SHNO: SYMPLECTIC OPERATOR LEARNING WITH SIZE-INDEPENDENT PARAMETER COUNTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning Hamiltonian dynamics across system sizes requires reusable parameterizations that preserve symplectic structure. We introduce the Shared Hénon Neural Operator (SHNO), which combines a tied local encoder–decoder pair shared across canonical patches, a shared HénonNet for latent evolution, and optional symplectic edge exchange between patches. The assembled operator preserves the symplectic form on the represented manifold. For fixed local architecture and propagation depth, its parameter count is independent of patch count, while computation scales linearly with the numbers of patches and edges. For compatible local target compositions, we establish a size-uniform normalized approximation bound under uniform block-approximation and Lipschitz assumptions. Experiments on Klein–Gordon, Fermi–Pasta–Ulam–Tsingou, and Toda lattices evaluate prediction accuracy, symplectic structure, and parameter reuse. With separate training at each size, the 5,574-parameter SHNO without edge exchange achieves lower Klein–Gordon rollout errors than Global Hénon ROM, FNO, and a shared residual baseline. Matched unshared models use 8–32 times as many parameters, with accuracy differences depending on the system and size. Fixed-checkpoint tests evaluate deployment from 64 to 128 and 256 canonical pairs without target-size training. On Toda, edge exchange reduces geometric mean rollout error by 62.8% relative to identity coupling with 3.6% more parameters. Code is available anonymously at `https://anonymous.4open.science/r/SHNO-6547`.

1 INTRODUCTION

Hamiltonian systems often retain the same interaction laws as the number of interacting units changes. This repeated structure motivates learned flow maps that reuse parameters across system sizes. We seek to approximate evolution over a fixed time interval while preserving symplectic structure. The challenge is to assemble reusable components into a learned flow map that preserves the symplectic form on the represented state space.

Structure-preserving reduced-order models (ROMs) provide a foundation through symplectic representations and latent evolution (Peng & Mohseni, 2016; Chen et al., 2025). An encoder maps the physical state to latent coordinates, a symplectic latent propagator advances them, and a compatible decoder reconstructs the state. In full-state implementations designed for a fixed dimension, the encoder and decoder depend on that dimension. Neural operators and mesh-based simulators share learned computation across discretizations (Kovachki et al., 2023; Li et al., 2021; Pfaff et al., 2020), but parameter sharing alone does not ensure symplecticity of the learned flow map. How can shared components be assembled into a symplectic map over a variable number of interacting units?

We introduce Shared Hénon Neural Operator (SHNO), which retains the encoder–propagator–decoder composition and uses fixed-width canonical patches as reusable units. Each patch contains paired position and momentum coordinates and uses the same encoder, HénonNet for latent propagation, and tied decoder. The encoder and decoder are constructed from a common invertible symplectic transformation and its inverse. Optional symplectic edge maps with shared parameters couple latent patches through ordered edge updates. As illustrated in Figure 1, the patch count and interaction graph determine how these modules are assembled, while their trainable tensor shapes remain fixed.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The construction preserves the symplectic form on the represented manifold and has a parameter count independent of the number of patches. At fixed local architecture and propagation depth, computation scales with the number of patches and interaction edges. Experiments on Klein–Gordon (KG), Fermi–Pasta–Ulam–Tsingou (FPUT), and Toda lattices evaluate the prediction accuracy and parameter efficiency of this construction. With separate training at each size, the same 5,574-parameter SHNO architecture achieves lower 128-step KG errors than Global Hénon ROM, FNO, and a similarly sized shared residual operator. On FPUT at 128 canonical pairs, SHNO with edge exchange achieves the lowest mean error among the listed models, reducing error by 62.2% relative to Global Hénon ROM with 5,772 rather than 36,422 parameters (Table 1).

Our contributions are fourfold. **(1) Shared local symplectic construction.** A tied local encoder–decoder, a shared HénonNet, and optional symplectic edge maps preserve the symplectic form on the represented manifold. **(2) Scaling and approximation guarantees.** For fixed local architectures and propagation depth, SHNO uses $O(1)$ parameters in patch count K and $O(K + |\mathcal{E}_K|)$ work per step, where $\mathcal{E}_K$ is the patch interaction edge set. We characterize graph dependence and establish a size-uniform normalized approximation bound for compatible local target compositions under uniform approximation and Lipschitz assumptions. **(3) Parameter sharing and cross-size reuse.** Matched unshared controls require 8–32 times as many parameters over the tested sizes. Experiments quantify the accuracy trade-offs of sharing and deploying a checkpoint trained at 64 canonical pairs at 128 and 256 pairs without target-size training. **(4) Effective cross-patch communication.** On Toda, symplectic edge exchange yields a geometric mean edge-to-identity rollout-error ratio of 0.372 (95% CI: $[0.317, 0.445]$) with 3.6% more parameters.

2 Problem Setting and Design Goal

2.1 Hamiltonian dynamics and learning task

We consider a Hamiltonian system with canonical state

$$x = (q, p) \in \mathbb{R}^{2N}, \qquad q, p \in \mathbb{R}^N, \tag{1}$$

where q and p denote position and momentum, and N is the number of canonical pairs (degrees of freedom). For a smooth Hamiltonian $H : \mathbb{R}^{2N} \to \mathbb{R}$, the dynamics satisfy

$$\dot{x} = J_N \nabla H(x), \qquad J_N = \begin{bmatrix} 0 & I_N \\ -I_N & 0 \end{bmatrix}. \tag{2}$$

Let $F_h : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ denote the exact flow map over a time interval $h > 0$ of Eq. (2). On its domain of definition, this flow preserves the canonical symplectic form, $\big(\mathrm{D}F_h(x)\big)^\top J_N \, \mathrm{D}F_h(x) = J_N$.

Given trajectories sampled at interval h, we aim to learn a time-step map that approximates F_h and apply it repeatedly for autoregressive prediction. The model should be symplectic on its represented manifold, defined below, and support systems with different numbers of canonical pairs.

2.2 Symplectic reduced representations

A symplectic ROM approximates F_h by the composition

$$F_N^{\mathrm{ROM}} = D_N \circ \Phi_N \circ E_N, \tag{3}$$

where

$$E_N : \mathbb{R}^{2N} \to \mathbb{R}^{2r_N}, \quad \Phi_N : \mathbb{R}^{2r_N} \to \mathbb{R}^{2r_N}, \quad D_N : \mathbb{R}^{2r_N} \to \mathbb{R}^{2N}. \tag{4}$$

Here, $r_N \leq N$ is the number of latent canonical pairs. $\mathcal{M}_N = D_N(\mathbb{R}^{2r_N})$ denotes the represented manifold (the trial manifold). The encoder E_N maps physical states to latent coordinates, the latent map Φ_N advances those coordinates, and the decoder D_N reconstructs the physical state.

Let D_N be a symplectic embedding, with E_N restricted to $\mathcal{M}_N$ equal to its inverse:

$$\big(\mathrm{D}D_N(z)\big)^\top J_N \, \mathrm{D}D_N(z) = J_{r_N}, \qquad E_N \circ D_N = \mathrm{id}_{\mathbb{R}^{2r_N}}. \tag{5}$$

If Φ_N is also symplectic, then F_N^{ROM} preserves the symplectic form induced on $\mathcal{M}_N$ by the ambient canonical form. Indeed, for a represented state $x = D_N(z)$, $F_N^{\mathrm{ROM}}(x) = D_N(\Phi_N(z))$. Thus, the decoder transports the latent symplectic dynamics to the represented manifold.

Meanwhile, in a dimension-specific full-state implementation, the encoder and decoder are parameterized for a fixed state dimension. SHNO retains this composition principle and employs shared local maps to extend the parameterization across system sizes.

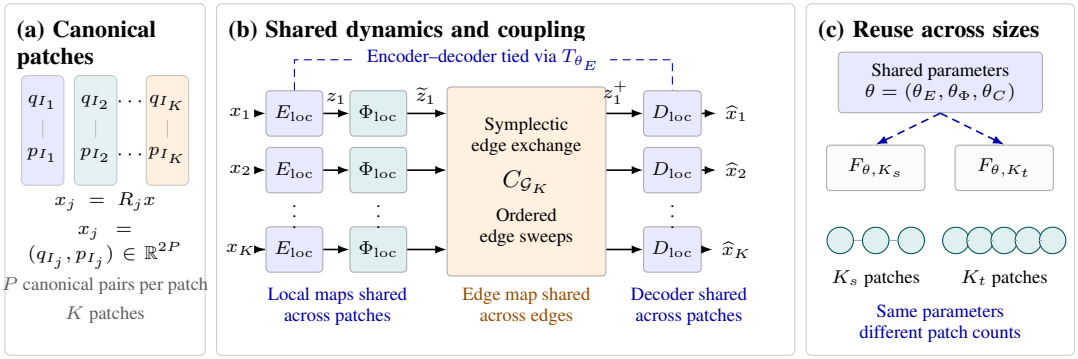

Figure 1: SHNO architecture. (a) The state is partitioned into canonical patches containing paired position and momentum coordinates. (b) Each patch uses the same local encoder, Hénon latent map, and tied decoder. The joint latent map $C_{\mathcal{G}_K}$ composes symplectic edge updates to exchange information between patches, with edge-map parameters shared across edges. The encoder uses a shared symplectic transformation T_{θ_E}, and the decoder uses its inverse. (c) The same parameter set θ defines the complete operator at different patch counts. Dashed arrows indicate parameter reuse.

2.3 Canonical patches and size-independent parameterization

We consider systems with recurring local interactions and organize their states into K canonical patches of fixed width P. Let $\mathcal{P} = (I_1, \ldots, I_K)$, where $I_j = (i_{j,1}, \ldots, i_{j,P})$ specifies the ordered coordinate indices of patch j. The underlying index sets form a disjoint partition of $\{1, \ldots, N\}$, so $KP = N$. Define the patch restriction and stacked representation by

$$R_j x = \big(q_{I_j}, p_{I_j}\big) \in \mathbb{R}^{2P}, \qquad \mathcal{R}_{\mathcal{P}} x = \big(R_1 x, \ldots, R_K x\big) \in (\mathbb{R}^{2P})^K, \tag{6}$$

where $q_{I_j} = (q_{i_{j,1}}, \ldots, q_{i_{j,P}})$, and p_{I_j} uses the same index order. Each patch therefore contains P complete position–momentum pairs.

The patch interaction graph is

$$\mathcal{G}_K = (\mathcal{V}_K, \mathcal{E}_K), \qquad \mathcal{V}_K = \{1, \ldots, K\}. \tag{7}$$

Vertex j represents patch I_j, and an undirected edge $\{i, j\} \in \mathcal{E}_K$ specifies an available communication link between patches i and j. The graph is supplied as part of the assembly description. For a chain divided into consecutive patches, adjacent patches form a path graph: $\mathcal{E}_K = \{\{j, j+1\} : 1 \le j < K\}$. The partition, graph, and coordinate ordering determine the assembly. Edge orientations and update schedules are specified by the coupling construction in Section 3.

Our design goal is a family of time-step models

$$F_{\theta,K} : \mathbb{R}^{2KP} \to \mathbb{R}^{2KP} \tag{8}$$

whose trainable parameter dimensions are independent of K for a fixed local architecture. Each model should preserve the symplectic form on its represented manifold $\mathcal{M}_K$.

Cross-size deployment uses the same learned parameter vector θ in F_{θ,K_s} and F_{θ,K_t}, with fixed local dimensions and coordinate conventions. We call this deployment zero-shot when θ is learned at the source size and reused without target-size training or parameter updates.

3 SHNO: Shared Local Symplectic Operators

3.1 Overview

SHNO retains the encoder–propagator–decoder composition introduced in Section 2 and builds the model from canonical patches. Every patch uses the same local representation and latent propagator, while a shared symplectic edge map exchanges information among patches. The assembled time-step model is

$$F_{\theta,K} = D_K \circ C_{\theta_C, \mathcal{G}_K, \sigma_K} \circ \Phi_K \circ E_K, \tag{9}$$

where E_K encodes the physical patches, Φ_K advances their latent states, $C_{\theta_C,\mathcal{G}_K,\sigma_K}$ performs edge exchange, and D_K reconstructs the global physical state. The schedule σ_K specifies edge orientations and the order of edge updates.

The parameter vector is $\theta = (\theta_E, \theta_\Phi, \theta_C)$. The encoder and decoder are tied through θ_E; θ_Φ parameterizes the shared local dynamics; and θ_C parameterizes the shared edge block. Figure 1 illustrates how these modules are assembled at different patch counts.

3.2 Tied local symplectic representations

For a patch state $x_j = R_j x \in \mathbb{R}^{2P}$, let $z_j = (a_j, b_j) = E_{\mathrm{loc}}(x_j) \in \mathbb{R}^{2r}, \quad a_j, b_j \in \mathbb{R}^r$, where $r \le P$ is the number of latent canonical pairs. We construct the local encoder and decoder from a common invertible symplectic transformation.

For $u, v, \eta \in \mathbb{R}^m$ and a twice continuously differentiable scalar potential $V : \mathbb{R}^m \to \mathbb{R}$, the elementary Hénon map and its inverse are

$$\mathcal{H}_{V,\eta}(u,v) = \big(v + \eta, -u + \nabla V(v)\big), \quad \mathcal{H}^{-1}_{V,\eta}(u,v) = \big(\nabla V(u-\eta) - v, u - \eta\big). \tag{10}$$

These maps preserve the canonical symplectic form. A Hénon layer applies its elementary map four times, and a HénonNet composes multiple layers (Burby et al., 2021).

We also employ symplectic G-reflectors (Mackey & Mackey, 2003), parametrized by $u \in \mathbb{R}^{2m}$ and $\beta \in \mathbb{R}$. For an input $y \in \mathbb{R}^{2m}$, a G-reflector is defined as:

$$\mathcal{R}_{u,\beta}(y) = y + \beta(u^\top J_m y)u. \tag{11}$$

This linear rank-one update is symplectic with inverse $\mathcal{R}_{u,-\beta}$, and every linear symplectic map on $\mathbb{R}^{2m}$ is a finite composition of such updates.

Let $T_{\theta_E} : \mathbb{R}^{2P} \to \mathbb{R}^{2P}$ compose Hénon layers and G-reflectors with $m = P$. Let $\pi_r : \mathbb{R}^{2P} \to \mathbb{R}^{2r}$ be the canonical projection onto the first r positions and their paired momenta, and $\iota_r : \mathbb{R}^{2r} \to \mathbb{R}^{2P}$ the canonical embedding that inserts them with zeros in the remaining coordinates. The local representation is

$$E_{\mathrm{loc}} = \pi_r \circ T_{\theta_E}, \qquad D_{\mathrm{loc}} = T^{-1}_{\theta_E} \circ \iota_r, \tag{12}$$

satisfying

$$E_{\mathrm{loc}} \circ D_{\mathrm{loc}} = \mathrm{id}_{\mathbb{R}^{2r}}, \qquad \big(\mathrm{D}D_{\mathrm{loc}}(z)\big)^\top J_P \,\mathrm{D}D_{\mathrm{loc}}(z) = J_r. \tag{13}$$

Thus, the decoder D_{loc} is a symplectic embedding into $\mathcal{M}_{\mathrm{loc}} = D_{\mathrm{loc}}(\mathbb{R}^{2r})$, and E_{loc} is its symplectic inverse. The reconstruction $D_{\mathrm{loc}} \circ E_{\mathrm{loc}}$ maps patch states onto $\mathcal{M}_{\mathrm{loc}}$ and leaves states on this manifold unchanged. We let all patches share θ_E.

3.3 Shared dynamics and symplectic edge exchange

Local evolution. Each latent patch advances independently as $\widetilde{z}_j = \Phi_{\mathrm{loc}}(z_j)$ using a HénonNet $\Phi_{\mathrm{loc}} : \mathbb{R}^{2r} \to \mathbb{R}^{2r}$ with parameters θ_Φ shared across patches. Optional edge exchange couples neighboring patches; identity coupling ($C = \mathrm{id}$) leaves them independent.

Shared edge block. Initialize the working states by $w_j = \widetilde{z}_j$. For an oriented edge (i, j), write its current states as $w_i = (a_i, b_i)$ and $w_j = (a_j, b_j)$. The edge block groups their position coordinates before their momentum coordinates:

$$y_{ij} = \mathrm{col}(a_i, a_j, b_i, b_j) \in \mathbb{R}^{4r}. \tag{14}$$

Thus the edge state contains $2r$ canonical pairs and uses J_{2r}. A shared map B_{θ_C} updates y_{ij}; its output, (a'_i, a'_j, b'_i, b'_j), is split back into the two patch states, updating $w_i = (a'_i, b'_i)$ and $w_j = (a'_j, b'_j)$.

The block composes d G-reflectors from Eq. (11), now with $m = 2r$:

$$B_{\theta_C} = \mathcal{R}_{u_d,\beta_d} \circ \cdots \circ \mathcal{R}_{u_1,\beta_1}, \tag{15}$$

with the ordered parameter tuple $\theta_C = ((u_k, \beta_k))_{k=1}^d$ shared across all edges. These parameters are separate from θ_E. The block has $d(4r + 1)$ parameters, independently of patch count.

Ordered edge updates. To schedule updates on edges sharing a patch, we partition $\mathcal{E}_K$ into c matchings $\mathcal{E}_K^{(1)}, \ldots, \mathcal{E}_K^{(c)}$, each containing edges with disjoint endpoints. A sweep U_ℓ applies B_{θ_C} in parallel to all edges in matching ℓ and leaves unmatched patches unchanged. Successive sweeps

use the updated patch states. The schedule σ_K records the matching order and edge orientations. Repeating this sequence for L rounds gives

$$C_{\theta_C,\mathcal{G}_K,\sigma_K} = (U_c \circ \cdots \circ U_1)^L. \tag{16}$$

Each sweep applies symplectic maps to disjoint blocks of canonical pairs and is therefore symplectic; as a composition of sweeps, the edge operator is symplectic as well.

The edge-exchange experiments use a path graph with two matchings: $\mathcal{E}_K^{(1)} = \{\{1,2\},\{3,4\},\dots\}$ and $\mathcal{E}_K^{(2)} = \{\{2,3\},\{4,5\},\dots\}$, with endpoints in $\{1,\dots,K\}$. We use one propagation round ($L = 1$), updating the first matching and then the second, with edges oriented from j to $j+1$. Under node relabeling, edge orientations and the matching order are carried along with the graph.

3.4 Canonical assembly and parameter reuse

The global operator encodes each physical patch, advances its latent state, applies edge exchange, and decodes the result back into the original physical coordinates. To express these operations in global canonical order, let S_K collect all latent position coordinates before all latent momenta:

$$Z = \mathsf{S}_K \operatorname{col}(z_1,\dots,z_K) = \operatorname{col}(a_1,\dots,a_K,b_1,\dots,b_K). \tag{17}$$

Here $z_j = (a_j, b_j)$ and $Z \in \mathbb{R}^{2Kr}$. The physical gather $\mathcal{R}_\mathcal{P}$ extracts complete patches, and its inverse $\mathcal{A}_\mathcal{P}$ returns decoded coordinates to their original positions.

Writing $f^{\oplus K}$ for patchwise application of the same map f, the assembled maps are

$$E_K = \mathsf{S}_K \circ E_{\mathrm{loc}}^{\oplus K} \circ \mathcal{R}_\mathcal{P}, \quad \Phi_K = \mathsf{S}_K \circ \Phi_{\mathrm{loc}}^{\oplus K} \circ \mathsf{S}_K^{-1}, \quad D_K = \mathcal{A}_\mathcal{P} \circ D_{\mathrm{loc}}^{\oplus K} \circ \mathsf{S}_K^{-1}. \tag{18}$$

The edge operator gathers and updates endpoint coordinates from the current global latent state using Eq. (14). Gather, packing, and scatter only reorder coordinates; all learned transformations occur in the local modules.

These coordinate changes express the product symplectic form in canonical order. Consequently, the local identities in Eq. (13) give $E_K \circ D_K = \mathrm{id}$ and a symplectic decoder onto $\mathcal{M}_K = D_K(\mathbb{R}^{2Kr})$. Together with the symplectic latent update, they yield the assembly guarantees in Section 4.

With the local architecture and coordinate conventions fixed, the same learned modules can be applied to a different number of patches. Deployment at a new system size retains all learned parameters and updates only the patch partition, interaction graph, and edge-update schedule.

3.5 Training objective

We train on segments $\{(x_i^0,\dots,x_i^T)\}_{i=1}^M$ sampled from the training trajectories at time interval h. Let $\ell_N(\widehat{x}, x) = \|\widehat{x} - x\|_2^2/(2Ns_{\Delta x})$, where $s_{\Delta x}$ is the mean squared state increment per coordinate, computed from the training pairs and held fixed (Appendix C). The prediction objective is

$$\mathcal{L}_{\mathrm{dyn}} = \frac{1}{M}\sum_{i=1}^{M}\left[\ell_N\big(F_{\theta,K}(x_i^0), x_i^1\big) + \frac{\lambda_{\mathrm{roll}}}{T}\sum_{t=1}^{T}\ell_N\big(F_{\theta,K}^t(x_i^0), x_i^t\big)\right], \tag{19}$$

where T is the training rollout horizon and $F_{\theta,K}^t$ denotes t successive model applications. Training first uses the one-step term ($\lambda_{\mathrm{roll}} = 0$), then adds autoregressive refinement with $T = 8$ and $\lambda_{\mathrm{roll}} = 1$. The encoder–decoder and local HénonNet parameters remain trainable in both phases, together with the edge parameters when enabled.

In the reported full-rank setting $P = r = 8$, the encoder and decoder are exact inverses on their respective spaces. Their parameters are learned jointly with those of the latent propagator and optional edge maps through the prediction objective. Appendix C details the representation-consistency stage, optimization, and checkpoint selection.

4 Structural Properties

Let $\Psi_K = C_{\theta_C,\mathcal{G}_K,\sigma_K} \circ \Phi_K$ denote the assembled latent update. The following properties connect local symplectic construction with parameter reuse across system sizes. Complete proofs and approximation assumptions are given in Appendix A.

Theorem 1 (Represented symplecticity). *Suppose D_K is a symplectic embedding with image $\mathcal{M}_K$, $E_K D_K = \mathrm{id}$, and Ψ_K is a symplectic diffeomorphism. Then $F_{\theta,K} = D_K \circ \Psi_K \circ E_K$ restricts to a symplectic diffeomorphism of $\mathcal{M}_K$.*

SHNO satisfies these conditions through its tied representation and composition of symplectic local and edge maps.

Scaling and graph dependence. For fixed local architectures and propagation depth, the parameter count is independent of K, and one forward step requires $O(K + |\mathcal{E}_K|)$ work. On uniformly bounded-degree graphs, this becomes $O(K)$. After L rounds of c ordered matching sweeps, each output patch depends only on inputs within cL graph hops.

Approximation across sizes. We establish a normalized approximation bound for target latent maps with the same local composition as SHNO. The bound is independent of patch count when block approximation errors, composition depth, and learned-block Lipschitz constants are uniformly bounded across sizes. Appendix A.4 gives the precise assumptions and proof.

5 Experiments

We evaluate SHNO on KG, FPUT, and Toda lattices, examining prediction accuracy, symplectic structure, parameter sharing, cross-size checkpoint reuse, and edge exchange.

5.1 Experimental setup

Systems and data. All three systems have nearest-neighbor interactions and free-end boundaries. We consider $N \in \{64, 128, 256\}$ canonical pairs and generate reference trajectories with Störmer–Verlet (Hairer et al., 2006) at step size $h = 0.01$. Each size has 32 training trajectories of 64 steps and eight validation and eight test trajectories of 128 steps. KG and FPUT use smooth initial conditions with nominal amplitude 0.04; Toda uses alternating initial conditions with amplitude 0.4. Appendix B specifies the Hamiltonians and initial-state distributions.

Models. SHNO uses patch width and latent rank $P = r = 8$, giving $K \in \{8, 16, 32\}$. SHNO-identity uses independent patch dynamics ($C = \mathrm{id}$; 5,574 parameters). SHNO-edge adds a shared six-reflector block with one two-color propagation round (5,772 parameters). KG comparisons use SHNO-identity; FPUT and Toda evaluate both configurations. We compare with Global Hénon ROM, which uses a full-state symplectic parameterization; Shared Residual, a convolutional operator with 5,483 parameters; and FNO, with 12,754 parameters. A direct sharing ablation uses independent copies of the SHNO local module at each patch while retaining the tied encoder–decoder construction within each copy. Appendix D gives the architecture details.

Training and evaluation. The main comparison trains models independently at each size using three random seeds. Training fits one-step predictions and then refines autoregressive rollouts with an eight-step horizon. Checkpoints are selected using validation trajectories. The fixed-checkpoint experiment instead deploys models trained at $N = 64$ directly at the two larger sizes. Toda compares identity and edge configurations under a separate paired protocol.

For each seed, we evaluate 128-step autoregressive predictions using pooled normalized root-mean-square error (NRMSE):

$$e_{\mathrm{roll}} = \frac{\|\widehat{X}_{1:128} - X_{1:128}\|_F}{\|X_{1:128}\|_F}, \tag{20}$$

where X contains the eight test trajectories. Tables report means and sample standard deviations across three independently trained seeds. Paired trajectory-error ratios are aggregated on the log scale, with 95% equal-tailed percentile confidence intervals from 20,000 hierarchical bootstrap resamples. Appendix C specifies optimization; Appendix C.3 describes the statistical procedure.

5.2 Prediction accuracy and symplectic structure

Table 1 compares rollout accuracy, parameter count, and numerical symplectic structure. For SHNO and Global Hénon ROM, the residual of $D_K \circ \Psi_K : \mathbb{R}^{2Kr} \to \mathbb{R}^{2N}$ with Jacobian A is $\delta_{\mathrm{repr}} = \frac{\|A^\top J_N A - J_{Kr}\|_F}{\|J_{Kr}\|_F}$. For Shared Residual and FNO, the diagnostic instead evaluates

Table 1: 128-step rollout NRMSE, parameter counts, and normalized symplectic residuals. Errors are means ± sample SD across three seeds at each separately trained size; residuals are means ± sample SD across the nine size–seed configurations. Residuals evaluate the represented map for geometric models and the physical prediction map for Shared Residual and FNO, as defined in Section 5. † denotes symplecticity by construction. Bold marks the lowest mean error among listed models for each system and size.

Model	Parameters	$N = 64$	$N = 128$	$N = 256$	Normalized residual
Klein–Gordon					
SHNO-identity†	5,574 fixed	$\mathbf{0.01286 \pm 0.00005}$	$\mathbf{0.00514 \pm 0.00025}$	$\mathbf{0.00187 \pm 0.00007}$	$(\mathbf{2.22 \pm 0.233}) \times 10^{-16}$
Global Hénon ROM†	23,622–62,022	0.02104 ± 0.00198	0.02659 ± 0.00158	0.03013 ± 0.00296	$(4.97 \pm 1.10) \times 10^{-16}$
Shared Residual	5,483 fixed	0.06328 ± 0.00695	0.05516 ± 0.00608	0.04996 ± 0.00513	$(1.66 \pm 0.505) \times 10^{-3}$
FNO	12,754 fixed	0.04057 ± 0.00522	0.03914 ± 0.00506	0.03887 ± 0.00482	$(6.67 \pm 1.87) \times 10^{-4}$
FPUT					
SHNO-identity†	5,574 fixed	0.01563 ± 0.00085	0.00674 ± 0.00043	0.00395 ± 0.00062	$(\mathbf{3.18 \pm 0.186}) \times 10^{-16}$
SHNO-edge†	5,772 fixed	$\mathbf{0.00826 \pm 0.00028}$	$\mathbf{0.00397 \pm 0.00042}$	$\mathbf{0.00187 \pm 0.00074}$	$(3.56 \pm 0.174) \times 10^{-16}$
Global Hénon ROM†	23,622–62,022	0.01900 ± 0.00100	0.01050 ± 0.00050	0.00680 ± 0.00030	$(5.93 \pm 1.16) \times 10^{-16}$
Shared Residual	5,483 fixed	0.02116 ± 0.00037	0.01183 ± 0.00045	0.01149 ± 0.00051	$(7.36 \pm 1.89) \times 10^{-4}$
FNO	12,754 fixed	0.02411 ± 0.00056	0.01454 ± 0.00112	0.01296 ± 0.00114	$(3.34 \pm 0.809) \times 10^{-4}$

$\|[\mathrm{D}F(x)]^\top J_N \mathrm{D}F(x) - J_N\|_F / \|J_N\|_F$ for the physical prediction map F. Appendix E provides additional residual statistics.

KG. SHNO-identity has lower mean rollout error than all three listed baselines at every size. Its error decreases from 0.01286 at $N = 64$ to 0.00187 at $N = 256$, an 85.5% reduction with fixed parameter count. Over the same sizes, Shared Residual and FNO errors decrease by 21.0% and 4.2%, respectively, while Global Hénon ROM error increases from 0.02104 to 0.03013 despite its larger parameter budget.

FPUT. SHNO-edge achieves the lowest mean error at every size, followed by SHNO-identity. At $N = 128$, their errors are 0.00397 and 0.00674, compared with 0.01050 for Global Hénon ROM, 0.01183 for Shared Residual, and 0.01454 for FNO. Edge exchange reduces mean error relative to identity coupling by 47.2%, 41.1%, and 52.7% at $N = 64, 128, 256$, respectively, with 198 additional parameters. From $N = 64$ to 256, errors decrease by 77.4% for SHNO-edge and 74.7% for SHNO-identity, versus 64.2% for Global Hénon ROM and approximately 46% for both non-symplectic baselines. On both systems, SHNO therefore exhibits a larger relative error reduction across the tested sizes without increasing its parameter count.

Numerical structure. Mean normalized represented-map residuals range from 2.22×10^{-16} to 5.93×10^{-16} for the geometric models, consistent with float64 roundoff. The mean physical-map residuals of Shared Residual and FNO range from 3.34×10^{-4} to 1.66×10^{-3}. These diagnostics complement the prediction results: SHNO achieves the reported accuracy while preserving its prescribed symplectic structure.

5.3 Direct effect of parameter sharing

We compare SHNO-identity with an unshared symplectic control that assigns independent representation and latent modules to each patch, keeping local architectures and training settings unchanged. Figure 2 shows the accuracy and parameter trade-off.

The unshared control uses 8–32 times as many parameters. On KG, it achieves lower mean errors, but the accuracy gap narrows as system size increases. On FPUT, the ordering varies with size, with no consistent advantage from independent patch parameters. Appendix F provides the size-resolved comparison and paired statistical analysis.

5.4 Fixed-checkpoint prediction across sizes

For each system and seed, we deploy the selected SHNO-identity $N = 64$ checkpoint at $N = 128, 256$ without parameter updates or normalization refitting. Target-size training gives lower mean errors in all four groups (Table 2). Figure 3 shows similar fixed-checkpoint errors at both KG target sizes, with a larger benefit from target-size training at $N = 256$. On FPUT, the fixed-checkpoint and

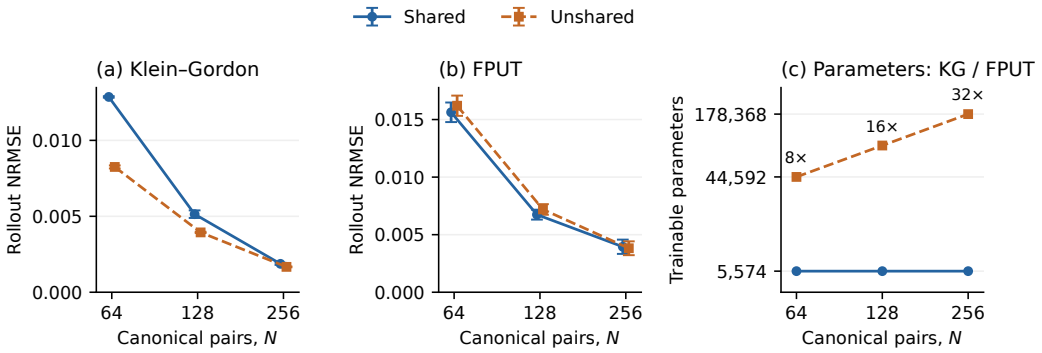

Figure 2: Effect of parameter sharing. (a,b) KG and FPUT rollout errors (mean ± sample standard deviation across three training seeds). (c) Parameter counts on a logarithmic scale; KG and FPUT use identical architectures and therefore share these curves. Annotations give the unshared/shared parameter ratios. Small horizontal offsets in (a,b) separate error bars.

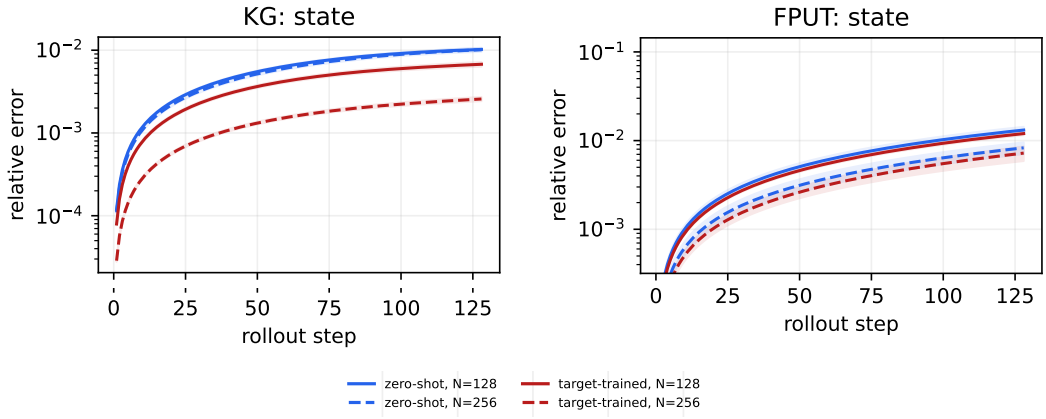

Figure 3: State error during fixed-checkpoint transfer on KG (left) and FPUT (right). Blue curves use the unchanged $N = 64$ checkpoint (zero-shot); red curves use target-size training. Solid and dashed lines denote $N = 128$ and $N = 256$. Errors are averaged over eight test trajectories within each seed, then over three seeds; shading is one sample standard deviation across seed-level curves. The vertical axes are logarithmic.

target-trained error curves are closer to each other. Appendix G provides Hamiltonian discrepancies and metric definitions.

5.5 Effect of Symplectic Edge Exchange

On Toda, we compare SHNO-identity and SHNO-edge with paired training seeds and test trajectories, adding only the shared 198-parameter edge block. Figure 4 shows geometric edge/identity error ratios below one at every size. Pooling paired log ratios across sizes gives 0.372 (95% interval: $[0.317, 0.445]$), a 62.8% reduction in geometric mean error with 3.6% more parameters. Together with the FPUT results, these gains support explicit symplectic communication between patches. Appendix H provides the hierarchical analysis and interaction diagnostics.

Additional results. The appendices provide size-resolved sharing comparisons (Appendix F), time-resolved state and Hamiltonian errors for fixed-checkpoint deployment (Appendix G), and Toda interaction and edge-response diagnostics (Appendix H). Appendix E reports full residual statistics, numerical assembly tests, and CPU forward-time scaling with the corresponding measurement protocol.

Target	Fixed $N = 64$	Target-trained
KG, 128	$.00704 \pm .00022$	$.00514 \pm .00025$
KG, 256	$.00679 \pm .00032$	$.00187 \pm .00007$
FPUT, 128	$.00741 \pm .00078$	$.00674 \pm .00043$
FPUT, 256	$.00462 \pm .00084$	$.00395 \pm .00062$

Table 2: Fixed-checkpoint transfer: 128-step NRMSE (mean $\pm$ sample SD, three seeds) for the unchanged $N = 64$ checkpoint and target-trained models.

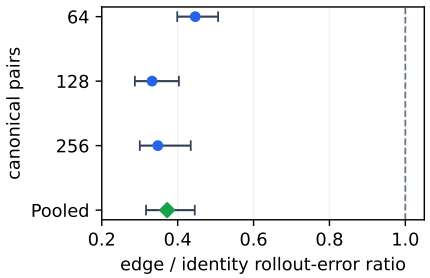

Figure 4: Toda edge/identity geometric error ratios with 95% bootstrap intervals (20,000 paired hierarchical resamples). The diamond pools sizes on the log scale; one denotes equal error.

6 Related Work

Learning Hamiltonian dynamics. Hamiltonian and Lagrangian neural networks learn energies or structured vector fields (Greydanus et al., 2019; Cranmer et al., 2019; Toth et al., 2020; Finzi et al., 2020). Related methods combine these formulations with numerical integration, control, and nonseparable dynamics (Chen et al., 2020; Zhong et al., 2020; Xiong et al., 2021). SympNets, HénonNets, and generating-function networks instead parameterize discrete symplectic maps directly (Jin et al., 2020; Burby et al., 2021; Chen & Tao, 2021). SHNO builds on this map-based approach through shared patch dynamics and symplectic edge updates.

Symplectic model reduction. Proper symplectic decomposition and nonlinear symplectic autoencoders construct reduced representations that preserve Hamiltonian geometry (Peng & Mohseni, 2016; Buchfink et al., 2023; Chen et al., 2025). SHNO applies the encoder–dynamics–decoder composition locally, with a tied encoder–decoder and shared latent map. Sharing these modules across patches makes the parameter count independent of patch count.

Operator learning and local composition. DeepONet and graph, Fourier, multipole, and wavelet operators share learned mappings across discretizations (Lu et al., 2021; Li et al., 2020a;b; 2021; Gupta et al., 2021), with extensions to varying geometries and nonuniform sampling (Li et al., 2023; Liu et al., 2023). Graph simulators reuse local interaction rules across systems with different numbers of nodes (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2020). Hamiltonian graph networks, port-Hamiltonian compositions, energy-consistent operators, and Symplectic Neural Operators incorporate physical structure into these settings (Sanchez-Gonzalez et al., 2019; Neary & Topcu, 2023; Tanaka et al., 2025; Makara et al., 2026). SHNO shares parameters across local and edge maps, whose ordered composition preserves the symplectic form on the represented manifold.

7 Conclusion

We introduced SHNO, which assembles shared local symplectic maps over canonical patches. Its tied representations, shared Hénon dynamics, and optional edge exchange preserve the symplectic form on the represented manifold with a parameter count independent of patch count. For fixed local architectures and composition depth, SHNO has $O(K+|\mathcal{E}_K|)$ computational cost and bounded graph dependence; under uniform approximation and stability assumptions, compatible local target compositions admit a size-uniform normalized error bound. Experiments on KG, FPUT, and Toda quantify the accuracy trade-offs of sharing and fixed-checkpoint deployment. Matched unshared models require 8–32 times as many parameters, while target-size training improves accuracy over checkpoint reuse. On Toda, edge exchange reduces geometric mean rollout error by 62.8% with 3.6% more parameters. These results support shared local symplectic composition for parameter-efficient learning across system sizes. Future work will examine reduced-rank representations ($r < P$) for cross-size prediction and the role of propagation depth in capturing longer-range interactions on more general graphs.

AI USE STATEMENT

We used generative AI tools solely to improve the grammar, clarity, and readability of the manuscript. The authors reviewed and approved all AI-assisted edits and take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

Section 5 describes the experimental setup and evaluation. Appendix A provides assumptions and proofs; Appendices B–D specify the systems, data generation, training, statistical procedures, and model architectures. Appendices E–H report additional results and diagnostics, and Appendix I documents the implementation environment.

REFERENCES

Patrick Buchfink, Silke Glas, and Bernard Haasdonk. Symplectic model reduction of hamiltonian systems on nonlinear manifolds and approximation with weakly symplectic autoencoder. *SIAM journal on scientific computing*, 45(2):A289–A311, 2023.

Joshua William Burby, Qi Tang, and Romit Maulik. Fast neural poincaré maps for toroidal magnetic fields. *Plasma Physics and Controlled Fusion*, 63(2):024001, 2021.

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

Yongsheng Chen, Wei Guo, Qi Tang, and Xinghui Zhong. Reduced-order modeling of hamiltonian dynamics based on symplectic neural networks. *arXiv preprint arXiv:2508.11911*, 2025.

Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. Symplectic recurrent neural networks. In *International Conference on Learning Representations*, 2020.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2019.

Marc Finzi, Ke Alexander Wang, and Andrew G Wilson. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. *Advances in neural information processing systems*, 33: 13880–13889, 2020.

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

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021.

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

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

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in neural information processing systems*, 33:6755–6766, 2020b.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.

Songming Liu, Zhongkai Hao, Chengyang Ying, Hang Su, Ze Cheng, and Jun Zhu. Nuno: A general framework for learning parametric pdes with non-uniform data. In *International Conference on Machine Learning*, pp. 21658–21671, 2023.

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

D Steven Mackey and Niloufer Mackey. *On the determinant of symplectic matrices*. Manchester Centre for Computational Mathematics Manchester, England, 2003.

Yeang Makara, Yusuke Tanaka, Takashi Matsubara, and Takaharu Yaguchi. Symplectic neural operators for learning infinite dimensional hamiltonian systems. *arXiv preprint arXiv:2605.15881*, 2026.

Cyrus Neary and Ufuk Topcu. Compositional learning of dynamical system models using port-hamiltonian neural networks. In *Learning for Dynamics and Control Conference*, pp. 679–691, 2023.

Liqian Peng and Kamran Mohseni. Symplectic model reduction of hamiltonian systems. *SIAM Journal on Scientific Computing*, 38(1):A1–A27, 2016.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ode integrators. *arXiv preprint arXiv:1909.12790*, 2019.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468, 2020.

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

Peter Toth, Danilo J. Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *International Conference on Learning Representations*, 2020.

Shiying Xiong, Yunjin Tong, Xingzhe He, Shuqi Yang, Cheng Yang, and Bo Zhu. Nonseparable symplectic neural networks. In *International Conference on Learning Representations*, 2021.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020.

A Theoretical Analysis

We consider disjoint patches containing complete canonical position–momentum pairs. The physical state has KP canonical pairs, and the latent state has Kr canonical pairs. Write

$$\Psi_K = C_{\theta_C, \mathcal{G}_K, \sigma_K} \circ \Phi_K, \qquad F_{\theta,K} = D_K \circ \Psi_K \circ E_K.$$

All symplectic forms below are canonical or induced by restriction to the represented manifold.

A.1 Symplecticity of the assembled operator

Let $D_K : \mathbb{R}^{2Kr} \to \mathbb{R}^{2KP}$ be a symplectic embedding with image $\mathcal{M}_K$, and suppose $E_K D_K = \mathrm{id}$. For a symplectic diffeomorphism Ψ_K, we show that $F_{\theta,K}|_{\mathcal{M}_K}$ is a symplectic diffeomorphism.

Proof. For every latent state Z,

$$F_{\theta,K}(D_K(Z)) = D_K(\Psi_K(Z)). \tag{21}$$

Thus the represented dynamics are conjugate to Ψ_K through the diffeomorphism $D_K : \mathbb{R}^{2Kr} \to \mathcal{M}_K$.

Let $\omega_{\mathcal{M}}$ be the induced form on $\mathcal{M}_K$ and ω_{lat} the canonical form on $\mathbb{R}^{2Kr}$. The symplectic embedding identity is $D_K^* \omega_{\mathcal{M}} = \omega_{\mathrm{lat}}$. Consequently,

$$\begin{aligned} D_K^* (F_{\theta,K}|_{\mathcal{M}_K})^* \omega_{\mathcal{M}} &= \Psi_K^* D_K^* \omega_{\mathcal{M}} \\ &= \Psi_K^* \omega_{\mathrm{lat}} \\ &= \omega_{\mathrm{lat}} = D_K^* \omega_{\mathcal{M}}. \end{aligned}$$

Pullback by D_K is invertible on forms on $\mathcal{M}_K$, so $F_{\theta,K}|_{\mathcal{M}_K}$ preserves $\omega_{\mathcal{M}}$. Its inverse is $D_K \circ \Psi_K^{-1} \circ E_K$ restricted to $\mathcal{M}_K$. □

For SHNO, the local representation $E_{\mathrm{loc}} = \pi_r \circ T_{\theta_E}$ and $D_{\mathrm{loc}} = T_{\theta_E}^{-1} \circ \iota_r$ satisfies

$$E_{\mathrm{loc}} D_{\mathrm{loc}} = \mathrm{id}, \qquad D_{\mathrm{loc}}^* \omega_P = \omega_r.$$

Direct sums of these maps, expressed through canonical packing and scatter, give $E_K D_K = \mathrm{id}$ and a symplectic global decoder. Local Hénon maps and disjoint edge updates are symplectic diffeomorphisms, as is their composition.

When $r = P$, the encoder and decoder are inverse transformations on the full physical state space. Hence

$$(\mathrm{D}F_{\theta,K}(x))^\top J_{KP}\, \mathrm{D}F_{\theta,K}(x) = J_{KP}. \tag{22}$$

A.2 Parameter count and computational complexity

Theorem 2 (Fixed parameters and sparse computation)**.** *Fix P, r, the local network architectures, edge-block depth d, and propagation count L. The trainable parameter count is independent of K. Given the partition, graph, and edge schedule, a forward evaluation requires $O(K + |\mathcal{E}_K|)$ fixed-width work. An implementation storing local node and edge activations requires $O(K + |\mathcal{E}_K|)$ activation storage. Both bounds reduce to $O(K)$ on uniformly bounded-degree graph families.*

Proof. The parameter vector contains one shared local representation, one shared latent map, and one shared edge block. Their tensor dimensions depend only on the fixed local architectures.

Encoding, local evolution, and decoding each apply a fixed-width map to K patches. The edge operator performs $L|\mathcal{E}_K|$ fixed-width block evaluations. Canonical packing and scatter take $O(K)$ work. Thus the total work is $O(K + |\mathcal{E}_K|)$.

Storing the node activations takes $O(K)$ space. For fixed block depth and propagation count, the local edge activations take $O(|\mathcal{E}_K|)$ space. A uniform degree bound gives $|\mathcal{E}_K| = O(K)$. □

A target assembly reuses the same parameter vector with a new partition, graph, and schedule, provided the local dimensions and coordinate conventions agree. The structural guarantees apply to the target assembly; prediction accuracy is evaluated separately by the fixed-checkpoint experiments.

A.3 Graph dependence and relabeling

Proposition 3 (Graph receptive field)**.** *Suppose each propagation round contains c ordered matching sweeps and all local maps act independently on individual patches. After L rounds, each output patch depends only on input patches within graph distance cL. For the two-color path schedule, this radius is at most $2L$.*

Proof. In a matching sweep, each node participates in at most one edge update. An output at that node can therefore acquire dependence from at most one neighboring node's current state. Each sweep increases the dependency radius by at most one hop. Composing cL sweeps gives the bound. Patch-local encoding, evolution, and decoding do not increase graph dependence. □

Relabeling transports the partition, edge orientations, and ordered matchings together. Each gather, local update, and scatter operation is then the relabeled version of the original operation. The assembled computation and its dependency bound therefore transform consistently under this joint relabeling.

A.4 Size-uniform compositional approximation

Define the normalized latent norm

$$\|Z\|_K = K^{-1/2}\|Z\|_2. \tag{23}$$

This norm measures root-mean-square error across patches.

Theorem 4 (Size-uniform compositional approximation)**.** *Let*

$$\Psi_K = T_{K,S} \circ \cdots \circ T_{K,1}, \qquad \Psi_K^\star = T_{K,S}^\star \circ \cdots \circ T_{K,1}^\star$$

be learned and target maps with the same sequence of S node or matching stages. Corresponding matching stages use the same edges and endpoint ordering and leave unmatched nodes unchanged.

On compact sets containing all relevant learned and target intermediate states, suppose:

1. *each learned block is B-Lipschitz in its Euclidean block norm;*
2. *corresponding learned and target node blocks differ uniformly by at most ϵ_{loc};*
3. *corresponding learned and target edge blocks differ uniformly by at most ϵ_{edge}.*

Assume these bounds and S are uniform in K. With $\overline{B} = \max\{1, B\}$, every admissible input satisfies

$$\|\Psi_K(Z) - \Psi_K^\star(Z)\|_K \le \left(\sum_{s=0}^{S-1} \overline{B}^{\,s}\right)(\epsilon_{\mathrm{loc}} + \epsilon_{\mathrm{edge}}). \tag{24}$$

Proof. A node stage acts on K disjoint blocks. Its squared Euclidean approximation error is at most $K\epsilon_{\mathrm{loc}}^2$, so its error in $\|\cdot\|_K$ is at most ϵ_{loc}.

A matching contains at most $K/2$ edges with disjoint output blocks. Its squared Euclidean approximation error is at most $(K/2)\epsilon_{\mathrm{edge}}^2$. Thus its normalized error is at most $\epsilon_{\mathrm{edge}}/\sqrt{2}$. Unmatched coordinates contribute no approximation error.

Each learned stage is $\overline{B}$-Lipschitz in $\|\cdot\|_K$: the nontrivial blocks are disjoint, and unmatched coordinates undergo the identity. Let Z_s and $Z_s^\star$ be the learned and target intermediate states, with $Z_0 = Z_0^\star = Z$. Writing $e_s = \|Z_s - Z_s^\star\|_K$, we obtain

$$\begin{aligned} e_s &\le \|T_{K,s}(Z_{s-1}) - T_{K,s}(Z_{s-1}^\star)\|_K \\ &\quad + \|T_{K,s}(Z_{s-1}^\star) - T_{K,s}^\star(Z_{s-1}^\star)\|_K \\ &\le \overline{B} e_{s-1} + \epsilon_{\mathrm{loc}} + \epsilon_{\mathrm{edge}}. \end{aligned}$$

Starting from $e_0 = 0$ and iterating proves the bound. □

For SHNO, one node stage followed by L rounds of c matching sweeps gives $S = 1 + cL$. A uniformly bounded-degree graph family admits schedules with a uniformly bounded number of colors. With such schedules and fixed L, the composition depth remains independent of K.

The theorem bounds approximation of a compatible latent target composition. Approximating a physical Hamiltonian flow also involves the represented manifold and the accuracy of the chosen local target composition.

A.5 Autoregressive error propagation

Suppose Ψ_K is Λ-Lipschitz in $\|\cdot\|_K$ on an admissible set and

$$\|\Psi_K(Z) - \Psi_K^\star(Z)\|_K \le \epsilon$$

throughout that set. For learned and target trajectories starting from the same state and remaining in the set,

$$\|\Psi_K^t(Z) - (\Psi_K^\star)^t(Z)\|_K \le \epsilon \sum_{j=0}^{t-1} \Lambda^j. \tag{25}$$

Indeed, adding and subtracting the learned map evaluated at the target state gives $e_{t+1} \le \Lambda e_t + \epsilon$ with $e_0 = 0$. The bound follows by iteration. Uniform one-step approximation therefore yields a patch-count-independent rollout bound when the stability constant is uniform and the trajectories remain admissible.

B Benchmark Systems and Data

B.1 Hamiltonian systems

We consider canonical states $x = (q_1, \dots, q_N, p_1, \dots, p_N) \in \mathbb{R}^{2N}$ with free-end nearest-neighbor interactions. Writing $d_i = q_{i+1} - q_i$, the Hamiltonians are

$$H_{\mathrm{KG}}(q,p) = \frac{1}{2}\sum_{i=1}^{N} p_i^2 + \frac{1}{2}\sum_{i=1}^{N-1} d_i^2 + \frac{\omega^2}{2}\sum_{i=1}^{N} q_i^2 + \frac{\gamma}{4}\sum_{i=1}^{N} q_i^4, \tag{26}$$

$$H_{\mathrm{FPUT}}(q,p) = \frac{1}{2}\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N-1}\left(\frac{1}{2}d_i^2 + \frac{\alpha}{3}d_i^3 + \frac{\beta}{4}d_i^4\right), \tag{27}$$

$$H_{\mathrm{Toda}}(q,p) = \frac{1}{2}\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N-1}\left(e^{d_i} - d_i - 1\right). \tag{28}$$

We use $(\omega, \gamma) = (1, 0.1)$ for Klein–Gordon and $(\alpha, \beta) = (0.25, 0.1)$ for FPUT. Reference trajectories are generated with Störmer–Verlet at step size $\Delta t = 0.01$.

B.2 Initial conditions and data splits

For Klein–Gordon and FPUT, the initial-condition amplitude is $A = 0.04$. For each trajectory identifier s, a deterministic generator selects $m_q, m_p \in \{1,2,3,4\}$, $\phi_q, \phi_p \in [0, 2\pi)$, and $a_s \in [0.85A, 1.15A]$. The initial state is

$$q_i^0 = a_s \sin\left(\frac{m_q \pi i}{N+1} + \phi_q\right), \qquad p_i^0 = \frac{a_s}{2}\cos\left(\frac{m_p \pi i}{N+1} + \phi_p\right), \quad i = 1, \dots, N. \tag{29}$$

For Toda, we use $A = 0.4$ and alternating initial conditions:

$$q_i^0 = A(1 + 0.03\sin s)(-1)^{i-1}, \qquad p_i^0 = 0.25A(-1)^{i-1+s}, \quad i = 1, \dots, N. \tag{30}$$

Each system and size has 32 training trajectories, eight validation trajectories, and eight test trajectories. Training trajectories contain 64 integration steps; validation and test trajectories contain 128. The splits use disjoint trajectory identifiers. All model-training seeds use the same data. Autoregressive training windows start from states in the training trajectories.

Table 3: Training configuration.

Setting	Value
Training / validation / test trajectories	32 / 8 / 8
Training / evaluation steps	64 / 128
Patch width and latent rank	$P = r = 8$
Training seeds	0, 1, 2
Representation-consistency stage	100 epochs
One-step training	240 epochs
Autoregressive refinement	160 epochs, horizon 8
Optimizer	Adam
Learning rates	10^{-3}, then 10^{-4}
Gradient clipping	Global norm 1
Checkpoint evaluation	Every 10 epochs
Rollout-loss weight	1
Auxiliary physics losses	None
Geometric-model precision	Float64

C Training and Evaluation

C.1 Optimization and checkpoint selection

Models are trained at $N \in \{64, 128, 256\}$ with training seeds $\{0, 1, 2\}$. SHNO uses patch width $P = 8$ and latent rank $r = 8$, giving $K = N/8$ patches. The independently trained comparisons fit a separate model at each size. The fixed-checkpoint experiment is described in Appendix G.

The implementation begins with 100 representation-consistency epochs using reconstruction and latent-cycle MSE. This stage involves only the encoder and decoder; at $P = r = 8$, both losses vanish in exact arithmetic and provide no effective learning signal. Prediction training then uses 240 one-step epochs and 160 autoregressive epochs with horizon eight, jointly optimizing the encoder–decoder and dynamics parameters.

Prediction losses use the fixed training increment scale

$$s_{\Delta x} = \frac{1}{2N|\mathcal{D}_{\mathrm{tr}}|} \sum_{(x,x^+)\in\mathcal{D}_{\mathrm{tr}}} \|x^+ - x\|_2^2. \tag{31}$$

For a predicted state $\widehat{x}$ and reference state x, the normalized squared error is

$$\ell(\widehat{x}, x) = \frac{\|\widehat{x} - x\|_2^2}{2N s_{\Delta x}}. \tag{32}$$

The scale is computed once from the training pairs and held fixed throughout optimization.

At $P = r$, the tied SHNO representation satisfies $D_K E_K = E_K D_K = \mathrm{id}$. Reconstruction and cycle residuals therefore serve as numerical consistency diagnostics. The prediction losses train the dynamics and the representation through the assembled operator.

Checkpoints are evaluated every ten epochs. Among those with mean validation one-step increment NRMSE at most 0.95, we select the checkpoint with the lowest validation rollout error. If none satisfies this threshold, we select the checkpoint with the lowest validation increment NRMSE.

C.2 Rollout error

For each trained model, prediction starts from the reference initial state and proceeds autoregressively for $T = 128$ steps. With $B = 8$ test trajectories, the pooled rollout error is

$$e_{\mathrm{roll}} = \left(\frac{\sum_{b=1}^{B} \sum_{t=1}^{T} \|\widehat{x}_t^{(b)} - x_t^{(b)}\|_2^2}{\sum_{b=1}^{B} \sum_{t=1}^{T} \|x_t^{(b)}\|_2^2} \right)^{1/2}. \tag{33}$$

Reported means and sample standard deviations summarize this scalar across the three independently trained seeds. The sample variance uses denominator $S - 1 = 2$ ($S = 3$, $\mathrm{ddof} = 1$); trajectories are aggregated within each seed.

C.3 PAIRED ERROR RATIOS

The sharing, fixed-checkpoint, and Toda comparisons also use trajectory-level rollout errors,

$$e^{(b)} = \left(\frac{\sum_{t=1}^{T} \|\widehat{x}_t^{(b)} - x_t^{(b)}\|_2^2}{\sum_{t=1}^{T} \|x_t^{(b)}\|_2^2} \right)^{1/2}. \tag{34}$$

For two models A and B, errors are paired by system, size, training seed, and test trajectory. The aggregate ratio is the geometric mean

$$R_{A/B} = \exp\left[\frac{1}{|\mathcal{I}|} \sum_{\iota \in \mathcal{I}} \log\left(\frac{e_A^{(\iota)}}{e_B^{(\iota)}} \right) \right], \tag{35}$$

where $\mathcal{I}$ indexes the paired evaluations. Values below one favor model A.

This statistic averages log ratios of trajectory-level errors, whereas Eq. (33) pools squared errors before taking a ratio. Consequently, a paired geometric ratio need not equal the ratio of the reported mean rollout errors. We use 20,000 hierarchical bootstrap resamples. At each level, groups are sampled with replacement within the selected parent group. Both model errors are selected together, preserving pairing by seed label and test-trajectory index within each system and size. Each replicate averages the sampled log ratios and exponentiates the result. The 95% equal-tailed percentile interval uses the empirical 2.5th and 97.5th percentiles of these bootstrap estimates. Across-seed sample SD describes training variation; bootstrap intervals describe uncertainty in the paired geometric ratio under the specified hierarchy. Experiment-specific hierarchies and random seeds are given below.

D MODEL ARCHITECTURES

D.1 SHARED LOCAL SYMPLECTIC MAPS

Each Hénon potential is a scalar-output tanh MLP with two hidden layers of width 32. The final layer is initialized to zero. The patch transformation and local latent dynamics each contain two Hénon layers, with four applications of the elementary map per layer.

The tied representation contains 2,804 parameters, and the local latent map contains 2,770. The identity-coupling model therefore has 5,574 trainable parameters. The six-reflector edge block adds 198 parameters, giving 5,772 for SHNO with edge exchange.

D.2 SHARED EDGE EXCHANGE

For $y, u \in \mathbb{R}^{2m}$ and scalar β, the elementary G-reflector is

$$\mathcal{R}_{u,\beta}(y) = y + \beta(u^\top J_m y)u. \tag{36}$$

The identity $u^\top J_m u = 0$ gives

$$\mathcal{R}_{u,\beta}^{-1} = \mathcal{R}_{u,-\beta}, \qquad (\mathrm{D}\mathcal{R}_{u,\beta})^\top J_m \mathrm{D}\mathcal{R}_{u,\beta} = J_m. \tag{37}$$

For an oriented edge (i, j) with patch latents $z_i = (a_i, b_i)$ and $z_j = (a_j, b_j)$, the edge block acts on

$$y = (a_i, a_j, b_i, b_j) \in \mathbb{R}^{4r}.$$

It composes six reflectors and shares their parameters across all graph edges. At $r = 8$, each reflector has $4r + 1 = 33$ parameters, giving $6(4r + 1) = 198$ parameters.

The edge-exchange experiments use a path graph with two ordered matchings:

$$\begin{aligned} \mathcal{E}_K^{(1)} &= \{\{1, 2\}, \{3, 4\}, \ldots\}, \\ \mathcal{E}_K^{(2)} &= \{\{2, 3\}, \{4, 5\}, \ldots\}, \end{aligned} \tag{38}$$

with endpoints in $\{1, \ldots, K\}$ and edges oriented from j to $j + 1$. One propagation round updates $\mathcal{E}_K^{(1)}$ and then $\mathcal{E}_K^{(2)}$. Edges within each matching have no common endpoints. Thus, the predictive

edge configuration has $c = 2$ colors and $L = 1$ propagation round; its dependency radius is at most two graph hops.

Graph relabeling transports the endpoint orientations and ordered edge sets together with the node labels.

D.3 Unshared symplectic control

The unshared control assigns an independent copy of the local representation and latent dynamics to each patch. The encoder and decoder remain tied within each patch. Local widths, depths, losses, and optimization settings match the shared model.

Its parameter count is $5{,}574K$, giving 44,592, 89,184, and 178,368 parameters at $N = 64, 128, 256$. The comparison isolates parameter sharing while preserving the local symplectic construction.

D.4 Baseline models

Global Hénon ROM. The global model uses tied symplectic encoding and decoding with a full-state latent map of potential width 64. It contains 23,622, 36,422, and 62,022 parameters at $N = 64, 128, 256$, respectively.

Shared Residual. This model is a shared one-dimensional convolutional operator with hidden width 29 and three blocks. It contains 5,483 trainable parameters.

FNO. FNO uses grid positional embedding, hidden width 16, four Fourier layers, and at most 16 modes. It predicts residual updates in normalized position and momentum channels. Normalization is fitted on the training data. The model contains 12,754 parameters. Fourier computations use float32, and predictions are converted to float64 for evaluation.

Shared Residual and FNO use the KG/FPUT data, normalized prediction objective, optimization schedule, checkpoint rule, and rollout metric described in Appendix C.

E Numerical Structure and Computational Scaling

E.1 Represented symplecticity

The main table summarizes residuals across the nine configurations formed by three system sizes and three training seeds for each system and model. Its sample standard deviation uses denominator $9 - 1 = 8$ and describes variation across both sizes and seeds. Rollout-error standard deviations instead summarize three seeds at a fixed size. The maxima below use the same normalized configuration-level measurements as the main table.

For $G_K = D_K \circ \Psi_K : \mathbb{R}^{2Kr} \to \mathbb{R}^{2N}$ with Jacobian A, the represented-map residual is

$$\delta_{\mathrm{repr}} = \frac{\|A^\top J_N A - J_{Kr}\|_F}{\|J_{Kr}\|_F}. \tag{39}$$

This diagnostic measures preservation of the canonical symplectic form by the latent-to-physical map G_K. For each size–seed configuration, we reload the validation-selected checkpoint and evaluate $A = \mathrm{D}G_K(Z_0)$ at $Z_0 = E_K(x_0)$, where x_0 is the initial state of the first test trajectory. Each configuration contributes one residual, without averaging over test trajectories or rollout steps. The geometric models use float64. The normalization factor is $\|J_{Kr}\|_F = \sqrt{2Kr}$, which equals $\sqrt{2N}$ in the reported full-rank setting. We normalize each configuration's residual before computing means, sample standard deviations, or maxima across configurations.

Table 4: Maximum normalized represented-map residuals across the nine size–seed configurations for each system and model. A dash indicates that the configuration was not evaluated with this diagnostic.

Model	KG	FPUT
SHNO-identity	2.543×10^{-16}	3.395×10^{-16}
SHNO-edge	–	3.795×10^{-16}
Global Hénon ROM	6.552×10^{-16}	7.466×10^{-16}

Table 5: Physical-map symplectic residuals for Shared Residual and FNO (mean $\pm$ sample standard deviation across three training seeds). All reported values are finite.

System	N	Shared Residual	FNO
KG	64	$(1.9722 \pm 0.71515) \times 10^{-3}$	$(9.0037 \pm 0.71188) \times 10^{-4}$
KG	128	$(1.5283 \pm 0.43298) \times 10^{-3}$	$(6.0378 \pm 0.21040) \times 10^{-4}$
KG	256	$(1.4682 \pm 0.30717) \times 10^{-3}$	$(4.9536 \pm 0.46240) \times 10^{-4}$
FPUT	64	$(8.9081 \pm 1.8440) \times 10^{-4}$	$(4.2887 \pm 0.20495) \times 10^{-4}$
FPUT	128	$(7.0704 \pm 1.6379) \times 10^{-4}$	$(3.2180 \pm 0.25819) \times 10^{-4}$
FPUT	256	$(6.1102 \pm 1.4861) \times 10^{-4}$	$(2.5112 \pm 0.32549) \times 10^{-4}$

E.2 Physical-map residuals for prediction baselines

For Shared Residual and FNO, we evaluate the complete physical prediction map $F : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ using

$$\delta_{\mathrm{phys}}(x) = \frac{\|[\mathrm{D}F(x)]^\top J_N \mathrm{D}F(x) - J_N\|_F}{\|J_N\|_F}. \tag{40}$$

Table 5 reports means and sample standard deviations across the three training seeds. These physical-map measurements are distinct from the latent-to-physical pullback residuals of the geometric models.

E.3 Assembly diagnostics

A separate 5,640-parameter configuration tests the assembly operations. The square edge-map residual is 3.8×10^{-16}. The decoder pullback residual, transported-schedule relabeling error, and finite-difference support leakage are zero at the reported precision.

The finite-difference probe detects no dependence beyond two graph hops after one two-color path round, consistent with the dependency bound. The same parameter state loads at $N = 64, 128, 256$. This structural configuration uses different widths from the 5,574-parameter and 5,772-parameter predictive models.

E.4 Timing protocol

The CPU assembly benchmark uses the 5,640-parameter configuration in float64 with batch size eight. Each graph size receives five untimed warm-up evaluations and five timed trials. Each trial averages 20 forward evaluations; the reported value is the median of the five trial averages.

For a path graph, define $w = K + |\mathcal{E}_K| = 2K - 1$. The measured times are

w	31	127	511	2047
Time (ms)	0.445	0.524	1.153	3.839

A linear fit gives

$$t(w) = 0.3307 + 0.001707w \quad \text{milliseconds}, \qquad R^2 = 0.9991.$$

These measurements characterize forward assembly cost for the benchmark configuration.

Table 6: Direct sharing ablation. Rollout errors are means ± sample standard deviations over three training seeds. The final column gives paired geometric error ratios with 95% equal-tailed percentile bootstrap intervals. Values below one favor shared SHNO.

System	N	Shared error	Unshared error	Shared/unshared ratio
KG	64	0.01286 ± 0.00005	0.00825 ± 0.00010	$1.401\ [1.239, 1.579]$
KG	128	0.00514 ± 0.00025	0.00394 ± 0.00004	$1.135\ [0.990, 1.301]$
KG	256	0.00187 ± 0.00007	0.00167 ± 0.00003	$0.958\ [0.854, 1.077]$
FPUT	64	0.01563 ± 0.00085	0.01620 ± 0.00088	$0.965\ [0.915, 1.044]$
FPUT	128	0.00674 ± 0.00043	0.00719 ± 0.00046	$0.937\ [0.898, 0.963]$
FPUT	256	0.00395 ± 0.00062	0.00382 ± 0.00060	$1.034\ [0.908, 1.204]$

F Direct Sharing Ablation

The shared and unshared models are trained on KG and FPUT at all three sizes and seeds using the protocol in Appendix C. Table 6 reports the complete size-resolved comparison.

The shared model has 5,574 parameters at every size. The unshared model uses 8, 16, and 32 times as many parameters at $N = 64, 128, 256$, respectively.

The resampling hierarchy is system, size within system, training seed, and paired test trajectory, using bootstrap random seed 20270921. Within each system–size group, intervals resample seeds and paired trajectories, with random seed $20270921 + N$. All intervals follow the procedure in Appendix C.3.

On KG, the unshared model achieves lower mean error at all three sizes, with relative reductions of 35.8%, 23.3%, and 10.7% compared with the shared model. The accuracy gain decreases as the parameter multiplier increases from 8 to 32. At $N = 256$, removing sharing reduces NRMSE from 0.00187 to 0.00167 while increasing the parameter count from 5,574 to 178,368.

On FPUT, the shared model has lower mean error at $N = 64, 128$, while the unshared model has lower mean error at $N = 256$. The paired shared-to-unshared ratio at $N = 128$ is 0.937 (95% interval: $[0.898, 0.963]$), favoring sharing; the intervals at $N = 64, 256$ include one. These results quantify the accuracy trade-off associated with parameter sharing rather than establishing equivalence between the two configurations.

G Fixed-Checkpoint Cross-Size Evaluation

For each system and training seed, we load the selected $N = 64$ SHNO checkpoint at $N = 128$ and $N = 256$. The complete parameter state is reused without target-size training, normalization refitting, or parameter updates. Prediction starts from the target test initial states. Models trained separately at each target size provide the comparison.

The target-trained models have lower mean pooled rollout error in all four system–target-size groups.

G.1 Time-resolved state and energy errors

Figure 5 presents the complete state and Hamiltonian error curves for both systems. Its state-error panels are reproduced in Figure 3 of the main text. The Hamiltonian panels provide complementary diagnostics over the same 128-step prediction horizon.

For trajectory b, the time-resolved diagnostics are

$$e_x^{(b)}(t) = \frac{\|\widehat{x}_t^{(b)} - x_t^{(b)}\|_2}{\max\{\|x_t^{(b)}\|_2, 10^{-12}\}}, \tag{41}$$

$$e_H^{(b)}(t) = \frac{|H(\widehat{x}_t^{(b)}) - H(x_t^{(b)})|}{\max\{|H(x_t^{(b)})|, 10^{-12}\}}. \tag{42}$$

The Hamiltonian diagnostic measures discrepancy from the reference trajectory at the same time. For each training seed, we average each diagnostic over the eight test trajectories. The curves then

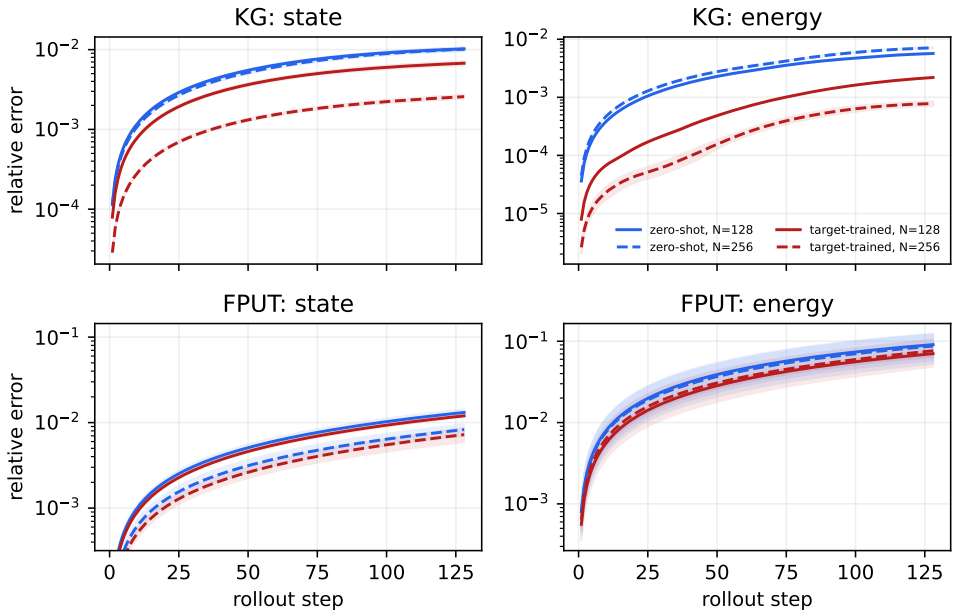

Figure 5: Time-resolved errors over 128 autoregressive steps for fixed-checkpoint deployment and target-size training. The panels show relative state error and relative Hamiltonian discrepancy on KG and FPUT. Blue curves use the unchanged $N = 64$ checkpoint (zero-shot); red curves use models trained at the target size. Curves average the eight test trajectories within each seed and then the three training seeds. Shading denotes one sample standard deviation across the three seed-level curves. Solid and dashed lines denote target sizes $N = 128$ and $N = 256$, respectively. Both vertical axes use logarithmic scales.

Table 7: Toda edge-exchange ablation. Ratios below one favor edge exchange. Intervals are 95% equal-tailed percentile intervals from 20,000 paired hierarchical bootstrap resamples. The final two columns report the cross-patch interaction index and mean relative edge response.

N	Edge/identity ratio	ρ_{cut}	u_{edge}
64	$0.446\ [0.399, 0.507]$	0.245	0.0092
128	$0.332\ [0.287, 0.403]$	0.252	0.0090
256	$0.348\ [0.300, 0.434]$	0.255	0.0123
Pooled	$0.372\ [0.317, 0.445]$	–	–

show the mean across three seed-level curves, with one sample standard deviation computed using $\mathrm{ddof} = 1$.

H Toda Edge-Exchange Ablation

H.1 Paired comparison

We compare identity coupling with a six-reflector edge block at $N = 64, 128, 256$ and three training seeds. Both models use the initial conditions in Eq. (30), $P = r = 8$, and the training schedule in Appendix C. Each model pair is evaluated on the same eight test trajectories.

The primary statistic is the geometric mean of paired edge-to-identity rollout-error ratios. Intervals follow Appendix C.3. Pooled resampling draws system sizes, training seeds within sizes, and paired trajectories within seeds, with bootstrap random seed 2027. Size-specific intervals resample seeds and paired trajectories, with random seed $2027 + N$.

The pooled ratio corresponds to a 62.8% reduction in geometric mean rollout error. Edge exchange increases the parameter count from 5,574 to 5,772, an increase of approximately 3.6%.

H.2 Interaction and edge-response diagnostics

Let H_{cross} contain the Hamiltonian interaction terms connecting different patches. The cross-patch interaction index is

$$\rho_{\mathrm{cut}} = \frac{\left(\mathbb{E}_{\mathrm{val}}\|\nabla H_{\mathrm{cross}}(x)\|_2^2\right)^{1/2}}{\left(\mathbb{E}_{\mathrm{val}}\|\nabla H(x)\|_2^2\right)^{1/2}}, \tag{43}$$

where the expectation averages over validation states. The values are 0.245, 0.252, and 0.255 at $N = 64, 128, 256$.

For each evaluation initial state x_b, let $z_b = \Phi_K(E_K(x_b))$ be the latent state immediately before edge exchange. The relative edge response of a trained model is

$$u_{\mathrm{edge}} = \frac{1}{B}\sum_{b=1}^{B} \frac{\|C_{\theta_C,\mathcal{G}_K,\sigma_K}(z_b) - z_b\|_2}{\|z_b\|_2}. \tag{44}$$

Averaging over the three training seeds gives 0.0092, 0.0090, and 0.0123 at the three sizes. Every trained edge model has a nonzero response, and the square edge-map symplectic residuals remain below 10^{-10}.

The interaction index describes the contribution of cross-patch terms to the Hamiltonian gradient. The edge-response diagnostic measures the magnitude of the learned latent update. The paired rollout comparison measures the predictive effect of introducing that update.

I Implementation Environment

The direct sharing and fixed-checkpoint experiments used PyTorch 2.11.0 and CUDA 12.9 on a server equipped with a NVIDIA A100-SXM4-40GB GPU and an AMD EPYC 7763 64-Core Processor.

The CPU assembly timings were obtained in a separate benchmark using the measurement procedure described in Appendix E.