PHYSICS-INDUCED KOOPMAN LEARNING FOR MESH-BASED DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Finite-dimensional Koopman learning seeks to represent nonlinear dynamics through linear evolution in learned observable coordinates, but a generic learned realization need not respect the physical structure of the target system. We characterize families of linear latent generators with finite interaction radius, consistency across isomorphic local environments, and orientation covariance, showing that each such family is uniquely represented by a shared matrix-valued kernel on local interaction types. When the set of local interaction types is finite, this yields a finite expansion in orientation-graded relation operators with uniquely determined coefficients. We instantiate this reduction as Koopman-MeshFT, combining local covariant encoders and decoders with a growth-controlled continuous-time generator. Across four public PDE benchmarks, Koopman-MeshFT attains the lowest mean rollout error among the evaluated Koopman baselines and improves several benchmark-specific physical diagnostics over them, while remaining competitive with FNO as a nonlinear neural-operator reference. The learned linear realization also enables non-iterative state estimation and forecasting from sparse observations. These results establish a physics-induced hypothesis class for finite-dimensional Koopman learning without assuming exact finite-dimensional closure.

1 INTRODUCTION

Learning dynamical models for continuum physics from observations on meshes is a central problem in scientific machine learning. Recent approaches have achieved strong predictive performance by learning state-update maps with graph-based simulators or operators between function spaces (Pfaff et al., 2021; Li et al., 2021; Wu et al., 2024; Lu et al., 2021; Kovachki et al., 2023; E & Yu, 2018; Gupta et al., 2021; Raissi et al., 2019; Sirignano & Spiliopoulos, 2018; Ummenhofer et al., 2020; Cao, 2021; Battaglia et al., 2018; Noguchi & Kawahara, 2026). A complementary viewpoint is provided by Koopman operator theory, which represents nonlinear state evolution through linear evolution of observables (Koopman, 1931; Mezić, 2005; Williams et al., 2015; Mezić, 2013; 2021; Folkestad et al., 2020; Azencot et al., 2020; Xiong et al., 2024). This provides a linear operator description of nonlinear dynamics and enables spectral and modal analysis. In practice, however, the appropriate observables are unknown, and unconstrained finite-dimensional approximations may learn latent dynamics that are unstable or inconsistent with the structure of the underlying physical system (Williams et al., 2015; Lusch et al., 2018).

For continuum systems represented on meshes, not every finite-dimensional linear dynamics is physically equally plausible. Instantaneous interactions are typically local, the physical law should not depend on arbitrary relabeling of mesh entities, oriented quantities should transform consistently with orientation conventions, and distinguished physical storage quantities may be conserved or dissipated. These requirements are defined for the physical dynamics, but should also restrict how those dynamics are represented in observable space. The central question of this work is therefore: *what class of finite-dimensional Koopman generators is compatible with such physical structure?*

Building on Mesh Field Theory (MeshFT) (Noguchi & Kawahara, 2026), we formulate **(L) locality**, **($\mathbf{P}_{\mathrm{loc}}$) local-law consistency**, a local sharpening of relabeling symmetry, and **(O) orientation covariance** as structural requirements on linear latent generators. These physical requirements characterize generators by shared local kernels: parameter sharing is derived, not independently imposed. For finitely many types, this gives an orientation-graded relation expansion with unique coefficients

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.

in a fixed basis and a coefficient count independent of mesh size. **(E) Storage conservation or dissipation** motivates a growth-controlled subfamily. Figure 1 summarizes this construction, which does not assume exact finite-dimensional Koopman closure.

We instantiate the reduction as *Koopman-MeshFT*: local covariant nonlinear encoders and decoders are learned together with a continuous-time generator in the reduced class. A growth-controlled subfamily regulates amplification of closure errors under the exact latent evolution. Across four public PDE benchmarks (Takamoto et al., 2022), Koopman-MeshFT achieves lower mean rollout error than the evaluated Koopman baselines (KNO (Xiong et al., 2024) and cKAE (Azencot et al., 2020)), while remaining competitive with FNO (Li et al., 2021), a representative nonlinear neural-operator simulator, and improving several conservation- and dissipation-related diagnostics. The same learned dynamics support non-iterative state estimation from sparse observations.

2 Related Work

Koopman learning and finite-dimensional approximation. Koopman operator theory represents nonlinear dynamics through linear evolution on an infinite-dimensional observable space (Mezić, 2005; 2013). Data-driven methods such as DMD and EDMD construct finite-dimensional approximations of the Koopman operator from snapshot data (Schmid, 2010; Williams et al., 2015). Kernel methods replace explicit dictionaries with features in reproducing kernel Hilbert spaces (Kawahara, 2016), while neural approaches learn observables directly, often through autoencoders or latent linear dynamics, to increase expressivity while retaining linear evolution in observable space (Azencot et al., 2020; Lusch et al., 2018; Otto & Rowley, 2019; Folkestad et al., 2020).

Symmetry, stability, and invariant structure in Koopman learning. Prior work has shown that symmetries of the underlying dynamics induce commutation or intertwining relations for Koopman operators and their finite-dimensional approximations, enabling symmetry-adapted representations (Salova et al., 2019; Sinha et al., 2020; Hochrainer & Kar, 2024). More generally, equivariance can be encoded through structured parameter sharing (Ravanbakhsh et al., 2017). Stability, dissipativity, and other structural constraints have also been incorporated into Koopman learning (Hara et al., 2020; Mamakoukas et al., 2023; Xu et al., 2025; Zhang et al., 2024; Khosravi, 2023; Mardt et al., 2020). Physics-informed DMD similarly restricts learned linear operators to matrix classes encoding prescribed physical structure (Baddoo et al., 2023). Related work has further emphasized Koopman-invariant subspaces and the role of learned observables in obtaining useful finite-dimensional representations (Haseli & Cortés, 2023; Takeishi et al., 2017; Brunton et al., 2016). Natural Graph Networks formalize kernel sharing under local isomorphisms (de Haan et al., 2020). We begin instead from the physical requirement that identical local environments obey the same local law and show that, with finite-range interaction, this is equivalent to NGN-style sharing for linear Koopman generators. Thus, within this class, sharing follows from local-law consistency rather than an independent architectural choice. We further impose orientation grading and distinguish finite-type bases from selected continuous-attribute kernel spaces.

Learning mesh-based physics and structure-preserving dynamics. Graph-based simulators learn mesh dynamics through local message passing (Battaglia et al., 2018; Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Brandstetter et al., 2022), with related graph methods incorporating conservation and physical symmetries directly into the architecture (Horie & Mitsume, 2024). Natural graph networks further formalize consistency across graph isomorphisms and local graph structure (de Haan et al., 2020). Neural operators instead learn mappings between function spaces and naturally support transfer across discretizations (Li et al., 2021; Kovachki et al., 2023; Xiong et al., 2024), with group-equivariant variants explicitly incorporating geometric symmetries (Helwig et al., 2023). Structure-preserving mesh methods based on discrete exterior calculus retain topological operators while learning metric-dependent components (Trask et al., 2022). Also, MeshFT derives a topology and metric separation from locality, relabeling equivariance, orientation covariance, and energy balance, and uses this reduction to construct a structure-preserving mesh simulator (Noguchi & Kawahara, 2026). Other approaches impose physical structure through Hamiltonian, Lagrangian, symplectic, or thermodynamic formulations (Greydanus et al., 2019; Cranmer et al., 2020; Zhong et al., 2020; Zhang et al., 2022; Lee et al., 2021; Hernández et al., 2021; Desai et al., 2021). Our work differs primarily in its target and construction. Rather than prescribing a state-space dynamics template or directly learning a nonlinear simulator, we use physical principles to specify the hypothesis class of a

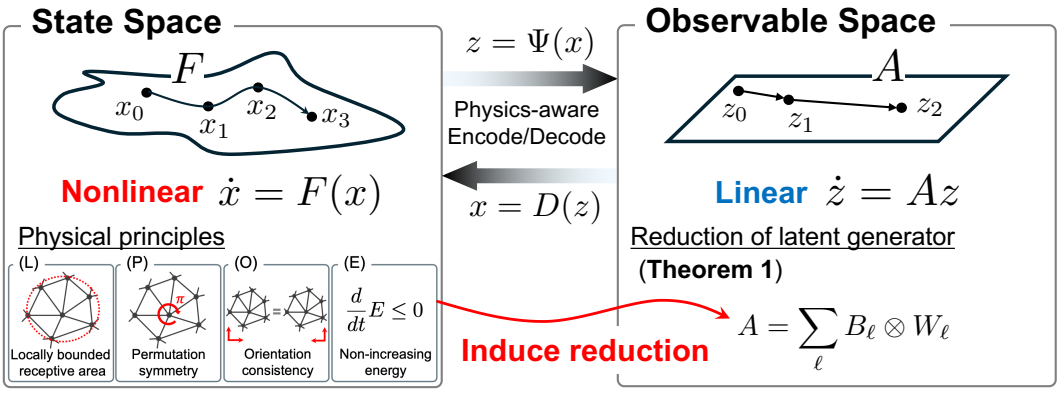


Figure 1: Overview of Koopman-MeshFT. Locality, local-law consistency, and orientation covariance characterize the latent generator through a shared type kernel. A finite type set gives the graded relation expansion. Storage considerations motivate growth control, while local covariant encoders and decoders connect the state and observable spaces.

finite-dimensional Koopman realization, including local shared observables and relation-wise shared generator maps, without requiring explicit knowledge of the governing equations.

3 Physical Principles and Generator Reduction

We first distinguish structural requirements on physical dynamics from those used to construct a finite-dimensional linear observable model. The latter admit a reduction in which shared coefficients follow from equality of local responses.

3.1 Mesh-based dynamics and physical principles

Let $\mathfrak{M}$ be a family of finite attributed relational meshes. Each $\mathcal{K}$ has entities $\Sigma_\mathcal{K}$, with $N_\mathcal{K} = |\Sigma_\mathcal{K}|$, and fixed primitive relations encoding connectivity, directions, and selected geometric, material, or boundary attributes. Their undirected support graph defines $d_\mathcal{K}$ and $B_q^\mathcal{K}(\sigma) = \{\tau : d_\mathcal{K}(\sigma, \tau) \leq q\}$. Physically, these relations describe the local environment in which a dynamical law acts: what can interact, along which directions, and under which local attributes. They specify this environment rather than the law itself, and are fixed before any generator is chosen. They are distinct from the complete relation basis constructed below. The physical field $x \in \mathcal{X}_\mathcal{K} \subseteq \mathbb{R}^{N_\mathcal{K}} \otimes V_X$, with $V_X \simeq \mathbb{R}^{d_x}$, evolves as $\dot{x} = F_\mathcal{K}(x)$ with flow $\phi_t^\mathcal{K}$. We work on compact trajectory tubes where $F_\mathcal{K}$ is locally Lipschitz and all derivatives and compositions used below exist. Following MeshFT (Noguchi & Kawahara, 2026), we consider four physical requirements.

(L) Locality. $F_{\mathcal{K},\sigma}(x)$ depends only on $x|_{B^\mathcal{K}_{R_F}(\sigma)}$ for a uniformly bounded interaction radius R_F. **(P) Permutation symmetry.** For a mesh isomorphism f with permutation matrix P_f, $F_{\mathcal{K}'}(\rho_X(f)x) = \rho_X(f)F_\mathcal{K}(x)$, where $\rho_X(f) = P_f \otimes I_{V_X}$. **(O) Orientation covariance.** A specified orientation involution ω preserves entity labels and the undirected support graph. With the channel grading $S_X = \mathrm{diag}(s_c^X)$, $s_c^X \in \{+1, -1\}$, and $\rho_X(\omega) = I \otimes S_X$, the law satisfies $F_{\omega\mathcal{K}}(\rho_X(\omega)x) = \rho_X(\omega)F_\mathcal{K}(x)$. **(E) Storage non-increase.** A physical storage $H_\mathcal{K}$ satisfies $DH_\mathcal{K}[F_\mathcal{K}] = 0$ for conservative dynamics, or $DH_\mathcal{K}[F_\mathcal{K}] \leq 0$ for dissipative dynamics.

The Koopman formulation represents the nonlinear flow through linear evolution of observables (Mezić, 2005; Budišić et al., 2012): for $\mathcal{U}_t^\mathcal{K} g = g \circ \phi_t^\mathcal{K}$, its infinitesimal generator is $\mathcal{L}_{F_\mathcal{K}} g = Dg[F_\mathcal{K}]$. The above physical principles then constrain distinct aspects of this exact observable dynamics. Locality (L) gives $\mathcal{L}_{F_\mathcal{K}} \mathcal{A}_q^\mathcal{K}(\sigma) \subseteq \mathcal{A}_{q+R_F}^\mathcal{K}(\sigma)$, where $\mathcal{A}_q^\mathcal{K}(\sigma)$ contains sufficiently smooth observables supported in $B_q^\mathcal{K}(\sigma)$. Relabeling (P) and orientation covariance (O) induce the corresponding pullback intertwining identities. Also, storage (E) gives $\mathcal{U}_t^\mathcal{K} H_\mathcal{K} = H_\mathcal{K}$ or $\mathcal{U}_t^\mathcal{K} H_\mathcal{K} \leq H_\mathcal{K}$. These consequences concern the exact Koopman dynamics and do not imply finite-dimensional closure.

3.2 LOCAL-LAW CONSISTENCY

We characterize linear latent dynamics $A_{\mathcal{K}}$ on $Z_{\mathcal{K}} = \bigoplus_{\sigma \in \Sigma_{\mathcal{K}}} V_Z$, where $V_Z = \mathbb{R}^{d_z}$ is a shared latent feature space attached to each mesh entity. The channel space is fixed across the mesh family: relabeling permutes entity indices, while orientation acts on designated latent channels through $S_Z = \mathrm{diag}(s_a^Z)$, $s_a^Z \in \{\pm 1\}$. Fix $0 \le R \le r$. The interaction radius R specifies the receptive field of the linear dynamics, with $A_{\mathcal{K},\sigma\leftarrow\tau} = 0$ whenever $d_{\mathcal{K}}(\sigma,\tau) > R$, while r specifies the local structural context allowed to determine these interaction coefficients. The patch $\mathcal{P}_r(\mathcal{K},\sigma)$ contains the primitive relations and attributes in $B_r^{\mathcal{K}}(\sigma)$, and two patches are isomorphic when they represent the same local environment up to relabeling, with the center entity and with relation types, directions, and specified attributes preserved. We assume this patch structure is preserved under relabeling and orientation.

Definition 1 (Local-law consistency)**.** For every patch isomorphism $f : \mathcal{P}_r(\mathcal{K},\sigma) \to \mathcal{P}_r(\mathcal{K}',\sigma')$ and every pair of latent fields whose features agree under the correspondence, $z'_{f(\tau)} = z_\tau$ on the patch,

$$(A_{\mathcal{K}'} z')_{\sigma'} = (A_{\mathcal{K}} z)_\sigma. \tag{1}$$

Local-law consistency requires matching latent states in isomorphic local physical environments to produce the same response, independently of the surrounding mesh. This strengthens whole-mesh relabeling consistency: $A_{\mathcal{K}} = -N_{\mathcal{K}} I$ is local and relabeling equivariant but gives different responses to matching patches in differently sized meshes. The condition holds for all latent fields, not only encoded training states.

3.3 REDUCTION TO SHARED RELATION GENERATORS

We characterize the full generator family under the prescribed local-law specification, rather than posit a generator basis. For $d_{\mathcal{K}}(\sigma,\tau) \le R$, let $\alpha(\mathcal{K},\sigma,\tau) = [\mathcal{P}_r(\mathcal{K},\sigma),\sigma,\tau]$ denote the local interaction type. Write $\mathcal{T}_{R,r}$ for all such types realized across $\mathfrak{M}$. Orientation induces an involution on this set. The following result shows that these local types contain exactly the structural information needed to parameterize every locally consistent generator.

Theorem 1 (Reduction of locally consistent linear generators)**.** *A family of linear generators has interaction radius at most R and satisfies local-law consistency if and only if there exists a unique shared kernel $W : \mathcal{T}_{R,r} \to \mathbb{R}^{d_z \times d_z}$ such that $(A_{\mathcal{K}} z)_\sigma = \sum_{\tau \in B_R^{\mathcal{K}}(\sigma)} W(\alpha(\mathcal{K},\sigma,\tau)) z_\tau$. Orientation covariance $A_{\omega\mathcal{K}}(I \otimes S_Z) = (I \otimes S_Z) A_{\mathcal{K}}$ holds exactly when $W(\omega\alpha) = S_Z W(\alpha) S_Z$. If $\mathcal{T}_{R,r}$ is finite, this kernel form admits a fixed scalar relation-family basis $\{B_{\ell,\mathcal{K}}\}_{\ell \in \Lambda_{\mathrm{red}}}$ in which every such covariant generator has the unique representation*

$$A_{\mathcal{K}} = \sum_{\ell \in \Lambda_{\mathrm{red}}} B_{\ell,\mathcal{K}} \otimes W_\ell, \quad S_Z W_\ell = \chi_\ell W_\ell S_Z. \tag{2}$$

Each basis family has radius at most R, is locally consistent, and satisfies $B_{\ell,\omega\mathcal{K}} = \chi_\ell B_{\ell,\mathcal{K}}$, $\chi_\ell \in \{+1,-1\}$. Conversely, any coefficients satisfying this grading rule define an admissible family. Uniqueness is relative to the fixed basis and channel coordinates over the mesh family.

The proof is in Appendix A.3. Thus, local-law consistency turns mesh-dependent interaction blocks into channel maps shared by local structural type, while orientation further constrains their allowed channel couplings. With finitely many relation types and fixed channel width, the resulting generator space has dimension independent of mesh size (Appendix A.4). Finite type is an additional assumption on the mesh family, for example bounded degree with a finite attribute alphabet. When continuous attributes make $\mathcal{T}_{R,r}$ infinite, restricting the kernel to a selected finite-dimensional function space still yields a finite graded relation representation for that restricted model class (Appendix A.5).

On a homogeneous one-dimensional periodic grid with fixed local attributes and period exceeding $2r+1$, the radius-one types are self, left, and right, so the reduction becomes the familiar shared stencil $(Az)_i = W_0 z_i + W_- z_{i-1} + W_+ z_{i+1}$, with the same channel maps at every site. Higher-dimensional directional grids analogously assign one shared map to self and each axis-aligned neighbor, giving the class used in our experiments (Appendix A.6). The theorem characterizes this admissible generator form, but does not guarantee exact finite-dimensional Koopman closure for arbitrary nonlinear dynamics.

Algorithm 1 Learning a reduced Koopman realization

Require: Training window $(x_0, \dots, x_h)$ and mesh $\mathcal{K}$
1: Encode $z_0 \leftarrow \Psi_{\theta,\mathcal{K}}(x_0)$
2: Build $A_{\eta,\mathcal{K}}$ from shared local relations and set $\widetilde{K}_\Delta \leftarrow \widetilde{\mathcal{E}}_\Delta^{M,s}(A_{\eta,\mathcal{K}})$
3: **for** $j = 0, \dots, h-1$ **do**
4: $\quad z_{j+1} \leftarrow \widetilde{K}_\Delta z_j, \quad \widehat{x}_{j+1} \leftarrow \mathcal{D}_{\zeta,\mathcal{K}}(z_{j+1})$
5: **end for**
6: Update (θ, η, ζ) using reconstruction, rollout, closure, and cycle-consistency losses

3.4 Finite-time propagation

Having characterized the admissible infinitesimal generator, we next ask how its locality appears at finite time. Locality of the reduced generator does not imply finite support of its exponential. For $R > 0$, entities in the same connected component, and $m = \lceil d_{\mathcal{K}}(\sigma, \tau)/R \rceil$, the induced Euclidean norm gives

$$\|(e^{tA})_{\sigma \leftarrow \tau}\| \le \sum_{k=m}^{\infty} \frac{(t\|A\|)^k}{k!} \le e^{t\|A\|} \frac{(t\|A\|)^m}{m!}, \qquad t \ge 0. \tag{3}$$

This sparse-matrix-function estimate (Benzi & Razouk, 2007) follows from the supports of the powers of A (Appendix A.7). Thus finite-time influence can extend beyond the generator radius, but is progressively suppressed with graph distance. Different components do not interact, and $R = 0$ gives block-diagonal propagation. For numerical propagation, we use the substepped Taylor action $\widetilde{\mathcal{E}}_\Delta^{M,s}(A) = \left[\sum_{k=0}^{M} \frac{1}{k!} \left(\frac{\Delta A}{s}\right)^k\right]^s$, where M is the order and s the number of substeps. Its support radius is at most MsR. Exact quasi-local evolution and finite-support numerical propagation are therefore distinct.

4 Learning the Reduced Koopman Realization

The reduction above specifies the hypothesis class for the latent linear dynamics. For each mesh, we learn an observable encoder $\Psi_{\theta,\mathcal{K}}$, a state-independent generator $A_{\eta,\mathcal{K}}$, and a decoder $\mathcal{D}_{\zeta,\mathcal{K}}$: $z_0 = \Psi_{\theta,\mathcal{K}}(x_0)$, $z(t) = e^{tA_{\eta,\mathcal{K}}} z_0$, and $\widehat{x}(t) = \mathcal{D}_{\zeta,\mathcal{K}}(z(t))$. Thus, nonlinear state dynamics are modeled by a nonlinear lift, linear latent evolution, and nonlinear decoding, with finite-dimensional closure learned approximately from data. The resultant procedure is shown in Algorithm 1.

4.1 Local observables and shared relation weights

Starting from the reduced relation families above, let $\{\widehat{B}_{\ell,\mathcal{K}}\}$ be a selected local dictionary, which may span the complete reduced class, as for the periodic stencils, or a smaller model class. We use the same relation structure to parameterize local nonlinear observables: $\Psi_{\theta,\mathcal{K}}(x)_\sigma = \psi_\theta(x_\sigma, \{(\widehat{B}_{\ell,\mathcal{K}} x)_\sigma\}_{\ell \in \Lambda_E})$, with the local network ψ_θ shared across entities, and construct the decoder analogously. This relation-feature parameterization is chosen to satisfy locality and the required relabeling and orientation covariance, but is not a complete characterization of all admissible nonlinear encoders and decoders. For the latent generator, the reduction gives the direct shared-relation parameterization $A_{\eta,\mathcal{K}} = \sum_{\ell \in \Lambda_A} \widehat{B}_{\ell,\mathcal{K}} \otimes W_{\eta,\ell}$. For orientation-aware channels, $(W_{\eta,\ell})_{ab} = 0$ unless $s_a^Z = \chi_\ell s_b^Z$, so even relations connect equal grades and odd relations connect opposite grades. With compatible encoder and decoder actions, relabeling and orientation covariance are inherited by both e^{tA} and its polynomial approximations.

4.2 Consistency learning and growth-controlled latent dynamics

For a training window $(x_0, \dots, x_h)$, let $\widetilde{K}_\Delta = \widetilde{\mathcal{E}}_\Delta^{M,s}(A)$ and $z_j = \widetilde{K}_\Delta^j \Psi(x_0)$. In addition to reconstruction and rollout losses, we use $\delta_j = \widetilde{K}_\Delta \Psi(x_j) - \Psi(x_{j+1})$, and $r_{\mathrm{cyc},j} = \Psi(\mathcal{D}(z_j)) - z_j$, where δ_j is the data-closure residual and $r_{\mathrm{cyc},j}$ is the manifold-cycle residual along the latent rollout. Their losses penalize mean squared residuals. Importantly, $\widetilde{K}_\Delta^h \Psi(x_0) - \Psi(x_h) = \sum_{j=0}^{h-1} \widetilde{K}_\Delta^{h-1-j} \delta_j$, so long-horizon error depends not only on closure defects but also on how the learned dynamics propagate them. Full loss definitions are given in Appendix C.1.

Table 1: Predictive fidelity and model size for benchmarks from PDEBench Takamoto et al. (2022). Rollout relative L^2 errors use training-standardized states, 19 transitions, and five seeds. Error cells give the mean above $\pm$ standard deviation. Boldface marks the lowest mean error and parameter count within the displayed *Koopman* group. FNO is a nonlinear reference. Parameter counts are real trainable scalars on CNS2D. M denotes million.

Method	Advection	Burgers	CNS1D	CNS2D	CNS2D parameters
Koopman-based models					
Koopman-MeshFT	**0.00709** $\pm$ 0.00033	**0.1556** $\pm$ 0.0089	**0.2880** $\pm$ 0.0147	**0.5171** $\pm$ 0.0113	**44,500**
KNO	0.04367 $\pm$ 0.00348	0.2917 $\pm$ 0.0072	0.5598 $\pm$ 0.0765	0.5851 $\pm$ 0.0186	306,764
cKAE	0.68165 $\pm$ 0.00529	0.6501 $\pm$ 0.0088	0.7500 $\pm$ 0.0310	0.7908 $\pm$ 0.0355	38.85 M
Nonlinear reference					
FNO	0.02495 $\pm$ 0.00250	0.1521 $\pm$ 0.0123	0.2515 $\pm$ 0.0206	0.4340 $\pm$ 0.0193	2.37 M

Motivated by this error propagation, the principal model uses a growth-controlled subfamily of the reduced generators. In transformed latent coordinates, we combine a skew-symmetric mixing term with on-site damping. Let $G_{\mathcal{K}} = \sum_{\ell} \widehat{B}_{\ell,\mathcal{K}} \otimes V_{\ell}$ and $T_{\mathcal{K}} = I_{N_{\mathcal{K}}} \otimes L$, where L is invertible and grade preserving. Assuming $\widehat{B}_{0,\mathcal{K}} = I$ and $\widehat{B}_{\ell,\mathcal{K}}^{\top} = \widehat{B}_{\bar{\ell},\mathcal{K}}$ with $\chi_{\bar{\ell}} = \chi_{\ell}$, we set

$$A_{\eta,\mathcal{K}} = T_{\mathcal{K}}^{-1} \left[\tfrac{1}{2}(G_{\mathcal{K}} - G_{\mathcal{K}}^{\top}) + I_{N_{\mathcal{K}}} \otimes (\gamma I - Q^{\top} Q) \right] T_{\mathcal{K}}, \tag{4}$$

with grade-compatible V_{ℓ}, Q and prescribed growth rate γ. This remains in the selected reduced relation class and guarantees $\|e^{tA}\|_{P_{\mathcal{K}}} \leq e^{\gamma t}$ for $P_{\mathcal{K}} = I \otimes L^{\top} L$ (Appendix A.9). This is a sufficient growth-controlled subfamily, not the complete class of dissipative generators, and the latent metric is distinct from physical storage and does not enforce decoded conservation or positivity.

5 Experiments

We assess the fidelity of a realization learned in the reduced generator class through state prediction and physical diagnostics. Ablations examine the class construction, consistency objectives, and latent amplification. A sparse-observation protocol tests reuse of the frozen dynamics for state estimation.

Benchmarks and evaluation. The benchmarks are Advection, Burgers, one-dimensional compressible Navier Stokes (CNS1D), and periodic compressible Navier Stokes at 64×64 resolution (CNS2D) from PDEBench (Takamoto et al., 2022). The main prediction and continuous/discrete comparisons use 19 forecast transitions from observed initial states and five seeds. The generator-class and loss ablations also use seeds 0 through 4. Means and standard deviations are reported across model seeds. Physical diagnostics use decoded states in physical coordinates. Evaluation definitions and run provenance are specified in Appendix C.1 and Appendix B.5.

Models and numerical realization. KNO (Xiong et al., 2024) and cKAE (Azencot et al., 2020) provide Koopman-based baselines. FNO (Li et al., 2021) is a nonlinear prediction reference without the requirement of a fixed linear latent evolution. Koopman-MeshFT uses the local continuous generator with the latent envelope described in Section 4.2. Continuous models in the main tables use the polynomial action (Appendix C).

5.1 Predictive fidelity of the learned Koopman realization

Table 1 reports learned-observable prediction. Koopman–MeshFT has the lowest error among evaluated Koopman models on four benchmarks. The improvements show that the principle-induced parameterization can represent these nonlinear trajectories while retaining linear latent evolution. On CNS2D, the reported propagation agrees with an all-mode matrix-exponential audit (Appendix C.1). Relative to FNO, it is better on Advection, similar on Burgers, and less accurate on CNS1D/2D.

Relation-wise sharing also gives a compact parameterization. For fixed relation types and channel widths, parameter count is independent of the number of mesh entities, while the latent state dimension

Table 2: Physical fidelity over five seeds (mean $\pm$ standard deviation). L^2 drift measures change in a conserved quantity. Burgers balance and increment errors assess energy dissipation and successive energy changes. CNS1D composite drift combines conservation-related components. CNS2D conservation error compares predicted and reference changes from their respective initial quantities, while tracking compares the quantity trajectories. Boldface marks the lowest mean among the displayed *Koopman-based* models. FNO is a nonlinear reference. Component results and metric specifications are in Appendix B.5.

(a) Conservation and dissipation

Method	Advection L^2 drift	Burgers balance error	Burgers increment error	CNS1D composite drift
Koopman-MeshFT	$\mathbf{0.00939} \pm 0.00046$	$\mathbf{0.804} \pm 0.138$	$\mathbf{2.016} \pm 0.414$	$\mathbf{0.264} \pm 0.070$
KNO	0.06209 ± 0.00480	3.936 ± 0.595	12.046 ± 1.997	0.889 ± 0.161
cKAE	0.37694 ± 0.02534	2.666 ± 0.272	7.808 ± 0.775	0.740 ± 0.153
FNO	0.00791 ± 0.00275	0.985 ± 0.388	2.740 ± 1.134	0.210 ± 0.033

(b) CNS2D physical-quantity evolution

Method	Mass conservation error	Energy conservation error	Quantity tracking
Koopman-MeshFT	$\mathbf{0.0274} \pm 0.0159$	$\mathbf{0.0539} \pm 0.0370$	$\mathbf{0.1531} \pm 0.0294$
KNO	0.1564 ± 0.0043	0.1865 ± 0.0203	0.1851 ± 0.0221
cKAE	0.0562 ± 0.0445	0.0939 ± 0.0842	0.2869 ± 0.0652
FNO	0.0146 ± 0.0038	0.0234 ± 0.0056	0.0879 ± 0.0384

is $N_{\mathcal{K}} d_z$. The model uses $44{,}500$ parameters on CNS2D, fewer than the three references in the table. This concerns the size of the learned rule, not execution speed, which also depends on mesh size and the repeated operator applications in the numerical exponential.

5.2 Physical fidelity of the learned dynamics

We next assess whether the learned realization reproduces the conservation and dissipation behavior of the target systems. Table 2 evaluates this physical fidelity in the decoded state space, alongside aggregate prediction error.

Conservation and dissipation. Koopman-MeshFT improves the reported conservation and dissipation diagnostics over KNO and cKAE on Advection, Burgers, and CNS1D. Reduced Advection L^2 drift and CNS1D composite drift are accompanied by lower Burgers balance and energy-increment errors. The latter distinguish matching the amount of dissipation from merely avoiding positive energy increments: KNO has fewer positive increments but larger dissipation errors. FNO gives lower Advection L^2 drift and CNS1D composite drift, whereas Koopman-MeshFT gives lower Burgers dissipation errors and CNS1D mass drift. Component-wise results are in Appendix B.5.

Mass and energy evolution on CNS2D. Koopman-MeshFT also gives lower mass and total-energy conservation errors and lower quantity-tracking error than KNO and cKAE. The conservation diagnostic compares changes from each trajectory's own initial quantities, while tracking additionally reflects initial reconstruction error. The two therefore measure different aspects of the predicted evolution. FNO has lower errors in these metrics. Mass- and energy-specific tracking results and additional one-dimensional diagnostics are reported in Appendix B.5. Note that the improvements in mass, energy, and tracking are empirical benefits of the learned linear realization, not hard conservation guarantees. Also, Appendix B.1 shows that Koopman-MeshFT preserves accuracy under an unseen orientation reversal and remains close to FNO under geometric mesh perturbations.

5.3 Ablation of the Koopman realization

The following comparisons examine the generator hypothesis class, temporal parameterization, and observable-consistency objectives.

Table 3: Generator comparison over five seeds. Both models include latent envelopes. Errors are rollout relative L^2 means $\pm$ standard deviations. Boldface marks the lower error and parameter count.

	Burgers		CNS1D	
Generator	Error	Params.	Error	Params.
Local/shared	$\mathbf{0.1556} \pm 0.0089$	$\mathbf{39{,}377}$	$\mathbf{0.2880} \pm 0.0147$	$\mathbf{40{,}659}$
Global low-rank	0.3609 ± 0.0166	71,881	0.5659 ± 0.0721	73,163

Table 4: Temporal parameterization and consistency learning over five seeds. Both panels share the CNS1D Full rollout reference. Rollout errors are means $\pm$ standard deviations. One-step and multi-step closure MSEs are reported in units of 10^{-4} and 10^{-3}, respectively. Boldface marks the lowest mean within each benchmark and panel. Full statistics are in Appendix B.2.

(a) Temporal parameterization and amplification control

Variant	Advection	Burgers	CNS1D	CNS2D
Continuous + envelope	$\mathbf{0.00709} \pm 0.00033$	$\mathbf{0.1556} \pm 0.0089$	$\mathbf{0.2880} \pm 0.0147$	$\mathbf{0.5171} \pm 0.0113$
Continuous	0.01186 ± 0.00073	0.8124 ± 0.2670	0.3595 ± 0.0399	2.1988 ± 3.6428
Discrete + envelope	0.3243 ± 0.0580	0.4637 ± 0.0285	1.1076 ± 0.3192	0.5463 ± 0.0033
Discrete	0.01097 ± 0.00392	1.2958 ± 0.8345	298.1 ± 80.3	20.90 ± 20.11

(b) Observable-consistency objectives

Variant	Rollout rel. L^2	One-step closure	Multi-step closure
Burgers			
Koopman-MeshFT	0.1556 ± 0.0089	$\mathbf{1.53}$	$\mathbf{6.43}$
w/o data closure	$\mathbf{0.1442} \pm 0.0135$	6.55	9.80
w/o manifold cycle	0.1623 ± 0.0122	1.90	9.35
CNS1D			
Koopman-MeshFT	0.2880 ± 0.0147	$\mathbf{4.04}$	$\mathbf{7.97}$
w/o data closure	$\mathbf{0.2870} \pm 0.0090$	13.8	21.6
w/o manifold cycle	0.3071 ± 0.0303	5.69	12.9

Local and shared generator. Table 3 compares local/shared and global low-rank generators with envelopes. Across five paired seeds, local/shared approximately halves mean rollout error on Burgers and CNS1D with 45% fewer parameters, while spatial mixing, sharing, rank, and realized amplification differ jointly. In a separate matched CNS2D comparison against a full-support shared generator, the local model has 52% lower mean physical-coordinate rollout error with 14.5% as many total parameters (Appendix B.3). Encoder-neighborhood features are evaluated separately (Appendix B.4).

Temporal parameterization and amplification control. Table 4 (a) compares the polynomial exponential with a directly learned local step operator, each with and without an envelope. The former constrains the support of the infinitesimal generator, while the latter constrains the finite-time step. The continuous model with an envelope gives the lowest mean rollout error on all four benchmarks. The envelope improves both parameterizations on Burgers and the compressible systems, but worsens the discrete Advection result. The finite-horizon identity in Section 4.2 separates closure defects from their subsequent propagation. On Burgers, the plain and envelope-equipped models have similar one-step closure errors, but their multi-step closure errors differ by about a factor of 39. This is consistent with the role of residual propagation in long-horizon error.

Data closure and manifold consistency. The objectives in Section 4.2 act on different residuals. Removing data closure lowers mean rollout error on Burgers and has little effect on CNS1D, but increases both one-step and multi-step latent closure MSEs (Table 4 (b)). Including data closure therefore improves measured latent consistency without consistently improving decoded prediction. Latent MSE depends on the scale of the learned coordinates. Removing the manifold cycle increases mean rollout error by 4.3% and 6.6% on Burgers and CNS1D. The generator parameterization, observable objectives, and envelope thus have distinct roles in the learned realization. Complete statistics and paired differences are in Appendix B.2.

Table 5: Sparse-observation state estimation and forecasting on Burgers and CNS2D. State-estimation error at t_3 and forecast error over $t_4, \ldots, t_{19}$ are physical relative L^2 errors. For CNS2D, mass and total-energy tracking errors are also reported. Values are mean $\pm$ sample standard deviation over five model seeds after averaging trajectories and sensor realizations, with three sensor placements additionally averaged for CNS2D. End-to-end latency includes state estimation and forecasting. CNS2D times are reference measurements. Placement-specific results are provided in Appendix D. Lower is better. Boldface indicates the lower mean within each benchmark and metric, not statistical significance.

Benchmark	Method	State estimation	Forecast Rel. L^2	E2E latency (ms)
Burgers	Koopman-MeshFT	0.2539 ± 0.0064	$\mathbf{0.2885 \pm 0.0036}$	$\mathbf{4.22 \pm 0.13}$
	Inverse FNO	$\mathbf{0.2528 \pm 0.0077}$	0.3151 ± 0.0308	1176.71 ± 26.69
CNS2D	Koopman-MeshFT	0.2925 ± 0.1121	0.2078 ± 0.0823	$\mathbf{5.17 \pm 0.35}$
	Inverse FNO	$\mathbf{0.2054 \pm 0.0091}$	$\mathbf{0.2073 \pm 0.0133}$	1375.86 ± 127.16

CNS2D physical-observable tracking error		
Method	Mass	Total energy
Koopman-MeshFT	$\mathbf{0.0279 \pm 0.0097}$	$\mathbf{0.0399 \pm 0.0103}$
Inverse FNO	0.0665 ± 0.0040	0.0894 ± 0.0066

5.4 Sparse-Observation State Estimation and Forecasting

Beyond forward prediction, the learned linear dynamics support a precomputable sparse-state estimator through an auxiliary affine readout fitted on training data. We test this capability with frozen dynamics, retaining the original nonlinear encoder and decoder for forecasting. For Burgers and CNS2D, observations over $t_0, \ldots, t_3$ estimate the state at t_3 and forecast $t_4, \ldots, t_{19}$ with all dynamical weights fixed. Koopman-MeshFT reduces this to a regularized linear inverse problem with a precomputable estimator $\widehat{x}_3 = M_S \mathbf{y} + a_S$, followed by one reencoding and native rollout. FNO inverse instead estimates its own initial state by iterative optimization through a frozen FNO from the same observations and fixed initializations. Full settings are given in Appendix D.

Table 5 shows that Koopman-MeshFT achieves competitive sparse-observation forecasting without per-query iterative optimization. On Burgers, the two methods have nearly identical state-estimation errors, while Koopman-MeshFT gives lower forecast error. On CNS2D, inverse FNO estimates the current state more accurately, whereas their mean forecast errors are nearly identical, with larger model-seed variation for Koopman-MeshFT. Koopman-MeshFT also gives lower CNS2D mass and total-energy tracking errors and substantially lower recorded end-to-end latency on both benchmarks. These results show that the learned linear realization can serve not only as a forward model, but also as an efficient inference model that converts partial observations into physically meaningful future predictions without solving a nonlinear inverse problem at each query.

6 Conclusion

We establish a physics-induced characterization of linear latent generators: locality, local-law consistency, and orientation covariance require shared kernels, with a complete graded relation expansion for finite local types. Koopman-MeshFT learns nonlinear observables and growth-controlled generators within this class, without assuming exact Koopman closure. Across four public PDE benchmarks, it outperforms the evaluated Koopman baselines in mean rollout error and improves several conservation and dissipation diagnostics. Ablations support locality, but better latent closure need not improve prediction. The same frozen realization enables non-iterative sparse-state estimation, providing a common structural basis for prediction and inference.

Scope and limitations. Finite-basis completeness requires a finite local type set or additional restrictions on attribute dependence. The experiments cover periodic grids and controlled geometric perturbations, but do not establish arbitrary geometry transfer. The latent envelope does not guarantee decoded conservation, and compact parameterization does not imply faster numerical propagation.

AI USE STATEMENT

We used generative AI tools to assist with theoretical formulation and proofs, experimental design, code implementation and debugging, analysis and interpretation of results, and manuscript drafting and editing. The authors reviewed and revised AI-assisted contributions, made all final research decisions, and take full responsibility for the claims, results, code, and text.

REPRODUCIBILITY STATEMENT

The assumptions and proofs are provided in Appendix A. Experimental settings and evaluation procedures are described in Appendices C and D. The supplementary source package provides model implementations, data-preparation scripts, configurations, and evaluation entry points.

REFERENCES

Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 475–485, 2020.

Peter J. Baddoo, Benjamin Herrmann, Beverley J. McKeon, J. Nathan Kutz, and Steven L. Brunton. Physics-informed dynamic mode decomposition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 479(2271):20220576, 2023.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Michele Benzi and Nader Razouk. Decay bounds and o (n) algorithms for approximating functions of sparse matrices. *Electronic Transactions on Numerical Analysis*, 28(16-39), 2007.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016.

Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.

Shuhao Cao. Choose a transformer: Fourier or galerkin. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24924–24940, 2021.

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

Pim de Haan, Taco Cohen, and Max Welling. Natural graph networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3636–3646, 2020.

Shaan A. Desai, Marios Mattheakis, David Sondak, Pavlos Protopapas, and Stephen J. Roberts. Port-hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Phys. Rev. E*, 104:034312, 2021.

Weinan E and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

Carl Folkestad, Daniel Pastor, Igor Mezic, Ryan Mohr, Maria Fonoberova, and Joel Burdick. Extended dynamic mode decomposition with learned koopman eigenfunctions for prediction and control. In *2020 american control conference (acc)*, pp. 3906–3913. IEEE, 2020.

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

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24048–24062, 2021.

Keita Hara, Masaki Inoue, and Noboru Sebe. Learning koopman operator under dissipativity constraints. *IFAC-PapersOnLine*, 53(2):1169–1174, 2020.

Masih Haseli and Jorge Cortés. Generalizing dynamic mode decomposition: Balancing accuracy and expressiveness in koopman approximations. *Automatica*, 153:111001, 2023.

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant Fourier neural operators for partial differential equations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12907–12930. PMLR, 2023.

Quercus Hernández, Alberto Badías, David González, Francisco Chinesta, and Elías Cueto. Structure-preserving neural networks. *Journal of Computational Physics*, 426:109950, 2021.

Thomas Hochrainer and Gurudas Kar. Approximation of translation invariant koopman operators for coupled non-linear systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(8), 2024.

Masanobu Horie and Naoto Mitsume. Graph neural PDE solvers with conservation and similarity-equivariance. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18785–18814. PMLR, 2024.

Yoshinobu Kawahara. Dynamic mode decomposition with reproducing kernels for koopman spectral analysis. *Advances in neural information processing systems*, 29, 2016.

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

Mohammad Khosravi. Representer theorem for learning koopman operators. *IEEE Transactions on Automatic Control*, 68(5):2995–3010, 2023.

B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

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.

Kookjin Lee, Nathaniel Trask, and Panos Stinis. Machine learning structure preserving brackets for forecasting irreversible processes. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

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.

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.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Giorgos Mamakoukas, Ian Abraham, and Todd D. Murphey. Learning stable models for prediction and control. *IEEE Transactions on Robotics*, 39(3):2255–2275, 2023. doi: 10.1109/TRO.2022.3228130.

Andreas Mardt, Luca Pasquali, Frank Noé, and Hao Wu. Deep learning markov and koopman models with physical constraints. In *Mathematical and Scientific Machine Learning*, pp. 451–475, 2020.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

Igor Mezić. Analysis of fluid flows via spectral properties of the koopman operator. *Annual review of fluid mechanics*, 45(1):357–378, 2013.

Igor Mezić. Koopman operator, geometry, and learning of dynamical systems. *Not. Am. Math. Soc*, 68(7):1087–1105, 2021.

Satoshi Noguchi and Yoshinobu Kawahara. Mesh Field Theory: Port–Hamiltonian formulation of mesh-based physics. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. Accepted at ICML 2026; `https://arxiv.org/abs/2605.00394`.

Samuel E. Otto and Clarence W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. doi: 10.1137/18M1177846.

Amnon Pazy. *Semigroups of linear operators and applications to partial differential equations*. Springer New York, 1983.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. 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, 2019. doi: 10.1016/j.jcp.2018.10.045.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabás Póczos. Equivariance through parameter-sharing. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 2892–2901, 2017.

Anastasiya Salova, Jeffrey Emenheiser, Adam Rupe, James P Crutchfield, and Raissa M D'Souza. Koopman operator and its approximations for systems with symmetries. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(9), 2019.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468, 2020.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Subhrajit Sinha, Sai Pushpak Nandanoori, and Enoch Yeung. Koopman operator methods for global phase space exploration of equivariant dynamical systems. *IFAC-PapersOnLine*, 53(2):1150–1155, 2020.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611, 2022. doi: 10.52202/068431-0117.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems*, 30, 2017.

Nathaniel Trask, Andy Huang, and Xiaozhe Hu. Enforcing exact physics in scientific machine learning: A data-driven exterior calculus on graphs. *Journal of Computational Physics*, 456: 110969, 2022.

Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2020.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 53681–53705, 21–27 Jul 2024.

Wei Xiong, Xiaomeng Huang, Ziyang Zhang, Ruixuan Deng, Pei Sun, and Yang Tian. Koopman neural operator as a mesh-free solver of non-linear partial differential equations. *Journal of Computational Physics*, 513:113194, 2024. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2024.113194.

Yuezhu Xu, S. Sivaranjani, and Vijay Gupta. Learning neural koopman operators with dissipativity guarantees. In *2025 IEEE 64th Conference on Decision and Control*, pp. 2064–2070, 2025. doi: 10.1109/CDC57313.2025.11312430.

Jingdong Zhang, Qunxi Zhu, and Wei Lin. Learning hamiltonian neural koopman operator and simultaneously sustaining and discovering conservation laws. *Phys. Rev. Res.*, 6:L012031, Feb 2024. doi: 10.1103/PhysRevResearch.6.L012031.

Zhen Zhang, Yeonjong Shin, and George Em Karniadakis. Gfinns: Generic formalism informed neural networks for deterministic and stochastic dynamical systems. *Philosophical Transactions of the Royal Society A*, 380(2229):20210207, 2022.

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

A Generator Reduction and Approximate Realizations

This appendix gives the reduction proof, its periodic-grid specialization, and the implementation restrictions used in Section 4. It also establishes the finite-time and approximation bounds relating latent evolution to decoded trajectories.

A.1 Geometric setup and local laws

Let $\mathfrak{M}$ be a nonempty family of finite, nonempty attributed relational meshes. Each mesh $\mathcal{K}$ has an entity set $\Sigma_{\mathcal{K}}$ with $N_{\mathcal{K}} = |\Sigma_{\mathcal{K}}|$, a fixed finite vocabulary of primitive relation types, such as adjacency or oriented incidence, together with specified geometric, material, or boundary attributes. Relations may be directed and weighted. Their undirected union defines a graph $I_{\mathcal{K}}$ and graph distance $d_{\mathcal{K}}$, with infinite distance between disconnected components. These primitive relations and attributes are fixed before the generator is chosen and describe the local environment available to the dynamical law. The complete relation basis used below is constructed from them.

The mesh family is closed under the transformations associated with **(P) relabeling consistency** and **(O) orientation covariance**. Relabeling preserves primitive relation types, directions, and specified attributes, while the orientation involution ω preserves entity labels and the undirected support graph, with $\omega^2\mathcal{K} = \mathcal{K}$, and changes only prescribed orientation-sensitive attributes or relation signs. Thus (P) changes only entity labels, whereas (O) changes the orientation convention without changing the underlying physical connectivity.

Fix integers $0 \le R \le r < \infty$ and write $B_q^{\mathcal{K}}(\sigma) = \{\tau : d_{\mathcal{K}}(\sigma, \tau) \le q\}$. The patch $\mathcal{P}_r(\mathcal{K}, \sigma)$ is the primitive attributed relation structure induced on $B_r^{\mathcal{K}}(\sigma)$, with σ as its root or center entity. Two patches are isomorphic when they represent the same local physical environment up to relabeling, with the center, relations, directions, and specified attributes preserved. This reflects the usual physical assumption that a local evolution law is determined by the state and local geometric, material, or boundary environment, rather than by arbitrary entity labels or distant graph-wide information. Absolute entity identifiers and graph-wide metadata are therefore excluded from the local law, while physically relevant quantities such as position, material coefficients, or boundary labels may be

included when specified in advance. The homogeneous-grid specialization below uses directional relations and fixed local attributes, not absolute coordinates.

We require the orientation action to be compatible with local patches:

$$\mathcal{P}_r(\omega\mathcal{K}, \sigma) = \omega\mathcal{P}_r(\mathcal{K}, \sigma). \tag{5}$$

Physically, reversing an orientation convention should commute with restricting attention to a local neighborhood. Likewise, a patch isomorphism remains an isomorphism after orientation reversal, so equivalent local environments remain equivalent under a change of orientation convention. This makes the orientation action well defined on local isomorphism classes, which whole-mesh relabeling alone would not ensure.

A patch isomorphism preserves the center and the local connectivity around it. Therefore, if an entity lies within the interaction radius R of the center in one patch, its corresponding entity lies within the same radius in the other patch. In other words, equivalent local environments contain the same pattern of possible direct interactions. This follows because, for $R \leq r$, any shortest path from the center to an entity within B_R is entirely contained in the larger patch B_r, so the patch preserves the relevant graph distances.

Let $V_Z = \mathbb{R}^{d_z}$, $d_z \geq 1$, denote the shared latent feature space attached to each entity, and define

$$Z_{\mathcal{K}} = \bigoplus_{\sigma \in \Sigma_{\mathcal{K}}} V_Z. \tag{6}$$

Ordinary patch isomorphisms relabel entity slots while acting identically on latent channels. Orientation acts on the common channel space through

$$S_Z = \mathrm{diag}(s_1, \ldots, s_{d_z}), \quad s_a \in \{+1, -1\}, \quad \Omega_{\mathcal{K}} = I_{N_{\mathcal{K}}} \otimes S_Z. \tag{7}$$

Thus each latent channel is assigned an orientation parity: even channels remain unchanged and odd channels change sign under orientation reversal. We restrict the present result to this shared grading and do not consider position-dependent changes of local frame or more general rotational mixing of channel components.

A.2 Assumptions on the generator family

Koopman realization. We consider a mesh-indexed family $A = (A_{\mathcal{K}})_{\mathcal{K} \in \mathfrak{M}}$. Adopting a finite-dimensional continuous-time Koopman realization means that each $A_{\mathcal{K}} : Z_{\mathcal{K}} \to Z_{\mathcal{K}}$ is an autonomous linear generator, with coefficients independent of latent state and time. The latent channel space V_Z is common across the mesh family, while parameter sharing will follow from the structural conditions below. We consider $\dot{z} = A_{\mathcal{K}} z$ without an additive forcing term.

(L) Finite interaction radius. Writing $A_{\mathcal{K},\sigma\leftarrow\tau} \in \mathrm{End}(V_Z)$ for the block mapping the latent feature at τ to the instantaneous update at σ, where $\mathrm{End}(V_Z)$ denotes linear maps from V_Z to itself,

$$A_{\mathcal{K},\sigma\leftarrow\tau} = 0 \quad \text{if} \quad d_{\mathcal{K}}(\sigma, \tau) > R. \tag{8}$$

Thus only entities within radius R can directly affect the instantaneous latent dynamics, although the interaction coefficient may depend on the larger local environment within radius r.

(P_{loc}) Local-law consistency. The relabeling principle (P) requires covariance under any isomorphism of the full mesh. Such an isomorphism may relabel only part of the mesh, provided that it extends consistently to the whole mesh structure. Here we impose a stronger local requirement: two isomorphic local environments must obey the same local law even when their surrounding meshes differ and no global mesh isomorphism exists. Formally, for every patch isomorphism $f : \mathcal{P}_r(\mathcal{K}, \sigma) \to \mathcal{P}_r(\mathcal{K}', \sigma')$ and every pair $z \in Z_{\mathcal{K}}$, $z' \in Z_{\mathcal{K}'}$ satisfying $z'_{f(\tau)} = z_\tau$ on $B_r^{\mathcal{K}}(\sigma)$,

$$(A_{\mathcal{K}'} z')_{\sigma'} = (A_{\mathcal{K}} z)_\sigma. \tag{9}$$

Values outside the patches are arbitrary. Thus r specifies the local structural context that may determine the law, while $R \leq r$ specifies which entities directly interact. Condition (P_{loc}) implies the original relabeling covariance (P), whereas (P) alone does not enforce consistency between matching patches embedded in different global meshes.

Original (P) alone is strictly weaker than (P_{loc}). For example, $A_{\mathcal{K}} = -N_{\mathcal{K}} I_{Z_{\mathcal{K}}}$, where $I_{Z_{\mathcal{K}}}$ is the identity on $Z_{\mathcal{K}}$, is local and relabeling equivariant, but assigns different responses to identical local patches embedded in meshes of different sizes.

(O) Orientation covariance.

$$A_{\omega\mathcal{K}} \Omega_{\mathcal{K}} = \Omega_{\mathcal{K}} A_{\mathcal{K}}, \tag{10}$$

where $\Omega_{\mathcal{K}} = I_{N_{\mathcal{K}}} \otimes S_Z$ is the latent orientation action defined above. Thus changing the orientation convention transforms the latent dynamics consistently without changing the represented physical evolution.

Conditions (L), (P_{loc}), and (O) characterize the admissible generator through a shared kernel and do not require finitely many local environments. A finite relation-basis representation additionally requires the set of relevant local interaction types to be finite. For $d_{\mathcal{K}}(\sigma, \tau) \le R$, define

$$\alpha(\mathcal{K}, \sigma, \tau) = [\mathcal{P}_r(\mathcal{K}, \sigma), \sigma, \tau], \tag{11}$$

where $[\cdot]$ denotes the patch-isomorphism class with output σ and input τ marked in order. Let $\mathcal{T}_{R,r}$ be the set of all such interaction types realized across $\mathfrak{M}$. The finite-basis part of Theorem 1 assumes $|\mathcal{T}_{R,r}| < \infty$. This is a condition on the target mesh family and its prescribed local attributes, not on the size of the training dataset. By equation 5, orientation induces an involution $\alpha \mapsto \omega\alpha$ on $\mathcal{T}_{R,r}$.

A.3 Proof of Theorem 1

Proof. **Shared-kernel characterization.** Fix $\mathcal{K} \in \mathfrak{M}$ and $z \in Z_{\mathcal{K}}$. For $\tau \in \Sigma_{\mathcal{K}}$ and $v \in V_Z$, let $e_{\mathcal{K},\tau}(v) \in Z_{\mathcal{K}}$ denote the latent field equal to v at τ and zero elsewhere. Since $A_{\mathcal{K}}$ is linear,

$$(A_{\mathcal{K}} z)_\sigma = \sum_{\tau \in \Sigma_{\mathcal{K}}} A_{\mathcal{K},\sigma\leftarrow\tau} z_\tau = \sum_{\tau \in B_R^{\mathcal{K}}(\sigma)} A_{\mathcal{K},\sigma\leftarrow\tau} z_\tau, \tag{12}$$

where the second equality follows from (L).

Now suppose $(\mathcal{K}, \sigma, \tau)$ and $(\mathcal{K}', \sigma', \tau')$ have the same marked local type. Then there is a patch isomorphism f with $f(\sigma) = \sigma'$ and $f(\tau) = \tau'$. Applying (P_{loc}) to the two fields $e_{\mathcal{K},\tau}(v)$ and $e_{\mathcal{K}',\tau'}(v)$ gives

$$A_{\mathcal{K},\sigma\leftarrow\tau} v = A_{\mathcal{K}',\sigma'\leftarrow\tau'} v \qquad \text{for all } v \in V_Z. \tag{13}$$

Hence, the two linear maps are equal. Therefore, the interaction block depends only on the marked local type $\alpha = \alpha(\mathcal{K}, \sigma, \tau)$, not on the particular mesh or entities realizing it. Denote this common block by $W(\alpha)$. Then

$$(A_{\mathcal{K}} z)_\sigma = \sum_{\tau \in B_R^{\mathcal{K}}(\sigma)} W(\alpha(\mathcal{K}, \sigma, \tau)) z_\tau. \tag{14}$$

Since every $\alpha \in \mathcal{T}_{R,r}$ is realized by at least one marked pair, this determines W uniquely.

Conversely, let $W : \mathcal{T}_{R,r} \to \mathrm{End}(V_Z)$ be arbitrary and define $A_{\mathcal{K}}$ by equation 14. The sum is restricted to $B_R^{\mathcal{K}}(\sigma)$, so the resulting family has interaction radius at most R. If $f : \mathcal{P}_r(\mathcal{K}, \sigma) \to \mathcal{P}_r(\mathcal{K}', \sigma')$ is a patch isomorphism, then f bijects the corresponding radius-R neighborhoods and preserves their marked interaction types. Thus, for latent fields satisfying $z'_{f(\tau)} = z_\tau$ on the patch,

$$\begin{aligned}(A_{\mathcal{K}'} z')_{\sigma'} &= \sum_{\tau \in B_R^{\mathcal{K}}(\sigma)} W(\alpha(\mathcal{K}', \sigma', f(\tau))) z'_{f(\tau)} \\ &= \sum_{\tau \in B_R^{\mathcal{K}}(\sigma)} W(\alpha(\mathcal{K}, \sigma, \tau)) z_\tau = (A_{\mathcal{K}} z)_\sigma.\end{aligned} \tag{15}$$

Hence (P_{loc}) also holds. This proves that finite interaction radius together with local-law consistency is equivalent to the shared-kernel form.

Orientation covariance. Let $\alpha = \alpha(\mathcal{K}, \sigma, \tau)$. The (σ, τ) block of (O) is

$$A_{\omega\mathcal{K},\sigma\leftarrow\tau} S_Z = S_Z A_{\mathcal{K},\sigma\leftarrow\tau}. \tag{16}$$

Under orientation reversal, the local type α becomes $\omega\alpha$. By the shared-kernel characterization, $A_{\mathcal{K},\sigma\leftarrow\tau} = W(\alpha)$ and $A_{\omega\mathcal{K},\sigma\leftarrow\tau} = W(\omega\alpha)$. Hence

$$W(\omega\alpha) S_Z = S_Z W(\alpha).$$

Since $S_Z^2 = I$, this is equivalent to

$$W(\omega\alpha) = S_Z W(\alpha) S_Z. \tag{17}$$

Conversely, this identity implies (O) for every block within radius R, while both sides are zero outside the interaction radius.

Finite graded relation basis. Now assume $\mathcal{T}_{R,r}$ is finite. For each realized type α, define the scalar indicator relation

$$(E_{\alpha,\mathcal{K}})_{\sigma\tau} = \begin{cases} 1, & d_{\mathcal{K}}(\sigma,\tau) \le R \text{ and } \alpha(\mathcal{K},\sigma,\tau) = \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

These relation families are local and locally consistent. They are also linearly independent over the mesh family: if $\sum_\alpha c_\alpha E_{\alpha,\mathcal{K}} = 0$ for every $\mathcal{K}$, evaluating at any realized pair of type β gives $c_\beta = 0$. Hence every generator has the unique expansion

$$A_{\mathcal{K}} = \sum_{\alpha \in \mathcal{T}_{R,r}} E_{\alpha,\mathcal{K}} \otimes W(\alpha). \tag{19}$$

The independence here is that of relation *families* over $\mathfrak{M}$, not necessarily their evaluations on a single mesh.

Orientation acts on the indicator relations by $E_{\alpha,\omega\mathcal{K}} = E_{\omega\alpha,\mathcal{K}}$. If a type is unchanged by orientation reversal, $\omega\alpha = \alpha$, we keep E_α as an even basis element. Otherwise, orientation exchanges a pair of types $\{\alpha, \beta\}$ with $\beta = \omega\alpha$. For each such pair, replace the two indicators by their symmetric and antisymmetric combinations,

$$B_+ = E_\alpha + E_\beta, \qquad B_- = E_\alpha - E_\beta, \tag{20}$$

which satisfy $B_{\pm,\omega\mathcal{K}} = \pm B_{\pm,\mathcal{K}}$. The corresponding channel matrices are

$$W_+ = \tfrac{1}{2}(W(\alpha) + W(\beta)), \qquad W_- = \tfrac{1}{2}(W(\alpha) - W(\beta)). \tag{21}$$

Using equation 17, $S_Z W_+ S_Z = W_+$ and $S_Z W_- S_Z = -W_-$, while a fixed type satisfies $S_Z W(\alpha) S_Z = W(\alpha)$. Hence the resulting relation basis $\{B_{\ell,\mathcal{K}}\}_{\ell \in \Lambda_{\mathrm{red}}}$ has a definite orientation parity $\chi_\ell \in \{+1, -1\}$,

$$B_{\ell,\omega\mathcal{K}} = \chi_\ell B_{\ell,\mathcal{K}}, \qquad S_Z W_\ell = \chi_\ell W_\ell S_Z. \tag{22}$$

Conversely, any coefficients satisfying this grading rule give

$$\begin{aligned} A_{\omega\mathcal{K}}(I \otimes S_Z) &= \sum_\ell \chi_\ell B_{\ell,\mathcal{K}} \otimes W_\ell S_Z \\ &= \sum_\ell B_{\ell,\mathcal{K}} \otimes S_Z W_\ell = (I \otimes S_Z) A_{\mathcal{K}}. \end{aligned} \tag{23}$$

Thus (O) holds, while locality and local-law consistency are preserved by the basis construction. Since this parity-adapted basis is obtained from the independent indicator basis by an invertible change of basis, its coefficient matrices are unique. This proves the finite-basis statement and completes the theorem. □

A.4 Consequences of the reduction

The reduction theorem has two immediate consequences for the learned model. First, the parity structure determines the number of independent generator coefficients. Second, the covariance imposed on the infinitesimal generator is inherited by its finite-time propagation.

Corollary 1 (Dimension of the reduced generator class)**.** *Let L_+ and L_- denote the numbers of even and odd relation-basis families in Theorem 1, and let d_+ and d_- denote the numbers of even and odd latent channels, respectively. Then the dimension of the admissible generator-family space is*

$$L_+(d_+^2 + d_-^2) + 2L_- d_+ d_-. \tag{24}$$

If orientation has f fixed local types and p pairs of types exchanged under reversal, the same dimension is

$$f(d_+^2 + d_-^2) + p(d_+ + d_-)^2. \tag{25}$$

Proof. For an even relation, $S_Z W = W S_Z$, so only the $(+,+)$ and $(-,-)$ channel blocks may be nonzero, giving $d_+^2 + d_-^2$ free coefficients. For an odd relation, $S_Z W = -W S_Z$, so only the $(+,-)$ and $(-,+)$ blocks may be nonzero, giving $2d_+d_-$ coefficients. Independence of the scalar relation families makes these dimensions additive. Since each exchanged pair produces one even and one odd basis element, $L_+ = f + p$ and $L_- = p$. □

The grading rule can also be read entrywise. Since $S_Z = \mathrm{diag}(s_1, \ldots, s_{d_z})$, the (a,b) entry of $S_Z W_\ell = \chi_\ell W_\ell S_Z$ gives

$$\big(s_a - \chi_\ell s_b\big)(W_\ell)_{ab} = 0. \tag{26}$$

Hence

$$(W_\ell)_{ab} = 0 \quad \text{unless} \quad s_a = \chi_\ell s_b. \tag{27}$$

Thus an even relation ($\chi_\ell = +1$) couples channels of the same parity, whereas an odd relation ($\chi_\ell = -1$) couples channels of opposite parity. This count concerns the independent coefficients of the reduced generator, not the number of trainable parameters in a particular implementation. For example, a factorized parametrization may use more trainable parameters than the generator has independent degrees of freedom.

Corollary 2 (Covariance of finite-time propagation)**.** *For any mesh isomorphism $f : \mathcal{K} \to \mathcal{K}'$ with permutation matrix P_f, every generator family in Theorem 1 satisfies*

$$A_{\mathcal{K}'}(P_f \otimes I_{V_Z}) = (P_f \otimes I_{V_Z}) A_{\mathcal{K}}. \tag{28}$$

The corresponding relabeling and orientation identities are preserved by powers, fixed-coefficient polynomials, and the exact exponential of the generator. If the encoder and decoder satisfy the analogous covariance identities, then so does the decoded predictor at every fixed time.

Proof. A whole-mesh isomorphism restricts to a patch isomorphism around every entity, so (P_{loc}) gives the generator identity above. Write either the relabeling or orientation identity as

$$A'Q = QA. \tag{29}$$

Then induction gives $(A')^k Q = QA^k$ for every $k \geq 0$, which proves the polynomial case. Absolute convergence of the finite-dimensional exponential series then gives

$$e^{tA'}Q = Qe^{tA}. \tag{30}$$

If the encoder and decoder obey the corresponding covariance identities, then inserting $e^{tA'}Q = Qe^{tA}$ between them shows that the full predictor $D_{\mathcal{K}} e^{tA_{\mathcal{K}}} \Psi_{\mathcal{K}}$ is covariant at every fixed time. This covariance statement does not by itself imply exact Koopman closure or an exact decoded semigroup. □

A.5 Finite types and selected relation dictionaries

The shared-kernel characterization does not require finitely many local types. Finiteness is needed only to represent the full kernel class by finitely many relation families. We first give a simple sufficient condition, then describe the finite-dimensional restriction used when local attributes are continuous.

Proposition 1 (Sufficient condition for finite local types)**.** *Fix finite alphabets for the primitive relation data, including any allowed relation weights, and for the local attributes. If $r < \infty$ and the undirected relation graphs have uniformly bounded degree, then $|\mathcal{T}_{R,r}| < \infty$.*

Proof. Let D be the uniform degree bound. Every radius-r patch contains at most $1 + D + \cdots + D^r$ entities. On a uniformly bounded number of entities, only finitely many attributed relational structures can be formed from finite alphabets, and each has finitely many choices of ordered marked entities. Passing to isomorphism classes preserves finiteness. □

Finiteness can fail when local attributes are continuous. For example, let $\mathcal{K}_a$ be a single isolated entity with attribute $a \in [0,1]$ and scalar latent state. Any rule

$$A_{\mathcal{K}_a} z = g(a) z \tag{31}$$

is radius-zero and locally consistent. Even restricting g to continuous functions gives an infinite-dimensional kernel class, as $1, a, a^2, \ldots$ are linearly independent. Hence no finite relation basis can represent all admissible kernels. Finite training data only make the observed type set finite, and bounded interaction radius alone does not imply finite type without a degree bound.

For continuous or otherwise rich attributes, we therefore restrict the dependence of $W(\alpha)$ to a selected finite-dimensional function class. The finite-type construction above is the special case in which $\mathscr{V}$ is the full space of scalar functions on the finite type set, with the indicator functions as a basis. The same construction extends to any selected finite-dimensional scalar function space, even when the type set itself is infinite.

Proposition 2 (Selected finite-dimensional kernel class)**.** *Let $\mathscr{V}$ be a fixed finite-dimensional space of real-valued scalar functions $b : \mathcal{T}_{R,r} \to \mathbb{R}$, invariant under orientation reversal, where $(\omega^* b)(\alpha) = b(\omega\alpha)$. Suppose $W(\omega\alpha) = S_Z W(\alpha) S_Z$ and each scalar entry $\alpha \mapsto [W(\alpha)]_{ab}$ belongs to $\mathscr{V}$. Then $\mathscr{V}$ admits a basis $\{b_\ell\}_{\ell \in \Lambda_{\mathrm{sel}}}$ with*

$$b_\ell(\omega\alpha) = \chi_\ell b_\ell(\alpha), \qquad \chi_\ell \in \{+1, -1\}. \tag{32}$$

Each b_ℓ induces a scalar weighted relation matrix $B_{\ell,\mathcal{K}} \in \mathbb{R}^{N_\mathcal{K} \times N_\mathcal{K}}$ by

$$(B_{\ell,\mathcal{K}})_{\sigma\tau} = \begin{cases} b_\ell(\alpha(\mathcal{K}, \sigma, \tau)), & d_\mathcal{K}(\sigma, \tau) \le R, \\ 0, & \textit{otherwise}. \end{cases} \tag{33}$$

The indicator relations $E_{\alpha,\mathcal{K}}$ used above are the special case obtained from indicator functions of individual types. Every generator in the selected class then has the unique form

$$A_\mathcal{K} = \sum_{\ell \in \Lambda_{\mathrm{sel}}} B_{\ell,\mathcal{K}} \otimes W_\ell, \qquad S_Z W_\ell = \chi_\ell W_\ell S_Z. \tag{34}$$

Proof. The finite-type proof applies verbatim once the indicator basis is replaced by a basis of $\mathscr{V}$, except that the basis must first be adapted to orientation. Since $\omega^2 = I$, the pullback $(\omega^* b)(\alpha) = b(\omega\alpha)$ is an involution on $\mathscr{V}$. Hence $\mathscr{V}$ admits an even–odd basis $\{b_\ell\}$ satisfying $b_\ell(\omega\alpha) = \chi_\ell b_\ell(\alpha)$.

Expanding

$$W(\alpha) = \sum_\ell b_\ell(\alpha) W_\ell \tag{35}$$

and using $W(\omega\alpha) = S_Z W(\alpha) S_Z$ gives

$$\sum_\ell b_\ell(\alpha) \big(\chi_\ell W_\ell - S_Z W_\ell S_Z\big) = 0. \tag{36}$$

Linear independence of the b_ℓ yields $S_Z W_\ell = \chi_\ell W_\ell S_Z$. Lifting each b_ℓ to the scalar relation matrix $B_{\ell,\mathcal{K}}$ then gives the stated expansion and uniqueness. □

This finite-dimensional dependence on local attributes is a modelling restriction, not a consequence of locality alone. In particular, a chosen primitive relation dictionary need not span all locally consistent kernels. For example, on an irregular unweighted graph, $(A_\mathcal{K} z)_i = -\deg(i)^2 z_i$ has interaction radius $R = 0$ and structural radius $r = 1$, but cannot generally be represented by a scalar combination of identity and adjacency. A practical implementation may therefore use either the complete type basis when available or a selected finite-dimensional subspace.

A.6 Periodic nearest-neighbor specialization

The general reduction recovers the standard shared directional stencil as a special case.

Corollary 3 (Directional periodic stencils)**.** *Consider periodic d-dimensional nearest-neighbor grids with periods $n_a > 2r + 1$, fixed local weights and attributes, and distinct relation labels for the $2d$ coordinate directions. Exclude absolute coordinates and total grid size from the local-law attributes. For $R = 1$, every locally consistent linear generator has the unique form*

$$A_\mathcal{K} = I \otimes W_0 + \sum_{a=1}^{d} \left(T_{a,+} \otimes W_{a,+} + T_{a,-} \otimes W_{a,-}\right), \tag{37}$$

where $e_a \in \mathbb{Z}^d$ is the unit lattice vector in direction a and $(T_{a,\pm} z)_\sigma = z_{\sigma \pm e_a}$.

Proof. The period condition ensures that every radius-r patch is isomorphic to the same labelled local lattice patch, with no periodic identification inside the patch. For $R = 1$, the only marked displacements are 0 and $\pm e_a$, which are distinct because direction labels are preserved. Their indicator relations are the identity and directional shifts, so Theorem 1 gives equation 37 and uniqueness. □

Thus, for $d = 1$ the reduction gives the familiar self, left, and right stencil. This is a fixed-local-attribute characterization, not a resolution-invariance result: changing grid spacing may change the local physical type.

A.7 Quasi-local finite-time action

For a fixed generator, each application of A can propagate information by at most R graph steps. Hence, for $R > 0$ and $m = \lceil d_{\mathcal{K}}(\sigma, \tau)/R \rceil$,

$$(A^k)_{\sigma \leftarrow \tau} = 0 \qquad \text{for } k < m. \tag{38}$$

Therefore, using the induced Euclidean norm,

$$\|(e^{tA})_{\sigma \leftarrow \tau}\| \le \sum_{k=m}^{\infty} \frac{t^k}{k!} \|(A^k)_{\sigma \leftarrow \tau}\| \le \sum_{k=m}^{\infty} \frac{(t\|A\|)^k}{k!}. \tag{39}$$

This proves the first inequality in equation 3. Setting the scalar $a = t\|A\| \ge 0$,

$$\sum_{k=m}^{\infty} \frac{a^k}{k!} = \frac{a^m}{m!} \sum_{j=0}^{\infty} \frac{a^j m!}{(m+j)!} \le \frac{a^m}{m!} \sum_{j=0}^{\infty} \frac{a^j}{j!} = e^a \frac{a^m}{m!},$$

which gives the second inequality. The case $a = 0$, $m = 0$ is understood by continuity. Disconnected components do not interact, and $R = 0$ gives block-diagonal propagation. Finally, the substepped Taylor action used in the main text is a polynomial in A of degree at most Ms, so its support radius is at most MsR. These are standard sparse-matrix-function decay arguments (Benzi & Razouk, 2007).

Uniform bound over a finite-type mesh family. The preceding estimate can be made uniform over the mesh family when $\mathcal{T}_{R,r}$ is finite and the kernel W is fixed. Define $q = \sup_{\mathcal{K},\sigma} |B_R^{\mathcal{K}}(\sigma)|$ and $w = \max_{\alpha \in \mathcal{T}_{R,r}} \|W(\alpha)\|_2$. Both are finite: finite local type uniformly bounds the relevant patch sizes, while W contains only finitely many coefficient matrices. With the block-row-sum norm, $\|A_{\mathcal{K}}\|_{\mathrm{br}} \le qw$, and hence, for $m = \lceil d_{\mathcal{K}}(\sigma, \tau)/R \rceil$,

$$\|(e^{tA_{\mathcal{K}}})_{\sigma \leftarrow \tau}\|_2 \le \sum_{k=m}^{\infty} \frac{(tqw)^k}{k!} \le e^{tqw} \frac{(tqw)^m}{m!}. \tag{40}$$

Thus the decay constants are independent of mesh size for the fixed mesh family and fixed local law. Distinct components remain uncoupled, and $R = 0$ gives block-diagonal propagation.

A.8 Selected relations and nonlinear local maps

For the nonlinear encoder and decoder, we choose finite dictionaries of local scalar relation matrices $\widehat{B}_{\ell,\mathcal{K}} \in \mathbb{R}^{N_{\mathcal{K}} \times N_{\mathcal{K}}}$ that satisfy the required relabeling and orientation covariance. The hats distinguish these selected implementation relations from a complete type basis. We use

$$\Psi_{\mathcal{K}}(x)_\sigma = \psi_\theta\big(x_\sigma, \{(\widehat{B}_{\ell,\mathcal{K}} x)_\sigma\}_{\ell \in \Lambda_E}\big), \quad \mathcal{D}_{\mathcal{K}}(z)_\sigma = d_\zeta\big(z_\sigma, \{(\widehat{B}_{\ell,\mathcal{K}} z)_\sigma\}_{\ell \in \Lambda_D}\big). \tag{41}$$

Here, the relation matrices act on entity indices and leave feature channels unchanged. The finite dictionaries, network widths, nonlinearities, and channel dimensions are modelling choices, not consequences of the linear-generator theorem. In general, linear aggregation may discard local information, although directional single-neighbor relations on a homogeneous grid retain the full one-hop stencil.

Let S_X and S_Z denote the orientation actions on physical and latent channels. If $\widehat{B}_{\ell,\omega\mathcal{K}} = \chi_\ell \widehat{B}_{\ell,\mathcal{K}}$, define $S_E = \mathrm{diag}(S_X, \{\chi_\ell S_X\}_{\ell \in \Lambda_E})$ and $S_D = \mathrm{diag}(S_Z, \{\chi_\ell S_Z\}_{\ell \in \Lambda_D})$. Orientation equivariance requires $\psi_\theta(S_E u) = S_Z \psi_\theta(u)$ and $d_\zeta(S_D v) = S_X d_\zeta(v)$. For any map f, this two-element covariance can be enforced by

$$f_{\mathrm{eq}}(u) = \tfrac{1}{2}\big(f(u) + S_{\mathrm{out}} f(S_{\mathrm{in}} u)\big), \tag{42}$$

when $S_{\mathrm{in}}^2 = S_{\mathrm{out}}^2 = I$. Shared relation features and shared nonlinear maps likewise give relabeling naturality at corresponding roots. This construction provides a compatible local and covariant parameterization, not a characterization of all admissible nonlinear maps or a guarantee of exact inversion.

A.9 Class preservation of the latent growth envelope

The latent growth envelope used in the main model restricts the reduced generator to a subfamily with controlled finite-time amplification. We show here that this restriction preserves the selected structural class. The reduction theorem determines the admissible structural form of the latent generator, but does not by itself control the growth of its finite-time evolution. Since excessive latent growth can amplify approximation and closure errors during rollout, we use a growth-controlled subfamily of the selected relation class. This construction is inspired by the storage principle (E), in the sense that it separates conservative-like mixing from dissipative contributions, but it does not identify the latent norm with the physical storage observable.

Let the selected dictionary contain $\widehat{B}_{0,\mathcal{K}} = I$ and be closed under transpose by fixed partner indices:

$$\widehat{B}_{\ell,\mathcal{K}}^{\top} = \widehat{B}_{\bar{\ell},\mathcal{K}}, \quad \bar{\bar{\ell}} = \ell, \quad \chi_{\bar{\ell}} = \chi_{\ell}. \tag{43}$$

Each dictionary family is assumed to satisfy the chosen locality, local-law condition, and orientation parity. In addition, we require transpose closure at the same structural radius. This does not impose symmetric interactions: it only requires that reversing the input-output direction of a selected relation yields another relation in the same dictionary, as for opposite directional shifts or symmetric adjacency. Transpose closure is natural for the stencil dictionaries used here, but is not automatic for a general output-root local relation class.

Choose matrices V_ℓ satisfying $S_Z V_\ell = \chi_\ell V_\ell S_Z$, an invertible L commuting with S_Z, and a matrix Q commuting with S_Z. For fixed $\gamma \in \mathbb{R}$, define

$$A_{\mathrm{w},\mathcal{K}} = \tfrac{1}{2}(G_{\mathcal{K}} - G_{\mathcal{K}}^{\top}) + I_{N_{\mathcal{K}}} \otimes (\gamma I - Q^{\top}Q), \qquad A_{\mathcal{K}} = T_{\mathcal{K}}^{-1} A_{\mathrm{w},\mathcal{K}} T_{\mathcal{K}},$$

where $G_{\mathcal{K}}$ and $T_{\mathcal{K}}$ are defined similarly in equation 4. In the transformed coordinates, the skew-symmetric term redistributes latent amplitude without changing the Euclidean quadratic norm, while $-Q^{\top}Q$ provides non-positive on-site dissipation and γ sets the permitted growth rate. Equivalently, $A_{\mathrm{w},\mathcal{K}}$ is the generator in the learned metric coordinates.

Proposition 3 (Class preservation and latent growth bound)**.** *The family equation 4 remains in the selected relation class, with*

$$W_\ell = \tfrac{1}{2} L^{-1}(V_\ell - V_{\bar{\ell}}^{\top})L + \mathbf{1}_{\{\ell=0\}} L^{-1}(\gamma I - Q^{\top}Q)L. \tag{44}$$

where $\mathbf{1}_{\{\ell=0\}}$ equals one for the identity relation $\widehat{B}_{0,\mathcal{K}} = I$ and zero otherwise. Thus the growth-control term contributes only to the on-site coefficient W_0. It is local, locally consistent, and orientation covariant. With $P_{\mathcal{K}} = I_{N_{\mathcal{K}}} \otimes L^{\top}L$, it also satisfies

$$A_{\mathcal{K}}^{\top} P_{\mathcal{K}} + P_{\mathcal{K}} A_{\mathcal{K}} \preceq 2\gamma P_{\mathcal{K}}, \qquad \|e^{tA_{\mathcal{K}}}\|_{P_{\mathcal{K}}} \le e^{\gamma t}, \quad t \ge 0, \tag{45}$$

where $\|z\|_{P_{\mathcal{K}}}^2 = z^{\top} P_{\mathcal{K}} z$ and the operator norm is induced by this metric.

Proof. By equation 43, $G_{\mathcal{K}}^{\top} = \sum_\ell \widehat{B}_{\ell,\mathcal{K}} \otimes V_{\bar{\ell}}^{\top}$. Using $T_{\mathcal{K}}^{-1}(B \otimes V)T_{\mathcal{K}} = B \otimes (L^{-1}VL)$ gives equation 44. Thus the factorization changes only the shared channel coefficients and introduces no new relation families or mesh-global dependence. Since S_Z is symmetric and involutive, $V_{\bar{\ell}}^{\top}$ has the same parity as V_ℓ. Commutation of L, L^{-1}, and $Q^{\top}Q$ with S_Z therefore preserves the grading rule for every W_ℓ, while the identity term is even. Locality and local-law consistency follow from the selected dictionary. Hence the family remains in the reduced structural class.

Now $P_{\mathcal{K}} = T_{\mathcal{K}}^{\top} T_{\mathcal{K}} \succ 0$, and

$$\begin{aligned} A_{\mathcal{K}}^{\top} P_{\mathcal{K}} + P_{\mathcal{K}} A_{\mathcal{K}} &= T_{\mathcal{K}}^{\top}(A_{\mathrm{w},\mathcal{K}}^{\top} + A_{\mathrm{w},\mathcal{K}})T_{\mathcal{K}} \\ &= 2\gamma P_{\mathcal{K}} - 2T_{\mathcal{K}}^{\top}(I \otimes Q^{\top}Q)T_{\mathcal{K}} \preceq 2\gamma P_{\mathcal{K}}. \end{aligned} \tag{46}$$

For $z(t) = e^{tA_{\mathcal{K}}} z_0$, this implies $\frac{d}{dt}\|z(t)\|_{P_{\mathcal{K}}}^2 \le 2\gamma \|z(t)\|_{P_{\mathcal{K}}}^2$, and hence $\|z(t)\|_{P_{\mathcal{K}}} \le e^{\gamma t}\|z_0\|_{P_{\mathcal{K}}}$. Taking the supremum over $z_0 \neq 0$ gives the operator bound. □

In the implementation, L is a parity-masked triangular factor with positive diagonal, Q is parity-masked, and $\gamma = \log(g_{\max})/\Delta_0$ for configured $g_{\max} > 0$ and $\Delta_0 > 0$. These learned or configured quantities are shared across the mesh family. The bound permits controlled growth when $\gamma > 0$ and guarantees non-expansion in the learned metric when $\gamma \leq 0$. Importantly, this learned quadratic metric is not the physical storage $H_{\mathcal{K}}$ in principle (E), so the bound does not by itself imply physical energy or storage dissipation.

This factorization is a sufficient growth-controlled subfamily, not a characterization of all dissipative local generators. In the particular envelope parameterization used here, the symmetric part in transformed coordinates contains only on-site channel dissipation, while spatial couplings enter through the skew-symmetric part. Hence, for example, a scalar symmetric nearest-neighbor Laplacian, although local and dissipative, is not generally represented by this parameterization. This restriction is specific to the envelope parameterization used here and can be relaxed to include local positive-semidefinite spatial dissipation, such as graph-Laplacian terms, provided the enlarged relation class preserves the required locality and covariance. The bound is an exact algebraic property of the parameterized generator, not a finite-precision certification.

A.10 Numerical propagation and approximate observable closure

For order M and s substeps, define

$$\widetilde{\mathcal{E}}_{\Delta}^{M,s}(A) = \left[\sum_{k=0}^{M} \frac{(\Delta A/s)^k}{k!}\right]^s. \tag{47}$$

This is a polynomial of degree at most Ms, hence has support radius at most MsR, and inherits the relabeling and orientation covariance of A. Unlike the exact exponential, however, it need not satisfy the semigroup identity or the growth bound in equation 45. The guaranteed estimate is

$$\|\widetilde{\mathcal{E}}_{\Delta}^{M,s}(A)\|_P \leq e^{\gamma\Delta} + \|\widetilde{\mathcal{E}}_{\Delta}^{M,s}(A) - e^{\Delta A}\|_P. \tag{48}$$

Thus numerical approximation properties remain separate from the structural guarantees.

The learned predictor is $\widehat{x}(t) = \mathcal{D}(e^{tA}\Psi(x_0))$, or its specified numerical approximation. The reduction theorem constrains the admissible generator class but does not guarantee exact observable closure $D\Psi_x[F(x)] = A\Psi(x)$, exact reconstruction $\mathcal{D}\Psi(x) = x$, injectivity, or an exact finite-dimensional invariant subspace. These properties are instead encouraged empirically through reconstruction, data-closure, rollout, and cycle-consistency losses. In our rollout, the state is encoded once, propagated in latent space, and then decoded. Repeated encoding and decoding would in general define a different dynamical model.

A.11 Closure defects and decoded prediction error

Fix a mesh and suppress its subscript. The following identities formalize the separation among observable closure, latent amplification, numerical propagation error, and reconstruction used in the experiments.

Assume that F and Ψ are continuously differentiable near the trajectory tube, and define the generator defect

$$r_A(x) = A\Psi(x) - D\Psi_x[F(x)].$$

Differentiating $s \mapsto e^{(t-s)A}\Psi(\phi_s(x))$ and integrating gives the finite-dimensional variation-of-constants identity (Pazy, 1983)

$$e^{tA}\Psi(x) - \Psi(\phi_t(x)) = \int_0^t e^{(t-s)A} r_A(\phi_s(x))\, ds. \tag{49}$$

Let $\epsilon_{\mathrm{rec}}(y) = \|\mathcal{D}\Psi(y) - y\|$, and assume that $\mathcal{D}$ is $L_{\mathcal{D}}$-Lipschitz on the relevant latent states. Then

$$\left\|\mathcal{D}e^{tA}\Psi(x) - \phi_t(x)\right\| \leq L_{\mathcal{D}} \int_0^t \|e^{(t-s)A}\| \|r_A(\phi_s(x))\|\, ds + \epsilon_{\mathrm{rec}}(\phi_t(x)). \tag{50}$$

Replacing the exact exponential by a numerical action $\widetilde{K}_t$ adds its approximation error:

$$\begin{aligned}\left\|\mathcal{D}\widetilde{K}_t\Psi(x) - \phi_t(x)\right\| \le L_{\mathcal{D}}\|(\widetilde{K}_t - e^{tA})\Psi(x)\| \\ + L_{\mathcal{D}}\int_0^t \|e^{(t-s)A}\|\|r_A(\phi_s(x))\|\,ds + \epsilon_{\mathrm{rec}}(\phi_t(x)).\end{aligned} \tag{51}$$

For a sampled trajectory $(x_0, \dots, x_h)$ and a fixed linear latent step K_Δ, define $\delta_j = K_\Delta\Psi(x_j) - \Psi(x_{j+1})$. Repeated substitution gives

$$K_\Delta^h\Psi(x_0) - \Psi(x_h) = \sum_{j=0}^{h-1} K_\Delta^{h-1-j}\delta_j, \tag{52}$$

and hence

$$\left\|\mathcal{D}K_\Delta^h\Psi(x_0) - x_h\right\| \le L_{\mathcal{D}}\sum_{j=0}^{h-1}\|K_\Delta^{h-1-j}\|\,\|\delta_j\| + \epsilon_{\mathrm{rec}}(x_h). \tag{53}$$

Thus long-horizon prediction depends not only on the closure defects, but also on their propagation and on reconstruction and numerical errors. These identities do not assume exact finite-dimensional closure or statistical identifiability.

A.12 Latent coordinates and physical storage

The latent growth envelope is expressed in a learned latent metric, whereas principle (E) concerns a distinguished physical storage observable. We first clarify which quantities are invariant under changes of latent coordinates.

To distinguish coordinate-invariant properties of the realization from metric-dependent latent quantities, consider an invertible latent coordinate change $z' = Tz$ and the corresponding reparameterization

$$\Psi' = T\Psi, \qquad A' = TAT^{-1}, \qquad \mathcal{D}' = \mathcal{D}\circ T^{-1}.$$

By construction, this change of latent coordinates leaves the decoded predictor unchanged, since $e^{tA'} = Te^{tA}T^{-1}$ and hence

$$\mathcal{D}'e^{tA'}\Psi' = \mathcal{D}e^{tA}\Psi.$$

More importantly, the generator defect transforms as $r'_{A'} = Tr_A$. Thus exact closure, $r_A = 0$, is coordinate invariant, whereas its Euclidean norm and Euclidean contraction properties generally are not. For a channel-local transformation $T_{\mathcal{K}} = I_{N_{\mathcal{K}}} \otimes T_Z$, the reduced relation structure is also preserved: only the channel matrices transform as $W_\ell \mapsto T_Z W_\ell T_Z^{-1}$. Orientation grading is preserved when $T_Z S_Z = S_Z T_Z$.

For a quadratic latent quantity $V(z) = \frac{1}{2}z^\top Pz$, non-increase under $\dot{z} = Az$ is characterized by $A^\top P + PA \preceq 0$ (Khalil & Grizzle, 2002). Under $z' = Tz$, the same scalar quantity is represented by $P' = T^{-\top}PT^{-1}$. Thus choosing the Euclidean norm corresponds to selecting one particular latent metric.

By contrast, let $\mathcal{H}_{\mathcal{K}}$ be a differentiable physical storage observable and define $h_{\mathcal{K}}(z) = \mathcal{H}_{\mathcal{K}}(\mathcal{D}_{\mathcal{K}}(z))$. Along the latent dynamics,

$$\frac{d}{dt}h_{\mathcal{K}}(z(t)) = D\mathcal{H}_{\mathcal{K}\mathcal{D}_{\mathcal{K}}(z)}\left[D\mathcal{D}_{\mathcal{K}z}[A_{\mathcal{K}}z]\right]. \tag{54}$$

Physical conservation or dissipation requires this quantity to vanish or be non-positive on the relevant decoded trajectories. Neither the relation reduction nor the latent metric bound implies this condition unless the physical storage is explicitly tied to that metric. The latent envelope is therefore inspired by principle (E), but does not itself enforce physical conservation, dissipation, or positivity. If $\mathcal{H}_{\mathcal{K}}$ is Lipschitz, accurate state prediction also controls storage-value error, but this is an accuracy consequence rather than a structural guarantee.

Table 6: Controlled structural validation over five seeds. State errors are rollout relative L^2. Covariance defects are computed at the prediction level. Lower is better, and boldface marks the lowest mean in each column.

(a) Zero-shot orientation reversal

Method	Canonical	Reversed	Orientation defect
Koopman-MeshFT	$\mathbf{0.00257} \pm 0.00009$	$\mathbf{0.00257} \pm 0.00009$	$\mathbf{0}$
KNO	0.02225 ± 0.00160	0.77597 ± 0.10056	1.0489 ± 0.1131
cKAE	0.57526 ± 0.00857	1.65813 ± 0.13096	1.2106 ± 0.0557
FNO	0.00790 ± 0.00079	1.95531 ± 1.06228	0.9231 ± 0.0585

(b) Geometrically irregular mesh transfer

Method	Held-out meshes	Strong-jitter OOD	Relabeling defect
Koopman-MeshFT	0.00458 ± 0.00040	0.01965 ± 0.00275	$(\mathbf{2.38} \pm \mathbf{0.08}) \times \mathbf{10^{-8}}$
KNO	0.00963 ± 0.00091	0.02468 ± 0.00272	0.1368 ± 0.0041
cKAE	0.64235 ± 0.08509	0.63529 ± 0.08088	0.9236 ± 0.0409
FNO	$\mathbf{0.00389} \pm 0.00076$	$\mathbf{0.01762} \pm 0.00424$	0.0486 ± 0.0116

Table 7: Reconstruction and latent-closure MSE with the re-evaluated CNS1D Full reference. Values are mean $\pm$ sample standard deviation over five seeds, with scale factors shown in the headings. Boldface marks the lowest unrounded mean within each benchmark and metric.

Variant	Reconstruction ($\times 10^{-4}$)	One-step closure ($\times 10^{-4}$)	Multi-step closure ($\times 10^{-3}$)
Burgers			
Koopman-MeshFT	0.663 ± 0.255	$\mathbf{1.53} \pm 0.11$	$\mathbf{6.43} \pm 0.989$
Global low-rank + envelope	2.25 ± 0.235	5.55 ± 0.543	27.8 ± 4.23
Without data-closure objective	$\mathbf{0.473} \pm 0.100$	6.55 ± 1.37	9.80 ± 1.70
Without manifold-cycle objective	0.688 ± 0.180	1.90 ± 0.0733	9.35 ± 2.03
CNS1D			
Koopman-MeshFT	9.17 ± 1.21	$\mathbf{4.04} \pm 0.353$	$\mathbf{7.97} \pm 1.01$
Global low-rank + envelope	14.3 ± 1.92	7.04 ± 0.571	22.7 ± 1.64
Without data-closure objective	$\mathbf{8.70} \pm 1.08$	13.8 ± 0.493	21.6 ± 4.34
Without manifold-cycle objective	8.77 ± 1.07	5.69 ± 0.416	12.9 ± 2.51

B Additional Experimental Results

B.1 Controlled two-dimensional structural validation

We additionally consider two controlled synthetic benchmarks to test the structural properties targeted by the reduction (Table 6). In the orientation benchmark, models are trained only in the canonical orientation and evaluated on the unseen reversed convention. In the irregular-mesh benchmark, models are trained on mildly perturbed periodic meshes and evaluated on held-out meshes and a stronger-jitter OOD range. The latter changes local geometry while keeping the torus topology and entity count fixed, so it is not a test of arbitrary unstructured-mesh or resolution generalization.

As shown in Table 6 (a), under orientation reversal, Koopman-MeshFT preserves its rollout error and has zero measured covariance defect. Its energy-curve relative MAE is likewise unchanged at 6.45×10^{-4}, with no positive energy increments. On the irregular benchmark, Table 6 (b) shows that its strong-jitter rollout error remains close to FNO and below KNO and cKAE, while its relabeling defect remains near numerical precision. These tests provide direct empirical checks of orientation covariance and relabeling naturality rather than additional claims of exact physical conservation or arbitrary mesh generalization.

Table 8: Paired rollout differences (ablated minus Koopman-MeshFT). CNS1D pairs archived ablation outputs with the re-evaluated Full reference. Values are mean ± sample standard deviation over five seeds. Also, counts indicate higher ablated error.

	Burgers		CNS1D	
Variant	Δ rollout error	Worse seeds	Δ rollout error	Worse seeds
Global low-rank + envelope	$+0.20529 \pm 0.02391$	$5/5$	$+0.27794 \pm 0.07714$	$5/5$
Without data-closure objective	-0.01145 ± 0.01432	$1/5$	-0.00098 ± 0.01994	$3/5$
Without manifold-cycle objective	$+0.00666 \pm 0.01084$	$3/5$	$+0.01912 \pm 0.02167$	$3/5$

Table 9: Matched CNS2D comparison of physical-coordinate rollout accuracy and mass/total-energy tracking on the test split over five paired seeds (mean ± sample standard deviation). Parameter counts are raw trainable scalars, shown as generator/total. Boldface marks the lower mean or parameter count.

Metric	Local	Global
Parameters (generator/total)	$\mathbf{456}/\mathbf{44{,}500}$	$262{,}280/306{,}324$
Rollout relative L^2	$\mathbf{0.2141} \pm 0.1109$	0.4477 ± 0.0613
Mass tracking	$\mathbf{0.0821} \pm 0.0624$	0.1590 ± 0.0149
Total-energy tracking	$\mathbf{0.1330} \pm 0.0671$	0.1938 ± 0.0481

B.2 Additional ablation statistics

This subsection supplements the generator and consistency-objective ablations in Section 5.3 with reconstruction, latent-closure statistics, and paired rollout differences.

These comparisons use five seeds. The Koopman-MeshFT reference has rollout relative L^2 error 0.1556 ± 0.0089 on Burgers and 0.2880 ± 0.0147 on CNS1D. Table 7 reports reconstruction and latent-closure statistics, while Table 8 reports paired rollout differences relative to Koopman-MeshFT. The encoder-feature comparison in Appendix B.4 uses a separate three-seed evaluation.

The global low-rank alternative has higher rollout error on every paired seed and larger closure errors on both tasks. This comparison favors the local shared construction, but jointly changes spatial mixing, rank, and parameter sharing rather than isolating locality alone. Removing either consistency objective increases the reported mean latent-closure errors. Its effect on rollout accuracy is smaller and task dependent: removing data closure improves the mean Burgers rollout error, while the manifold-cycle comparison has mixed signs across seeds.

B.3 Matched local/global generator comparison

We complement the low-rank comparison with separately retrained local/global models on a fixed periodic grid. Both retain translation sharing, the same codec architecture, objective, envelope construction, training and validation rules, numerical action, and all-mode FFT backend. In whitened coordinates, the spatial skew part uses self and nearest-neighbor shifts (local) or all periodic shifts (global). The on-site damping construction is unchanged. Neither low-rank factorization nor Fourier-mode truncation is imposed. Identical per-seed initial codecs and generators are trained independently end-to-end, with extra global coefficients initialized to trainable zeros. These separately retrained models use matrix exponentials for training and evaluation and report 19-transition physical-coordinate errors, unlike the training-standardized errors in Table 1.

The local generator achieves lower rollout relative L^2 in all five seed pairs and lower mean mass and total-energy tracking errors, while using substantially fewer trainable parameters (Table 9). These results support locality as a useful inductive bias for parameter-efficient prediction and mass/total-energy tracking under this matched protocol.

Table 10: Encoder-feature comparison over three seeds. Values are rollout relative L^2 means $\pm$ sample standard deviations. Boldface marks the lower mean within each benchmark.

Encoder	Burgers	CNS1D
Relation-feature encoder	0.1580 ± 0.0103	$\mathbf{0.2831} \pm 0.0086$
State-only encoder	$\mathbf{0.1488} \pm 0.0026$	0.3084 ± 0.0160

Table 11: Supplementary physical diagnostics for the one-dimensional benchmarks over five seeds (mean $\pm$ sample standard deviation). Boldface marks the lowest mean among the displayed *Koopman-based* models. FNO is a nonlinear reference.

Method	Advection phase error	Burgers energy increase rate	CNS1D mass drift
Koopman-MeshFT	$\mathbf{0.0231} \pm 0.0135$	0.1414 ± 0.0427	$\mathbf{0.0196} \pm 0.0031$
KNO	0.0803 ± 0.1072	$\mathbf{0.0520} \pm 0.0054$	0.1975 ± 0.1471
cKAE	1.3473 ± 0.1552	0.0974 ± 0.0618	0.2494 ± 0.0557
FNO	0.0948 ± 0.0936	0.3454 ± 0.1513	0.0423 ± 0.0119

B.4 Neighborhood features in the observable map

A state-only shared encoder remains local and relabeling equivariant, but removes neighborhood-derived features from the observable map. It gives slightly lower mean rollout error on Burgers and higher error on CNS1D (Table 10), indicating that the benefit of relation features is task dependent. This comparison uses a separate three-seed evaluation, with unchanged parameter counts.

B.5 Additional physical diagnostics

Diagnostics use denormalized states and spatial means $\langle\cdot\rangle$ (volume-weighted on lines). Trajectory scores are averaged before computing seed means and sample standard deviations.

Advection and Burgers. For Advection, we measure relative drift of $\langle u^2\rangle$ from the predicted initial value and relative phase-speed error estimated by circular cross-correlation. For Burgers, we evaluate normalized energy-balance residuals, errors in successive energy changes, and the frequency of energy increases, using $E = \langle u^2\rangle/2$.

Compressible systems. We evaluate mass, momentum, and total energy through $f = (\rho, \rho\boldsymbol{v}, \mathcal{E})$ and $\mathcal{Q} = \langle f\rangle$, where $\mathcal{E} = p/(\gamma_{\text{gas}} - 1) + \rho|\boldsymbol{v}|^2/2$ and $\gamma_{\text{gas}} = 1.4$. CNS1D measures drift from each predicted initial quantity, normalized by $\langle|f_{0,c}|\rangle$, where $f_{0,c}$ denotes the reference initial mass, momentum, or total-energy density ($c = 1, 2, 3$, respectively). CNS2D uses training-derived scales and distinguishes drift mismatch from tracking error:

$$\begin{aligned} d_j^{\mathcal{Q}} &= (\widehat{\mathcal{Q}}_j - \widehat{\mathcal{Q}}_0) - (\mathcal{Q}_j - \mathcal{Q}_0), \\ e_j^{\mathcal{Q}} &= \widehat{\mathcal{Q}}_j - \mathcal{Q}_j = d_j^{\mathcal{Q}} + (\widehat{\mathcal{Q}}_0 - \mathcal{Q}_0). \end{aligned} \tag{55}$$

Component metrics average absolute normalized residuals over future times. CNS1D composite drift and CNS2D quantity tracking use the root-mean-square across three and four components, respectively, before temporal averaging. Tracking therefore retains initial reconstruction offsets. Table 9 uses the same component tracking metrics. Detailed normalizations and numerical tolerances are specified in the supplementary evaluation code.

C Experimental and Numerical Details

C.1 Model construction and training specification

Data and model configurations. The public datasets use 128-point periodic lines for Advection, Burgers, and CNS1D, with train, validation, and test splits of 128, 32, and 32 trajectories and 20 states

Table 12: CNS2D mass and total-energy diagnostics over five seeds (mean ± sample standard deviation). Boldface marks the lowest mean among the displayed *Koopman-based* models. FNO is a nonlinear reference.

(a) Drift mismatch

Method	Mass	Total energy
Koopman-MeshFT	$\mathbf{0.0274} \pm 0.0159$	$\mathbf{0.0539} \pm 0.0370$
KNO	0.1564 ± 0.0043	0.1865 ± 0.0203
cKAE	0.0562 ± 0.0445	0.0939 ± 0.0842
FNO	0.0146 ± 0.0038	0.0234 ± 0.0056

(b) Quantity tracking

Method	Mass	Total energy
Koopman-MeshFT	$\mathbf{0.0307} \pm 0.0157$	$\mathbf{0.0657} \pm 0.0380$
KNO	0.1718 ± 0.0087	0.2153 ± 0.0355
cKAE	0.2486 ± 0.0203	0.4122 ± 0.1738
FNO	0.0146 ± 0.0038	0.0234 ± 0.0056

Table 13: Benchmark-specific Koopman-MeshFT configurations. The envelope gain $g_{\max}$ specifies the permitted gain over the nominal model step $\Delta_0 = 1$.

Benchmark	Mesh and state channels	Selected relation dictionary	Batch	$g_{\max}$
Advection	Periodic line, 1 channel	Self, left, right	16	1.00
Burgers	Periodic line, 1 channel	Self, left, right	16	1.00
CNS1D	Periodic line, 3 channels	Self, left, right	8	1.05
CNS2D	Periodic 64×64 grid, 4 channels	Self, north, south, west, east	4	1.05

per trajectory. The CNS2D manifest records 128, 32, and 32 trajectories with 21 states each on a periodic 64×64 grid. Advection uses $\beta = 0.4$, Burgers uses viscosity $\nu = 0.01$, and the compressible systems use $\eta = \zeta = 0.01$, with Mach number 0.1 for CNS2D. Training means μ_c and population standard deviations σ_c over trajectory, time, and entity axes define channelwise standardized inputs. σ_c is floored at 10^{-6}. For each trajectory, stack the 19 predicted future states in $\widehat{X}_b$ and targets in X_b. The main prediction error is $\mathrm{mean}_b \, \|(\widehat{X}_b - X_b)/\sigma\|_F / \max(\|(X_b - \mu)/\sigma\|_F, 10^{-12})$. Physical-coordinate errors use $\mu = 0$, $\sigma = 1$ in this formula. Physical diagnostics use denormalized states.

All four configurations use eight latent channels and separate shared local encoder and decoder MLPs with two hidden layers of width 128 and SiLU activations. The encoder, decoder, and generator use the relations in Table 13. The public-benchmark configurations use the trivial even grading for all physical and latent channels. The main model uses a local continuous generator with the latent growth envelope. Its metric floor and channel-damping initialization are both 10^{-3}, and its raw relation-channel matrices use initialization scale 0.02. The periodic-line and grid dictionaries are shared across mesh entities. The two-dimensional orientation experiment uses a separate nontrivial grading and is reported with the synthetic structural validation.

Optimization and numerical propagation. Training uses windows of five transitions, or six consecutive states, with stride one. Validation uses the same window length with stride five. We train for at most 100 epochs using AdamW with learning rate 10^{-3}, weight decay 10^{-6}, gradient-norm clipping at 1, and early stopping after 25 validation checks without improvement. The checkpoint with the lowest total validation objective is used for test evaluation. Results are aggregated over model seeds. For the main continuous models, training and CNS1D rollout evaluation use $(M, s) = (6, 2)$. CNS2D evaluation uses $(8, 4)$, with an additional $(12, 4)$ audit for the plain model. For the five frozen CNS2D envelope checkpoints, we compare the reported action with a complex128 all-mode matrix-exponential reference over 19 test transitions, without retraining or changing the reported solver. Both

yield standardized rollout error 0.5171 ± 0.0113 and unchanged mass/total-energy tracking at the reported precision. For standardized decoded trajectories $X = (\hat{x}_0, \ldots, \hat{x}_{19})$, including the shared initial reconstruction, the relative discrepancy $\|X_{\mathrm{exp}} - X_{\mathrm{native}}\|_F / \|X_{\mathrm{native}}\|_F$ is $(5.88 \pm 1.46) \times 10^{-6}$. Values are trajectory averages followed by seed means and sample standard deviations. The computed one-step P-norm operator difference is $(5.74 \pm 0.51) \times 10^{-4}$ for $(8, 4)$ and $(2.28 \pm 0.33) \times 10^{-7}$ for $(12, 4)$. Also, the 19-step P-norm gains are 2.234 ± 0.200 for both $(8, 4)$ and the reference (mean $\pm$ sample standard deviation across seeds). Using $(6, 2)$ over 19 transitions gives standardized rollout error 29.21 ± 25.84.

For the main model, let $\bar{z}_j = \Psi(x_j)$ denote the encoded teacher state and let $z_0 = \bar{z}_0$, $z_{j+1} = \widetilde{K}_\Delta z_j$ denote the autonomous latent rollout, where $\widetilde{K}_\Delta$ is the implemented polynomial action. The optimized objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{closure}} + \mathcal{L}_{\mathrm{rollout}} + 0.1\mathcal{L}_{\mathrm{cyc}} + 10^{-3}\mathcal{L}_{\mathrm{nc}} + 10^{-2}\mathcal{L}_{\mathrm{sg}}. \tag{56}$$

All terms use mean squared errors over their applicable sample, time, mesh-entity, and channel indices, except the non-collapse penalty. Specifically, $\mathcal{L}_{\mathrm{rec}} = \mathrm{MSE}(\mathcal{D}(z_0), x_0)$, $\mathcal{L}_{\mathrm{closure}} = \mathrm{MSE}(\widetilde{K}_\Delta \bar{z}_j, \bar{z}_{j+1})$ over all adjacent teacher-forced pairs, $\mathcal{L}_{\mathrm{rollout}} = \mathrm{MSE}(\mathcal{D}(z_j), x_j)$ for $j \geq 1$, and $\mathcal{L}_{\mathrm{cyc}} = \mathrm{MSE}(\Psi(\mathcal{D}(z_j)), z_j)$ for $j = 0, \ldots, h$, including the initial latent state. The non-collapse term is $\mathcal{L}_{\mathrm{nc}} = d_z^{-1} \sum_a [0.05 - \mathrm{Var}(z_{0,a})]_+$, where the variance is taken over minibatch samples and mesh entities. Finally, $\mathcal{L}_{\mathrm{sg}} = \mathrm{MSE}(\widetilde{K}_{2\Delta} z_0, \widetilde{K}_\Delta \widetilde{K}_\Delta z_0)$ penalizes numerical semigroup inconsistency. All active terms are evaluated every minibatch and optimized jointly.

At inference, the initial state is encoded once, propagated entirely in latent space, and decoded at the requested times. No decode–reencode operation is applied between latent propagation steps.

Reference-model configurations. The main comparison reports KNO with retained mode count $m = 12$. cKAE uses latent width 8 on the one-dimensional benchmarks and width 1 on CNS2D. The supplied CNS2D runner evaluates KNO modes $4, 8, 12$ at latent width 8, cKAE widths 1 and 2, and an FNO with width 32, 12 retained modes per spatial axis, and four spectral layers. KNO mode selection uses the lowest mean validation objective and does not use test metrics. The external Koopman baseline code uses two hidden layers of width 128 for its encoder and decoder, Adam with learning rate 10^{-3} and weight decay 10^{-6}, gradient clipping at 1, and objective weights 1 for reconstruction, latent linearity, and rollout prediction, together with a 10^{-3} non-collapse penalty with floor 0.05. cKAE additionally uses backward-prediction and forward-backward consistency weights 1 and 0.1.

D Sparse-Observation Estimation and Forecasting Details

This appendix provides the implementation details for the sparse-observation experiment in Section 5.4. It specifies the observation protocol, the Koopman-MeshFT and inverse-FNO state estimators, the subsequent forecasting procedures, and the evaluation and timing conventions.

Observation protocol. A four-frame sequence of partial observations at $t_0, \ldots, t_3$ is used to estimate the full state at t_3 and forecast $t_4, \ldots, t_{19}$. Burgers uses 16 sensors on 128 periodic sites. CNS2D observes all four state channels at 128 sites of a periodic 64×64 grid and averages uniform, random, and coarse-fine sensor placements. Sensor locations remain fixed during the observation window. Independent Gaussian noise with standard deviation 0.01 is added in training-standardized coordinates.

All dynamical models are trained on full-field trajectories and held fixed during this experiment. Readouts and priors are fitted using training data only, and hyperparameters are selected using validation data. Each method receives the same observations for each trajectory and sensor realization. No observations are provided after t_3, and no true-state reset or positivity clipping is applied. Inverse computations are performed in training-standardized coordinates, whereas reported state errors and physical diagnostics use denormalized physical states.

Linear Koopman estimator. Let S denote the sensor-selection operator and $\widetilde{K} = \widetilde{K}_\Delta$ the frozen numerical latent step. An auxiliary affine readout $x \approx Cz + b$ is fitted from training states and their

encodings. Let μ_z and $\Sigma_z \succ 0$ denote the mean and regularized covariance of encoded training initial states. Stacking the four observation frames gives

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}, \qquad H_S = \begin{bmatrix} SC \\ SC\widetilde{K} \\ SC\widetilde{K}^2 \\ SC\widetilde{K}^3 \end{bmatrix}, \qquad \mathbf{b}_S = \begin{bmatrix} Sb \\ Sb \\ Sb \\ Sb \end{bmatrix}, \tag{57}$$

so that $\mathbf{y} \approx H_S z_0 + \mathbf{b}_S$.

For each sensor configuration, we use $R_S = \lambda_S I$, with $\lambda_S = \sigma_{\text{obs}}^2 + \lambda_{\text{floor}}$, where $\sigma_{\text{obs}} = 0.01$ is the observation-noise standard deviation in standardized coordinates and $\lambda_{\text{floor}} \in \{10^{-6}, 10^{-4}, 10^{-2}, 10^{-1}, 1\}$ is selected using validation forecasts through the auxiliary affine readout and then held fixed for the native forecast-restart route. The resulting estimator is

$$\begin{aligned} G_S &= \Sigma_z H_S^\top \left(H_S \Sigma_z H_S^\top + R_S\right)^{-1}, \\ \widehat{z}_0 &= \mu_z + G_S \left(\mathbf{y} - \mathbf{b}_S - H_S \mu_z\right). \end{aligned} \tag{58}$$

Equivalently, $\widehat{z}_0$ minimizes $\|H_S z + \mathbf{b}_S - \mathbf{y}\|^2_{R_S^{-1}} + \|z - \mu_z\|^2_{\Sigma_z^{-1}}$. The diagonal observation regularizer stabilizes the observation-space solve, while the latent-prior penalty constrains directions not identified by the measurements. Neither is interpreted as a calibrated uncertainty model. Propagating the inferred latent state to t_3 gives the affine full-state estimate

$$\widehat{x}_3 = C\widetilde{K}^3 \widehat{z}_0 + b = M_S \mathbf{y} + a_S, \tag{59}$$

where $M_S = C\widetilde{K}^3 G_S$ and $a_S = C\widetilde{K}^3[\mu_z - G_S(\mathbf{b}_S + H_S \mu_z)] + b$. For each sensor configuration, G_S, M_S, and a_S can therefore be precomputed. The state-estimation error is evaluated on $\widehat{x}_3$ before reencoding.

The affine readout is used only for sparse-state estimation. Forecasting re-enters the original Koopman-MeshFT realization by encoding $\widehat{x}_3$ once and using the learned nonlinear decoder:

$$\widehat{x}_{3+h} = \mathcal{D}\left(\widetilde{K}^h \Psi(\widehat{x}_3)\right), \qquad h = 1, \ldots, 16. \tag{60}$$

Inverse FNO baseline. Let $\mathcal{F}_\theta$ denote the frozen FNO one-step map in the same standardized coordinates. A prior over physical initial states is estimated from the training initial conditions. Writing $\Sigma_x = L_x L_x^\top$, the initial state is parameterized as $x_0 = \mu_x + L_x w$ and inferred by minimizing

$$J_\lambda(w) = \sum_{j=0}^{3} \left\| S\mathcal{F}_\theta^j(\mu_x + L_x w) - y_j \right\|_2^2 + \lambda \|w\|_2^2. \tag{61}$$

Only w is optimized, while the FNO weights remain fixed. We perform 50 Adam updates with learning rate 0.05 from each of two fixed initializations, the prior mean and an observation-based interpolated state, and retain the solution with the lower regularized observation-window objective. The prior regularization is selected using validation data without access to future test targets.

The inferred state and subsequent forecast are

$$\widehat{x}_3^{\text{FNO}} = \mathcal{F}_\theta^3(\widehat{x}_0^{\text{FNO}}), \qquad \widehat{x}_{3+h}^{\text{FNO}} = \mathcal{F}_\theta^h(\widehat{x}_3^{\text{FNO}}), \qquad h = 1, \ldots, 16. \tag{62}$$

Numerical observation integration. The latent step is the order-eight, four-substep Taylor action $\widetilde{K}_\Delta = [\sum_{r=0}^{8} (\Delta A/4)^r / r!]^4$. For CNS2D, periodic structure permits evaluation through Fourier blocks with all spatial modes retained. The regularized latent covariance is represented as a sample factor plus a diagonal term, and the gain is obtained from a solve in the $4 \times 128 \times 4 = 2048$ dimensional observation space without materializing the full covariance. These are computational representations of the specified estimator, not additional truncations of the latent dynamics.

Evaluation and timing. State-estimation error uses $\widehat{x}_3$ before reencoding. Forecast error uses $t_4, \ldots, t_{19}$. Both are trajectory-wise physical relative L^2 errors. For CNS2D tracking, use the densities f and spatial means from Appendix B.5, but RMS training scales $s_c^{\text{sp}} =$

Table 14: Placement-specific CNS2D sparse-observation results with 128 sensors. Uniform uses a regularly spaced periodic subgrid, random uses randomly selected spatial sites, and coarse-fine combines 64 globally distributed coarse-grid sensors with 64 additional sensors concentrated in a local patch. The placement geometry is determined independently of the field values. Values are mean $\pm$ sample standard deviation over five model seeds after averaging trajectories and sensor realizations within each seed. Panel (a) reports physical relative L^2 errors, and panel (b) reports train-scaled physical-quantity tracking errors. Boldface marks the lower mean between the two methods within each placement and metric.

(a) State estimation and forecasting

Placement	Method	State estimation at t_3	Forecast Rel. L^2
Uniform	Koopman-MeshFT	0.2909 ± 0.1109	0.2050 ± 0.0824
	FNO inverse	$\mathbf{0.1947} \pm 0.0152$	$\mathbf{0.1983} \pm 0.0196$
Random	Koopman-MeshFT	0.2936 ± 0.1156	$\mathbf{0.2105} \pm 0.0826$
	FNO inverse	$\mathbf{0.2205} \pm 0.0208$	0.2218 ± 0.0297
Coarse-fine	Koopman-MeshFT	0.2931 ± 0.1104	0.2079 ± 0.0823
	FNO inverse	$\mathbf{0.2010} \pm 0.0240$	$\mathbf{0.2019} \pm 0.0183$

(b) Physical-quantity tracking

Placement	Method	Mass	Total energy
Uniform	Koopman-MeshFT	$\mathbf{0.0262} \pm 0.0089$	$\mathbf{0.0388} \pm 0.0108$
	FNO inverse	0.0616 ± 0.0058	0.0846 ± 0.0064
Random	Koopman-MeshFT	$\mathbf{0.0280} \pm 0.0109$	$\mathbf{0.0405} \pm 0.0104$
	FNO inverse	0.0691 ± 0.0058	0.0923 ± 0.0129
Coarse-fine	Koopman-MeshFT	$\mathbf{0.0293} \pm 0.0095$	$\mathbf{0.0402} \pm 0.0098$
	FNO inverse	0.0688 ± 0.0071	0.0912 ± 0.0062

$\max\{[\mathrm{mean}_{\mathrm{tr}}\langle|f_c|\rangle^2]^{1/2}, 10^{-12}\}$. Each reported component averages $|\widehat{\mathcal{Q}}_{j,c} - \mathcal{Q}_{j,c}|/s_c^{\mathrm{sp}}$ over $j = 4, \ldots, 19$ and trajectories, before the sensor/seed aggregation below.

End-to-end latency begins with standardized observation frames on the host and ends when the forecast returns to the host. It includes state estimation, one-time Koopman reencoding when applicable, forecasting, and required host-device transfers. Model loading, readout and prior fitting, validation selection, and sensor-layout-specific gain preparation are offline costs and are excluded. The measurements use warmed batch-one execution with synchronization at device boundaries. The reported hardware is an AMD EPYC 9124 CPU for Burgers and an NVIDIA H100 PCIe GPU for CNS2D. CNS2D timings are reference measurements, not a dedicated controlled latency benchmark.

Trajectories and sensor realizations are first averaged within each model seed. The reported means and sample standard deviations are then computed over five model seeds. For CNS2D, Table 5 reports results after averaging the three sensor-placement families within each model seed, while Table 14 reports the same comparison separately for each placement. The placement-specific results show the same qualitative pattern as the placement-averaged comparison. FNO inverse estimates the current state more accurately for all three placements, while forecast-error ordering varies across placements. Koopman-MeshFT gives lower mass and total-energy tracking errors for every placement. Thus the aggregate physical-tracking result is not driven by a single sensor layout.