PARAMETRIC KOOPMAN NEURAL OPERATORS FOR CONDITION-DEPENDENT PDE DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators have emerged as powerful surrogates for learning evolution operators underlying PDE dynamics in complex dynamical systems. However, time-varying physical conditions, such as forcing and background fields, induce condition-dependent dynamics, posing the challenge of adapting evolution operators to changing conditions while preserving temporal composition across intervals. We introduce the **Parametric Koopman Neural Operator (PKNO)**, which learns condition-dependent Koopman dynamics in shared observable coordinates. We construct shared hybrid observables over finite field histories, combining prescribed physical quantities with learned supplementary observables to capture temporal context in condition-independent coordinates. On these shared coordinates, we learn a condition-to-propagator map that generates mode-wise spectral Koopman propagators, enabling ordered composition under time-varying physical conditions. We further condition propagator generation on the rolling predicted history and derive a rollout-error bound characterizing how rolling-history sensitivity amplifies local prediction errors. Across four PDE benchmarks, PKNO is competitive under standard fixed conditions and demonstrates substantial gains over conditional baselines in challenging time-varying regimes.

1 INTRODUCTION

Partial differential equations (PDEs) compactly describe a broad range of physical processes, many of which exhibit dynamics that vary systematically with physical conditions. Neural operators learn mappings between function spaces by parameterizing nonlocal operator transformations (Kovachki et al., 2023), classically motivated by integral-kernel formulations, with representative architectures including spectral Fourier Neural Operators (FNOs) (Li et al., 2021; Park et al., 2025), multiresolution Wavelet Neural Operators (WNOs) (Tripura & Chakraborty, 2023; Lei & Li, 2025), and the attention-based General Neural Operator Transformer (GNOT) (Hao et al., 2023; Wang et al., 2026), and applications spanning fluid dynamics (Lyu et al., 2023; Hu et al., 2025), plasma physics (Kim et al., 2025), and weather prediction (Hu et al., 2026). As physical conditions evolve, however, the underlying evolution law changes accordingly, making PDE evolution inherently condition dependent and potentially non-autonomous. This raises a central question:

> How can we learn a composable family of condition-dependent evolution operators for non-autonomous PDE dynamics?

Learning a composable family of condition-dependent evolution operators for non-autonomous PDE dynamics involves two coupled challenges. First, learning condition-dependent operators introduces a fundamental trade-off between adaptability and generalization: explicitly assigning operators to different conditions increases model complexity and limits transfer to unseen conditions, whereas a fully shared operator fails to capture condition-dependent dynamics. Existing neural operators mitigate this trade-off through direct incorporation of coefficient or parameter fields as inputs (Li et al., 2021), multi-input architectures (Jin et al., 2022; Hao et al., 2023), and hypernetwork-based modulation (Jing et al., 2026). However, these approaches still represent condition dependence within a shared predictor, leaving unresolved how to simultaneously achieve adaptive dynamics and generalizable evolution operators across varying physical conditions. Second, non-autonomous evolution requires preserving temporal composability among a sequence of condition-dependent operators.

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.

Existing neural operators, however, typically perform rollout through repeated prediction rather than explicitly composing operators associated with changing physical conditions. Autoregressive neural PDE solvers realize temporal rollout through repeated field prediction, where long rollouts can amplify errors associated with unresolved memory or neglected spatial frequencies (Lippe et al., 2023; Buitrago Ruiz et al., 2025). Methods such as PDE-Refiner primarily improve rollout accuracy and stability, while semigroup-aware operator learning explicitly enforces temporal consistency for autonomous dynamics (Chen & Wu, 2023). These approaches mainly address rollout stability or temporal consistency under fixed evolution dynamics, rather than composability among operators induced by changing physical conditions.

The Koopman operator is a linear, infinite dimensional operator that acts on functions over the state space of the dynamical system, trading nonlinear dynamics for linear, but infinite-dimensional dynamics (Koopman, 1931). From this perspective, condition-dependent evolution operators can be represented within a common observable space, establishing a natural basis for composing condition-dependent dynamics over time. However, existing Koopman formulations capture only part of this structure: parametric models can adapt the operator to physical parameters but are mainly formulated for finite-dimensional states (Guo et al., 2025), whereas Koopman neural operators act on PDE fields but typically use a fixed latent propagator (Xiong et al., 2024), preventing the latent evolution law from explicitly adapting to changing conditions across intervals.

In this work, we introduce the **Parametric Koopman Neural Operator (PKNO)**, which extends parametric Koopman learning to PDE fields through condition-generated, mode-wise spectral propagators. We first formulate condition-dependent PDE evolution as a family of interval-indexed flow maps and lift their temporal composition to Koopman dynamics on finite histories (Section 2). We then show that prescribed target observables admit a condition-independent finite representation uniformly over admissible conditions, without requiring a shared finite-dimensional invariant subspace (Theorem 1 in Section 2.3). Guided by this result, PKNO separates condition-independent observable representation from condition-dependent evolution. It constructs hybrid observables by combining prescribed physical quantities with learned supplementary fields (Section 3.1), and learns a condition-to-propagator map that generates mode-wise spectral Koopman propagators (Section 3.2; Figure 1). Because these propagators act in shared coordinates, their ordered composition directly represents non-autonomous evolution across successive intervals. For history-adaptive autoregressive rollout, predicted fields update the history used to generate subsequent propagators. Theorem 2 characterizes the resulting long-horizon error in terms of local one-step prediction error and sensitivity to perturbations in the rolling history. Empirically, PKNO substantially improves over both direct conditional prediction and condition-dependent Koopman baselines on time-varying Navier–Stokes dynamics (Section 4). We hope our results will contribute to AI for Science by moving beyond condition-aware prediction toward explicit learning of physical evolution laws under changing conditions.

2 From Condition-Dependent Flows to Composable Koopman Dynamics

Although Koopman operators recast nonlinear dynamics as linear evolution in observable space, extending this viewpoint to condition-dependent PDE dynamics is non-trivial. First, physical conditions may include global parameters, distributed fields, and time-varying inputs, so the dynamics form a family of evolution laws indexed by interval conditions rather than a single fixed operator. Second, partial field observations may not form a Markov state, making the observed evolution depend on recent history. Third, successive condition-specific dynamics must remain composable in a common finite representation, yet a condition-dependent Koopman family need not admit a shared finite-dimensional invariant subspace. We address these challenges by formulating condition-indexed PDE flows on a common state space, lifting them to Koopman dynamics on finite histories, and seeking shared finite coordinates through uniform approximation of prescribed target evolution.

2.1 Condition-Indexed PDE Flows

We model condition-dependent PDE evolution as a family of flow maps indexed by the physical conditions acting over each time interval. Let $\Omega \subset \mathbb{R}^d$ be the physical domain and $\mathcal{X}$ a common Banach space of $\mathbb{R}^m$-valued fields after boundary lifting. Over a time horizon $T > 0$, the physical

condition path $\mathbf{c} : [0, T] \to \mathcal{C}_\mathrm{g} \times \mathcal{C}_\mathrm{f} =: \mathcal{C}$ is written as $\mathbf{c}(t) = (\mathbf{c}_\mathrm{g}, \mathbf{c}_\mathrm{f}(t))$, where $\mathbf{c}_\mathrm{g} = \boldsymbol{\beta}$ and $\mathbf{c}_\mathrm{f}(t) = (\mathbf{b}(\cdot, t), \mathbf{r}(\cdot, t), \mathbf{q}(\cdot, t))$. Here $\mathcal{C}_\mathrm{g} \subset \mathbb{R}^{d_\mathrm{g}}$ contains global parameters, and $\mathcal{C}_\mathrm{f}$ contains boundary data on $\partial\Omega$, forcing and spatially varying physical fields on Ω. We equip $\mathcal{C}$ with the product topology. For $\mathbf{x}(t) := \mathbf{x}(\cdot, t) \in \mathcal{X}$, consider

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathcal{A}_{\mathbf{c}(t)}\big(\mathbf{x}(t)\big), \qquad \mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{X},$$

where $\mathcal{A}_{\mathbf{c}(t)}$ is the condition-dependent differential operator governing the evolution. This formulation covers standard evolutionary PDEs and their dynamical-systems view (Brunton & Kutz, 2024; Brunton et al., 2022).

For $0 \le s \le t \le T$, let $\mathbf{c}_{[s,t]} := \mathbf{c}|_{[s,t]}$ denote the restricted condition path. Under standard regularity and boundary-compatibility conditions ensuring well-posedness (Amann, 1995), the corresponding condition-indexed flow map is

$$\mathbf{F}_s^t[\mathbf{c}_{[s,t]}] : \mathcal{X} \to \mathcal{X}, \qquad \mathbf{x}(s) \mapsto \mathbf{x}(t),$$

and, for $s \le \tau \le t$, the maps satisfy

$$\mathbf{F}_s^t[\mathbf{c}_{[s,t]}] = \mathbf{F}_\tau^t[\mathbf{c}_{[\tau,t]}] \circ \mathbf{F}_s^\tau[\mathbf{c}_{[s,\tau]}], \qquad \mathbf{F}_s^s = \mathrm{Id}_\mathcal{X}\,. \tag{1}$$

Remark 1 (Temporal Composability.)**.** *Although the physical flows $\mathbf{F}_s^t[\mathbf{c}_{[s,t]}]$ compose exactly on the common state space $\mathcal{X}$, a learned representation may not: different interval conditions can induce incompatible coordinates. This motivates a shared observable space for condition-indexed evolution.*

Suppose the continuous trajectory is observed at $0 = t_0 < \cdots < t_N \le T$, with $\mathbf{x}_n := \mathbf{x}(t_n)$ and $\Delta t_n := t_{n+1} - t_n$ for $n < N$. For fixed-step data, $\Delta t_n \equiv \Delta t$. Let $\mathcal{C}_{\Delta t}$ denote the admissible interval-condition descriptors, and $\mathbf{c}_n \in \mathcal{C}_{\Delta t}$ describe the physical conditions on $[t_n, t_{n+1}]$. The corresponding discrete update is

$$\mathbf{x}_{n+1} = \mathbf{F}_{\Delta t}[\mathbf{c}_n](\mathbf{x}_n).$$

Here $\mathbf{c}_n$ contains exogenous physical conditions, such as static parameters, spatially varying fields, or sampled condition paths, and is distinct from state-derived inputs such as the field history. The primary learning object is therefore the condition-indexed family

$$\{\mathbf{F}_{\Delta t}[\mathbf{c}] : \mathcal{X} \to \mathcal{X}\}_{\mathbf{c} \in \mathcal{C}_{\Delta t}}\,. \tag{2}$$

From an operator-learning perspective, the goal is to approximate this family from observed trajectories while retaining its dependence on the interval conditions. We next lift this nonlinear flow family to a common Koopman observable space, where condition-specific evolution can be represented and composed in shared coordinates.

2.2 Lifting Temporal Composition to Observable Dynamics

To preserve the temporal composition structure of the condition-indexed flow family in Eq. (2), we lift each interval-conditioned evolution to a Koopman operator on a common observable space.

Although each full-state update associated with Eq. (2) is Markovian, operator-learning data often provide only a partial field description, with unobserved state variables, forcing, or unresolved physical processes still influencing the observed dynamics. The resulting reduced description need not be Markovian and can exhibit effective memory (Lin et al., 2021). For example, in our shallow-water benchmark (Appendix A.1), only the water-depth channel is observed, while momentum variables remain unobserved, so a single depth field does not constitute the full Markov state. We therefore approximate this unresolved memory with a finite field history. For $N_h \ge 1$ and $n \ge N_h - 1$, define $\mathbf{h}_n = (\mathbf{x}_{n-N_h+1}, \ldots, \mathbf{x}_n) \in \mathcal{X}^{N_h}$. Finite history approximates this memory without assuming exact closure; $N_h = 1$ recovers the full-state description. The map $\mathbf{H}_{\Delta t}[\mathbf{c}_n] : \mathcal{X}^{N_h} \to \mathcal{X}^{N_h}$ updates the history as

$$\mathbf{h}_{n+1} := \mathbf{H}_{\Delta t}[\mathbf{c}_n](\mathbf{h}_n) = (\mathbf{x}_{n-N_h+2}, \ldots, \mathbf{x}_n, \mathbf{F}_{\Delta t}[\mathbf{c}_n](\mathbf{x}_n))\,.$$

Following the standard Koopman composition operator (Koopman, 1931; Brunton et al., 2022), let $\mathcal{G}$ be a complex vector space of scalar observables $\varphi : \mathcal{X}^{N_h} \to \mathbb{C}$, preserved under composition with every admissible history update. Define

$$\mathcal{K}_{\Delta t}[\mathbf{c}] : \mathcal{G} \to \mathcal{G}, \qquad \text{s.t.} \qquad \mathcal{K}_{\Delta t}[\mathbf{c}]\varphi := \varphi \circ \mathbf{H}_{\Delta t}[\mathbf{c}]. \tag{3}$$

Thus $(\mathcal{K}_{\Delta t}[\mathbf{c}_n]\varphi)(\mathbf{h}_n) = \varphi(\mathbf{h}_{n+1})$, linearly in φ. For $r \geq 1$, define the grid-based multi-step map by

$$\mathbf{H}_{t_n}^{t_{n+r}}[\mathbf{c}_{[t_n,t_{n+r}]}] := \mathbf{H}_{\Delta t}[\mathbf{c}_{n+r-1}] \circ \cdots \circ \mathbf{H}_{\Delta t}[\mathbf{c}_n].$$

The corresponding multi-step Koopman operator is defined by $\mathcal{K}_{t_n}^{t_{n+r}}\varphi := \varphi \circ \mathbf{H}_{t_n}^{t_{n+r}}$. For $r = 1$ it reduces to $\mathcal{K}_{\Delta t}[\mathbf{c}_n]$. The multi-step history map composes the interval updates inherited from Eq. (1). Applying the Koopman definition in Eq. (3), consecutive operators satisfy

$$\mathcal{K}_{t_n}^{t_{n+2}}[\mathbf{c}_{[t_n,t_{n+2}]}] = \mathcal{K}_{t_n}^{t_{n+1}}[\mathbf{c}_{[t_n,t_{n+1}]}] \circ \mathcal{K}_{t_{n+1}}^{t_{n+2}}[\mathbf{c}_{[t_{n+1},t_{n+2}]}], \tag{4}$$

where the condition path on $[t_n, t_{n+2}]$ is obtained by concatenating the two interval paths.

Equation (4) is the Koopman pullback counterpart of the flow-composition relation in Eq. (1). Since Koopman operators act on observables by composition, their product order is reversed relative to forward state propagation. This identity holds on the common observable space $\mathcal{G}$ without requiring a finite dictionary. The finite-coordinate realization introduced later translates this temporal composition into an ordered matrix product for forward rollout.

2.3 Shared Observable Coordinates beyond Invariant Closure

To preserve Koopman composition in finite coordinates, we seek a shared representation for the condition-dependent evolution of task-relevant observables. We prescribe a field-valued map $\mathbf{g}_{\mathrm{phys}}(\mathbf{h}_n)$ of task-relevant physical quantities and select scalar target observables $g_j : \mathcal{X}^{N_h} \to \mathbb{R}$, $j \in [p] := \{1, \ldots, p\}$. In PKNO, these scalar targets are obtained from the prescribed physical fields as the real and imaginary parts of their retained Fourier coefficients.

On a fixed observation grid with step Δt, repeated application of a single Koopman operator $\mathcal{K}_{\Delta t}[\mathbf{c}]$ under a fixed interval condition $\mathbf{c}$ generates a Krylov sequence (Williams et al., 2015; Arbabi & Mezić, 2017). Under changing conditions, observable evolution instead involves ordered products of condition-dependent Koopman operators. Define

$$\mathcal{V}_{\mathrm{cond}} := \mathrm{span}\big\{\mathcal{K}_{\Delta t}[\mathbf{c}_0] \cdots \mathcal{K}_{\Delta t}[\mathbf{c}_{\ell-1}]g_j : \ell \geq 0,\ \mathbf{c}_i \in \mathcal{C}_{\Delta t},\ j \in [p]\big\}.$$

The empty product for $\ell = 0$ is the identity, and the products follow the pullback composition order in Eq. (4). In general, $\mathcal{V}_{\mathrm{cond}}$ need not be finite-dimensional, so exact finite-dimensional invariance is unnecessarily restrictive. We therefore ask a weaker question: can one condition-independent finite dictionary containing $g_1, \ldots, g_p$ uniformly approximate the one-step evolution of prescribed targets $\mathcal{K}_{\Delta t}[\mathbf{c}]g_j$ across admissible conditions?

We show that prescribed target images admit a uniform finite-dimensional approximation over compact sets of complete interval descriptors. We state this result in real $L^2(\mu)$ over compact interval descriptors, without requiring a finite-dimensional invariant subspace. Equip $\mathcal{X}^{N_h}$ with its product topology and Borel σ-algebra, and let μ be a finite Borel measure.

Theorem 1 (Uniform one-step approximation of finite target observables)**.** *Assume $\mathcal{C}_{\Delta t}$ is compact and let $g_j \in C_b(\mathcal{X}^{N_h}; \mathbb{R})$, $j = 1, \ldots, p$. Suppose each history update $\mathbf{H}_{\Delta t}[\mathbf{c}]$ is Borel measurable and $\mathbf{c} \mapsto \mathbf{H}_{\Delta t}[\mathbf{c}](\mathbf{h})$ is continuous for μ-almost every $\mathbf{h}$. Then, for every $\varepsilon > 0$, there exist a condition-independent finite dictionary*

$$\mathbf{\Psi} = (g_1, \ldots, g_p, \psi_{p+1}, \ldots, \psi_{N_\Psi})^\top \in L^2(\mu)^{N_\Psi}$$

and a continuous map $\mathbf{A} : \mathcal{C}_{\Delta t} \to \mathbb{R}^{p \times N_\Psi}$ such that

$$\sup_{\mathbf{c} \in \mathcal{C}_{\Delta t}} \left\| \begin{bmatrix} \mathcal{K}_{\Delta t}[\mathbf{c}]g_1 \\ \vdots \\ \mathcal{K}_{\Delta t}[\mathbf{c}]g_p \end{bmatrix} - \mathbf{A}(\mathbf{c})\mathbf{\Psi} \right\|_{L^2(\mu;\mathbb{R}^p)} \leq \varepsilon.$$

The complete proof is provided in Appendix B. The dictionary is shared across conditions, while $\mathbf{A}(\mathbf{c})$ adapts the coefficients of the prescribed targets' one-step images. Theorem 1 shows that shared finite coordinates do not require exact closure of the full condition-dependent Koopman family: the dictionary can remain fixed while the coefficients representing prescribed target evolution vary with the condition. In PKNO, the prescribed targets are realized by retained physical Fourier coefficients, while learned supplementary fields provide additional shared coordinates.

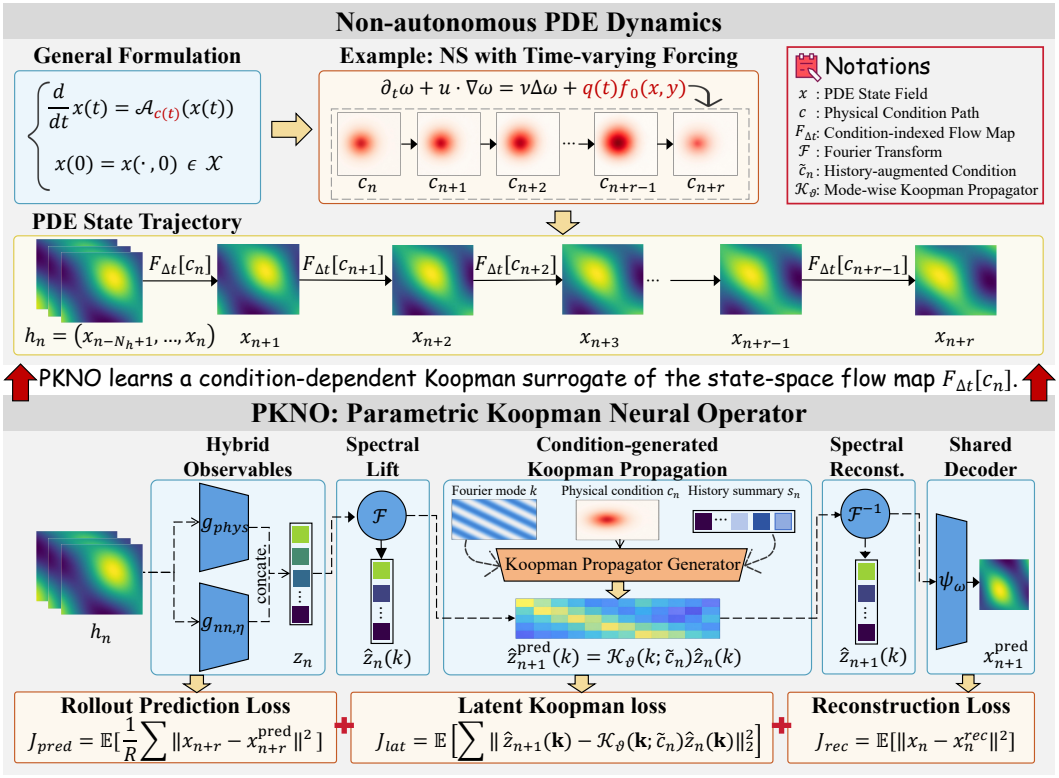

Figure 1: PKNO for condition-dependent non-autonomous PDE dynamics.

3 PKNO: Parametric Koopman Neural Operator

Motivated by the shared finite-target approximation in Theorem 1, PKNO constructs a finite-coordinate surrogate for the condition-indexed flow family in Eq. (2). It separates condition-independent observable representation from condition-dependent evolution through hybrid field-level observables, condition-generated mode-wise propagators, and their ordered composition for non-autonomous rollout (Figure 1).

3.1 Conditional-Independent Hybrid Observables

To realize a condition-independent observable representation at the field level, PKNO uses a shared encoder–decoder across all physical conditions. The encoder combines prescribed physical quantities, corresponding to the target observables $g_1, \ldots, g_p$, with learned supplementary fields that enrich the shared representation.

Hybrid Observables. Let $\mathcal{Z}$ denote the latent observable field space, and let $\mathbf{g}_{\mathrm{nn},\eta}$ be a supplementary field map with trainable parameters η. The hybrid mapping $\mathbf{g}_\eta : \mathcal{X}^{N_h} \to \mathcal{Z}$ is

$$\mathbf{z}_n = \mathbf{g}_\eta(\mathbf{h}_n) := \begin{bmatrix} \mathbf{g}_{\mathrm{phys}}(\mathbf{h}_n) \\ \mathbf{g}_{\mathrm{nn},\eta}(\mathbf{h}_n) \end{bmatrix},$$

where $\mathbf{g}_{\mathrm{phys}}$ collects the prescribed physical quantities from the history. The physical and learned fields are concatenated along the channel dimension, and $\mathbf{g}_\eta$ is shared across all conditions.

Shared Decoder. Let $\psi_\omega : \mathcal{Z} \to \mathcal{X}$ denote a decoder parameterized by ω, with reconstruction $\mathbf{x}_n^{\mathrm{rec}} = \psi_\omega(\mathbf{z}_n)$. The same decoder acts on encoded fields for reconstruction and propagated fields for prediction. Physical conditions are therefore supplied only to the propagator generator, not to the encoder or decoder, keeping the observable coordinates shared across conditions.

3.2 CONDITION-GENERATED SPECTRAL PROPAGATION

With the observable coordinates fixed across conditions, PKNO represents changing physical dynamics through condition- and frequency-dependent Koopman propagators. The observable basis remains fixed while the propagator varies with the interval condition, enabling successive evolution laws to compose directly in temporal order.

Spectral Representation. Let p_z denote the number of channels in the shared latent observable field space. The spatial Fourier transform acts channel-wise, with coefficients $\widehat{\mathbf{z}}_n(\mathbf{k}) := (\mathcal{F}\mathbf{z}_n)(\mathbf{k}) \in \mathbb{C}^{p_z}$ and inverse $\mathbf{z}_n = \mathcal{F}^{-1}\widehat{\mathbf{z}}_n$. PKNO retains $Z_{k_{\max}} := \{\mathbf{k} \in \mathbb{Z}^d : \|\mathbf{k}\|_\infty \le k_{\max}\}$, with spectral projection $\Pi_{k_{\max}}$. The retained physical coefficients, represented by their real and imaginary parts, provide finite target observables compatible with Theorem 1.

Condition-Dependent Propagation. For each retained mode $\mathbf{k} \in Z_{k_{\max}}$ and interval condition $\mathbf{c}_n$, let $\mathbf{K}_\vartheta(\mathbf{k}; \mathbf{c}_n) \in \mathbb{C}^{p_z \times p_z}$ denote the propagation matrix with trainable parameters ϑ. The mathematical core advances the latent coefficient as

$$\widehat{\mathbf{z}}^{\mathrm{pred}}_{n+1}(\mathbf{k}) = \mathbf{K}_\vartheta(\mathbf{k}; \mathbf{c}_n)\widehat{\mathbf{z}}_n(\mathbf{k}), \qquad \mathbf{k} \in Z_{k_{\max}}. \tag{5}$$

PKNO learns the map $(\mathbf{k}, \mathbf{c}_n) \mapsto \mathbf{K}_\vartheta(\mathbf{k}; \mathbf{c}_n)$ with a shared generator that jointly encodes Fourier-mode coordinates and the scalar and field-valued components of $\mathbf{c}_n$. Extending frequency-coordinate amortization (Xiao et al., 2024) to condition-dependent Koopman dynamics, the generator combines a condition-independent spectral base with a learned condition-dependent correction. This construction allows different spatial frequencies to respond differently to changing physical conditions while sharing parameters across modes; Hermitian symmetry recovers redundant coefficients for real-valued fields.

Ordered Rollout. Given an observed history $\mathbf{h}_n$, the shared encoder initializes the latent state $\widehat{\mathbf{z}}_n$. For an r-step forecast, let $\{\mathbf{c}_n, \ldots, \mathbf{c}_{n+r-1}\}$ denote the prescribed sequence of interval conditions, where $\mathbf{c}_{n+j}$ describes $[t_{n+j}, t_{n+j+1}]$. Repeated application of Eq. (5) in forward time gives

$$\widehat{\mathbf{z}}^{\mathrm{pred}}_{n+r}(\mathbf{k}) = \mathbf{K}_\vartheta(\mathbf{k}; \mathbf{c}_{n+r-1}) \cdots \mathbf{K}_\vartheta(\mathbf{k}; \mathbf{c}_n)\widehat{\mathbf{z}}_n(\mathbf{k}). \tag{6}$$

Equation (6) is the finite-coordinate forward realization of the Koopman pullback composition rule in Eq. (4). The ordered product preserves the temporal sequence of changing interval conditions, yielding a direct non-autonomous rollout in the shared observable coordinates.

The retained-mode, fixed-step formulation admits complementary spatial and temporal extensions. Output-space high-frequency refinement targets fine-scale structure omitted by spectral truncation, while a continuous-time generator supports variable prediction intervals and conditions that evolve within an interval. Together, they broaden the resolved spatial bandwidth and temporal flexibility of PKNO without changing its shared observable coordinates, as detailed in Appendix D.

3.3 HISTORY-ADAPTIVE AUTOREGRESSIVE ROLLOUT

Equation (6) describes ordered composition in the prescribed-condition mathematical core. In the history-adaptive implementation, each predicted field becomes part of the history used to generate subsequent propagators, coupling current prediction errors with future dynamics. We therefore formulate this closed-loop rollout, specify its training objective, and characterize how local prediction errors accumulate through repeated history-conditioned propagation.

History-Adaptive Implementation. The propagator generator can augment each prescribed interval condition with a summary of the current field history, allowing the selected evolution law to adapt to state-dependent context. We write the resulting context as $\widetilde{\mathbf{c}}_n = (\mathbf{c}_n, \mathbf{s}_n)$, where $\mathbf{s}_n = S(\mathbf{h}^{\mathrm{roll}}_n) \in \mathcal{S}$ summarizes the recent field history. Thus, the one-step propagation in Eq. (5) is evaluated with $\widetilde{\mathbf{c}}_n$ in place of $\mathbf{c}_n$. In the reported fixed-metadata benchmarks, $\mathbf{c}_n = \mathbf{m}$ within each dataset-specific run, so $\widetilde{\mathbf{c}}_n = (\mathbf{m}, \mathbf{s}_n)$ (Appendix A.1). During rollout, field-history autoregression updates the summary as the predicted trajectory evolves, producing a history-conditioned sequence of propagators. In implementation, $\mathbf{s}_n$ is realized by a state encoder that maps the current rolling field history to a fixed-dimensional context embedding, which is recomputed after each autoregressive update. Such history dependence can be especially informative when physical drivers are only

partially observed. For example, Reinbold et al. (2021) reconstructed an unmeasured forcing field from velocity data by incorporating physical constraints. This suggests that response histories may carry indirect information about forcing even when it cannot be measured directly.

Learning Objective. PKNO jointly optimizes observable reconstruction, latent Koopman evolution, and physical-field prediction. The reconstruction and rollout objectives are defined as $\mathcal{J}_{\mathrm{rec}} = \mathbb{E}[\|\mathbf{x}_n - \mathbf{x}_n^{\mathrm{rec}}\|_{\mathcal{X}}^2]$ with $\mathbf{x}_n^{\mathrm{rec}} = \boldsymbol{\psi}_\omega(\mathbf{z}_n)$ and $\mathcal{J}_{\mathrm{pred}} = \mathbb{E}[\frac{1}{R}\sum_{r=1}^{R}\|\mathbf{x}_{n+r} - \mathbf{x}_{n+r}^{\mathrm{pred}}\|_{\mathcal{X}}^2]$, which ensure accurate state recovery and long-horizon forecasting, respectively. To explicitly enforce condition-dependent Koopman evolution in the shared observable space, we introduce the latent consistency objective:

$$\mathcal{J}_{\mathrm{lat}} = \mathbb{E}\left[\sum_{\mathbf{k}\in Z_{k_{\max}}} \|\widehat{\mathbf{z}}_{n+1}(\mathbf{k}) - \mathbf{K}_\vartheta(\mathbf{k};\widetilde{\mathbf{c}}_n)\widehat{\mathbf{z}}_n(\mathbf{k})\|_2^2\right]. \tag{7}$$

Equation (7) penalizes the one-step residual of the condition-dependent propagation in Eq. (5), with the prescribed condition augmented by the rolling-history summary. The overall objective is

$$\mathcal{J} = \lambda_{\mathrm{rec}}\mathcal{J}_{\mathrm{rec}} + \lambda_{\mathrm{lat}}\mathcal{J}_{\mathrm{lat}} + \lambda_{\mathrm{pred}}\mathcal{J}_{\mathrm{pred}}. \tag{8}$$

Error Bound. To analyze the closed-loop history-adaptive realization of Eq. (5), we view PKNO as a family of one-step predictors $\mathcal{P}_{\mathbf{c}} : \mathcal{X}^{N_h} \to \mathcal{X}$ indexed by interval conditions $\mathbf{c} \in \mathcal{C}_{\Delta t}$. For a history $\mathbf{h}$, the decoded core prediction is $\mathcal{P}_{\mathbf{c}}(\mathbf{h}) := \mathcal{D}_\omega(\mathsf{K}(\mathbf{c}, S(\mathbf{h}))\mathcal{E}_\eta(\mathbf{h}))$. Here $\mathcal{E}_\eta$, $\mathcal{D}_\omega$, and K denote retained-mode encoding, decoding, and block-diagonal propagation, respectively. For histories $\mathbf{h} = (\mathbf{u}_1, \ldots, \mathbf{u}_{N_h})$ and $\widetilde{\mathbf{h}} = (\widetilde{\mathbf{u}}_1, \ldots, \widetilde{\mathbf{u}}_{N_h})$, define $\|\mathbf{h} - \widetilde{\mathbf{h}}\|_{\mathrm{hist}} := \max_{1\le\ell\le N_h}\|\mathbf{u}_\ell - \widetilde{\mathbf{u}}_\ell\|_{\mathcal{X}}$, and let $G_r(a) := \sum_{q=0}^{r-1} a^q$. Following the consistency–stability view of time stepping and recent analyses of autoregressive neural PDE solvers (Hairer et al., 1993; Brandstetter et al., 2022; McCabe et al., 2023), we characterize long-horizon error through local prediction error and sensitivity to perturbations in the rolling history.

Theorem 2 (Condition-dependent autoregressive rollout error)**.** *For any horizon $r \ge 1$, consider an admissible condition sequence $\mathbf{c}_n, \ldots, \mathbf{c}_{n+r-1} \in \mathcal{C}_{\Delta t}$. Let $\mathbf{h}_n^{\mathrm{roll}} = \mathbf{h}_n$. For $j = n, \ldots, n+r-1$, define $\mathbf{x}_{L,j+1}^{\mathrm{pred}} = \mathcal{P}_{\mathbf{c}_j}(\mathbf{h}_j^{\mathrm{roll}})$, and obtain $\mathbf{h}_{j+1}^{\mathrm{roll}}$ by shifting $\mathbf{h}_j^{\mathrm{roll}}$ and appending $\mathbf{x}_{L,j+1}^{\mathrm{pred}}$. Suppose the local one-step error along the reference trajectory satisfies $\|\mathbf{x}_{j+1} - \mathcal{P}_{\mathbf{c}_j}(\mathbf{h}_j)\|_{\mathcal{X}} \le \varepsilon_{\mathrm{loc}}$ for $j = n, \ldots, n+r-1$, and the predictor family is uniformly Lipschitz in its history argument: for every admissible $\mathbf{c}$ and histories $\mathbf{h}, \widetilde{\mathbf{h}}$ in the rollout region,*

$$\|\mathcal{P}_{\mathbf{c}}(\mathbf{h}) - \mathcal{P}_{\mathbf{c}}(\widetilde{\mathbf{h}})\|_{\mathcal{X}} \le \Gamma\|\mathbf{h} - \widetilde{\mathbf{h}}\|_{\mathrm{hist}}.$$

Then,

$$\|\mathbf{x}_{n+r} - \mathbf{x}_{L,n+r}^{\mathrm{pred}}\|_{\mathcal{X}} \le \varepsilon_{\mathrm{loc}} G_r(\Gamma).$$

Appendix B provides the proof and component-wise sufficient conditions. The factor Γ measures the sensitivity of the complete condition-indexed predictor to perturbations in its rolling history. Accordingly, the bound separates local prediction accuracy from rollout-history amplification: errors remain uniformly controlled for $\Gamma < 1$, grow at most linearly for $\Gamma = 1$, and may grow geometrically for $\Gamma > 1$. Thus, long-horizon accuracy depends not only on one-step prediction quality but also on how prediction errors propagate through the rolling history.

4 Experiments

In this section, we evaluate PKNO on diverse PDE systems and investigate the following research questions (RQs): (1) Can PKNO achieve accurate and stable long-horizon prediction across diverse PDE systems? (2) Can PKNO accurately compose non-autonomous evolution under time-varying physical conditions? (3) How do shared hybrid observables and condition-dependent propagators contribute to performance? (4) Does PKNO achieve a favorable accuracy–parameter trade-off compared with existing operator models?

Table 1: Main forecasting performance under fixed and time-varying physical conditions. Bold indicates the lowest error within each evaluation block.

Data	Metric	*Fixed physical condition*					*Time-varying physical condition*		
		FNO	**AM-FNO**	**KNO**	**IKNO**	**PKNO**	**Cond. FNO**	**Cond. KNO**	**PKNO**
Burgers	$L_2^{\text{Step}} \downarrow$	0.0091	0.0502	0.0079	0.0094	**0.0050**	–	–	–
	$L_2^{\text{Full}} \downarrow$	0.0091	0.0502	0.0079	0.0094	**0.0050**	–	–	–
	nRMSE $\downarrow$	0.0127	0.0420	0.0098	0.0137	**0.0033**	–	–	–
	fRMSE $\downarrow$	472.0893	495.2618	144.0765	468.4480	**43.0326**	–	–	–
$\text{NS}_{\nu=10^{-3}}$	$L_2^{\text{Step}} \downarrow$	0.0172	0.0333	0.0146	**0.0098**	0.0146	0.0193	0.0250	**0.0161**
	$L_2^{\text{Full}} \downarrow$	0.0180	0.0345	0.0151	**0.0100**	0.0150	0.0193	0.0248	**0.0158**
	nRMSE $\downarrow$	0.0181	0.0347	0.0153	**0.0101**	0.0152	0.0197	0.0251	**0.0159**
	fRMSE $\downarrow$	**886.7193**	930.2107	1583.0093	2850.5215	948.0957	**37.7916**	191.6041	118.5927
$\text{NS}_{\nu=10^{-4}}$	$L_2^{\text{Step}} \downarrow$	0.3967	0.4649	0.4109	**0.3954**	0.4007	0.0308	0.0230	**0.0170**
	$L_2^{\text{Full}} \downarrow$	**0.4575**	0.5169	0.4664	0.4735	0.4604	0.0323	0.0245	**0.0180**
	nRMSE $\downarrow$	0.4798	0.5338	0.4863	0.5028	**0.4788**	0.0325	0.0247	**0.0181**
	fRMSE $\downarrow$	1.1886	**1.0875**	3.1252	2.7301	2.5109	**1.1895**	6.5008	5.3291
SWE	$L_2^{\text{Step}} \downarrow$	0.0110	0.0219	0.0077	**0.0045**	0.0116	**0.0204**	0.0259	0.0431
	$L_2^{\text{Full}} \downarrow$	0.0111	0.0226	0.0077	**0.0045**	0.0117	**0.0197**	0.0248	0.0416
	nRMSE $\downarrow$	0.0112	0.0227	0.0077	**0.0046**	0.0117	**0.0021**	0.0026	0.0042
	fRMSE $\downarrow$	8.9342	5.7214	**1.8082**	1.8602	3.6718	**0.8088**	1.7398	1.9310

4.1 Experimental Setup

Datasets. We evaluate PKNO on four PDE prediction settings, including viscous Burgers, incompressible Navier–Stokes, and shallow-water equations. The Burgers and Navier–Stokes benchmarks follow standard neural operator settings (Li et al., 2021; Xiong et al., 2024), with Navier–Stokes evaluated at viscosities $\nu = 10^{-3}$ and $\nu = 10^{-4}$ to test increasingly challenging autoregressive rollouts. Following PDEBench (Takamoto et al., 2022a;b), we further include a two-dimensional radial dam-break shallow-water dataset for nonlinear wave propagation with sharp fronts. For evaluating non-autonomous evolution, we construct time-varying condition variants for Navier–Stokes and shallow-water systems by introducing interval-wise varying physical conditions during PDE evolution. Detailed descriptions and configurations are provided in Appendix A.1.

Baselines. We compare PKNO with representative neural operator and Koopman-based methods. The neural operator baselines include FNO (Li et al., 2021) and AM-FNO (Xiao et al., 2024), while the Koopman-based baselines include KNO (Xiong et al., 2024), AM-KNO, and IKNO (Jin et al., 2025). For time-varying-condition experiments, we compare with direct conditional FNO (Cond. FNO) and conditional KNO (Cond. KNO) that directly incorporates physical conditions as inputs. Baseline definitions and implementation settings are provided in Appendices A.2 and A.4.

Evaluation Metrics. We evaluate prediction quality using four complementary metrics. Step relative L_2 measures average per-step error, whereas full-rollout relative L_2 evaluates each predicted trajectory as a whole. We additionally report normalized RMSE (nRMSE), which measures global field error over the complete prediction output, and high-frequency spectral RMSE (fRMSE), which isolates prediction error in the prescribed high-frequency Fourier band. Together, the four metrics assess stepwise accuracy, trajectory-level error accumulation, global field fidelity, and fine-scale spectral fidelity. Exact definitions are provided in Appendix A.3.

4.2 Experimental Results

▷ **Long-Horizon Prediction under Fixed Physical Conditions (RQ1).** Table 1 reports results on the standard fixed-condition benchmarks. ❶ **PKNO performs strongly on Burgers.** PKNO achieves the lowest error across all four metrics, reducing full-rollout relative L_2 from 7.9×10^{-3} for KNO to 5.0×10^{-3}. ❷ **PKNO remains competitive on Navier–Stokes.** At $\nu = 10^{-3}$, PKNO closely matches KNO in full-rollout relative L_2 (0.0150 vs. 0.0151), while IKNO performs best. At $\nu = 10^{-4}$, PKNO remains comparable to FNO in full-rollout error and achieves slightly lower nRMSE. ❸ **Performance is weaker on shallow water.** The weaker shallow-water result coincides with a substantially more conservative optimization and propagation configuration, including

a smaller learning rate and residual-update scale and a reduced decomposition depth. We analyze this failure mode further in Appendix A.5.

▷ **Condition-Dependent Propagation (RQ2).** The time-varying-condition results in Table 1 provide two key observations. ❶ **PKNO consistently improves over conditional KNO on Navier–Stokes.** At $\nu = 10^{-3}$ and 10^{-4}, PKNO achieves lower error than Cond. KNO across all four metrics. In particular, full-rollout relative L_2 is reduced from 0.0248 to 0.0158 and from 0.0245 to 0.0180, corresponding to reductions of 36.3% and 26.5%, respectively. This indicates that conditioning the propagator alone is insufficient, and that the shared observable representation together with history-adaptive propagation provides additional benefit. ❷ **PKNO also improves over direct conditional prediction on the Navier–Stokes tasks.** Compared with Cond. FNO, PKNO reduces full-rollout relative L_2 from 0.0193 to 0.0158 at $\nu = 10^{-3}$ and from 0.0323 to 0.0180 at $\nu = 10^{-4}$, with consistent gains in step relative L_2 and nRMSE. This suggests that explicitly parameterizing condition-dependent latent evolution can be advantageous over inputting the condition directly into a nonlinear field predictor.

▷ **Ablation Study (RQ3).** Table 2 isolates the contributions of hybrid observables and condition-dependent propagation on time-varying Navier–Stokes. ❶ **Hybrid observables improve rollout accuracy.** Adding the hybrid dictionary reduces full-rollout relative L_2 from 0.0361 to 0.0248, with consistent gains in step relative L_2 and nRMSE. ❷ **Condition-dependent propagation provides a further gain.** With both components enabled, full-rollout relative L_2 further decreases to 0.0158, and the other physical-space metrics improve accordingly. The fRMSE trend is less monotonic, indicating that gains in global rollout accuracy do not necessarily translate to improved high-frequency fidelity.

Table 2: Ablation on $\mathrm{NS}_{\nu=10^{-3}}$: condition-dependent $\mathbf{K}$ and hybrid observables.

Cond. $\mathbf{K}$	Hybrid dict.	*Time-varying physical condition*			
		$L_2^{\text{Step}}\downarrow$	$L_2^{\text{Full}}\downarrow$	nRMSE↓	fRMSE↓
✗	✗	0.0351	0.0361	0.0370	56.5347
✗	✓	0.0250	0.0248	0.0251	191.6041
✓	✗	0.0215	0.0213	0.0214	**103.6762**
✓	✓	**0.0161**	**0.0158**	**0.0159**	118.5927

▷ **Parameter–Accuracy Trade-off (RQ4).** Figure 2 compares prediction error against the number of trainable parameters on the four fixed-condition benchmarks. ❶ **PKNO achieves competitive accuracy with sub-million parameterization.** PKNO uses 0.411M parameters on Burgers and 0.916M on the two-dimensional tasks, substantially fewer than FNO and AM-FNO on the latter, both of which exceed 2M parameters. ❷ **The trade-off is favorable on Burgers and Navier–Stokes.** PKNO attains the lowest measured rollout MSE on Burgers and remains close to the best-performing models on both Navier–Stokes regimes, while using considerably fewer parameters than the larger neural-operator baselines.

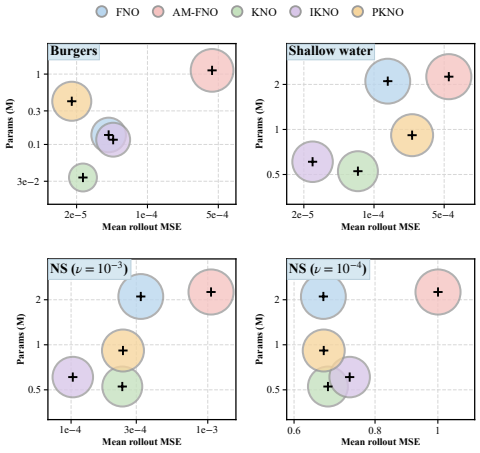

Figure 2: Parameter–error trade-off on the four fixed-condition benchmarks.

5 Conclusion

In this paper, we present PKNO, a parametric Koopman neural operator for condition-dependent PDE dynamics that learns composable evolution operators in shared observable coordinates. We further introduce hybrid observables and condition-generated spectral propagators, together with theoretical results characterizing shared finite representations and autoregressive rollout error. Extensive experiments across multiple PDE benchmarks show that PKNO is competitive under fixed conditions and achieves substantial gains over conditional baselines in time-varying regimes. We hope PKNO will inspire future research on structured operator learning for non-autonomous and condition-dependent physical systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, improving the clarity and readability of the manuscript. We did not use generative AI to generate experimental data or results, conduct the core scientific analysis, or make research conclusions. All AI-assisted text was carefully reviewed, revised, and verified by the authors. The authors take full responsibility for the final content of this work, including any text, code, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable or sensitive data, or other ethical concerns requiring special consideration. The experiments are conducted on standard PDE benchmarks and numerical simulations solely for research purposes, and the authors are not aware of any potential ethical issues arising from the methodology, datasets, or applications presented in this work.

REPRODUCIBILITY STATEMENT

Section 4 presents the experimental protocol and results. Appendices A.1–A.4 specify the datasets, baselines, metrics, and parameter settings, and Appendix A.5 reports additional results. Appendix B provides complete proofs. The source code will be released to facilitate reproduction of the reported results

REFERENCES

Herbert Amann. *Linear and Quasilinear Parabolic Problems. Volume I: Abstract Linear Theory*, volume 89 of *Monographs in Mathematics*. Birkhäuser, 1995. ISBN 978-3-7643-5114-4. doi: 10.1007/978-3-0348-9221-6.

Hassan Arbabi and Igor Mezić. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16 (4):2096–2126, 2017. doi: 10.1137/17M1125236.

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

Steven L. Brunton and J. Nathan Kutz. Promising directions of machine learning for partial differential equations. *Nature Computational Science*, 4:483–494, 2024. doi: 10.1038/s43588-024-00643-2.

Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern Koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, 2022. doi: 10.1137/21M1401243.

Ricardo Buitrago Ruiz, Tanya Marwah, Albert Gu, and Andrej Risteski. On the benefits of memory for modeling time-dependent PDEs. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=o9kqa5K3tB.

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

Junfeng Chen and Kailiang Wu. Deep-OSG: Deep learning of operators in semigroup. *Journal of Computational Physics*, 493:112498, 2023. doi: 10.1016/j.jcp.2023.112498.

Rares Grozavescu, Pengyu Zhang, Etienne Meunier, and Mark Girolami. Koopman autoencoders with continuous-time latent dynamics for fluid dynamics forecasting, 2026. URL https://arxiv.org/abs/2602.02832.

Yue Guo, Milan Korda, Ioannis G. Kevrekidis, and Qianxiao Li. Learning parametric Koopman decompositions for prediction and control. *SIAM Journal on Applied Dynamical Systems*, 24(1): 744–781, 2025. doi: 10.1137/23M1604576.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2 edition, 1993. doi: 10.1007/978-3-540-78862-1.

Wenjian Hao, Bowen Huang, Wei Pan, Di Wu, and Shaoshuai Mou. Deep Koopman learning of nonlinear time-varying systems. *Automatica*, 159:111372, 2024. doi: 10.1016/j.automatica.2023.111372.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12556–12569, 2023.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Benjamin Holzschuh, Qiang Liu, Georg Kohl, and Nils Thuerey. PDE-transformer: Efficient and versatile transformers for physics simulations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23562–23602. PMLR, 2025.

Xuebin Hu, Qinglong Ma, Peizhi Zhao, and Xun Wang. Physics-aware neural operator for high-fidelity fluid dynamics modeling with geometric and spectral priors. *Physics of Fluids*, 37(11):115111, 2025. doi: 10.1063/5.0299765.

Yifan Hu, Fukang Yin, Weimin Zhang, Kaijun Ren, Junqiang Song, and Kefeng Deng. A hybrid framework for global weather forecasting via low-resolution dynamical core and multigrid neural operator. *npj Climate and Atmospheric Science*, 9(1):112, 2026. doi: 10.1038/s41612-026-01374-z.

Pengzhan Jin, Shuai Meng, and Lu Lu. MIONet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022. doi: 10.1137/22M1477751.

Yuhong Jin, Andong Cong, Lei Hou, Qiang Gao, Xiangdong Ge, Chonglong Zhu, Yongzhi Feng, and Jun Li. Invertible koopman neural operator for data-driven modeling of partial differential equations, 2025. URL https://arxiv.org/abs/2503.19717.

Cheng Jing, Uvini Balasuriya Mudiyanselage, Abhishek Verma, Kallol Bera, Shahid Rauf, and Kookjin Lee. Extending fourier neural operators for modeling parameterized and coupled PDEs. In *International Conference on Learning Representations*, 2026.

Taeyoung Kim, Youngsoo Ha, and Myungjoo Kang. Neural operators learn the local physics of magnetohydrodynamics. *Computers & Fluids*, 297:106661, 2025. doi: 10.1016/j.compfluid.2025.106661.

Bernard O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315–318, 1931. doi: 10.1073/pnas.17.5.315.

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.

Wei-Min Lei and Hou-Biao Li. U-WNO: U-Net enhanced wavelet neural operator for solving parametric partial differential equations. *Computers & Mathematics with Applications*, 194:272–287, 2025. doi: 10.1016/j.camwa.2025.06.024.

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

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

Yen Ting Lin, Yifeng Tian, Daniel Livescu, and Marian Anghel. Data-driven learning for the Mori–Zwanzig formalism: A generalization of the Koopman learning framework. *SIAM Journal on Applied Dynamical Systems*, 20(4):2558–2601, 2021. doi: 10.1137/21M1401759.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 67398–67433, 2023. doi: 10.52202/075280-2946.

Siyu Lou, Hao Xu, Wenguan Wang, Lu Lu, Hao Sun, Yang Liu, Linfeng Zhang, Dongxiao Zhang, and Yuntian Chen. Data-driven discovery of governing differential equations across physical systems, 2026. URL https://arxiv.org/abs/2606.09638.

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*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022. doi: 10.1016/j.cma.2022.114778.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9:4950, 2018. doi: 10.1038/s41467-018-07210-0.

Yanfang Lyu, Xiaoyu Zhao, Zhiqiang Gong, Xiao Kang, and Wen Yao. Multi-fidelity prediction of fluid flow based on transfer learning using fourier neural operator. *Physics of Fluids*, 35(7): 077118, 2023. doi: 10.1063/5.0155555.

Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards stability of autoregressive neural operators. *arXiv preprint arXiv:2306.10619*, 2023.

Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Regaldo-Saint Blancard, Francois Rozet, Cristiana Diaconu, Lucas Meyer, Kaze W. K. Wong, Hadi Sotoudeh, Alberto Bietti, Irina Espejo, Rio Fear, Siavash Golkar, Tom Hehir, Keiya Hirashima, Geraud Krawezik, Francois Lanusse, Rudy Morel, Ruben Ohana, Liam Parker, Mariel Pettee, Jeff Shen, Kyunghyun Cho, Miles Cranmer, and Shirley Ho. Walrus: A cross-domain foundation model for continuum dynamics. *arXiv preprint arXiv:2511.15684*, 2025. URL https://arxiv.org/abs/2511.15684.

Seungtae Park, Heejoon Jeon, and Hyung Ju Hwang. Enhancing fourier neural operators with CNNs architectures: Pooling, groupwise convolution and inverted block. *Neurocomputing*, 634:129905, 2025. doi: 10.1016/j.neucom.2025.129905.

Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Generalizing Koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018. doi: 10.1137/16M1062296.

Mahindra Singh Rautela, Alexander Most, Siddharth Mansingh, Bradley C. Love, Ayan Biswas, Diane Oyen, and Earl Lawrence. MORPH: Shape-agnostic pde foundation models. *arXiv preprint arXiv:2509.21670*, 2025.

Patrick A. K. Reinbold, Logan M. Kageorge, Michael F. Schatz, and Roman O. Grigoriev. Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nature Communications*, 12:3219, 2021. doi: 10.1038/s41467-021-23479-0. URL https://www.nature.com/articles/s41467-021-23479-0.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflueger, and Mathias Niepert. PDEBench Datasets. DaRUS, University of Stuttgart, 2022a. Dataset DOI: 10.18419/darus-2986.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflueger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611, 2022b. URL https://arxiv.org/abs/2210.07182.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023. doi: 10.1016/j.cma.2022.115783.

Rui Wang, Yihe Dong, Sercan O. Arik, and Rose Yu. Koopman neural forecaster for time-series with temporal distributional shifts. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kUmdmHxK5N.

Wei Wang, Yuanyuan Gao, and Zhuogeng Fan. KAN-ONets: Improved expressivity and performance of neural operators on diverse grids with Kolmogorov–Arnold networks. *Neurocomputing*, 663:131775, 2026. doi: 10.1016/j.neucom.2025.131775.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-FNO: An enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022. doi: 10.1016/j.advwatres.2022.104180.

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. doi: 10.1007/s00332-015-9258-5.

Zipeng Xiao, Siqi Kou, Zhongkai Hao, Bokai Lin, and Zhijie Deng. Amortized fourier neural operators. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3651.

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. doi: 10.1016/j.jcp.2024.113194.

A Experimental Details

This appendix makes the empirical comparisons reproducible by specifying the datasets and tasks, baselines and evaluation protocol, evaluation metrics, parameter settings, and additional results, in the order in which these elements enter the experimental study. Together, these details define how forecasting accuracy and the measured parameter–error trade-off are computed.

A.1 Datasets and Tasks

The benchmark suite contains one single-step operator-mapping task and three autoregressive 40-step forecasting tasks. The standard benchmarks evaluate spatial operator approximation and long-horizon error accumulation across distinct PDE regimes, while controlled variants of the three temporal tasks further evaluate adaptation to time-varying physical conditions.

Table 3 records the archived filenames and paper-facing protocols. For the standard archived benchmarks, the first N_{train} samples form the training set and the final N_{test} samples form the held-out test set. These fields are loaded as 32-bit floating-point tensors without data augmentation. The controlled time-varying variants are generated independently for training, validation, and testing using disjoint condition schedules.

Table 3: Datasets and task protocols. "History / target" denotes the numbers of observed and forecast fields, and "Split" gives the numbers of training and test samples for the standard archived benchmarks. Standard benchmarks use $\mathbf{c}_n = \mathbf{m}$; the final column reports the prescribed interval condition for the time-varying extension, where available.

Dataset	Archived data file	Prediction task	Grid	History / target	Split	$\mathbf{c}_n^{\mathbf{tv}}$
Burgers	`burgers_data_R10.mat`	Initial condition to solution field at the prescribed target time	256	1/1	1000/200	—
NS 10^{-3}	`ns_V1e-3_N5000_T50.mat`	Autoregressive vorticity forecasting at $\nu = 10^{-3}$	64^2	10/40	1000/200	$(\mathbf{m}, q_n),\ q_n \in [0.7, 1.3]$
NS 10^{-4}	`ns_V1e-4_N10000_T30.mat`	Autoregressive vorticity forecasting at $\nu = 10^{-4}$	64^2	10/40	1000/200	$(\mathbf{m}, q_n),\ q_n \in [0.7, 1.3]$
SWE	`2D_rdb_NA_NA.h5`	Autoregressive radial dam-break water-depth forecasting	128^2	10/40	900/100	$(\mathbf{m}, q_n),\ q_n \in [0.25, 0.75]$

The scalar q_n denotes the forcing amplitude prescribed on $[t_n, t_{n+1}]$ and is held constant within that snapshot interval. Each time-varying condition path is generated from eight randomly sampled knot amplitudes followed by linear interpolation and mild temporal smoothing. For Navier–Stokes, q_n scales the vorticity forcing; for shallow water, it scales the external momentum source.

Burgers. We use the one-dimensional viscous Burgers benchmark employed by FNO and KNO (Li et al., 2021; Xiong et al., 2024). The dynamics satisfy

$$\partial_t u(x,t) + \partial_x \left(\frac{u(x,t)^2}{2} \right) = \nu \, \partial_{xx} u(x,t), \qquad x \in (0,1), \quad t \in (0,1],$$

with periodic boundary conditions and initial condition $u(x,0) = u_0(x)$. The `R10` benchmark corresponds to $\mathrm{Re} = 10$ (equivalently $\nu = 0.1$ under $\mathrm{Re} = 1/\nu$), and the learning task maps $u_0 \mapsto u(\cdot, 1)$. The data are obtained following the access instructions in the official KoopmanLab repository (https://github.com/Koopman-Laboratory/KoopmanLab). The MATLAB fields `a` and `u` contain the initial condition and target solution, respectively. Spatial subsampling by a factor of 32 yields 256 retained grid points. Because this archived benchmark provides initial-to-terminal solution pairs rather than interval-resolved trajectories, we retain it as a fixed-condition single-step task rather than constructing an interval-wise time-varying condition path.

Navier–Stokes. The two-dimensional incompressible Navier–Stokes files are obtained through the same official KoopmanLab data instructions and correspond to the vorticity-form benchmark used by FNO and KNO (Li et al., 2021; Xiong et al., 2024). We evaluate $\nu = 10^{-3}$ and $\nu = 10^{-4}$ separately on 64×64 fields, using 10 observed and 40 forecast frames. Although the 10^{-4} filename contains `T30`, the archived file used for these runs contains 50 temporal samples, allowing the same

10/40 slicing for all four methods. The lower-viscosity setting provides the more demanding long-horizon test.

For experiments with time-varying physical conditions, we construct time-varying-forcing variants of both viscosity regimes,

$$\partial_t \omega + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega + q(t) f_0(x, y),$$

with fixed spatial forcing pattern

$$f_0(x, y) = 0.1 \left[\sin\big(2\pi(x + y)\big) + \cos\big(2\pi(x + y)\big) \right].$$

The viscosity ν remains fixed within each dataset, while the scalar forcing amplitude varies across snapshot intervals. Each trajectory contains 50 state snapshots and 49 interval amplitudes $\{q_n\}_{n=0}^{48}$, with

$$q(t) = q_n, \qquad t \in [t_n, t_{n+1}).$$

The same value q_n is used by the numerical solver throughout all internal integration steps in that snapshot interval. For both viscosity regimes, $q_n \in [0.7, 1.3]$. Training, validation, and in-distribution test trajectories use independently sampled smooth-random condition paths generated from random knot amplitudes, linear interpolation, and mild temporal smoothing.

Shallow-water. We use the PDEBench two-dimensional radial dam-break shallow-water dataset (Takamoto et al., 2022b;a). The dataset is distributed by the DaRUS repository under DOI `10.18419/darus-2986`, and its trajectories are generated by a PyClaw finite-volume solver. We retain the native 128×128 grid, use a temporal interval of 0.01, and expose only the water-depth field h to the learned models. This task evaluates sharp-front propagation and boundary-sensitive long rollouts.

For experiments with time-varying physical conditions, we augment the shallow-water system with a time-varying momentum source,

$$\partial_t \begin{bmatrix} h \\ hu \\ hv \end{bmatrix} + \partial_x \begin{bmatrix} hu \\ hu^2 + \frac{1}{2} g h^2 \\ huv \end{bmatrix} + \partial_y \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2} g h^2 \end{bmatrix} = q(t) \begin{bmatrix} 0 \\ \tau_x(x, y) \\ \tau_y(x, y) \end{bmatrix},$$

where

$$\tau_x(x, y) = \tau_y(x, y) = 0.05 \exp\left(-\frac{x^2 + y^2}{4} \right).$$

Gravity and the sampled initial dam radius remain fixed along each trajectory, while the scalar forcing amplitude varies across snapshot intervals. Each trajectory contains 50 state snapshots and 49 interval amplitudes $\{q_n\}_{n=0}^{48}$, with

$$q(t) = q_n, \qquad t \in [t_n, t_{n+1}).$$

The corresponding value q_n is applied in the PyClaw source update throughout that interval, with $q_n \in [0.25, 0.75]$. Training, validation, and in-distribution test trajectories use independently sampled smooth-random condition paths generated using the same schedule construction as above.

Condition and history context. For all experiments, the propagator generator receives a prescribed interval condition together with a summary of the current rollout history,

$$\widetilde{\mathbf{c}}_n = (\mathbf{c}_n, \mathbf{s}_n), \qquad \mathbf{s}_n = S(\mathbf{h}_n^{\mathrm{roll}}).$$

For the standard fixed-condition benchmarks, $\mathbf{c}_n = \mathbf{m}$, where $\mathbf{m}$ denotes dataset-specific static physical and discretization metadata. For the time-varying Navier–Stokes and shallow-water experiments,

$$\mathbf{c}_n = (\mathbf{m}, q_n).$$

The amplitude q_n is aligned with the same physical interval as the corresponding one-step prediction. In particular, with observed history $(x_0, \ldots, x_9)$, the first forecast x_{10} uses q_9, and the complete 40-step rollout uses $q_9, \ldots, q_{48}$. The history summary is recomputed after each autoregressive update, and the resulting propagator is $\mathbf{K}_\vartheta(\mathbf{k}; \widetilde{\mathbf{c}}_n)$.

A.2 BASELINES

Fixed-condition benchmarks. For the standard fixed-condition benchmarks, we compare against representative neural-operator and Koopman-based methods covering global spectral modeling, multiresolution operator learning, fixed Koopman propagation, and invertible latent dynamics.

- **FNO** (Li et al., 2021): A Fourier-domain neural operator that learns global spectral convolution kernels for mapping between PDE fields. It provides a standard spectral operator-learning reference for both single-step and temporal prediction tasks.
- **AM-FNO** (Xiao et al., 2024): A frequency-amortized Fourier neural operator that generates spectral operator parameters from Fourier coordinates using a shared parameterization, rather than learning each spectral mode independently.
- **KNO** (Xiong et al., 2024): A Koopman neural operator that lifts PDE fields into learned observable coordinates and evolves the retained Fourier modes using fixed mode-wise linear propagators. We follow the official KoopmanLab implementation.
- **IKNO** (Jin et al., 2025): An invertible Koopman-based model that combines invertible residual-coupling transformations with latent linear evolution. It provides an alternative structured latent-dynamics baseline for long-horizon prediction.

Time-varying-condition benchmarks. For the controlled Navier–Stokes and shallow-water benchmarks, we compare against methods that explicitly receive the interval condition $\mathbf{c}_n = (\mathbf{m}, q_n)$ and can therefore adapt their prediction to the prescribed condition at each rollout step.

- **Cond. FNO** (Li et al., 2021): A condition-aware FNO that receives the current field history together with the interval-aligned condition $\mathbf{c}_n$ at each autoregressive step and directly predicts the next field. It represents direct conditional neural operator learning without an explicit condition-dependent latent propagator.
- **Cond. KNO** (Xiong et al., 2024): A condition-aware extension of KNO in which the mode-wise latent propagators depend on the prescribed interval condition $\mathbf{c}_n$. The observable representation remains shared across conditions, providing a controlled reference for condition-dependent Koopman propagation.

All methods within each benchmark group use the same data splits, observed history, prediction horizon, and evaluation protocol whenever applicable. For the time-varying-condition tasks, the condition supplied at each rollout step is aligned with the corresponding physical evolution interval.

A.3 EVALUATION METRICS

We use four complementary metrics to characterize forecasting error at different aggregation scales and spatial frequencies: step relative L_2, full-rollout relative L_2, normalized RMSE (nRMSE), and high-frequency spectral RMSE (fRMSE). The first two measure stepwise and trajectory-level relative error, respectively; nRMSE provides a dataset-level normalized error in physical space, while fRMSE isolates errors in fine-scale spectral content.

For test sample $i \in \{1, \dots, N\}$ and forecast step $t \in \{1, \dots, T\}$, let $\widehat{\mathbf{u}}_{i,t}, \mathbf{u}_{i,t} \in \mathbb{R}^P$ denote the predicted and target fields after flattening spatial locations and physical channels, where P is the number of entries per field. For the single-step Burgers task, $T = 1$.

For the parameter–error comparison in Figure 2, the mean rollout MSE is defined as

$$\mathrm{MSE}_{\mathrm{roll}} = \frac{1}{NTP} \sum_{i=1}^{N} \sum_{t=1}^{T} \|\widehat{\mathbf{u}}_{i,t} - \mathbf{u}_{i,t}\|_2^2 . \tag{9}$$

Step relative L_2. The step metric averages the relative field error over samples and forecast steps:

$$L_2^{\mathrm{Step}} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\|\widehat{\mathbf{u}}_{i,t} - \mathbf{u}_{i,t}\|_2}{\|\mathbf{u}_{i,t}\|_2}. \tag{10}$$

This metric reflects average local forecast accuracy without coupling errors across the prediction horizon.

Full-rollout relative L_2. To evaluate each predicted trajectory as a whole, we aggregate errors over all forecast steps before normalization:

$$L_2^{\text{full}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left(\sum_{t=1}^{T} \|\widehat{\mathbf{u}}_{i,t} - \mathbf{u}_{i,t}\|_2^2\right)^{1/2}}{\left(\sum_{t=1}^{T} \|\mathbf{u}_{i,t}\|_2^2\right)^{1/2}}. \tag{11}$$

Unlike L_2^{Step}, this metric preserves error accumulation over the complete rollout and then averages the trajectory-level relative errors across test samples.

Normalized RMSE. We further report a dataset-level normalized RMSE over the complete test rollout:

$$\text{nRMSE} = \left(\frac{\sum_{i=1}^{N}\sum_{t=1}^{T} \|\widehat{\mathbf{u}}_{i,t} - \mathbf{u}_{i,t}\|_2^2}{\sum_{i=1}^{N}\sum_{t=1}^{T} \|\mathbf{u}_{i,t}\|_2^2} \right)^{1/2}. \tag{12}$$

Unlike L_2^{full}, which normalizes each trajectory before averaging, e_{nRMSE} aggregates prediction and target energy over the entire test set before normalization, providing a global measure of field fidelity.

High-frequency spectral RMSE. To assess fine-scale spatial fidelity, we compute a normalized spectral error over a prescribed high-frequency band. Let $\mathcal{F}$ denote the channel-wise spatial Fourier transform and $\Pi_{\mathcal{H}}$ the projection onto the high-frequency set $\mathcal{H}$:

$$\text{fRMSE} = \left(\frac{\sum_{i=1}^{N}\sum_{t=1}^{T} \|\Pi_{\mathcal{H}}\mathcal{F}\left(\widehat{\mathbf{u}}_{i,t} - \mathbf{u}_{i,t}\right)\|_2^2}{\sum_{i=1}^{N}\sum_{t=1}^{T} \|\Pi_{\mathcal{H}}\mathcal{F}\left(\mathbf{u}_{i,t}\right)\|_2^2} \right)^{1/2}. \tag{13}$$

This metric isolates degradation of high-frequency structure that can be obscured by aggregate physical-space errors. Restricting the metric to $\mathcal{H}$ also avoids redundancy with nRMSE over the full spectrum.

Gradient relative L_2. For the shallow-water benchmark, we additionally evaluate the spatial variation of the predicted field using a gradient relative L_2 error. Let Δ_x and Δ_y denote finite-difference operators along the two spatial directions. We define

$$\text{Grad-}L_2 = \left(\frac{\sum_{i=1}^{N}\sum_{t=1}^{T} \left(\|\Delta_x \hat{u}_{i,t} - \Delta_x u_{i,t}\|_2^2 + \|\Delta_y \hat{u}_{i,t} - \Delta_y u_{i,t}\|_2^2\right)}{\sum_{i=1}^{N}\sum_{t=1}^{T} \left(\|\Delta_x u_{i,t}\|_2^2 + \|\Delta_y u_{i,t}\|_2^2\right)} \right)^{1/2}, \tag{14}$$

where $\hat{u}_{i,t}$ and $u_{i,t}$ are the predicted and reference water-depth fields, respectively. Lower values indicate better agreement in spatial variations over the complete test rollout. Unlike physical-space field errors, this metric emphasizes the fidelity of local gradients and is therefore more sensitive to the location and sharpness of propagating fronts and boundaries. On the time-varying shallow-water benchmark, Cond. FNO, fixed KNO, and PKNO obtain gradient relative L_2 errors of 0.0921, 0.1142, and 0.2013, respectively, further indicating that the current PKNO realization has difficulty preserving sharp spatial variations.

A.4 Implementation and Training Details

This section summarizes the training protocol, architectural settings, dataset-specific optimization hyperparameters, and implementation environment used in the reported experiments.

Training protocol. All models are trained for 500 epochs with random seed 42. We use Adam and decay the learning rate by a factor of 0.5 every 100 epochs. PKNO is trained with the composite objective in Eq. (8), where $\lambda_{\text{pred}} = 5$, $\lambda_{\text{rec}} = 0.5$, and $\lambda_{\text{lat}} = 10^{-2}$. The latent-consistency weight is linearly warmed up from zero to 10^{-2} during the first 10 epochs.

For the temporal tasks, training follows the same autoregressive field-history protocol used at inference time. At each rollout step, the predicted field is appended to the sliding input history before the next prediction is made. Thus, the temporal model is repeatedly re-evaluated from the updated field history rather than propagated through a single encode-once latent trajectory. No additional high-frequency refinement loss is used in the reported core experiments.

Architecture settings. Unless otherwise stated, the observable dimension is 32 and 16 Fourier modes are retained. The Koopman decomposition depth is $D = 8$ for Navier–Stokes and $D = 4$ for shallow water. KNO and PKNO use the same retained-mode resolution in the controlled Koopman comparisons.

The PKNO observable dictionary and propagator-correction networks each use two hidden layers of width 128. The condition and state-summary embeddings have dimensions 128 and 64, respectively. IKNO uses four invertible residual-coupling blocks together with an internal Koopman operator. Dataset-specific optimization settings are reported in Table 4.

Optimization settings. PKNO uses a default weight decay of 10^{-4} for all parameter groups except the state encoder, static-condition encoder, and dynamic-condition encoder, for which weight decay is set to zero. We adopt this parameter-group-specific regularization because uniform weight decay can drive these encoders toward numerical collapse late in training, suppressing their sensitivity to the conditioning variables. All other optimizer settings are unchanged. Gradient clipping is applied only where listed in Table 4.

Table 4: Dataset-specific optimization settings for the reported KNO and PKNO experiments. All configurations use 500 training epochs and random seed 42. PKNO uses zero weight decay for the state and condition encoders and 10^{-4} for all remaining parameter groups.

Dataset	Method	Batch size	Learning rate	Gradient clipping
Burgers	KNO	64	1×10^{-3}	–
	PKNO	64	1×10^{-3}	–
NS 10^{-3}	KNO	10	1×10^{-3}	–
	PKNO	10	5×10^{-4}	1.0
NS 10^{-4}	KNO	10	1×10^{-3}	–
	PKNO	10	5×10^{-4}	1.0
SWE	KNO	5	1×10^{-3}	–
	PKNO	5	5×10^{-5}	0.1

Implementation environment. Experiments are run with Python 3.10.20, PyTorch 2.5.1, CUDA 12.1, and a single NVIDIA RTX A6000 GPU. Complex-valued trainable tensors are counted as two real parameters when reporting model size.

A.5 Additional Results

Shallow-water failure analysis. PKNO is less competitive on shallow water, whose radial dam-break dynamics contain sharp fronts and substantial fine-scale structure. This task also uses a more conservative optimization and propagation configuration than Navier–Stokes, including a smaller learning rate and residual-update scale and a reduced Koopman decomposition depth ($D = 4$ versus $D = 8$). While these choices improve numerical stability, they may restrict the magnitude and expressiveness of the learned propagator corrections.

Table 5 provides a frequency-resolved diagnostic. The performance gap is small at low frequencies, but grows markedly toward finer scales: the high-frequency relative-L_2 error is 0.841 for Cond. FNO, compared with 2.132 for fixed KNO and 2.442 for PKNO. PKNO also exhibits a larger mid-

Table 5: Frequency-resolved diagnostic on the time-varying shallow-water benchmark. Lower is better. KNO is included as a condition-agnostic diagnostic reference.

Method	**Low-freq.** L_2	**Mid-freq.** L_2	**High-freq.** L_2	**Gradient** L_2
Cond. FNO	**0.0157**	**0.1818**	**0.8410**	**0.0921**
KNO	0.0203	0.1843	2.1320	0.1142
PKNO	0.0313	0.4500	2.4419	0.2013

frequency and gradient error. These results suggest that the current retained-mode Koopman configuration has difficulty recovering sharp, fine-scale dam-break dynamics, and that the conservative shallow-water training configuration may further limit the learned corrections. We therefore treat this result as a limitation of the present spectral-propagation and optimization design rather than of condition-dependent Koopman modeling in general.

Computational Cost. To complement the parameter–accuracy comparison in the main paper, we report measured training cost on the two controlled Navier–Stokes tasks in Table 6. All measurements are obtained from the 500-epoch runs under the hardware environment described in Appendix A.4. Training time denotes the accumulated epoch-level training time, and GPU memory denotes the peak allocated memory recorded during training.

Table 6: Measured computational cost on the controlled Navier–Stokes tasks. Training time is measured over 500 epochs on an NVIDIA RTX A6000. GPU memory denotes peak allocated memory.

Dataset	**Method**	**Params (M)**	**Time / epoch (s)**	**Train time (h)**	**Peak memory (GB)**
NS 10^{-3}	Cond. FNO	4.203	39.7	5.51	3.91
	Fixed KNO	0.689	96.6	13.42	14.54
	PKNO	0.949	137.3	19.07	15.11
NS 10^{-4}	Cond. FNO	4.203	43.1	5.98	3.91
	Fixed KNO	0.689	95.3	13.23	14.54
	PKNO	0.949	141.5	19.66	15.11

Relative to fixed KNO, PKNO increases the parameter count by 37.6% and incurs additional training computation due to condition- and history-dependent propagator generation. Across the two Navier–Stokes tasks, training time increases by 42.1–48.6%, whereas peak allocated GPU memory increases by approximately 4%. Thus, PKNO remains a sub-million-parameter model with moderate memory overhead, while its adaptive propagator generation introduces additional training-time cost.

B PROOFS

In this section, we prove Theorems 1 and 2 in turn. The first establishes a shared finite dictionary for condition-indexed target evolution, while the second bounds error accumulation in the closed-loop history-adaptive autoregressive rollout.

Proof sketch for Theorem 1. View the stacked condition-indexed target images as a continuous map into $L^2(\mu;\mathbb{R}^p)$. Compactness yields a finite ε-net whose components, together with the prescribed targets, span a shared finite subspace. Orthogonal projection gives the uniform error bound, while its coordinates vary continuously with the interval condition.

Proof of Theorem 1. Let

$$\mathcal{H} := L^2(\mu), \qquad Y_j(\mathbf{c}) := \mathcal{K}_{\Delta t}[\mathbf{c}]g_j,$$

and define

$$\mathbf{Y}(\mathbf{c}) := \big(Y_1(\mathbf{c}), \dots, Y_p(\mathbf{c})\big) \in \mathcal{H}^p = L^2(\mu;\mathbb{R}^p).$$

We first show that $\mathbf{Y}$ is continuous. Let $\mathbf{c}_n \to \mathbf{c}$ in $\mathcal{C}_{\Delta t}$. By the assumed continuity of the history update with respect to the condition, for μ-almost every $\mathbf{h}$,

$$\mathbf{H}_{\Delta t}[\mathbf{c}_n](\mathbf{h}) \longrightarrow \mathbf{H}_{\Delta t}[\mathbf{c}](\mathbf{h}).$$

Since each g_j is continuous,

$$g_j(\mathbf{H}_{\Delta t}[\mathbf{c}_n](\mathbf{h})) \longrightarrow g_j(\mathbf{H}_{\Delta t}[\mathbf{c}](\mathbf{h}))$$

for μ-almost every $\mathbf{h}$. Moreover,

$$\left|g_j(\mathbf{H}_{\Delta t}[\mathbf{c}_n](\mathbf{h})) - g_j(\mathbf{H}_{\Delta t}[\mathbf{c}](\mathbf{h}))\right|^2 \le 4\|g_j\|_\infty^2.$$

Because μ is finite, the right-hand side is integrable. Hence the dominated convergence theorem gives

$$\|Y_j(\mathbf{c}_n) - Y_j(\mathbf{c})\|_{\mathcal{H}} \longrightarrow 0.$$

Since the number of targets is finite,

$$\|\mathbf{Y}(\mathbf{c}_n) - \mathbf{Y}(\mathbf{c})\|_{\mathcal{H}^p}^2 = \sum_{j=1}^{p} \|Y_j(\mathbf{c}_n) - Y_j(\mathbf{c})\|_{\mathcal{H}}^2 \longrightarrow 0.$$

Thus $\mathbf{Y} : \mathcal{C}_{\Delta t} \to \mathcal{H}^p$ is continuous.

Since $\mathcal{C}_{\Delta t}$ is compact,

$$\mathcal{Q} := \mathbf{Y}(\mathcal{C}_{\Delta t}) \subset \mathcal{H}^p$$

is compact. Therefore, for the prescribed $\varepsilon > 0$, there exist finitely many elements

$$\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(M)} \in \mathcal{Q}, \qquad \mathbf{v}^{(q)} = \big(v_1^{(q)}, \dots, v_p^{(q)}\big),$$

such that for every $\mathbf{c} \in \mathcal{C}_{\Delta t}$ there is an index $q = q(\mathbf{c})$ satisfying

$$\|\mathbf{Y}(\mathbf{c}) - \mathbf{v}^{(q)}\|_{\mathcal{H}^p} \le \varepsilon.$$

Define the condition-independent finite-dimensional subspace

$$V := \operatorname{span}\Big(\{g_1, \dots, g_p\} \cup \{v_j^{(q)} : 1 \le j \le p,\ 1 \le q \le M\}\Big) \subset \mathcal{H}.$$

Let $P_V : \mathcal{H} \to V$ be the orthogonal projection, and let $\mathbf{P}_V : \mathcal{H}^p \to V^p$ denote its componentwise extension. Because every $\mathbf{v}^{(q)}$ belongs to V^p, the best-approximation property of orthogonal projection gives

$$\begin{aligned}\|\mathbf{Y}(\mathbf{c}) - \mathbf{P}_V\mathbf{Y}(\mathbf{c})\|_{\mathcal{H}^p} &= \operatorname{dist}_{\mathcal{H}^p}\big(\mathbf{Y}(\mathbf{c}), V^p\big) \\ &\le \|\mathbf{Y}(\mathbf{c}) - \mathbf{v}^{(q(\mathbf{c}))}\|_{\mathcal{H}^p} \\ &\le \varepsilon.\end{aligned}$$

Consequently,

$$\sup_{\mathbf{c} \in \mathcal{C}_{\Delta t}} \|\mathbf{Y}(\mathbf{c}) - \mathbf{P}_V\mathbf{Y}(\mathbf{c})\|_{\mathcal{H}^p} \le \varepsilon.$$

Let $e_1, \dots, e_m$ be an orthonormal basis of V, and define

$$\mathbf{\Psi} := (g_1, \dots, g_p, e_1, \dots, e_m)^\top.$$

Then $N_\Psi = p + m \ge p$, every entry of $\mathbf{\Psi}$ belongs to $L^2(\mu)$, and $\psi_j = g_j$ for $j = 1, \dots, p$.

Define

$$\mathbf{A} : \mathcal{C}_{\Delta t} \longrightarrow \mathbb{R}^{p \times (p+m)}$$

by

$$A_{j,\ell}(\mathbf{c}) := 0, \qquad 1 \le \ell \le p,$$

and

$$A_{j,p+r}(\mathbf{c}) := \langle Y_j(\mathbf{c}), e_r \rangle_{\mathcal{H}}, \qquad 1 \le r \le m.$$

It follows that

$$\big(\mathbf{A}(\mathbf{c})\mathbf{\Psi}\big)_j = \sum_{r=1}^{m} \langle Y_j(\mathbf{c}), e_r\rangle_{\mathcal{H}} e_r$$
$$= P_V Y_j(\mathbf{c}).$$

Furthermore, $Y_j(\mathbf{c})$ is continuous in $\mathcal{H}$, so every coefficient

$$\mathbf{c} \longmapsto \langle Y_j(\mathbf{c}), e_r\rangle_{\mathcal{H}}$$

is continuous. Since $\mathbf{A}$ has finitely many entries, $\mathbf{A}$ is continuous.

Finally,

$$\mathbf{P}_V \mathbf{Y}(\mathbf{c}) = \mathbf{A}(\mathbf{c})\mathbf{\Psi}.$$

Combining this identity with the uniform projection estimate yields

$$\sup_{\mathbf{c}\in\mathcal{C}_{\Delta t}} \left\| \begin{bmatrix} \mathcal{K}_{\Delta t}[\mathbf{c}]g_1 \\ \vdots \\ \mathcal{K}_{\Delta t}[\mathbf{c}]g_p \end{bmatrix} - \mathbf{A}(\mathbf{c})\mathbf{\Psi} \right\|_{L^2(\mu;\mathbb{R}^p)} \le \varepsilon,$$

which proves the result. □

Before proving Theorem 2, we connect its abstract constants $\varepsilon_{\mathrm{loc}}$ and Γ to the PKNO components. Proposition B.1 derives the component-wise bound for Γ; the following residual estimate bounds $\varepsilon_{\mathrm{loc}}$, and the subsequent decomposition identifies the history-feedback contribution.

For the component-wise sensitivity bound, let

$$Z := \bigoplus_{\mathbf{k}\in Z_{k_{\max}}} \mathbb{C}^{p_z}, \qquad \|\widehat{\mathbf{v}}\|_Z := \left(\sum_{\mathbf{k}\in Z_{k_{\max}}} \|\widehat{\mathbf{v}}(\mathbf{k})\|_2^2 \right)^{1/2},$$

where p_z is the number of shared latent observable channels defined in Section 3.2, and define

$$\mathcal{E}_\eta(\mathbf{h}) := \Pi_{k_{\max}} \mathcal{F}\mathbf{g}_\eta(\mathbf{h}), \qquad \mathcal{D}_\omega(\widehat{\mathbf{v}}) := \boldsymbol{\psi}_\omega(\mathcal{F}^{-1}\Pi_{k_{\max}}\widehat{\mathbf{v}}).$$

For $\widetilde{\mathbf{c}} = (\mathbf{c}, \mathbf{s})$, let $(\mathsf{K}(\mathbf{c},\mathbf{s})\widehat{\mathbf{v}})(\mathbf{k}) := \mathbf{K}_\vartheta(\mathbf{k};\widetilde{\mathbf{c}})\widehat{\mathbf{v}}(\mathbf{k})$, and set $Q_{\mathbf{c}}(\mathbf{h}) := \mathsf{K}(\mathbf{c}, S(\mathbf{h}))$. Thus, $\mathcal{P}_{\mathbf{c}}(\mathbf{h}) = \mathcal{D}_\omega(Q_{\mathbf{c}}(\mathbf{h})\mathcal{E}_\eta(\mathbf{h}))$.

Proposition B.1 (Component-wise bound on rollout sensitivity)**.** *Suppose that, uniformly over* $\mathbf{c} \in \mathcal{C}_{\Delta t}$ *and on the rollout region,*

$$\|\mathcal{E}_\eta(\mathbf{h}) - \mathcal{E}_\eta(\widetilde{\mathbf{h}})\|_Z \le L_E \|\mathbf{h} - \widetilde{\mathbf{h}}\|_{\mathrm{hist}},$$
$$\|Q_{\mathbf{c}}(\mathbf{h}) - Q_{\mathbf{c}}(\widetilde{\mathbf{h}})\|_{Z\to Z} \le L_Q \|\mathbf{h} - \widetilde{\mathbf{h}}\|_{\mathrm{hist}},$$

with $\|Q_{\mathbf{c}}(\mathbf{h})\|_{Z\to Z} \le \kappa$ *and* $\|\mathcal{E}_\eta(\mathbf{h})\|_Z \le B_z$*. Assume also that* $\mathcal{D}_\omega$ *is* L_D*-Lipschitz from* Z *to* $\mathcal{X}$*. Then*

$$\|\mathcal{P}_{\mathbf{c}}(\mathbf{h}) - \mathcal{P}_{\mathbf{c}}(\widetilde{\mathbf{h}})\|_{\mathcal{X}} \le \Gamma_{\mathrm{suff}} \|\mathbf{h} - \widetilde{\mathbf{h}}\|_{\mathrm{hist}}, \qquad \Gamma_{\mathrm{suff}} := L_D\big(\kappa L_E + B_z L_Q\big).$$

Proof. Add and subtract $Q_{\mathbf{c}}(\mathbf{h})\mathcal{E}_\eta(\widetilde{\mathbf{h}})$ before applying the decoder Lipschitz bound. Then

$$\|\mathcal{P}_{\mathbf{c}}(\mathbf{h}) - \mathcal{P}_{\mathbf{c}}(\widetilde{\mathbf{h}})\|_{\mathcal{X}} \le L_D\big(\kappa L_E + B_z L_Q\big)\|\mathbf{h} - \widetilde{\mathbf{h}}\|_{\mathrm{hist}}.$$

□

The latent one-step residual below is the pointwise counterpart of the residual penalized by Eq. (7). Along the reference trajectory, suppose the reconstruction and latent residuals satisfy

$$\|\mathbf{x}_{j+1} - \mathcal{D}_\omega(\mathcal{E}_\eta(\mathbf{h}_{j+1}))\|_{\mathcal{X}} \le \varepsilon_{\mathrm{rec}},$$
$$\|\mathcal{E}_\eta(\mathbf{h}_{j+1}) - Q_{\mathbf{c}_j}(\mathbf{h}_j)\mathcal{E}_\eta(\mathbf{h}_j)\|_Z \le \varepsilon_{\mathrm{cl}}$$

imply the local predictor bound

$$\varepsilon_{\mathrm{loc}} \le \varepsilon_{\mathrm{rec}} + L_D \varepsilon_{\mathrm{cl}}.$$

If S is L_S-Lipschitz and $\mathsf{K}(\mathbf{c},\cdot)$ is uniformly L_{K}-Lipschitz over $\mathbf{c} \in \mathcal{C}_{\Delta t}$, then $L_Q \le L_{\mathsf{K}} L_S$, giving

$$\Gamma_{\mathrm{suff}} \le L_D \kappa L_E + L_D B_z L_{\mathsf{K}} L_S.$$

This decomposition separates sensitivity transmitted through the encoded history from that introduced by history-dependent propagator generation.

Proof sketch for Theorem 2. The local one-step bound controls the error evaluated on the reference history, while uniform closed-loop history sensitivity controls the additional error caused by replacing that history with the rolling predicted history. Their sum gives a scalar recurrence for the rollout error, and induction yields the stated accumulation factor $G_r(\Gamma)$.

Proof of Theorem 2. Let $e_j := \|\mathbf{x}_j - \mathbf{x}_{L,j}^{\mathrm{pred}}\|_{\mathcal{X}}$, with zero errors in the exact initial history. Applying the same prescribed condition $\mathbf{c}_j$ to the reference and rolling histories, the local one-step error bound and history-sensitivity assumption imply

$$e_{j+1} \le \varepsilon_{\mathrm{loc}} + \Gamma \max_{0 \le \ell < N_h} e_{j-\ell}.$$

Since $G_q(\Gamma)$ is nondecreasing in q, induction yields $e_{n+r} \le \varepsilon_{\mathrm{loc}} G_r(\Gamma)$ for every $r \ge 1$. □

C Related Work

PKNO lies at the intersection of neural operators and Koopman operator learning. We organize prior work around these two lines to position the specific combination addressed here: a condition-dependent Koopman family acting in shared field-level observable coordinates.

Neural Operators. Neural operators learn function-space maps for reusable PDE surrogates (Kovachki et al., 2023). DeepONet uses branch–trunk networks (Lu et al., 2021), whereas FNO learns spectral kernels that transfer across discretizations (Li et al., 2021); their relative advantages vary across regimes (Lu et al., 2022). Subsequent work has explored localized wavelets, attention-based operator architectures, U-shaped spectral paths, learned geometric deformations, and amortized spectral kernels (Tripura & Chakraborty, 2023; Cao, 2021; Hao et al., 2023; Wen et al., 2022; Li et al., 2023; Xiao et al., 2024). Recent PDE foundation models extend this direction toward broad pretraining and cross-equation transfer. Poseidon uses a multiscale operator transformer pretrained on fluid-dynamics PDEs and adapts to diverse downstream equations (Herde et al., 2024); PDE-Transformer develops a scalable multiscale transformer with channel-wise spatiotemporal tokenization for joint learning across many PDE systems (Holzschuh et al., 2025). MORPH targets heterogeneous 1D–3D, multi-field PDE data through shape-agnostic autoregressive pretraining (Rautela et al., 2025), while Walrus further scales transformer pretraining to cross-domain continuum dynamics spanning multiple physical domains (McCabe et al., 2025). These foundation models emphasize transferable representations across equations and datasets, whereas PKNO focuses on explicitly parameterizing how interval conditions alter the latent evolution law within shared observable coordinates. *Difference compared to existing work:* Existing neural operators and PDE foundation models primarily learn condition-aware predictors or transferable representations. PKNO instead models condition dependence at the evolution-law level through an ordered family of linear propagators acting in shared observable coordinates.

Koopman Operator Learning. Extended dynamic mode decomposition approximates Koopman evolution in a prescribed finite dictionary (Williams et al., 2015), whereas deep Koopman models learn observables jointly with finite-dimensional linear dynamics (Lusch et al., 2018). KNO combines this viewpoint with neural-operator field representations for mesh-independent PDE forecasting (Xiong et al., 2024). Controlled and time-varying Koopman models address inputs, distribution shifts, or online-varying embeddings (Proctor et al., 2018; Wang et al., 2023; Hao et al., 2024). Most closely, PK-NN learns a shared dictionary and parameter-dependent Koopman family for static or time-varying parameters (Guo et al., 2025). Continuous-time Koopman autoencoders allow variable propagation intervals (Grozavescu et al., 2026). *Difference compared to existing work:* PKNO parameterizes mode-wise latent Koopman evolution in shared spatial coordinates, allowing noncommuting propagators to compose without coordinate changes. In the framework of Lou et al. (2026), this is condition-dependent parameterization, not symbolic equation recovery, portable-mechanism discovery, or identification of irreducible variation. Distilling explicit parameter laws or closure relations from the learned propagator remains future work.

D SPATIOTEMPORAL EXTENSIONS

Beyond the core condition-dependent propagation, PKNO supports optional extensions for recovering omitted spatial frequencies and handling variable prediction intervals. Both retain the shared observable representation used by the discrete core.

D.1 HIGH-FREQUENCY REFINEMENT

PKNO can augment the decoded core prediction with an output-only high-frequency correction inspired by the high-frequency complement in KNO (Xiong et al., 2024). Let $\mathbf{x}^{\mathrm{pred}}_{L,n+1} := \boldsymbol{\psi}_\omega(\mathcal{F}^{-1}\Pi_{k_{\max}}\widehat{\mathbf{z}}^{\mathrm{pred}}_{n+1})$ denote the decoded core prediction. The optional corrected prediction is

$$\mathbf{x}^{\mathrm{pred}}_{n+1} = \mathbf{x}^{\mathrm{pred}}_{L,n+1} + \Pi_H U_\rho(\mathbf{h}_n, \mathbf{c}_n, \mathbf{x}^{\mathrm{pred}}_{L,n+1}),$$

where U_ρ is a trainable U-Net and Π_H retains frequencies outside the low-frequency band. The refinement acts only on the decoded output and therefore leaves the latent Koopman recurrence unchanged. Table 1 reported PKNO results do not use this branch

D.2 CONTINUOUS-TIME GENERATOR

For variable prediction intervals, PKNO replaces the discrete propagator with a mode- and condition-dependent generator while retaining the shared encoder and decoder. Ordered composition then has a continuous-time counterpart when physical conditions vary within a prediction interval. The resulting transition matrices evolve in the same shared observable coordinates. For each retained mode, $\mathbf{L}_\vartheta(\mathbf{k}; \mathbf{c}(t)) \in \mathbb{C}^{p_z \times p_z}$ evolves the shared spectral observable according to

$$\partial_t \widehat{\mathbf{z}}(\mathbf{k}, t) = \mathbf{L}_\vartheta(\mathbf{k}; \mathbf{c}(t))\widehat{\mathbf{z}}(\mathbf{k}, t).$$

The generator is produced from the Fourier mode and current condition, yielding mode- and condition-dependent evolution in the shared coordinates. For $s \le t$, let $\boldsymbol{\Phi}_\vartheta(\mathbf{k}; t, s)$ denote the finite-time transition matrix defined by

$$\begin{aligned} \partial_t \boldsymbol{\Phi}_\vartheta(\mathbf{k}; t, s) &= \mathbf{L}_\vartheta(\mathbf{k}; \mathbf{c}(t))\boldsymbol{\Phi}_\vartheta(\mathbf{k}; t, s), \\ \boldsymbol{\Phi}_\vartheta(\mathbf{k}; s, s) &= \mathbf{I}_{p_z}, \qquad \widehat{\mathbf{z}}^{\mathrm{pred}}(\mathbf{k}, t) = \boldsymbol{\Phi}_\vartheta(\mathbf{k}; t, s)\widehat{\mathbf{z}}(\mathbf{k}, s). \end{aligned} \tag{15}$$

If the condition and generator are constant over $[s, t]$, then $\boldsymbol{\Phi}_\vartheta(\mathbf{k}; t, s) = \exp((t-s)\mathbf{L}_\vartheta(\mathbf{k}; \mathbf{c}))$. For time-varying conditions, Eq. (15) is integrated while reevaluating $\mathbf{c}(t)$. The encoder and decoder remain shared with the discrete PKNO, and Hermitian symmetry is enforced as in the discrete model. This optional construction does not specify a continuous-time history-summary feedback mechanism and is outside the discrete rollout theorem.