Frequency-Structured Hamiltonian Neural Network for Multi-Timescale Dynamics

Yaojun Li* **Yulong Yang*** **Christine Allen-Blanchette**
Princeton University
{yl3425, yulong.yang, ca15}@princeton.edu

Abstract

Learning coupled fast and slow dynamics accurately remains challenging for structure-preserving neural models. Hamiltonian Neural Networks preserve a learned Hamiltonian, but their neural parameterizations are susceptible to spectral bias, which favors slowly varying components and can leave fast oscillations poorly resolved. We introduce the Frequency-Structured Hamiltonian Neural Network (FS-HNN), which organizes Hamiltonian learning across temporal frequency scales. FS-HNN learns Hamiltonian components from frequency-filtered views of observed trajectories and combines them nonlinearly into a single scalar Hamiltonian. This enables the model to represent interactions across timescales while conserving the learned energy in continuous time. For PDEs, FS-HNN represents Hamiltonian components as neural functionals and learns a spatial operator, using orthogonal projection to conserve the learned energy. Experiments on ODE and PDE benchmarks spanning nonlinear oscillators, coupled lattices, shallow water flows, and dispersive waves show that FS-HNN improves long-horizon rollout accuracy over existing baselines, with accompanying gains in physical-energy fidelity. Code is available at `https://github.com/CAB-Lab-Princeton/FS-HNN.git`.

1 Introduction

Multi-timescale dynamics arise throughout engineering and science. In nonlinear lattices, fast local oscillations coexist with slow energy transfer across modes [1], and shallow water flows exhibit rapid gravity waves alongside slower advective transport [2]. Predicting these systems over long horizons requires resolving both timescales and their interactions. When the timescales are widely separated, models must resolve rapid oscillations over intervals long enough to capture slow evolution. Neural models face an additional challenge from spectral bias, the tendency to learn low-frequency structure before high-frequency components [3, 4, 5]. Consequently, neural models may reproduce slowly varying behavior while misrepresenting the amplitude or phase of fast oscillations, potentially distorting energy transfer between modes and degrading long-horizon accuracy in coupled systems.

Hamiltonian Neural Networks incorporate physical structure into learned dynamics by conserving a learned Hamiltonian in continuous time [6, 7]. Symplectic methods impose geometric constraints through neural architectures or numerical integration schemes [8, 9, 10]. However, neither energy conservation nor symplecticity alone addresses the learning imbalance between fast and slow dynamics. Multiscale methods offer a complementary approach. Multirate and multigrid methods organize numerical computation across scales [11, 12, 13], and neural forecasting architectures model temporal structure at multiple resolutions [14, 15]. However, these forecasting architectures do not enforce Hamiltonian structure when combining information across temporal scales. Variational multirate methods preserve mechanical structure in slow–fast systems [16], but do not address learning Hamil-

* Equal contribution.

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

tonian representations from data. The remaining challenge is to learn and recombine information across interacting timescales while preserving Hamiltonian structure in the resulting dynamics.

We introduce the Frequency-Structured Hamiltonian Neural Network (FS-HNN), which organizes Hamiltonian learning across temporal frequency scales. FS-HNN constructs nested frequency-filtered views of observed trajectories that expose progressively coarser temporal structure while retaining shared low-frequency information. Each view provides supervision for a Hamiltonian component, and a learned nonlinear aggregator combines these components into a single scalar Hamiltonian that can represent cross-scale interactions. For ODE systems, the aggregated Hamiltonian generates dynamics through the canonical symplectic operator, conserving the learned Hamiltonian in continuous time. For PDE systems, the operator relating Hamiltonian gradients to the dynamics depends on the equation family, boundary conditions, and discretization [17, 18]. FS-HNN therefore represents Hamiltonian components as neural functionals, learns a convolutional spatial operator, and uses orthogonal projection to constrain the resulting dynamics to conserve the learned Hamiltonian.

We evaluate FS-HNN on ODE and PDE benchmarks spanning integrable, chaotic, and multiscale dynamics. FS-HNN achieves lower long-horizon rollout error than the evaluated structure-preserving and black-box baselines, with accompanying gains in physical-energy fidelity. Ablations show that distinct temporal scales and nonlinear Hamiltonian aggregation improve rollout accuracy. PDE ablations further demonstrate that frequency structure and energy preservation provide complementary improvements in prediction accuracy.

Our main contributions are threefold:

- We introduce a frequency-structured Hamiltonian parameterization that uses nested frequency-filtered trajectory views to learn dynamics across temporal scales.
- We develop a nonlinear aggregation scheme that combines these components into a single scalar Hamiltonian, allowing the model to represent cross-scale interactions while conserving the learned Hamiltonian in continuous time.
- We extend the framework to PDE discretizations using neural Hamiltonian functionals, a learned spatial operator, and orthogonal projection to constrain the learned dynamics to conserve the Hamiltonian.

2 Related Work

Conservation constraints in deep learning. Traditional system identification algorithms achieve accurate dynamics through prescribing model classes using prior knowledge [19, 20, 21, 22, 23, 24, 25]. Black-box learning architectures circumvent this restriction by trading interpretability for expressivity [26, 27, 28]. Recent methods seek to strike a balance between system identification and black-box learning by embedding physical laws as inductive biases. Hamiltonian [6, 7, 29, 30, 31, 32] and Lagrangian [33, 34, 35] based neural networks parameterize the system energy and recover dynamics through the corresponding equations of motion. However, encoding energy structure alone does not prevent numerical errors from accumulating over long rollouts [9, 8]. Subsequent work strengthens these inductive biases either by enforcing configuration-space structure [36, 32, 37, 38, 39, 40], which requires prior knowledge of system geometry, or by enforcing symplecticity in the dynamics. Symplecticity is typically achieved either by rolling out trajectories with symplectic integrators [9, 8, 41] or by learning a symplectic map directly [10, 42, 43, 44]. These refinements govern how the learned dynamics are constrained and integrated, but they do not by themselves resolve the learning imbalance between fast and slow dynamics. Among these, only Chen et al. [9] explicitly engages with stiff systems, studying contact discontinuities rather than the slow–fast coupling on which FS-HNN focuses. FS-HNN addresses this imbalance by structuring the Hamiltonian across temporal frequency scales, learning a separate component from each of a set of nested frequency-filtered trajectory views.

Multiscale modeling of dynamical systems. To mitigate the difficulty posed by stiff systems, we draw inspiration from numerical analysis, where multi-scale and multi-resolution methods have long been used to improve performance and stability for stiff dynamics [11, 12, 13]. Deep learning has adopted analogous ideas: black-box architectures decompose time series through autocorrelation [14], seasonal structure [15], or mixture-of-experts routing across timescales [45, 46, 47, 48] to improve long-horizon forecasting. These statistical decompositions, however, do not explicitly preserve the physical structure of the underlying dynamics. More physically informed work has

incorporated fast–slow structure into NeuralODEs for singularly perturbed systems [49], but focuses on dissipative fast dynamics rather than the persistent fast oscillations typical of Hamiltonian systems. From a structure-preserving perspective, Ober-Blöbaum et al. [16] show that slow–fast Lagrangian systems admit variational decompositions whose components retain Lagrangian dynamics. Their analysis is theoretical, however, and does not address how to realize such decompositions in neural architectures. FS-HNN addresses this gap by combining multi-timescale supervision with a structure-preserving Hamiltonian parameterization, allowing cross-scale interactions to be learned within a single conservative model.

3 Background

FS-HNN incorporates Hamiltonian energy conservation and frequency-structured learning to model coupled multi-timescale dynamics. The skew-symmetric structure of Hamilton's equations motivates the energy-conserving formulation of FS-HNN, while differences in characteristic frequencies motivate its nested frequency-filtered trajectory views.

Symplectic structure and energy conservation. Hamiltonian mechanics reframes classical dynamics in the language of energy [50, 51]. Instead of tracking the forces acting on a system, it summarizes the entire dynamics in a single scalar function, the Hamiltonian $\mathcal{H}(\mathbf{q}, \mathbf{p})$, which maps generalized coordinates $\mathbf{q}$ and momenta $\mathbf{p}$ to total energy. The gradients of $\mathcal{H}$ with respect to coordinates and momenta then specify how the system evolves, through Hamilton's equations,

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

Letting $\mathbf{z} = (\mathbf{q}, \mathbf{p})$ denote a point in the phase space of a dynamical system, Hamilton's equations take the symplectic form

$$\dot{\mathbf{z}} = \mathbf{J}\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z}), \qquad \mathbf{J} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{bmatrix}, \tag{2}$$

where $\mathbf{J}$ is the skew-symmetric canonical symplectic matrix. Energy conservation then follows by differentiating $\mathcal{H}$ along a trajectory and using the skew-symmetry of $\mathbf{J}$, which gives

$$\frac{d\mathcal{H}}{dt} = \nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})^{\intercal}\dot{\mathbf{z}} = \nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})^{\intercal}\mathbf{J}\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z}) = 0. \tag{3}$$

Multi-timescale Hamiltonian systems. Many Hamiltonian systems of practical interest evolve on multiple, widely separated characteristic timescales [52, 53]. For a stiff Hamiltonian system, let $\varepsilon \ll 1$ be the stiffness parameter of the fast potential, and suppose the phase space splits into slow coordinates $(\mathbf{q}^{\mathrm{s}}, \mathbf{p}^{\mathrm{s}})$ and fast coordinates $(\mathbf{q}^{\mathrm{f}}, \mathbf{p}^{\mathrm{f}})$. The Hamiltonian then takes the form

$$\mathcal{H}\left(\mathbf{q}^{\mathrm{s}}, \mathbf{p}^{\mathrm{s}}, \mathbf{q}^{\mathrm{f}}, \mathbf{p}^{\mathrm{f}}\right) = \frac{1}{2}\left(\mathbf{p}^{\mathrm{s}}\right)^{\intercal}\mathbf{M}_{\mathrm{s}}^{-1}\mathbf{p}^{\mathrm{s}} + \frac{1}{2}\left(\mathbf{p}^{\mathrm{f}}\right)^{\intercal}\mathbf{M}_{\mathrm{f}}^{-1}\mathbf{p}^{\mathrm{f}} + V\left(\mathbf{q}^{\mathrm{s}}, \mathbf{q}^{\mathrm{f}}\right) + \frac{1}{\varepsilon}\bar{W}\left(\mathbf{q}^{\mathrm{f}}\right), \tag{4}$$

where the stiff potential $1/\varepsilon \bar{W}(\mathbf{q}^{\mathrm{f}})$ drives fast oscillations and $V\left(\mathbf{q}^{\mathrm{s}}, \mathbf{q}^{\mathrm{f}}\right)$ couples the two subsystems. The characteristic frequency of the fast subsystem follows from a leading-order analysis of the stiff potential. Expanding $\bar{W}$ to second order about a local minimizer $\mathbf{q}^{\mathrm{f}}_*$ and writing $\boldsymbol{\eta} = \mathbf{q}^{\mathrm{f}} - \mathbf{q}^{\mathrm{f}}_*$, the fast equation of motion reduces to

$$\ddot{\boldsymbol{\eta}} \approx -\varepsilon^{-1}\mathbf{M}_{\mathrm{f}}^{-1}\nabla^2\bar{W}\left(\mathbf{q}^{\mathrm{f}}_*\right)\boldsymbol{\eta}, \tag{5}$$

so the fast modes oscillate with characteristic frequency $\omega \sim \varepsilon^{-1/2}$ while the slow modes evolve on $\mathcal{O}(1)$ timescales.

Coupling and frequency structure. In the idealized case of vanishing coupling (i.e., when V depends only on $\mathbf{q}^{\mathrm{s}}$), the Hamiltonian separates additively into independent slow and fast components,

$$\mathcal{H}\left(\mathbf{q}^{\mathrm{s}}, \mathbf{p}^{\mathrm{s}}, \mathbf{q}^{\mathrm{f}}, \mathbf{p}^{\mathrm{f}}\right) = \mathcal{H}_{\mathrm{s}}\left(\mathbf{q}^{\mathrm{s}}, \mathbf{p}^{\mathrm{s}}\right) + \mathcal{H}_{\mathrm{f}}\left(\mathbf{q}^{\mathrm{f}}, \mathbf{p}^{\mathrm{f}}\right). \tag{6}$$

However, in general V couples slow and fast coordinates and the additive form holds only as a leading-order approximation [16, 54, 55]. The timescale separation nonetheless has a direct frequency-domain consequence. Because slow and fast components evolve at characteristically different rates, multi-timescale trajectories concentrate their spectral power in well-separated regions of the temporal frequency spectrum, even when coupling prevents a clean modal decomposition. FS-HNN leverages this observation through a frequency-structured parameterization based on nested frequency-filtered trajectory views, with Hamiltonian components recombined nonlinearly to account for the coupling. Additional details are provided in Appendix A.2.

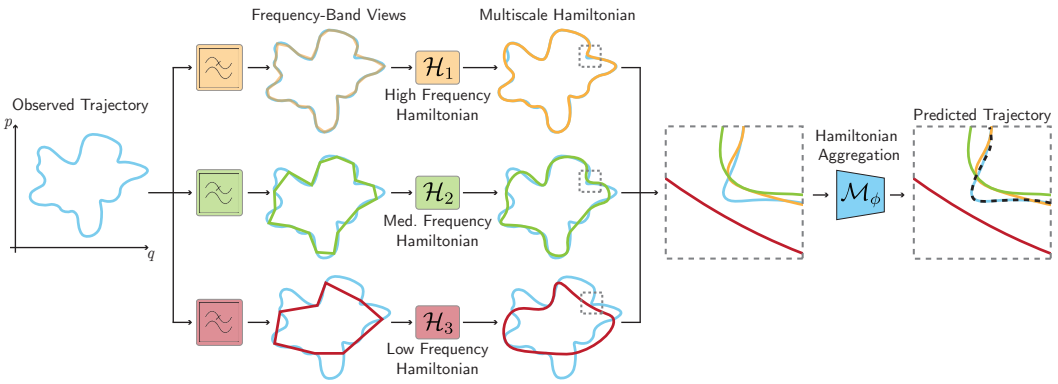


Figure 1: **Frequency-Structured Hamiltonian Neural Network.** An observed trajectory $\mathbf{z}_{\text{train}}$ is filtered into a nested family of low-pass views $\mathbf{z}^{(k)}$ at decreasing cutoffs, which expose progressively coarser temporal structure. Each view provides supervision for a per-band Hamiltonian $\mathcal{H}_{\theta_k}^{(k)}$ that models the dynamics resolved at its cutoff. The aggregator $\mathcal{M}$ combines the per-band Hamiltonians into a single scalar Hamiltonian $\mathcal{H}_{\text{agg}}$, whose canonical dynamics conserve the learned energy and generate the predicted trajectory.

4 Method

The Frequency-Structured Hamiltonian Neural Network (FS-HNN) organizes Hamiltonian learning across temporal frequency scales, as illustrated in Figure 1. For ODE systems, nested frequency-filtered views of observed trajectories provide supervision for individual Hamiltonian components. A learned nonlinear aggregator combines these components into a single scalar Hamiltonian, whose canonical dynamics conserve the learned Hamiltonian in continuous time (Section 4.1). For PDE systems, FS-HNN represents the Hamiltonian components as neural functionals, learns a problem-dependent spatial operator, and uses orthogonal projection to constrain the predicted dynamics to conserve the learned Hamiltonian (Section 4.2).

4.1 Frequency-Structured Parameterization of the Hamiltonian

FS-HNN incorporates the frequency structure of multi-timescale dynamics into its Hamiltonian parameterization. Rather than parameterizing the Hamiltonian with a single network, FS-HNN learns components associated with nested frequency-filtered trajectory views and combines them into a single scalar Hamiltonian. Nonlinear aggregation allows FS-HNN to represent interactions between coupled slow and fast modes.

ODE Frequency-Filtered Views. We construct K nested low-pass-filtered views of the training trajectory $\mathbf{z}_{\text{train}}$ using decreasing cutoff frequencies $\omega_1 > \omega_2 > \ldots > \omega_K$, with ω_1 equal to the Nyquist frequency:

$$\mathbf{z}^{(k)} = \mathcal{F}^{-1}\left(\mathbf{1}_{\{|\omega| \leq \omega_k\}} \cdot \mathcal{F}\left(\mathbf{z}_{\text{train}}\right)\right), \tag{7}$$

where $\mathcal{F}$ denotes the discrete Fourier transform and $\mathbf{1}_{\{|\omega| \leq \omega_k\}}$ selects frequencies below ω_k. where $\mathcal{F}$ denotes the discrete Fourier transform and $\mathbf{1}_{\{|\omega| \leq \omega_k\}}$ retains frequency components at or below ω_k. The resulting views have nested spectral support: $\mathbf{z}^{(1)}$ retains the full spectrum, while each subsequent view removes additional high-frequency components.

The default configuration uses $K = 3$ nested views with cutoff frequencies of $\{1.0, 0.5, 0.25\}$ times the Nyquist frequency. We also evaluate spectrum-adaptive cutoffs to account for differences in spectral content across systems. For spectra with a distinct dominant peak at normalized frequency f_{peak}, we use $\{1, f_{\text{peak}}/2, f_{\text{peak}}/20\}$. For broadband spectra, we use $\{1, q_{0.99}, q_{0.90}\}$, where q_α denotes the normalized frequency below which a fraction α of the total spectral power lies.

Per-Band Hamiltonian. Each frequency-filtered trajectory is used to train a separate Hamiltonians $\mathcal{H}_{\theta_k}^{(k)}$ parameterized by MLPs, which learns a Hamiltonian approximation to the dynamics resolved by its filtered trajectory. Because the views are nested rather than disjoint, each branch captures the dynamics resolved at its cutoff while retaining all lower-frequency content.

Aggregation. A natural starting point for combining K per-band Hamiltonians into a single dynamical system is the additive form

$$\mathcal{H}_{\mathrm{add},\theta}\left(\mathbf{z}\right) = \sum_{k=1}^{K} \mathcal{H}_{\theta_k}^{(k)}\left(\mathbf{z}^{(k)}\right), \tag{8}$$

which is exact when the system admits a clean slow-fast decomposition (Appendix A.2) and the subsystems decouple. Coupling through V breaks this additivity in general, so we parameterize the aggregated Hamiltonian $\mathcal{H}_{\mathrm{agg},\theta}\left(\mathbf{z}\right)$ as a learned scalar function of the per-band Hamiltonians,

$$\mathcal{H}_{\mathrm{agg},\theta}\left(\mathbf{z}\right) = \mathcal{M}_{\phi}\left(\mathcal{H}_{\theta_1}^{(1)}\left(\mathbf{z}^{(1)}\right), \ldots, \mathcal{H}_{\theta_K}^{(K)}\left(\mathbf{z}^{(K)}\right)\right), \tag{9}$$

where $\mathcal{M}_{\phi}$ is a small MLP. The additive case is recovered when $\mathcal{M}_{\phi} = \sum_k$, yielding the additive case of $\mathcal{H}_{\mathrm{agg},\theta} = \mathcal{H}_{\mathrm{add},\theta}$. Outside that limit, $\mathcal{M}_{\phi}$ captures non-additive interactions among the per-band components. The aggregated Hamiltonian then generates the dynamics through the canonical symplectic form

$$\dot{\mathbf{z}} = \mathbf{J}\nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}\left(\mathbf{z}\right). \tag{10}$$

Structure Preservation. Although the aggregation function $\mathcal{M}_{\phi}$ is nonlinear, $\mathcal{H}_{\mathrm{agg},\theta}$ remains a scalar Hamiltonian, and the dynamics factor through the canonical skew-symmetric $\mathbf{J}$. The conservation argument of Section 3 then carries over directly, giving

$$\frac{d}{dt}\mathcal{H}_{\mathrm{agg},\theta}\left(\mathbf{z}\left(t\right)\right) = \nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}^{\intercal}\mathbf{J}\nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta} = 0, \tag{11}$$

along trajectories. The learned aggregation therefore enriches the functional form of the Hamiltonian relative to the additive decomposition without breaking Hamiltonian structure or energy conservation in the continuous-time limit.

Training. FS-HNN is trained by minimizing the standard HNN loss function

$$\mathcal{L} = \left\|\dot{\mathbf{z}} - \mathbf{J}\nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}\left(\mathbf{z}\right)\right\|^2, \tag{12}$$

on the unfiltered training trajectory. The per-band Hamiltonian $\mathcal{H}_{\theta_k}^{(k)}$ and aggregation function $\mathcal{M}_{\phi}$ are trained jointly through $\mathcal{L}$.

4.2 Extension to PDE Systems

The construction in Section 4.1 leans the fact that the symplectic operator $\mathbf{J}$ is fixed, finite-dimensional, and skew-symmetric by construction. As a result $\langle \nabla\mathcal{H}, \mathbf{J}\nabla\mathcal{H}\rangle = 0$ holds automatically and the per-band Hamiltonians fully determine the dynamics. In the PDE setting, the state $\mathbf{z}$ is a continuous field, and in many discretizations this fixed canonical structure is no longer available. The corresponding operator becomes a differential object (e.g., ∂_x) whose form depends on the equation family, boundary conditions, and discretization scheme [17, 18], and is rarely available in closed form. We therefore learn its action and recover energy conservation through projection rather than by construction. The frequency-filtered views and the aggregation of Section 4.1 carry over unchanged, while each per-band Hamiltonian becomes a neural functional.

Convolutional Spatial Operator. Rather than constructing $\mathbf{J}$ analytically, we learn its action directly through a spatial operator $\mathbf{J}_{\theta}$ that maps the gradient of the aggregated Hamiltonian to the rate of change of the state,

$$\Delta\tilde{\mathbf{z}} = \mathbf{J}_{\theta}\left(\nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}\left(\mathbf{z}\right)\right). \tag{13}$$

This avoids explicit matrix representations of the operator. We implement $\mathbf{J}_{\theta}$ as a residual CNN [56], since local convolutional kernels naturally capture neighborhood coupling on discretized spatial grids and emulate differential operators.

Energy Preservation via Orthogonal Projection. Unlike the canonical $\mathbf{J}$, the learned operator $\mathbf{J}_{\theta}$ is not skew-symmetric by construction, so the conservation identity of the canonical setting no longer holds automatically. Rather than constraining $\mathbf{J}_{\theta}$ to a prescribed skew-symmetric or Poisson form, we enforce energy conservation directly by projecting its output $\Delta\tilde{\mathbf{z}}$ onto the subspace orthogonal to $\nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z})$,

$$\Delta\mathbf{z} = \Delta\tilde{\mathbf{z}} - \frac{\langle \nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z}), \Delta\tilde{\mathbf{z}}\rangle}{\langle \nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z}), \nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z})\rangle + \epsilon}\nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z}), \tag{14}$$

where $\Delta\mathbf{z} \approx \dot{\mathbf{z}}$ is the projected vector-field estimate and $\epsilon \ll 1$ provides numerical stability. Writing $\mathbf{g} = \nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z})$, the projection satisfies

$$\langle \mathbf{g}, \Delta\mathbf{z} \rangle = \frac{\epsilon}{\|\mathbf{g}\|^2+\epsilon} \langle \mathbf{g}, \Delta\tilde{\mathbf{z}} \rangle. \tag{15}$$

Thus, for $\epsilon = 0$ and $\mathbf{g} \neq 0$, the projected vector field is exactly orthogonal to the Hamiltonian gradient, giving

$$\frac{d}{dt}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z}) = \langle \nabla_{\mathbf{z}}\mathcal{H}_{\mathrm{agg},\theta}(\mathbf{z}), \dot{\mathbf{z}} \rangle = 0, \tag{16}$$

when the continuous-time dynamics follow the projected field. A positive ϵ leaves the explicit conservation residual shown above. Moreover, away from critical points, exact orthogonality implies the existence of a skew-symmetric operator $\mathbf{J}(\mathbf{z})$ satisfying $\Delta\mathbf{z} = \mathbf{J}(\mathbf{z})\mathbf{g}$, placing the projected dynamics within the skew-gradient class associated with energy-preserving integration [57, 58]. Thus, in the $\epsilon = 0$ continuous-time limit, the construction conserves the learned Hamiltonian without requiring $\mathbf{J}_\theta$ itself to be skew-symmetric. It does not, however, guarantee symplecticity or exact conservation after numerical time discretization (Appendix A.3). Importantly, $\mathcal{H}_{\mathrm{agg},\theta}$ is a learned invariant and need not equal the physical energy. Tables 7 and 10 therefore evaluate the analytic energies of the benchmark systems on the predicted states.

PDE Frequency-Band Partitions. We apply temporal filtering (Section 4.1) to PDE systems, producing nested views that retain progressively less fine-scale temporal variation. The use of overlapping and nested views is analogous to multiscale techniques in PDE analysis, including multiresolution analysis with nested approximation spaces [59], multigrid methods across mesh resolutions [60], and variational coupling of resolved and unresolved scales [61]. Similarly, our construction preserves interactions across scales instead of treating frequency components as independent.

Per-Band Hamiltonian Functional. Whereas each ODE Hamiltonian $\mathcal{H}_{\theta_k}^{(k)}$ is a scalar function represented by an MLP, a PDE Hamiltonian is a functional of a field. asWe therefore represent each $\mathcal{H}_{\theta_k}^{(k)}$ with a DeepONet [28], mapping a field representation to a scalar Hamiltonian. The resulting neural functionals are aggregated and differentiated to construct the structure-preserving PDE dynamics.

5 Experiments

The experiments examine how the frequency-structured Hamiltonian components developed above affect long-horizon rollout accuracy and physical-energy behavior in ODE and PDE systems. On ODE benchmarks ranging from integrable mechanics to chaotic and multiscale dynamics, FS-HNN is compared with black-box and structure-preserving baselines. On PDE system, the extension with its learned operator and projection is compared with structure-preserving and neural-operator baselines.

5.1 ODE Systems

The four ODE benchmarks range from simple periodic motion to systems in which fast and slow dynamics are explicitly coupled. The single pendulum has no separation of timescales, while the double pendulum introduces chaos and sensitivity to initial conditions. The Fermi–Pasta–Ulam–Tsingou (FPUT) chain [1] combines fast oscillations with slow energy transfer across modes. The fast–slow oscillator proposed in [62] couples a unit-frequency oscillator to a slow oscillator of frequency $\varepsilon = 10^{-2}$ through a nonlinear potential. The fast mode of this system lies at only 0.032 of the Nyquist frequency, so FS-HNN uses the spectrum-selected cutoffs $\{1.0, 0.016, 0.0016\}$ of Section 4.1 on this system to align the frequency hierarchy with its spectral scales. Models are trained on the first 10 steps and rolled out for 1000 steps. We compare against a black-box MLP, HNN [6], and SympNet [10]. Rollout error for FS-HNN and baselines are reported in Table 1 and Figure 2.

Rollout accuracy. FS-HNN attains the lowest rollout MSE on all four systems (Table 1). On the single pendulum, the phase-space portrait of FS-HNN remains stable over the full rollout while the MLP drifts visibly (Figure 2a), so the frequency-structured parameterization retains stable rollouts even when pronounced timescale separation is absent. On the chaotic double pendulum, HNN and SympNet diverge from the reference trajectory within the first few hundred steps, whereas FS-HNN tracks it for several hundred steps before phase error accumulates (Figure 2b). The two

Table 1: **Rollout accuracy for ODE systems.** We report the rollout MSE of FS-HNN, MLP, HNN, and SympNet on the pendulum, double pendulum, Fermi–Pasta–Ulam–Tsingou, and fast–slow oscillator systems. FS-HNN achieves the lowest prediction error on all four benchmarks.

Model	Single Pend.	Double Pend.	FPUT	Fast–Slow
FS-HNN	$(\mathbf{3.25}\pm\mathbf{1.02})\times\mathbf{10^{-7}}$	$(\mathbf{3.73}\pm\mathbf{1.50})\times\mathbf{10^{-2}}$	$(\mathbf{2.10}\pm\mathbf{1.03})\times\mathbf{10^{-5}}$	$(\mathbf{1.45}\pm\mathbf{0.63})\times\mathbf{10^{-2}}$
MLP	$(3.51\pm1.11)\times10^{-4}$	$(4.99\pm1.87)\times10^{-1}$	$(3.28\pm1.31)\times10^{-2}$	$(3.30\pm0.18)\times10^{-2}$
HNN	$(9.27\pm1.06)\times10^{-4}$	$(6.39\pm3.11)\times10^{-1}$	$(1.06\pm0.38)\times10^{-2}$	$(5.75\pm2.80)\times10^{-2}$
SympNet	$(3.26\pm1.85)\times10^{-6}$	$(4.56\pm1.58)\times10^{-1}$	$(3.43\pm0.05)\times10^{-3}$	$(3.48\pm0.06)\times10^{-2}$

coupled-timescale systems test the frequency structure directly. On FPUT, FS-HNN retains the slow energy-exchange envelope that the baselines flatten or distort (Figure 2c), and on the fast–slow oscillator it follows the slow drift of the coupled oscillator while the baselines lag behind or depart from the reference (Figure 2d). These results support the use of frequency-structured Hamiltonian components to capture interacting fast and slow dynamics.

Energy preservation across systems. The nonlinear aggregator retains the Hamiltonian structure described in Section 4.1, so the learned Hamiltonian is conserved in continuous time. Because this learned invariant need not coincide with the physical energy, each system's analytic energy is evaluated on the predicted states and compared with that of the reference trajectory, separately from the full-state error. FS-HNN has the lowest reported energy error across all four systems over the 1000-step rollout in Table 7 in Appendix C.1.

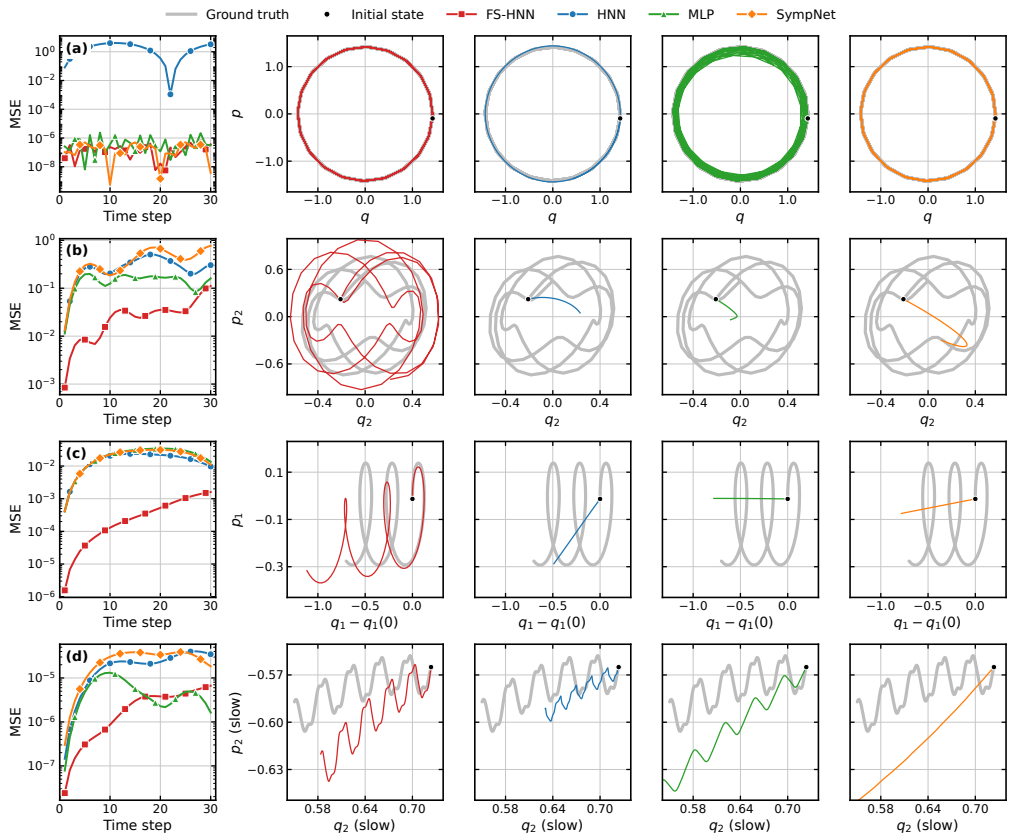


Figure 2: **ODE systems rollout comparison.** Rollout MSE and phase-space portraits of FS-HNN, HNN, MLP, and SympNet on the **(a)** single pendulum, **(b)** double pendulum, **(c)** Fermi–Pasta–Ulam–Tsingou chain, and **(d)** fast–slow oscillator, shown in its slow coordinates. FS-HNN attains the lowest rollout MSE on all four systems (Table 1).

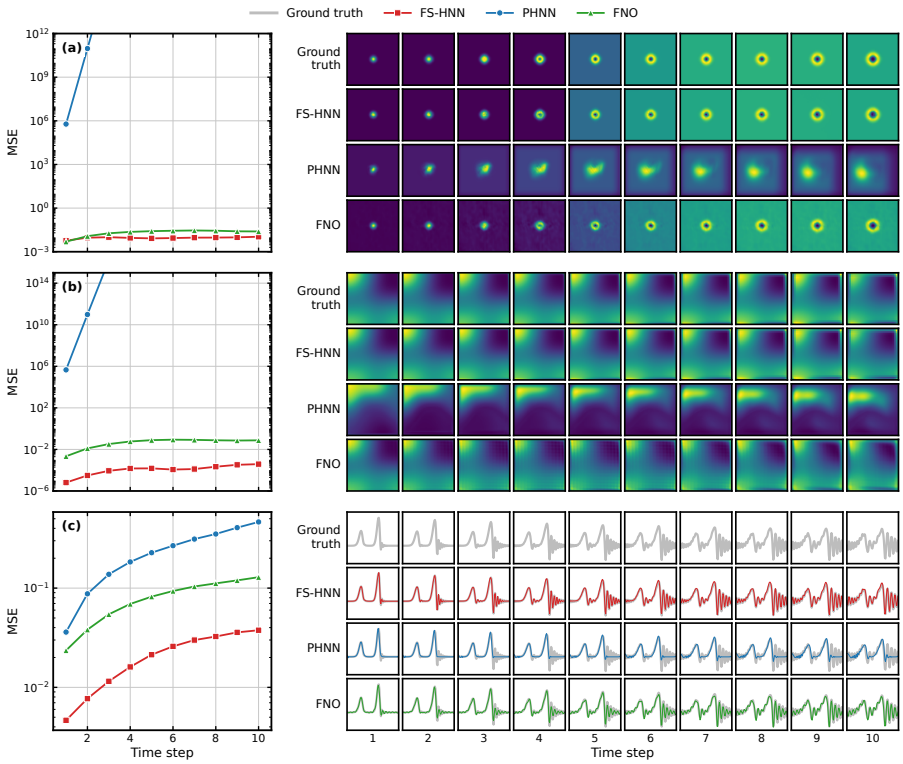


Figure 3: **PDE system rollout comparison.** We compare FS-HNN, PHNN, and FNO on **(a)** SWE with Gaussian-pulse initialization, **(b)** SWE with random initialization, and **(c)** the KdV equation. FS-HNN most closely reproduces the reference fields and achieves the lowest rollout error.

5.2 PDE Systems

Section 4.2 extends FS-HNN to PDEs through a learned spatial operator $\mathbf{J}_\theta$ and orthogonal projection. We evaluate this construction on the 2D shallow water equations (SWE) [2] with Gaussian-pulse and random initializations, and on the one-dimensional Korteweg–de Vries (KdV) equation. The SWE configurations are inviscid and unforced, so they conserve energy in the continuous setting [63]. The random initializations produce more complex spatial structure than the localized Gaussian pulse and are typically harder to learn [64]. KdV follows the two-soliton setup of Eidnes and Lye [44], with $u_t + 6uu_x - u_{xxx} = 0$ solved on the periodic domain $[0, 20]$. Data generation for both systems is detailed in Appendices B.5 and B.7. Models are trained on the first 5 steps and rolled out for the subsequent 50 steps. We compare with PHNN [44] as the closest structure-preserving baseline, together with Fourier Neural Operator (FNO) [27] and Energy-Consistent Neural Operator (ENO) [65]. Rollout accuracy is reported in Table 2 and Figure 3..

Rollout accuracy. Across the systems studied, FS-HNN attains the lowest rollout MSE on all three benchmarks (Table 2). Under both SWE initializations, FS-HNN closely reproduces the reference

Table 2: **Rollout accuracy for PDE systems.** We report the rollout MSE of FS-HNN, ENO, PHNN, and FNO on SWE with Gaussian-pulse and random initial conditions and on the KdV equation. The KdV entries come from single training runs on a shared train/validation split and are averaged over 20 held-out trajectories. FS-HNN achieves the lowest prediction error on all three benchmarks.

Model	SWE Pulse	SWE Random	KdV
FS-HNN	$(\mathbf{3.67}\pm\mathbf{0.54})\times\mathbf{10^{-4}}$	$(\mathbf{2.86}\pm\mathbf{0.13})\times\mathbf{10^{-5}}$	$(\mathbf{4.62}\pm\mathbf{4.05})\times\mathbf{10^{-2}}$
ENO	$(4.41\pm2.68)\times10^{-3}$	$(6.88\pm2.27)\times10^{-2}$	$(7.68\pm3.89)\times10^{-1}$
PHNN	$(8.57\pm3.91)\times10^{-2}$	$(4.65\pm2.16)\times10^{-2}$	$(2.97\pm1.65)\times10^{-1}$
FNO	$(5.31\pm1.17)\times10^{-3}$	$(2.19\pm0.28)\times10^{-2}$	$(1.15\pm0.86)\times10^{-1}$

Table 3: **Ablation on nonlinear aggregation.** We compare a linear and MLP aggregation. Nonlinear aggregation improves performance on the main nonlinear ODE benchmarks.

System	Linear	MLP
Pendulum	1.01×10^{-6}	$\mathbf{3.25 \times 10^{-7}}$
Double Pendulum	6.99×10^{-2}	$\mathbf{3.73 \times 10^{-2}}$
FPUT	3.47×10^{-3}	$\mathbf{2.10 \times 10^{-5}}$
Fast–Slow	1.26×10^{0}	$\mathbf{1.45 \times 10^{-2}}$

fields (Figure 3a,b). Its advantage over PHNN is larger for the random initializations, suggesting that the learned spatial operator remains effective as the field dynamics become more complex. On KdV, FS-HNN reproduces the soliton interaction and the dispersive tail (Figure 3c), so the PDE formulation also captures nonlinear wave interactions and dispersive evolution beyond the shallow water setting. These results show that FS-HNN extends frequency-structured Hamiltonian learning to PDE systems without requiring their dynamics operator in closed form.

Energy behavior across systems. The projection in Section 4.2 is designed to preserve the learned Hamiltonian, which need not coincide with the physical energy. We therefore evaluate the analytic SWE energy and KdV Hamiltonian on the predicted fields and compare them with the reference trajectories. FS-HNN follows the reference energy more closely than PHNN and FNO across the PDE benchmarks, including the small numerical energy variation in the SWE data (Figure 4 and Table 10 in Appendix C.2). As in the ODE setting, FS-HNN rollouts remain faithful to the energy of each system as well as to its state.

5.3 Ablation Studies

The ablations below isolate the components of FS-HNN behind these gains. In the ODE setting they examine the aggregator, the use of distinct temporal scales, the Hamiltonian structure, and the cutoffs, and in the PDE setting they separate the contributions of frequency structure and energy preservation.

Nonlinear aggregation. Table 3 compares the MLP aggregator $\mathcal{M}_\phi$ with linear aggregation of the same per-band Hamiltonians. Nonlinear aggregation lowers rollout error on all ODE benchmarks, with the largest gains on FPUT and the fast–slow oscillator, where fast and slow dynamics are explicitly coupled. This is consistent with the motivation for $\mathcal{M}_\phi$ in Section 4.1, since coupling breaks the additivity of the per-band Hamiltonians.

Distinct temporal scales. Table 4 compares FS-HNN with controls that keep its multi-branch architecture and aggregator but filter every branch at the same cutoff, so that capacity is matched and only the distinct temporal scales are removed. These controls underperforms FS-HNN on all four systems, so the improvement comes from combining views at distinct cutoffs rather than from added capacity or the aggregation network.

Hamiltonian structure. Applying the same frequency-filtered views and aggregation to an unconstrained NeuralODE [26] isolates the role of Hamiltonian structure. The combined NeuralODE gives errors similar to its single-scale version on the pendulum and FPUT and higher errors on the double pendulum and fast–slow oscillator (Table 5), while FS-HNN benefits from combining its components. Each FS-HNN branch contributes a scalar Hamiltonian and aggregation yields a single Hamiltonian with canonical dynamics, whereas NeuralODE combines unconstrained vector fields. Together with

Table 4: **Ablation on distinct temporal scales.** All control models use the same multi-branch architecture and aggregation module, but all branches operate at the same temporal interval. The last row is the full multiscale FS-HNN using multiple temporal intervals.

Setting	Pendulum	Double Pendulum	FPUT	Fast–slow
Low frequency only	2.61×10^{-6}	3.83×10^{-2}	2.78×10^{-3}	1.25×10^{0}
Mid frequency only	3.53×10^{-3}	7.23×10^{-2}	3.42×10^{-3}	1.80×10^{2}
High frequency only	8.95×10^{-1}	3.59×10^{-1}	3.47×10^{-3}	2.00×10^{2}
FS-HNN	$\mathbf{3.25 \times 10^{-7}}$	$\mathbf{3.73 \times 10^{-2}}$	$\mathbf{2.10 \times 10^{-5}}$	$\mathbf{1.45 \times 10^{-2}}$

Table 5: **NeuralODE trained with frequency-filtered inputs.** We compare a single-scale NeuralODE with a combined NeuralODE using the same multi-timescale branch aggregation strategy. The combined model gives similar errors on the pendulum and FPUT systems and higher errors on the double pendulum and fast–slow oscillator.

System	Single	Combined
Pendulum	1.20×10^{-9}	$\mathbf{1.16 \times 10^{-9}}$
Double Pendulum	$\mathbf{1.66 \times 10^{-2}}$	1.52×10^{-1}
FPUT	3.47×10^{-3}	$\mathbf{3.44 \times 10^{-3}}$
Fast–slow oscillator	$\mathbf{4.07 \times 10^{-2}}$	7.83×10^{-2}

the previous ablation, this comparison supports combining information across temporal scales at the level of the Hamiltonian rather than the vector field.

Number of views and cutoff placement. The number of nested views K and their cutoffs determine which temporal scales the views expose. Tables 8 and 9 in Appendix C.1 vary K and compare fixed cutoffs with cutoffs selected from the observed spectrum. Adding nested views improves accuracy on FPUT, whereas on the fast–slow oscillator, aligning the cutoffs with the system's spectral scales is more effective than adding views with cutoffs obtained by repeated halving. Spectrum-selected cutoffs with $K = 3$ give the lowest error of all tested configurations on both systems. The same choice also outperforms the fixed cutoffs on the remaining ODE benchmarks and on KdV.

Frequency structure and energy preservation. The PDE construction combines frequency structure, through the filtered views and aggregation, with energy preservation, through the learned operator and projection. Either ingredient could account for the gains in Table 2. Table 11 in Appendix C.2 separates them on random-initialization SWE and KdV with two controls, each removing one ingredient. The single-scale conservative control removes the frequency structure and keeps the DeepONet Hamiltonian, learned $\mathbf{J}_\theta$, and projection. The non-conservative DeepONet removes energy preservation and keeps the frequency structure. FS-HNN gives lower rollout error than either control on both benchmarks, so its PDE gains come from combining frequency structure with energy preservation rather than from either ingredient alone.

6 Conclusion and Discussion

Hamiltonian neural networks build energy conservation into the learned dynamics but inherit spectral bias, which can leave fast oscillations poorly resolved. FS-HNN addresses this by organizing the Hamiltonian around the frequency structure of the observed trajectories. Nested frequency-filtered views supervise per-band Hamiltonian components, and a learned nonlinear aggregator combines them into a single scalar Hamiltonian whose dynamics capture interactions between fast and slow modes and conserve the learned energy by construction. For PDEs, FS-HNN represents the components as neural functionals, learns the action of the dynamics operator, and conserves the learned energy through orthogonal projection. Across the ODE and PDE benchmarks, FS-HNN attains lower rollout error than black-box and structure-preserving baselines while remaining faithful to the physical energy of each system. Ablations attribute these gains to combining distinct temporal scales at the level of a scalar Hamiltonian rather than to added capacity, and show that frequency structure and energy preservation are complementary for PDEs. These results suggest that structure-preserving models benefit from dedicating part of the model to coarser temporal structure rather than learning from the full-spectrum trajectory alone.

Scope and limitations. The PDE experiments focus on smooth conservative dynamics on fixed spatial grids. Shock-dominated or strongly dissipative regimes require entropy-aware or non-conservative formulations. Conservation of the learned Hamiltonian holds in continuous time, so exact conservation after time discretization depends on the integrator, and the projected PDE dynamics are not guaranteed to be symplectic. Filtering a Hamiltonian trajectory does not in general produce a Hamiltonian flow, so each branch fits a Hamiltonian approximation, and the structure guarantees apply to the aggregated dynamics rather than to the individual branches. The projection also adds computational cost through gradient evaluations. Extending the implementation across spatial resolutions and boundary conditions remains future work.

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 acknowledge the NeurIPS 2026 reviewers and area chair for their thoughtful feedback. The authors declare no competing interests.

References

[1] Enrico Fermi, P Pasta, Stanislaw Ulam, and Mary Tsingou. Studies of the nonlinear problems. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 1955.

[2] Cornelis Boudewijn Vreugdenhil. *Numerical methods for shallow-water flow*, volume 13. Springer Science & Business Media, 2013.

[3] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

[4] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[5] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

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

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

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

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

[10] 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.

[11] Jan Mandel, Steve McCormick, and John Ruge. An algebraic theory for multigrid methods for variational problems. *SIAM Journal on Numerical Analysis*, 25(1):91–110, 1988.

[12] Jan Mandel and Seymour V Parter. On the multigrid f-cycle. *Applied mathematics and computation*, 37(1):19–36, 1990.

[13] Yoonsang Lee and Bjorn Engquist. Variable step size multiscale methods for stiff and highly oscillatory dynamical systems. *arXiv preprint arXiv:1304.4336*, 2013.

[14] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[15] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

[16] Sina Ober-Blöbaum, Theresa Wenger, Tobias Gail, and Sigrid Leyendecker. Variational multirate integrators. *arXiv preprint arXiv:2406.12991*, 2024.

[17] Thomas J Bridges. Multi-symplectic structures and wave propagation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 121, pages 147–190. Cambridge University Press, 1997.

[18] Thomas J Bridges and Sebastian Reich. Multi-symplectic integrators: numerical schemes for hamiltonian pdes that conserve symplecticity. *Physics Letters A*, 284(4-5):184–193, 2001.

[19] Nicholas Galioto and Alex A Gorodetsky. Bayesian identification of hamiltonian dynamics from symplectic data. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1190–1195. IEEE, 2020.

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

[21] 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.

[22] Jonathan P Epperlein, Bassam Bamieh, and Karl J Astrom. Thermoacoustics and the rijke tube: Experiments, identification, and modeling. *IEEE Control Systems Magazine*, 35(2):57–77, 2015.

[23] 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.

[24] 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.

[25] 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.

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

[27] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[28] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[29] Tom Bertalan, Felix Dietrich, Igor Mezić, and Ioannis G Kevrekidis. On learning hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12), 2019.

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

[31] Jiayin Liu, Yulong Yang, Vineet Bansal, and Christine Allen-Blanchette. Physically plausible multi-system trajectory generation and symmetry discovery. *arXiv preprint arXiv:2509.23003*, 2025.

[32] Yaofeng Desmond Zhong and Naomi Leonard. Unsupervised learning of lagrangian dynamics from images for prediction and control. *Advances in Neural Information Processing Systems*, 33:10741–10752, 2020.

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

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

[35] 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.

[36] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International conference on machine learning*, pages 3165–3176. PMLR, 2020.

[37] Justice Mason, Christine Allen-Blanchette, Nicholas Zolman, Elizabeth Davison, and Naomi Leonard. Learning interpretable dynamics from images of a freely rotating 3d rigid body. *arXiv preprint arXiv:2209.11355*, 2022.

[38] 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.

[39] Thai Duong, Abdullah Altawaitan, Jason Stanley, and Nikolay Atanasov. Port-hamiltonian neural ode networks on lie groups for robot dynamics learning and control. *IEEE Transactions on Robotics*, 2024.

[40] Philipp Horn, Veronica Saz Ulibarrena, Barry Koren, and Simon Portegies Zwart. A generalized framework of neural networks for hamiltonian systems. *Journal of Computational Physics*, 521:113536, 2025.

[41] Chen-Di Han, Bryan Glaz, Mulugeta Haile, and Ying-Cheng Lai. Adaptable hamiltonian neural networks. *Physical Review Research*, 3(2):023156, 2021.

[42] Joshua William Burby, Qi Tang, and Romit Maulik. Fast neural poincaré maps for toroidal magnetic fields. *Plasma Physics and Controlled Fusion*, 63(2):024001, 2020.

[43] 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.

[44] Sølve Eidnes and Kjetil Olsen Lye. Pseudo-hamiltonian neural networks for learning partial differential equations. *Journal of Computational Physics*, 500:112738, 2024.

[45] Yifan Hu, Peiyuan Liu, Peng Zhu, Dawei Cheng, and Tao Dai. Adaptive multi-scale decomposition framework for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17359–17367, 2025.

[46] Junaid Farooq, Danish Rafiq, Pantelis R Vlachas, and Mohammad Abid Bazaz. Refreshnet: learning multiscale dynamics through hierarchical refreshing. *Nonlinear Dynamics*, 112(16):14479–14496, 2024.

[47] Junjie Yao, Yuxiao Yi, Liangkai Hang, Weizong Wang, Yaoyu Zhang, Tianhan Zhang, Zhi-Qin John Xu, et al. Solving multiscale dynamical systems by deep learning. *Computer Physics Communications*, page 109802, 2025.

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

[49] Daniel A Serino, Allen Alvarez Loya, Joshua W Burby, Vitaliy Gyrya, and Qi Tang. Fast-slow neural networks for learning singularly perturbed dynamical systems. *Journal of Computational Physics*, 537:114090, 2025.

[50] Herbert Goldstein, Charles P. Poole, and John L. Safko. *Classical Mechanics*. Addison-Wesley, 3 edition, 2002.

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

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

[53] Jan A Sanders, Ferdinand Verhulst, and James Murdock. *Averaging methods in nonlinear dynamical systems*, volume 59. Springer, 2007.

[54] Ludwig Gauckler, Ernst Hairer, and Christian Lubich. Energy separation in oscillatory hamiltonian systems without any non-resonance condition. *Communications in Mathematical Physics*, 321(3):803–815, 2013.

[55] Miguel D Bustamante, Kevin Hutchinson, Yuri V Lvov, and Miguel Onorato. Exact discrete resonances in the fermi-pasta-ulam–tsingou system. *Communications in Nonlinear Science and Numerical Simulation*, 73:437–471, 2019.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[57] Robert I McLachlan, G Reinout W Quispel, and Nicolas Robidoux. Geometric integration using discrete gradients. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 357(1754):1021–1045, 1999.

[58] Elena Celledoni, Volker Grimm, Robert I McLachlan, David I McLaren, Dion O'Neale, Brynjulf Owren, and G Reinout W Quispel. Preserving energy resp. dissipation in numerical PDEs using the "Average Vector Field" method. *Journal of Computational Physics*, 231(20):6770–6789, 2012.

[59] Stephane G Mallat. Multiresolution approximations and wavelet orthonormal bases of l^2 (r). *Transactions of the American mathematical society*, 315(1):69–87, 1989.

[60] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.

[61] Thomas JR Hughes. Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer methods in applied mechanics and engineering*, 127(1-4):387–401, 1995.

[62] Valentin Duruisseaux, Joshua W Burby, and Qi Tang. Approximation of nearly-periodic symplectic maps via structure-preserving neural networks. *Scientific Reports*, 13(1):8351, 2023.

[63] Eitan Tadmor and Weigang Zhong. Energy-preserving and stable approximations for the two-dimensional shallow water equations. In *Mathematics and Computation, a Contemporary View: The Abel Symposium 2006 Proceedings of the Third Abel Symposium, Alesund, Norway, May 25–27, 2006*, pages 67–94. Springer, 2008.

[64] Rüdiger Brecht, Elsa Cardoso-Bihlo, and Alex Bihlo. Physics-informed neural networks for tsunami inundation modeling. *Journal of Computational Physics*, 536:114066, 2025.

[65] Yusuke Tanaka, Takaharu Yaguchi, Tomoharu Iwata, and Naonori Ueda. Energy-consistent neural operators for hamiltonian and dissipative partial differential equations. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

A Hamiltonian Structure and Conservation

The derivations below expand on the Hamiltonian and multi-timescale structure used in the main text. Hamilton's equations and the HNN objective are restated in the notation used throughout. The slow–fast splitting of a stiff Hamiltonian makes precise the separation of characteristic frequencies that motivates the frequency-filtered views. The conservation analysis establishes that the orthogonal projection conserves learned Hamiltonians in continuous time and admits a skew-gradient form, and explains why each branch fits a Hamiltonian approximation rather than an exact Hamiltonian flow.

A.1 Hamiltonian Neural Networks

For a conservative Hamiltonian system, we write the state at time t as

$$\mathbf{z}(t) = \begin{bmatrix} \mathbf{q}(t) \\ \mathbf{p}(t) \end{bmatrix} \in \mathbb{R}^{2d},$$

where $\mathbf{q}(t) \in \mathbb{R}^d$ and $\mathbf{p}(t) \in \mathbb{R}^d$ are the generalized coordinates and momenta. The scalar Hamiltonian $\mathcal{H}(\mathbf{q}, \mathbf{p})$ typically represents the total energy.

The dynamics follow Hamilton's equations,

$$\dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \qquad \dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}}. \tag{17}$$

To write these equations compactly, we introduce the canonical symplectic matrix

$$\mathbf{J} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{bmatrix}, \qquad \mathbf{J}^\intercal = -\mathbf{J}, \tag{18}$$

where $\mathbf{I}$ is the $d \times d$ identity matrix. Writing the state gradient as

$$\boldsymbol{\nabla}_{\mathbf{z}} \mathcal{H}(\mathbf{z}) = \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial \mathbf{q}}(\mathbf{q}, \mathbf{p}) \\ \frac{\partial \mathcal{H}}{\partial \mathbf{p}}(\mathbf{q}, \mathbf{p}) \end{bmatrix},$$

Hamilton's equations take the canonical form

$$\dot{\mathbf{z}} = \mathbf{J} \boldsymbol{\nabla}_{\mathbf{z}} \mathcal{H}(\mathbf{z}). \tag{19}$$

We can check energy conservation by differentiating the Hamiltonian along a trajectory,

$$\frac{d\mathcal{H}}{dt} = \boldsymbol{\nabla}_{\mathbf{z}} \mathcal{H}(\mathbf{z})^\intercal \dot{\mathbf{z}}. \tag{20}$$

Substituting (19) into (20) gives

$$\frac{d\mathcal{H}}{dt} = \boldsymbol{\nabla}_{\mathbf{z}} \mathcal{H}(\mathbf{z})^\intercal \mathbf{J} \boldsymbol{\nabla}_{\mathbf{z}} \mathcal{H}(\mathbf{z}). \tag{21}$$

Since $\mathbf{J}^\intercal = -\mathbf{J}$, the quadratic form $\mathbf{x}^\intercal \mathbf{J} \mathbf{x}$ vanishes for any vector $\mathbf{x}$, and therefore

$$\frac{d\mathcal{H}}{dt} = 0. \tag{22}$$

The continuous-time trajectories therefore remain on level sets of the Hamiltonian.

A Hamiltonian Neural Network (HNN) represents the unknown Hamiltonian with a neural network $\mathcal{H}_\theta(\mathbf{z})$ and defines the vector field by

$$\dot{\mathbf{z}}_\theta = \mathbf{J} \boldsymbol{\nabla}_{\mathbf{z}} \mathcal{H}_\theta(\mathbf{z}). \tag{23}$$

Given training data $\{(\mathbf{z}_i, \dot{\mathbf{z}}_i)\}_{i=1}^N$, the parameters θ are typically learned by minimizing

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| \dot{\mathbf{z}}_i - \mathbf{J} \boldsymbol{\nabla}_{\mathbf{z}} \mathcal{H}_\theta(\mathbf{z}_i) \right\|_2^2. \tag{24}$$

This parameterization enforces Hamiltonian structure and conserves the learned Hamiltonian in continuous time.

A.2 Slow–Fast Splitting in Hamiltonian Form

Following the multirate setting in [16], assume the potential energy can be additively decomposed into slow and fast parts,

$$U(\mathbf{q}) = V(\mathbf{q}) + W(\mathbf{q}), \qquad W(\mathbf{q}) = \frac{1}{\varepsilon}\widetilde{W}(\mathbf{q}), \quad 0 < \varepsilon \ll 1. \tag{25}$$

The scaled potential W generates stiff forces. Split the configuration into slow and fast variables

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}^s \\ \mathbf{q}^f \end{bmatrix}, \qquad \mathbf{q}^s \in Q_s,\ \mathbf{q}^f \in Q_f,\ Q_s \times Q_f = Q, \tag{26}$$

and assume (i) the fast potential depends only on the fast variables, $W = W(\mathbf{q}^f)$, while the slow potential may depend on all variables, $V = V(\mathbf{q}^s, \mathbf{q}^f)$, and (ii) the kinetic energy is quadratic with a block-diagonal mass matrix,

$$T(\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^\intercal \mathbf{M}\dot{\mathbf{q}}, \qquad \mathbf{M} = \begin{bmatrix} \mathbf{M}_s & 0 \\ 0 & \mathbf{M}_f \end{bmatrix}. \tag{27}$$

Under these assumptions, the Hamiltonian admits the slow–fast decomposition used in the multirate formulation below.

We first write the corresponding Lagrangian as

$$L(\mathbf{q}^s, \mathbf{q}^f, \dot{\mathbf{q}}^s, \dot{\mathbf{q}}^f) = \frac{1}{2}(\dot{\mathbf{q}}^s)^\intercal \mathbf{M}_s \dot{\mathbf{q}}^s + \frac{1}{2}(\dot{\mathbf{q}}^f)^\intercal \mathbf{M}_f \dot{\mathbf{q}}^f - V(\mathbf{q}^s, \mathbf{q}^f) - W(\mathbf{q}^f). \tag{28}$$

Define the conjugate momenta by the Legendre map

$$\mathbf{p}^s := \frac{\partial L}{\partial \dot{\mathbf{q}}^s} = \mathbf{M}_s \dot{\mathbf{q}}^s, \qquad \mathbf{p}^f := \frac{\partial L}{\partial \dot{\mathbf{q}}^f} = \mathbf{M}_f \dot{\mathbf{q}}^f, \tag{29}$$

The velocities are therefore

$$\dot{\mathbf{q}}^s = \mathbf{M}_s^{-1}\mathbf{p}^s, \qquad \dot{\mathbf{q}}^f = \mathbf{M}_f^{-1}\mathbf{p}^f. \tag{30}$$

The Legendre transform gives

$$\mathcal{H}(\mathbf{q}^s, \mathbf{q}^f, \mathbf{p}^s, \mathbf{p}^f) = (\mathbf{p}^s)^\intercal \dot{\mathbf{q}}^s + (\mathbf{p}^f)^\intercal \dot{\mathbf{q}}^f - L(\mathbf{q}^s, \mathbf{q}^f, \dot{\mathbf{q}}^s, \dot{\mathbf{q}}^f). \tag{31}$$

Here, the velocities $\dot{\mathbf{q}}^s, \dot{\mathbf{q}}^f$ are expressed in terms of $\mathbf{p}^s, \mathbf{p}^f$ using (30). Substituting (28)–(30) gives

$$\mathcal{H}(\mathbf{q}^s, \mathbf{q}^f, \mathbf{p}^s, \mathbf{p}^f) = \frac{1}{2}(\mathbf{p}^s)^\intercal \mathbf{M}_s^{-1}\mathbf{p}^s + \frac{1}{2}(\mathbf{p}^f)^\intercal \mathbf{M}_f^{-1}\mathbf{p}^f + V(\mathbf{q}^s, \mathbf{q}^f) + W(\mathbf{q}^f). \tag{32}$$

Hamilton's equations are

$$\dot{\mathbf{q}}^s = \frac{\partial \mathcal{H}}{\partial \mathbf{p}^s}, \quad \dot{\mathbf{p}}^s = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}^s}, \quad \dot{\mathbf{q}}^f = \frac{\partial \mathcal{H}}{\partial \mathbf{p}^f}, \quad \dot{\mathbf{p}}^f = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}^f}. \tag{33}$$

Applying (32) yields the split system

$$\dot{\mathbf{q}}^s = \mathbf{M}_s^{-1}\mathbf{p}^s, \qquad \dot{\mathbf{p}}^s = -\frac{\partial V}{\partial \mathbf{q}^s}(\mathbf{q}^s, \mathbf{q}^f), \tag{34}$$

$$\dot{\mathbf{q}}^f = \mathbf{M}_f^{-1}\mathbf{p}^f, \qquad \dot{\mathbf{p}}^f = -\frac{\partial V}{\partial \mathbf{q}^f}(\mathbf{q}^s, \mathbf{q}^f) - \boldsymbol{\nabla}_{\mathbf{q}^f} W(\mathbf{q}^f), \tag{35}$$

Next, substituting the stiffness scaling (25), with $W(\mathbf{q}^f) = \varepsilon^{-1}\bar{W}(\mathbf{q}^f)$, gives

$$\boldsymbol{\nabla}_{\mathbf{q}^f} W(\mathbf{q}^f) = \frac{1}{\varepsilon}\boldsymbol{\nabla}_{\mathbf{q}^f} \bar{W}(\mathbf{q}^f), \tag{36}$$

so the fast momentum equation becomes

$$\dot{\mathbf{p}}^f = -\frac{\partial V}{\partial \mathbf{q}^f}(\mathbf{q}^s, \mathbf{q}^f) - \frac{1}{\varepsilon}\boldsymbol{\nabla}_{\mathbf{q}^f}\bar{W}(\mathbf{q}^f), \tag{37}$$

so the fast subsystem contains an $\mathcal{O}(\varepsilon^{-1})$ force, whereas the slow subsystem in (34) contains no such stiff term.

To identify the fast timescale, let $\mathbf{q}^f_*$ be a local minimizer of $\bar{W}$ with $\boldsymbol{\nabla}_{\mathbf{q}^f}\bar{W}(\mathbf{q}^f_*) = 0$, and define $\boldsymbol{\eta} := \mathbf{q}^f - \mathbf{q}^f_*$. Linearizing $\boldsymbol{\nabla}_{\mathbf{q}^f}\bar{W}(\mathbf{q}^f)$ near $\mathbf{q}^f_*$ gives $\boldsymbol{\nabla}_{\mathbf{q}^f}\bar{W}(\mathbf{q}^f) \approx \mathbf{K}_f\boldsymbol{\eta}$ with $\mathbf{K}_f = \boldsymbol{\nabla}^2_{\mathbf{q}^f}\bar{W}(\mathbf{q}^f_*) \succeq 0$. Ignoring lower-order coupling terms in $\partial V/\partial \mathbf{q}^f$ relative to ε^{-1}, we combine

$$\dot{\mathbf{q}}^f = \mathbf{M}_f^{-1}\mathbf{p}^f, \qquad \dot{\mathbf{p}}^f \approx -\frac{1}{\varepsilon}\mathbf{K}_f\boldsymbol{\eta}$$

to obtain the leading fast oscillator

$$\ddot{\boldsymbol{\eta}} \approx -\frac{1}{\varepsilon}\mathbf{M}_f^{-1}\mathbf{K}_f\,\boldsymbol{\eta}. \tag{38}$$

The fast frequencies thus scale as $\omega \sim \varepsilon^{-1/2}$, giving a shorter characteristic timescale for $\mathbf{q}^f$ than for $\mathbf{q}^s$.

Finally, collecting the variables into the phase-space vector

$$\mathbf{z} := \begin{bmatrix}\mathbf{q}^s & \mathbf{q}^f & \mathbf{p}^s & \mathbf{p}^f\end{bmatrix}^\intercal,$$

we can write the system in (33) in canonical symplectic form

$$\dot{\mathbf{z}} = \mathbf{J}\boldsymbol{\nabla}_{\mathbf{z}}\mathcal{H}(\mathbf{z}), \qquad \mathbf{J} = \begin{bmatrix} 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \\ -\mathbf{I} & 0 & 0 & 0 \\ 0 & -\mathbf{I} & 0 & 0 \end{bmatrix}, \tag{39}$$

where the block sizes match $(\mathbf{q}^s, \mathbf{q}^f, \mathbf{p}^s, \mathbf{p}^f)$.

If the slow potential depends only on the slow variables, $V = V(\mathbf{q}^s)$, the block-diagonal kinetic energy in (27) allows the Hamiltonian to separate as

$$\mathcal{H}(\mathbf{z}) = \mathcal{H}_s(\mathbf{q}^s, \mathbf{p}^s) + \mathcal{H}_f(\mathbf{q}^f, \mathbf{p}^f),$$

and the slow and fast Hamiltonian subsystems evolve independently.

A.3 Orthogonality constraint versus strict conservation.

FS-HNN imposes orthogonality on the learned vector field. It does not directly require

$$\mathcal{H}(\mathbf{z}_{t+1}) = \mathcal{H}(\mathbf{z}_t) \tag{40}$$

at each rollout step or enforce exact symplectic or multisymplectic preservation. The orthogonality condition makes the continuous-time vector field tangent to the level sets of the Hamiltonian.

Let $\mathcal{H}(\mathbf{z})$ denote the learned Hamiltonian and let $\mathbf{f}_\theta(\mathbf{z})$ denote the unconstrained vector-field prediction before projection. We impose the condition

$$\boldsymbol{\nabla}_{\mathbf{z}}\mathcal{H}(\mathbf{z})^\intercal\mathbf{f}(\mathbf{z}) = 0, \tag{41}$$

where $\mathbf{f}(\mathbf{z})$ is the projected vector field. This condition implies that, in continuous time,

$$\frac{d\mathcal{H}}{dt} = \boldsymbol{\nabla}_{\mathbf{z}}\mathcal{H}(\mathbf{z})^\intercal\dot{\mathbf{z}} = \boldsymbol{\nabla}_{\mathbf{z}}\mathcal{H}(\mathbf{z})^\intercal\mathbf{f}(\mathbf{z}) = 0. \tag{42}$$

This continuous-time conservation property does not guarantee exact conservation after numerical time integration.

The orthogonality condition also gives a skew-gradient representation. At points where $\boldsymbol{\nabla}_{\mathbf{z}}\mathcal{H}(\mathbf{z}) \neq 0$, if

$$\boldsymbol{\nabla}_{\mathbf{z}}\mathcal{H}(\mathbf{z})^\intercal\mathbf{f}(\mathbf{z}) = 0, \tag{43}$$

then there exists a skew-symmetric structure operator $\mathbf{J}(\mathbf{z})$ such that

$$\mathbf{f}(\mathbf{z}) = \mathbf{J}(\mathbf{z})\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z}). \tag{44}$$

One way to see this is to construct

$$\mathbf{J}(\mathbf{z}) = \frac{\mathbf{f}(\mathbf{z})\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})^\intercal - \nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})\mathbf{f}(\mathbf{z})^\intercal}{\|\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})\|_2^2}. \tag{45}$$

This operator satisfies

$$\mathbf{J}(\mathbf{z})^\intercal = -\mathbf{J}(\mathbf{z}), \tag{46}$$

and, using the orthogonality condition $\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})^\intercal\mathbf{f}(\mathbf{z}) = 0$, we obtain

$$\begin{aligned}\mathbf{J}(\mathbf{z})\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z}) &= \frac{\mathbf{f}(\mathbf{z})\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})^\intercal\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z}) - \nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})\mathbf{f}(\mathbf{z})^\intercal\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})}{\|\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})\|_2^2} \\ &= \frac{\mathbf{f}(\mathbf{z})\|\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})\|_2^2 - \nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})(\mathbf{f}(\mathbf{z})^\intercal\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z}))}{\|\nabla_{\mathbf{z}}\mathcal{H}(\mathbf{z})\|_2^2} \\ &= \mathbf{f}(\mathbf{z}).\end{aligned} \tag{47}$$

Under exact orthogonality, the learned vector field therefore admits a skew-symmetric structure operator away from critical points of the Hamiltonian. This establishes energy preservation in the continuous-time skew-gradient sense, but does not establish symplectic or multisymplectic structure.

The learned functional need not coincide with the physical energy. Away from critical points, a system with a first integral I admits a local representation $\dot{\mathbf{z}} = \mathbf{J}(\mathbf{z})\nabla_{\mathbf{z}}I(\mathbf{z})$ with a non-unique skew-symmetric $\mathbf{J}(\mathbf{z})$ [57]. FS-HNN learns such an invariant–operator pair from the observed flow. Identifying the learned $\mathcal{H}$ with the analytic energy, or recovering the physical Poisson operator, requires additional physical priors or direct supervision [17, 18].

Hamiltonian approximation of filtered trajectories. The frequency-filtered trajectories provide training targets at different temporal resolutions. Filtering does not generally produce exact Hamiltonian flows, so each branch fits a Hamiltonian approximation to its target. The aggregated scalar Hamiltonian then defines the final ODE vector field through the canonical symplectic matrix.

B Data Generation

The subsections below specify the governing equations, numerical integrator, and initial conditions of each benchmark, covering the four ODE systems, the two shallow water configurations, and KdV. Because every reference trajectory comes from numerical integration, its energy is conserved only up to discretization error, so the physical-energy error reported in the main text is measured against the energy of the reference trajectory rather than a constant.

B.1 Pendulum.

For the simple pendulum, we write the state as $\mathbf{z} = (\boldsymbol{\theta}, \boldsymbol{\omega})^\intercal$, with dynamics

$$\dot{\mathbf{z}} = \begin{bmatrix}\dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\omega}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\omega} \\ -(g/L)\sin\boldsymbol{\theta}\end{bmatrix}.$$

We generate data using a symplectic (leapfrog) scheme: first update $\boldsymbol{\omega}$ to a half step, then advance $\boldsymbol{\theta}$, and finally update $\boldsymbol{\omega}$ to the full step $(\boldsymbol{\omega}_{n+\frac{1}{2}}, \boldsymbol{\theta}_{n+1}, \boldsymbol{\omega}_{n+1})$. A single random initial condition is sampled with $\boldsymbol{\theta}_0 \sim \mathcal{U}[-\pi, \pi]$ and $\boldsymbol{\omega}_0 \sim \mathcal{U}[-1, 1]$, and the system is integrated for $N_{\text{traj}} \times N_{\text{steps}}$ steps, after which the long trajectory is reshaped into N_{traj} segments. Optionally, i.i.d. Gaussian noise with standard deviation σ can be added to both $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$. The underlying continuous-time pendulum is conservative (no damping/forcing), and the leapfrog integrator is symplectic, which yields a bounded energy error and avoids systematic energy drift over long horizons.

B.2 Double Pendulum.

For the double pendulum, we write the state as $\mathbf{z} = (\boldsymbol{\theta}, \boldsymbol{\omega})^\intercal$, where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)^\intercal$ and $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)^\intercal$, with parameters (m_1, m_2, L_1, L_2, g). The acceleration vector $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)^\intercal$

is computed from the standard coupled equations, so the dynamics can be written as

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\alpha}(\boldsymbol{\theta}, \boldsymbol{\omega}) \end{bmatrix}.$$

Time stepping is performed by a semi-implicit *symplectic Euler* update: first update $\boldsymbol{\omega}$, then update $\boldsymbol{\theta}$ using the new velocities. Angles are wrapped back to $(-\pi, \pi]$ after each update to avoid unbounded growth. As in the pendulum case, we sample one random initial condition $\boldsymbol{\theta}_0 = (\boldsymbol{\theta}_{1,0}, \boldsymbol{\theta}_{2,0})^\intercal$ with $\boldsymbol{\theta}_{1,0}, \boldsymbol{\theta}_{2,0} \sim \mathcal{U}[-\pi, \pi]$, and $\boldsymbol{\omega}_0 = (\boldsymbol{\omega}_{1,0}, \boldsymbol{\omega}_{2,0})^\intercal$ with $\boldsymbol{\omega}_{1,0}, \boldsymbol{\omega}_{2,0} \sim \mathcal{U}[-1, 1]$. We evolve the system for $N_{\text{traj}} \times N_{\text{steps}}$ steps and reshape the long roll-out into N_{traj} trajectory segments. Optional additive Gaussian noise is supported for both $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$. The continuous double pendulum (without dissipation) is energy-conserving, and the symplectic Euler discretization preserves the symplectic structure, typically producing stable long-term behavior with bounded (though not exactly zero) energy error.

B.3 Fermi-Pasta-Ulam-Tsingou.

For this 1D particle chain, the state is $(\mathbf{q}, \mathbf{p}) \in \mathbb{R}^N \times \mathbb{R}^N$, where $\mathbf{q}$ and $\mathbf{p}$ collect the particle displacements and momenta. The vector dynamics are Hamiltonian:

$$\dot{\mathbf{q}} = \frac{\mathbf{p}}{m}, \qquad \dot{\mathbf{p}} = \mathbf{F}(\mathbf{q}),$$

where the i-th component of the force vector is

$$\mathbf{F}_i(\mathbf{q}) = \mathbf{f}_{i-1}(\mathbf{q}) - \mathbf{f}_i(\mathbf{q}).$$

We use periodic boundary conditions $\mathbf{q}_{N+1} \equiv \mathbf{q}_1$ and $\mathbf{q}_0 \equiv \mathbf{q}_N$ unless otherwise specified. Let $\mathbf{r}_i(\mathbf{q}) = \mathbf{q}_{i+1} - \mathbf{q}_i$ be the nearest-neighbor stretch. The scalar spring force magnitude associated with bond i is

$$\mathbf{f}_i(\mathbf{q}) = k\,\mathbf{r}_i + \alpha \mathbf{r}_i^2 + \beta \mathbf{r}_i^3.$$

Equivalently, the system corresponds to the Hamiltonian

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m} + \sum_{i=1}^{N} \left(\frac{k}{2}\mathbf{r}_i^2 + \frac{\alpha}{3}\mathbf{r}_i^3 + \frac{\beta}{4}\mathbf{r}_i^4 \right).$$

In many experiments we set $\alpha = 0$ and $\beta > 0$.

Data are generated using a symplectic velocity–Verlet integrator. Given $(\mathbf{q}_n, \mathbf{p}_n)$, compute the force $\mathbf{F}(\mathbf{q}_n)$, update positions

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\mathbf{p}_n}{m}\Delta\mathbf{t} + \frac{1}{2}\frac{\mathbf{F}(\mathbf{q}_n)}{m}\Delta\mathbf{t}^2,$$

recompute $\mathbf{F}(\mathbf{q}_{n+1})$, and then update momenta

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \frac{1}{2}(\mathbf{F}(\mathbf{q}_n) + \mathbf{F}(\mathbf{q}_{n+1}))\Delta\mathbf{t}.$$

A random initial condition is sampled as

$$\mathbf{q}_0 \sim \mathcal{N}(0, \sigma_q^2 \mathbf{I}), \qquad \mathbf{p}_0 \sim \mathcal{N}(0, \sigma_p^2 \mathbf{I}).$$

The system is integrated for $N_{\text{traj}} \times N_{\text{steps}}$ steps with fixed $\Delta\mathbf{t}$, saving every s steps. The resulting long roll-out is reshaped into N_{traj} trajectory segments of length

$$T = \left\lfloor \frac{N_{\text{steps}}}{s} \right\rfloor + 1.$$

Optionally, i.i.d. Gaussian noise with standard deviation σ can be added to both $\mathbf{q}$ and $\mathbf{p}$. We also record the total energy $E = \mathcal{H}(\mathbf{q}, \mathbf{p})$ at each saved frame. The underlying continuous-time system is conservative (no damping or forcing), and the velocity–Verlet scheme is symplectic, which typically yields stable long-horizon simulations with bounded energy error rather than systematic energy drift.

B.4 Fast–Slow Oscillator.

We use the nearly periodic Hamiltonian system of Duruisseaux et al. [62], with state $\mathbf{z} = (q_1, q_2, p_1, p_2)^\intercal$ and Hamiltonian

$$\mathcal{H}_\varepsilon(q_1, q_2, p_1, p_2) = \frac{1}{2}\left(q_1^2 + p_1^2\right) + \frac{\varepsilon}{2}\left(q_2^2 + p_2^2\right) + \varepsilon\, U(q_1, q_2), \qquad U(q_1, q_2) = q_1 q_2 \sin(2q_1 + 2q_2),$$

Here, (q_1, p_1) forms a unit-frequency fast oscillator, (q_2, p_2) forms a slow oscillator with frequency of order ε, and εU couples the two. Hamilton's equations are

$$\dot{q}_1 = p_1, \qquad \dot{q}_2 = \varepsilon p_2, \qquad \dot{p}_1 = -q_1 - \varepsilon\, \partial_{q_1} U, \qquad \dot{p}_2 = -\varepsilon q_2 - \varepsilon\, \partial_{q_2} U.$$

We set $\varepsilon = 10^{-2}$ and integrate 300 trajectories for 3000 steps using classical RK4 with $\Delta t = 0.1$; this step size resolves the fast period with approximately 63 samples. Following Duruisseaux et al. [62], we draw equal numbers of initial conditions from three phase-space windows. The general window uses $q_1, p_1 \sim \mathcal{U}[-1.6, 1.6]$, $q_2 \sim \mathcal{U}[-0.9, 1.6]$, and $p_2 \sim \mathcal{U}[-1.1, 1.1]$. The second window contains small slow-oscillator orbits centered to the right, with $q_2 \sim \mathcal{U}[0.6, 1.5]$ and $p_2 \sim \mathcal{U}[-0.5, 0.5]$; the third contains small orbits around the origin, with $q_2, p_2 \sim \mathcal{U}[-0.8, 0.8]$. In both restricted windows, $q_1, p_1 \sim \mathcal{U}[-1.6, 1.6]$. We record the total energy $\mathcal{H}_\varepsilon$ at every saved frame.

B.5 2D Shallow Water Equations (SWE) with Random Initial Condition.

We generate a two-dimensional shallow-water dataset on a periodic square domain with physical parameters $L_x = L_y = 10^6$, gravity $g = 9.81$, and mean depth $H = 100$. The computational grid is uniform with $N_x = N_y = 64$, and periodicity is enforced via circular shifts in the spatial discretization. All optional source terms are disabled (Coriolis, β-effect, friction, wind, sources/sinks), yielding an inviscid, unforced configuration.

The solver evolves conservative variables $(\mathbf{h}, \mathbf{m_x}, \mathbf{m_y})$ in flux-divergence form, where the mass equation is

$$\boldsymbol{\partial_t} \mathbf{h} = -\left(\boldsymbol{\partial_x} \mathbf{m_x} + \boldsymbol{\partial_y} \mathbf{m_y}\right),$$

and the momentum fluxes include the hydrostatic pressure contribution $\frac{1}{2} g \mathbf{h}^2$. Time integration uses a CFL-based step size

$$\Delta \mathbf{t} = 0.1\, \frac{\min(\Delta \mathbf{x}, \Delta \mathbf{y})}{\sqrt{gH}},$$

together with a fourth-order Runge-Kutta (RK4) scheme. At each saved step, diagnostic velocities are recovered as $\mathbf{u} = \mathbf{m_x}/\mathbf{h}$ and $\mathbf{v} = \mathbf{m_y}/\mathbf{h}$ (using a small depth floor), and the free-surface anomaly is computed as $\boldsymbol{\eta} = \mathbf{h} - H$.

Initial conditions are random free-surface perturbations $\boldsymbol{\eta}_0$ constructed by averaging two Gaussian random fields and then repeatedly applying Gaussian smoothing until the maximum nearest-neighbor jump falls below a prescribed threshold, enforcing spatial regularity. We generate 1000 independent rollouts and save the resulting time series (including $\mathbf{u}$, $\mathbf{v}$, and the stored height field $\mathbf{h}$). In the continuous setting, the inviscid, unforced shallow-water equations conserve mass and admit an energy invariant; in our implementation, the conservative flux form promotes mass conservation (up to numerical error), while energy is not exactly preserved due to the explicit RK4 time stepping.

B.6 Gaussian Pulse Initial Condition under SWE

We generate a two-dimensional shallow-water dataset on a periodic square domain using the same solver and physical configuration described above (uniform 64×64 grid on $[-5 \times 10^5,\, 5 \times 10^5]^2$, periodic boundaries enforced via circular shifts in the spatial discretization, inviscid and unforced). The prognostic fields are the layer depth (free-surface height) $\mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{t})$ and the horizontal velocity components $\mathbf{u}(\mathbf{x}, \mathbf{y}, \mathbf{t})$, $\mathbf{v}(\mathbf{x}, \mathbf{y}, \mathbf{t})$ (equivalently the conservative variables $(\mathbf{h}, \mathbf{m_x}, \mathbf{m_y})$ with $\mathbf{m_x} = \mathbf{hu}$ and $\mathbf{m_y} = \mathbf{hv}$). We record time series of $(\mathbf{u}, \mathbf{v}, \mathbf{h})$ together with the corresponding time stamps $\mathbf{t}$.

Unlike random-field initialization, each rollout here is initialized by a deterministic *localized Gaussian pulse* in the free-surface anomaly $\boldsymbol{\eta}_0$. On the discrete grid, we set

$$\boldsymbol{\eta}_0(i, j) = A \exp\left(-\frac{(i - i_c)^2 + (j - j_c)^2}{2\sigma^2}\right),$$

with amplitude $A = 0.1$, width $\sigma = 2$ grid cells, and center (i_c, j_c) at the domain midpoint by default. The pulse is converted to the initial height field using the same convention as the implementation (e.g., $\mathbf{h}(\cdot, \cdot, \mathbf{0}) = H + \boldsymbol{\eta}_0$ for mean depth H), after which the shallow-water equations are integrated forward in time.

We generate 500 independent rollouts (each starting from the same centered pulse unless the center is randomized) and save the resulting arrays:

$$\mathbf{u} \in \mathbb{R}^{N_{\text{traj}} \times N_t \times N_y \times N_x}, \quad \mathbf{v} \in \mathbb{R}^{N_{\text{traj}} \times N_t \times N_y \times N_x}, \quad \mathbf{h} \in \mathbb{R}^{N_{\text{traj}} \times N_t \times N_y \times N_x}, \quad \mathbf{t} \in \mathbb{R}^{N_{\text{traj}} \times N_t},$$

together with auxiliary stored diagnostics (e.g., hm) and the constructed pulse $\boldsymbol{\eta}_0$.

In the continuous setting, the inviscid, unforced shallow-water equations conserve total mass and admit an energy invariant; in our discrete implementation, the flux-form evolution promotes near-conservation of mass up to numerical error, while exact energy conservation is not enforced due to explicit time stepping.

B.7 Korteweg–de Vries Equation.

We follow the KdV setup of Eidnes and Lye [44], which is also used to evaluate ENO [65]. The equation is

$$u_t + \eta \left(\frac{1}{2}u^2\right)_x - \gamma^2 u_{xxx} = 0, \qquad \eta = 6, \quad \gamma = 1,$$

on the periodic domain $[0, P]$ with $P = 20$, discretized on $M = 100$ uniform grid points. In pseudo-Hamiltonian form the equation reads $u_t = \partial_x \, \delta\mathcal{H}/\delta u$ with the conserved functional

$$\mathcal{H}[u] = -\int \left(\frac{\eta}{6}u^3 + \frac{\gamma^2}{2}u_x^2\right) dx.$$

Initial conditions follow their two-soliton family,

$$u(x,0) = 2\sum_{l=1}^{2} c_l^2 \operatorname{sech}^2\Big(c_l((x + \frac{P}{2} - d_l P) \bmod P - \frac{P}{2})\Big), \qquad c_l \sim \mathcal{U}(\frac{1}{2}, 2), \quad d_l \sim \mathcal{U}(0,1),$$

corresponding to two solitons of heights $2c_l^2$ centered at $d_l P$. We use a Fourier pseudospectral discretization and integrating-factor RK4, which treats the stiff dispersive term exactly. Saved frames are separated by $\Delta t = 0.01$, with 20 solver substeps per saved interval. We generate 100 trajectories up to $t = 2$. Across the generated trajectories, the median relative drift in $\mathcal{H}$ is 6.3×10^{-6}; doubling the spatial resolution changes the solution by a relative L_2 error of 4.9×10^{-5}.

C Additional Experimental Results

Supplementary results on the benchmarks generated above follow the main-text division into ODE and PDE systems, covering physical-energy error, additional PDE rollouts, the ablations referenced in Section 5.3, and training cost. All models have parameter counts of the same order of magnitude (Table 6), so the comparisons are made at comparable capacity.

C.1 ODE Systems

Table 7 reports the physical-energy error of each model over the 1000-step rollout. The analytic energy of each system is evaluated on the predicted states and compared with the energy of the

Table 6: **Model parameter count.** Parameter counts of FS-HNN and the baselines are of the same order of magnitude, so all models are compared at comparable capacity. Some baselines are not scaled further because their accuracy degrades as the networks become deeper and wider.

	FS-HNN (ODE)	FS-HNN (PDE)	HNN	MLP	SympNet	PHNN	FNO
Params.	3.21×10^6	2.04×10^6	2.82×10^6	1.32×10^6	0.76×10^6	4.80×10^6	4.22×10^6

Table 7: **Physical-energy error for ODE rollouts.** We report the median relative error over the rollout for the pendulum, double pendulum, and fast–slow oscillator, and the final relative error for FPUT. FS-HNN has the lowest energy error on all four systems, so its rollouts stay faithful to the physical energy as well as to the state.

Model	Pendulum	Double pendulum	FPUT	Fast–slow
FS-HNN	$\mathbf{1.4 \times 10^{-3}}$	$\mathbf{2.3 \times 10^{-1}}$	$\mathbf{2.3 \times 10^{-2}}$	$\mathbf{5.4 \times 10^{-4}}$
HNN	8.7×10^{-1}	4.5×10^{0}	9.2×10^{3}	1.1×10^{-2}
MLP	9.5×10^{-1}	7.9×10^{0}	4.5×10^{-2}	1.5×10^{-2}
SympNet	–	–	2.6×10^{3}	5.3×10^{-3}

reference trajectory, which fluctuates slightly because the reference is generated numerically. The table gives the median relative energy error over the rollout for the pendulum, double pendulum, and fast–slow oscillator, and the final relative energy error for FPUT**, where the energy error of HNN and SympNet grows over the rollout rather than fluctuating about a fixed level.**

Tables 8 and 9 report the ablations on the number of nested views and on cutoff placement discussed in Section 5.3, with the cutoff comparison extended to KdV.

C.2 PDE Systems

Figure 4 shows the physical-energy error of FS-HNN, PHNN, and FNO on all three PDE benchmarks over the first 10 rollout steps. As in the ODE setting, the analytic energy is evaluated on the predicted fields and compared with that of the reference trajectory. Table 10 summarizes the energy error for random-initialization SWE and KdV.

Table 11 reports the ablation from Section 5.3 that separates the effects of frequency structure and energy preservation.

C.3 Computational Cost

All models are trained on an NVIDIA RTX 6000 Pro GPU. Depending on the batch size and epoch configuration, running all benchmarks requires approximately 30 GB of VRAM at minimum, and training typically does not exceed one hour per benchmark. Table 12 reports per-epoch training time on the PDE benchmarks, with FS-HNN reported separately for a single branch and for the combination stage. Although the full model involves multiple training stages, the per-epoch cost of

Table 8: **Number of nested views.** We report 50-step rollout MSE from one seed on held-out initial conditions, a protocol that differs from that of Table 1. Fixed cutoffs are obtained by repeated halving from the Nyquist frequency, and spectrum-selected cutoffs are evaluated at $K = 3$. Parentheses give the ratio of HNN MSE to the MSE of each configuration, with HNN using the architecture of Table 1. Adding views with halved cutoffs improves FPUT but degrades the fast–slow oscillator, while spectrum-selected cutoffs give the lowest error on both systems.

Model	K	FPUT		Fast–slow	
		MSE	Gain	MSE	Gain
Baseline					
HNN	–	2.03×10^{-2}	$1.0\times$	9.64×10^{-5}	$1.0\times$
Fixed					
FS-HNN	1	3.37×10^{-3}	$6.0\times$	2.29×10^{-5}	$4.2\times$
FS-HNN	2	3.32×10^{-3}	$6.1\times$	3.07×10^{-5}	$3.1\times$
FS-HNN	3	2.58×10^{-3}	$7.9\times$	8.44×10^{-3}	$0.01\times$
FS-HNN	4	1.38×10^{-3}	$14.7\times$	1.21×10^{-2}	$0.008\times$
Spectrum					
FS-HNN	3	$\mathbf{9.58 \times 10^{-4}}$	$\mathbf{21.2}\times$	$\mathbf{2.26 \times 10^{-5}}$	$\mathbf{4.3}\times$

Table 9: **Fixed versus spectrum-selected cutoffs at $K = 3$.** We compare rollout MSE using fixed normalized cutoffs $1.0/0.5/0.25$ with the spectrum-selected cutoffs from Section 4.1, listed in the final column. Gain is the ratio of the fixed-cutoff MSE to the spectrum-selected MSE. Spectrum-selected cutoffs never underperform the fixed cutoffs, and the gain is largest on the fast–slow oscillator, whose spectral scales lie far below the fixed cutoffs.

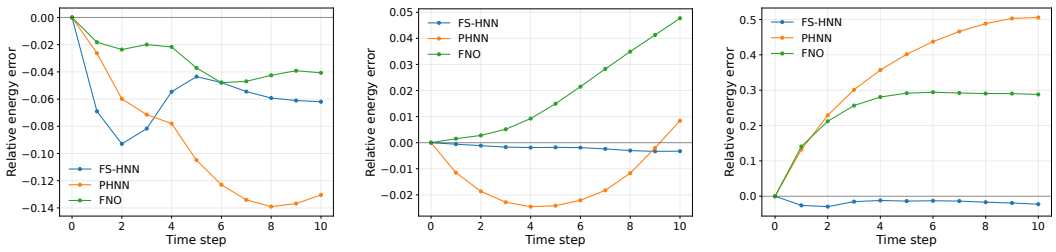
Benchmark	Fixed	Spectrum	Gain	Cutoffs
Pendulum	2.76×10^{-5}	$\mathbf{1.67 \times 10^{-5}}$	$1.7\times$	$1.0/0.05/0.005$
Double Pendulum	9.58×10^{-2}	$\mathbf{8.46 \times 10^{-2}}$	$1.1\times$	$1.0/0.692/0.164$
FPUT	2.58×10^{-3}	$\mathbf{9.58 \times 10^{-4}}$	$2.7\times$	$1.0/0.0122/0.0012$
Fast–Slow	8.44×10^{-3}	$\mathbf{2.26 \times 10^{-5}}$	$373\times$	$1.0/0.016/0.0016$
KdV	4.74×10^{-2}	$\mathbf{4.62 \times 10^{-2}}$	$1.0\times$	$1.0/0.88/0.41$

each stage is comparable to that of the baselines on SWE. On KdV, an FS-HNN stage is approximately three times slower than PHNN and five to six times slower than FNO.

D Impact Statement

This paper focuses on conservative and stable prediction for dynamical systems. We achieve both accuracy and interpretability. Moreover, our results show that, rather than relying solely on high-resolution data, allocating some modeling capacity to coarser-scale information can further improve predictive performance.

The authors do not intend for the methods described harmful behaviors. We do not believe the contribution introduces distinct societal risks beyond those commonly associated with general-purpose ML techniques; nonetheless, we support careful, responsible deployment.



(a) SWE Pulse (b) SWE Random (c) KdV

Figure 4: **Physical-energy error for PDE rollouts.** We evaluate the relative error of the SWE energy $1/2h\left(u^2+v^2\right)+1/2g\eta^2$ for (a) Gaussian-pulse and (b) random initial conditions, and of the KdV Hamiltonian $\mathcal{H}[u]$ from Appendix B.7 for (c) KdV. The reference energies are conserved up to discretization error. Table 10 summarizes panels (b) and (c). FS-HNN stays closest to the reference energy on random-initialization SWE and KdV, and ends closer than PHNN on the Gaussian pulse, where FNO tracks the reference most closely.

Table 10: **Physical-energy error for PDE rollouts.** For random-initialization SWE and KdV, we report relative error at the beginning of the rollout, the median relative error over the rollout, and the final relative error. The reference energy changes by -8.2×10^{-3} for SWE and -7.2×10^{-7} for KdV over the respective windows. FS-HNN stays closest to the reference energy on both benchmarks, over the rollout and at its end.

	SWE Random			KdV		
Model	$t = 0$	Median	Final	$t = 0$	Median	Final
FS-HNN	0	$\mathbf{1.9 \times 10^{-3}}$	-3.3×10^{-3}	0	$\mathbf{1.6 \times 10^{-2}}$	-2.2×10^{-2}
PHNN	0	1.8×10^{-2}	$+8.4 \times 10^{-3}$	0	4.2×10^{-1}	$+5.1 \times 10^{-1}$
FNO	0	1.5×10^{-2}	$+4.8 \times 10^{-2}$	0	2.9×10^{-1}	$+2.9 \times 10^{-1}$

Table 11: **Frequency structure versus energy preservation.** We report 10-step rollout MSE on random-initialization SWE and KdV. The single-scale conservative HNN retains the DeepONet Hamiltonian, learned $\mathbf{J}_\theta$, and orthogonal projection, while the frequency-structured DeepONet instead retains the frequency decomposition and aggregation. Removing either ingredient raises the error on both benchmarks, so FS-HNN relies on frequency structure and energy preservation together.

Model	SWE Random	KdV
FS-HNN	$\mathbf{2.38 \times 10^{-4}}$	$\mathbf{4.62 \times 10^{-2}}$
HNN	1.55×10^{-3}	1.47×10^{-1}
DeepONet	2.62×10^{-1}	7.77×10^{-1}

Table 12: **Per-epoch training time on PDE benchmarks.** We report the average training time in seconds per epoch. FS-HNN is reported separately for a single branch and for the combination stage.

Model / Stage	SWE Pulse	SWE Random	KdV
PHNN	7.83×10^{-2}	3.51×10^{-2}	5.08×10^{-2}
FNO	6.98×10^{-2}	5.93×10^{-2}	2.61×10^{-2}
FS-HNN single branch	4.09×10^{-2}	5.56×10^{-2}	1.31×10^{-1}
FS-HNN combination	5.57×10^{-2}	5.97×10^{-2}	1.54×10^{-1}

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 claims in the abstract and introduction are aligned with the proposed frequency-band Hamiltonian parameterization, the PDE extension via a learned operator and projection, and the ODE/PDE experimental scope. The main supporting material appears in Sections 4 and 5, with limitations discussed in Section 6.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 discusses limitations related to dataset fidelity and computational cost, including the use of approximately conservative numerical solvers and the overhead introduced by orthogonal projection operations.

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: The theoretical components are presented as derivations rather than formal theorem statements. The assumptions and derivations for Hamiltonian energy preservation, slow-fast splitting, frequency separation, and the orthogonality projection are given in Sections 3–4 and Appendix A.

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: The paper describes the model architecture, benchmark tasks, baseline comparisons, rollout protocols, and data-generation procedures in Sections 4–5 and Appendix B.

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: [No]

   Justification: The current submission does not provide an anonymized public code/data release or executable reproduction instructions. The datasets are generated by simulation and their generation procedures are described in Appendix B. All code will be released if accepted.

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: The paper reports the benchmark systems, training and rollout horizons, baseline models, data-generation procedures, model sizes, and PDE timing information, as well as hyper-parameter settings and random seeds.

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: The paper reports rollout MSE values and energy-behavior curves for the main benchmarks and error bars for our model in all benchmarks.

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: Appendix C.3 reports per-epoch training times, the hardware type, memory for the PDE benchmarks and Table 6 reports model parameter counts.

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 work uses simulated physical dynamical systems and standard numerical benchmarks, does not involve human subjects or personal data, and does not introduce privacy-sensitive or high-risk data collection. The submission is anonymized.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The current paper focuses on foundational methods for learning multiscale physical dynamics and include a dedicated discussion of positive and negative societal impacts in Appendix D. The work is not tied to a deployed system or a direct high-risk application.

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: [N/A]

    Justification: The paper does not release high-risk pretrained models, internet-scraped datasets, image generators, language models, or other assets with clear misuse risks. The experiments are based on simulated physical systems.

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: [N/A]

    Justification: The experiments use newly generated simulation data and cite prior benchmark systems and baseline methods. The paper does not rely on third-party released datasets, pretrained models, or external asset packages in the current manuscript.

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: The current paper does not release new datasets, code packages, model checkpoints, or other standalone assets. The simulation procedures used to generate the experimental data are documented in Appendix B.

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: The paper does not involve crowdsourcing, human-subject experiments, participant data, or human annotation labor.

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: The paper does not involve crowdsourcing or research with human subjects, so IRB approval or equivalent review is not applicable.

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: The core method development, experiments, and scientific claims do not involve LLMs as an important, original, or non-standard research component. Any use of LLMs limited to writing, editing, or formatting would not affect the methodology, rigor, or originality of the work.