Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Neural operator discovery from heterogeneous trajectories

Sili Deng

silideng@mit.edu

Massachusetts Institute of Technology https://orcid.org/0000-0002-3421-7414

Zituo Chen

Massachusetts Institute of Technology https://orcid.org/0009-0006-3156-1857

Qiaofeng Li

Zhejiang University

Jiaxin Hu

Zhejiang University

Article

Keywords: Neural operator, Operator discovery, Dynamical systems, Self-supervised learning, Representation learning

Posted Date: July 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10108057/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

Neural operator discovery from heterogeneous trajectories

Zituo Chen[1†], Qiaofeng Li[1,2†], Jiaxin Hu[2], Sili Deng[1*]

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, 02139, MA, USA.
[2]State Key Laboratory of Fluid Power & Mechatronic Systems, Zhejiang University, 866 Yuhangtang Road, Hangzhou, 310058, Zhejiang Province, China.

*Corresponding author(s). E-mail(s): silideng@mit.edu;
†These authors contributed equally to this work.

Abstract

Neural operators provide data-driven mappings for modeling dynamical systems. Extending them to families of systems typically requires explicit conditioning variables such as physical parameters, geometries, or boundary conditions. In many real-world settings, these quantities are unobserved. Here, we formulate neural operator discovery (NOD) as the problem of learning both shared solution operators and system-specific variation directly from heterogeneous trajectories without access to labeled governing factors. We introduce a factorized latent-conditioning formulation that jointly learns a neural operator and a low-dimensional latent representation through factorized prediction, trajectory-decoupled sampling, and dimension selection. Across diverse systems, the learned latent representation captures the intrinsic dimensionality of system variation and organizes system instances in a smooth and approximately invertible latent structure aligned with the underlying governing factors. This organization enables generalization to previously unseen system instances, including zero-shot extrapolation across regimes and stable long-horizon prediction. These results establish an interpretable paradigm for operator learning in the absence of explicit factor supervision.

Keywords: Neural operator, Operator discovery, Dynamical systems, Self-supervised learning, Representation learning

Across the physical sciences, trajectory data are often collected from systems that share a common governing law but differ across experimental conditions, geometries, or boundary settings. In many experimental datasets and legacy simulation archives, the factors that distinguish individual system instances are unobserved, unrecorded, or not directly measurable. The resulting data therefore consist of heterogeneous trajectories generated by multiple underlying configurations without explicit annotation of the variables driving their variation. A central challenge is to learn predictive models that capture both the shared dynamics of a system family and the differences between system instances directly from trajectory data, without access to labeled governing factors.

One established approach to this challenge is symbolic regression, which seeks to recover governing equations together with system-specific coefficients directly from trajectory data without requiring factor labels [1–3]. However, its expressivity is constrained by the chosen primitive library, while computational cost grows rapidly with system complexity (Supplementary Section S3.6). In practice, these limitations restrict symbolic regression largely to low-dimensional systems and hinder its application to high-dimensional spatiotemporal dynamics governed by partial differential equations.

Neural operator learning, by contrast, provides fast and accurate data-driven surrogates for such spatiotemporal dynamics [4–7]. Prevailing formulations either model a single dynamical system or, when extended to families, learn the solution operator in a supervised manner with factors as explicit inputs (*parametric neural operators*, PNO) [8–11]. This assumption is violated in the setting considered here, where the factors distinguishing system instances are unobserved.

Alternative approaches partially address this limitation. Foundation-model-style surrogates (*autoregressive neural operators*, ANO) [12–15] avoid explicit parameter supervision by conditioning prediction on long histories of observed states, referred to here as the *past-state window*. However, these representations are high-dimensional and unstructured with respect to the underlying governing factors. As a result, autoregressive conditioning accumulates rollout error during self-prediction [16, 17] and does not yield an identifiable representation of system variation.

A separate line of work infers latent representations from trajectory data and uses them to condition prediction. Existing approaches, however, trade off expressivity, scalability, or interpretability. Variational autoencoders compress low-complexity dynamics into latent representation under Gaussian priors [18], restricting the capacity of the learned solution operator. Gradient-based meta-learning methods recover system descriptors through inner-loop adaptation [19], but backpropagation through the inner loop becomes prohibitively expensive for high-dimensional spatiotemporal systems.

We observe that both the past-state window used in ANO and the inferred latent representation used in latent-conditioned models serve the same role: encoding the system-specific variation underlying a shared solution operator. Motivated by this perspective, we formulate the problem of learning both a shared solution operator and system-specific variation from heterogeneous trajectories without parameter supervision as **Neural Operator Discovery** (NOD).

To address this problem, we consider a factorized realization consisting of two components (Fig. 1a): a trajectory encoder f_ϕ, which maps a past-state window from a system instance to a low-dimensional latent representation $\mathbf{c}$, and a state-evolution operator g_θ, which predicts future states conditioned on this representation. Two complementary constraints are imposed on $\mathbf{c}$. First, the representation is restricted to be low-dimensional, with its dimension selected as the smallest value sufficient to capture variation across system instances. Second, the conditioning trajectory and prediction trajectory are sampled from different initial conditions of the same system instance (Fig. 1b), encouraging the encoder to retain only cross-trajectory invariants. Together, these constraints promote latent representations that capture governing factors while suppressing trajectory-specific features.

Under this formulation, the learned representations organize system instances in a structured latent space $\mathcal{C}$ that reflects the underlying physical variation. Across multiple systems, this latent space exhibits a smooth and approximately invertible correspondence with the true governing factor space $\mathcal{P}$, despite being learned without parameter supervision. The resulting model supports both accurate long-horizon prediction and generalization to previously unseen system instances. In particular, new systems can be handled either by inferring the latent representation from a single observed trajectory (one-shot inference) or by sampling directly in the latent space to obtain the corresponding latent representation $\mathbf{c}$ (zero-shot inference), as illustrated in Fig. 1b.

Results

This section first illustrates the NODnet structure, then validates it on three families of partial differential equations: the Burgers' equation with varying viscosity, which tests the capture of sharp, regime-dependent dynamics under shocks of $\mathcal{O}(\nu)$ width; the cylinder flow wake on an unstructured mesh, which tests scalability under sparse supervision; and the FitzHugh–Nagumo reaction–diffusion system with two independent kinetic factors, which tests recovery of multi-factor variation. Four additional case studies in the Supplementary Information extend the evaluation to geometric, boundary-condition, and resolution variation, spanning convolutional, recurrent, Fourier Neural Operator, and Neural-ODE implementations of f_ϕ and g_θ (Supplementary Table S1).

Method overview

We consider families of dynamical systems governed by a shared equation form but differing in system-specific factors such as physical coefficients, boundary conditions, or geometric parameters. Let $p_i \in \mathcal{P}$ denote the factors associated with system instance i, where $\mathcal{P} \subset \mathbb{R}^d$ is the underlying factor space. The dynamics evolve according to an operator

$$\frac{\partial \mathbf{x}}{\partial t} = \mathcal{G}(\mathbf{x}_t; p_i), \tag{1}$$

where the governing operator $\mathcal{G}$ and the factors p_i are both unknown.

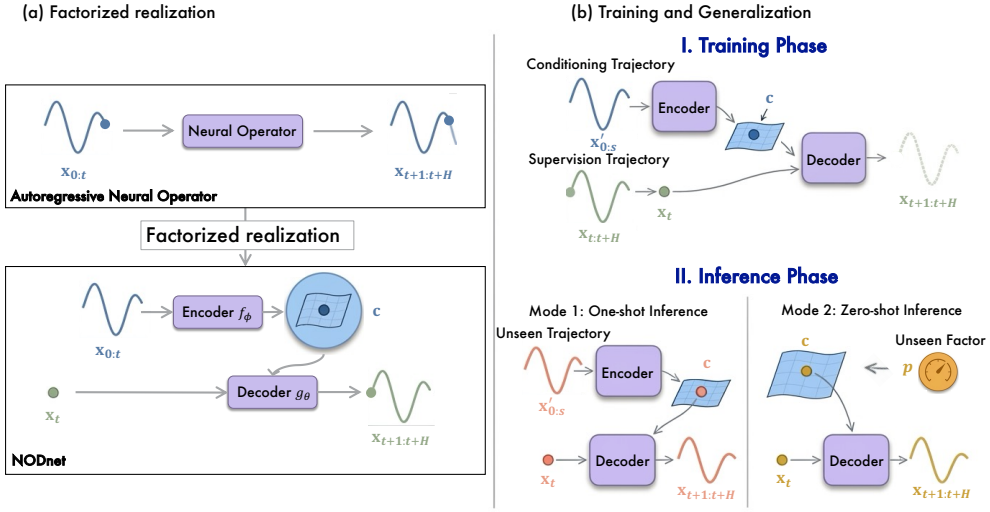


Fig. 1 Neural Operator Discovery. (a) Comparison between conventional history-conditioned neural operators and Neural Operator Discovery. Conventional autoregressive neural operators condition prediction directly on a past-state trajectory window $\mathbf{x}_{0:t}$. In NODnet, the trajectory window is first mapped to a low-dimensional latent representation $\mathbf{c}$ through an encoder f_ϕ, and a conditional decoder g_θ then predicts future dynamics from initial state $\mathbf{x}_t$ conditioned on $\mathbf{c}$. (b) Training and inference with NODnet. During training, trajectory-decoupled sampling pairs two trajectories $\mathbf{x}'_{0:s}$ and $\mathbf{x}_{t:t+H}$ from the same system instance but different initial conditions, using one trajectory for latent inference and the other for supervision. During inference, prediction can proceed either by inferring the latent representation from an observed trajectory (one-shot inference) or by directly sampling the latent space (zero-shot inference). Both modes generalize to system instances unseen during training, denoted as "unseen trajectory" in one-shot inference and "unseen factor" in zero-shot inference.

Each system instance generates trajectories $\tau = \mathbf{x}_{0:T}$ on the computation domain Ω with a time window $[0, T]$, yielding a heterogeneous dataset $\mathcal{D} = \tau_{ij}$, where i denotes system instance, j denotes trajectory initial state $\mathbf{x}_0$. In the setting considered here, only the trajectories and their grouping by system instance are available; the governing factors themselves are unobserved. Importantly, the grouping information does not directly specify the underlying factors. For example, a two-dimensional factor space sampled at five values along each dimension produces 25 distinct system instances, while the observed data contain only trajectories grouped by instance rather than explicit factor coordinates.

Our goal is to learn both a shared solution operator in discrete form,

$$\mathbf{x}_{t+1} = \mathcal{G}'(\mathbf{x}_t; p_i), \tag{2}$$

and a low-dimensional latent representation that captures the underlying system-specific variation. To achieve this, our formulation combines three elements: factorized prediction, trajectory-decoupled sampling, and dimension selection.

Factorized prediction.

Conventional autoregressive neural operators predict future states directly from a window of past observations:

$$\mathbf{x}_{t+1} = \mathcal{G}'_{\psi}(\mathbf{x}_t; \mathbf{x}_{0:t}), \tag{3}$$

where the past-state window $\mathbf{x}_{0:t}$ implicitly carries both dynamical information and system-specific variation. In NODnet, we instead factorize prediction into two components: a latent representation inferred from past trajectories, and a conditional state-evolution operator:

$$\mathbf{c} = f_{\phi}(\mathbf{x}_{0:t}), \hat{\mathbf{x}}_{t+1} = g_{\theta}(\mathbf{x}_t; \mathbf{c}). \tag{4}$$

Here, the encoder f_{ϕ} maps a trajectory segment to a low-dimensional latent representation $\mathbf{c}$ that captures system-specific variation, while the decoder g_{θ} predicts future states conditioned on this representation. Training minimizes mean-squared error of prediction:

$$\mathcal{L} = \|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|_2. \tag{5}$$

Trajectory-decoupled sampling.

To prevent the latent representation from encoding trajectory-specific initial-condition information, we train the encoder and decoder using trajectories generated from different initial conditions of the same system instance. Specifically, the encoder infers a latent representation

$$\mathbf{c} = f_{\phi}(\mathbf{x}'_{0:s}), \tag{6}$$

from a conditioning trajectory $\mathbf{x}'_{0:s}$, while the decoder predicts the evolution of a separate trajectory $\mathbf{x}_{t:t+1}$ from the same system instance:

$$\hat{\mathbf{x}}_{t+1} = g_{\theta}(\mathbf{x}_t; \mathbf{c}). \tag{7}$$

Because the conditioning and prediction trajectories share governing factors but differ in initial conditions, the encoder is encouraged to retain only cross-trajectory invariants rather than transient trajectory-specific features.

Dimension selection.

To ensure that the latent representation captures only system-specific variation, we select its dimension as the smallest value sufficient to explain differences across system instances. The selection follows a two-stage procedure over a sweep of representation dimensions $d = 0, 1, 2, \ldots$. In the first stage, candidate dimensions are identified from the training-loss curve as the smallest dimensions beyond which increasing d no longer yields improvements exceeding the variability across random seeds. In the second stage, the final dimension d^* is selected as the smallest candidate whose corresponding $d{+}1$ model does not produce additional factor-aligned structure in the learned latent space, assessed using principal component analysis of the per-instance latent representations. Applied to the three main case studies, this procedure yields

$d^* = (1, 1, 2)$, matching the number of independent governing factors in each system family. Full details, threshold criteria, seed configurations, and latent-space analysis are provided in Supplementary Sections S2.1–S2.3.

Evaluation overview

Inference modes of NODnet.

Once trained, the learned latent space organizes system instances according to their underlying physical variation, enabling two modes of inference. In zero-shot inference, a latent representation is sampled directly from the learned latent space and used to condition the decoder for prediction. In one-shot inference, the latent representation is first inferred from a short trajectory segment using the encoder, after which prediction proceeds using the same conditional decoder.

For Burgers' equation and cylinder flow wake, we evaluate one-shot inference, where the latent representation must be inferred from observed trajectories. For the FitzHugh–Nagumo reaction–diffusion system, we evaluate both one-shot and zero-shot inference to examine the structure and generalization properties of the learned latent space.

Baselines and fair comparison.

We compare against two baselines that isolate different conditioning strategies while maintaining matched model capacity within each case study. The autoregressive baseline (ANO) conditions predictions directly on a window of past states, without learning an explicit latent representation of system instances. The single-system baseline (SNO) removes the encoder entirely, corresponding to a shared neural operator trained across all trajectories without distinguishing between instances. The predictive capacity of SNO is matched to the decoder component of our model, while ANO is matched to the full model capacity. These baselines isolate the contribution of learning a low-dimensional latent representation for system-specific variation, independent of architectural scaling. A parameter-supervised reference (PNO), in which the true governing factors are provided as input, is reported in Supplementary Section S2.4 as a non-comparable upper-bound benchmark.

Burgers' Equation

As a one-dimensional special case of the Navier–Stokes equations, Burgers' equation describes the interplay between nonlinear convection and viscous diffusion. As viscosity decreases, solutions develop shock fronts of $\mathcal{O}(\nu)$ width, producing sharp regime-dependent dynamics that are difficult to resolve. The governing equation is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \tag{8}$$

where u is the fluid velocity and ν is the kinematic viscosity, the only varying factor across the system family.

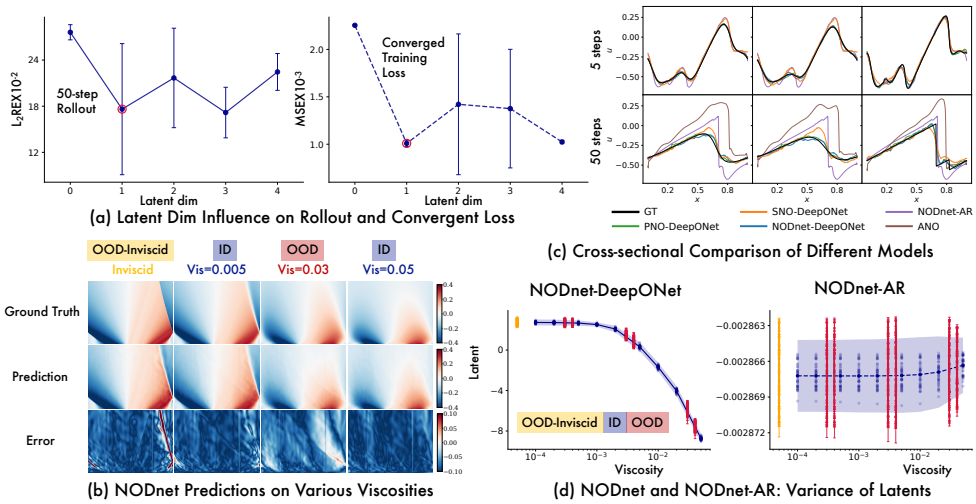


Fig. 2 Burgers' equation results. (a) Dimension selection. The 50-step rollout error (left, L_2RE $\times 10^{-2}$) and converged training MSE (right, $\times 10^{-3}$) are evaluated across latent dimensions and three random seeds. The dimension-selection procedure identifies d^*=1, consistent with viscosity being the only varying governing factor (Supplementary Section S2.1). (b) Predictions across viscosity regimes, including the inviscid limit (ν=0), in-distribution viscosities (ν=0.005, 0.05), and an unseen viscosity within the training range (ν=0.03). Rows show ground truth, prediction, and absolute error. (c) Comparison of long-horizon prediction profiles for the parameter-supervised upper bound (PNO), the single-system baseline (SNO), the latent-conditioned model (NODnet), the autoregressive latent-conditioned variant (NODnet-AR), and the autoregressive baseline (ANO). Results are shown at rollout horizons H=5 (top) and H=50 (bottom). (d) Relationship between the learned latent representation and viscosity. The full-field predictor learns a smooth monotonic latent organization extending continuously to the inviscid limit, whereas the autoregressive variant collapses to a nearly constant representation and fails to separate viscosity regimes.

We evaluate generalization across three settings: unseen initial conditions at trained viscosities (ID), unseen viscosities within the training range (OOD), and the inviscid limit ν=0, which tests extrapolation to a singular regime. Training data cover nine viscosities in the viscous regime.

Sweeping the latent representation dimension from d=0 to 4 reveals an optimum at d^*=1, consistent with the fact that viscosity is the only varying factor across the system family (Fig. 2a). Predictions remain accurate across all three evaluation splits (Fig. 2b). At ν=0, the ground truth corresponds to the entropy solution of the inviscid Burgers' equation [20, 21], which classical numerical integration typically reaches only by switching from parabolic to hyperbolic discretization schemes. In contrast, the learned latent representation varies smoothly and monotonically with viscosity, extending continuously to the inviscid regime (Fig. 2d, left). One-shot inference from a single unseen trajectory recovers a consistent latent representation for the inviscid system instance.

We next examine how decoder supervision influences factor identification. Under shock dynamics, most of the spatial domain remains smooth, while the shock front

occupies only a small region whose position depends strongly on viscosity. We therefore compare two variants that share the same encoder but differ in prediction strategy. The first (NODnet-DeepONet) predicts the full spatiotemporal field in a single forward pass using a DeepONet-style decoder [7], receiving supervision across all (x, t) locations simultaneously. The second (NODnet-AR) replaces the decoder with an autoregressive stepper trained only on one-step transitions. In the autoregressive setting, the training loss is dominated by smooth-region errors that vary weakly with viscosity, providing limited gradient signal for separating system instances. As a result, the autoregressive variant fails to distinguish viscosities and collapses to a nearly constant latent representation, whereas the full-field predictor learns a monotonic latent organization aligned with viscosity (Fig. 2d, Table 1).

This effect is also reflected in long-horizon prediction performance. The autoregressive baseline achieves low one-step error but accumulates substantial rollout drift, producing the largest 50-step prediction error among all models (Fig. 2c, Table 1). By contrast, the latent-conditioned full-field predictor maintains the lowest 50-step error across the ID, OOD, and inviscid splits, and remains within a bounded gap of the parameter-supervised reference reported in Supplementary Section S2.4. The no-conditioning baseline converges to a family-averaged solution because it receives no information distinguishing one viscosity regime from another.

Cylinder Flow Wake

The cylinder flow wake provides a challenging setting for evaluating Neural Operator Discovery under sparse and geometrically complex data. In the laminar Reynolds-number regime considered here ($\mathrm{Re} \in [75, 400]$), vortex shedding patterns and drag vary smoothly with viscosity. The governing equations are the 2D incompressible Navier–Stokes equations:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} &= \nu \nabla^2 \mathbf{u} - \nabla p/\rho \\ \mathbf{u}\,|_{\partial\Omega_{in}} &= u_\infty \end{aligned} \tag{9}$$

where ν is the kinematic viscosity, u_∞ is the inlet velocity, and $\mathbf{u} = (u, v)$ is the velocity field. Across the system family, viscosity ν is the only varying governing factor, while variations in u_∞ modify the initial flow development through the boundary condition.

This case probes scalability along two dimensions: the flow is defined on an unstructured mesh, and the training data are sparse, consisting of only 16 trajectories spanning four viscosity values and four inlet velocities. The ID split evaluates generalization to unseen u_∞ at trained viscosities, while the OOD split evaluates extrapolation to viscosities outside the training range (Fig. 5II(c)).

Sweeping the representation dimension from d=0 to 4 reveals an optimum at d^*=1, consistent with viscosity being the only governing factor varying across the system family (Supplementary Section S2.2). Larger representations degrade long-horizon rollout performance, indicating that additional dimensions capture trajectory-specific

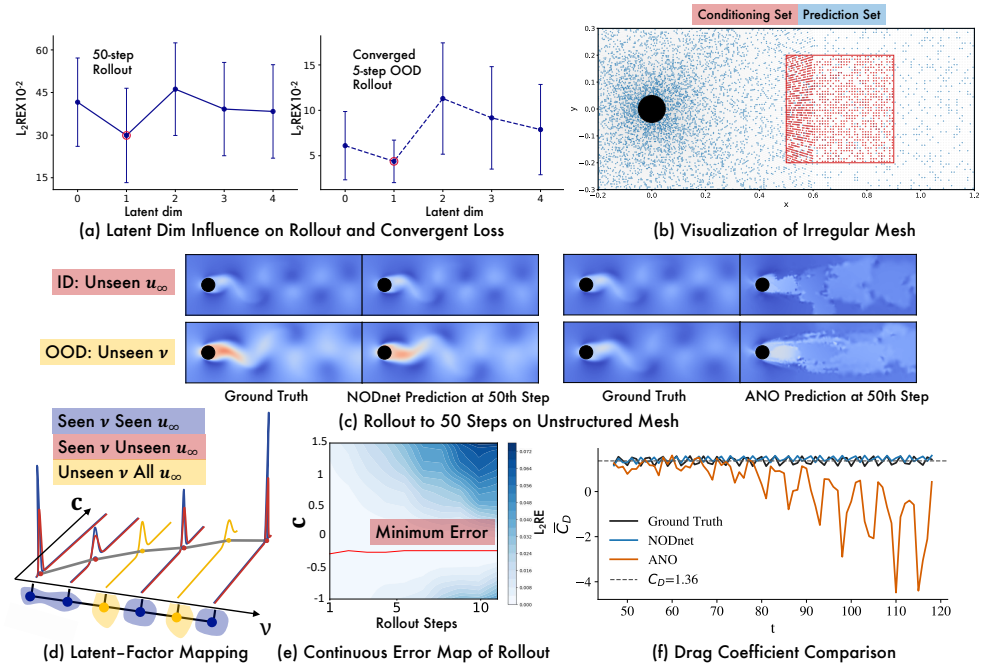


Fig. 3 Cylinder flow wake results. (a) Dimension selection based on long-horizon training and OOD rollout errors identifies d^*=1 (Supplementary Section S2.2). (b) Irregular-mesh prediction setting. The encoder infers the latent representation from a conditioning set (red), while the decoder predicts flow evolution on a separate prediction set (blue). (c) 50-step rollouts on the ID split (unseen u_∞) and OOD split (unseen ν). (d) Relationship between the inferred latent coordinate and viscosity. System instances organize along a smooth monotonic curve in latent space, with a convex loss landscape across prediction horizons. (e) Continuous rollout-error landscape as the latent representation value is perturbed. The error-minimizing latent value remains aligned with the inferred representation as rollout horizons increase. (f) Drag coefficient C_D at Re=300. The latent-conditioned model recovers the reference drag coefficient $\bar{C}_D$=1.36 and the correct vortex shedding frequency, whereas the autoregressive baseline exhibits frequency drift and develops nonphysical drag behavior.

features rather than meaningful system variation. At d=1, the inferred representation varies monotonically with viscosity (Fig. 3d), suggesting that the latent space organizes system instances according to the underlying physical structure.

To examine how the learned representation influences prediction, we perturb the inferred latent coordinate and evaluate the resulting rollout error. The error landscape remains smooth across rollout horizons, with prediction error minimized near the inferred latent representation (Fig. 3e). On the OOD split, representations inferred from previously unseen trajectories remain consistent with the learned structure (Fig. 3d), and one-shot inference achieves comparable performance. Together, these results suggest that the learned latent space generalizes smoothly beyond the training viscosity range.

To evaluate robustness under sparse and geometrically complex data, we train the model using only 16 trajectories defined on an unstructured mesh with 4,869

nodes per prediction step. Both the encoder and decoder are implemented using Point Transformer backbones [22], enabling latent conditioning and prediction directly on irregular geometries. Additional architectural details and ablation studies on encoder–decoder capacity allocation are provided in Supplementary Section S2.2 and Table S4.

Sparse supervision presents a second challenge: the model must simultaneously learn a generalizable solution operator and distinguish between different viscosity regimes from limited trajectories. To improve robustness, the encoder and decoder operate on spatially separated regions, where the encoder infers the latent representation from a conditioning window while the decoder predicts flow evolution on a distinct prediction domain (Fig. 3b). We additionally introduce a supervised contrastive regularization term [23] that encourages separation between latent representations associated with different system instances during early training. Together, these mechanisms stabilize latent-space organization under sparse supervision.

Despite the limited training data, the latent-conditioned model remains stable across both ID and OOD splits, achieving the lowest 50-step rollout error among the compared models (Table 1); a comparison to the viscosity-supervised reference is reported in Supplementary Section S2.4. The autoregressive baseline remains competitive at short horizons but accumulates rollout drift under distribution shift (Fig. 3c). The no-conditioning baseline performs well at very short horizons, where distinguishing viscosity is less critical, but deteriorates at longer horizons (Table 1), highlighting the importance of inferring system-specific variation.

Beyond rollout error metrics, we evaluate physical consistency through the drag coefficient at Re=300 (Fig. 3f). The latent-conditioned model recovers the reference drag coefficient $\bar{C}_D$=1.36 [24–26] and Strouhal number 0.205 [24] (St = 0.2037), whereas the autoregressive baseline develops frequency drift (St = 0.2451) and nonphysical drag behavior.

FitzHugh-Nagumo Reaction-Diffusion

The FitzHugh–Nagumo reaction–diffusion system [27, 28] models nerve impulse propagation through excitable wave dynamics, where the qualitative regime of the system (quiescent, oscillatory, traveling pulses, or spiral patterns) is jointly determined by the stimulus parameter k and the recovery rate β. The governing equations are

$$\begin{aligned} \frac{\partial u}{\partial t} &= \gamma_u \nabla^2 u + u - u^3 + k - v \\ \frac{\partial v}{\partial t} &= \gamma_v \nabla^2 v + \beta\,(u - v) \end{aligned} \tag{10}$$

where u is the membrane potential, v is the recovery variable, and periodic boundary conditions are imposed.

This system provides a setting for evaluating whether the learned latent space captures interpretable system-level variation and supports generalization beyond the training regime. Training uses 9 (k, β) pairs. The ID split evaluates unseen initial conditions at these parameter pairs, the OOD-Intra split evaluates 4 unseen parameter

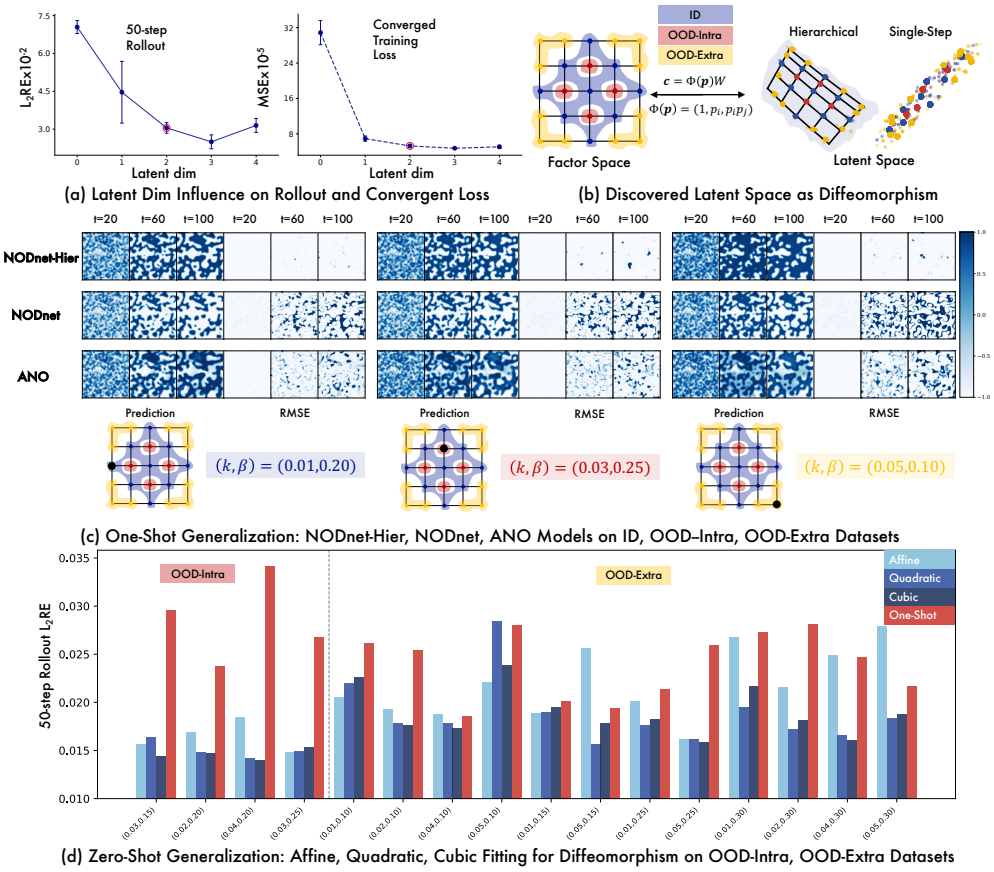


Fig. 4 FitzHugh–Nagumo reaction–diffusion results. (a) Dimension selection across three random seeds identifies d^*=2, consistent with the two governing factors (k, β) (Supplementary Section S2.3). (b) Learned latent-space organization for the 9 training system instances together with a quadratic mapping between latent and parameter spaces. Representations inferred from unseen system instances remain consistent with the learned structure. (c) Predictions and error maps for the hierarchical latent-conditioned model, the single-step latent-conditioned model, and the autoregressive baseline from the same initial condition. (d) Zero-shot prediction error as a function of the mapping complexity between latent and parameter spaces. Quadratic mappings improve prediction accuracy relative to affine mappings, while cubic mappings provide negligible additional benefit. One-shot inference is shown as a reference.

pairs inside the training hull, and the OOD-Extra split evaluates 12 pairs outside the training range (Fig. 5III(c)).

Sweeping the representation dimension from d=0 to 4 reveals an optimum at d^*=2, consistent with the two independent governing factors (k, β). Although prediction error continues to decrease slightly at d=3, principal component analysis of the learned representations shows that the third dimension remains effectively unused (Supplementary Section S2.3). The resulting latent space organizes system instances according

Table 1 Generalization performance across the three dynamical systems. Errors are reported as L_2 relative error ($\times 10^{-2}$, mean±std) at 1-, 5-, and 50-step rollout horizons. Latent-conditioned models (NODnet), no-conditioning baselines (SNO), and autoregressive baselines (ANO) are compared across in-distribution and out-of-distribution settings. The parameter-supervised reference (PNO), provided for context, is reported in Supplementary Section S2.4. Reported statistics for latent-conditioned and no-conditioning models are averaged over three random seeds, with variability computed across both seeds and evaluation trajectories (Supplementary Sections S2.1–S2.3). Model definitions and evaluation splits are provided in Methods.

Model	Subset	L_2 error ($\times 10^{-2}$)		
		1-step	5-step	50-step
Burgers' Equation				
SNO-DeepONet (5.0M)	ID	5.14±2.50	12.78±9.15	27.62±25.80
	OOD	5.38±2.47	14.05±9.05	30.35±25.84
	OOD-inviscid	5.29±1.49	12.90±4.85	24.98±9.38
NODnet-DeepONet (10M)	ID	4.46±1.87	7.56±3.60	**17.62±13.07**
	OOD	4.59±1.85	7.52±3.65	**17.71±14.09**
	OOD-inviscid	4.74±1.42	9.17±4.09	**20.34±9.05**
NODnet-AR (10M)	ID	3.47±3.68	12.02±10.92	61.48±36.14
	OOD	4.09±4.02	13.76±11.56	64.42±36.38
	OOD-inviscid	2.15±0.98	8.95±4.71	55.18±20.97
ANO$^{\downarrow}$ (9.8M)	ID	**0.92±0.67**	**6.19±3.41**	62.88±34.34
	OOD	**0.99±0.73**	**6.49±3.53**	64.73±34.31
	OOD-inviscid	**1.31±0.74**	**6.72±2.89**	64.86±29.31
Cylinder Flow Wake				
SNO (3.5M)	ID	**1.47±0.65**	**3.46±1.93**	36.09±15.17
	OOD	**1.63±0.84**	6.11±3.76	34.53±13.76
NODnet (15M)	ID	1.69±0.79	4.00±2.60	**29.89±16.61**
	OOD	1.72±0.86	**4.38±2.34**	**31.04±15.11**
ANO$^{\downarrow}$ (14M)	ID	2.21±0.82	5.33±2.90	31.63±12.26
	OOD	1.99±0.95	4.52±2.32	36.31±9.43
FitzHugh-Nagumo Reaction-Diffusion				
SNO-Hier (2.5M)	ID	1.31±0.09	1.78±0.10	7.33±1.63
	OOD-Intra	1.28±0.06	1.75±0.07	7.05±1.24
	OOD-Extra	1.38±0.11	1.96±0.16	9.31±2.28
NODnet-Hier (3.1M)	ID	0.91±0.06	**1.20±0.10**	**2.52±0.67**
	OOD-Intra	0.90±0.04	**1.19±0.09**	**3.05±0.93**
	OOD-Extra	1.03±0.19	**1.61±0.66**	**5.49±4.38**
NODnet (3.1M)	ID	**0.79±0.03**	2.21±0.15	22.56±2.81
	OOD-Intra	**0.79±0.02**	2.18±0.11	21.83±2.41
	OOD-Extra	**0.79±0.04**	2.27±0.16	23.70±3.41
ANO$^{\downarrow}$ (2.8M)	ID	1.06±0.20	2.07±0.55	46.70±5.73
	OOD-Intra	0.97±0.13	1.85±0.36	44.68±2.80
	OOD-Extra	1.34±0.24	2.69±0.58	50.79±8.05

to the underlying parameter variation (Fig. 4b). A quadratic mapping between the latent space and the parameter space generalizes to held-out system instances with R^2=0.9937 and a Jacobian condition number of 13.6±7.0, indicating that the learned correspondence remains smooth and stable beyond the training set (Eq. S7).

We next examine how supervision structure influences factor identification. At matched model capacity, replacing the hierarchical decoder in NODnet-Hier with a single-step decoder in NODnet increases long-horizon rollout error by approximately an order of magnitude (Table 1). The hierarchical decoder supervises prediction simultaneously at lead times 1, 3, and 9, feeding longer-horizon dynamical information back into the encoder and stabilizing recovery of the latent organization (Fig. 4b, right). This parallels the Burgers' system, where the supervision geometry of the decoder strongly influences the ability to identify governing factors.

The learned latent space further enables zero-shot generalization to previously unseen system instances. To evaluate this behavior, we fit analytical mappings of increasing complexity between the latent space and the parameter space and generate predictions directly from sampled latent representations. The mappings include: affine (first-order), quadratic (second-order), and cubic (third-order), with details in Supplementary Information S2.3. A quadratic mapping improves prediction accuracy relative to an affine fit, while higher-order polynomial fits provide negligible additional benefit (Fig. 4d). The comparable performance of one-shot and zero-shot inference (Fig. 4c) indicates that the latent space captures physically meaningful system variation that extrapolates beyond the training regime.

Baseline comparisons further support the role of latent conditioning in long-horizon prediction. Within the (k, β) training hull, the latent-conditioned hierarchical model achieves nearly uniform performance across the ID and OOD-Intra splits, with the lowest 5- and 50-step rollout error among the compared models (Table 1). The no-conditioning baseline fails to distinguish between parameter regimes and accumulates substantially larger long-horizon error. The autoregressive baseline performs worse still, as it conditions prediction only on past-state histories and accumulates rollout drift over time (Fig. 4c). Notably, even without hierarchical prediction, the latent-conditioned model achieves roughly half the 50-step prediction error of the autoregressive baseline, indicating that latent conditioning together with trajectory-decoupled sampling accounts for much of the performance gain.

Discussion

We introduced Neural Operator Discovery as the problem of learning both a shared solution operator and a representation of system-specific variation directly from heterogeneous trajectories without parameter supervision. Across the systems studied here, the learned latent space aligns consistently with the underlying governing factors, while long-horizon prediction outperforms autoregressive and no-conditioning baselines (Table 1) and remains within a bounded gap of the parameter-supervised reference reported in Supplementary Section S2.4. These results suggest that latent conditioning can provide an interpretable and scalable alternative to explicit parameter supervision in operator learning.

A central observation of this work is that the learned latent space exhibits smooth structure across system instances and supports both one-shot and zero-shot generalization. In particular, the FitzHugh–Nagumo system demonstrates extrapolation beyond

the convex hull of the training regime, despite the model never observing those parameter combinations during training. Although the method is trained only to minimize prediction error, two aspects of the formulation appear important for this behavior. First, trajectory-decoupled sampling encourages the encoder to retain only information shared across trajectories from the same system instance, suppressing dependence on transient initial-condition features. Second, restricting the latent representation to the minimal dimension sufficient to distinguish system instances limits the capacity available for encoding trajectory-specific variation. Together, these constraints encourage a compact latent organization that reflects system-level structure.

The results also suggest that factor identification depends strongly on the supervision geometry of the decoder. In both the Burgers' and FitzHugh–Nagumo systems, decoders receiving longer-horizon or full-field supervision recover more structured latent representations and achieve more stable long-term rollouts than autoregressive predictors trained only on local transitions. This indicates that accurate system identification may require supervision signals that remain sensitive to governing-factor variation over extended temporal horizons.

The Neural Operator Discovery setting also connects naturally to broader questions of identifiability in latent-variable models. Khemakhem et al. [29] establish identifiability in nonlinear ICA through auxiliary variables, while Locatello et al. [30] analyze weakly supervised disentanglement under shared latent factors. Hyvärinen et al. [31] identify latent structure through temporal contrastive learning. In our setting, trajectories grouped by system instance provide an implicit supervisory signal, while latent dimension selection introduces an additional model-selection component absent from most existing approaches. Supplementary Sections S3.2–S3.6 further discuss connections to latent-variable models for scientific data [32, 33], gradient-based meta-learning [19], symbolic regression, and generative modeling [34].

Several limitations remain. The present study considers continuously varying factors in low-dimensional Euclidean spaces; discrete system classes, bifurcating dynamics, and topologically nontrivial factor manifolds remain unexplored. All experiments are conducted on 1D and 2D systems, and extension to large-scale 3D dynamics remains an important direction for future work. The datasets considered here are simulation-generated; robustness under experimental noise, sparse sensing, and missing observations requires further investigation. In addition, the minimum number of system instances required for reliable latent-space formation remains unclear, particularly in sparse-data regimes. Finally, while the learned latent spaces exhibit smooth and approximately invertible correspondence with governing factors across the systems studied here, a formal theoretical characterization of these properties remains open.

Methods

This section summarizes the implementations used across the three primary case studies, including dataset generation and encoder and decoder architectures. The Burgers' system uses a CNN encoder paired with a DeepONet-style decoder; the cylinder flow wake uses Point Transformer backbones for both encoder and decoder;

and the FitzHugh–Nagumo system uses a recurrent encoder with a hierarchical U-CNN decoder. Full architectural details, hyperparameters, and dimension-selection procedures are provided in Supplementary Sections S2.1–S2.3.

Burgers' Equation

Burgers' equation is solved using spatial and temporal discretizations satisfying the CFL stability condition ($\Delta x{=}5 \times 10^{-4}$, $\Delta t{=}5 \times 10^{-3}$). Simulations are generated on a 4001×1001 grid and downsampled to 401×101 for learning. The training set spans nine viscosity values in the viscous regime, while evaluation includes both interpolation within the training range and extrapolation to the inviscid limit $\nu{=}0$ (Fig. 5I(c)). For each system instance, 50 trajectories are generated from different initial conditions.

Following the factorized formulation in Eq. 4, the encoder infers a latent representation from a conditioning trajectory, while the decoder predicts the full spatiotemporal field from an initial condition and the inferred representation.

Encoder

The encoder is implemented as a convolutional neural network that processes the conditioning trajectory as a (T, X) image and outputs a scalar latent representation through global pooling and a small multilayer perceptron (MLP).

Decoder

The decoder is implemented using a DeepONet-style architecture that predicts the solution field at arbitrary spatiotemporal coordinates in a single forward pass:

$$\hat{u}(t,x) = u_0(x) + \alpha_{\text{res}} \left\langle\, b(\mathbf{u}_0, c),\, \tau\big(\gamma(x,t)\big) \,\right\rangle, \tag{11}$$

Branch and trunk networks are implemented as MLPs, and the coordinate inputs are sinusoidally positional encoded to improve representation of sharp shock fronts (Supplementary Section S2.1). Training minimizes mean-squared prediction error over randomly sampled spatiotemporal query points. The NODnet-AR architecture can also be found in Supplementary Section S2.1.

Cylinder Flow Wake

The dataset is generated from high-fidelity simulations of incompressible flow past a cylinder on the 2D domain $\Omega = [-9, 21] \times [-6, 6]$. Inlet velocity takes values $u_\infty \in 15, 16, 17, 18, 19, 20$ and viscosity $\nu \in 0.005, 0.008, 0.011, 0.014, 0.017, 0.020$. With cylinder diameter fixed at 0.1, the resulting system family spans Reynolds numbers from 75 to 400, and the simulations are validated against reference statistics [24]. To evaluate learning under sparse supervision, training uses only 16 trajectories sampled across (ν, u_∞) combinations (Fig. 5II(c)). Generalization is evaluated on two settings: unseen inlet velocities at trained viscosities, and unseen viscosities outside the training range.

The decoder predicts the next-step velocity field autoregressively from the current state and a latent representation inferred by the encoder from a separate conditioning

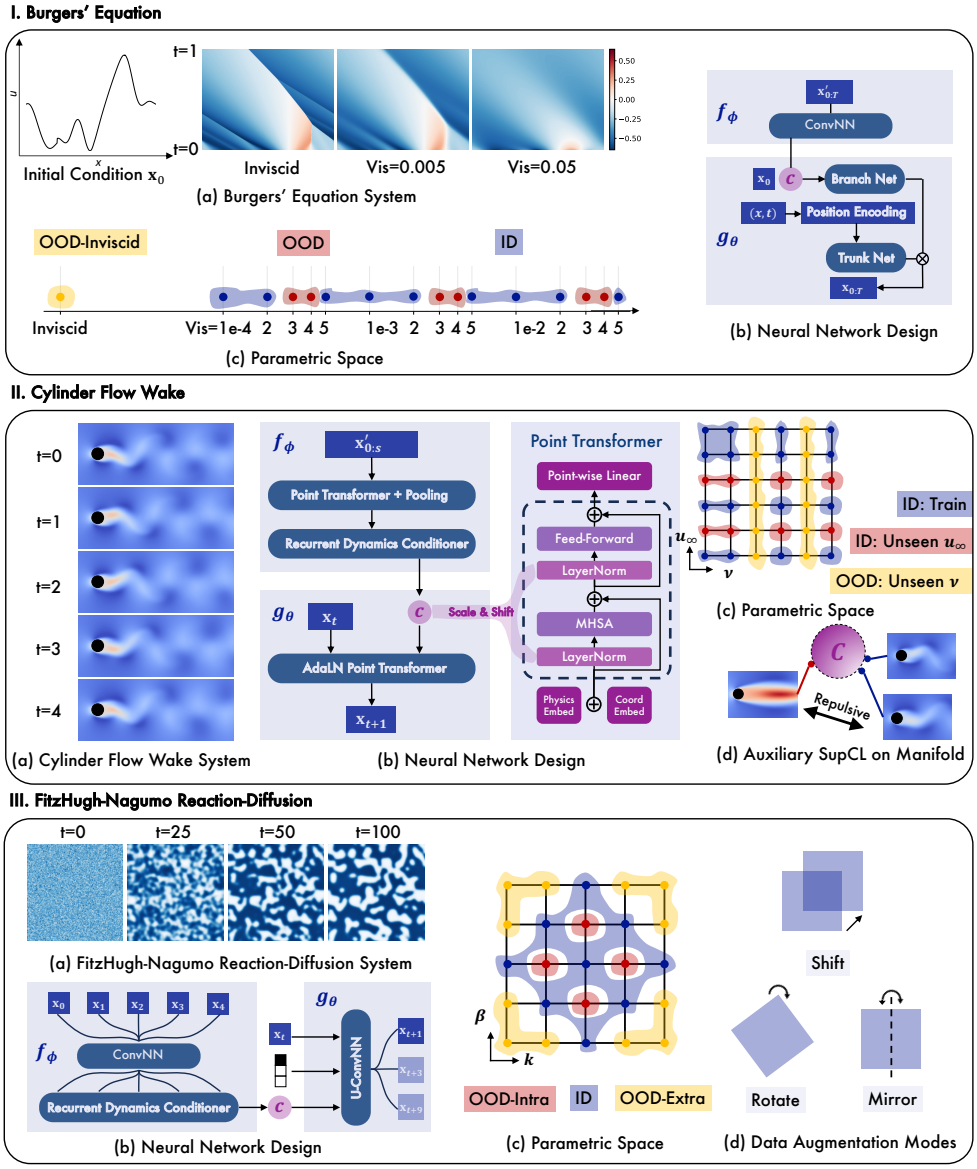


Fig. 5 System descriptions and learning configurations. I. Burgers' equation: (a) sampled spatiotemporal fields across viscosity regimes, (b) CNN encoder and DeepONet-style decoder with positional encoding, (c) training and evaluation splits in viscosity space. **II. Cylinder flow wake:** (a) sampled trajectory on the irregular mesh, (b) Point Transformer encoder and AdaLN-conditioned Transformer decoder, (c) training and evaluation splits across viscosity and inlet velocity, (d) supervised contrastive regularization of the latent space under sparse supervision. **III. FitzHugh–Nagumo reaction–diffusion:** (a) sampled trajectory, (b) recurrent encoder and hierarchical U-CNN decoder, (c) parameter-space splits across (k, β) pairs, (d) symmetry-based data augmentation under O(2) transformations.

trajectory generated under the same viscosity but different initial conditions. Conditioning trajectories are chosen adjacent to the prediction window so that the prediction setting remains comparable to history-conditioned autoregressive neural operators.

Encoder and decoder

Both encoder and decoder are implemented using Transformer architectures designed for irregular meshes [22, 35] (Fig. 5II(b)). The encoder processes a localized spatiotemporal conditioning window to infer the latent representation from wake dynamics, while the decoder predicts flow evolution over the target prediction region using AdaLN-Zero conditioning [36]. Rotary Position Embedding (RoPE) [37] is used to preserve relative spatial information on the irregular mesh. Additional architectural details and ablation studies are provided in Supplementary Section S2.2.

Latent space regularization

To stabilize latent space organization under sparse supervision, we introduce a supervised contrastive regularization term [23] that encourages separation between representations associated with different system instances (Fig. 5II(d)). The contribution of this regularization is gradually annealed during training so that the prediction objective dominates optimization at later stages.

FitzHugh-Nagumo Reaction-Diffusion

The FitzHugh–Nagumo reaction–diffusion dataset is generated on a 128^2 grid across 25 distinct system instances: 9 for training, 4 for the OOD-Intra split (interpolated parameter pairs), and 12 for the OOD-Extra split (extrapolated parameter pairs) (Fig. 5III(c)). For each system instance, 50 trajectories are generated from Gaussian random-field initial conditions, with 40 used for training and 10 reserved for evaluation of initial-condition generalization.

The decoder predicts the next-step fields $(\mathbf{u}, \mathbf{v})'_{t+1}$ autoregressively from the current state $(\mathbf{u}, \mathbf{v})'_t$ and a latent representation inferred by the encoder from a separate conditioning trajectory generated under the same (k, β) pair but different initial conditions.

Encoder

To process high-dimensional spatiotemporal trajectories, the encoder incorporates a recurrent inductive bias:

$$\begin{aligned} \mathbf{h}_i &= \Psi(\mathbf{h}_{i-1}, \psi(\mathbf{x}_i)), \quad i = 0, \ldots, s, \\ c &= o(\mathbf{h}_s), \end{aligned} \tag{12}$$

where $\Psi(\cdot, \cdot)$ denotes a recurrent unit (GRU [38] or LSTM [39]) operating on learned per-frame representations $\psi(\mathbf{x}_i)$. The final hidden state is projected to the latent representation c (Fig. 5III(b)).

Decoder

Because the dynamics are high-dimensional and evolve over long temporal horizons, the decoder predicts future states autoregressively using a hierarchical prediction strategy inspired by hierarchical temporal aggregation [40] and multi-output forecasting [41]. Three steppers with lead times of 1, 3, and 9 steps share a U-shaped CNN backbone (Fig. 5III(b)). Longer-horizon supervision stabilizes rollout prediction and improves recovery of system-level variation by feeding long-timescale dynamical information back into the latent-conditioning process.

Representing prediction horizons hierarchically also reduces the number of network function evaluations required for long rollouts from $\mathcal{O}(N)$ to $\mathcal{O}(\log_3 N)$ for the N-th prediction step. For example,

$$\hat{\mathbf{x}}_{25} = g_9^{(2)} \circ g_3^{(2)} \circ g_1^{(1)}(\mathbf{x}_0, c).$$

Symmetry-aware data augmentation

Under periodic boundary conditions, the FitzHugh–Nagumo system is equivariant to spatial translation, mirroring, and rotation (Fig. 5III(d)). Applying these transformations enlarges the effective dataset by approximately 10× while preserving the underlying system instance. This augmentation encourages the encoder to identify governing factors independently of the spatial orientation or position of excitation patterns and improves robustness across initial conditions.

References

[1] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the national academy of sciences **113**(15), 3932–3937 (2016)

[2] M. Jacobs, B.W. Brunton, S.L. Brunton, J.N. Kutz, R.V. Raut. Hypersindy: Deep generative modeling of nonlinear stochastic governing equations (2023). URL https://arxiv.org/abs/2310.04832

[3] S. d'Ascoli, S. Becker, A. Mathis, P. Schwaller, N. Kilbertus. Odeformer: Symbolic regression of dynamical systems with transformers (2023). URL https://arxiv.org/abs/2310.05573

[4] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces with applications to pdes. Journal of Machine Learning Research **24**(89), 1–97 (2023)

[5] K. Azizzadenesheli, N. Kovachki, Z. Li, M. Liu-Schiaffini, J. Kossaifi, A. Anandkumar, Neural operators for accelerating scientific simulations and design. Nature Reviews Physics **6**(5), 320–328 (2024)

[6] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential

equations. arXiv preprint arXiv:2010.08895 (2020)

[7] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nature machine intelligence **3**(3), 218–229 (2021)

[8] H. Zhou, Y. Ma, H. Wu, H. Wang, M. Long, Unisolver: Pde-conditional transformers are universal pde solvers. arXiv preprint arXiv:2405.17527 (2024)

[9] J. Sun, Y. Liu, Z. Zhang, H. Schaeffer, Towards a foundation model for partial differential equations: Multioperator learning and extrapolation. Physical Review E **111**(3), 035304 (2025)

[10] H. Wu, H. Luo, H. Wang, J. Wang, M. Long, Transolver: A fast transformer solver for pdes on general geometries. arXiv preprint arXiv:2402.02366 (2024)

[11] Z. Fang, S. Wang, P. Perdikaris, Learning only on boundaries: a physics-informed neural operator for solving parametric partial differential equations in complex geometries. Neural computation **36**(3), 475–498 (2024)

[12] Y. Cao, Y. Liu, L. Yang, R. Yu, H. Schaeffer, S. Osher, Vicon: Vision in-context operator networks for multi-physics fluid dynamics prediction. arXiv preprint arXiv:2411.16063 (2024)

[13] Z. Hao, C. Su, S. Liu, J. Berner, C. Ying, H. Su, A. Anandkumar, J. Song, J. Zhu, Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. arXiv preprint arXiv:2403.03542 (2024)

[14] Z. Chen, S. Deng, Flow marching for a generative PDE foundation model. arXiv preprint arXiv:2509.18611 (2025)

[15] M. McCabe, B.R.S. Blancard, L.H. Parker, R. Ohana, M. Cranmer, A. Bietti, M. Eickenberg, S. Golkar, G. Krawezik, F. Lanusse, et al., Multiple physics pretraining for physical surrogate models. arXiv preprint arXiv:2310.02994 (2023)

[16] K. Arora, L.E. Asri, H. Bahuleyan, J.C.K. Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation (2023). URL https://arxiv.org/abs/2204.01171

[17] Z. Chen, S. Deng. Latent generative solvers for generalizable long-term physics simulation (2026). URL https://arxiv.org/abs/2602.11229

[18] P.Y. Lu, S. Kim, M. Soljačić, Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning. Physical Review X **10**(3) (2020). https://doi.org/10.1103/physrevx.10.031056. URL http://dx.doi.org/10.1103/PhysRevX.10.031056

[19] Q. Li, T. Wang, V. Roychowdhury, M.K. Jawed, Metalearning generalizable dynamics from trajectories. Physical Review Letters **131**(6), 067301 (2023)

[20] O.A. Oleinik, Discontinuous solutions of non-linear differential equations. Uspekhi Matematicheskikh Nauk **12**(3), 3–73 (1957)

[21] S.N. Kružkov, First order quasilinear equations in several independent variables. Mathematics of the USSR-Sbornik **10**(2), 217–243 (1970)

[22] H. Zhao, L. Jiang, J. Jia, P.H. Torr, V. Koltun, in Proceedings of the IEEE/CVF international conference on computer vision (2021), pp. 16259–16268

[23] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning. Advances in neural information processing systems **33**, 18661–18673 (2020)

[24] R.D. Henderson, Details of the drag curve near the onset of vortex shedding. Physics of Fluids **7**(9), 2102 (1995)

[25] C.H.K. Williamson, Vortex dynamics in the cylinder wake. Annual Review of Fluid Mechanics **28**, 477–539 (1996)

[26] M. Schäfer, S. Turek, F. Durst, E. Krause, R. Rannacher, in Flow simulation with high-performance computers II: DFG priority research programme results 1993–1995 (Springer, 1996), pp. 547–566

[27] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane. Biophysical journal **1**(6), 445–466 (1961)

[28] J. Nagumo, S. Arimoto, S. Yoshizawa, An active pulse transmission line simulating nerve axon. Proceedings of the IRE **50**(10), 2061–2070 (1962)

[29] I. Khemakhem, D. Kingma, R. Monti, A. Hyvarinen, in International Conference on Artificial Intelligence and Statistics (PMLR, 2020), pp. 2207–2217

[30] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, M. Tschannen, in International Conference on Machine Learning (PMLR, 2020), pp. 6348–6359

[31] A. Hyvärinen, H. Sasaki, R. Turner, in International Conference on Artificial Intelligence and Statistics (PMLR, 2019), pp. 859–868

[32] B. Chen, K. Huang, S. Raghupathi, I. Chandratreya, Q. Du, H. Lipson, Automated discovery of fundamental variables hidden in experimental data. Nature Computational Science **2**(7), 433–442 (2022)

[33] X. Chen, B.W. Soh, Z.E. Ooi, E. Vissol-Gaudin, H. Yu, K.S. Novoselov, K. Hippalgaonkar, Q. Li, Constructing custom thermodynamics using deep learning. Nature Computational Science **4**(1), 66–85 (2024)

[34] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial networks (2014). URL https://arxiv.org/abs/1406.2661

[35] B. Alkin, A. Fürst, S. Schmid, L. Gruber, M. Holzleitner, J. Brandstetter, Universal physics transformers: A framework for efficiently scaling neural operators. Advances in Neural Information Processing Systems **37**, 25152–25194 (2024)

[36] W. Peebles, S. Xie, in Proceedings of the IEEE/CVF international conference on computer vision (2023), pp. 4195–4205

[37] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, Y. Liu. Roformer: Enhanced transformer with rotary position embedding (2023). URL https://arxiv.org/abs/2104.09864

[38] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)

[39] S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)

[40] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, Q. Tian, Accurate medium-range global weather forecasting with 3D neural networks. Nature **619**(7970), 533–538 (2023)

[41] R. Green, G. Stevens, Z. Abdallah, T.M.S. Filho. Stratify: Unifying multi-step forecasting strategies (2024). URL https://arxiv.org/abs/2412.20510

Acknowledgments

Z.C. and S.D. acknowledge financial support from MIT Sea Grant. Q.L. and J.H. are supported by the Major Research Plan of the National Natural Science Foundation of China (Grant No. 92470109) and the Zhejiang Provincial Team of Leading Talents in Innovation and Entrepreneurship (Grant No. 2024R01002). The authors acknowledge the MIT Office of Research Computing and Data (ORCD) for providing high-performance computing resources that have contributed to the research results reported within this paper. A portion of this research was performed using computational resources sponsored by the U.S. Department of Energy's Office of Critical Minerals and Energy Innovation and located at the National Laboratory of the Rockies. We thank Huaibo Chen, Benjamin Koenig, and Nicolas Tricard for advice on numerical methods for data generation in the early stage of this project.

Author contributions

Z.C.: conceptualization, methodology, investigation, software, visualization, writing–original draft, writing–review and editing. Q.L.: conceptualization, methodology, software (early prototyping), writing–review and editing. J.H.: investigation, software, visualization. S.D.: supervision, funding acquisition, writing–review and editing.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information. Supplementary Sections S1–S3 provide additional benchmark cases (bistable oscillator, wall-bouncing system, mass-spring chain, KPP equation), extended simulation and architecture details for each case study, the drag-force computation procedure, efficiency comparison with iMODE, and alternative theoretical perspectives on the diffeomorphic latent space.

Data availability. The Burgers' dataset used in this work is available at https://doi.org/10.5281/zenodo.20372988. The FitzHugh-Nagumo reaction-diffusion dataset and cylinder flow wake dataset can be reproduced using MATLAB or OpenFOAM with the simulation parameters detailed in Supplementary Section S3.

Code availability. The source code for all seven case studies (three in the main text, four in Supplementary Information) is available at https://doi.org/10.5281/zenodo.20406331.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplement.pdf