SympFNO: Structure-Preserving Fourier Neural Operators for Physical Surrogate Modeling

Luong Doan
N2TP Technology Solutions JSC
Hanoi, Vietnam

Hai Nguyen
Nanyang Technological University
Singapore

Khanh Nguyen
Nanyang Technological University
Singapore

Hung Mai
National Economics University
Hanoi, Vietnam

Ngoc Vu
N2TP Technology Solutions JSC
Hanoi, Vietnam

Hieu Bui
Posts and Telecommunications
Institute of Technology
Hanoi, Vietnam

Nhung Duong
Hanoi University of Pharmacy
Hanoi, Vietnam

Phong Ho
N2TP Technology Solutions JSC
Hanoi, Vietnam

Long Dang
Posts and Telecommunications
Institute of Technology
Hanoi, Vietnam

Tuan Do *
Phenikaa University
Hanoi, Vietnam

Abstract

Neural operators can efficiently learn solution maps for partial differential equations, but standard architectures often violate fundamental physical structure during inference, leading to unstable long-horizon rollouts. We introduce the Symplectic Fourier Neural Operator (SympFNO), which is composed of three symplectic layers that embed structure preservation directly into Fourier-space and physical-space transformations, thereby preserving the geometric structure of Hamiltonian dynamics. Our theory provides a rigorous foundation for the architecture by showing that the structure-preserving design is mathematically consistent with Hamiltonian evolution and inherently more efficient than unconstrained operator parameterizations. Across five well-known Hamiltonian PDE problems, SympFNO achieves stable long-term rollouts, substantially lower trajectory error, more accurate conservation of physical invariants, and up to two orders of magnitude fewer parameters than state-of-the-art baselines, namely FNO and PINO.

1 Introduction

Partial differential equations (PDEs) are central to modeling real-world systems in science and engineering, from fluid flow and electromagnetic waves to chemical reactions, biological processes,

*Corresponding author: tuan.dongoc@phenikaa-uni.edu.vn

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

and material deformation. In many applications, one must solve not just a single PDE, but many related PDEs under varying conditions, parameters, geometries, or inputs. While classical solvers such as finite difference, finite volume, and finite element methods are accurate, they can be costly for high-resolution, long-time, repeated-query, or inverse-design problems.

Neural operators (NOs) learn mappings between functions rather than fixed-dimensional vectors, making them useful as fast surrogate PDE solvers. Once trained, they can predict solutions for new parameters much faster than traditional methods and may generalize across resolutions or geometries, making them valuable in climate modeling, fluid dynamics, materials design, digital twins, uncertainty quantification, and real-time control.

Despite their success, neural operators remain limited for long-time physical systems. Models such as DeepONet and Fourier neural operator (FNO) are efficient but largely data-driven, so they may not exactly preserve geometric or conservation structures, causing quantities like energy, momentum, invariants, or symplectic form to drift during rollout (Lu et al., 2021; Li et al., 2021). Physics-informed variants such as PINO only reduce drift through soft constraints, while hard-constrained methods often require projection, post-processing, extra computation, or apply only to limited PDE classes (Li et al., 2024; Tanaka et al., 2025; Baez et al., 2024; Liu et al., 2024a; Cardoso-Bihlo and Bihlo, 2025).

1.1 Our Contributions

In this paper, we introduce *SympFNO*, a structure-preserving Fourier neural operator for physical PDEs. The model acts directly on the canonical variables of the current state and predicts the future state through a learned symplectic transformation, mirroring the evolution of Hamiltonian systems. Our architecture is composed of three symplectic function-to-function mapping components: the Fourier Shear layer for global spectral interactions, the Nonlinear Shear layer for pointwise nonlinear Hamiltonian effects, and the Physical Rotation layer for amplitude-dependent local rotations. We prove that all three layers and their compositions are exactly symplectic (Theorems 1, 2, 3, and 4). We further justify the effectiveness of the architecture through key theoretical results: (i) the Fourier Shear layer provides stable and efficient representations of linear Hamiltonian dynamics (Theorems 5, 6, and 7); (ii) the Nonlinear Shear layer captures nonlinear potential kicks beyond linear symplectic maps (Theorem 8); and (iii) the Physical Rotation layer exactly preserves mass for Schrödinger-like problems (Theorem 9).

We validate the architecture and theory on five Hamiltonian benchmarks in Section 5. We show that SympFNO achieves more accurate long-horizon rollouts (Figure 2 and Table 2) and better energy and mass preservation than DeepONet and FNO-based baselines (Figure 3 and Table 2). We also include ablation experiments to confirm the role of each layer: Fourier Shear for linear Hamiltonian systems, Nonlinear Shear for nonlinear dynamics, and Physical Rotation for mass-preserving systems such as the Schrödinger equation. Sensitivity studies further show that SympFNO remains effective with moderate numbers of Fourier modes, layers, and training samples, indicating improved learning efficiency rather than reliance on large model capacity or extensive data.

1.2 Related Works

Physics-informed neural networks (PINNs) originated from early neural-network methods for differential equations (Lagaris et al., 1998) and were later formalized by Raissi et al. (2019) through PDE residual minimization. Later work improved optimization and scalability via NTK analyses (Wang et al., 2022), adaptive weighting and residual attention (McClenny and Braga-Neto, 2023; Anagnostopoulos et al., 2023), and domain-decomposition or variational extensions (Jagtap et al., 2020; Jagtap and Karniadakis, 2020; Kharazmi et al., 2021; Penwarden et al., 2023). In contrast, these approaches do not directly enforce exact geometric structure in operator learning at inference time; our work instead embeds symplectic structure into the neural operator architecture itself.

Symplectic neural networks preserve the geometry of Hamiltonian flow, not only conserved scalars. Hamiltonian Neural Networks (Greydanus et al., 2019) learn Hamiltonians from data. SympNets (Jin et al., 2020) instead use symplectic modules and prove universal approximation of symplectic maps. Later work extended this direction to general coordinates (Chen et al., 2021), nonseparable Hamiltonians (Xiong et al., 2021), improved training objectives (David and Méhats, 2023), Hamiltonian matching and backward-error perspectives (Canizares et al., 2024), and unified symplectic-architecture frameworks (Horn et al., 2025). These symplectic networks operate on fixed-dimensional

state vectors, so their dense parameterizations become costly on higher-dimensional spatial grids and do not directly transfer across grid resolutions. SympFNO addresses these limitations by embedding symplecticity into a neural operator with Fourier and shared pointwise transformations.

Transformer-based PDE solvers use attention (Cao, 2021), multiscale hierarchies (Liu et al., 2024b), and geometry-aware tokens (Wu et al., 2024). Diffusion methods handle partial observations (Huang et al., 2024), physical conditioning (Jacobsen et al., 2025), and fluid forecasting (Yang and Sommer, 2023). However, transformers may struggle on very large meshes, while diffusion methods are often costly due to iterative sampling or learned priors.

Neural operators learn mappings between infinite-dimensional function spaces and are powerful PDE surrogate models. Early operator-learning views arose from model reduction (Bhattacharya et al., 2021), with key architectures including DeepONet (Lu et al., 2021), FNO (Li et al., 2021), Geo-FNO (Li et al., 2023), Convolutional Neural Operator (Raonic et al., 2023), and Koopman Neural Operator (Xiong et al., 2024). Despite strong accuracy and efficiency, standard neural operators are largely data-driven and typically lack explicit physical structure or conservation laws, allowing rollout errors to accumulate into unphysical long-horizon dynamics.

Physics-informed neural operators (PINO) use soft constraints by combining data loss with PDE residuals to encourage physical consistency (Li et al., 2024). Related soft methods add energy penalties, adaptive activations, or improved sampling for stability and accuracy (Tanaka et al., 2025; Zhang and Ding, 2025; Daw et al., 2022). In contrast, hard-constraint methods enforce laws through projection (Baez et al., 2024), structural conservation in neural operators (Liu et al., 2024a), or invariant-manifold projection for first integrals (Cardoso-Bihlo and Bihlo, 2025). Overall, these methods improve consistency but are often limited to linear continuity-type laws or require costly projection-based post-processing.

Unlike prior models, our approach builds the operator as a sequence of symplectic layers in Fourier and real space, with linear and nonlinear transformations that preserve the geometry of the Hamiltonian dynamics. By acting directly on physically meaningful state variables, it enforces symplectic evolution rather than merely encouraging it through penalties or post-processing. As a result, SympFNO combines physical fidelity, accuracy, and efficiency, particularly in long-horizon rollouts.

2 Preliminaries

Many physical systems, including discrete systems (such as particle systems, rigid bodies, coupled oscillators, and lattice models) and continuous systems (such as elastic media, wave equations, and field models), can be formulated, or approximated after suitable discretization, in Hamiltonian form. This formulation is especially useful because it expresses the dynamics in terms of energy and provides a natural framework for structure-preserving numerical simulation.

For a Hamiltonian system, the state of the system at a given time t is described by q and p, which denote the generalized coordinates and generalized momenta, respectively. Each system has its own Hamiltonian function, which is associated with the system's energy and governs its dynamics, which we denote as $\mathcal{H} = \mathcal{H}(q, p, t)$. In numerical simulation, however, a computer cannot directly handle a continuous system with infinitely many degrees of freedom. Therefore, one considers a finite-dimensional discrete Hamiltonian system of the form

$$\mathcal{H}(q, p, t) = \sum_{i=1}^{N} \mathcal{H}_i(q, p, t), \tag{1}$$

with N denotes the number of degrees of freedom, or equivalently, the number of grid points in the discretized system, and $\mathcal{H}_i(q, p)$ denotes the contribution of the i-th or grid point to the total Hamiltonian. The time evolution of the system is then governed by Hamilton's equations,

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \qquad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i}, \qquad i = 1, \ldots, N, \tag{2}$$

where the dot denotes differentiation with respect to time.

The evolution of a Hamiltonian system is governed by symplectic geometry. In particular, the time evolution of the system preserves the underlying symplectic structure. Consequently, any exact transition map associated with a Hamiltonian system, is a symplectic map.

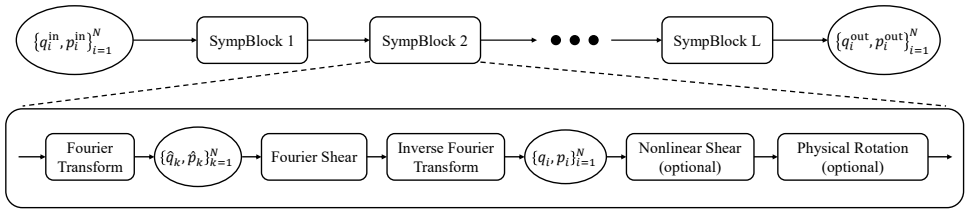


Figure 1: The architecture of SympFNO.

Definition 1 (Symplectic map)**.** *Let $z = (q, p) \in \mathbb{R}^{2N}$ denote the state vector. A map $\Phi : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ is called symplectic if its Jacobian matrix $D\Phi(z)$ satisfies $D\Phi(z)^T J\, D\Phi(z) = J$, $\forall z \in \mathbb{R}^{2N}$, where $J = \begin{pmatrix} 0 & I_N \\ -I_N & 0 \end{pmatrix}$ is the canonical symplectic matrix. Here, I_N denotes the $N \times N$ identity matrix, and "0" denotes the $N \times N$ zero matrix.*

Therefore, if a model is constructed as a symplectic map, it can naturally capture the physical state transition of a Hamiltonian system while preserving its intrinsic geometric structure. Although the exact flow of a time-independent Hamiltonian conserves energy, symplecticity alone does not guarantee this. We therefore also evaluate energy conservation and, where applicable, mass conservation.

3 Methodology

Our architecture of SympFNO, demonstrated in Figure 1, is designed to learn time-evolution maps for Hamiltonian PDEs while preserving the underlying symplectic structure. It operates directly on the canonical variables (q, p) and is built by composing three structure-preserving components: the Fourier Shear layer, the Nonlinear Shear layer, and the Physical Rotation layer. Each layer maps spatial fields to spatial fields, treating grid values as samples of functions rather than unstructured vectors. The Fourier Shear layer captures global interactions through transformations in Fourier space, the Nonlinear Shear layer introduces pointwise nonlinear expressivity in physical space, and the Physical Rotation layer models amplitude-dependent local rotations. Together, these layers provide a flexible yet symplectic architecture for representing both linear and nonlinear Hamiltonian dynamics.

3.1 Fourier Shear

The Fourier Shear layer provides the global mixing component of SympFNO by applying symplectic shear transformations to each retained Fourier mode of the canonical variables (q, p).

Definition 2 (Fourier Shear layer)**.** *The Fourier Shear Layer takes q and p as inputs, where $q, p \in \mathbb{R}^N$ and N denotes the number of grid points. Let $\hat{q}$ and $\hat{p}$ denote their corresponding retained complex Fourier coefficients. For each Fourier mode $k \le K$, the layer is defined as the composition of three consecutive shear maps with learnable scalar parameters $w_1^{(k)}, w_2^{(k)}, w_3^{(k)} \in \mathbb{R}$:*

$$\begin{aligned} S_1 &: (\hat{q}_k, \hat{p}_k) \mapsto \big(\hat{q}_k,\ \hat{p}_k + w_1^{(k)} \hat{q}_k\big), \\ S_2 &: (\hat{q}_k, \hat{p}_k) \mapsto \big(\hat{q}_k + w_2^{(k)} \hat{p}_k,\ \hat{p}_k\big), \\ S_3 &: (\hat{q}_k, \hat{p}_k) \mapsto \big(\hat{q}_k,\ \hat{p}_k + w_3^{(k)} \hat{q}_k\big). \end{aligned} \tag{3}$$

The output coefficients are obtained by successively applying these three transformations, first S_1, then S_2, and finally S_3. The modes with $k > K$ remain unchanged.

Remark 1 (No lifting to higher-dimensional channels)**.** *Operating directly on the two-dimensional phase space (q_k, p_k) for each mode is essential, since any lifting from $\mathbb{R}^2$ to $\mathbb{R}^{2n}$ generally destroys the symplectic structure associated with the original phase space. The trade-off is that each mode has fewer trainable parameters; when necessary, this limitation can be overcome by stacking additional layers.*

With the layer definition above, Theorem 2 shows that the Fourier Shear Layer defines a symplectic map for any choice of parameters $w_1^{(k)}$, $w_2^{(k)}$, and $w_3^{(k)}$. Fourier coefficients are the weights of Fourier basis functions representing the fields. Transforming these coefficients therefore defines a

function-to-function mapping that captures global spatial interactions while preserving the symplectic structure. We therefore retain Fourier Shear in every configuration as the core component for learning global spatial interactions through Fourier modes. Without it, the remaining pointwise layers cannot capture spatial coupling, even in the simplest benchmark dynamics.

3.2 Nonlinear Shear

The Nonlinear Shear layer applies pointwise nonlinear transformations in physical space, allowing SympFNO to model nonlinear Hamiltonian dynamics beyond mode-wise linear Fourier operations, such as the Sine-Gordon equation.

Definition 3 (Nonlinear Shear layer)**.** *The Nonlinear Shear layer takes q and p as inputs, where $q, p \in \mathbb{R}^N$ and N denotes the number of grid points. For each grid point $i = 1, \ldots, N$, the output is obtained by successively applying two transformations:*

$$L_g : (q_i, p_i) \mapsto \big(q_i,\ p_i + g(q_i)\big), \qquad U_h : (q_i, p_i) \quad \mapsto \big(q_i + h(p_i),\ p_i\big). \tag{4}$$

Here, g and h are learnable pointwise polynomial functions. More explicitly, $g(x) = \lambda_g^{(1)} x + \lambda_g^{(2)} x^2 + \cdots + \lambda_g^{(d)} x^d$, where $\lambda_g^{(1)}, \ldots, \lambda_g^{(d)} \in \mathbb{R}$ are learnable parameters and d is the prechosen polynomial degree of g. The function h is defined analogously.

Remark 2 (High order stability)**.** *When g or h is chosen to be a high-degree polynomial, learning the coefficient $\lambda_g^{(d)}$ can become unstable, especially during the early training epochs. To mitigate this issue, we use a scaling factor A_g^{i-1} for each coefficient term, so that $\lambda_g^{(i)} \mapsto \lambda_g^{(i)} / A_g^{i-1}$ with $i = 1, \ldots, d$, where $A_g \in \mathbb{R}$ is a prescribed constant. An analogous rescaling is applied to the coefficients of h.*

The symplecticity of the Nonlinear Shear layer is established in Theorem 3. Unlike SympNets (Jin et al., 2020), this layer defines a nonlinear operator on spatial fields through shared pointwise polynomials, using only $2d$ parameters and supporting compatible resolution changes.

3.3 Physical Rotation

The Physical Rotation layer is designed to preserve spatially local amplitude, making it especially suitable for Hamiltonian systems with mass or norm conservation, such as the Schrödinger equation.

Definition 4 (Physical Rotation layer)**.** *The Physical Rotation layer takes q and p as inputs, where $q, p \in \mathbb{R}^N$ and N denotes the number of grid points. For each grid point $i = 1, \ldots, N$, the output is obtained by applying the transformation*

$$R_\alpha : (q_i, p_i) \mapsto \big(q_i \cos\alpha_i - p_i \sin\alpha_i,\ q_i \sin\alpha_i + p_i \cos\alpha_i\big), \tag{5}$$

where the rotation angle α_i depends on the local amplitude through $\alpha_i = \alpha(q_i^2 + p_i^2)$. Here, α is a learnable polynomial function, defined analogously to g and h in Definition 3. The same coefficient rescaling described in Remark 2 is also applied to ensure stable learning for higher-order terms.

We verify that the Physical Rotation layer preserves the symplectic structure in Theorem 4. The shared rotation rule defines a mapping between functions by acting on their values at each spatial point.

Number of parameters. Let K be the number of retained Fourier modes, d the polynomial degree, L the number of layers, and m the hidden width of a standard FNO. For SympFNO, each layer contains $3K$ Fourier-shear parameters and $3d$ polynomial parameters. Thus, $N_{\mathrm{SympFNO}} = L(3K + 3d)$.

For a standard FNO (Li et al., 2021), each layer contains a complex spectral convolution with $2Km^2$ real parameters and a pointwise linear map with $m^2 + m$ parameters. Excluding input lifting and output projection, the total parameter count is $N_{\mathrm{FNO}} = L(2Km^2 + m^2 + m)$.

For a 2D problem retaining the lowest six Fourier modes in each coordinate, and using two independent Fourier blocks, the number of retained modes is $K = 2 \times 6^2 = 72$. With standard $d = 6$ and $m = 16$, we obtain $N_{\mathrm{FNO}} / N_{\mathrm{SympFNO}} \approx 158$. Thus, with the same number of Fourier modes and layers, SympFNO uses symplectic structure to require approximately 158 times fewer parameters than a standard FNO, while still achieving better accuracy (Figure 2 and Table 2).

4 Theoretical Framework

4.1 Symplecticity

We first establish exact symplecticity on the finite-dimensional phase space obtained after spatial discretization. The composition theorem reduces this task to verifying the Fourier Shear, Nonlinear Shear, and Physical Rotation layers individually. Each layer is symplectic for any learnable parameter values, so their composition preserves this structure by construction.

Theorem 1 (Symplecticity of composition)**.** *If two maps $\Phi_1, \Phi_2 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ are symplectic, then their composition $\Phi_2 \circ \Phi_1$ is also symplectic. The detailed proof is given in Appendix C.1.*

Theorem 2 (Symplecticity and analyticity of Fourier Shear)**.** *Every Fourier Shear layer defined in Definition 2 is a real-linear symplectic map on $\mathbb{R}^{2N}$, and the composition of Fourier Shear layers $\mathcal{G}_\theta$ is exactly symplectic for any parameter values θ. Moreover, since the map is linear, it is real-analytic on $\mathbb{R}^{2N}$. The detailed proof is given in Appendix C.2*

Theorem 3 (Symplecticity of the Nonlinear Shear layer)**.** *Every Nonlinear Shear layer in Definition 3 is a symplectic map on $\mathbb{R}^{2N}$ for any differentiable pointwise scalar functions $g, h : \mathbb{R} \to \mathbb{R}$, including polynomials and multilayer perceptrons (MLPs). The detailed proof is given in Appendix C.3.*

Theorem 4 (Symplecticity of the Physical Rotation layer)**.** *Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be any smooth function, and define the amplitude-dependent rotation $Ro(q,p) = \big(q\cos\alpha - p\sin\alpha,\ q\sin\alpha + p\cos\alpha\big)$, where $\alpha = f(q^2+p^2)$. Then Ro is exactly symplectic on $\mathbb{R}^2$ for any choice of differentiable f. Therefore, the Physical Rotation layer defined in Definition 4 is symplectic on $\mathbb{R}^{2N}$. The detailed proof is given in Appendix C.4.*

4.2 Effectiveness

For linear quadratic systems, the following theorem relates the learned Fourier Shear map to a modified Hamiltonian and bounds energy drift under the stated assumptions. The Harmonic Chain and Shallow Water rollouts in Figures 4 and 10 illustrate the corresponding energy behavior; the nonlinear benchmarks lie outside this guarantee.

Theorem 5 (Quadratic modified Hamiltonian and energy drift bound)**.** *Consider the linear Hamiltonian system $H(z) = \frac{1}{2} z^\top K z$, with $K = K^\top$, $z = (q,p) \in \mathbb{R}^{2N}$. Let $A_\star = \exp(\Delta t\, JK)$ denote the exact time-Δt flow map, and let A_θ be the linear map realized by the Fourier Shear layers. Assume (i) $\|A_\theta - A_\star\| \leq \epsilon$ with ϵ sufficiently small; and (ii) $\sigma(\Delta t\, JK) \subset \{z \in \mathbb{C} : -\pi < \operatorname{Im} z < \pi\}$ (This is CFL-like condition $|\omega_j \Delta t| < \pi$ for oscillatory modes with linear frequencies ω_j, and ensures $A_\star$ has no eigenvalues on $\mathbb{R}^-$). Then*

(a) Modified Hamiltonian. *The principal matrix logarithm $X_\theta = \log A_\theta$ exists, it is a Hamiltonian matrix (i.e. $X_\theta^\top J + J X_\theta = 0$), and $A_\theta = e^{\Delta t\, J\tilde{K}}$ for a symmetric matrix $\tilde{K} := -\frac{1}{\Delta t} J X_\theta$. Hence the Fourier Shear layers is the exact Δt flow of a quadratic modified Hamiltonian: $\tilde{H}(z) = \frac{1}{2} z^\top \tilde{K} z$ with $\|\tilde{K} - K\| \leq C\epsilon$.*

(b) Bounded energy drift. *Along the iterated trajectory $z_n = A_\theta^n z_0$ with $\|z_n\| \leq R$, we have $|H(z_n) - H(z_0)| \leq R^2 C\epsilon$, for all $n \geq 0$. In particular, the bound is independent of n. Hence there is no secular energy growth over time. The detailed proof appears in Appendix C.5.*

The next theorem characterizes the expressive power of Fourier Shear layers: two layers suffice to realize any symplectic Fourier multiplier on the retained modes, with identity action on the remaining modes. The Harmonic Chain and Shallow Water architecture ablations in Table 6 support the practical effectiveness of these layers for linear dynamics.

Theorem 6 (Exact representation of truncated symplectic Fourier multipliers)**.** *Assume the model contains at least two Fourier Shear layers, i.e. $L \geq 2$. Consider transformations of the form*

$$\hat{z}'_k = M_k \hat{z}_k \quad \text{for } k < K, \qquad \hat{z}'_k = \hat{z}_k \quad \text{for } k \geq K, \tag{6}$$

where $\hat{z}_k = (\hat{q}_k, \hat{p}_k) \in \mathbb{C}^2$ and $M_k \in Sp(2,\mathbb{R})$, where $Sp(2,\mathbb{R})$ denotes the set of real 2×2 matrices satisfying $M_k^\top J M_k = J$. Then, for any prescribed collection $\{M_k\}_{k<K}$ of symplectic matrices, there exists a choice of model parameters that realizes these transformations exactly. The detailed proof is given in Appendix C.6.

Remark 3 (Implementation note)**.** *In Theorem 6, each active frequency k carries a complex pair $\hat{z}_k = (\hat{q}_k, \hat{p}_k) \in \mathbb{C}^2$ acted on by a real 2×2 matrix M_k. In the real Fourier basis used in the proof of Theorem 2, each nonzero frequency corresponds to two identical real 2×2 blocks, one for cosine and one for sine coordinates. These two descriptions are equivalent and yield the same physical map.*

The comparison below shows how the symplectic constraint reduces the dimension factor in a complexity bound for the realized linear class relative to unconstrained Fourier multipliers.

Theorem 7 (Generalization bound for the realized linear class)**.** *Let $\mathcal{F}_{\mathrm{symp}}$ be the class of Fourier shear operators with depth $L \geq 2$ and K Fourier modes. Let $\mathcal{F}_{\mathrm{unc}}$ be the class of unconstrained Fourier multipliers whose per-mode maps are arbitrary $M_k \in \mathbb{R}^{2\times 2}$ with $\|M_k\|_{\mathrm{op}} \leq B$. Assume $\|x_i\|_2 \leq R_x$ for all samples i, and the loss ℓ is L_ℓ-Lipschitz. Then, for a universal constant C_0, the empirical Rademacher complexities satisfy*

$$\mathfrak{R}_n(\ell \circ \mathcal{F}_{\mathrm{symp}}) \leq C_0 L_\ell B R_x \sqrt{\frac{3K}{n}}, \qquad \mathfrak{R}_n(\ell \circ \mathcal{F}_{\mathrm{unc}}) \quad \leq C_0 L_\ell B R_x \sqrt{\frac{4K}{n}}. \tag{7}$$

Hence the leading dimension factor improves by $\sqrt{3}/2 \approx 0.866$. The detailed proof is given in Appendix C.7.

The following theorem explains why nonlinear potential kicks require more than linear symplectic maps: polynomial Nonlinear Shears represent polynomial kicks exactly and approximate continuous ones. The Pendulum Chain and Sine-Gordon ablations in Table 6 illustrate the benefit of this component, while Table 7 examines sensitivity to its polynomial degree.

Theorem 8 (Nonlinear-potential kick separation)**.** *Let $q \sim \mu$ on $[-R, R]$ with $\mathbb{E}_\mu q^2 > 0$, and consider initial states $(q, 0)$. For a potential kick $\Phi_f(q, 0) = (q, f(q))$, with $f \in L^2(\mu)$ and $L^2(\mu)$ denotes square-integrable functions, so $\mathbb{E}_\mu[f(q)^2] < \infty$. Compare the linear symplectic class $\mathcal{A}_{\mathrm{lin}}$ with the degree-d lower nonlinear shear class $\mathcal{A}^{(d)}_{\mathrm{ns}}$:*

$$\mathcal{A}_{\mathrm{lin}} = \{(q,p) \mapsto A(q,p) : A \in Sp(2,\mathbb{R})\}, \qquad \mathcal{A}^{(d)}_{\mathrm{ns}} = \{(q,p) \mapsto (q, p + g(q)) : g \in \mathcal{P}_d\}, \tag{8}$$

where $\mathcal{P}_d$ is the set of real polynomials in q of degree at most d. Then

$$\begin{aligned} \inf_{A \in \mathcal{A}_{\mathrm{lin}}} \mathbb{E}_\mu \|A(q,0) - \Phi_f(q,0)\|_2^2 &= \inf_{c \in \mathbb{R}} \mathbb{E}_\mu \big(f(q) - cq\big)^2, \\ \inf_{g \in \mathcal{P}_d} \mathbb{E}_\mu \|(q, g(q)) - \Phi_f(q,0)\|_2^2 &= \inf_{g \in \mathcal{P}_d} \mathbb{E}_\mu \big(g(q) - f(q)\big)^2. \end{aligned} \tag{9}$$

Consequently, if $f \in \mathcal{P}_d$, the nonlinear shear realizes the kick exactly with zero error. If f is continuous on $[-R, R]$, polynomial nonlinear shears approximate the kick arbitrarily well as $d \to \infty$. If f is not a scalar multiple of the function q, every linear symplectic map has strictly positive irreducible error. The detailed proof is given in Appendix C.8

Physical Rotation complements symplecticity with exact preservation of discrete mass, as established below. The Schrödinger architecture ablations in Table 6 show reduced mass drift when rotation is included alongside Nonlinear Shear, illustrating its practical role within composed models; the exact guarantee applies to the rotation layer alone.

Theorem 9 (Mass preservation of Physic Rotation layer)**.** *The mass $M = \sum_{i=1}^{N} \left(q_i^2 + p_i^2\right)$ is exactly preserved under the Physical Rotation layer R_α for all parameter values. The detailed proof is given in Appendix C.9.*

5 Numerical Experiments

We evaluate SympFNO on five Hamiltonian benchmarks in Table 1: the 1D Harmonic and Pendulum Chains, and the 2D Shallow Water, Nonlinear Schrödinger, and Sine-Gordon equations. Datasets are generated with the `dopri5` integrator (Dormand and Prince, 1980) in `PyTorch` (Paszke et al., 2019) using `torchdiffeq` (Chen, 2018). We use resolutions 32, 64, and 128 for 1D problems, and 32×32 and 256×256 for 2D problems, downsampling the latter to 64×64 for training/evaluation and testing by interpolation to 256×256, following Li et al. (2021). For each problem and grid size, we generate 1000 trajectories split into 800/100/100 for training, validation, and testing, recording states every $\Delta t = 10^{-2} s$ for 300 steps.

Table 1: Hamiltonians of benchmark problems.

Problem	Linearity	Hamiltonian
1D Harmonic Chain	Linear	$\mathcal{H} = \sum_i \left[\frac{1}{2}p_i^2 + \frac{1}{2}(q_{i+1} - q_i)^2\right]$
1D Pendulum Chain	Nonlinear	$\mathcal{H} = \sum_i \left[\frac{1}{2}p_i^2 + \frac{1}{2}(q_{i+1} - q_i)^2 - \cos(q_i)\right]$
2D Shallow Water	Linear	$\mathcal{H} = \sum_i \left[\frac{1}{2}gh_i^2 + \frac{1}{2}H_0\lvert\nabla\Phi_i\rvert^2\right]; \quad h_i = p_i, \phi_i = q_i$
2D Sine-Gordon	Nonlinear	$\mathcal{H} = \sum_i \left[\frac{1}{2}p_i^2 + \frac{1}{2}\lvert\nabla q_i\rvert^2 - \cos(q_i)\right]$
2D Schrödinger	Nonlinear	$\mathcal{H} = \sum_i \left[\frac{1}{2}\lvert\nabla\psi_i\rvert^2 - \frac{1}{4}\lvert\psi_i\rvert^4\right]; \quad \psi_i = q_i + ip_i$

Each model is trained for one-step prediction, $(q_t, p_t) \mapsto (q_{t+\Delta t}, p_{t+\Delta t})$, by minimizing mean squared error with Adam (Kingma and Ba, 2014) in `PyTorch`; PINO additionally uses its physics-informed loss. We use batch size 1024, with learning rate initialized at 10^{-3} and halved every 100 epochs. We evaluate long-term rollout accuracy, energy drift across all systems, and mass drift for the Schrödinger equation, since small one-step errors may accumulate into physically unrealistic trajectories. Full experimental settings are given in Appendix D.

Comparison with Baselines. We compare SympFNO with several neural-operator baselines: DirectFNO, which uses an FNO (Li et al., 2021) to predict the next state directly; ResFNO, which predicts the residual for improved long-time rollouts; residual versions of DeepONet (Lu et al., 2021); PINO (Li et al., 2024); SympNet (Jin et al., 2020); and ProjFNO. ProjFNO adds an energy-preserving projection inspired by Baez et al. (2024), rescaling the predicted state by $\sqrt{\mathcal{H}_{\text{in}}/\mathcal{H}_{\text{pred}}}$, but is limited to systems with quadratic energy, such as the Harmonic Chain and Shallow Water benchmarks. All models are trained for 300 epochs using consistent baseline settings: four layers, 12 Fourier modes in 1D and 4-6 per dimension in 2D for FNO-based baselines and SympFNO; width 256 with 128 branches in 1D and eight times more in 2D for DeepONet. We also include MiniResFNO, a compact residual FNO with a parameter count comparable to SympFNO. Full details are provided in Appendix D.1.

Table 2, Figure 2, and Figure 3 compare the baseline models in terms of rollout accuracy, energy drift, and mass drift over 200-step rollouts on high-resolution tasks. The results for low-resolution tasks and additional baselines are reported in the full comparison in Table 4 and 5. Across all benchmarks and resolutions, SympFNO consistently achieves the best rollout accuracy. For conservation drift, SympFNO also outperforms most baselines, with only a few exceptions. ProjFNO achieves the lowest energy drift in several cases by explicitly projecting outputs onto the conservation manifold; however, this often comes at the cost of substantially lower accuracy and requires the exact energy form, limiting its applicability mainly to simple structures such as quadratic energies. Overall, SympFNO provides the best balance of rollout accuracy, energy conservation, and mass conservation among the baselines.

Ablation Study. We provide ablation studies across all problems using grid resolutions of 32 and 64. These include architecture ablations; degree and scale-down sweeps for the Nonlinear Shear and Physical Rotation layers; sweeps over Fourier modes, number of layers, and training set size. For this purpose, each model is trained for 300 epochs and evaluated using 100-step autoregressive rollouts.

To assess each component in SympFNO, we ablate four variants: Fourier Shear only, Fourier Shear with Nonlinear Shear, Fourier Shear with Physical Rotation, and the full architecture. Full settings and results are given in Appendix D.2, matching our theoretical predictions. For the linear Harmonic Chain and Shallow Water problems, Theorems 6 and 7 suggest that Fourier Shear layers are sufficient; empirically, they achieve the lowest rollout error and energy drift, while nonlinear components may distract from the true dynamics. For the nonlinear Pendulum Chain and Sine-Gordon problems, Nonlinear Shear improves accuracy and energy conservation, consistent with Theorem 8, whereas additional Physical Rotation can complicate training and cause unstable rollouts. For the Schrödinger

Table 2: Comparison of baseline models in rollout accuracy, energy drift, and mass drift on high-resolution tasks. The full version is provided in Table 4. For each benchmark, the first row reports the mean relative rollout error over 200 time steps, the second row reports the final relative energy drift, and the third row, shown only for the Schrödinger equation, reports the final relative mass drift. The last row reports the number of real scalar model parameters. Metric values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	MiniResFNO	ResFNO	DeepONet	PINO	ProjFNO	**SympFNO**
Harmonic grid 64	(2.69 ± 0.19)E-02 (2.80 ± 0.93)E-03 1 058	(9.91 ± 2.93)E-03 (7.14 ± 1.21)E-04 26 946	(2.61 ± 0.16)E-02 (2.55 ± 0.18)E-03 559 875	(1.45 ± 0.24)E-02 (6.82 ± 0.39)E-04 26 946	(2.00 ± 0.75)E-02 (**7.57 ± 0.88**)**E-06** 26 946	(**8.91 ± 0.92**)**E-05** (3.23 ± 0.45)E-05 **144**
Harmonic grid 128	(4.71 ± 0.53)E-02 (4.03 ± 1.68)E-03 1 058	(1.09 ± 0.26)E-02 (7.82 ± 1.78)E-04 26 946	(2.70 ± 0.07)E-02 (2.21 ± 0.10)E-03 592 643	(8.80 ± 1.12)E-03 (6.14 ± 0.72)E-04 26 946	(1.14 ± 0.31)E-02 (**8.24 ± 5.13**)**E-06** 26 946	(**9.31 ± 0.36**)**E-05** (4.26 ± 0.58)E-05 **144**
Pendulum grid 64	(6.86 ± 1.20)E-02 (7.37 ± 0.17)E-03 1 058	(1.67 ± 0.34)E-02 (8.91 ± 1.58)E-04 26 946	(1.42 ± 0.23)E-01 (1.36 ± 0.52)E-02 559 875	(1.72 ± 0.08)E-02 (9.62 ± 1.38)E-04 26 946	– – –	(**2.15 ± 0.07**)**E-03** (**1.09 ± 0.01**)**E-04** **224**
Pendulum grid 128	(2.74 ± 1.38)E-01 (7.00 ± 2.50)E-03 1 058	(1.73 ± 0.43)E-02 (7.21 ± 1.76)E-04 26 946	(1.53 ± 0.03)E-01 (1.20 ± 0.38)E-02 592 643	(1.74 ± 0.35)E-02 (6.53 ± 1.09)E-04 26 946	– – –	(**3.49 ± 0.09**)**E-03** (**1.84 ± 0.29**)**E-04** **224**
Shallow Water grid 64^2	(1.19 ± 0.27)E-01 (5.57 ± 1.28)E-03 4 422	(2.91 ± 0.31)E-02 (1.94 ± 0.24)E-03 149 234	(3.25 ± 0.05)E-01 (9.18 ± 0.68)E-03 50 356 227	(5.09 ± 0.66)E-02 (2.95 ± 0.10)E-02 149 234	(3.05 ± 0.21)E-02 (**1.50 ± 0.43**)**E-06** 149 234	(**5.84 ± 0.66**)**E-04** (5.03 ± 0.26)E-05 **792**
Shallow Water grid 256^2	(1.20 ± 0.22)E-01 (5.96 ± 1.05)E-03 4 422	(4.28 ± 0.17)E-02 (2.50 ± 0.42)E-03 149 234	(3.06 ± 0.03)E-01 (8.74 ± 0.44)E-03 50 356 227	(7.04 ± 0.54)E-02 (2.95 ± 0.06)E-02 149 234	(4.36 ± 0.13)E-02 (**1.88 ± 0.01**)**E-03** 149 234	(**2.87 ± 0.00**)**E-02** (1.89 ± 0.01)E-03 **792**
Sine-Gordon grid 64^2	(8.88 ± 0.86)E-02 (2.48 ± 0.18)E-03 4 422	(2.56 ± 0.21)E-02 (1.07 ± 0.17)E-03 149 234	(7.31 ± 0.42)E-01 (1.90 ± 0.08)E-02 50 356 227	(4.96 ± 0.32)E-02 (1.99 ± 0.09)E-02 149 234	– – –	(**7.49 ± 0.65**)**E-03** (**1.16 ± 0.01**)**E-04** **872**
Sine-Gordon grid 256^2	(1.03 ± 0.11)E-01 (2.56 ± 0.72)E-03 4 422	(4.35 ± 0.21)E-02 (1.96 ± 0.35)E-03 149 234	(6.89 ± 0.29)E-01 (1.78 ± 0.23)E-02 50 356 227	(7.26 ± 0.31)E-02 (1.96 ± 0.07)E-02 149 234	– – –	(**3.02 ± 0.02**)**E-02** (**1.40 ± 0.00**)**E-03** **872**
Schrödinger grid 64^2	(2.65 ± 0.10)E-01 (1.02 ± 0.09)E-02 (9.21 ± 0.49)E-03 4 422	(1.06 ± 0.09)E-01 (6.68 ± 1.42)E-03 (6.59 ± 1.42)E-03 149 234	(1.52 ± 1.71)E+00 (8.08 ± 9.89)E-02 (5.97 ± 7.27)E-02 50 356 227	(7.71 ± 0.61)E-01 (3.77 ± 0.32)E-01 (2.91 ± 0.31)E-01 149 234	– – – –	(**7.34 ± 0.69**)**E-02** (**5.54 ± 0.40**)**E-04** (**4.53 ± 0.49**)**E-04** **832**
Schrödinger grid 256^2	(2.75 ± 0.04)E-01 (1.08 ± 0.15)E-02 (1.03 ± 0.13)E-02 4 422	(1.15 ± 0.08)E-01 (7.11 ± 0.86)E-03 (6.48 ± 0.90)E-03 149 234	(1.58 ± 1.78)E+00 (8.41 ± 10.45)E-02 (5.99 ± 7.21)E-02 50 356 227	(8.07 ± 0.54)E-01 (4.14 ± 0.04)E-01 (3.26 ± 0.04)E-01 149 234	– – – –	(**8.48 ± 0.44**)**E-02** (**4.68 ± 0.42**)**E-04** (**5.27 ± 0.82**)**E-04** **832**

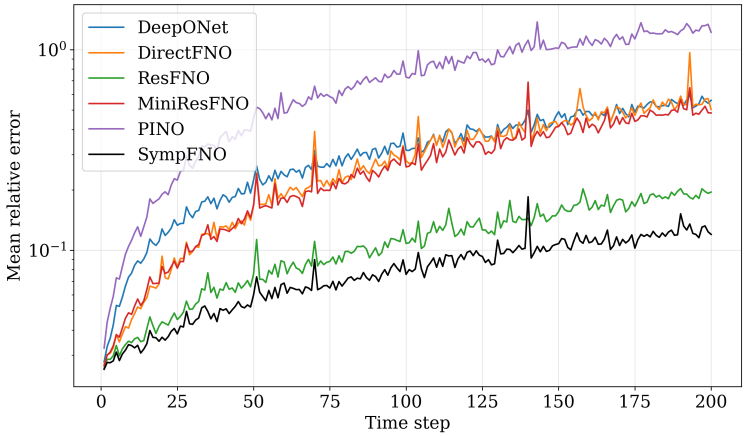
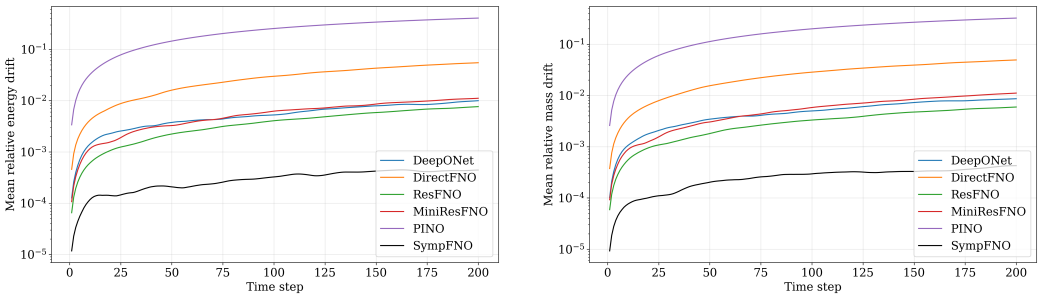


Figure 2: Mean relative error for each baseline compared with SympFNO across 200 rollout steps on the 2D Schrödinger super-resolution task with 256×256 grid points (seed 636).

equation, Physical Rotation is designed to preserve mass, as supported by Theorem 9, and improves energy and mass conservation with comparable accuracy when added alongside Nonlinear Shear.

We study sensitivity to polynomial degree and scale-down factor in Nonlinear Shear and Physical Rotation layers (Appendices D.3 and D.4). Linear problems favor lower degrees or larger scale-down factors. For nonlinear problems, Nonlinear Shear attains its lowest mean rollout errors with degrees 4-10 and scale-down factors 4-7. Nonlinear Shear achieves lower mean rollout error than Physical Rotation at their respective best tested degrees. For Schrödinger, Nonlinear Shear needs higher



(a) Energy drift

(b) Mass drift

Figure 3: Mean relative drift of energy (a) and mass (b) for each baseline against SympFNO across 200 rollout steps on the 2D Schrödinger super-resolution task with 256×256 grid points (seed 636).

degrees due to stiffness and stronger nonlinearity, while Physical Rotation rollout error varies little across the tested degrees.

Finally, we examine the required layers, Fourier modes, and training samples (Appendix D.5) by sweeping them with the selected architecture for each problem. We find that 4-6 layers usually suffice: fewer layers may underfit, whereas additional layers, especially those with high polynomial degree, make the dynamics harder to learn. Fourier modes follow a similar pattern, with the optimum depending on resolution and nonlinearity; we recommend 8-14 modes for 1D problems and 4-7 modes per dimension for 2D problems. For data, easier problems require only small number of trajectories to learn stable physics, although the best accuracy is achieved using all 800 trajectories in most cases.

6 Discussion

Our results suggest that enforcing symplectic structure architecturally is a useful inductive bias for Hamiltonian PDEs. Unlike soft physics-informed penalties, SympFNO preserves the symplectic form by construction at every layer and rollout step, which is crucial for long-time predictions. Our experiments show that this design improves long-horizon stability and reduces physical drift compared with standard neural-operator baselines.

Exact symplecticity does not imply exact conservation of all physical invariants. A symplectic map preserves phase-space geometry, but not necessarily the exact Hamiltonian energy unless it matches the true flow or an appropriate symplectic integrator. Thus, SympFNO can reduce energy drift without guaranteeing perfect conservation, providing structural stability rather than replacing all physical constraints. Combining it with energy, mass, or momentum constraints, or projection-based post-processing, may further improve fidelity. Multi-step training may also enhance rollout stability, but we leave these directions to future work.

Our proposed approach preserves symplectic structure and is therefore suited to closed systems with exact symplectic or Hamiltonian dynamics, such as wave equations, Maxwell equations in vacuum, ideal fluids, and Vlasov-type kinetic equations. However, dissipative systems, including heat diffusion, viscous flows, reaction-diffusion systems, and Navier-Stokes dynamics, are not purely symplectic, since these systems are not closed and energy can dissipate. Thus, our architecture is not guaranteed to learn them properly. One possible extension is to decompose the model into a symplectic component and a corrective component that drives the evolution toward the observed dissipative solution, similar to perturbation theory, but this remains beyond the scope of this work.

Acknowledgments and Disclosure of Funding

Computing resources for this work were provided by N2TP Technology Solutions JSC. This research received no external funding. The authors declare no competing interests.

References

Al-Mohy, A. H., Higham, N. J., and Relton, S. D. (2013). Computing the Fréchet derivative of the matrix logarithm and estimating the condition number. *SIAM Journal on Scientific Computing*, 35(4):C394–C410.

Anagnostopoulos, S. J., Toscano, J. D., Stergiopulos, N., and Karniadakis, G. E. (2023). Residual-based attention and connection to information bottleneck theory in PINNs. *arXiv preprint arXiv:2307.00379v1*.

Arslan, B. (2021). Structure preserving algorithm for the logarithm of symplectic matrices. *Mathematical Sciences and Applications E-Notes*, 9(3):133–141.

Baez, A., Zhang, W., Ma, Z., Das, S., Nguyen, L. M., and Daniel, L. (2024). Guaranteeing conservation laws with projection in physics-informed neural networks. In *NeurIPS 2024 Workshop on Data-driven and Differentiable Simulations, Surrogates, and Solvers (D3S3)*.

Bhattacharya, K., Hosseini, B., Kovachki, N. B., and Stuart, A. M. (2021). Model reduction and neural networks for parametric PDEs. *The SMAI Journal of Computational Mathematics*, 7:121–157.

Canizares, P., Murari, D., Schönlieb, C.-B., Sherry, F., and Shumaylov, Z. (2024). Hamiltonian matching for symplectic neural integrators. *arXiv preprint arXiv:2410.18262*.

Cao, S. (2021). Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, pages 24924–24940. Curran Associates, Inc.

Cardoso-Bihlo, E. and Bihlo, A. (2025). Exactly conservative physics-informed neural networks and deep operator networks for dynamical systems. *Neural Networks*, 181:106826.

Chen, R. T. Q. (2018). torchdiffeq.

Chen, Y., Matsubara, T., and Yaguchi, T. (2021). Neural symplectic form: Learning Hamiltonian equations on general coordinate systems. In *Advances in Neural Information Processing Systems*, volume 34, pages 16659–16670. Curran Associates, Inc.

David, M. and Méhats, F. (2023). Symplectic learning for Hamiltonian neural networks. *Journal of Computational Physics*, 494:112495.

Daw, A., Bu, J., Wang, S., Perdikaris, P., and Karpatne, A. (2022). Rethinking the importance of sampling in physics-informed neural networks. *arXiv preprint arXiv:2207.02338v1*.

Dormand, J. R. and Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Horn, P., Saz Ulibarrena, V., Koren, B., and Portegies Zwart, S. (2025). A generalized framework of neural networks for Hamiltonian systems. *Journal of Computational Physics*, 521:113536.

Huang, J., Yang, G., Wang, Z., and Park, J. J. (2024). DiffusionPDE: Generative PDE-solving under partial observation. In *Advances in Neural Information Processing Systems*, volume 37, pages 130291–130323. Curran Associates, Inc.

Jacobsen, C., Zhuang, Y., and Duraisamy, K. (2025). CoCoGen: Physically consistent and conditioned score-based generative models for forward and inverse problems. *SIAM Journal on Scientific Computing*, 47(2):C399–C425.

Jagtap, A. D. and Karniadakis, G. E. (2020). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5):2002–2041.

Jagtap, A. D., Kharazmi, E., and Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028.

Jin, P., Zhang, Z., Zhu, A., Tang, Y., and Karniadakis, G. E. (2020). SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132:166–179.

Kharazmi, E., Zhang, Z., and Karniadakis, G. E. (2021). hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000.

Li, Z., Huang, D. Z., Liu, B., and Anandkumar, A. (2023). Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*.

Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., and Anandkumar, A. (2024). Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27.

Liu, N., Fan, Y., Zeng, X., Klöwer, M., Zhang, L., and Yu, Y. (2024a). Harnessing the power of neural operators with automatically encoded conservation laws. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 30965–30997. PMLR.

Liu, X., Xu, B., Cao, S., and Zhang, L. (2024b). Mitigating spectral bias for the multiscale operator learning. *Journal of Computational Physics*, 506:112944.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229.

McClenny, L. D. and Braga-Neto, U. M. (2023). Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Penwarden, M., Jagtap, A. D., Zhe, S., Karniadakis, G. E., and Kirby, R. M. (2023). A unified scalable framework for causal sweeping strategies for physics-informed neural networks (PINNs) and their temporal decompositions. *Journal of Computational Physics*, 493:112464.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Raonic, B., Molinaro, R., De Ryck, T., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., and de Bézenac, E. (2023). Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pages 77187–77200. Curran Associates, Inc.

Tanaka, Y., Yaguchi, T., Iwata, T., and Ueda, N. (2025). Energy-consistent neural operators for Hamiltonian and dissipative partial differential equations. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 1882–1890. PMLR.

Wang, S., Yu, X., and Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768.

Wu, H., Luo, H., Wang, H., Wang, J., and Long, M. (2024). Transolver: A fast transformer solver for PDEs on general geometries. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F., editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 53681–53705. PMLR.

Xiong, S., Tong, Y., He, X., Yang, S., Yang, C., and Zhu, B. (2021). Nonseparable symplectic neural networks. In *International Conference on Learning Representations*.

Xiong, W., Huang, X., Zhang, Z., Deng, R., Sun, P., and Tian, Y. (2024). Koopman neural operator as a mesh-free solver of non-linear partial differential equations. *Journal of Computational Physics*, 513:113194.

Yang, G. and Sommer, S. (2023). A denoising diffusion model for fluid field prediction. *arXiv preprint arXiv:2301.11661*.

Zhang, J. and Ding, C. (2025). Simple yet effective adaptive activation functions for physics-informed neural networks. *Computer Physics Communications*, 307:109428.

A Broader Impacts

SympFNO may contribute to more reliable and efficient scientific machine learning by improving long-horizon surrogate modeling for Hamiltonian PDEs. This can benefit applications such as wave propagation, plasma modeling, electromagnetics, ideal-fluid simulation, uncertainty quantification, and real-time control, where accurate and stable physical prediction is important.

We do not identify any clear direct negative societal impact from SympFNO itself, since it is a methodological contribution for structure-preserving scientific surrogate modeling and does not involve personal data, human-subject decisions, or generative content.

B Additional Theory

Theorem 10 (Approximation of split-step symplectic rotation maps)**.** *Let $\mathcal{K} \subset \mathbb{R}^{2N}$ be compact, and let $\Psi = \mathrm{Ro}_M \circ \mathrm{SM}_M \circ \cdots \mathrm{Ro}_1 \circ \mathrm{SM}_1$ where each SM_m is a real symplectic Fourier multiplier and each Ro_m is a pointwise amplitude-dependent rotation.*

Assume that each rotation angle satisfies $\alpha_m \in C^1([0,\rho^2])$ and $\alpha_m(0) = 0$ for some $\rho > \rho_\star$. Here, $C^1[0,\rho^2]$ denotes the class of continuously differentiable functions on $[0,\rho^2]$, and $\rho_\star := \max_{0\le j\le 2M-1} \sup_{z\in K_j} \|z\|$ with $K_0 = \mathcal{K}$ and $K_j = F_j(K_{j-1})$, where

$$F_1 = \mathrm{SM}_1, \quad F_2 = \mathrm{Ro}_1, \quad \ldots, \quad F_{2M-1} = \mathrm{SM}_M, \quad F_{2M} = \mathrm{Ro}_M. \tag{10}$$

Then, for every $\varepsilon > 0$, there exists a SympFNO model $\mathcal{G}_\theta$, composed of sufficiently many Fourier Shear and Physical Rotation layers with sufficiently large polynomial degree, such that

$$\sup_{z\in\mathcal{K}} \|\mathcal{G}_\theta(z) - \Psi(z)\| < \varepsilon. \tag{11}$$

Proof. We prove the result by approximating each factor separately and then controlling the error of the composition.

Step 1: Fourier shear realization. By Theorem 6, each real symplectic Fourier multiplier SM_m can be realized exactly by a SympFNO using at most two Fourier-shear layers. Thus all Fourier multiplier factors can be realized exactly using at most $2M$ Fourier-shear layers in total.

Step 2: Rotation approximation. Let the rotation angle be $g_m(r_i^2)$. We approximate it by a polynomial angle $\tilde{p}_m(r_i^2) = \sum_{j=1}^{P} a_j (r_i^2)^j$ on $[0,\rho^2]$.

Since $g_m \in C^1([0,\rho^2])$ and $\rho > \rho_\star$, Weierstrass approximation gives, for any $\delta > 0$, a polynomial q_m such that $\|g_m' - q_m\|_{\infty,[0,\rho^2]} < \delta$. Define

$$\tilde{p}_m(s) := \int_0^s q_m(t)\, dt. \tag{12}$$

Then $\tilde{p}_m(0) = 0$, $\tilde{p}_m$ has no constant term, and

$$\|g_m - \tilde{p}_m\|_{\infty,[0,\rho^2]} \le \rho^2\delta, \qquad \|g_m' - \tilde{p}_m'\|_{\infty,[0,\rho^2]} < \delta. \tag{13}$$

Thus $\|\tilde{p}_m'\|_\infty \le \|g_m'\|_\infty + \delta$. For $\|x\| \le \rho$, the angle error gives

$$|\mathrm{Rot}_\alpha(x) - \mathrm{Rot}_\beta(x)| \le \|x\|\,|\alpha - \beta| \le \rho^3\delta, \tag{14}$$

and hence at most $\sqrt{N}\rho^3\delta$ in the full-space norm.

Step 3: Telescoping composition. Using the factor labeling and forward images K_j from the theorem statement, let $\eta := \rho - \rho_\star > 0$. Let $\hat{F}_j$ approximate F_j, exactly for Fourier multipliers and by polynomial rotations for rotation factors. When SM_m requires two shear layers, use an identity rotation in the first layer and the nontrivial rotation in the second.

Define $G_\star := \max_m \|g_m'\|_{\infty,[0,\rho^2]}$ and

$$\Lambda_0 := \max\big(B,\ 1 + 2\rho^2(G_\star + 1)\big). \tag{15}$$

Restricting to $\delta \leq 1$, we have $\|\tilde{p}'_m\|_\infty \leq G_\star + 1$, so each approximate rotation factor has Lipschitz constant at most Λ_0 on B_ρ.

Since $\hat{F}_j \to F_j$ uniformly on B_ρ as $\delta \to 0$, and there are finitely many factors, choose δ small enough so that all mixed intermediate states in the telescoping expansion remain in B_ρ. Therefore,

$$\|\hat{\Psi}(z) - \Psi(z)\| \leq \sum_{j=1}^{2M} \left(\prod_{i=j+1}^{2M} \mathrm{Lip}(\hat{F}_i; B_\rho) \right) \|\hat{F}_j(z_j) - F_j(z_j)\|, \tag{16}$$

where $z_j \in K_{j-1} \subset B_{\rho_\star} \subset B_\rho$. Fourier multiplier errors are zero, while each rotation error is at most $\sqrt{N}\rho^3\delta$. The Lipschitz products are bounded by Λ_0^{2M}. Choose

$$\delta_0 := \min\left\{1,\ \frac{\varepsilon}{4M\sqrt{N}\rho^3\Lambda_0^{2M}}\right\}. \tag{17}$$

Approximating each g_m in C^1 within δ_0 yields $\|\hat{\Psi}(z) - \Psi(z)\| < \varepsilon$ for all $z \in \mathcal{K}$. Hence $\mathcal{G}_\theta := \hat{\Psi}$ gives the desired SympFNO approximation. □

C Full Proofs

C.1 The detailed proof of Theorem 1

Proof. Let $z \in \mathbb{R}^{2n}$. By the chain rule,

$$D(\Phi_2 \circ \Phi_1)(z) = D\Phi_2(\Phi_1(z))\, D\Phi_1(z). \tag{18}$$

Set $A := D\Phi_1(z), B := D\Phi_2(\Phi_1(z))$. Then $D(\Phi_2 \circ \Phi_1)(z) = BA$. Therefore,

$$\big(D(\Phi_2 \circ \Phi_1)(z)\big)^\top J\, D(\Phi_2 \circ \Phi_1)(z) = (BA)^\top J(BA). \tag{19}$$

Using $(BA)^\top = A^\top B^\top$, we obtain

$$(BA)^\top J(BA) = A^\top B^\top JBA. \tag{20}$$

Since Φ_2 is symplectic, we have $B^\top JB = J$, and hence

$$A^\top B^\top JBA = A^\top JA. \tag{21}$$

Since Φ_1 is symplectic, we also have $A^\top JA = J$. Thus,

$$\big(D(\Phi_2 \circ \Phi_1)(z)\big)^\top J\, D(\Phi_2 \circ \Phi_1)(z) = J. \tag{22}$$

Because $z \in \mathbb{R}^{2n}$ was arbitrary, it follows that $\Phi_2 \circ \Phi_1$ is symplectic. □

C.2 The detailed proof of Theorem 2

Lemma 1 (Symplecticity of scalar shear)**.** *For any $w \in \mathbb{R}$, $S_a(q,p) = (q,\ p + wq)$ and $S_b(q,p) = (q + wp,\ p)$ on $\mathbb{R}^2$ is symplectic.*

Proof. We verify the condition $DS^\top J_1\, DS = J_1$ where $J_1 = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$.

For S_a, the Jacobian is $DS_a = \left(\begin{smallmatrix} 1 & 0 \\ w & 1 \end{smallmatrix}\right)$. Compute:

$$DS_a^\top J_1\, DS_a = \begin{pmatrix} 1 & w \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ w & 1 \end{pmatrix}. \tag{23}$$

Multiplying the first two factors:

$$\begin{pmatrix} 1 & w \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} -w & 1 \\ -1 & 0 \end{pmatrix}. \tag{24}$$

Then:

$$\begin{pmatrix} -w & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ w & 1 \end{pmatrix} = \begin{pmatrix} -w+w & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = J_1. \tag{25}$$

The cancellation $-w + w = 0$ holds for every $w \in \mathbb{R}$ without any symmetry condition, because in dimension $n = 1$ every scalar is trivially symmetric. The proof for S_b is analogous: $DS_b = \left(\begin{smallmatrix} 1 & w \\ 0 & 1 \end{smallmatrix}\right)$ and the same calculation with rows and columns interchanged yields J_1. □

Remark 4 (Symplecticity of linear matrix shear)**.** *In dimension $n > 1$, the shear $S(q, p) = (q, p + Wq)$ with $W \in \mathbb{R}^{n\times n}$ is symplectic if and only if $W = W^\top$. The off-diagonal block of $DS^\top J\, DS - J$ equals $W - W^\top$, which vanishes precisely when W is symmetric. Our architecture uses $n = 1$ throughout, so this subtlety does not arise.*

Lemma 2 (Symmetric physical space matrix induced by Fourier shear)**.** *Let $D_w = \mathrm{diag}(w_0, w_1, w_1, w_2, w_2, \ldots)$ be a real diagonal matrix of mode weights in the real orthogonal Fourier basis $F_\mathbb{R}$. Then the matrix $C_w := F_\mathbb{R}^\top D_w F_\mathbb{R}$ is real and symmetric.*

Proof. Since D_w is a real diagonal matrix, it is real and symmetric, so $D_w^\top = D_w$. As $F_\mathbb{R}$ is real orthogonal, the matrix $C_w = F_\mathbb{R}^\top D_w F_\mathbb{R}$ is real, and $C_w^\top = (F_\mathbb{R}^\top D_w F_\mathbb{R})^\top = F_\mathbb{R}^\top D_w^\top F_\mathbb{R} = F_\mathbb{R}^\top D_w F_\mathbb{R} = C_w$. Hence C_w is real and symmetric. □

Proof of Theorem 2. We work throughout in the real orthogonal Fourier basis, which avoids the complex-coordinate subtlety. The real basis uses the DC mode, cosine/sine pairs for $1 \le k \le N/2-1$, and the Nyquist mode if N is even.

Step 1. Symplecticity of each layer Let $F_\mathbb{R} \in \mathbb{R}^{N\times N}$ denote the real orthogonal Fourier matrix, so that $F_\mathbb{R} F_\mathbb{R}^\top = I$. Let $D_w = \mathrm{diag}(w_0, w_1, w_1, w_2, w_2, \ldots)$ be the real diagonal matrix of mode weights, with each nonzero frequency weight w_k repeated on both its cosine and sine coordinates. Then the lower Fourier Shear in physical space is

$$(q, p) \mapsto (q,\ p + C_w q), \tag{26}$$

with $C_w := F_\mathbb{R}^\top D_w F_\mathbb{R}$. By Lemma 2, the matrix C_w is real and symmetric. It therefore follows from Remark 4 that each Fourier Shear layer is symplectic. The proof for the upper Fourier Shear is entirely analogous.

Step 2. Symplecticity of the composition. By Theorem 1, the composition of symplectic maps is again symplectic. Since each Fourier Shear layer is symplectic by Step 1, it follows that the composition of all layers is symplectic.

Step 3. Analyticity Every component of Fourier Shear, the real Fourier transform $F_\mathbb{R}$, the diagonal multiplications, and the inverse transform $F_\mathbb{R}^\top$, is a linear map. A composition of linear maps is linear, hence entire real-analytic. This analyticity is relevant for the Benettin–Giorgilli extension to nonlinear Hamiltonians (see Remark after Theorem 2), where the BEA theory requires the map to be analytic in a complex strip. □

C.3 The detailed proof of Theorem 3

Proof. For the lower shear, set

$$G(q) = \mathrm{diag}(g_1'(q_1), \ldots, g_N'(q_N)). \tag{27}$$

Because the functions are applied pointwise, $G(q)$ is diagonal and hence $G(q) = G(q)^\top$. The Jacobian of L_g is

$$DL_g(q, p) = \begin{pmatrix} I_N & 0 \\ G(q) & I_N \end{pmatrix}. \tag{28}$$

A direct multiplication gives

$$\big(DL_g\big)^\top J DL_g = \begin{pmatrix} G(q) - G(q)^\top & I_N \\ -I_N & 0 \end{pmatrix} = J. \tag{29}$$

Thus the lower shear is symplectic.

For the upper shear, set

$$H(p) = \mathrm{diag}(h_1'(p_1), \ldots, h_N'(p_N)). \tag{30}$$

Again $H(p) = H(p)^\top$, and

$$DU_h(q,p) = \begin{pmatrix} I_N & H(p) \\ 0 & I_N \end{pmatrix}. \tag{31}$$

Hence

$$\big(DU_h\big)^\top J DU_h = \begin{pmatrix} 0 & I_N \\ -I_N & H(p)^\top - H(p) \end{pmatrix} = J. \tag{32}$$

Therefore U_h is symplectic.

Finally, by the chain rule in Theorem 1, the full Nonlinear Shear layer is symplectic. □

Remark 5 (Coupled MLPs require a symmetric Jacobian)**.** *The pointwise assumption is essential. A fully coupled vector MLP $g : \mathbb{R}^N \to \mathbb{R}^N$ gives the lower shear $(q,p) \mapsto (q, p + g(q))$, whose Jacobian is symplectic if and only if $Dg(q) = Dg(q)^\top$ for all q. Equivalently, on a simply connected domain, g should be parameterized as a gradient field $g = \nabla G$ for a scalar potential network G. The same condition applies to a coupled $h(p)$.*

C.4 The detailed proof of Theorem 4

Proof. Define $F : \mathbb{R}_{\geq 0} \to \mathbb{R}$ by $F'(s) = f(s)$, and consider the Hamiltonian $G(q,p) = -\frac{1}{2}F(q^2 + p^2)$. Hamilton's equations give

$$\dot{q} = \frac{\partial G}{\partial p} = -p\, f(r^2), \qquad \dot{p} = -\frac{\partial G}{\partial q} = q\, f(r^2),$$

where $r^2 = q^2 + p^2$. We claim that r^2 is a first integral:

$$\frac{d}{dt}(q^2 + p^2) = 2q\dot{q} + 2p\dot{p} = 2q \cdot (-pf(r^2)) + 2p \cdot (qf(r^2)) = 0.$$

Since r^2 is constant along the flow, we have $f(r^2) = f(r_0^2) = \text{const}$ along trajectories, and the dynamics reduces to a linear rotation with angular velocity $f(r_0^2)$:

$$\dot{q} = -f(r_0^2)\, p, \qquad \dot{p} = f(r_0^2)\, q.$$

The time-t flow of this system is rotation by angle $t\, f(r_0^2)$:

$$\Phi_G^t(q,p) = \big(q\cos(tf) - p\sin(tf),\ \ q\sin(tf) + p\cos(tf)\big).$$

Evaluating at $t = 1$:

$$\Phi_G^1(q,p) = \big(q\cos\alpha - p\sin\alpha,\ \ q\sin\alpha + p\cos\alpha\big), \qquad \alpha = f(r_0^2) = f(q^2 + p^2).$$

This is exactly $R(q,p)$. Since the time flow of any smooth Hamiltonian is symplectic, R is symplectic. □

C.5 The detailed proof of Theorem 5

Proof of Theorem 5. For **part (a)**, since A_θ is symplectic (Theorem 2) and ϵ-close to $A_\star$, it lies in a neighbourhood whose spectrum avoids $\mathbb{R}^-$ (because hypothesis (ii) ensures $A_\star$'s spectrum avoids $\mathbb{R}^-$, and the spectrum is continuous). For a real symplectic matrix with no eigenvalues on $\mathbb{R}^-$, the principal matrix logarithm exists and is Hamiltonian (Arslan, 2021), i.e. $X_\theta = \log A_\theta$ satisfies $X_\theta^\top J + J X_\theta = 0$.

This implies $-JX_\theta$ is symmetric (since $(-JX_\theta)^\top = X_\theta^\top(-J^\top) = X_\theta^\top J = -JX_\theta$, using the Hamiltonian condition). Define $\tilde{K} := -\frac{1}{\Delta t} JX_\theta$, which is symmetric. Then $A_\theta = e^{X_\theta} = e^{\Delta t\, J\tilde{K}}$, which is exactly the time-Δt flow of $\tilde{H}(z) = \frac{1}{2} z^\top \tilde{K} z$.

For the bound on $\tilde{K}$: the principal matrix logarithm is analytic on the set of matrices whose spectrum avoids $\mathbb{R}_-$, hence locally Lipschitz there (Al-Mohy et al., 2013). So

$$\|X_\theta - \log A_\star\| \le C_0 \|A_\theta - A_\star\| \le C_0 \epsilon. \tag{33}$$

Hypothesis (ii) ensures that $\log A_\star = \Delta t\, JK$ (i.e., $\Delta t\, JK$ is the principal logarithm of $A_\star$, not a branch-shifted one). Without this condition, the principal logarithm of $A_\star$ could differ from $\Delta t\, JK$ by branch wrapping. For example, a rotation by $4\pi/3$ has principal logarithm corresponding to rotation by $-2\pi/3$, not $4\pi/3$. With (ii) holding, we get

$$\|\tilde{K} - K\| = \frac{1}{\Delta t} \|J(\log A_\star - X_\theta)\| \le C\epsilon. \tag{34}$$

For **part (b)**, since A_θ is the exact flow of $\tilde{H}$, the modified Hamiltonian is exactly conserved

$$\tilde{H}(z_n) = \tilde{H}(z_0) \qquad \text{for all } n \ge 0. \tag{35}$$

The difference between the true and modified Hamiltonians is

$$|H(z) - \tilde{H}(z)| = \frac{1}{2} |z^\top (K - \tilde{K}) z| \le \frac{1}{2} \|K - \tilde{K}\| \cdot \|z\|^2. \tag{36}$$

By the triangle inequality,

$$\begin{aligned} |H(z_n) - H(z_0)| &\le |H(z_n) - \tilde{H}(z_n)| + |\tilde{H}(z_n) - \tilde{H}(z_0)| + |\tilde{H}(z_0) - H(z_0)| \\ &\le \frac{1}{2} R^2 \|K - \tilde{K}\| + 0 + \frac{1}{2} R^2 \|K - \tilde{K}\| = R^2 \|K - \tilde{K}\| \\ &\le R^2 C \epsilon. \end{aligned} \tag{37}$$

This bound is independent of n, i.e. there is no secular energy growth for all time. □

C.6 The detailed proof of Theorem 6

Proof. One direction to show that every Fourier Shear map is a symplectic Fourier multiplier is immediate from the architecture: the FFT is linear, each shear is a per-mode 2×2 linear map, and the composition of 2×2 symplectic matrices is symplectic (Lemma 1, Theorem 1). The IFFT is linear. So the full map is a block-diagonal symplectic Fourier multiplier.

For the converse, we must show every $M_k \in Sp(2, \mathbb{R})$ is realisable per mode. Write $M_k = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ with $ad - bc = 1$. One layer computes

$$S_3(w_3)\, S_2(w_2)\, S_1(w_1) = \begin{pmatrix} 1 + w_1 w_2 & w_2 \\ w_1 + w_3 + w_1 w_2 w_3 & 1 + w_2 w_3 \end{pmatrix}. \tag{38}$$

Case 1: $b \neq 0$. Set $w_2 = b$, $w_1 = (a-1)/b$, $w_3 = (d-1)/b$. Then:

The $(1,1)$ entry is $1 + w_1 w_2 = 1 + \frac{(a-1)}{b} \cdot b = a$.

The $(1,2)$ entry is $w_2 = b$.

The $(2,2)$ entry is $1 + w_2 w_3 = 1 + b \cdot \frac{(d-1)}{b} = d$.

The $(2,1)$ entry is $w_1 + w_3 + w_1 w_2 w_3 = \frac{a-1}{b} + \frac{d-1}{b} + \frac{(a-1)(d-1)}{b} = \frac{(a-1)+(d-1)+(a-1)(d-1)}{b} = \frac{ad-1}{b} = c$,

where the last equality uses $ad - bc = 1$, i.e., $ad - 1 = bc$. So one layer suffices.

Case 2: $b = 0$. Then $d = 1/a$ (since $ad = 1$) and $M_k = \left(\begin{smallmatrix} a & 0 \\ c & 1/a \end{smallmatrix}\right)$. No single layer can realise $b = 0$ with $a \neq 1$ (since $w_2 = b = 0$ forces the $(1, 1)$ entry to 1). Use two layers:

$$M_k = \underbrace{S_3(c/a)\, S_2(1)\, S_1(0)}_{\text{layer 2}} \; \underbrace{S_3(a-1)\, S_2(-1/a)\, S_1(a(1-a))}_{\text{layer 1}}. \tag{39}$$

Layer 1 with $w_1 = a(1 - a)$, $w_2 = -1/a$, $w_3 = a - 1$: the one-layer formula gives

$$L_1 = \begin{pmatrix} 1 + a(1-a)(-1/a) & -1/a \\ \cdots & 1 + (-1/a)(a-1) \end{pmatrix} = \begin{pmatrix} a & -1/a \\ 0 & 1/a \end{pmatrix}, \tag{40}$$

where the $(1, 1)$ entry simplifies as $1 - (1 - a) = a$, the $(2, 2)$ entry as $1 - (a - 1)/a = 1/a$, and the $(2, 1)$ entry equals 0 by the determinant constraint $\det L_1 = 1$ with $a \cdot (1/a) - (-1/a) \cdot (2, 1) = 1$.

Layer 2 with $w_1 = 0$, $w_2 = 1$, $w_3 = c/a$:

$$L_2 = \begin{pmatrix} 1 & 1 \\ c/a & 1 + c/a \end{pmatrix}. \tag{41}$$

The product gives:

$$L_2 L_1 = \begin{pmatrix} 1 & 1 \\ c/a & 1 + c/a \end{pmatrix} \begin{pmatrix} a & -1/a \\ 0 & 1/a \end{pmatrix} = \begin{pmatrix} a & 0 \\ c & 1/a \end{pmatrix} = M_k. \tag{42}$$

The $(1, 1)$ entry: $1 \cdot a + 1 \cdot 0 = a$. The $(1, 2)$ entry: $1 \cdot (-1/a) + 1 \cdot (1/a) = 0$. The $(2, 1)$ entry: $(c/a) \cdot a + (1 + c/a) \cdot 0 = c$. The $(2, 2)$ entry: $(c/a)(-1/a) + (1 + c/a)(1/a) = -c/a^2 + 1/a + c/a^2 = 1/a$.

Therefore every $M_k \in Sp(2, \mathbb{R})$ is realisable with at most two layers. Applying this independently to each mode $k < K$ and leaving higher modes as identity proves the result. □

C.7 The detailed proof of Theorem 7

Lemma 3 (Covering numbers of $Sp(2, \mathbb{R})$ vs $\mathbb{R}^{2\times 2}$)**.** *Let $B_B := \{M \in \mathbb{R}^{2\times 2} : \|M\|_{\text{op}} \leq B\}$, where $\|M\|_{\text{op}}$ is the operator norm. Then, for some constant $C_B > 0$ depending only on B, The η-covering numbers satisfy*

$$\begin{aligned} \log \mathcal{N}(\eta,\, Sp(2, \mathbb{R}) \cap B_B,\, \|\cdot\|_{\text{op}}) &\leq 3 \log\left(\frac{C_B}{\eta}\right), \\ \log \mathcal{N}(\eta,\, \mathbb{R}^{2\times 2} \cap B_B,\, \|\cdot\|_{\text{op}}) &\leq 4 \log\left(\frac{C_B}{\eta}\right). \end{aligned} \tag{43}$$

Proof. For the unconstrained class, $\mathbb{R}^{2\times 2} \cap B_B$ is a bounded subset of $\mathbb{R}^4$. A standard volumetric grid therefore yields $\log \mathcal{N}(\eta) \leq 4 \log(C_B/\eta)$.

Now consider $Sp(2, \mathbb{R}) \cap B_B$. Write $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, so $ad - bc = 1$ and $\|M\|_{\text{op}} \leq B$. Since $ad - bc = 1$, at least one of $|a|, |b|, |c|, |d|$ is at least $1/\sqrt{2}$; otherwise $|ad - bc| \leq |a||d| + |b||c| < 1$, a contradiction.

Thus the set is covered by four charts. On the a-chart, where $|a| \geq 1/\sqrt{2}$, use (a, b, c) as coordinates and recover $d = (1 + bc)/a$. Similarly, on the b-, c-, and d-charts, recover $c = (ad - 1)/b$, $b = (ad - 1)/c$, and $a = (1 + bc)/d$, respectively.

Because $\|M\|_{\text{op}} \leq B$, all coordinates lie in a bounded set, and each reconstruction map is Lipschitz on its chart. Hence each chart has 3-dimensional covering complexity, and

$$\log \mathcal{N}(\eta, \text{chart}) \leq 3 \log(C_B/\eta). \tag{44}$$

Taking the union over four charts only adds $\log 4$, which is absorbed into C_B. Therefore $\log \mathcal{N}(\eta, Sp(2,\mathbb{R}) \cap B_B, \|\cdot\|_{\mathrm{op}}) \le 3\log(C_B/\eta)$. □

Proof of Theorem 7. The key observation is that depth L does not enlarge the realized per-mode linear class. After any number of layers, each Fourier mode is represented by a single matrix $M_k \in Sp(2,\mathbb{R})$, since products of symplectic matrices remain symplectic. Hence the effective dimension is 3 per mode rather than $3L$.

Step 1. Metric transfer. Define the empirical metric $d_x(f_A, f_{A'}) = \left(\frac{1}{n}\sum_{i=1}^n \|(A-A')x_i\|^2\right)^{1/2}$. Then $d_x(f_A, f_{A'}) \le R_x \|A - A'\|_{\mathrm{op}}$. So operator-norm covers transfer to empirical covers.

Step 2. Product covering number. The full class is a product over K independent modes, so log-covering numbers add. Using the per-mode covering bounds (Lemma 3) gives

$$\begin{aligned} \log \mathcal{N}(\eta,\, \mathcal{A}_{\mathrm{symp}}(B)) &\le K \cdot 3\log(C_B/\eta), \\ \log \mathcal{N}(\eta,\, \mathcal{A}_{\mathrm{unc}}(B)) &\le K \cdot 4\log(C_B/\eta). \end{aligned} \tag{45}$$

Step 3. Dudley entropy integral. The Rademacher complexity is bounded by the Dudley integral

$$\mathfrak{R}_n(\mathcal{F}) \le \frac{12}{\sqrt{n}} \int_0^D \sqrt{\log \mathcal{N}(\eta, \mathcal{F})}\, d\eta \tag{46}$$

where $D = \sup_{f\in\mathcal{F}} \|f\|$. For classes with log-covering number $p\log(C/\eta)$, this integral evaluates to $O(\sqrt{p/n})$ (with constants absorbing B, R_x, L_ℓ, and logarithmic factors). Applying Lipschitz contraction for the loss gives the stated bound. □

C.8 The detailed proof of Theorem 8

Proof. Any $A \in Sp(2,\mathbb{R})$ can be written as $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ with $ad - bc = 1$. On inputs $(q, 0)$,

$$A(q,0) = (aq, cq). \tag{47}$$

Therefore

$$\mathbb{E}_\mu \|A(q,0) - \Phi_f(q,0)\|_2^2 = (a-1)^2 \mathbb{E}_\mu q^2 + \mathbb{E}_\mu (cq - f(q))^2. \tag{48}$$

The first term is minimized by $a = 1$, and this choice is compatible with the symplectic constraint by taking, for instance, $b = 0$ and $d = 1$. Minimizing over c gives the stated linear-ablation value.

For the nonlinear shear, the output on $(q, 0)$ is exactly $(q, g(q))$, so the loss is $\mathbb{E}_\mu(g(q) - f(q))^2$. If $f \in \mathcal{P}_d$, choose $g = f$. If f is continuous on a compact interval, the Weierstrass approximation theorem gives polynomials g converging uniformly to f, hence also converging in $L^2(\mu)$. Finally, the error of any linear map in A_{lin} is the squared $L^2(\mu)$ distance from f to the one-dimensional subspace $\mathrm{span}\{q\}$, which is positive exactly when f is not in its closure. □

C.9 The detailed proof of Theorem 9

Proof. At each spatial point i, the rotation R_i maps $(q_i, p_i) \mapsto (q_i', p_i')$ with:

$$q_i'^2 + p_i'^2 = (q_i \cos\alpha_i - p_i \sin\alpha_i)^2 + (q_i \sin\alpha_i + p_i \cos\alpha_i)^2. \tag{49}$$

Expanding:

$$\begin{aligned} q_i'^2 + p_i'^2 &= q_i^2 \cos^2\alpha_i - 2q_i p_i \cos\alpha_i \sin\alpha_i + p_i^2 \sin^2\alpha_i \\ &\quad + q_i^2 \sin^2\alpha_i + 2q_i p_i \sin\alpha_i \cos\alpha_i + p_i^2 \cos^2\alpha_i \\ &= q_i^2(\cos^2\alpha_i + \sin^2\alpha_i) + p_i^2(\sin^2\alpha_i + \cos^2\alpha_i) \\ &= q_i^2 + p_i^2. \end{aligned} \tag{50}$$

The cross terms cancel exactly. Summing over i gives $M' = \sum_i (q_i'^2 + p_i'^2) = \sum_i (q_i^2 + p_i^2) = M$. This holds regardless of α_i, so it holds for any parameter values. □

Table 3: Distributions of random initial conditions for each benchmark.

Problem	M	$k_{1,m}^{q,p}, k_{2,m}^{q,p}$	a_m^q	a_m^p
Harmonic Chain	$U(+2,+6)$	$U(+1,+4)$	$N(0,0.5^2)$	$N(0,0.3^2)$
Pendulum Chain	$U(+2,+6)$	$U(+1,+4)$	$N(0,0.5^2)$	$N(0,0.3^2)$
Shallow Water	$U(+2,+6)$	$U(-3,+3)$	$U(0,0.3)$	$U(0,0.2)$
Sine-Gordon	$U(+2,+6)$	$U(-3,+3)$	$U(0,0.3)$	$U(0,0.2)$
Schrödinger	$U(+2,+4)$	$U(-3,+3)$	$U(0,0.2)$	$U(0,0.2)$

D Experiment Details

We evaluate SympFNO on five representative Hamiltonian systems, as summarized in Table 1. The first two benchmarks are one-dimensional discrete chains. The Harmonic Chain describes a lattice of coupled oscillators with quadratic interaction energy, leading to linear Hamiltonian dynamics. This problem is useful for testing whether the proposed Fourier Shear layer can capture global linear wave propagation while preserving the underlying symplectic structure. The Pendulum Chain extends this setting by replacing the quadratic on-site potential with a nonlinear pendulum potential, producing nonlinear oscillatory dynamics while retaining a canonical Hamiltonian form. This benchmark therefore tests whether the nonlinear components of SympFNO can model nontrivial Hamiltonian effects beyond linear spectral evolution.

For two-dimensional problems, we consider three PDE benchmarks. The Shallow Water equation is included as a linear Hamiltonian problem, where the dynamics describe conservative wave propagation through the evolution of height and momentum-like variables. It provides a structured test case for evaluating long-range spatial interactions and energy behavior in two dimensions. The Nonlinear Schrödinger equation is a nonlinear dispersive Hamiltonian PDE whose state can be written in terms of real and imaginary components. In addition to its Hamiltonian structure, it possesses an important mass invariant, making it a suitable benchmark for testing the Physical Rotation layer. Finally, the Sine-Gordon equation is a nonlinear wave equation with a periodic potential, combining wave propagation with strongly nonlinear local interactions. Together, these five systems cover linear and nonlinear dynamics, one- and two-dimensional settings, and both discrete-chain and PDE-based Hamiltonian evolution.

Datasets are generated by numerically solving each benchmark system with the `dopri5` (Dormand and Prince, 1980) integrator implemented in `PyTorch` (Paszke et al., 2019) through the `torchdiffeq` package (Chen, 2018). This solver is used to produce reference trajectories, which serve as the ground-truth data for training and evaluation. All trajectories are defined on the periodic spatial domain $[0, 2\pi)$. For the one-dimensional benchmarks, we generate trajectories at spatial resolutions of 32, 64, and 128 grid points. For the two-dimensional benchmarks, trajectories are generated on both 32×32 and 256×256 grids. The high-resolution 256×256 data are then downsampled to 64×64 for training and evaluation, while predictions are interpolated back to 256×256 during testing. This follows the super-resolution setting of Li et al. (2021), where the model is trained on a coarser grid but evaluated against high-resolution reference solutions.

For each benchmark problem and each grid resolution, we generate 1000 independent trajectories from randomly sampled initial conditions. The initial states are generated as truncated Fourier series of the form

$$q_0(x,y),\, p_0(x,y) = \sum_{m=1}^{M} a_m^{q,p} \sin\!\left(k_{1,m}^{q,p} x + k_{2,m}^{q,p} y + \theta_m^{q,p}\right), \tag{51}$$

where $k_{2,m}^{q,p} = 0$ for one-dimensional problems. The number of modes, frequencies, amplitudes, and phases are sampled independently according to the random distributions and parameter ranges specified in Table 3. The resulting trajectories are split into 800 training trajectories, 100 validation trajectories, and 100 testing trajectories. Along each trajectory, the solution state is recorded every $\Delta t = 10^{-2}s$ for 300 time steps.

We train each model to learn the one-step solution map,

$$(q_t, p_t) \mapsto (q_{t+\Delta t}, p_{t+\Delta t}), \tag{52}$$

by minimizing mean squared error with the Adam optimizer (Kingma and Ba, 2014) in `PyTorch`. For PINO only, we additionally include its physics-informed loss term. We use batch size 1024, with

learning rate initialized at 10^{-3} and halved every 100 epochs. We evaluate performance using long-horizon autoregressive rollouts. We select the checkpoint with the lowest validation mean squared error. Since the benchmark systems are Hamiltonian, their exact flows preserve the Hamiltonian energy over time. Therefore, beyond trajectory prediction error, we also measure energy drift to assess whether the learned dynamics remain physically consistent during rollout. For the Nonlinear Schrödinger equation, we additionally report mass drift, since mass is an important conserved quantity for this system. These metrics are particularly important because small one-step errors may accumulate over many rollout steps, eventually leading to unstable or physically unrealistic solutions. All training and evaluation runs are repeated with three random seeds (42, 636, 792), and we report the mean and standard deviation across these runs.

We measure rollout accuracy by averaging the pointwise relative error $|u^{\mathrm{pred}} - u|/(|u| + \varepsilon)$ over both state components $u \in \{q, p\}$, all spatial grid points, and test trajectories, where u^{pred} and u denote predicted and reference values, respectively, and $\varepsilon = 2^{-23}$ is float32 machine epsilon. Error curves report this average at each rollout step; tabulated errors additionally average over the evaluated time steps, excluding the initial state.

The training of each model, including SympFNO and all baselines, was performed on a single NVIDIA RTX 6000 Ada Generation GPU with 48 GB of VRAM. On this hardware, completed comparison runs for 300 epochs with batch size 1024 required less than 5 hours per model and had recorded peak GPU memory usage below 8 GB per model.

D.1 Comparison with baselines

We compare SympFNO with several neural-operator and symplectic-network baselines. DirectFNO uses a standard FNO (Li et al., 2021) to directly predict the next state $(q_{t+\Delta t}, p_{t+\Delta t})$ from the current state (q_t, p_t). Since learning the update between two consecutive states is often easier than learning the full next-state map and can lead to more stable long-horizon rollouts, ResFNO instead trains the same FNO architecture in residual form. That is, the model predicts the increment from the current state to the next state. MiniResFNO is a compact version of ResFNO designed to have a parameter count comparable to that of SympFNO. For a fair comparison, the other baselines, DeepONet (Lu et al., 2021), PINO (Li et al., 2024), and ProjFNO, are also trained in residual form.

We also compare with LA-SympNet and G-SympNet (Jin et al., 2020). LA-SympNet uses three linear modules, each containing two linear shear sublayers, interleaved with two sigmoid activation modules. G-SympNet uses five gradient modules of width 30 with sigmoid activations. Both are evaluated on the Harmonic and Pendulum Chains at 32, 64, and 128 grid points. We restrict SympNet rollout evaluation to these 1D benchmarks, which we consider sufficient to illustrate the rollout behavior of the tested configurations.

The PINO baseline is built on the same FNO backbone as ResFNO, but augments the data-fitting objective with an additional physics-informed penalty. Specifically, we penalize the mismatch between the time derivatives implied by the governing equations evaluated at the predicted state and the finite-difference derivatives computed from the reference trajectory. The PDE residual loss is defined as

$$\mathcal{L}_{\mathrm{pde}} = \frac{1}{2N} \sum_{i=1}^{N} \begin{bmatrix} \left\| \dot{q}\left(q_t^{\mathrm{pred}}, p_t^{\mathrm{pred}}\right) - \dfrac{q_{t+\Delta t} - q_{t-\Delta t}}{2\Delta t} \right\|_2^2 \\ + \left\| \dot{p}\left(q_t^{\mathrm{pred}}, p_t^{\mathrm{pred}}\right) - \dfrac{p_{t+\Delta t} - p_{t-\Delta t}}{2\Delta t} \right\|_2^2 \end{bmatrix}, \tag{53}$$

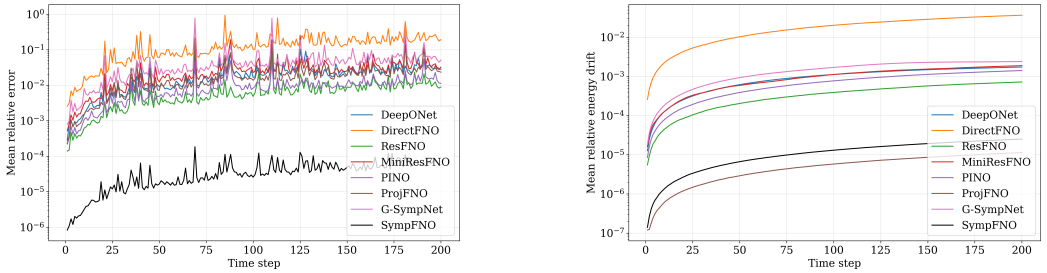
where $\dot{q}$ and $\dot{p}$ denote the right-hand sides of the Hamiltonian dynamics, and N is the number of training samples. The final training loss combines the data loss and the physics penalty as

$$\mathcal{L} = 0.9\, \mathcal{L}_{\mathrm{data}} + 0.1\, \mathcal{L}_{\mathrm{pde}}. \tag{54}$$

Thus, PINO is encouraged to match the observed data while also satisfying the underlying differential equations.

ProjFNO augments ResFNO with an energy-preserving projection step, analogous in spirit to the projection method of Baez et al. (2024). In our implementation, the predicted state is rescaled by the factor

$$\sqrt{\mathcal{H}_{\mathrm{in}}/\mathcal{H}_{\mathrm{pred}}}, \tag{55}$$



(a) Accuracy

(b) Energy drift

Figure 4: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 1D Harmonic Chain task with 32 grid points (seed 636).

so that the projected output has the same energy level as the input state. However, this projection is only valid for systems whose Hamiltonian is quadratic and homogeneous in the state variables, such as $\mathcal{H}(q, p) = aq^2 + bp^2$. Therefore, we apply ProjFNO only to the Harmonic Chain and Shallow Water benchmarks, where this simple rescaling preserves energy in the intended form.

All models are trained for 300 epochs using the configurations specified here. The FNO-based models and SympFNO use four layers. The FNO-based baselines, except MiniResFNO, use 12 Fourier modes in 1D and 4 Fourier modes per spatial dimension at 32^2 and 6 at 64^2 in 2D, with 16 hidden channels. For these models, the spatial coordinates are concatenated to the input channels as positional information. DeepONet uses width 256 with 128 latent features per output component in 1D, and width 2048 with 1024 latent features per output component in 2D to account for the larger spatial input dimension. Its branch and trunk networks each have four hidden layers.

For SympFNO, we use 12 Fourier modes for 1D problems. For 2D problems, we use 4 Fourier modes per spatial dimension for 32^2 grids, and 6 Fourier modes per spatial dimension for 64^2 grids. The nonlinear components are selected according to the structure of each system. For the linear problems, namely the Harmonic Chain and the Shallow Water equation, we use only the Fourier Shear layer. For the Pendulum Chain, we combine the Fourier Shear layer with a Nonlinear Shear layer of polynomial degree 10. The scale-down factor is set to 7.5 for all grid sizes. For the Sine-Gordon equation, we use the Fourier Shear layer together with a Nonlinear Shear layer of polynomial degree 10 and a scale-down factor of 7.5. For the Nonlinear Schrödinger equation, we use the Fourier Shear layer together with a Nonlinear Shear layer of polynomial degree 3 and a scale-down factor of 3. In addition, we include a Physical Rotation layer with polynomial degree 4 and a scale-down factor of 3 to better capture the equation's mass-preserving structure.

Tables 4 and 5 compare the baseline models in terms of rollout accuracy, energy drift, and mass drift over 200-step autoregressive rollouts. The results show that SympFNO consistently achieves the best rollout accuracy across all benchmark problems and grid resolutions, often by a substantial margin. In the SympNet comparison, the 1D empty entries correspond to non-finite LA-SympNet rollout metrics , while the 2D SympNet metrics are omitted because those cases were not evaluated. In terms of conservation behavior, SympFNO also outperforms most baselines in energy drift and mass drift, with only a few exceptions. ProjFNO obtains the smallest energy drift in several quadratic-energy cases because it explicitly projects the predicted state back onto the prescribed energy level. However, this improvement in energy conservation comes at the cost of noticeably lower rollout accuracy, and the method requires prior knowledge of the exact energy form, which limits its applicability mainly to systems with quadratic Hamiltonians. Overall, these results indicate that SympFNO provides the best balance between accurate long-term prediction and preservation of physical invariants among the compared baselines.

D.2 Architecture ablation

For the architecture ablation study, we evaluate the contribution of each SympFNO component across all five benchmark systems: Harmonic Chain, Pendulum Chain, Shallow Water, Nonlinear Schrödinger, and Sine-Gordon. For each problem, models are trained for 301 epochs and evaluated using 100-step autoregressive rollouts. We compare four architectural variants: Fourier Shear only;

Table 4: Comparison of baseline models in rollout accuracy, energy drift and mass drift. For each benchmark, the first row reports the mean relative rollout error over 200 time steps, the second row reports the final relative energy drift, and the third row, shown only for the Schrödinger equation, reports the final relative mass drift. The last row reports the number of real scalar model parameters. Metric values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	MiniResFNO	ResFNO	DeepONet	PINO	ProjFNO	**SympFNO**
Harmonic grid 32	(2.87 ± 0.28)E-02 (2.13 ± 0.27)E-03 1 058	(8.50 ± 0.68)E-03 (6.90 ± 0.49)E-04 26 946	(1.97 ± 0.20)E-02 (1.73 ± 0.07)E-03 543 491	(1.11 ± 0.09)E-02 (8.92 ± 3.71)E-04 26 946	(1.20 ± 0.64)E-02 $(\mathbf{9.07 \pm 1.84})$**E-06** 26 946	$(\mathbf{4.39 \pm 0.30})$**E-05** (2.42 ± 0.14)E-05 **144**
Harmonic grid 64	(2.69 ± 0.19)E-02 (2.80 ± 0.93)E-03 1 058	(9.91 ± 2.93)E-03 (7.14 ± 1.21)E-04 26 946	(2.61 ± 0.16)E-02 (2.55 ± 0.18)E-03 559 875	(1.45 ± 0.24)E-02 (6.82 ± 0.39)E-04 26 946	(2.00 ± 0.75)E-02 $(\mathbf{7.57 \pm 0.88})$**E-06** 26 946	$(\mathbf{8.91 \pm 0.92})$**E-05** (3.23 ± 0.45)E-05 **144**
Harmonic grid 128	(4.71 ± 0.53)E-02 (4.03 ± 1.68)E-03 1 058	(1.09 ± 0.26)E-02 (7.82 ± 1.78)E-04 26 946	(2.70 ± 0.07)E-02 (2.21 ± 0.10)E-03 592 643	(8.80 ± 1.12)E-03 (6.14 ± 0.72)E-04 26 946	(1.14 ± 0.31)E-02 $(\mathbf{8.24 \pm 5.13})$**E-06** 26 946	$(\mathbf{9.31 \pm 0.36})$**E-05** (4.26 ± 0.58)E-05 **144**
Pendulum grid 32	(8.75 ± 1.80)E-02 (4.96 ± 0.19)E-02 1 058	(2.46 ± 0.90)E-02 (3.81 ± 1.45)E-03 26 946	(1.25 ± 0.25)E-01 (5.48 ± 1.66)E-02 543 491	(2.02 ± 0.27)E-02 (4.69 ± 0.67)E-03 26 946	– – –	$(\mathbf{9.72 \pm 2.47})$**E-04** $(\mathbf{2.04 \pm 1.10})$**E-04** **224**
Pendulum grid 64	(6.86 ± 1.20)E-02 (7.37 ± 0.17)E-03 1 058	(1.67 ± 0.34)E-02 (8.91 ± 1.58)E-04 26 946	(1.42 ± 0.23)E-01 (1.36 ± 0.52)E-02 559 875	(1.72 ± 0.08)E-02 (9.62 ± 1.38)E-04 26 946	– – –	$(\mathbf{2.15 \pm 0.07})$**E-03** $(\mathbf{1.09 \pm 0.01})$**E-04** **224**
Pendulum grid 128	(2.74 ± 1.38)E-01 (7.00 ± 2.50)E-03 1 058	(1.73 ± 0.43)E-02 (7.21 ± 1.76)E-04 26 946	(1.53 ± 0.03)E-01 (1.20 ± 0.38)E-02 592 643	(1.74 ± 0.35)E-02 (6.53 ± 1.09)E-04 26 946	– – –	$(\mathbf{3.49 \pm 0.09})$**E-03** $(\mathbf{1.84 \pm 0.29})$**E-04** **224**
Shallow Water grid 32^2	(1.22 ± 0.32)E-01 (4.92 ± 0.25)E-03 4 422	(2.59 ± 0.14)E-02 (1.90 ± 0.55)E-03 67 314	(2.64 ± 0.03)E-01 (6.82 ± 0.44)E-03 37 773 315	(5.02 ± 0.65)E-02 (2.95 ± 0.27)E-02 67 314	(2.67 ± 0.12)E-02 $(\mathbf{1.02 \pm 0.08})$**E-06** 67 314	$(\mathbf{1.43 \pm 0.20})$**E-04** (8.90 ± 0.23)E-05 **336**
Shallow Water grid 64^2	(1.19 ± 0.27)E-01 (5.57 ± 1.28)E-03 4 422	(2.91 ± 0.31)E-02 (1.94 ± 0.24)E-03 149 234	(3.25 ± 0.05)E-01 (9.18 ± 0.68)E-03 50 356 227	(5.09 ± 0.66)E-02 (2.95 ± 0.10)E-02 149 234	(3.05 ± 0.21)E-02 $(\mathbf{1.50 \pm 0.43})$**E-06** 149 234	$(\mathbf{5.84 \pm 0.66})$**E-04** (5.03 ± 0.26)E-05 **792**
Shallow Water grid 256^2	(1.20 ± 0.22)E-01 (5.96 ± 1.05)E-03 4 422	(4.28 ± 0.17)E-02 (2.50 ± 0.42)E-03 149 234	(3.06 ± 0.03)E-01 (8.74 ± 0.44)E-03 50 356 227	(7.04 ± 0.54)E-02 (2.95 ± 0.06)E-02 149 234	(4.36 ± 0.13)E-02 $(\mathbf{1.88 \pm 0.01})$**E-03** 149 234	$(\mathbf{2.87 \pm 0.00})$**E-02** (1.89 ± 0.01)E-03 **792**
Sine-Gordon grid 32^2	(9.34 ± 1.21)E-02 (2.32 ± 0.23)E-03 4 422	(2.65 ± 0.43)E-02 (8.55 ± 1.98)E-04 67 314	(3.66 ± 0.02)E-01 (7.70 ± 1.44)E-03 37 773 315	(4.48 ± 0.34)E-02 (2.00 ± 0.03)E-02 67 314	– – –	$(\mathbf{1.17 \pm 0.08})$**E-02** $(\mathbf{2.37 \pm 0.01})$**E-04** **416**
Sine-Gordon grid 64^2	(8.88 ± 0.86)E-02 (2.48 ± 0.18)E-03 4 422	(2.56 ± 0.21)E-02 (1.07 ± 0.17)E-03 149 234	(7.31 ± 0.42)E-01 (1.90 ± 0.08)E-02 50 356 227	(4.96 ± 0.32)E-02 (1.99 ± 0.09)E-02 149 234	– – –	$(\mathbf{7.49 \pm 0.65})$**E-03** $(\mathbf{1.16 \pm 0.01})$**E-04** **872**
Sine-Gordon grid 256^2	(1.03 ± 0.11)E-01 (2.56 ± 0.72)E-03 4 422	(4.35 ± 0.21)E-02 (1.96 ± 0.35)E-03 149 234	(6.89 ± 0.29)E-01 (1.78 ± 0.23)E-02 50 356 227	(7.26 ± 0.31)E-02 (1.96 ± 0.07)E-02 149 234	– – –	$(\mathbf{3.02 \pm 0.02})$**E-02** $(\mathbf{1.40 \pm 0.00})$**E-03** **872**
Schrödinger grid 32^2	(2.61 ± 0.16)E-01 (1.08 ± 0.38)E-02 (9.57 ± 2.54)E-03 4 422	(1.33 ± 0.19)E-01 (1.20 ± 0.08)E-02 (1.17 ± 0.02)E-02 67 314	(2.94 ± 0.18)E-01 (1.02 ± 0.07)E-02 (8.19 ± 0.83)E-03 37 773 315	(7.54 ± 0.63)E-01 (3.91 ± 0.31)E-01 (3.05 ± 0.30)E-01 67 314	– – – –	$(\mathbf{8.92 \pm 0.17})$**E-02** $(\mathbf{5.88 \pm 0.70})$**E-04** $(\mathbf{6.30 \pm 1.47})$**E-04** **376**
Schrödinger grid 64^2	(2.65 ± 0.10)E-01 (1.02 ± 0.09)E-02 (9.21 ± 0.49)E-03 4 422	(1.06 ± 0.09)E-01 (6.68 ± 1.42)E-03 (6.59 ± 1.42)E-03 149 234	(1.52 ± 1.71)E+00 (8.08 ± 9.89)E-02 (5.97 ± 7.27)E-02 50 356 227	(7.71 ± 0.61)E-01 (3.77 ± 0.32)E-01 (2.91 ± 0.31)E-01 149 234	– – – –	$(\mathbf{7.34 \pm 0.69})$**E-02** $(\mathbf{5.54 \pm 0.40})$**E-04** $(\mathbf{4.53 \pm 0.49})$**E-04** **832**
Schrödinger grid 256^2	(2.75 ± 0.04)E-01 (1.08 ± 0.15)E-02 (1.03 ± 0.13)E-02 4 422	(1.15 ± 0.08)E-01 (7.11 ± 0.86)E-03 (6.48 ± 0.90)E-03 149 234	(1.58 ± 1.78)E+00 (8.41 ± 10.45)E-02 (5.99 ± 7.21)E-02 50 356 227	(8.07 ± 0.54)E-01 (4.14 ± 0.04)E-01 (3.26 ± 0.04)E-01 149 234	– – – –	$(\mathbf{8.48 \pm 0.44})$**E-02** $(\mathbf{4.68 \pm 0.42})$**E-04** $(\mathbf{5.27 \pm 0.82})$**E-04** **832**

Fourier Shear with Nonlinear Shear; Fourier Shear with Physical Rotation; and the full architecture combining Fourier Shear, Nonlinear Shear, and Physical Rotation. All variants use four SympFNO layers. For one-dimensional problems, we use 12 Fourier modes, while for two-dimensional problems we use 4 Fourier modes per spatial dimension on 32×32 grids and 6 modes per dimension on 64×64 grids. The nonlinear and rotation layers use polynomial degree 3 for both one- and two-dimensional problems, with scale-down factor 5 in both cases. This setup allows us to isolate the role of each proposed component and identify which layers are most important for linear, nonlinear, and mass-preserving Hamiltonian dynamics.

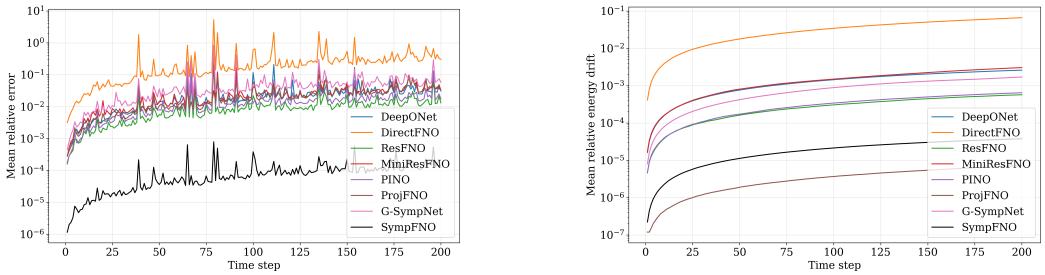
The numerical results are reported in Table 6 and are consistent with our theoretical analysis. For the linear Hamiltonian problems, namely the Harmonic Chain and Shallow Water equation, Theorems 7 and 6 indicate that Fourier Shear layers are sufficient to represent the relevant linear symplectic dynamics. The experiments support this prediction, with the Fourier-Shear-only model achieving the lowest rollout error and energy drift, while nonlinear components can make the dynamics harder to learn. For the nonlinear Hamiltonian problems, namely the Pendulum Chain and Sine-Gordon equation, adding Nonlinear Shear layers improves both trajectory accuracy and energy conservation, showing that these layers capture nonlinear Hamiltonian effects that cannot be represented by Fourier Shear alone. By contrast, adding Physical Rotation layers to these systems is usually unnecessary

Table 5: Comparison of baseline models in rollout accuracy, energy drift and mass drift. For each benchmark, the first row reports the mean relative rollout error over 200 time steps, the second row reports the final relative energy drift, and the third row, shown only for the Schrödinger equation, reports the final relative mass drift. The last row reports the number of real scalar model parameters. Metric values are mean $\pm$ population standard deviation across seeds 42, 636, and 792. The 256^2 rows use 64^2 model parameter counts.

Task	LA-SympNet	G-SympNet	DirectFNO	ResFNO	**SympFNO**
Harmonic grid 32	– – 6 400	(4.98 ± 0.18)E-02 (2.25 ± 0.11)E-03 5 100	(1.63 ± 0.16)E-01 (3.85 ± 0.44)E-02 26 946	(8.50 ± 0.68)E-03 (6.90 ± 0.49)E-04 26 946	$\mathbf{(4.39 \pm 0.30)}$**E-05** $\mathbf{(2.42 \pm 0.14)}$**E-05** **144**
Harmonic grid 64	– – 25 088	(5.05 ± 0.31)E-02 (1.70 ± 0.09)E-03 9 900	(2.33 ± 0.41)E-01 (6.92 ± 0.71)E-02 26 946	(9.91 ± 2.93)E-03 (7.14 ± 1.21)E-04 26 946	$\mathbf{(8.91 \pm 0.92)}$**E-05** $\mathbf{(3.23 \pm 0.45)}$**E-05** **144**
Harmonic grid 128	– – 99 328	(5.08 ± 0.19)E-02 (1.87 ± 0.15)E-03 19 500	(2.39 ± 0.19)E-01 (7.57 ± 1.44)E-02 26 946	(1.09 ± 0.26)E-02 (7.82 ± 1.78)E-04 26 946	$\mathbf{(9.31 \pm 0.36)}$**E-05** $\mathbf{(4.26 \pm 0.58)}$**E-05** **144**
Pendulum grid 32	(1.87 ± 1.48)E+02 (2.50 ± 1.79)E+05 6 400	(1.38 ± 0.14)E-01 (1.91 ± 0.08)E-02 5 100	(3.06 ± 3.22)E+00 (1.54 ± 2.18)E+04 26 946	(2.46 ± 0.90)E-02 (3.81 ± 1.45)E-03 26 946	$\mathbf{(9.72 \pm 2.47)}$**E-04** $\mathbf{(2.04 \pm 1.10)}$**E-04** **224**
Pendulum grid 64	– – 25 088	(1.99 ± 0.17)E-01 (7.34 ± 1.07)E-03 9 900	(2.20 ± 0.22)E-01 (5.25 ± 1.33)E-02 26 946	(1.67 ± 0.34)E-02 (8.91 ± 1.58)E-04 26 946	$\mathbf{(2.15 \pm 0.07)}$**E-03** $\mathbf{(1.09 \pm 0.01)}$**E-04** **224**
Pendulum grid 128	– – 99 328	(2.06 ± 0.10)E-01 (6.64 ± 1.21)E-03 19 500	(2.44 ± 0.14)E-01 (3.94 ± 1.03)E-02 26 946	(1.73 ± 0.43)E-02 (7.21 ± 1.76)E-04 26 946	$\mathbf{(3.49 \pm 0.09)}$**E-03** $\mathbf{(1.84 \pm 0.29)}$**E-04** **224**
Shallow Water grid 32^2	– – 6 299 648	– – 4 924 800	(4.02 ± 1.91)E-01 (4.90 ± 1.94)E-02 67 314	(2.59 ± 0.14)E-02 (1.90 ± 0.55)E-03 67 314	$\mathbf{(1.43 \pm 0.20)}$**E-04** $\mathbf{(8.90 \pm 0.23)}$**E-05** **336**
Shallow Water grid 64^2	– – 100 696 064	– – 39 340 800	(4.26 ± 1.37)E-01 (3.97 ± 0.18)E-02 149 234	(2.91 ± 0.31)E-02 (1.94 ± 0.24)E-03 149 234	$\mathbf{(5.84 \pm 0.66)}$**E-04** $\mathbf{(5.03 \pm 0.26)}$**E-05** **792**
Shallow Water grid 256^2	– – 100 696 064	– – 39 340 800	(4.11 ± 1.20)E-01 (4.72 ± 0.42)E-02 149 234	(4.28 ± 0.17)E-02 (2.50 ± 0.42)E-03 149 234	$\mathbf{(2.87 \pm 0.00)}$**E-02** $\mathbf{(1.89 \pm 0.01)}$**E-03** **792**
Sine-Gordon grid 32^2	– – 6 299 648	– – 4 924 800	(3.27 ± 0.22)E-01 (2.62 ± 0.87)E-02 67 314	(2.65 ± 0.43)E-02 (8.55 ± 1.98)E-04 67 314	$\mathbf{(1.17 \pm 0.08)}$**E-02** $\mathbf{(2.37 \pm 0.01)}$**E-04** **416**
Sine-Gordon grid 64^2	– – 100 696 064	– – 39 340 800	(3.63 ± 0.98)E-01 (2.68 ± 0.47)E-02 149 234	(2.56 ± 0.21)E-02 (1.07 ± 0.17)E-03 149 234	$\mathbf{(7.49 \pm 0.65)}$**E-03** $\mathbf{(1.16 \pm 0.01)}$**E-04** **872**
Sine-Gordon grid 256^2	– – 100 696 064	– – 39 340 800	(4.04 ± 1.15)E-01 (2.87 ± 0.80)E-02 149 234	(4.35 ± 0.21)E-02 (1.96 ± 0.35)E-03 149 234	$\mathbf{(3.02 \pm 0.02)}$**E-02** $\mathbf{(1.40 \pm 0.00)}$**E-03** **872**
Schrödinger grid 32^2	– – – 6 299 648	– – – 4 924 800	(3.10 ± 0.15)E-01 (1.03 ± 0.84)E-01 (1.18 ± 1.15)E-01 67 314	(1.33 ± 0.19)E-01 (1.20 ± 0.08)E-02 (1.17 ± 0.02)E-02 67 314	$\mathbf{(8.92 \pm 0.17)}$**E-02** $\mathbf{(5.88 \pm 0.70)}$**E-04** $\mathbf{(6.30 \pm 1.47)}$**E-04** **376**
Schrödinger grid 64^2	– – – 100 696 064	– – – 39 340 800	(2.36 ± 0.23)E-01 (4.88 ± 1.10)E-02 (4.39 ± 1.04)E-02 149 234	(1.06 ± 0.09)E-01 (6.68 ± 1.42)E-03 (6.59 ± 1.42)E-03 149 234	$\mathbf{(7.34 \pm 0.69)}$**E-02** $\mathbf{(5.54 \pm 0.40)}$**E-04** $\mathbf{(4.53 \pm 0.49)}$**E-04** **832**
Schrödinger grid 256^2	– – – 100 696 064	– – – 39 340 800	(2.61 ± 0.31)E-01 (1.02 ± 0.52)E-01 (1.00 ± 0.58)E-01 149 234	(1.15 ± 0.08)E-01 (7.11 ± 0.86)E-03 (6.48 ± 0.90)E-03 149 234	$\mathbf{(8.48 \pm 0.44)}$**E-02** $\mathbf{(4.68 \pm 0.42)}$**E-04** $\mathbf{(5.27 \pm 0.82)}$**E-04** **832**

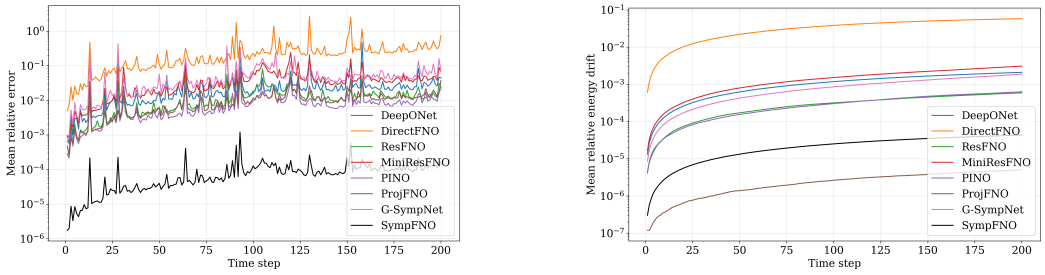
and can complicate training, sometimes leading to less stable rollouts. For the Nonlinear Schrödinger equation, however, the Physical Rotation layer is particularly beneficial because it is designed to preserve mass, as supported by Theorem 9 and Theorem 10. Including this layer alongside Nonlinear Shear improves energy and mass conservation with comparable prediction accuracy.

As discussed earlier, high-degree learned polynomial functions in the nonlinear layers can introduce instability when applied to problems where the corresponding architectural component is unnecessary. The empty entries reported in our examples for the full architecture and low scaledown factor on unsuitable problems serve as examples of this effect. In these cases, either the training is incomplete or the autoregressive rollout typically becomes unstable within the 100-step rollout horizon. This instability can be mitigated by training with multiple rollout steps or by selecting layers that better



(a) Accuracy (b) Energy drift

Figure 5: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 1D Harmonic Chain task with 64 grid points (seed 636).



(a) Accuracy (b) Energy drift

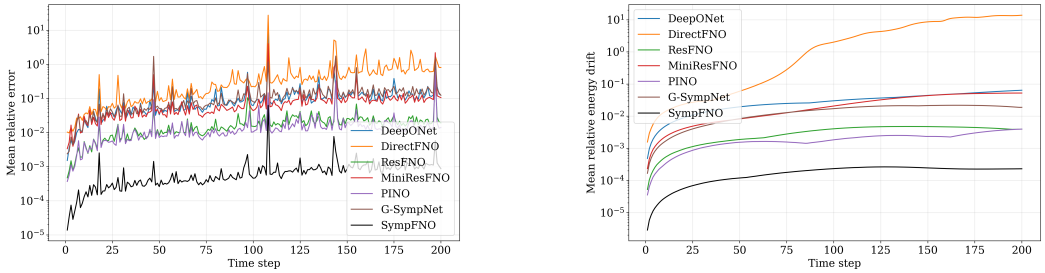
Figure 6: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 1D Harmonic Chain task with 128 grid points (seed 636).

match the structure of the target system, for example, by using only Fourier Shear for linear problems or combining Fourier Shear with Nonlinear Shear for standard nonlinear Hamiltonian problems. It can also be reduced by increasing the degree scale-down factor, as described in Remark 2.

D.3 Nonlinear Shear degree and scale-down sweep

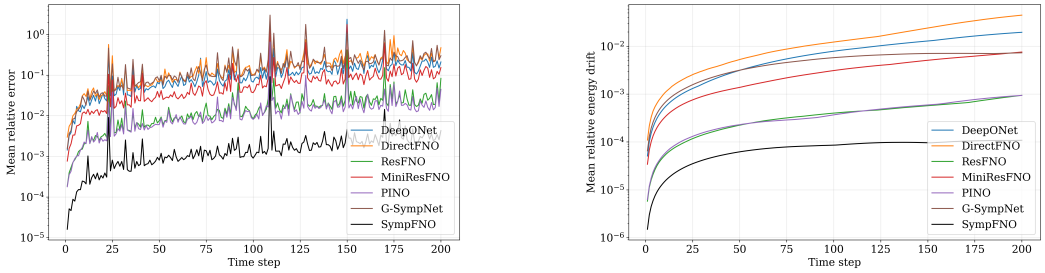
We further study the sensitivity of the Nonlinear Shear layers to two key hyperparameters: the polynomial degree and the degree scale-down factor. In this study, the architecture consists of Fourier Shear layers together with Nonlinear Shear layers, without the Physical Rotation layer. All models use four layers, are trained for 301 epochs, and are evaluated using 100-step autoregressive rollouts. For one-dimensional problems, we use 12 Fourier modes, while for two-dimensional problems we use 4 modes per spatial dimension on 32×32 grids and 6 modes per spatial dimension on 64×64 grids. For the degree sweep, we vary the polynomial degree over $\{2, 4, 6, 8, 10\}$ while fixing the scale-down factor to 5. For the scale-down sweep, we fix the polynomial degree to 6 and vary the scale-down factor over $\{3, 4, 5, 6, 7\}$. This setup allows us to examine how the expressiveness and rollout stability of the Nonlinear Shear layers depend on the order and scaling of the learned polynomial transformations.

Tables 7 and 8 report the sensitivity of the Nonlinear Shear layers to the polynomial degree and scale-down factor. Overall, the results show that these hyperparameters affect both rollout accuracy and conservation behavior, but the best choice depends on the benchmark problem. For linear systems such as the Harmonic Chain and Shallow Water equation, lower-complexity nonlinear layers are generally sufficient and often perform best, since the underlying dynamics do not require strong nonlinear expressiveness. For nonlinear systems such as the Pendulum Chain, Sine-Gordon equation, and Schrödinger equation, appropriately chosen degrees and scale-down factors can provide a better balance between expressiveness and stability. Very large or poorly scaled polynomial transformations do not consistently improve performance and may degrade rollout accuracy or conservation.



(a) Accuracy (b) Energy drift

Figure 7: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 1D Pendulum Chain task with 32 grid points (seed 636).



(a) Accuracy (b) Energy drift

Figure 8: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 1D Pendulum Chain task with 64 grid points (seed 636).

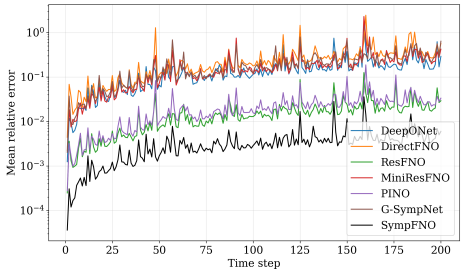
D.4 Physical Rotation degree and scale-down sweep

We also study the sensitivity of the Physical Rotation layer to its polynomial degree and degree scale-down factor. This experiment is analogous to the Nonlinear Shear degree and scale-down sweep, but here the architecture consists of Fourier Shear layers together with Physical Rotation layers, without the Nonlinear Shear layers. All models use four layers, are trained for 301 epochs, and are evaluated using 100-step autoregressive rollouts. For one-dimensional problems, we use 12 Fourier modes, while for two-dimensional problems we use 4 modes per spatial dimension on 32×32 grids and 6 modes per spatial dimension on 64×64 grids. For the degree sweep, we vary the Physical Rotation polynomial degree over $\{2, 4, 6, 8, 10\}$ while fixing the scale-down factor to 5. For the scale-down sweep, we fix the polynomial degree to 6 and vary the scale-down factor over the same range used in the Nonlinear Shear sweep. This setup allows us to evaluate how the expressiveness and stability of the Physical Rotation layer depend on the order and scaling of its learned amplitude-dependent rotations.

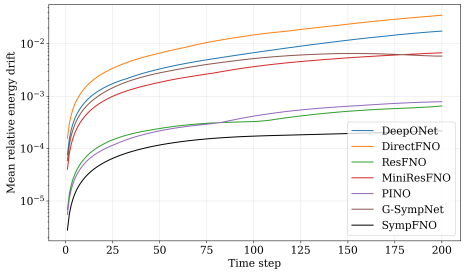
Tables 9 and 10 summarize the sensitivity of the Physical Rotation layer to its polynomial degree and scale-down factor. Overall, the results show that Physical Rotation is most useful for systems with rotation-like or mass-preserving structure, especially the Schrödinger equation, where performance remains stable across a range of settings. For other systems, the benefit is less consistent, and poorly chosen degrees or scale-down factors can lead to noticeably larger rollout errors and conservation drift. This suggests that the Physical Rotation layer should be used selectively, when its structure matches the target dynamics, and that moderate hyperparameter choices are generally preferable for stable long-term prediction.

D.5 Fourier modes, number of layers, and training data size sweep

We further evaluate the sensitivity of SympFNO to three model and data hyperparameters: the number of Fourier modes, the number of layers, and the amount of training data. For each benchmark, we use the architecture selected for its dynamics: Fourier Shear only for the linear Harmonic Chain

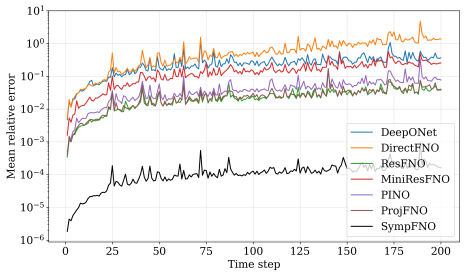


(a) Accuracy

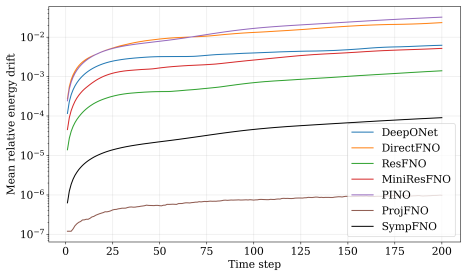


(b) Energy drift

Figure 9: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 1D Pendulum Chain task with 128 grid points (seed 636).



(a) Accuracy



(b) Energy drift

Figure 10: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 2D Shallow Water task with 32×32 grid points (seed 636).

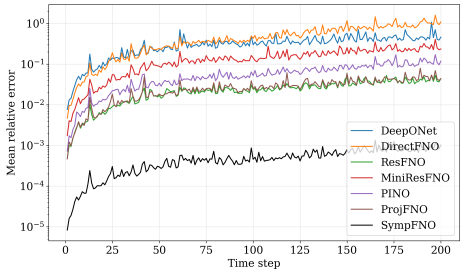
and Shallow Water equation, Fourier Shear with Nonlinear Shear for the Pendulum Chain and Sine-Gordon equation, and the full architecture with Fourier Shear, Nonlinear Shear, and Physical Rotation for the Schrödinger equation. All models are trained for 301 epochs and evaluated using 100-step autoregressive rollouts.

For the Fourier-mode sweep, we use grid resolutions 32 and 64 and vary the number of modes over $\{6, 8, 10, 12, 14\}$ for one-dimensional problems and $\{3, 4, 5, 6, 7\}$ per spatial dimension for two-dimensional problems, while keeping the number of layers fixed at four. For the layer-depth sweep, we vary the number of SympFNO layers over $\{2, 4, 6, 8\}$ using both grid resolutions 32 and 64. For the training-data sweep, we again use grid resolutions 32 and 64 and vary the number of training trajectories over $\{200, 400, 600, 800\}$, while keeping the architecture fixed. These experiments are designed to assess whether SympFNO requires large spectral bandwidth, deep architectures, or large amounts of training data to achieve stable long-horizon prediction.

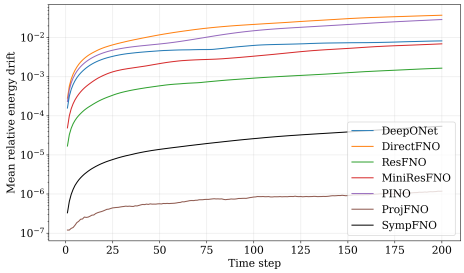
Tables 11,12, 13, and 14 summarize the sensitivity of SympFNO to Fourier modes, network depth, and training data size. Overall, the results show that SympFNO does not require very large spectral bandwidth, very deep architectures, or excessive training data to obtain stable long-horizon predictions.

For the Fourier-mode sweep, across the 1D benchmarks, the performance is relatively stable once a moderate number of modes is used, indicating that very large spectral bandwidth is not necessary for accurate long-horizon prediction. For the 2D benchmarks, however, using too few modes can substantially degrade the rollout accuracy, while increasing from four to seven modes approximately halves the mean rollout error for Sine-Gordon at both resolutions. This suggests that SympFNO benefits from sufficient low-frequency spectral resolution, but does not consistently improve by simply adding more modes.

For the layer-depth sweep, intermediate-depth networks, especially 4 or 6 layers, provide a reasonable compromise across the tested systems. Very shallow models can underfit several nonlinear or higher-dimensional systems, while very deep models may be harder to optimize.

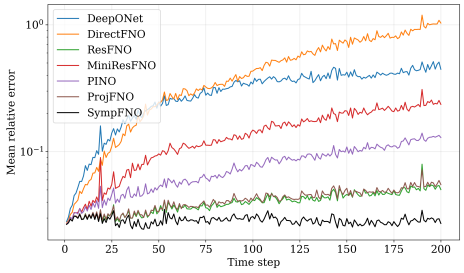


(a) Accuracy

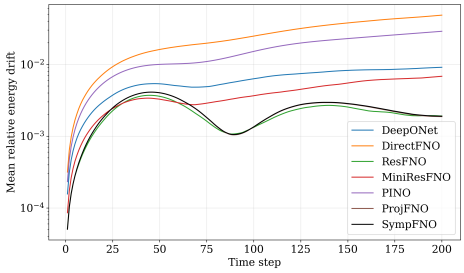


(b) Energy drift

Figure 11: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 2D Shallow Water task with 64×64 grid points (seed 636).



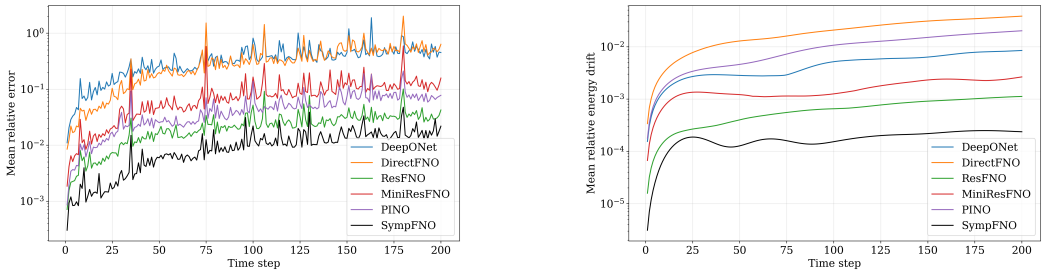
(a) Accuracy



(b) Energy drift

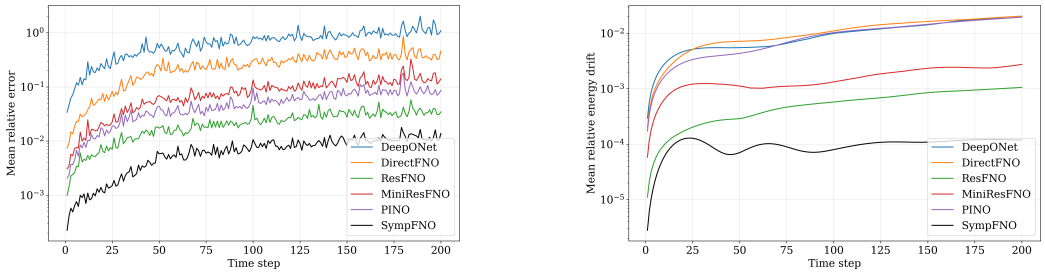
Figure 12: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 2D Shallow Water task with 256×256 grid points (seed 636).

For the training-data sweep, as expected, increasing the number of trajectories generally improves rollout accuracy and conservation behavior, with the improvement being most visible for hard problems such as Schrödinger equations. Nevertheless, several tasks already achieve competitive performance with smaller number of trajectories, showing that SympFNO can learn stable dynamics from a moderate amount of data.



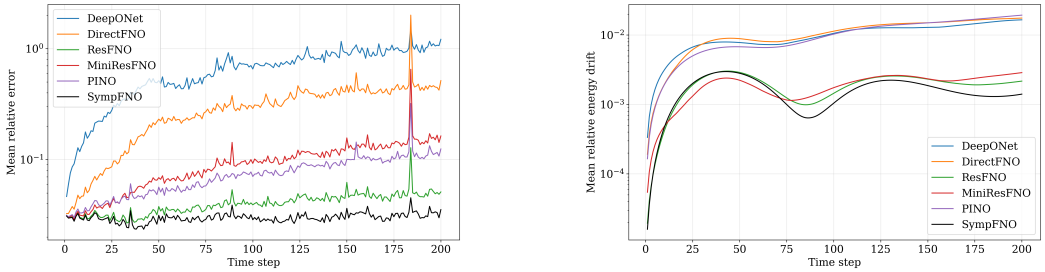
(a) Accuracy (b) Energy drift

Figure 13: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 2D Sine-Gordon task with 32×32 grid points (seed 636).



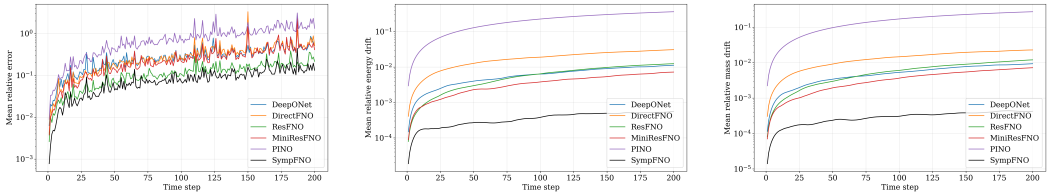
(a) Accuracy (b) Energy drift

Figure 14: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 2D Sine-Gordon task with 64×64 grid points (seed 636).



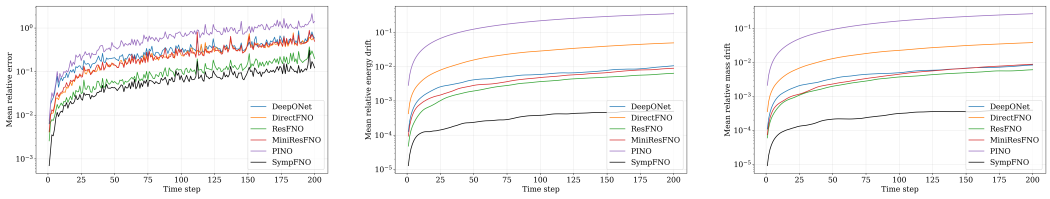
(a) Accuracy (b) Energy drift

Figure 15: Mean relative error (a) and mean relative energy drift (b) for each baseline against SympFNO across 200 rollout steps on the 2D Sine-Gordon task with 256×256 grid points (seed 636).



(a) Accuracy (b) Energy drift (c) Mass drift

Figure 16: Mean relative error (a), mean relative energy drift (b) and mean relative mass drift (c) for each baseline against SympFNO across 200 rollout steps on the 2D Schrödinger task with 32×32 grid points (seed 636).



(a) Accuracy (b) Energy drift (c) Mass drift

Figure 17: Mean relative error (a), mean relative energy drift (b) and mean relative mass drift (c) for each baseline against SympFNO across 200 rollout steps on the 2D Schrödinger task with 64×64 grid points (seed 636).

Table 6: Architecture ablation results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean $\pm$ population standard deviation across seeds 42, 636, and 792.

Task	No nonlinear layers	No Physical Rotation	No Nonlinear Shear	Full
Harmonic	$(\mathbf{2.34 \pm 0.06})$**E-05**	(8.12 ± 0.96)E-05	(9.32 ± 13.17)E+03	–
grid 32	$(\mathbf{1.13 \pm 0.05})$**E-05**	(3.82 ± 0.17)E-05	(6.85 ± 9.68)E+09	–
Harmonic	$(\mathbf{4.65 \pm 0.36})$**E-05**	(1.84 ± 0.19)E-04	(1.15 ± 0.31)E-01	–
grid 64	$(\mathbf{1.42 \pm 0.05})$**E-05**	(5.70 ± 0.14)E-05	(4.89 ± 1.56)E-02	–
Pendulum	(2.63 ± 0.12)E-01	(4.68 ± 0.51)E-02	(1.40 ± 0.25)E-01	$(\mathbf{2.06 \pm 0.12})$**E-02**
grid 32	(9.17 ± 1.60)E-02	(1.24 ± 0.34)E-02	(5.75 ± 2.02)E-02	$(\mathbf{5.86 \pm 1.68})$**E-03**
Pendulum	(3.79 ± 0.20)E-01	$(\mathbf{7.82 \pm 1.18})$**E-02**	(2.45 ± 0.55)E-01	–
grid 64	(3.40 ± 0.43)E-02	$(\mathbf{7.03 \pm 2.05})$**E-03**	(2.89 ± 3.74)E-01	–
Shallow Water	$(\mathbf{7.73 \pm 0.36})$**E-05**	(2.64 ± 0.14)E-04	(5.26 ± 2.02)E-04	(4.88 ± 1.48)E-04
grid 32^2	$(\mathbf{4.44 \pm 0.12})$**E-05**	(1.76 ± 0.04)E-04	(1.75 ± 0.03)E-04	(1.73 ± 0.02)E-04
Shallow Water	$(\mathbf{3.13 \pm 0.06})$**E-04**	(3.42 ± 0.03)E-04	(4.24 ± 0.53)E-04	(6.18 ± 0.25)E-04
grid 64^2	$(\mathbf{2.49 \pm 0.11})$**E-05**	(8.69 ± 0.11)E-05	(8.76 ± 0.12)E-05	(8.86 ± 0.23)E-05
Sine-Gordon	(1.42 ± 0.02)E-02	$(\mathbf{7.44 \pm 0.50})$**E-03**	(1.53 ± 0.08)E-02	(1.29 ± 0.16)E-02
grid 32^2	(1.37 ± 0.09)E-04	$(\mathbf{1.07 \pm 0.05})$**E-04**	(1.70 ± 0.13)E-04	(1.72 ± 0.42)E-04
Sine-Gordon	(1.23 ± 0.15)E-02	$(\mathbf{5.88 \pm 0.62})$**E-03**	(1.30 ± 0.12)E-02	(1.00 ± 0.07)E-02
grid 64^2	(1.49 ± 0.07)E-04	$(\mathbf{5.68 \pm 0.33})$**E-05**	(2.03 ± 0.29)E-04	(1.13 ± 0.24)E-04
Schrödinger	(1.73 ± 0.07)E-01	$(\mathbf{5.91 \pm 0.30})$**E-02**	(1.21 ± 0.04)E-01	(5.99 ± 0.16)E-02
grid 32^2	(1.09 ± 0.32)E-03	(4.18 ± 0.35)E-04	(8.49 ± 1.30)E-04	$(\mathbf{3.98 \pm 0.20})$**E-04**
	(1.23 ± 0.51)E-03	(4.40 ± 0.89)E-04	(1.34 ± 0.07)E-03	$(\mathbf{3.83 \pm 0.16})$**E-04**
Schrödinger	(1.71 ± 0.09)E-01	(4.17 ± 0.21)E-02	(5.66 ± 0.29)E-02	$(\mathbf{4.16 \pm 0.25})$**E-02**
grid 64^2	(1.16 ± 0.15)E-03	(4.07 ± 0.04)E-04	(4.66 ± 0.26)E-04	$(\mathbf{3.84 \pm 0.31})$**E-04**
	(1.48 ± 0.14)E-03	(4.38 ± 0.57)E-04	(3.90 ± 0.42)E-04	$(\mathbf{3.65 \pm 0.33})$**E-04**

Table 7: Nonlinear Shear degree sweep results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	Degree 2	Degree 4	Degree 6	Degree 8	Degree 10
Harmonic	(6.97 ± 0.85)E-05	$(\mathbf{6.81 \pm 0.92})$**E-05**	(7.66 ± 0.22)E-05	(2.20 ± 0.37)E-04	(3.82 ± 1.50)E-04
grid 32	(3.87 ± 0.06)E-05	(4.05 ± 0.11)E-05	$(\mathbf{3.82 \pm 0.23})$**E-05**	(4.40 ± 0.27)E-05	(7.11 ± 1.69)E-05
Harmonic	$(\mathbf{1.78 \pm 0.17})$**E-04**	(1.90 ± 0.30)E-04	(2.31 ± 0.28)E-04	(1.08 ± 0.21)E-03	(3.40 ± 1.85)E-03
grid 64	(6.14 ± 0.25)E-05	$(\mathbf{5.97 \pm 0.18})$**E-05**	(6.27 ± 0.76)E-05	(1.56 ± 0.65)E-04	(6.28 ± 2.73)E-04
Pendulum	(2.66 ± 0.13)E-01	(4.51 ± 0.38)E-02	(3.58 ± 0.47)E-03	(4.49 ± 1.13)E-04	$(\mathbf{2.78 \pm 0.33})$**E-04**
grid 32	(9.24 ± 1.42)E-02	(1.37 ± 0.19)E-02	(8.69 ± 2.63)E-04	(1.20 ± 0.30)E-04	$(\mathbf{8.68 \pm 1.09})$**E-05**
Pendulum	(3.99 ± 0.30)E-01	(5.29 ± 1.55)E-02	(1.67 ± 0.21)E-02	(3.60 ± 0.62)E-03	$(\mathbf{3.08 \pm 0.18})$**E-03**
grid 64	(3.33 ± 0.47)E-02	(3.96 ± 0.33)E-03	(1.16 ± 0.22)E-03	(1.98 ± 0.66)E-04	$(\mathbf{1.94 \pm 0.09})$**E-04**
Shallow Water	$(\mathbf{2.62 \pm 0.11})$**E-04**	(2.63 ± 0.13)E-04	(2.86 ± 0.33)E-04	(2.70 ± 0.18)E-04	(2.64 ± 0.11)E-04
grid 32^2	(1.76 ± 0.04)E-04	(1.76 ± 0.04)E-04	$(\mathbf{1.76 \pm 0.03})$**E-04**	(1.76 ± 0.04)E-04	(1.76 ± 0.04)E-04
Shallow Water	$(\mathbf{3.29 \pm 0.09})$**E-04**	(3.45 ± 0.08)E-04	(3.39 ± 0.09)E-04	(3.44 ± 0.16)E-04	(3.55 ± 0.23)E-04
grid 64^2	(8.86 ± 0.17)E-05	(8.85 ± 0.02)E-05	(9.02 ± 0.21)E-05	(8.80 ± 0.11)E-05	$(\mathbf{8.71 \pm 0.13})$**E-05**
Sine-Gordon	(1.74 ± 0.30)E-02	(7.61 ± 0.35)E-03	$(\mathbf{7.22 \pm 0.36})$**E-03**	(7.68 ± 0.53)E-03	(7.84 ± 0.41)E-03
grid 32^2	(1.95 ± 0.44)E-04	(1.06 ± 0.04)E-04	$(\mathbf{1.00 \pm 0.01})$**E-04**	(1.03 ± 0.02)E-04	(1.11 ± 0.04)E-04
Sine-Gordon	(1.35 ± 0.10)E-02	$(\mathbf{4.76 \pm 0.17})$**E-03**	(5.36 ± 0.53)E-03	(4.86 ± 0.36)E-03	(5.00 ± 0.43)E-03
grid 64^2	(1.40 ± 0.04)E-04	(5.75 ± 0.25)E-05	(5.98 ± 0.38)E-05	$(\mathbf{5.43 \pm 0.16})$**E-05**	(5.64 ± 0.21)E-05
Schrödinger	(1.41 ± 0.06)E-01	(5.92 ± 0.10)E-02	(6.16 ± 0.29)E-02	(6.05 ± 0.29)E-02	$(\mathbf{5.88 \pm 0.08})$**E-02**
grid 32^2	(9.62 ± 0.96)E-04	(4.32 ± 0.65)E-04	$(\mathbf{4.20 \pm 0.27})$**E-04**	(4.63 ± 1.01)E-04	(4.31 ± 0.58)E-04
	(1.34 ± 0.09)E-03	(5.16 ± 1.78)E-04	$(\mathbf{4.82 \pm 0.57})$**E-04**	(5.95 ± 2.52)E-04	(5.11 ± 1.52)E-04
Schrödinger	(1.41 ± 0.07)E-01	(4.11 ± 0.23)E-02	(4.36 ± 0.15)E-02	(4.12 ± 0.10)E-02	$(\mathbf{3.98 \pm 0.13})$**E-02**
grid 64^2	(1.06 ± 0.09)E-03	(3.55 ± 0.25)E-04	(4.13 ± 0.47)E-04	$(\mathbf{3.53 \pm 0.11})$**E-04**	(4.11 ± 0.25)E-04
	(1.49 ± 0.21)E-03	(3.12 ± 0.48)E-04	(4.33 ± 1.46)E-04	$(\mathbf{2.96 \pm 0.06})$**E-04**	(4.36 ± 1.18)E-04

Table 8: Nonlinear Shear scale-down sweep results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	Scale 3	Scale 4	Scale 5	Scale 6	Scale 7
Harmonic	–	(7.86 ± 0.90)E-05	(7.66 ± 0.22)E-05	(7.70 ± 0.57)E-05	$(\mathbf{6.84 \pm 0.57})$**E-05**
grid 32	–	$(\mathbf{3.75 \pm 0.15})$**E-05**	(3.82 ± 0.23)E-05	(3.86 ± 0.20)E-05	(3.82 ± 0.17)E-05
Harmonic	–	(7.54 ± 3.71)E-04	(2.31 ± 0.28)E-04	(3.13 ± 0.64)E-04	$(\mathbf{1.86 \pm 0.21})$**E-04**
grid 64	–	(1.00 ± 0.02)E-04	(6.27 ± 0.76)E-05	$(\mathbf{5.05 \pm 0.34})$**E-05**	(5.52 ± 0.16)E-05
Pendulum	(4.55 ± 0.79)E-03	(4.11 ± 0.69)E-03	$(\mathbf{3.58 \pm 0.47})$**E-03**	(3.99 ± 1.07)E-03	(3.64 ± 0.48)E-03
grid 32	(8.70 ± 1.57)E-04	(1.09 ± 0.38)E-03	(8.69 ± 2.63)E-04	(9.00 ± 2.52)E-04	$(\mathbf{8.60 \pm 2.06})$**E-04**
Pendulum	(1.52 ± 0.11)E-02	$(\mathbf{1.29 \pm 0.15})$**E-02**	(1.67 ± 0.21)E-02	(1.75 ± 0.14)E-02	(2.10 ± 0.28)E-02
grid 64	(9.46 ± 1.50)E-04	$(\mathbf{6.82 \pm 0.64})$**E-04**	(1.16 ± 0.22)E-03	(1.12 ± 0.13)E-03	(1.14 ± 0.16)E-03
Shallow Water	(2.59 ± 0.08)E-04	(2.62 ± 0.03)E-04	(2.86 ± 0.33)E-04	(2.69 ± 0.03)E-04	$(\mathbf{2.58 \pm 0.05})$**E-04**
grid 32^2	(1.76 ± 0.04)E-04	(1.76 ± 0.04)E-04	$(\mathbf{1.76 \pm 0.03})$**E-04**	(1.76 ± 0.04)E-04	(1.76 ± 0.04)E-04
Shallow Water	(3.32 ± 0.05)E-04	(3.34 ± 0.01)E-04	(3.39 ± 0.09)E-04	$(\mathbf{3.29 \pm 0.02})$**E-04**	(3.32 ± 0.12)E-04
grid 64^2	(8.99 ± 0.14)E-05	(9.03 ± 0.24)E-05	(9.02 ± 0.21)E-05	$(\mathbf{8.91 \pm 0.06})$**E-05**	(8.92 ± 0.14)E-05
Sine-Gordon	(7.25 ± 0.39)E-03	(7.20 ± 0.25)E-03	(7.22 ± 0.36)E-03	(7.21 ± 0.17)E-03	$(\mathbf{7.13 \pm 0.35})$**E-03**
grid 32^2	$(\mathbf{9.70 \pm 0.26})$**E-05**	(1.01 ± 0.01)E-04	(1.00 ± 0.01)E-04	(1.01 ± 0.03)E-04	(1.05 ± 0.02)E-04
Sine-Gordon	(5.24 ± 0.55)E-03	$(\mathbf{5.19 \pm 0.62})$**E-03**	(5.36 ± 0.53)E-03	(5.39 ± 0.50)E-03	(5.21 ± 0.53)E-03
grid 64^2	$(\mathbf{5.73 \pm 0.46})$**E-05**	(6.05 ± 0.49)E-05	(5.98 ± 0.38)E-05	(6.14 ± 0.52)E-05	(5.83 ± 0.55)E-05
Schrödinger	(6.16 ± 0.25)E-02	$(\mathbf{6.14 \pm 0.30})$**E-02**	(6.16 ± 0.29)E-02	(6.19 ± 0.24)E-02	(6.19 ± 0.26)E-02
grid 32^2	(4.27 ± 0.10)E-04	(4.22 ± 0.26)E-04	$(\mathbf{4.20 \pm 0.27})$**E-04**	(4.24 ± 0.21)E-04	(4.28 ± 0.20)E-04
	(4.97 ± 0.69)E-04	$(\mathbf{4.66 \pm 0.44})$**E-04**	(4.82 ± 0.57)E-04	(4.94 ± 0.69)E-04	(5.13 ± 0.92)E-04
Schrödinger	(4.36 ± 0.14)E-02	$(\mathbf{4.35 \pm 0.15})$**E-02**	(4.36 ± 0.15)E-02	(4.38 ± 0.15)E-02	(4.41 ± 0.14)E-02
grid 64^2	$(\mathbf{4.06 \pm 0.40})$**E-04**	(4.07 ± 0.45)E-04	(4.13 ± 0.47)E-04	(4.18 ± 0.49)E-04	(4.22 ± 0.51)E-04
	$(\mathbf{4.24 \pm 1.33})$**E-04**	(4.27 ± 1.44)E-04	(4.33 ± 1.46)E-04	(4.39 ± 1.44)E-04	(4.46 ± 1.38)E-04

Table 9: Physical Rotation degree sweep results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	Degree 2	Degree 4	Degree 6	Degree 8	Degree 10
Harmonic grid 32	$(\mathbf{8.58 \pm 12.06})$**E-02** $(\mathbf{5.81 \pm 8.21})$**E-01**	(2.28 ± 0.77)E+01 (1.09 ± 0.73)E+02	(1.93 ± 0.34)E+01 (3.56 ± 1.25)E+01	(1.94 ± 0.80)E+01 (5.89 ± 6.75)E+01	(1.16 ± 0.17)E+01 (8.47 ± 3.21)E+00
Harmonic grid 64	$(\mathbf{9.25 \pm 7.49})$**E-03** $(\mathbf{3.13 \pm 2.52})$**E-03**	(6.82 ± 5.16)E+01 (4.84 ± 6.01)E+03	(2.56 ± 0.66)E+01 (1.89 ± 1.46)E+02	(1.80 ± 0.51)E+01 (4.31 ± 2.11)E+01	(2.22 ± 0.45)E+01 (9.39 ± 4.01)E+01
Pendulum grid 32	$(\mathbf{6.35 \pm 0.47})$**E-02** $(\mathbf{2.48 \pm 1.03})$**E-02**	(1.90 ± 0.43)E-01 (1.52 ± 1.11)E-01	(1.99 ± 1.50)E+01 (1.33 ± 1.44)E+02	(1.93 ± 0.38)E+01 (1.71 ± 1.10)E+01	(1.68 ± 0.22)E+01 (1.34 ± 0.61)E+01
Pendulum grid 64	$(\mathbf{2.14 \pm 0.07})$**E-01** $(\mathbf{2.25 \pm 0.04})$**E-02**	(2.91 ± 1.79)E+01 (3.95 ± 4.19)E+01	(1.92 ± 0.82)E+01 (4.82 ± 3.36)E+00	(1.58 ± 0.25)E+01 (3.31 ± 1.03)E+00	(1.54 ± 0.16)E+01 (3.93 ± 1.04)E+00
Shallow Water grid 32^2	$(\mathbf{3.05 \pm 0.14})$**E-04** (1.75 ± 0.04)E-04	(1.65 ± 0.33)E-03 $(\mathbf{1.71 \pm 0.04})$**E-04**	(7.13 ± 2.00)E-03 (2.49 ± 0.15)E-04	(1.58 ± 0.63)E-02 (6.77 ± 2.13)E-04	(4.77 ± 5.94)E-01 (7.70 ± 5.30)E-02
Shallow Water grid 64^2	$(\mathbf{3.47 \pm 0.20})$**E-04** $(\mathbf{8.71 \pm 0.05})$**E-05**	(9.86 ± 1.96)E-04 (9.10 ± 0.20)E-05	(7.85 ± 1.59)E-03 (3.98 ± 0.94)E-04	(3.58 ± 2.12)E-02 (4.30 ± 3.30)E-03	(1.93 ± 2.68)E+00 (3.01 ± 4.26)E+01
Sine-Gordon grid 32^2	$(\mathbf{1.59 \pm 0.06})$**E-02** $(\mathbf{1.53 \pm 0.07})$**E-04**	(1.96 ± 0.10)E-02 (2.87 ± 0.21)E-04	(5.51 ± 1.38)E+00 (3.78 ± 2.39)E+01	(1.52 ± 0.56)E+01 (5.74 ± 3.58)E+01	(2.26 ± 0.44)E+01 (9.15 ± 3.38)E+01
Sine-Gordon grid 64^2	$(\mathbf{1.27 \pm 0.08})$**E-02** $(\mathbf{1.50 \pm 0.01})$**E-04**	(4.32 ± 0.47)E-02 (3.52 ± 2.26)E-03	(2.30 ± 2.24)E+01 (1.16 ± 1.16)E+03	(1.76 ± 0.12)E+01 (2.79 ± 1.71)E+02	(2.53 ± 0.67)E+01 (4.70 ± 3.61)E+02
Schrödinger grid 32^2	(1.19 ± 0.09)E-01 (9.15 ± 2.51)E-04 (1.48 ± 0.60)E-03	(1.22 ± 0.04)E-01 (8.56 ± 1.40)E-04 (1.17 ± 0.51)E-03	(1.18 ± 0.05)E-01 $(\mathbf{6.73 \pm 0.48})$**E-04** $(\mathbf{7.55 \pm 2.09})$**E-04**	$(\mathbf{1.16 \pm 0.05})$**E-01** (8.06 ± 1.55)E-04 (1.15 ± 0.52)E-03	(1.25 ± 0.01)E-01 (8.94 ± 1.64)E-04 (1.14 ± 0.11)E-03
Schrödinger grid 64^2	(5.75 ± 0.20)E-02 (4.74 ± 0.46)E-04 (4.49 ± 0.45)E-04	(5.87 ± 0.10)E-02 (5.14 ± 0.69)E-04 (5.19 ± 1.42)E-04	$(\mathbf{5.71 \pm 0.29})$**E-02** (5.38 ± 0.66)E-04 (4.60 ± 0.64)E-04	(5.85 ± 0.14)E-02 $(\mathbf{4.55 \pm 0.26})$**E-04** $(\mathbf{4.37 \pm 1.37})$**E-04**	(5.81 ± 0.14)E-02 (5.01 ± 0.49)E-04 (4.57 ± 1.42)E-04

Table 10: Physical Rotation scale-down sweep results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	Scale 3	Scale 4	Scale 5	Scale 6	Scale 7
Harmonic grid 32	$(\mathbf{1.41 \pm 0.61})$**E+01** $(\mathbf{1.37 \pm 1.02})$**E+01**	(1.63 ± 0.42)E+01 (2.33 ± 2.04)E+01	(1.93 ± 0.34)E+01 (3.56 ± 1.25)E+01	(1.84 ± 0.38)E+01 (3.10 ± 2.62)E+01	(1.65 ± 0.41)E+01 (2.87 ± 1.46)E+01
Harmonic grid 64	(1.61 ± 0.84)E+01 (7.72 ± 9.29)E+01	$(\mathbf{1.17 \pm 0.32})$**E+01** $(\mathbf{1.80 \pm 1.08})$**E+01**	(2.56 ± 0.66)E+01 (1.89 ± 1.46)E+02	(2.24 ± 1.44)E+01 (1.28 ± 1.45)E+02	(1.70 ± 0.23)E+01 (4.22 ± 1.71)E+01
Pendulum grid 32	(2.34 ± 1.48)E+01 (1.97 ± 2.72)E+02	(1.21 ± 0.32)E+01 (8.62 ± 3.59)E+00	(1.99 ± 1.50)E+01 (1.33 ± 1.44)E+02	(9.02 ± 6.27)E+00 (1.56 ± 1.30)E+01	$(\mathbf{3.35 \pm 3.48})$**E+00** $(\mathbf{1.91 \pm 1.86})$**E+00**
Pendulum grid 64	(2.11 ± 0.70)E+01 (1.01 ± 0.72)E+01	$(\mathbf{1.25 \pm 0.23})$**E+01** $(\mathbf{2.12 \pm 0.89})$**E+00**	(1.92 ± 0.82)E+01 (4.82 ± 3.36)E+00	(2.72 ± 1.25)E+01 (1.94 ± 2.00)E+01	(1.84 ± 0.24)E+01 (5.58 ± 2.24)E+00
Shallow Water grid 32^2	(7.14 ± 3.97)E-02 (2.51 ± 1.41)E-03	(1.34 ± 0.77)E-02 (7.75 ± 0.59)E-04	(7.13 ± 2.00)E-03 (2.49 ± 0.15)E-04	(3.10 ± 2.22)E-03 (2.05 ± 0.31)E-04	$(\mathbf{9.83 \pm 4.70})$**E-04** $(\mathbf{1.77 \pm 0.09})$**E-04**
Shallow Water grid 64^2	(1.52 ± 1.43)E-01 (3.53 ± 4.19)E-01	(1.44 ± 0.76)E-02 (9.99 ± 12.94)E-03	(7.85 ± 1.59)E-03 (3.98 ± 0.94)E-04	(1.99 ± 0.61)E-03 (1.31 ± 0.18)E-04	$(\mathbf{1.24 \pm 0.26})$**E-03** $(\mathbf{1.03 \pm 0.07})$**E-04**
Sine-Gordon grid 32^2	(1.47 ± 0.66)E+01 (4.85 ± 2.66)E+01	(1.53 ± 0.82)E+01 (7.84 ± 5.66)E+01	(5.51 ± 1.38)E+00 (3.78 ± 2.39)E+01	(1.16 ± 0.64)E-01 (3.54 ± 2.52)E-03	$(\mathbf{2.75 \pm 0.43})$**E-02** $(\mathbf{4.68 \pm 1.72})$**E-04**
Sine-Gordon grid 64^2	(2.30 ± 0.68)E+01 (4.45 ± 2.22)E+02	(1.41 ± 0.14)E+01 (3.47 ± 2.14)E+02	(2.30 ± 2.24)E+01 (1.16 ± 1.16)E+03	(9.68 ± 9.52)E+00 (4.53 ± 6.17)E+02	$(\mathbf{6.90 \pm 5.98})$**E-01** $(\mathbf{6.81 \pm 8.61})$**E+01**
Schrödinger grid 32^2	$(\mathbf{1.17 \pm 0.04})$**E-01** $(\mathbf{6.60 \pm 0.43})$**E-04** $(\mathbf{7.22 \pm 1.58})$**E-04**	(1.18 ± 0.05)E-01 (6.66 ± 0.43)E-04 (7.29 ± 1.65)E-04	(1.18 ± 0.05)E-01 (6.73 ± 0.48)E-04 (7.55 ± 2.09)E-04	(1.18 ± 0.05)E-01 (6.78 ± 0.54)E-04 (7.60 ± 2.23)E-04	(1.19 ± 0.05)E-01 (6.85 ± 0.59)E-04 (7.77 ± 2.33)E-04
Schrödinger grid 64^2	(5.67 ± 0.27)E-02 (4.80 ± 0.12)E-04 (4.15 ± 0.31)E-04	$(\mathbf{5.66 \pm 0.27})$**E-02** $(\mathbf{4.78 \pm 0.16})$**E-04** $(\mathbf{4.04 \pm 0.31})$**E-04**	(5.71 ± 0.29)E-02 (5.38 ± 0.66)E-04 (4.60 ± 0.64)E-04	(5.70 ± 0.26)E-02 (4.84 ± 0.14)E-04 (4.09 ± 0.22)E-04	(5.72 ± 0.32)E-02 (5.37 ± 0.68)E-04 (4.53 ± 0.53)E-04

Table 11: Fourier modes sweep of 1D problems results for rollout error and energy drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	6 modes	8 modes	10 modes	12 modes	14 modes
Harmonic	(2.47 ± 0.23)E-05	(2.50 ± 0.21)E-05	(2.47 ± 0.07)E-05	$(\mathbf{2.34 \pm 0.06})$**E-05**	(2.47 ± 0.42)E-05
grid 32	(1.13 ± 0.08)E-05	$(\mathbf{1.10 \pm 0.08})$**E-05**	(1.16 ± 0.01)E-05	(1.13 ± 0.05)E-05	(1.15 ± 0.05)E-05
Harmonic	(4.30 ± 0.37)E-05	(4.38 ± 0.43)E-05	(4.29 ± 0.40)E-05	(4.65 ± 0.36)E-05	$(\mathbf{4.15 \pm 0.26})$**E-05**
grid 64	(1.49 ± 0.13)E-05	(1.47 ± 0.03)E-05	(1.54 ± 0.05)E-05	$(\mathbf{1.42 \pm 0.05})$**E-05**	(1.53 ± 0.06)E-05
Pendulum	(5.50 ± 0.47)E-02	(4.91 ± 0.31)E-02	(4.45 ± 1.01)E-02	$(\mathbf{4.39 \pm 0.75})$**E-02**	(4.60 ± 0.74)E-02
grid 32	$(\mathbf{1.04 \pm 0.05})$**E-02**	(1.28 ± 0.39)E-02	(1.23 ± 0.15)E-02	(1.41 ± 0.39)E-02	(1.21 ± 0.13)E-02
Pendulum	(2.87 ± 0.11)E-02	(3.07 ± 0.26)E-02	(2.53 ± 0.33)E-02	(2.32 ± 0.25)E-02	$(\mathbf{2.14 \pm 0.08})$**E-02**
grid 64	(1.95 ± 0.12)E-03	(2.15 ± 0.33)E-03	(1.93 ± 0.11)E-03	$(\mathbf{1.86 \pm 0.26})$**E-03**	(2.01 ± 0.24)E-03

Table 12: Fourier modes sweep of 2D problems results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	3 modes	4 modes	5 modes	6 modes	7 modes
Shallow Water	(5.88 ± 0.14)E+00	(7.73 ± 0.36)E-05	(8.10 ± 0.70)E-05	(8.14 ± 0.36)E-05	$(\mathbf{7.51 \pm 0.43})$**E-05**
grid 32^2	$(\mathbf{4.31 \pm 0.08})$**E-05**	(4.44 ± 0.12)E-05	(4.45 ± 0.09)E-05	(4.47 ± 0.11)E-05	(4.46 ± 0.10)E-05
Shallow Water	(5.90 ± 0.14)E+00	$(\mathbf{2.43 \pm 0.96})$**E-04**	(2.91 ± 0.17)E-04	(3.13 ± 0.06)E-04	(3.34 ± 0.34)E-04
grid 64^2	(2.67 ± 0.28)E-05	$(\mathbf{2.40 \pm 0.16})$**E-05**	(2.70 ± 0.14)E-05	(2.49 ± 0.11)E-05	(2.57 ± 0.06)E-05
Sine-Gordon	(1.32 ± 0.05)E+00	(7.46 ± 0.48)E-03	(7.08 ± 0.40)E-03	(6.10 ± 0.76)E-03	$(\mathbf{3.84 \pm 0.10})$**E-03**
grid 32^2	(1.60 ± 0.05)E-02	(1.09 ± 0.05)E-04	(1.04 ± 0.06)E-04	(1.07 ± 0.04)E-04	$(\mathbf{1.01 \pm 0.02})$**E-04**
Sine-Gordon	(1.26 ± 0.06)E+00	(7.32 ± 0.43)E-03	(6.91 ± 0.52)E-03	(6.01 ± 0.71)E-03	$(\mathbf{3.70 \pm 0.10})$**E-03**
grid 64^2	(1.60 ± 0.05)E-02	(6.73 ± 0.18)E-05	(5.94 ± 0.80)E-05	(5.89 ± 0.40)E-05	$(\mathbf{5.24 \pm 0.14})$**E-05**
Schrödinger	(4.25 ± 0.02)E+00	(5.97 ± 0.18)E-02	$(\mathbf{2.95 \pm 0.07})$**E-02**	(4.06 ± 0.15)E-02	(4.20 ± 0.20)E-02
grid 32^2	(1.86 ± 0.53)E-03	(3.97 ± 0.20)E-04	$(\mathbf{2.65 \pm 0.33})$**E-04**	(3.69 ± 0.29)E-04	(5.36 ± 0.54)E-04
	(2.37 ± 0.76)E-03	(3.76 ± 0.18)E-04	$(\mathbf{3.75 \pm 0.59})$**E-04**	(4.00 ± 0.52)E-04	(3.88 ± 0.80)E-04
Schrödinger	(4.30 ± 0.09)E+00	(7.03 ± 1.05)E-02	$(\mathbf{3.02 \pm 0.04})$**E-02**	(4.17 ± 0.25)E-02	(4.21 ± 0.36)E-02
grid 64^2	(1.90 ± 0.53)E-03	(5.26 ± 1.43)E-04	$(\mathbf{2.38 \pm 0.49})$**E-04**	(3.85 ± 0.31)E-04	(5.80 ± 0.52)E-04
	(2.37 ± 0.77)E-03	(7.33 ± 4.91)E-04	$(\mathbf{3.49 \pm 0.96})$**E-04**	(3.65 ± 0.34)E-04	(3.71 ± 0.71)E-04

Table 13: Number of layers sweep results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	2 layers	4 layers	6 layers	8 layers
Harmonic	(2.46 ± 0.24)E-05	$(\mathbf{2.34 \pm 0.06})$**E-05**	(2.70 ± 0.17)E-05	(2.47 ± 0.30)E-05
grid 32	$(\mathbf{1.11 \pm 0.04})$**E-05**	(1.13 ± 0.05)E-05	(1.14 ± 0.04)E-05	(1.22 ± 0.06)E-05
Harmonic	$(\mathbf{4.44 \pm 0.42})$**E-05**	(4.65 ± 0.36)E-05	(4.74 ± 0.61)E-05	(4.69 ± 0.63)E-05
grid 64	(1.49 ± 0.08)E-05	$(\mathbf{1.42 \pm 0.05})$**E-05**	(1.59 ± 0.12)E-05	(1.61 ± 0.07)E-05
Pendulum	$(\mathbf{4.30 \pm 0.33})$**E-02**	(4.39 ± 0.75)E-02	(4.83 ± 0.53)E-02	(5.17 ± 0.23)E-02
grid 32	(1.24 ± 0.16)E-02	(1.41 ± 0.39)E-02	(1.21 ± 0.48)E-02	$(\mathbf{1.16 \pm 0.30})$**E-02**
Pendulum	(8.58 ± 0.66)E-02	(2.32 ± 0.25)E-02	(2.83 ± 0.32)E-02	$(\mathbf{2.17 \pm 0.01})$**E-02**
grid 64	(6.25 ± 1.40)E-03	(1.86 ± 0.26)E-03	(1.87 ± 0.20)E-03	$(\mathbf{1.51 \pm 0.05})$**E-03**
Shallow Water	(7.57 ± 0.37)E-05	(7.73 ± 0.36)E-05	$(\mathbf{7.41 \pm 0.15})$**E-05**	(7.94 ± 0.80)E-05
grid 32^2	(4.46 ± 0.12)E-05	$(\mathbf{4.44 \pm 0.12})$**E-05**	(4.49 ± 0.10)E-05	(4.52 ± 0.13)E-05
Shallow Water	(2.52 ± 0.56)E-04	(3.13 ± 0.06)E-04	$(\mathbf{2.25 \pm 0.20})$**E-04**	(2.59 ± 0.45)E-04
grid 64^2	(2.68 ± 0.21)E-05	$(\mathbf{2.49 \pm 0.11})$**E-05**	(2.50 ± 0.14)E-05	(2.72 ± 0.25)E-05
Sine-Gordon	$(\mathbf{6.19 \pm 0.05})$**E-03**	(7.46 ± 0.48)E-03	(7.24 ± 0.29)E-03	(7.71 ± 0.17)E-03
grid 32^2	$(\mathbf{4.54 \pm 0.11})$**E-05**	(1.09 ± 0.05)E-04	(1.53 ± 0.05)E-04	(1.98 ± 0.02)E-04
Sine-Gordon	$(\mathbf{3.92 \pm 0.12})$**E-03**	(6.01 ± 0.71)E-03	(5.31 ± 0.28)E-03	(4.77 ± 0.08)E-03
grid 64^2	$(\mathbf{2.09 \pm 0.10})$**E-05**	(5.89 ± 0.40)E-05	(7.77 ± 0.28)E-05	(8.96 ± 0.21)E-05
Schrödinger	(6.09 ± 0.59)E-02	(5.97 ± 0.18)E-02	$(\mathbf{5.82 \pm 0.18})$**E-02**	(5.95 ± 0.20)E-02
grid 32^2	(4.87 ± 0.87)E-04	$(\mathbf{3.97 \pm 0.20})$**E-04**	(4.37 ± 0.37)E-04	(4.78 ± 0.34)E-04
	(5.89 ± 2.73)E-04	$(\mathbf{3.76 \pm 0.18})$**E-04**	(5.74 ± 0.82)E-04	(5.67 ± 0.69)E-04
Schrödinger	(4.67 ± 0.17)E-02	(4.17 ± 0.25)E-02	(3.93 ± 0.09)E-02	$(\mathbf{3.78 \pm 0.10})$**E-02**
grid 64^2	(4.86 ± 0.26)E-04	(3.85 ± 0.31)E-04	$(\mathbf{3.54 \pm 0.26})$**E-04**	(3.64 ± 0.33)E-04
	$(\mathbf{3.09 \pm 0.93})$**E-04**	(3.65 ± 0.34)E-04	(4.05 ± 0.55)E-04	(4.47 ± 1.00)E-04

Table 14: Training data size sweep results for rollout error, energy drift, and mass drift. For each benchmark, the first row reports the mean relative rollout error over 100 time steps, and the second row reports the final relative energy drift. The third row, shown only for the Schrödinger equation, reports the final relative mass drift. Values are mean ± population standard deviation across seeds 42, 636, and 792.

Task	200 trajectories	400 trajectories	600 trajectories	800 trajectories
Harmonic	(3.93 ± 0.06)E-05	(2.99 ± 0.48)E-05	(2.41 ± 0.09)E-05	$(\mathbf{2.34 \pm 0.06})$**E-05**
grid 32	$(\mathbf{1.06 \pm 0.07})$**E-05**	(1.14 ± 0.11)E-05	(1.08 ± 0.06)E-05	(1.13 ± 0.05)E-05
Harmonic	(6.19 ± 2.20)E-05	(4.36 ± 0.29)E-05	$(\mathbf{4.33 \pm 0.28})$**E-05**	(4.65 ± 0.36)E-05
grid 64	(1.53 ± 0.44)E-05	(1.45 ± 0.03)E-05	(1.44 ± 0.09)E-05	$(\mathbf{1.42 \pm 0.05})$**E-05**
Pendulum	(5.10 ± 0.65)E-02	(5.48 ± 0.74)E-02	(4.94 ± 0.56)E-02	$(\mathbf{4.39 \pm 0.75})$**E-02**
grid 32	(1.26 ± 0.21)E-02	(1.36 ± 0.29)E-02	$(\mathbf{1.21 \pm 0.06})$**E-02**	(1.41 ± 0.39)E-02
Pendulum	(9.23 ± 1.38)E-02	(8.18 ± 0.49)E-02	(5.14 ± 0.16)E-02	$(\mathbf{2.32 \pm 0.25})$**E-02**
grid 64	(8.54 ± 1.23)E-03	(5.67 ± 0.20)E-03	(3.49 ± 0.11)E-03	$(\mathbf{1.86 \pm 0.26})$**E-03**
Shallow Water	(7.90 ± 0.19)E-05	(8.71 ± 0.25)E-05	$(\mathbf{7.58 \pm 0.29})$**E-05**	(7.73 ± 0.36)E-05
grid 32^2	(4.47 ± 0.08)E-05	(4.47 ± 0.15)E-05	(4.46 ± 0.12)E-05	$(\mathbf{4.44 \pm 0.12})$**E-05**
Shallow Water	$(\mathbf{1.39 \pm 0.07})$**E-04**	(1.85 ± 0.49)E-04	(2.57 ± 0.37)E-04	(3.13 ± 0.06)E-04
grid 64^2	$(\mathbf{2.33 \pm 0.12})$**E-05**	(2.41 ± 0.09)E-05	(2.47 ± 0.14)E-05	(2.49 ± 0.11)E-05
Sine-Gordon	(9.88 ± 3.05)E-03	(8.80 ± 0.37)E-03	(8.21 ± 0.20)E-03	$(\mathbf{7.46 \pm 0.48})$**E-03**
grid 32^2	$(\mathbf{1.08 \pm 0.39})$**E-04**	(1.12 ± 0.03)E-04	(1.11 ± 0.04)E-04	(1.09 ± 0.05)E-04
Sine-Gordon	$(\mathbf{4.21 \pm 0.23})$**E-03**	(4.27 ± 0.31)E-03	(5.16 ± 0.27)E-03	(6.01 ± 0.71)E-03
grid 64^2	(3.88 ± 0.27)E-05	$(\mathbf{3.65 \pm 0.38})$**E-05**	(5.51 ± 0.17)E-05	(5.89 ± 0.40)E-05
Schrödinger	(6.23 ± 0.38)E-02	(6.15 ± 0.25)E-02	(6.07 ± 0.24)E-02	$(\mathbf{5.97 \pm 0.18})$**E-02**
grid 32^2	(4.21 ± 0.20)E-04	(4.17 ± 0.21)E-04	(4.63 ± 0.53)E-04	$(\mathbf{3.97 \pm 0.20})$**E-04**
	(5.18 ± 0.79)E-04	(4.24 ± 0.53)E-04	(5.04 ± 0.84)E-04	$(\mathbf{3.76 \pm 0.18})$**E-04**
Schrödinger	(6.02 ± 0.20)E-02	(5.79 ± 0.11)E-02	(4.61 ± 0.08)E-02	$(\mathbf{4.17 \pm 0.25})$**E-02**
grid 64^2	(7.52 ± 0.91)E-04	(7.27 ± 0.24)E-04	(5.58 ± 0.31)E-04	$(\mathbf{3.85 \pm 0.31})$**E-04**
	(4.34 ± 0.81)E-04	(5.41 ± 1.52)E-04	(5.23 ± 0.98)E-04	$(\mathbf{3.65 \pm 0.34})$**E-04**

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 abstract and introduction accurately summarize the proposed SympFNO architecture, its symplectic guarantees, and its empirical evaluation on Hamiltonian benchmarks.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes].

   Justification: The Discussion section states that exact symplecticity does not guarantee exact conservation of all physical invariants. It also explains that SympFNO is mainly suited to closed Hamiltonian systems and is not guaranteed for dissipative systems.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: Each theorem states its assumptions and is accompanied by a complete proof in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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 benchmark systems, data generation, splits, training setup, baselines, and evaluation metrics.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes].

   Justification: Code for data generation, training, and evaluation of SympFNO and all baselines is provided in the supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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 train/validation/test split, optimizer, batch size, learning rate schedule, model settings, baselines, and rollout protocol.

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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 experiments are repeated with three random seeds. Tables report the mean and standard deviation across these runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: The paper reports that all models were trained on a single NVIDIA RTX 6000 Ada Generation GPU with 48 GB of VRAM. Completed comparison runs for 300 epochs with a batch size of 1024 took less than 5 hours per model and had recorded peak GPU memory usage below 8 GB per model.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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 synthetic PDE data and does not involve human subjects, private data, or high-risk deployment. We are not aware of any deviation from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

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 Broader Impacts section in Appendix discusses potential benefits for scientific surrogate modeling and states that no clear direct negative societal impact is identified.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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 models or scraped datasets. The experiments use synthetic numerical PDE data.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes].

    Justification: Existing methods and open-source software used in the paper are properly cited. Their licenses and terms of use are respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

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 submission does not release a new dataset, pretrained model, or software package as a standalone asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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, annotation labor, or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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 human subjects or personally identifiable data.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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: LLMs are not used as part of the proposed method, theory, experiments, or data generation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.