Supplementary Information: Neural operator discovery from heterogeneous trajectories

Zituo Chen, Qiaofeng Li, Jiaxin Hu, and Sili Deng

S1 Supplementary Results

Table S1 summarises the seven case studies reported across the main text and this Supplementary Information. Each case stresses a different aspect of the minimal conditioning principle and is implemented with a different backbone pair; the recovered d^* matches the number of physically independent governing factors in every case.

Table S1: Seven NODnet case studies. Section, observation structure, governing factors, recovered latent dimension d^*, and the encoder/decoder backbones. Distinct backbones used across the seven cases: CNN, U-CNN, DenseNet, DeepONet, Point Transformer, recurrent (LSTM/GRU), Fourier Neural Operator, Neural ODE.

Case	Section	Domain	Factors	d^*	Encoder	Decoder
Bistable oscillator	S1.1	2-DoF ODE	(k_1, k_3)	2	DenseNet	DenseNet
Wall-bouncing ball	S1.2	2D billiard	(W, H)	2	DenseNet	DenseNet
Mass-spring chain	S1.3	10-DoF ODE	(c, k)	2	DenseNet	Neural ODE
KPP reaction-diffusion	S1.4	1D PDE	(r, D)	2	FNO + LSTM	FNO
Burgers' equation	Main / S2.1	1D PDE	ν	1	CNN	DeepONet (PE trunk)
Cylinder-flow wake	Main / S2.2	2D N–S, mesh	ν	1	Point Transformer	Point Transformer (AdaLN)
FitzHugh–Nagumo RD	Main / S2.3	2D PDE	(k, β)	2	Recurrent (LSTM/GRU)	Hierarchical U-CNN

The four subsections below complement the main text by stressing aspects of the design that the Burgers', FN-RD, and cylinder-wake cases do not: low-dimensional ODE families (bistable oscillator), geometric variation (wall-bouncing system), boundary-condition generalization (mass-spring chain), and resolution-agnostic discovery (KPP). In every case, the same design, namely a compact encoder producing a latent at the gauged minimum dimension and a decoder conditioned on it, is used; only the backbone architectures change to match the observation structure.

S1.1 Bistable system: dimension gauging on a low-dimensional ODE family

This case establishes the elbow signal and the emergent diffeomorphism on the simplest setting: a two-parameter family with no spatial structure. We consider a bistable oscillator with two stable equilibrium states (Fig. S1(a)), whose simplest form is characterized by a cubic polynomial stiffness featuring a negative linear coefficient and a positive cubic coefficient.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ k_1 & -c \end{bmatrix}}_{\text{linear part}} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -k_3 x^3 \end{bmatrix}}_{\text{nonlinear part}} \tag{S1}$$

Bistable systems exhibit excitation-dependent nonlinear behaviors: small excitations result in intra-well oscillations, where the system oscillates around one equilibrium state, while large excitations trigger inter-well oscillations characterized by transitions between the two stable states. In this example, we demonstrate that NODnet is able to learn the shared dynamics of the system family without knowing the actual factors.

The training dataset contains 112 system instances, each taking a factor combination between $k_1 = [-1.0 : -0.1 : -0.4]$ N/m and $k_3 = [2.0 : 0.2 : 5.0]$ N/m^3. For each system instance 10 trajectories, with initial positions $x_0 = [0.1 : 0.1 : 1.0]$ m, initial velocities $\dot{x}_0 = 0$ m/s, and time marching stepsize $\Delta t = 0.1$ s are generated. We randomly sample trajectories of length 500 within

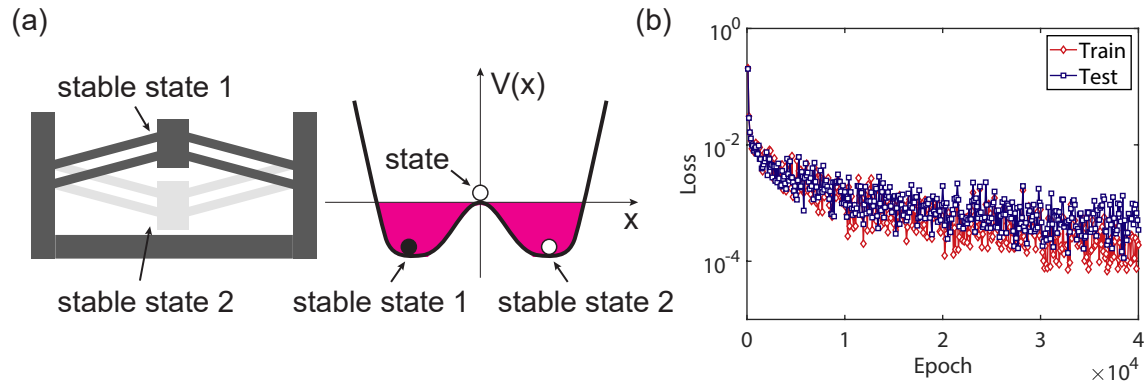

Figure S1: (a) An illustrative bistable system. Two potential wells typically exist in the potential function of a bistable system, marking two stable states. The system thus exhibits local intra-well and global inter-well motion patterns; (b) Sample training and test loss histories for the bistable system.

the long 1000-step trajectories to mimic random initial conditions for conditioning sequences. Sample train and test results are shown in Fig. S1(b). The test trajectories are generated in the same way as the training sequences but of a different set of factors. After training, the physically meaningful latent space is formed.

We sweep the dimension of the latent $\mathbf{c}$ and track the converged training loss. The loss curve forms an "elbow" shape as the latent dimension increases; the "elbow" point marks the optimal dimension for $\mathbf{c}$, equal to the true number of independent factors of the system.

The $\mathbf{c}$-space is empirically diffeomorphic to the true factor space. We validate this by training a neural ODE [1] that transports the learnt $\mathbf{c}$ to the true factor $\boldsymbol{\phi}$ [2], i.e., finding weights $\boldsymbol{\Theta}$ of a neural network $\mathbf{v}$ such that

$$\begin{aligned} &\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{v}_{\boldsymbol{\Theta}}(\mathbf{z}) \\ &\mathbf{z}(0) = \mathbf{c}_i,\ \mathbf{z}(1) = \boldsymbol{\phi}_i,\ i = 1, \ldots, N_s \end{aligned} \tag{S2}$$

The learnt NODE realises a continuous and differentiable transformation from the $\mathbf{c}$ space into the true factor space, instantiating the diffeomorphism. The latent space is physically meaningful and fully interpretable, with each point indexing a system instance and enabling one-shot generalization to unseen system instances for response prediction.

S1.2 Wall-bouncing system: geometry as the governing factor

This case tests whether the minimal conditioning principle extends to a governing factor that is *geometric* rather than a coefficient in an equation. A ball bounces elastically inside a rectangular region whose width and height vary across system instances (Fig. S2(a)); the two factors control the domain boundary rather than a differential term, and the encoder sees only trajectories, not the domain dimensions. The NODnet architecture is shown in Fig. S2(b): the encoder receives trajectories, and the decoder receives the concatenation of an initial condition and the latent produced by the encoder, trained against trajectory targets.

We utilize a dataset comprising 25 system instances. Each instance corresponds to a specific combination of factors width $= [2.1, 2.3, 2.5, 2.7, 2.9]$ m and height $= [2.1, 2.3, 2.5, 2.7, 2.9]$ m. For each system instance, 25 trajectories are generated with different initial velocities. Trajectory time sampling stepsize $\Delta t = 0.1$ s. We divide the dataset into a training set (13 system instances) and a test set (12 system instances), separated by odd/even numbering (Fig. S2(c)).

We employ DenseNet[3] architecture for both encoder and decoder (see Fig. S2(b)). The learnt latent space for training and testing are shown in Figs. S2(d) and (e). After constructing a diffeomorphism between the latent space and true geometric factor space, the transported latents are shown in Figs. S2(f) and (g), with true (width, height) pairs indicated by black dots. Figure S2(h) illustrates the learned diffeomorphism mapping from the latents to the true geometric factors.

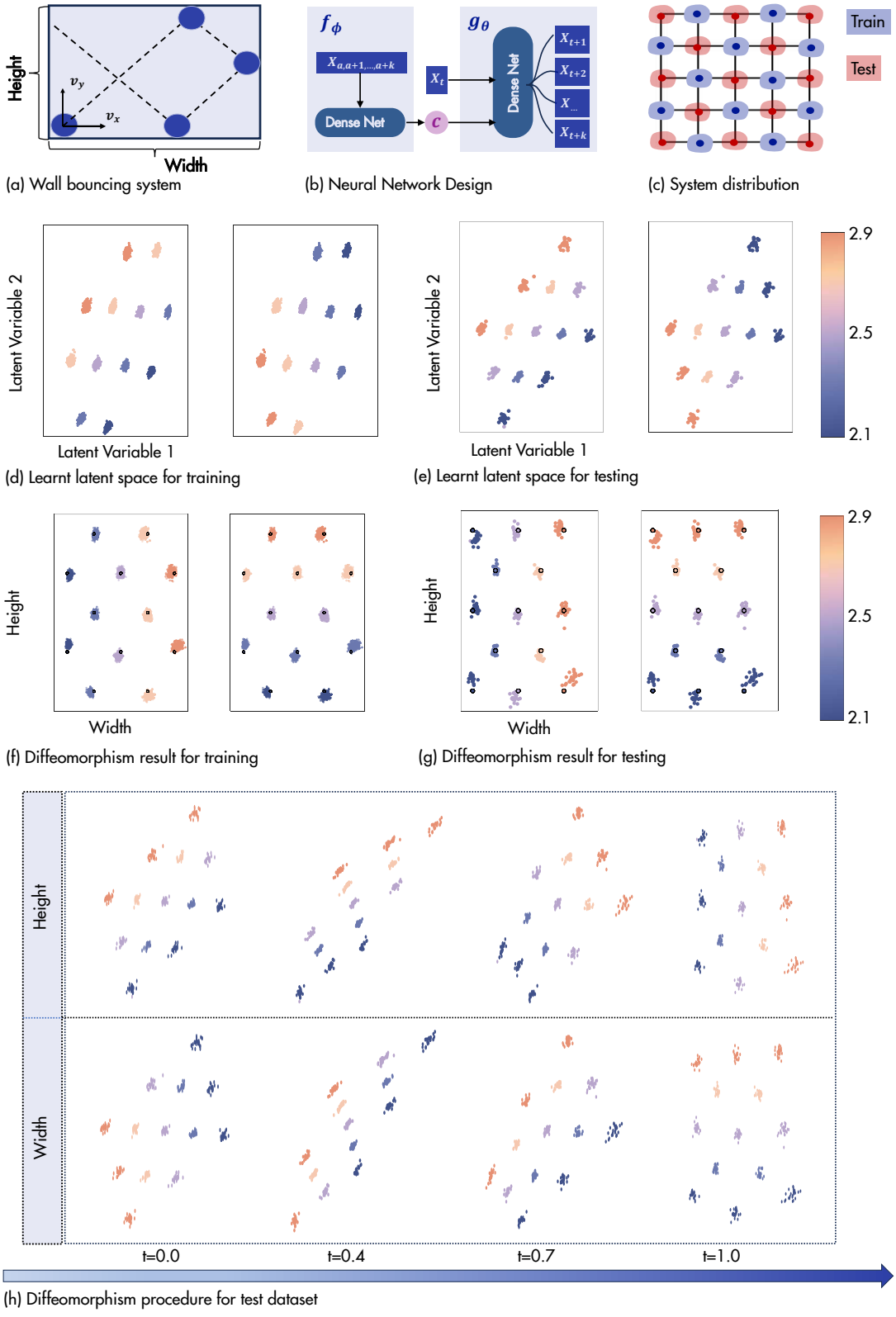

Figure S2: (a) An illustrative wall bouncing system; (b) The adopted NODnet architecture in the wall bouncing system case; (c) The distribution of system instances with respect to geometric factors; (d) The learnt latent space for training; (e) The learnt latent space for testing; (f) The diffeomorphism result for training; (g) The diffeomorphism result for testing; (h) Illustration of the diffeomorphism procedure from latent space to true geometric factor space for test dataset.

S1.3 Mass-spring chain: boundary-condition generalization via a differential decoder

This case tests whether the minimal conditioning principle, together with a decoder built on a differential operator, supports generalization across *unseen boundary conditions* at test time. The system is a 10-DoF mass-spring chain (Fig. S3(a)). The decoder is a neural ordinary differential equation (NODE, Fig. S3(b)), which exposes the boundary through its state rather than through its parameters. Because the latent produced by the encoder carries only the governing factors (damping and stiffness), and not the boundary specification, changing the boundary at test time leaves the learned operator family intact.

The training dataset contains 16 system instances, each taking a factor combination between $c = [0.1, 0.3, 0.5, 0.7]$ N $\cdot$ s/m and $k = [2.0, 3.0, 4.0, 5.0]$ N/m. For each system instance 11 trajectories with random initializations and time marching stepsize $\Delta t = 0.1$ s are generated.

First we train and test on the same boundary conditions. We train with trajectories of 20 time steps; after training, we generate trajectories of 50 time steps. The results are shown in Fig. S3(d). NODnet performs accurate extrapolation outside of the training time span. This proves that NODnet with NODE as decoder is able to perform accurate long-term rollouts.

The learnt latent space (training denoted as circles, testing as asterisks) is shown in Fig. S3(c). Two principal directions exist in the latent space, respectively corresponding to damping and stiffness increases of the system family.

Then we train with the right boundary *fixed* and test with the right boundary *free*. Test is performed on system instances excluded in the training dataset. When trajectories with the *fixed* boundary condition are available for the test system instances, we simply pass these trajectories through encoder to generate the latents. These variables together with the NODE are directly applied to the *free* boundary condition for test-time prediction. The results are shown in Fig. S4(a).

When only trajectories with the *free* boundary condition are available for the test system instances, the decoder can be used as a encoder, similar to the approach in [4].

$$\mathbf{c} = \arg\min_{\mathbf{c}} \|\mathbf{X} - \mathrm{NODE}(\mathbf{x}_0, \mathbf{c})\|_2 \tag{S3}$$

where $\mathbf{X}$ is a trajectory of the test system instance with free boundary condition. After solving Eq. (S3), the NODE parameterized by its solution $\mathbf{c}$ is capable of trajectory prediction given new initial conditions. The results are shown in Fig. S4(b).

S1.4 Kolmogorov-Petrovsky-Piskunov (KPP) equation: resolution-agnostic discovery with FNO backbones

This case tests whether the minimal conditioning principle holds when the encoder and decoder are both Fourier Neural Operators (FNOs), and whether the resulting operator family transfers across grid resolutions without retraining. Two questions follow. First, does the encoder, built on FNO layers, still discover the correct latent dimension on a resolution-agnostic representation? Second, does a latent inferred on a low-resolution trajectory parametrize the decoder correctly when the decoder is evaluated on a high-resolution grid? A positive answer to both would mean the principle is compatible with resolution generalization that label-conditioned methods cannot deliver without re-training.

The system governing equation is

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + ru(1-u) \tag{S4}$$

In this case, the training dataset contains 20 system instances, each taking a factor combination between $r = [0.01, 0.02, 0.03, 0.04, 0.05]$ and $D = [0.005, 0.010, 0.015, 0.020, 0.025]$.

The NODnet architecture and overall pipeline are shown in Figs. S5(a) and (b). A time sequence of the solution field $\{\mathbf{u}_j^c \in \mathcal{R}^{x_N}\}$ $(j = 1, 2, ..., N)$ is passed through a modified FNO paralleled by the spatial coordinates, where we keep the Fast Fourier Transform but discard the Inverse Fast Fourier Transform. The resulting vectors are passed through an LSTM to process the temporal evolution, outputting the latent vector $\mathbf{c} \in \mathcal{R}^2$, which acts as the system indicator. The decoder is also an FNO. Its input is a matrix formed by four vectors in $\mathcal{R}^{x_N}$: the spatial coordinate vector $\mathbf{x}$, the solution field

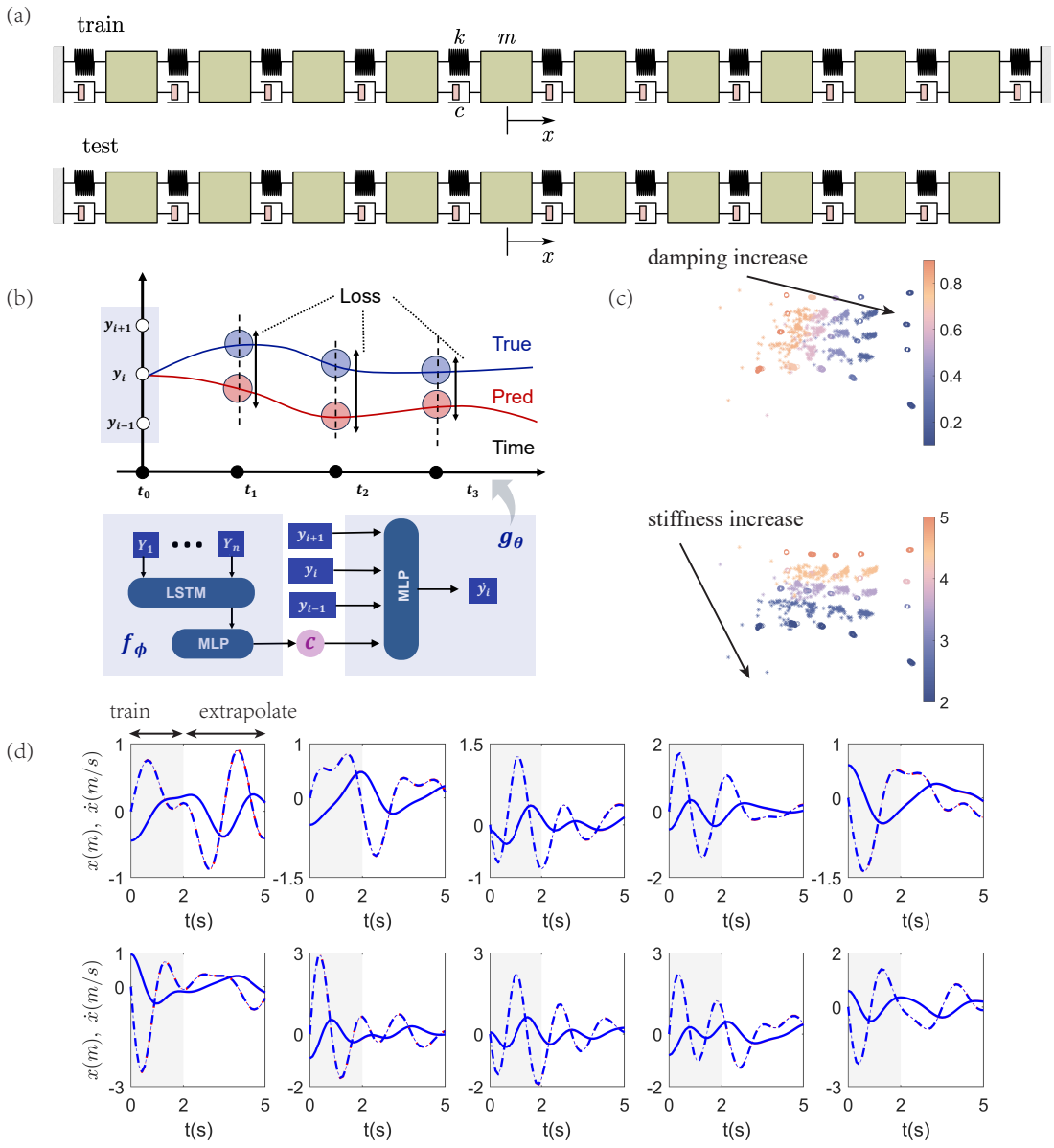

Figure S3: (a) An illustrative 10-DoF mass-spring chain system. We train with fixed boundary condition, test with free boundary condition on the right; (b) The NODnet architecture for the mass-spring-chain case. NODE is adopted as the decoder; (c) The learnt latent space. Clearly 2 principal directions exist corresponding to damping and stiffness increases of the physical system family; (d) Train and extrapolation results. With NODE as decoder, NODnet is able to perform accurate long-time rollouts outside of the training time span.

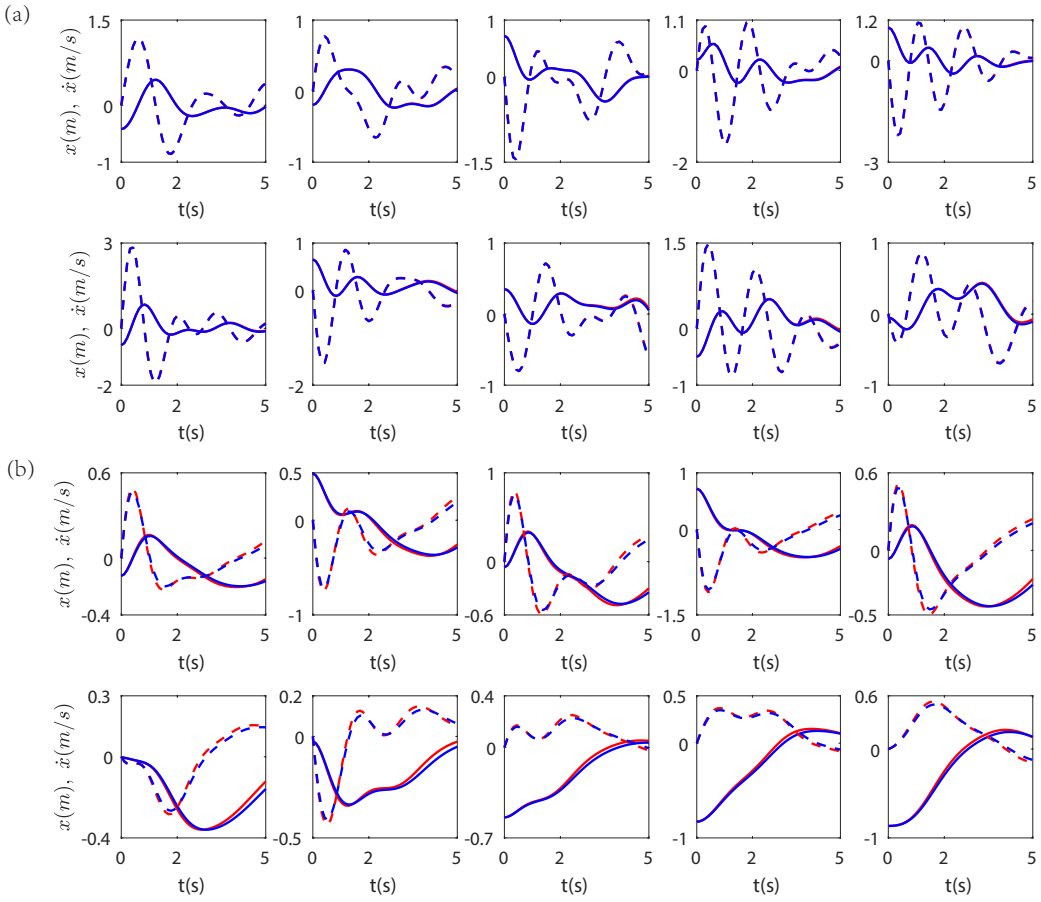

Figure S4: (a) Boundary condition generalization when conditioning data is of the same boundary condition as the training data. In this scenario, the encoder can be used to generate the latent $\mathbf{c}$; (b) Boundary condition generalization when conditioning data is of a different boundary condition from the training data. In this scenario, the encoder cannot be used. $\mathbf{c}$ has to be generated via solving an optimization problem (Eq. (S3)).

vector $\mathbf{u}_t$ at time t, and two copies of the latent vector $\mathbf{c}$ broadcast across the spatial grid. The output of decoder is the solution field at the next time step.

As shown in Fig. S5, we train NODnet on low-resolution grid and directly test on high-resolution grid. The 0-shot high-resolution prediction results are illustrated in Fig. S5(c), with small errors existing near the boundaries. The learnt latent space is shown in Fig. S5(d). Two principal directions exist corresponding to reaction and diffusion coefficients increases.

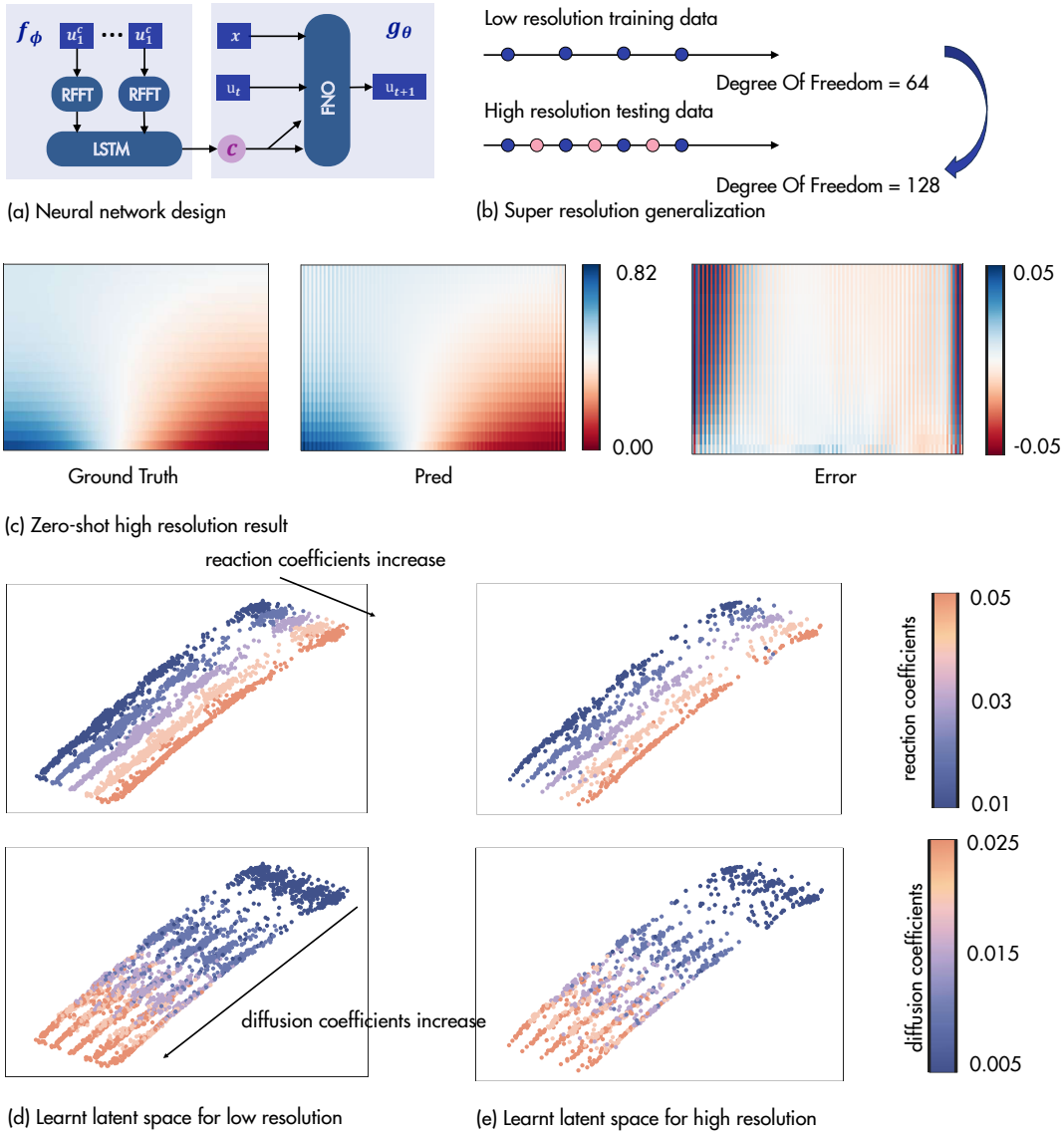

Figure S5: (a) The NODnet architecture for the KPP system case; (b) NODnet is trained on low-resolution grid and directly tested on high-resolution grid; (c) Zero-shot high-resolution test result; (d) The learnt latent space. Two principal directions exist corresponding to reaction and diffusion coefficients increases.

S2 Extended Results

This section reports detailed numerical results, ablations, and case-specific design rationale for the three main-text cases. Each subsection collects the simulation setup, decoder choice, latent-dimension

sweep, and split-by-split tables that the main text references but does not display in full.

S2.1 Details of Burgers' equation

Simulation details The dataset is generated by solving the one-dimensional viscous Burgers' equation $\partial_t u + u\partial_x u = \nu \partial_{xx} u$ on a periodic domain $\Omega = [-1, 1]$ with a spatial resolution of 4001 grid points. For the viscous cases ($\nu > 0$), an IMEX scheme is employed, with explicit central-difference advection with implicit spectral diffusion under adaptive CFL-constrained timestepping. For the inviscid case ($\nu = 0$), a Godunov scheme with MUSCL reconstruction (minmod limiter) and TVD Runge-Kutta 2nd-order time integration is used. The solution is integrated up to $t = 5$ s and sampled every 50 steps, yielding 101 temporal snapshots. The spatial output is downsampled to 401 points. The viscosity coefficient ν takes 16 values: $\nu = 0$ and $\nu \in 1, 2, 3, 4, 5 \times 10^{-4,-3,-2}$, giving 16 distinct simulation instances per initial condition.

Choice of decoder Two requirements shape the decoder choice. First, the decoder must supervise the encoder with signals from across the entire (x, t) field, not from the immediate next step: under shock dynamics the field is smooth almost everywhere and the shock front (where ν actually appears) occupies a small fraction of the domain, so one-step supervision is dominated by smooth-region MSE that varies little with ν and leaves the encoder no ν-dependent gradient. We therefore use a single-shot full-field decoder, namely DeepONet (branch-and-trunk MLP), shared between NODnet and PNO. Second, within the DeepONet the trunk must express the shock fronts at arbitrary continuous (x, t); coordinate-input MLPs smooth them out under their low-frequency spectral bias [5, 6], so we equip the trunk with sinusoidal positional encoding (PE).

Given the initial frame $\mathbf{u}_0$ and the latent c, the branch produces a latent vector $b(\mathbf{u}_0, c) \in \mathbb{R}^{512}$; the trunk takes the PE-encoded coordinates $\gamma(x, t) \in \mathbb{R}^{26}$ and produces a basis $\tau(\gamma(x, t)) \in \mathbb{R}^{512}$; the prediction is $\hat{u}(t, x) = u_0(x) + \alpha_{\text{res}} \langle b, \tau \rangle + b_{\text{out}}$, with α_{res} a learnable residual scale and b_{out} a learnable bias. Both branch and trunk are four-layer Softplus MLPs (hidden 800 / 1100 respectively). The PE follows the standard NeRF form [6]: each coordinate $c \in \{x, t\}$ is mapped to $\gamma(c) = [c, \sin(2^k \pi c), \cos(2^k \pi c)]_{k=0}^{L_c - 1}$, with L_x=8 matched to the spatial Nyquist of the 401-grid and L_t=4 matched to the temporal smoothness of Burgers' dynamics. Training uses u-MSE on 8 192 randomly sampled (t, x) queries per case. ANO is a separate architecture (1D U-Net autoregressive stepper with a four-frame past-state window) used as the LB and is described in Methods of the main text.

Both requirements leave direct empirical signatures (Table S2, Fig. S6). Removing PE preserves the supervision geometry but breaks the representational class: the trunk MLP smooths shocks, short-horizon prediction error grows by 4–5×, and the encoder's latent distribution collapses with all nine trained ν mapping to the same scalar within training noise. NODnet-AR (main text) preserves the representational class but breaks the supervision geometry: the same encoder and latent, but a U-Net autoregressive stepper supervises only one step at a time, where smooth-region MSE dominates the loss, and the latent again collapses (Fig. 2(d) of the main text). Both routes leave the encoder without a ν-dependent gradient. Encoder identifiability therefore requires both, and the DeepONet+PE combination provides both.

NODnet-AR architectural details NODnet-AR (reported alongside NODnet in Table 1 of the main text) shares NODnet's CNN encoder and uses an alternative decoder: a 1D U-Net autoregressive stepper conditioned via FiLM injection of the latent at each resolution level, $\hat{\mathbf{u}}_{t+1} = \mathbf{u}_t + g_\theta(\mathbf{u}_t, c)$, rolled out step-by-step at inference. The U-Net has 5 resolution levels with base-channels 44, matching the parameter count of the DeepONet decoder used by NODnet/PNO. PE does not apply to this decoder: the AR stepper operates on the discrete spatial grid rather than continuous coordinates. Training is u-MSE on adjacent $(\mathbf{u}_t, \mathbf{u}_{t+1})$ pairs with rollout depth 4. Both NODnet and NODnet-AR receive identical conditioning information through the same encoder; the comparison isolates the decoder architectural choice. The autoregressive choice is the source of the encoder collapse seen in main-text Fig. 2d: one-step AR supervision restricts the loss to local increments dominated by smooth-region MSE, eliminating the ν-dependent gradient the encoder needs to separate viscosities (mechanism detailed under *Choice of decoder* above).

Table S2: Removing-PE ablation on NODnet and PNO for Burgers' (L_2 relative error $\times 10^{-2}$ at 1-, 5-, and 50-step horizons). Same encoder, same loss, same training schedule; the only change is whether the trunk's (x, t) input is PE-encoded. All four rows are single-seed (seed=1234) since this is a focused mechanism ablation; the with-PE row is therefore the seed=1234 reference of Table 1's NODnet-DeepONet, the row whose multi-seed mean shifts modestly to 17.62/17.71/20.34 at H=50 (Table S3). The seed-pass does not change the qualitative gap to the no-PE row ($\sim 3\times$ at H=50) or the encoder-collapse signature in Fig. S6.

Model	Subset	1-step	5-step	50-step
NODnet (with PE)	ID	4.26±1.74	6.44±2.93	12.80±6.33
	OOD	4.37±1.67	6.24±2.67	12.42±6.48
	OOD-inviscid	4.59±1.34	8.30±3.68	16.90±5.24
NODnet (no PE, raw (x, t) trunk)	ID	20.36±5.89	27.86±8.54	43.11±26.34
	OOD	20.42±5.97	27.77±8.41	44.93±25.80
	OOD-inviscid	20.27±5.60	29.48±7.24	41.52±15.28
PNO$^{\uparrow}$ (with PE)	ID	3.89±1.74	6.04±2.58	13.43±6.39
	OOD	4.03±1.76	5.79±2.43	12.85±6.20
	OOD-inviscid	4.19±1.19	7.68±3.12	16.90±5.70
PNO$^{\uparrow}$ (no PE)	ID	14.69±4.61	21.94±8.01	30.91±14.09
	OOD	14.63±4.75	21.64±8.18	30.59±14.06
	OOD-inviscid	15.55±4.46	24.26±7.61	35.01±12.42

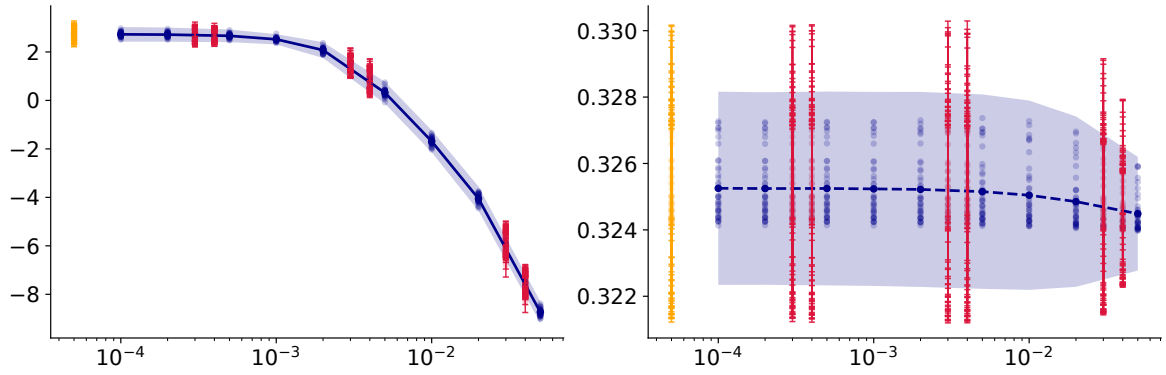

Figure S6: NODnet encoder latent distribution by viscosity, with and without PE on the DeepONet trunk (same CNN encoder, same u-only MSE, same training schedule on both panels). Left: with PE, the encoder learns a clean monotonic $\nu \to c$ curve (within-ν std small, inter-ν spread large), and the inviscid limit ν=0 extends the curve smoothly upward. Right: without PE, the same encoder collapses to a constant latent regardless of ν (the entire y-axis range is $\sim 10^{-3}$, comparable to within-ν noise). Removing PE thus disables both the decoder's shock representation and, via gradient feedback, the encoder's ν discrimination.

PNO calibration: $\sqrt{\nu}$ versus $\log\nu$ The PNO$^{\uparrow}$ replaces the learned latent with an analytic transform $c = \alpha\sqrt{\nu} + \beta$ of the true viscosity, with (α, β) calibrated so that the nine trained viscosities map to $[-2, +2]$ and the inviscid case ν=0 maps to $\beta \approx -2.19$, a finite latent just below the smallest trained latent and inside the smooth extension of the calibrated interval. The $\sqrt{\nu}$ form is chosen over the more common $\log\nu$: log sends ν=0 to $-\infty$ and forces an arbitrary ε-clamp that lands the inviscid latent far outside calibration, whereas $\sqrt{\nu}$ remains finite at the singular limit by construction. Inviscid prediction under either NODnet or $\sqrt{\nu}$-PNO is therefore reached by continuation along the calibrated latent curve, in contrast to the textbook $\log+\varepsilon$ encoding which sends ν=0 to a latent value never seen during training.

Latent-dimension sweep (supports Fig. 2(a)) The minimal-conditioning principle predicts that the latent dimension d should saturate at the number of independent governing factors, so d^*=1 for Burgers' since viscosity is the only governing factor. To probe both sides of the elbow, we train NODnet at $d \in \{0, 1, 2, 3, 4\}$ over three independent seeds $\{1234, 5678, 9012\}$ under identical training settings (same CNN encoder for $d\geq1$, same DeepONet+PE decoder, same u-only MSE, 200 epochs each); the d=0 run drops the encoder entirely and conditions the decoder only on the initial state, recovering the naive single-system neural-operator baseline that ignores per-instance variation. The expected pattern is two-sided: at $d<d^*$ the decoder receives no factor information and falls back to family-averaged dynamics; at $d>d^*$ test MSE surges because spare dimensions absorb initial-condition-specific structure that does not generalise, the structural-slack mechanism described in the Discussion of the main text. Numerical values are in Table S3. The two-stage rule from the main-text Methods is applied as follows. *Stage 1.* Seed-mean Train MSE values $2.25, 1.01, 1.42, 1.38, 1.02$ ($\times10^{-3}$) for $d = 0, \ldots, 4$ give step-wise relative changes $-55\%, +41\%, -3\%, -26\%$. The d=1→d=2 change of $+0.41$ is comparable to the d=2 across-seed envelope ($\sigma_s = 0.74$), so the candidate set is $\{1, 2\}$. *Stage 2.* The trained d=2 encoder's per-instance latent means populate a spurious second axis (PCA variance ratios with PC2 carrying within-instance shock-front noise rather than ν, mirroring the CFW d=2 pattern in Fig. S8), while d=1 produces a clean monotonic ν-curve (main-text Fig. 2d). Stage 2 therefore picks d^*=1. The non-monotonicity of long-horizon error past d^* (d=3 marginally below d=1 on H=50 within $\pm1\sigma_s$, Table S3) is consistent with structural slack scattering across multiple axes at d>2; the latent space verification rules out using long-horizon argmin as the selection criterion.

S2.2 Details of flow past a cylinder

Simulation details The dataset is a simulation on an irregular mesh on a 2D domain $\Omega = [-9, 21] \times [-6, 6]$ and boundary of the cylinder on a circle with radius 0.5 center at origin, with 159,720 points in total. The time span is 1.2 s, and integration timestep is 0.00001 s. Data is downsampled every 1000 steps, forming 120 steps. ν species a system, which takes the value of $[0.005, 0.008, 0.011, 0.014, 0.017, 0.020]$. For each ν, six u_∞ are simulated, which are $[15, 16, 17, 18, 19, 20]$. The dataset consists of 36 trajectories in total. There are 16 trajectories in the training dataset, which are $[\nu, u_\infty]_{\text{train}} = [0.005, 0.008, 0.014, 0.02] \times [15, 17, 19, 20]$; so the training data is limited. The first test set consists of 8 trajectories $[\nu, u_\infty]_{\text{test,1}} = [0.005, 0.008, 0.014, 0.02] \times [16, 18]$. It tests the model capability to generate correct $\boldsymbol{\eta}$ and correct prediction on unseen velocities for trained ν. The second test set consists of another 12 trajectories $[\nu, u_\infty]_{\text{test,2}} = [0.011, 0.017] \times [15, 16, 17, 18, 19, 20]$. It tests the model to generalize to unseen systems with different ν from the training set, given arbitrary velocities.

Formulation of SupCL

$$\ell_i = -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\big(\text{sim}(\mathbf{z}_i, \mathbf{z}_p)/\tau\big)}{\sum_{\substack{a=1 \\ a \neq i}}^{N} \exp\big(\text{sim}(\mathbf{z}_i, \mathbf{z}_a)/\tau\big)}, \tag{S5}$$

where $\text{sim}(\mathbf{z}_i, \mathbf{z}_j)$ is a similarity function. We employ L_2 distance as the similarity metric, a choice motivated by the simplicity of our one-dimensional latent space. In this formulation, τ denotes the temperature parameter and N represents the total number of samples within the batch. To gradually shift the optimization focus, we anneal the temperature over time, reducing the model's reliance on the supervised contrastive loss (SupCL) and transitioning toward the standard training objective. Additionally, we inject a small amount of random noise into the generated ϕ to mitigate mode collapse.

Table S3: Latent-dimension sweep on NODnet for Burgers' (L_2 relative error $\times 10^{-2}$ at H=1, 5, 50; last-50-epoch train MSE $\times 10^{-3}$). Cells are reported as $\mu \pm \sigma_{w \pm \sigma_s, s}$, where σ_w is within-eval std on the test cases (single-seed) and σ_s is across-seed std over $\{1234, 5678, 9012\}$. Identical training to the main-text NODnet row except for the latent dimension d; the d=0 row drops the encoder and conditions the decoder only on the initial state. Bold marks the lowest seed-mean in each column; the two-stage rule (Sec. S2.1 paragraph above) selects d^*=1 as the smallest plateau dimension passing the latent space verification.

d	Params	Subset	1-step	5-step	50-step	Train MSE
0	5.03M	ID	5.14±2.49 $_{\pm 0.21_s}$	12.78±9.15 $_{\pm 0.16_s}$	27.62±25.78 $_{\pm 0.99_s}$	2.25 $_{\pm 0.01_s}$
		OOD	5.38±2.47 $_{\pm 0.16_s}$	14.05±9.04 $_{\pm 0.08_s}$	30.35±25.83 $_{\pm 0.87_s}$	—
		OOD-inviscid	5.29±1.46 $_{\pm 0.29_s}$	12.90±4.83 $_{\pm 0.42_s}$	24.98±9.35 $_{\pm 0.76_s}$	—
1	10.08M	ID	**4.46±1.84** $_{\pm 0.31_s}$	**7.56±3.14** $_{\pm 1.76_s}$	17.62±9.90 $_{\pm 8.52_s}$	**1.01** $_{\pm 0.03_s}$
		OOD	**4.59±1.82** $_{\pm 0.32_s}$	**7.52±3.02** $_{\pm 2.05_s}$	17.71±10.54 $_{\pm 9.36_s}$	—
		OOD-inviscid	**4.74±1.39** $_{\pm 0.27_s}$	**9.17±3.83** $_{\pm 1.44_s}$	20.34±6.99 $_{\pm 5.74_s}$	—
2	10.08M	ID	4.82±2.10 $_{\pm 0.38_s}$	9.58±5.23 $_{\pm 2.85_s}$	21.67±15.56 $_{\pm 6.47_s}$	1.42 $_{\pm 0.74_s}$
		OOD	4.99±2.09 $_{\pm 0.42_s}$	9.97±5.10 $_{\pm 3.64_s}$	22.56±15.75 $_{\pm 7.97_s}$	—
		OOD-inviscid	5.04±1.46 $_{\pm 0.33_s}$	10.67±4.31 $_{\pm 2.05_s}$	22.69±8.40 $_{\pm 3.75_s}$	—
3	10.08M	ID	4.69±2.01 $_{\pm 0.24_s}$	8.29±4.07 $_{\pm 1.54_s}$	**17.18±10.84** $_{\pm 3.29_s}$	1.38 $_{\pm 0.63_s}$
		OOD	4.88±2.00 $_{\pm 0.32_s}$	8.53±4.04 $_{\pm 1.91_s}$	**17.39±11.38** $_{\pm 3.54_s}$	—
		OOD-inviscid	4.92±1.41 $_{\pm 0.08_s}$	9.55±4.03 $_{\pm 0.93_s}$	**19.55±6.65** $_{\pm 2.09_s}$	—
4	10.08M	ID	4.89±1.97 $_{\pm 0.22_s}$	9.15±3.81 $_{\pm 1.04_s}$	22.46±13.77 $_{\pm 2.40_s}$	1.02 $_{\pm 0.02_s}$
		OOD	5.01±1.97 $_{\pm 0.23_s}$	9.15±3.70 $_{\pm 0.99_s}$	22.88±14.36 $_{\pm 2.16_s}$	—
		OOD-inviscid	5.11±1.51 $_{\pm 0.18_s}$	10.47±4.26 $_{\pm 0.91_s}$	23.60±9.25 $_{\pm 2.17_s}$	—

Rollout augmentation We train on 2 step rollout result (1 step prediction with 1 step rollout $\mathbf{x}_{t+2} = \mathbf{f}^{(2)}(\mathbf{x}_t, \boldsymbol{\eta})$) to increase the training stability.

Drag force computation

$$\begin{aligned} \boldsymbol{\sigma} &= -p\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^\top), \\ F_D &= \int_{\partial\Omega_c} \boldsymbol{\sigma}\mathbf{n} \cdot \hat{\mathbf{e}}_x, ds, \\ &= -\int_{\partial\Omega_c} p n_x ds + \mu \int_{\partial\Omega_c} \left[(\nabla\mathbf{u} + \nabla\mathbf{u}^\top)\mathbf{n}\right]_x ds. \end{aligned} \tag{S6}$$

In our dataset, velocity predictions are available on scattered spatial points. To compute surface forces consistently across ground truth, NODnet, and Baseline, the scattered velocity field is interpolated onto a narrow band of points surrounding the cylinder surface to approximate velocity gradients and surface normals; since pressure is not directly predicted, we recover it by solving the pressure Poisson equation derived from incompressible Navier–Stokes with Neumann boundary condition on the cylinder surface consistent with the normal momentum balance. This ensures that the reconstructed pressure field yields physically consistent surface traction. This procedure ensures that drag evaluation is consistent with the governing equations and avoids biases arising from missing pressure information.

Strouhal number computation The Strouhal number $\mathrm{St} = fD/u_\infty$ (with f the vortex shedding frequency, D the cylinder diameter, u_∞ the inlet velocity) is the dimensionless analogue of the shedding frequency in the laminar wake. We extract f from the lift coefficient time series $C_L(t)$ via FFT and take the peak frequency in the band consistent with vortex shedding ($\sim 0.2\, u_\infty/D$ at the Reynolds numbers considered here). C_L is computed from the same surface integral as C_D but along the transverse direction $\hat{\mathbf{e}}_y$. Together, $\bar{C}_D$ and St characterize the two canonical features of the laminar wake: the time-averaged momentum transfer and the periodic vortex emission. A predictor that recovers both is reproducing the dynamics; a predictor that drifts in either is failing in a physically interpretable way.

Architectural inductive bias We adopt the same autoregressive encoder and decoder setting as the FN-RD system. Our parametric studies (4-step rollout averaged L_2 prediction error, Table S4) indicate that a "Heavy-Encode" architecture, in which the encoder f_ϕ has significantly higher capacity than the decoder g_θ, yields superior training stability and generalization. This is consistent with the minimal conditioning principle: identifying which system one is observing is the more demanding task, while evolving the state, once the governing factors are compactly available, is relatively simpler. We additionally use a multi-step rollout augmentation strategy, training on two-step rollout outputs ($\mathbf{x}_{t+2} = g_\theta^{(2)}(\mathbf{x}_t, c)$, $\hat{\mathbf{y}} = (\mathbf{x}_{t+1}, \mathbf{x}_{t+2})$), to enforce temporal stability and mitigate numerical drift during long-horizon inference.

Table S4: Architecture parametric experiments for the cylinder flow wake case. 4-step rollout averaged L_2 prediction error across train and test sets.

Mode	Parameter	Train	Unseen u_∞	Unseen ν	Performance
Balanced	23M	**0.023**	0.084	0.042	Overfit
Heavy Encode	**15M**	0.027	**0.030**	**0.028**	**Generalize**
Heavy Decode	42M	0.025	0.089	0.052	Overfit

Latent-dimension sweep The cylinder wake is parameterized by viscosity ν and inlet velocity u_∞, but at the Reynolds numbers considered here ($u_\infty \in [15, 20]$, $\nu \in [0.005, 0.020]$) the wake dynamics are governed effectively by the single dimensionless group $\mathrm{Re} = u_\infty D/\nu$. The minimal-conditioning principle therefore predicts d^*=1. To probe both sides of the elbow, we evaluate the heavy-encode model with $d \in \{0, 1, 2, 3, 4\}$ on the same Train / Unseen-u_∞ / Unseen-ν splits used in Table S4, at horizons $H \in \{1, 5, 50\}$, with three independent training seeds $\{1234, 5678, 9012\}$ at every d. The d=0 run drops the encoder (3.46M params) and conditions the decoder only on the initial state, recovering the naive single-system neural-operator baseline. For $d \geq 1$ we use the same encoder/decoder shape and ∼15M parameter budget; only latent dimension d differs (contributing ∼1k parameters per increment). The training schedule uses a warmup+cosine latent-noise schedule (constant $5{\times}10^{-2}$ for the first 200 epochs, then cosine decay to 0 over the remaining 800 epochs; $1{\times}10^{-1}$ peak for d=3 to overcome an encoder-collapse failure mode at the lower noise level), and at d=1 the supervised-contrastive weight is downweighted to 0.3×rep_force because the full-strength contrastive loss is geometrically over-constrained on a 1-D latent (36 system labels cannot be maximally separated on a line). Numerical values are in Table S5 (the same sweep is visualised in main-text Fig. 3a).

Why CFW needs a long-horizon (or OOD) gauge. Unlike Burgers' and FN-RD, the 1-step Train L_2RE for CFW does not elbow at the predicted d^*=1: seed-mean values $1.47, 1.69, 2.49, 1.91, 2.07$ ($\times 10^{-2}$) for $d = 0, \ldots, 4$ give step-wise relative changes +15%, +47%, −23%, +8%, with d=0 the lowest. This is the encoder-free overfit signature: at d=0 the decoder receives a 19 M-parameter budget free of any latent-conditioning bottleneck and absorbs all family variability into per-step memorisation of the training distribution, beating d=1 on the very horizon at which the bottleneck has the least room to pay off. The cost of that flexibility surfaces as soon as the model has to roll out (errors compound system-agnostically) or generalise (no latent to interpolate over): on Train H=50 the seed-means are $36.09, 29.89, 39.40, 38.42, 31.52$ (step-wise −17%, +32%, −2%, −18%), and on OOD H=5 they are $6.11, 4.38, 11.30, 9.17, 7.88$ (step-wise −28%, +158%, −19%, −14%). Both long-horizon gauges minimise at d=1.

Applying the two-stage rule. *Stage 1.* On the Train H=50 gauge, d=1 is the minimum and d=1→d=2 jumps by +32% ($\sim 9.5 \times \sigma_s$, well outside seed envelope), so the candidate set is $\{1, 2\}$. *Stage 2.* The trained d=2 encoder populates a clear second axis (PCA variance ratios $(0.74, 0.26)$, Fig. S8), absorbing within-system / IC-window variability rather than carrying a second governing factor; the d=1 encoder produces a clean monotonic ν-curve with signal-to-noise $\sim$ 36 (Fig. S7). Stage 2 selects d^*=1.

Three patterns are consistent with the principle. (i) d=0 wins short-horizon ID (H=1, 5) and OOD-H=1 through encoder-free overfit, but d=1 wins all four long-horizon and OOD-H=5 cells: the encoder bottleneck pays off precisely at the rollout horizons where slack matters. (ii) d=0 at H=50 on OOD (34.5%) sits below the entire d>1 curve and within the seed envelope of d=1 (31.0%): the structural-slack mechanism in compact form. d=0 has no pathway for IC-specific noise to enter the

latent and drift the rollout, whereas d>1 provides exactly such a pathway through its spare dimensions, so at long horizons "no encoder" beats "encoder with slack". (iii) The d>1 curve is non-monotonic but strictly above d=1 on Train H=50, with d=2 the worst (OOD-H=5 rises by a factor of ~ 2.6 over d=1, the "surge past sufficient capacity" signature seen in Burgers'), while d=3, 4 partially recover at long horizons by spreading the spare dimensions into a near-degenerate submanifold but never beat d=1 on the long-horizon Train metric.

Table S5: Latent-dimension sweep on the heavy-encode CFW model (L_2 relative error $\times 10^{-2}$, at H=1, 5, 50). Cells reported as $\mu \pm \sigma_{w \pm \sigma_s, s}$; σ_w is within-eval std on the test cases (single-seed) and σ_s is across-seed std over $\{1234, 5678, 9012\}$. ID = Train split; OOD = Unseen-u_∞ and Unseen-ν pooled. **Bold marks the lowest seed-mean in each column**: d=0 wins short-horizon ID through encoder-free overfit; d=1 wins long-horizon and OOD-H=5 — the elbow that motivates d^*=1.

d	Params	Subset	H=1	H=5	H=50
0 (no encoder)	3.46M	ID	**1.47±0.59** $_{\pm 0.29_s}$	**3.46±1.76** $_{\pm 0.81_s}$	36.09±14.17 $_{\pm 5.43_s}$
		OOD	**1.63±0.75** $_{\pm 0.39_s}$	6.11±3.76 $_{\pm 0.18_s}$	34.53±13.43 $_{\pm 3.01_s}$
1	15.24M	ID	1.69±0.79 $_{\pm 0.31_s}$	4.00±2.60 $_{\pm 1.23_s}$	**29.89±16.61** $_{\pm 6.11_s}$
		OOD	1.72±0.86 $_{\pm 0.36_s}$	**4.38±2.34** $_{\pm 0.45_s}$	**31.04±15.11** $_{\pm 4.51_s}$
2	15.24M	ID	2.49±0.97 $_{\pm 0.47_s}$	8.75±5.47 $_{\pm 4.77_s}$	39.40±17.01 $_{\pm 7.35_s}$
		OOD	2.84±1.06 $_{\pm 0.60_s}$	11.30±6.13 $_{\pm 5.41_s}$	38.33±13.42 $_{\pm 6.36_s}$
3	15.24M	ID	1.91±0.74 $_{\pm 0.43_s}$	5.54±2.94 $_{\pm 1.86_s}$	38.42±15.40 $_{\pm 6.17_s}$
		OOD	2.78±1.37 $_{\pm 0.91_s}$	9.17±5.64 $_{\pm 2.08_s}$	34.54±13.67 $_{\pm 4.36_s}$
4	15.24M	ID	2.07±0.87 $_{\pm 0.36_s}$	4.67±2.24 $_{\pm 1.44_s}$	31.52±14.61 $_{\pm 6.93_s}$
		OOD	2.58±1.25 $_{\pm 0.72_s}$	7.88±4.97 $_{\pm 2.11_s}$	33.03±14.36 $_{\pm 3.30_s}$

Why does d>1 regress? Latent-space inspection Figures S7, S8, and S9 render the trained d=1, d=2, and d=3 encoders evaluated on all 36 cylinder-wake systems (16 train + 8 unseen-u_∞ + 12 unseen-ν, with 16 conditioning-window draws per system). The right panels colour each per-system mean by ν. The d=1 encoder lays out the 36 systems as six tight clusters with one cluster per viscosity, monotonically ordered along the latent axis from low to high ν (inter-system range 2.10 vs. mean within-system std 0.06, signal-to-noise ratio ∼36); the six u_∞ within each ν collapse to a single point up to noise, confirming that the encoder has discovered a single governing dimension dominated by ν in this Re regime. PCA on the 36 system means then yields:

- d=2: variance ratios $(0.739, 0.261)$. PC2 retains over a quarter of the total variance — nowhere near the empty-axis degeneracy seen in FN-RD's d=3 run (variance ratios $(0.80, 0.20, \mathbf{0.00})$, Sec. S2.3). Both available dimensions are actively populated.
- d=3: variance ratios $(0.685, 0.231, 0.083)$. Even the third PC carries an 8% share, so all three dimensions remain populated; the encoder does not spontaneously collapse onto a 1-D submanifold.

This is the empirical signature of the structural-slack mechanism. The CFW case is parameterized by a single dominant dimensionless group ($\mathrm{Re} = u_\infty D/\nu$, hence d^*=1), but neither the SupCL contrastive force nor the rollout-MSE objective penalizes the extra latent axes for being non-empty. Unlike FN-RD, CFW lacks a strong O(2) augmentation that would multiply each system into many equivalent draws and pull degenerate axes to zero during training; the only within-system variability the encoder sees is the conditioning-window subsampling. Consequently, when given two or three latent dimensions, the encoder is free to spread within-system noise and IC-window fluctuations into the slack axes, producing the populated PC2/PC3 we observe. At inference time the decoder must integrate this extra non-governing structure across the 50-step rollout, which is exactly the OOD-degradation pattern in Table S5: the more slack the encoder is given to fill, the further the trajectory drifts from the true-Re flow. The non-monotonicity (d=2 being worst, d=3, 4 partially recovering) is also consistent with this picture: at d=3, 4 the slack is divided across multiple axes so that no single axis carries the full burden

of IC absorption (PC3 contribution 0.083, PC2 contribution drops from 0.261 at d=2 to 0.231 at d=3), reducing the per-axis impact on rollout but not eliminating it.

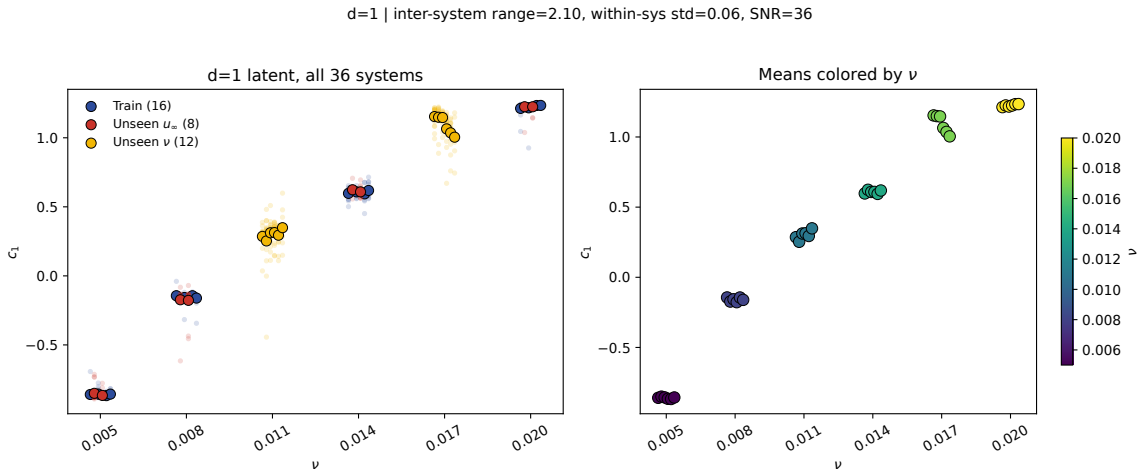

Figure S7: d=1 latent representation of all 36 CFW systems (the original heavy-encode checkpoint reported in the main text). **Left:** ν vs. c_1 scatter coloured by split (blue: 16 train, red: 8 unseen-u_∞, orange: 12 unseen-ν); faint dots are 16 conditioning-window draws per system, solid dots are per-system means. The u_∞ axis is shown as a small horizontal jitter inside each ν column. **Right:** the same per-system means, coloured by ν, on a viridis colourbar. The encoder produces a clean monotonic $\nu \to c_1$ map with $\sim$36$\times$ inter-system / within-system separation, and the six u_∞ values within each ν collapse to one point — the discovered 1-D latent aligns with ν in this Re regime, contrast directly with the populated slack axes in Figs. S8, S9.

What the SNO/NODnet convergence at H=50 on OOD does and does not imply. The seed-mean comparison d=1 vs. d=0 across horizons is asymmetric. At H=1, 5 on ID and H=1 on OOD, d=0 is at or below d=1: the encoder-free baseline absorbs all family variability into per-step memorisation of the training distribution, an overfit that the bottleneck at d^* cannot match by construction. The advantage flips at OOD-H=5 (d=1 at 4.38% vs. d=0 at 6.11%, a 28% relative reduction) and at long horizons on both splits (d=1 at 29.9% ID-H=50 vs. d=0 at 36.1%, a 17% reduction; OOD-H=50 within seed envelope). Two readings support the structural-slack interpretation. *First*, the asymmetric ordering — d=0 winning short-horizon ID and d=1 winning long-horizon and OOD-H=5 — is exactly what the bottleneck-as-regulariser claim predicts: d=0 has no factor-aware latent so it cannot interpolate across instances, but it also has no slack axis for IC noise to drift the rollout, so it catches up at H=50 through the absence of slack rather than the presence of factor information. *Second*, NODnet's drag and Strouhal-number recovery at Re=300 (main-text Fig. 3f) remains physically valid where SNO produces family-averaged drag with no per-instance shedding frequency. The rollout-error metric saturates at long horizons; the physical-observable metric continues to separate the two.

S2.3 Details of FHN reaction-diffusion network

Simulation details The dataset is a simulation on a uniform 2D mesh grid in a domain $\Omega = [0, 1]^2$ with $\Delta t = 0.001$ s up to $t = 10$ s and sampled every 100 timesteps, using finite difference method with the 4th-order Runge-Kutta time integration. k and β are tuned to model different neural activities, with k ranging from 0.01 to 0.05 and β ranging from 0.1 to 0.3, giving 25 distinct system instances.

Design of hierarchical mask The prediction rule applies smaller-lead-time steppers first, which suits a diffusion-reaction system whose transient phase happens early. This contrasts with the weather-forecasting setting [7], where predictions over different horizons can be issued in any order without compounding the early-transient error.

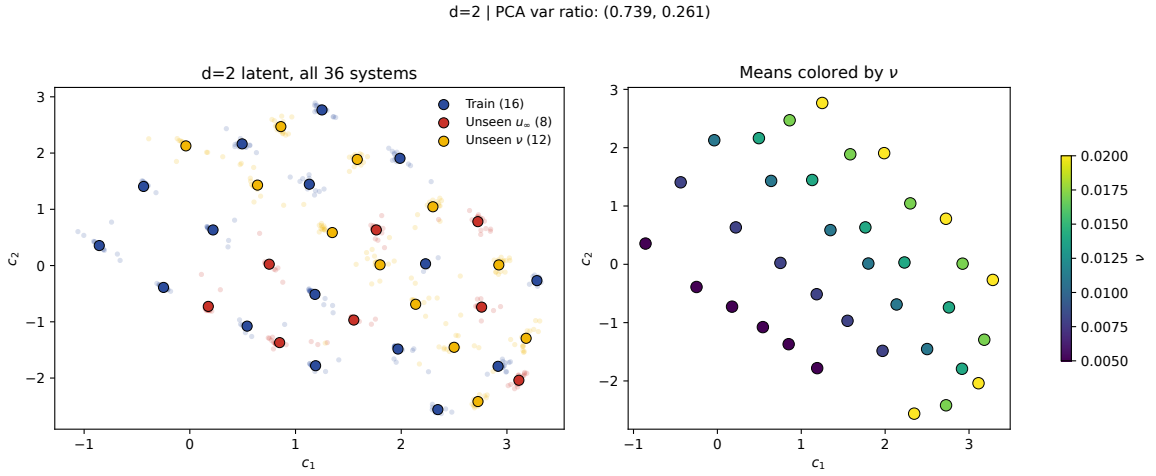

Figure S8: d=2 latent representation of all 36 CFW systems. **Left:** 2-D scatter coloured by split (blue: 16 train, red: 8 unseen-u_∞, orange: 12 unseen-ν), faint dots are 16 conditioning-window draws per system, solid dots are per-system means. **Right:** per-system means coloured by viscosity ν. PCA on the 36 means yields variance ratios $(0.739, 0.261)$: the second axis is clearly populated, in contrast to the empty-axis collapse seen for FN-RD when given more dimensions than d^*. This is the structural-slack signature explaining the H=5 rollout surge in Table S5.

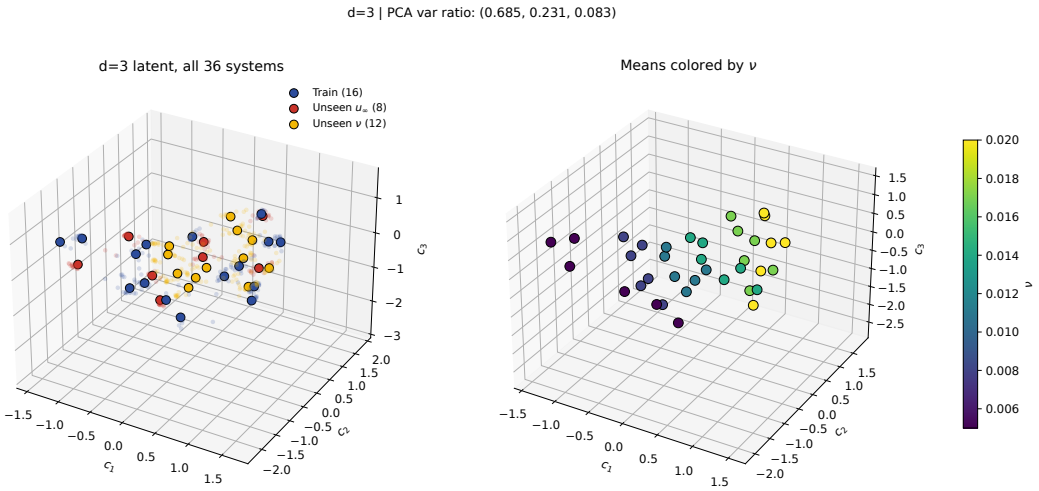

Figure S9: d=3 latent representation of all 36 CFW systems. Layout matches Fig. S8: **left** coloured by split, **right** per-system means coloured by ν. PCA on the 36 means yields variance ratios $(0.685, 0.231, 0.083)$. Unlike the FN-RD d=3 run (Sec. S2.3, variance ratios $(0.80, 0.20, 0.00)$, where the trained encoder discovers a 2-D submanifold matching d^*=2), the CFW encoder uses all three available axes; the third axis still holds 8% of the inter-system variance. The slack is spread rather than collapsed — consistent with d^*=1 being the only setting that recovers the original heavy-encode performance.

Data augmentation To enhance the model's capability to generalize across initial conditions without enlarging the simulation budget, we apply data augmentation to replicate trajectories. The governing equations together with periodic boundary conditions exhibit O(2) equivariance: any spatiotemporal trajectory transformed via the O(2) action remains a valid solution of the same FN-RD system instance. We therefore augment each trajectory by shifting to the four corners, mirroring horizontally and vertically, and rotating by 90°, 180°, 270°, yielding a 10× enlargement of the effective dataset. This enforces the encoder to identify factors that are invariant to spatial orientation while keeping the U-CNN decoder robust across diverse initial states.

Latent-dimension sweep The minimal-conditioning principle predicts d^*=2 for FN-RD, since (k, β) are the two factors that vary across systems. To probe both sides of the elbow, we train the hierarchical NODnet with $d \in \{0, 1, 2, 3, 4\}$ over three independent seeds $\{42, 5678, 9012\}$ under identical training settings (same encoder for $d \geq 1$, same hierarchical decoder with $\{1, 3, 9\}$-step modules, same MSE-on-rollout objective, same ∼650 epochs of data-augmented training); the d=0 run drops the encoder entirely, recovering the naive single-system neural-operator baseline that ignores per-instance variation. Numerical values are in Table S6. Applying the two-stage rule from the main-text Methods. *Stage 1.* Seed-mean Train MSE values $30.87, 6.92, 5.26, 4.74, 5.06$ ($\times 10^{-5}$) for $d = 0, \ldots, 4$ give stepwise relative changes $-78\%, -24\%, -10\%, +7\%$. The d=3→d=4 change of +0.32 is comparable to the across-seed envelope ($\sigma_s = 0.35$ at d=4), and the d=2→d=3 change of −0.52 is at the boundary of the envelope ($\sigma_s = 0.34$ at d=2, 0.12 at d=3); the candidate set is therefore $\{2, 3\}$. *Stage 2.* The trained d=3 encoder spontaneously collapses to a 2-D submanifold, with PCA on per-instance latent means returning variance ratios $(0.80, 0.20, 0.00)$ — the third axis is empty to numerical precision (Sec. S2.3, Fig. S10). Stage 2 selects d^*=2. The failure modes below d^* are graded on the long-rollout extrapolation split (H=50): d=0 collapses worst (L_2RE 9.31±2.28, no encoder so no factor information at all), d=1 collapses partially (L_2RE 7.87±4.48, a 1-D latent cannot resolve the (k, β) pair), and $d \geq 2$ are within a factor of 1.5 of each other with d=3, 4 marginally lower on Train and OOD-Intra metrics.

Table S6: Latent-dimension sweep on the hierarchical NODnet for FN-RD (L_2 relative error $\times 10^{-2}$ at H=1, 5, 50; last-50-epoch train MSE $\times 10^{-5}$). Cells are reported as $\mu \pm \sigma_{w \pm \sigma_s, s}$, where σ_w is within-eval std on the test cases (single-seed) and σ_s is across-seed std over $\{42, 5678, 9012\}$. Splits: Train (= ID), Interpolated (= OOD-Intra), Extrapolated (= OOD-Extra). Identical training to the main-text NODnet-Hier row except for the latent dimension d; the d=0 row drops the encoder. **Bold marks the lowest seed-mean in each column**; the two-stage rule (Sec. S2.3 paragraph above) selects d^*=2 as the smallest plateau dimension passing the latent space verification (the d=3 encoder spontaneously collapses to a 2-D submanifold).

d	Params	Subset	1-step	5-step	50-step	Train MSE
0	2.518M	Train	1.31±0.09 $_{\pm 0.01_s}$	1.78±0.09 $_{\pm 0.03_s}$	7.33±1.62 $_{\pm 0.21_s}$	30.87 $_{\pm 2.73_s}$
		Interpolated	1.28±0.06 $_{\pm 0.01_s}$	1.75±0.07 $_{\pm 0.03_s}$	7.05±1.22 $_{\pm 0.26_s}$	—
		Extrapolated	1.38±0.11 $_{\pm 0.02_s}$	1.96±0.16 $_{\pm 0.04_s}$	9.31±2.27 $_{\pm 0.21_s}$	—
1	3.108M	Train	0.94±0.05 $_{\pm 0.03_s}$	1.25±0.09 $_{\pm 0.10_s}$	2.98±1.24 $_{\pm 1.42_s}$	6.92 $_{\pm 0.60_s}$
		Interpolated	0.94±0.05 $_{\pm 0.03_s}$	1.29±0.11 $_{\pm 0.13_s}$	4.46±1.61 $_{\pm 1.22_s}$	—
		Extrapolated	1.13±0.18 $_{\pm 0.07_s}$	1.76±0.46 $_{\pm 0.11_s}$	7.87±4.48 $_{\pm 1.20_s}$	—
2	3.108M	Train	0.91±0.05 $_{\pm 0.02_s}$	1.20±0.09 $_{\pm 0.04_s}$	2.52±0.66 $_{\pm 0.07_s}$	5.26 $_{\pm 0.34_s}$
		Interpolated	0.90±0.03 $_{\pm 0.02_s}$	1.19±0.07 $_{\pm 0.05_s}$	3.05±0.91 $_{\pm 0.21_s}$	—
		Extrapolated	1.03±0.16 $_{\pm 0.10_s}$	1.61±0.50 $_{\pm 0.43_s}$	5.49±3.34 $_{\pm 2.84_s}$	—
3	3.109M	Train	0.89±0.05 $_{\pm 0.02_s}$	**1.13±0.09** $_{\pm 0.03_s}$	**2.05±0.49** $_{\pm 0.07_s}$	**4.74** $_{\pm 0.12_s}$
		Interpolated	0.87±0.03 $_{\pm 0.01_s}$	**1.13±0.06** $_{\pm 0.03_s}$	**2.50±0.68** $_{\pm 0.28_s}$	—
		Extrapolated	1.01±0.12 $_{\pm 0.07_s}$	1.48±0.32 $_{\pm 0.25_s}$	4.96±2.83 $_{\pm 2.59_s}$	—
4	3.110M	Train	**0.88±0.04** $_{\pm 0.01_s}$	1.15±0.08 $_{\pm 0.01_s}$	2.55±0.72 $_{\pm 0.24_s}$	5.06 $_{\pm 0.35_s}$
		Interpolated	**0.87±0.03** $_{\pm 0.01_s}$	1.16±0.06 $_{\pm 0.01_s}$	3.15±0.77 $_{\pm 0.27_s}$	—
		Extrapolated	**0.98±0.15** $_{\pm 0.05_s}$	**1.38±0.34** $_{\pm 0.15_s}$	**4.13±2.38** $_{\pm 1.54_s}$	—

The marginal d=3 advantage Table S6 shows that d=3 is marginally lower than d=2 on every long-rollout split. The principled gauge above already identifies d^*=2 as the smallest plateau dimension; three observations explain why d=3 remains marginally lower without contradicting that gauge.

(i) The plateau dimensions show only marginal differences. Seed-mean Train MSE at d=2, 3, 4 ($5.26, 4.74, 5.06 \times 10^{-5}$) varies by $\sim 10\%$ across d, smaller than the d=1→d=2 elbow drop ($\sim 24\%$). On long-horizon test, the 50-step interpolation seed-means 3.05 (d=2) and 2.50 (d=3) overlap within seed envelope ($\sigma_s = 0.21$ at d=2, 0.28 at d=3). The marginal d=3 advantage is one realisation of a flat plateau.

(ii) Optimization slack at d=3. The (k, β)→c map is genuinely curved (quadratic ≻ affine for the zero-shot diffeomorphism, Table S7). A 2-D latent must embed that curve exactly, while a 3-D ambient gives the optimizer a smoother landscape to find the same 2-D submanifold; d=3 buys optimization slack at no representational cost.

(iii) Structural slack appears past the plateau, at d=4. Train MSE worsens from 4.74 (d=3) to 5.06 (d=4), the first upward step in the sweep, and the 50-step interpolation column tracks the same trend (2.50 → 3.15). The shift is small in absolute terms because the 10× O(2) data augmentation acts as an additional regularizer that suppresses the structural-slack signature within $d \leq 3$, but the direction is consistent with the Burgers'-style "test surges past sufficient capacity" pattern.

Empirical confirmation that the d=3 representation is intrinsically 2-D. Figure S10 renders the per-system means of the trained d=3 model in its 3-D ambient space. PCA on the 25 system-mean latents returns singular values $\sigma = (0.4003, 0.1996, 0.0022)$, with σ_3 two orders of magnitude below σ_1, and the corresponding variance ratios $\sigma_i^2 / \sum_j \sigma_j^2 = (0.8009, 0.1991, \mathbf{0.0000})$, with the third entry below numerical resolution. Within numerical noise the d=3 model has discovered a strictly 2-D submanifold inside its 3-D latent space. This is the cleanest possible reconciliation with the principle: the model sees three available dimensions, but uses two — exactly d^* — with the third dimension carrying no signal. The marginal performance gain of d=3 over d=2 is the optimization-slack effect described in (ii), not a representational one.

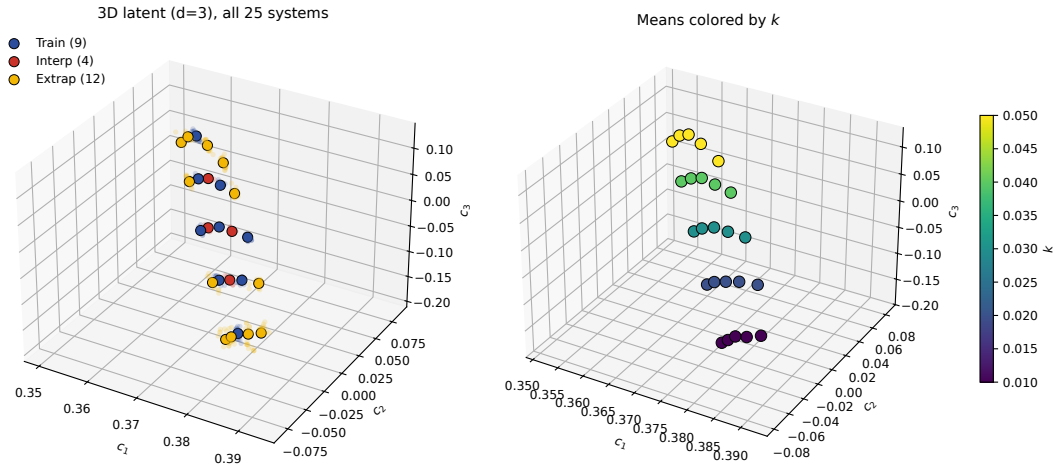

Figure S10: d=3 latent representation of all 25 FN-RD systems. **Left:** 3-D scatter in latent space, coloured by split (blue: 9 train SIs, red: 4 interpolated SIs, orange: 12 extrapolated SIs). Faint markers are 20 init+augmentation draws per system, solid markers are per-system means. **Right:** per-system means coloured by k, showing a smooth monotonic factor sweep along the manifold. PCA on the 25 means yields variance ratios $(0.8009, 0.1991, 0.0000)$: the third dimension is empty up to numerical noise, so the trained d=3 model has spontaneously embedded a 2-D submanifold (exactly d^*=2) inside its 3-D ambient.

Zero-shot generalization: effect of diffeomorphism order The zero-shot mode described in the main text uses a fitted polynomial diffeomorphism to map factors directly to latents for unseen

systems. The fitted quadratic diffeomorphism used in the main text takes the explicit form

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} -0.08 & -3.76 & 0.75 & 12.75 & -4.49 & -0.75 \\ -0.22 & 2.01 & 0.57 & -1.53 & 0.68 & -0.78 \end{bmatrix} \begin{bmatrix} 1 \\ k \\ \beta \\ k^2 \\ k\beta \\ \beta^2 \end{bmatrix} \tag{S7}$$

fitted by least squares on the 9 training SIs.

To assess whether the nonlinear structure of this map is essential, we compare polynomial diffeomorphisms of increasing order, all fitted on the same 9 training SIs via least-squares:

- **Order 1 (affine):** Features $[1, k, \beta]$ (3 coefficients per latent dimension).
- **Order 2 (quadratic):** Features $[1, k, \beta, k^2, k\beta, \beta^2]$ (6 coefficients; Eq. S7 above).
- **Order 3 (cubic):** Features up to k^3, $k^2\beta$, $k\beta^2$, β^3 (10 coefficients—underdetermined with 9 training points, so ridge regularization is applied).
- **One-shot:** The encoder f_ϕ maps a single test trajectory to the latent space. No factor values are required; included as a reference baseline.

We evaluate all four approaches on the 16 held-out FN-RD systems (4 interpolated, 12 extrapolated), measuring L_2 relative error over 50-step rollouts (Table S7).

Table S7: Zero-shot generalization on FN-RD (L_2 relative error, 50-step rollout) as a function of diffeomorphism polynomial order. One-shot is included as a non-zero-shot reference.

Method	Interpolated	Extrapolated	Overall
Order 1 (affine)	0.0165±0.0047	0.0219±0.0080	0.0205±0.0077
Order 2 (quadratic)	0.0151±0.0045	**0.0188±0.0056**	0.0179±0.0056
Order 3 (cubic)	**0.0146±0.0043**	0.0189±0.0054	**0.0178±0.0055**
One-shot	0.0285±0.0106	0.0239±0.0073	0.0250±0.0085

Results are summarized in Table S7 and visualized in main-text Fig. 4d. The affine map already achieves reasonable zero-shot accuracy, but the quadratic diffeomorphism reduces overall error by 13% (0.0179 vs. 0.0205), confirming that the nonlinear cross-terms k^2, $k\beta$, β^2 capture meaningful structure in the factor-to-latent mapping. Adding cubic terms yields negligible further improvement (0.0178), and the fit becomes underdetermined with only 9 training points, suggesting that the quadratic form already captures the essential geometry. The one-shot baseline (0.0250) is notably worse, with the gap attributable to the generalization limit of the encoder when applied to out-of-distribution systems.

We further visualize the decoder output when sampling latents on a uniform grid spanning the training-set latent space (Fig. S11). The decoder produces physically plausible dynamics at every sampled location, confirming that the learned manifold supports continuous, smooth generation of valid PDE solutions across the entire latent space.

S2.4 Parameter-supervised reference (PNO) performance

The parameter-supervised neural operator (PNO) receives the true governing factor of each system instance as an explicit input and shares the decoder architecture of NODnet within each case. PNO is not a peer baseline: it relies on factor labels that the NOD setting does not provide. Its role is to set a reference for the prediction error attainable when the per-instance factor is known, against which the gap incurred by factor inference can be read off. Calibration details (e.g., the $\sqrt{\nu}$ transform used for the Burgers' case) are reported in Section S2.1.

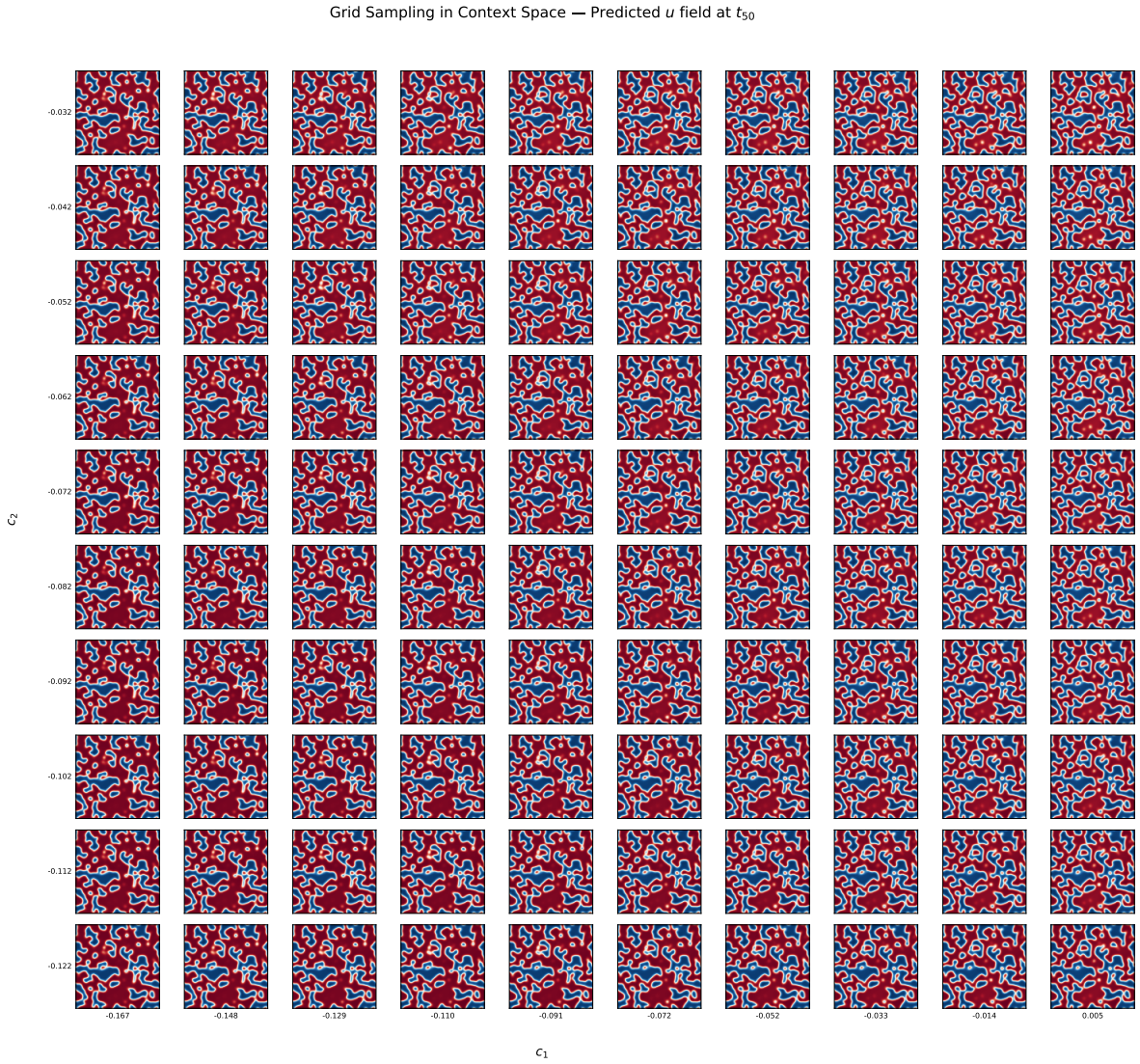

Figure S11: Grid sampling in latent space. Each cell shows the predicted FN-RD field at t=50 for a uniformly sampled latent $\mathbf{c}$, demonstrating that the decoder generates physically plausible dynamics across the entire learned manifold.

Table S8: **Parameter-supervised reference (PNO) performance.** L_2 relative error ($\times 10^{-2}$, mean±std) at 1-, 5-, and 50-step rollout horizons across the same splits used in main-text Table 1. The PNO row receives the true governing factor as input; values are single-seed.

Model	Subset	1-step	5-step	50-step
Burgers' equation				
PNO-DeepONet$^{\uparrow}$ (5.0M)	ID	3.89±1.74	6.04±2.58	13.43±6.39
	OOD	4.03±1.76	5.79±2.43	12.85±6.20
	OOD-inviscid	4.19±1.19	7.68±3.12	16.90±5.70
Cylinder flow wake				
PNO$^{\uparrow}$ (4.5M)	ID	3.65±1.51	12.27±5.53	36.52±12.26
	OOD	2.02±1.51	5.01±3.71	28.11±13.58
FitzHugh–Nagumo reaction–diffusion				
PNO-Hier$^{\uparrow}$ (2.5M)	ID	0.77±0.06	1.16±0.09	1.83±0.28
	OOD-Intra	0.74±0.04	1.10±0.06	1.68±0.16
	OOD-Extra	0.76±0.08	1.23±0.14	2.33±0.55

Table S9: Ablation on Trajectory-Decoupled Sampling. Model convergence is evaluated on the test dataset at 1-step prediction. All pairs share the same optimizer and learning rate schedule. CFW uses unseen systems as the test set; the ablation retains the SupCL regularization.

	w/ TDS			**w/o TDS**		
Case	Epochs	MSE	L2RE%	Epochs	MSE	L2RE%
1D Burgers' Equation	1.5e4	2.85e-4	0.56	3e4	4.71e-4	0.92
FN-RD	67	3.02e-5	0.92	208	6.02e-5	1.96
CFW	422	0.101	1.60	976	0.490	5.61

S3 Extended Discussion

This section gathers extended discussion in four groups: empirical validation of design choices (§ S3.1 TDS ablation, § S3.2 efficiency vs. iMODE); theoretical positioning (§ S3.3 alternative perspectives on the diffeomorphism, § S3.4 connections to identifiability theory); comparisons with alternative paradigms (§ S3.5 foundation-model surrogates, § S3.6 symbolic regression); and cross-case synthesis (§ S3.7 structural slack), closing with a practitioner's checklist (§ S3.8).

S3.1 Ablation on Trajectory-Decoupled Sampling

Trajectory-Decoupled Sampling (TDS) ensures that the conditioning trajectory $\mathbf{x}_{0:t}$ (fed to the encoder) and the prediction trajectory $\mathbf{x}'_{0:s}$ (predicted by the decoder) are randomly sampled from different trajectories of the same system instance. Table S9 compares convergence speed and final test-set accuracy with and without TDS across three cases.

TDS consistently accelerates convergence (2–3× fewer epochs) and improves final test-set accuracy across all cases. Without TDS, the encoder attempts to encode trajectory-specific transient information (e.g., initial conditions, temporal phase) into the latent c, producing noisy optimization signals and prolonging training. TDS prevents this by forcing the encoder to extract only the invariant governing dynamics shared across trajectories of the same SI. The resulting improvement in generalization (lower L2RE on held-out data) confirms that TDS acts as an implicit regularizer against overfitting to trajectory-level features.

S3.2 Efficiency comparison between NODnet and iMODE

We substitute the neural-ODE process in iMODE with a fixed-interval predictor f for like-for-like comparison with NODnet in computational complexity. The iMODE formulation is

$$\text{Prediction} : \hat{\mathbf{x}}_{t+l;\,i} = f_\theta^{(l)}(\mathbf{x}_t, \eta_i^{(j)}) \tag{S8}$$

$$\text{Loss} : \mathcal{L}_i(\theta, \eta_i^{(j)}) = \frac{1}{k}\sum_{h=1}^{k} \text{MSE}(\hat{\mathbf{x}}_{t+h;\,i}, \mathbf{x}_{t+h;\,i}) \tag{S9}$$

$$\text{Initialization} : \eta_i^{(0)} = \eta^*, \quad \eta_i \equiv \eta_i^{(p)} \tag{S10}$$

$$\text{Inner update} : \eta_i^{(j+1)} = \eta_i^{(j)} - \alpha\, \nabla_\eta\, \mathcal{L}_i(\theta, \eta_i^{(j)}) \tag{S11}$$

$$\text{Outer update} : \theta \leftarrow \theta - \frac{\beta}{N}\sum_{i=1}^{N} \nabla_\theta\, \mathcal{L}_i(\theta, \eta_i) \tag{S12}$$

where p is the number of inner update steps and m is the number of recurrent neural-network evaluations. The NODnet formulation is

$$\text{Latent} : c_i = f_\phi(\mathbf{x}_{s:s+m;\,i}) = o\big(\Phi_\phi^{(m+1)}(\mathbf{x}_{s:s+m;\,i})\big) \tag{S13}$$

$$\text{Prediction} : \hat{\mathbf{x}}_{t+l;\,i} = g_\theta^{(l)}(\mathbf{x}_t, c_i) \tag{S14}$$

$$\text{Loss} : \mathcal{L}_i = \frac{1}{k}\sum_{h=1}^{k} \text{MSE}(\hat{\mathbf{x}}_{t+h;\,i}, \mathbf{x}_{t+h;\,i}) \tag{S15}$$

$$\text{Update encoder} : \phi \leftarrow \phi - \frac{\alpha}{N}\sum_{i=1}^{N} \nabla_\phi\, \mathcal{L}_i(\phi, \theta) \tag{S16}$$

$$\text{Update decoder} : \theta \leftarrow \theta - \frac{\alpha}{N}\sum_{i=1}^{N} \nabla_\theta\, \mathcal{L}_i(\phi, \theta) \tag{S17}$$

In practice, neural-network backpropagation takes β times longer than the forward pass (typically $\beta > 1$) because of intermediate-state storage. The computational complexities of iMODE and NODnet are

$$\begin{aligned} \mathcal{O}_{\text{iMODE}} &= N(p+1)(\beta+1)\, k\, \mathcal{O}_\theta, \\ \mathcal{O}_{\text{NODnet}} &= N(\beta+1)\big(m\, \mathcal{O}_\phi + k\, \mathcal{O}_\theta\big), \end{aligned} \tag{S18}$$

giving the ratio

$$\frac{\mathcal{O}_{\text{iMODE}}}{\mathcal{O}_{\text{NODnet}}} = \frac{k(p+1)\, \mathcal{O}_\theta}{m\, \mathcal{O}_\phi + k\, \mathcal{O}_\theta}. \tag{S19}$$

We compare the training of iMODE and NODnet on the bistable system case where $k = m = p = 5$. The training costs for both methods are shown in Table S10. iMODE consumes approximately the same amount of memory as NODnet because only first-order gradient information is stored. However, the repetitive inner loop optimizations end up with 3 times computational complexity than NODnet.

Table S10: Efficiency comparison between NODnet and iMODE, on an RTX Ada 6000 NVIDIA GPU. Note that time* is simply calculated as Epoch × Computation time per epoch for reference.

Metrics	NODnet	iMODE
Computation time per epoch	8.4 s	24.7 s
GRAM per batch	2360 MB	2852 MB
Epoch reaching error 1e-3	31	12
Time* reaching error 1e-3	260.4 s	296.4 s
Epoch reaching error 1e-4	800	500
Time* reaching error 1e-4	6720 s	12350 s

S3.3 Alternative Perspectives for Diffeomorphic Latent Space

In this section, we offer alternative perspectives to understand the diffeomorphic structure of latent space.

GAN perspective The original generative adversarial network (GAN) solves the following min-max optimisation problem,

$$\min_{\mathbf{G}} \max_{\mathbf{D}} V(\mathbf{D}, \mathbf{G}) = \mathbb{E}_{\mathbf{x}}[\log \mathbf{D}(\mathbf{x})] + \mathbb{E}_{\mathbf{z}}[\log(1 - \mathbf{D}(\mathbf{G}(\mathbf{z})))], \tag{S20}$$

where $\mathbf{x} \sim p_{\text{data}}$, $\mathbf{z} \sim p_{\mathbf{z}}$, $\mathbf{D}$ is a discriminator optimised to distinguish real samples $\mathbf{x}$ from synthetic ones $\mathbf{G}(\mathbf{z})$, and $\mathbf{G}$ is a generator that produces synthetic samples from latent input $\mathbf{z}$ and seeks to maximise $\mathbf{D}$'s evaluation.

NODnet admits an analogous reading: the encoder f_ϕ serves as the generator (producing latents from conditioning trajectories) and the decoder g_θ serves as the discriminator (using the latent to predict the next state and being scored by prediction error). For interpretation, in each training epoch we manually separate the latents into two sets, $c = \{c_x, c_z\}$, where c_x acts as the real factors up to a diffeomorphism in the current epoch and c_z are the synthetic ones. The NODnet discriminator (the decoder g_θ) is optimised by

$$\max_{g_\theta} -\mathbf{MSE}\big(g_\theta(\mathbf{x}_t, c_x), \mathbf{x}_{t+1}\big) \equiv \min_{g_\theta} \mathbf{MSE}\big(g_\theta(\mathbf{x}_t, c_x), \mathbf{x}_{t+1}\big). \tag{S21}$$

The NODnet generator (the encoder f_ϕ) is optimised by

$$\min_{f_\phi} \mathbf{MSE}\big(g_\theta(\mathbf{x}_t, c_z), \mathbf{x}_{t+1}\big) = \mathbf{MSE}\big(g_\theta\big(\mathbf{x}_t, f_\phi(\mathbf{x}_{s:s+m;\,z})\big), \mathbf{x}_{t+1}\big). \tag{S22}$$

f_ϕ and g_θ are simultaneously optimised by

$$\begin{aligned} &\min_{f_\phi} \max_{g_\theta} \left[-\mathbf{MSE}\big(g_\theta(\mathbf{x}_t, c_x), \mathbf{x}_{t+1}\big) + \mathbf{MSE}\big(g_\theta\big(\mathbf{x}_t, f_\phi(\mathbf{x}_{s:s+m;\,z})\big), \mathbf{x}_{t+1}\big)\right] \\ &\equiv \min_{f_\phi,\, g_\theta} \mathbf{MSE}\big(g_\theta\big(\mathbf{x}_t, f_\phi(\mathbf{x}_{s:s+m})\big), \mathbf{x}_{t+1}\big). \end{aligned} \tag{S23}$$

In words, g_θ is optimised for prediction accuracy given the reference latents c_x, while f_ϕ is optimised so that the inferred latents $c_z = f_\phi(\mathbf{x}_{s:s+m;\,z})$ produce predictions consistent with g_θ. Equation (S23) gives a GAN-inspired interpretation of the coupled encoder–decoder training objective, rather than a formal equivalence to adversarial training.

Partial-ordering perspective To further explain the ordering in the latent space, we formulate:

$$\mathcal{S}(\mathbf{x}_t \mid p_i) = g_\theta(\mathbf{x}_t \mid c_i) = \mathbf{x}_{t+1} \tag{S24}$$

where $\mathcal{S}$ is the true physics solution operator. In classical physics, $\mathcal{S}$ is generally continuous and differentiable. Since neural operator g_θ is usually also continuous and differentiable, it follows that:

$$c_i = g_\theta^{-1}\mathcal{S}(p_i), \tag{S25}$$

where the operator $g_\theta^{-1}\mathcal{S}(\cdot)$ is also continuous and differentiable. For any triplet of factors $(p_i, p_j, p_k) \in \mathcal{P}$, continuity of the mapping suggests that if p_j lies between p_i and p_k, the discovered latent c_j should take a corresponding intermediate position between c_i and c_k. This behavior is related to Brouwer's Invariance of Domain [8]: under the additional assumptions of continuity, injectivity, and matched manifold dimension, the mapping preserves neighborhood structure. Trajectory-decoupled sampling encourages the encoder f_ϕ to assign a consistent latent to each system instance; when this empirical map is approximately injective and smooth, the observed latent space $\mathcal{C}$ behaves as a smooth image of the factor manifold $\mathcal{P}$.

S3.4 Connections to identifiability theory in latent-variable models

NOD's recovery of per-instance factors from heterogeneous trajectories sits adjacent to several lines of identifiability theory in latent-variable models. The comparison clarifies what NOD assumes and does not assume, and why TDS is the natural training objective in this regime.

Nonlinear ICA with auxiliary variables. Khemakhem et al. [9] establish that nonlinear ICA is identifiable up to component-wise transformations when the latents z are conditionally distributed given an observed auxiliary variable u and $p(z \mid u)$ lies in a sufficiently rich exponential family; the auxiliary u breaks the rotational symmetry that otherwise renders nonlinear ICA non-identifiable. NOD's instance grouping (knowing which trajectories belong to the same system instance) supplies exactly such an auxiliary: each grouping label u_i indexes a distinct conditional $p(z \mid u_i)$, and $p(z \mid u_i) \neq p(z \mid u_j)$ whenever the underlying factors differ ($p_i \neq p_j$). The formal subtlety is that this conditional is degenerate (a point mass at the deterministic factor p_i) rather than a continuous parametric family. NOD therefore lies in the limit where Khemakhem's auxiliary-variable identifiability degenerates to deterministic-factor identifiability, with the symmetry-breaking role preserved and the identification enforced through TDS rather than through likelihood maximisation over a parametric conditional.

Disentanglement under weak supervision. Locatello et al. [10] prove that fully unsupervised disentanglement is non-identifiable: for any generative model, an alternative model with entangled latents is observationally equivalent. NOD escapes this negative result because instance grouping provides weak supervision. The closer reference is Locatello et al. [11]: under pair sampling where two observations share K of d generative factors, the latent is identifiable up to component-wise reparametrization. NOD's TDS instantiates this with $K = d$ (the conditioning and supervision trajectories share all factors, only initial conditions differ), the maximally-shared case, giving identifiability up to permutation and sign of the latent coordinates without further structural assumptions.

Temporal structure as auxiliary. Hyvärinen et al. [12] use the time index as an auxiliary variable for nonlinear ICA: segments at different time points have different conditional distributions over the latents, and the resulting contrast identifies them. Klindt et al. [13] extend this to nonlinear disentanglement of natural data via temporal sparse coding. NOD's TDS uses the same kind of structural index to break symmetry but swaps the contrast axis from time-segment to instance: the auxiliary indexes which system instance generated the trajectory, not which time window within one trajectory. The resulting latent identifies the per-instance factor rather than the per-segment temporal mode, and the contrast is across the instance ensemble rather than within a single trajectory.

What NODnet adds beyond these. Existing identifiability results constrain the latent representation but assume the latent dimension d is given. NODnet's minimal conditioning principle adds dimension identification: d^* is gauged from data as the smallest value at which prediction loss saturates. The recovered d^* matches the number of physically independent governing factors in every case (Table S1), giving an identifiability claim that is both factor-wise (latent coordinates correspond to factors) and dimension-wise (the number of recovered factors is correct). The decoder side is also distinct: g_θ is a full neural operator advancing states under continuous-time dynamics, not a static generative model, so identification is enforced through forward dynamical consistency rather than reconstruction of static observations. The identifiability claim NODnet supports is therefore: under instance grouping (auxiliary in the Khemakhem sense, weak supervision in the Locatello [11] sense), TDS pairing with $K = d$ shared factors, and dimension gauging at d^*, the recovered latent is unique up to permutation, sign, and component-wise reparametrization, with the dimension of the recovered representation matching the dimension of the governing factor space.

From extrapolation to extrapolation range. Identifiability ensures the latent-factor map is unique on the training support; it does not specify how far the map remains valid beyond the training hull. NODnet's case studies exhibit useful extrapolation in every instance (Burgers' crosses the inviscid singularity ν=0, CFW generalizes outside the trained Reynolds-number range, FN-RD generalizes to factor pairs outside the training (k, β) box). Characterizing the *radius* of stability of the diffeomorphism, the maximum factor distance from the training hull at which one-shot or zero-shot inference still recovers the correct dynamics, is one step beyond identifiability and is not formally addressed here. Empirically, generalization fails when the test factor leaves the smooth latent-factor manifold (e.g. at bifurcation thresholds or topology changes in the dynamics); a formal theory linking the recovered Jacobian condition number and the loss landscape's local curvature to the resulting stability radius would close this gap.

S3.5 Foundation-model surrogates: scope of comparison

Foundation-model-style surrogates [14, 15, 16, 17] predict the next state from a long window of past states, with the conditioning produced inside a single forward pass through a large transformer-class decoder. Two independent design axes contribute to their reported accuracy: (i) the conditioning structure, an unstructured past-state window with no per-instance latent, and (ii) the decoder capacity, a foundation-scale backbone trained across many system families.

A direct head-to-head between NODnet and a full foundation surrogate (MPP, DPOT, ViCon, Multi-Physics Pretraining) would conflate these two axes: a larger decoder typically wins regardless of conditioning structure, so a side-by-side test would not isolate the contribution of the minimal conditioning principle. We therefore use ANO$^{\downarrow}$ as the controlled foundation-style baseline at matched per-case decoder capacity, with the conditioning channel replaced by the same window-of-past-states structure that foundation surrogates use (given history, predict next). The ablation between NODnet and ANO$^{\downarrow}$ thereby measures the conditioning-structure effect with decoder capacity held constant; the comparison against PNO$^{\uparrow}$ (true factors as input) and SNO (d=0, no encoder) on the same axis completes the conditioning sweep from oracle to absent.

NODnet's encoder–decoder split is itself architecture-agnostic: the foundation-scale decoders cited above could be substituted for the per-case backbone in g_θ, with the minimal conditioning channel preserved between f_ϕ and g_θ (a foundation-scale g_θ with a d^*-dim latent on the side). Such a hybrid is a natural extension complementary to foundation-scale training, not a competitor. The principle prescribes how the conditioning channel is structured, not how the decoder is scaled; holding decoder capacity fixed in the present comparison ensures that the gap between NODnet and ANO$^{\downarrow}$ is the conditioning-structure effect rather than a backbone-scale effect.

S3.6 Comparison with symbolic regression

Symbolic regression (SR) and NODnet address overlapping problems but sit at different points on the expressivity–interpretability frontier; the comparison clarifies when each is the appropriate tool.

What each method returns. SR returns a closed-form symbolic equation, typically a sparse combination of hand-designed primitives such as polynomials, partial derivatives, and trigonometric functions [18, 19, 20]. NODnet returns a learned operator pair (f_ϕ, g_θ) together with a d^*-dimensional latent space empirically diffeomorphic to the per-instance factor space; the form of the underlying law is not recovered.

Scope of dynamics. SR's expressivity is bounded by its primitive library. Systems whose governing equations involve primitives outside the library, or whose effective dynamics are not sparsely representable, fall outside SR's reach. The cylinder-flow wake on an unstructured 32k-node mesh is one such regime: the spatial discretization, the boundary geometry, and the multi-scale wake structure jointly resist sparse symbolic representation. NODnet's neural-operator decoder has no analogous bound and recovers a usable surrogate in this regime (main-text Fig. 3, Table 1). The same scope distinction applies to experimentally observed systems where the underlying physics is itself partially unknown: NODnet requires only trajectories.

Family recovery. Modern SR variants extend the original sparse-regression framework to families with varying coefficients, e.g. parametric SINDy [21]. When the parametric dependence is itself sparse and library-expressible, these variants recover the family. NODnet recovers operator and per-instance factor jointly in one training run, without requiring the parametric form to be analytically clean; the gauged d^* identifies the correct number of factors directly from data.

Inference cost. A symbolic equation must be numerically integrated to make predictions, with cost set by the discretization required for the recovered equation (CFL-bounded Δt, fine spatial grids, stiffness-dependent solver choice). For 1D PDEs this is modest (milliseconds to seconds per trajectory); for 2D PDEs the cost reaches seconds to minutes on CPU; 3D pushes higher. NODnet inference is a forward pass of g_θ rolled forward from an initial state, costing milliseconds on GPU for the cases here. The practical speedup ranges from $\sim 10^2\times$ for 1D systems to $\sim 10^4$–$10^5\times$ for 2D problems with fine resolution.

What SR offers that NODnet does not. A recovered closed-form law generalises beyond the training distribution to arbitrary initial and boundary conditions, composes naturally with other equations in multi-physics settings, and provides physical insight from the equation form itself. NODnet generalises only as far as its training distribution and the diffeomorphism extends, and its interpretability is at the factor-structure level rather than equation-structure level.

When to pick which. SR is appropriate when the dynamics are within library expressivity, a closed-form law is the deliverable, and the inference budget tolerates numerical integration. NODnet is appropriate when the dynamics are too complex for library expressivity (e.g. the cylinder wake), only trajectory data is available (experimental settings), or inference must be fast. The two are complementary rather than substitutes.

S3.7 Structural slack: cross-case synthesis

The minimal conditioning principle predicts that test error worsens above d^* because the encoder's spare dimensions absorb initial-condition-specific signal that the decoder then propagates as drift through the rollout. The three case studies provide complementary evidence of this structural-slack mechanism, with the strength of the signal modulated by data availability and augmentation.

Burgers' (S2.1). Seed-mean Train MSE rises from 1.01 at d=1 to 1.42 at d=2 (Table S3), and the 50-step rollout error rises from $\sim 17.6\%$ at d=1 to $\sim 21.7\%$ at d=2 on ID, with d=2 the worst plateau dimension. The encoder's extra dimensions absorb shock-front micro-variation that does not depend on the governing factor ν, and the autoregressive loss amplifies this absorption. The d=3, 4 rows return closer to the d=1 envelope because at higher dimensions the slack scatters across multiple axes rather than concentrating in one, an effect also seen on CFW (next paragraph).

Cylinder flow wake (S2.2). The clearest geometric demonstration. PCA on the trained d=2 encoder's per-system mean latents returns variance ratios $(0.739, 0.261)$: the second axis is populated, not empty (Fig. S8), and what populates it is within-system conditioning-window variability rather than ν. The non-monotonic d>1 test error in Table S5 is the rollout consequence: d=2 is worst because the slack is concentrated in one axis; d=3, 4 partially recover by spreading the slack across multiple axes (PC3 contribution 0.083, PC2 dropping from 0.261 to 0.231).

FN-RD (S2.3). The 50-step interpolation column shows a soft surge at d=4: seed-means $4.46 \rightarrow 3.05 \rightarrow 2.50 \rightarrow 3.15$ as d grows from 1 to 4, with Train MSE worsening from 4.74 at d=3 to 5.06 at d=4. The surge is delayed by one step relative to Burgers' and smaller in magnitude because the 10× O(2) data augmentation acts as an additional regularizer, suppressing the slack effect within $d \leq 3$ and partially absorbing it at d=4.

The d=0 anchor. At long horizons in the cylinder wake the d=0 baseline (no encoder, no slack to fill) is competitive with the entire d>1 curve (Table S5, H=50 on OOD: d=0=34.5%, d=2=38.3%, d=3=34.5%, d=4=33.0%, all within seed envelope of one another and of d^*=1 at 31.0%). "No encoder" is therefore not strictly dominated by "encoder with slack", direct empirical evidence that slack rather than absent factor information is the dominant source of $d > d^*$ rollout error: once the rollout has time to amplify the IC absorption, the bottleneck-free baseline catches up because it lacks the slack channel that drives drift.

S3.8 Applying NODnet to a new dataset

This subsection condenses the workflow for applying NODnet to a new heterogeneous-trajectory dataset, drawing on the seven reported cases.

Data. Organise as $\mathcal{D} = \{\mathcal{T}_{ij}\}$ with i indexing the system instance and j indexing trajectories within an instance. NODnet requires (i) at least two trajectories per instance under distinct initial conditions (for trajectory-decoupled sampling), and (ii) trajectories long enough that a conditioning window

$\mathbf{x}_{0:t}$ captures factor-bearing dynamics. The instance grouping is the only supervision NODnet uses; per-instance factor labels are not required at any stage.

Backbone selection. Choose f_ϕ and g_θ to match the observation structure: CNN/U-CNN for regular grids, Point Transformer for irregular meshes, recurrent (LSTM/GRU) for time series, FNO for resolution-agnostic recovery, Neural ODE for differential decoders. Table S1 maps observation structure to backbone choice across the seven reported cases. The principle is architecture-agnostic; matching the backbone to the data is an engineering choice.

Dimension gauging. Sweep $d \in \{0, 1, \ldots, d_{\max}\}$ with $d_{\max}$ a few above the suspected factor count, training each d over ≥ 3 independent seeds under identical settings (same encoder for $d \geq 1$, same decoder, same loss, same epoch budget); the d=0 run drops the encoder, recovering the SNO baseline. Apply the two-stage rule (Methods). *Stage 1.* On a long-horizon training-side gauge — Train MSE for autoregressive cases, Train H=50 L_2 relative error for cases where short-horizon Train MSE is overfit by the encoder-free baseline (CFW) — identify the smallest d at which the next-step relative change falls within the across-seed envelope, and admit it together with the next dimension as the candidate set. *Stage 2.* Inspect PCA variance ratios on per-instance latent means at the candidate dimensions: select the smallest d^* whose d+1 run either populates a spurious slack axis or collapses to a d^*-dimensional submanifold. Sections S2.1, S2.2, S2.3 give worked examples. Three seeds per d sufficed across all reported cases, and the runs are independent so they parallelise across GPUs.

Trajectory-decoupled sampling. At each training step, sample the conditioning trajectory $\mathbf{x}_{0:t}$ from one initial condition of instance i and the supervision trajectory $\mathbf{x}'_{0:s}$ from a different initial condition of the same instance. TDS integrates as a sampler choice in the data loader and adds no extra forward passes; the ablation in S3.1 shows 2–3× faster convergence and lower test error than naive single-trajectory sampling.

Diagnostics after training at d^*. Three checks confirm the principle has taken effect: (i) per-instance latent means $\bar{c}_i$ versus a factor proxy should form a smooth, monotonic curve (the empirical diffeomorphism signature, Figs. S7, S10); (ii) PCA on the per-instance latents should return rank matching d^* to numerical noise, with a non-empty principal component beyond d^* signalling structural slack; (iii) one-shot inference on held-out instances should yield a latent consistent with the manifold geometry.

Failure modes. (i) *Encoder collapse*: the latent maps all instances to the same value. Causes include short conditioning windows, supervision-geometry mismatch (Burgers' NODnet-AR is the canonical case, Section S2.1), or insufficient instance diversity; diagnose by plotting per-instance latents; remedy by lengthening the conditioning window, adopting a full-field decoder, or adding instance diversity. (ii) *Latent degeneration with sparse instances*: with very few instances ($\lesssim 16$), prediction loss alone may not separate the latents enough; add a manifold regularizer (the supervised-contrastive loss in S2.2 CFW handles 16 instances). (iii) *Discrete or topologically nontrivial factor spaces*: NODnet's diffeomorphism premise assumes continuous factor variation; bifurcating regimes and topology-changing dynamics lie outside the present scope.

References

[1] R.T. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations. Advances in neural information processing systems **31** (2018)

[2] J.M. Lee, J.M. Lee, *Smooth manifolds* (Springer, 2003)

[3] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 4700–4708

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

[5] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F.A. Hamprecht, Y. Bengio, A. Courville, in *Proceedings of the 36th International Conference on Machine Learning (ICML)* (2019), pp. 5301–5310

[6] M. Tancik, P.P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J.T. Barron, R. Ng, in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33 (2020), pp. 7537–7547

[7] 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)

[8] E.H. Spanier, *Algebraic topology* (Springer Science & Business Media, 2012)

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

[10] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, O. Bachem, in *International Conference on Machine Learning* (PMLR, 2019), pp. 4114–4124

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

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

[13] D.A. Klindt, L. Schott, Y. Sharma, I. Ustyuzhaninov, W. Brendel, M. Bethge, D. Paiton, in *International Conference on Learning Representations* (2021)

[14] 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)

[15] 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)

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

[17] 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)

[18] S.M. Udrescu, M. Tegmark, AI Feynman: A physics-inspired method for symbolic regression. Science advances **6**(16), eaay2631 (2020)

[19] 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)

[20] M. Cranmer, Interpretable machine learning for science with pysr and symbolicregression. jl. arXiv preprint arXiv:2305.01582 (2023)

[21] S. Rudy, A. Alla, S.L. Brunton, J.N. Kutz, Data-driven identification of parametric partial differential equations. SIAM Journal on Applied Dynamical Systems **18**(2), 643–660 (2019)