NEURALPDE: LEARNING LATENT PDE DYNAMICS FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series analysis extracts information from historical observations and predicts their future behavior. Many existing models make predictions only by learning historical data patterns, and the underlying evolution laws, which are often described by partial differential equations (PDEs) or other dynamical rules in real world physical systems, may be overlooked. We therefore propose NeuralPDE, which encodes historical observations to latent functions of an auxiliary variable and learns their evolution through an explicit advection–diffusion–reaction PDE on an auxiliary domain, and a learned decoder then maps the evolving functions to future predictions. NeuralPDE learns PDE-based evolution laws of complex systems in latent space to improve the stability and accuracy of multi-step forecasting. The roles of the transport and diffusion terms in NeuralPDE are theoretically verified through establishing energy dissipation for the linear component. We apply NeuralPDE on real-world time series and NeuralPDE demonstrates competitiveness in time-series multi-step forecasting tasks in comparison with existing models.

1 INTRODUCTION

Many time-series models, including recurrent models such as GRUs (Cho et al., 2014) and LSTMs (Hochreiter & Schmidhuber, 1997), learn to forecast future observations from patterns in historical data without explicitly modeling dynamics evolution.

Latent models based on neural ODEs offer a way to learn continuous-time dynamics: they encode an observation history as a point in a finite-dimensional latent space and evolve it under a learned vector field (Chen et al., 2018; Rubanova et al., 2019). However, under the usual uniqueness conditions, two trajectories of the same ODE that reach the same complete latent state at the same time share the same continuation, and their predictions coincide under identical decoder conditions (Dupont et al., 2019). Any deterministic evolution therefore relies on its state retaining the information needed to distinguish histories that require different forecasts. Compressing this information into a compact vector may make this distinction difficult. We therefore seek a richer representation of the evolving state to help preserve these distinctions, together with an encoder that can infer predictive information from historical observations.

Motivated by this, we propose NEURALPDE, which encodes observation histories into a latent information function space. We aim to preserve historical trends and reduce the risk of conflating histories with different predictive information. We introduce an auxiliary variable to organize this information within each function, providing a domain for local coupling, transport, and smoothing. The resulting latent functions evolve according to

$$\partial_\tau z = \mathcal{G}(z)$$

where $\mathcal{G}$ is the evolution operator. We hope that this operator can provide local coupling, continuous information propagation, scale-dependent smoothing, and stable dynamics.

This leads to the heat equation, whose diffusion operator provides local coupling and scale-dependent smoothing. We add a learnable reaction term to capture nonlinear interactions in the latent evolution, together with an advection term for directional transport. The resulting evolution takes the form

$$\partial_\tau z(\tau, s) = -A\partial_s z(\tau, s) + B\partial_{ss} z(\tau, s) + R_\theta(z(\tau, s)). \tag{1}$$

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

We refer to this as advection–diffusion–reaction (ADR) dynamics, where τ denotes latent evolution time and s is the auxiliary coordinate. A learned decoder maps the evolving latent functions to future observations. When the latent function is constant in s, both spatial derivatives vanish and the evolution reduces to $\partial_\tau z = R_\theta(z)$. Thus, the model includes an autonomous neural ODE over the latent channels as a special case.

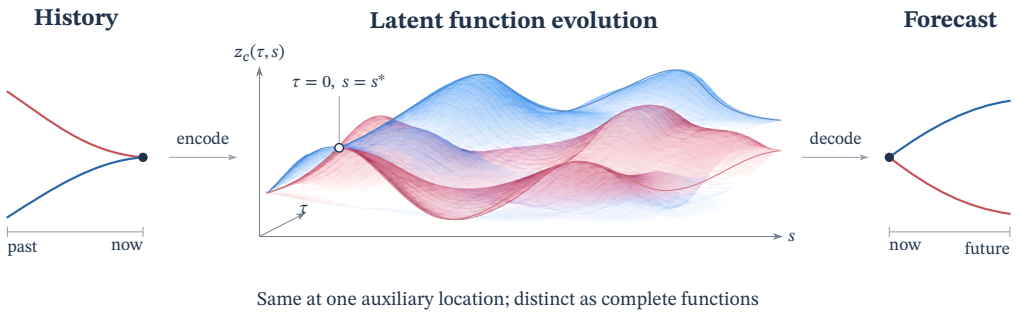


Figure 1: Schematic motivation for a history-dependent latent function. Two histories share a last observed value but require different forecasts. Their encoded functions can agree at one auxiliary location (open circle, $\tau = 0$) while remaining distinct as complete profiles. The colored surfaces illustrate one latent channel evolving toward the forecast decoder.

To make the PDE evolution numerically tractable, we apply a Fourier transform and retain a finite set of modes. The retained coefficients satisfy a coupled ODE system, which we integrate with differentiable ODE solvers. This formulation also permits Neural ODE adjoint backpropagation, with $O(1)$ memory in the number of internal solver steps, at the cost of solving an additional adjoint system during the backward pass (Chen et al., 2018). Meanwhile, like Augmented Neural ODEs (ANODEs) (Dupont et al., 2019), NEURALPDE can also overcome the planar separation constraint in the disk–annulus task by lifting inputs into a latent function space. Section 5.4 explains this behavior using Figure 3.

In the part of theory, We analyze the initial value sensitivity of this operator structure and characterize the modal action and dissipation of the linear ADR operator, then establish a conditional perturbation bound for the sampled nonlinear reaction and the forecast decoder. The assumptions distinguish linear smoothing from contraction of the full nonlinear evolution.

We evaluate sparse-sensor forecasting on public Burgers solutions and measured cylinder-wake, oscillator, and sea surface temperature data, comparing NEURALPDE with a matched latent ODE and direct predictors under documented capacity and training settings. Encoder and ADR-component ablations examine where the structure helps. A disk–annulus experiment visualizes the representation capacity of the latent function. Section 5 presents the forecasting and representation results; full protocols and supplementary results are collected in Appendices A–G.

2 RELATED WORK

Continuous latent dynamics and state representation. Neural ODEs learn continuous vector fields, and Latent ODEs infer an initial latent vector from observation histories and then evolve it (Chen et al., 2018; Rubanova et al., 2019). Augmented Neural ODEs analyze restrictions imposed by topology-preserving flows and expand the state space to address them (Dupont et al., 2019). We encode histories as latent functions and learn their ADR evolution on an auxiliary domain. Compared with evolving encoded observations in a latent vector space, this representation provides additional coordinates for distinguishing predictive states. Two latent functions can agree at some locations while differing elsewhere, reflecting differences in the historical trends they encode. Their full function profiles therefore remain distinct. Like ANODEs, this lifting permits separation without requiring trajectories to intersect in the complete state space. Our disk–annulus pilot illustrates this capacity on the evaluated grid, where both NEURALPDE and ANODEs overcome the separation difficulty encountered by the planar Neural ODE.

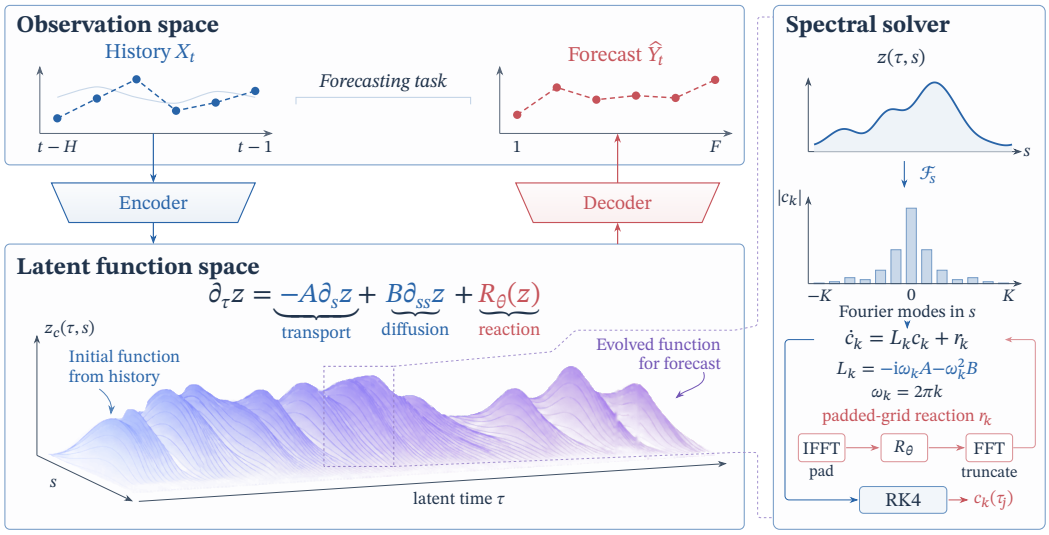


Figure 2: The pipeline of NEURALPDE

Latent PDEs and PDE-based forecasting. Iakovlev et al. learn space-time continuous latent PDEs from noisy partial observations on irregular spatiotemporal grids. Their latent field is indexed by locations in the observed spatial domain; interpolation and spatial neighborhoods define its dynamics, and variational inference estimates the latent states (Iakovlev et al., 2023).

PDETime treats multivariate time series as samples of a continuous system on a fixed but unknown spatial domain associated with channel attributes. Its encoder uses history and temporal features to predict a latent integrand, which is integrated and decoded into forecasts (Qi et al., 2024).

A key distinction from both approaches is that NEURALPDE explicitly introduces an auxiliary coordinate s to organize a predictive latent function. Both approaches associate their spatial domain with locations or attributes of the observed system. Our auxiliary domain is chosen independently of the observed spatial domain, and s has no prescribed correspondence to sensor locations or channel attributes. Ordered sensor histories initialize $z(0, s)$, which then evolves according to our ADR equation.

Fourier representations and differentiable PDE solvers. Fourier Neural Operators (FNOs) learn mappings between functions using integral kernels parameterized in Fourier space (Li et al., 2021). We use Fourier coordinates to make PDE integration tractable: retaining a finite set of modes yields a coupled ODE system for the coefficients, which we can solve with differentiable integrators. This formulation also permits Neural ODE adjoint backpropagation with $O(1)$ memory in the number of internal solver steps, when the state dimension and requested output times are held fixed (Chen et al., 2018). For convenience, our reported experiments use direct differentiation. Our use of spectral coordinates to represent and transform latent functions is also partly inspired by FNOs.

3 METHODOLOGY

NEURALPDE learns a predictive future from some sensors history. A history-to-function encoder supplies the initial condition, an ADR generator evolves the function, and a global decoder maps it to future observations. Figure 2 outlines this forecasting pipeline.

3.1 PROBLEM FORMULATION

The task is to predict F future at an anchor sensor from H historical observations at M fixed sensors. Each sensor supplies C variables, and the anchor occupies slot zero. Writing $x_r^{(0:M)}$ for the

concatenated observations from slots $0, \ldots, M-1$, the input and output are

$$X_t = [x_{t-H}^{(0:M)}, \ldots, x_{t-1}^{(0:M)}] \in \mathbb{R}^{H \times MC}, \qquad \widehat{Y}_t = [\widehat{y}_1, \ldots, \widehat{y}_F]^\top \in \mathbb{R}^{F \times C}. \tag{2}$$

The ordered feature slots identify the sensors. Physical coordinates, unobserved fields, and graph connectivity are not model inputs.

Encoder. A encoder maps X_t to $u_0 = E_\eta(X_t) \in \mathbb{R}^q$, the real coordinates of a C_z-channel function z_0 on $\mathbb{T} = \mathbb{R}/\mathbb{Z}$. Retaining K positive Fourier modes gives $q = C_z(2K+1)$. The auxiliary coordinate s has no assigned physical location or history age. Appendix A.4 specifies the encoder and the map from its output to the initial function.

3.2 Latent dynamics

We evolve the initialized function with Equation 1, subject to $z(0, s) = z_0(s)$ and periodic boundary conditions. The three terms act on the same representation in complementary ways. Their parameters are shared over the auxiliary domain and forecast time, so the generator is autonomous.

Advection. $-A\partial_s z$ propagates patterns along the domain and couples their channels. We use

$$A = \tfrac{1}{2}(A_{\mathrm{raw}} + A_{\mathrm{raw}}^{\mathsf{T}}). \tag{3}$$

For a scalar channel, this term translates the function. For symmetric channel coupling, its eigenvectors define channel combinations with their own transport speeds. Periodicity and symmetry ensure that transport preserves the function's total L^2 energy in the linear system. Transport can therefore move features without directly amplifying or damping that energy.

Diffusion. $B\partial_{ss} z$ suppresses rapid variation along the auxiliary coordinate. Its parameterization is

$$B = LL^{\mathsf{T}} + \epsilon I, \qquad \epsilon > 0. \tag{4}$$

For continuous Fourier amplitude c_k at mode k, diffusion contributes $-(2\pi k)^2 B c_k$. The quadratic frequency dependence imposes stronger linear damping at finer auxiliary scales, while leaving the constant mode unchanged. These scales belong to the latent domain; they do not directly identify temporal noise or physical wavelengths in the observations.

Reaction. The pointwise network $R_\theta : \mathbb{R}^{C_z} \to \mathbb{R}^{C_z}$ mixes channels at each location and uses the same parameters everywhere. It is a two-hidden-layer SiLU MLP. Applying this nonlinear map to the function can couple Fourier modes, complementing the mode-wise linear derivative terms. The reaction supplies nonlinear growth, decay, and interaction; its weights are not constrained to do not assume that the full nonlinear evolution is contractive.

Together, these operators specify a class of latent evolution laws with shared transport, smoothing, and nonlinear transformations. Section 4 derives the linear energy identity and gives the conditions for nonlinear sensitivity bounds. But the contribution of each component to predictive performance should be assessed empirically through experiments.

3.3 Fourier solver

We solve the latent PDE with a Fourier discretization, evaluating the linear derivatives spectrally and the nonlinear reaction on an oversampled grid. For $\omega_k = 2\pi k$, the retained amplitudes satisfy

$$\frac{\mathrm{d}c_k}{\mathrm{d}\tau} = (-\mathrm{i}\omega_k A - \omega_k^2 B)c_k + \mathcal{Q}_{k,n_p}[R_\theta(z)], \qquad 0 \le k \le K, \tag{5}$$

where $\mathcal{Q}_{k,n_p}$ denotes the k-th discrete Fourier coefficient obtained from the nonlinear reaction evaluated on $n_p = \lceil 1.5(2K+1) \rceil$ auxiliary grid points. The solver packs the real DC coefficients and the real and imaginary parts of the positive coefficients into independent real coordinates. It applies the linear terms directly in the spectral representation. To evaluate the reaction, it pads the spectrum, transforms to the auxiliary grid, applies R_θ pointwise, and transforms back before truncation. Padding reduces aliasing; it does not exactly remove all harmonics produced by a nonpolynomial SiLU reaction.

3.4 DECODER AND TRAINING

A decoder comprising a global spectral projection and an MLP converts each evolved latent function into an anchor prediction. With feature index r, it computes

$$g_r(\tau) = \mathrm{Re}\sum_{c=1}^{C_z}\sum_{k=0}^{K}\overline{W_{rck}}\widehat{z}_{ck}(\tau), \qquad \widehat{y}_j = x_{t-1}^{(0)} + D_\psi(g(\tau_j)). \tag{6}$$

The global spectral projection combines information across the whole function. A two-layer MLP D_ψ maps features to C output variables. The last observed anchor value provides a residual reference.

We train the encoder, ADR , and decoder jointly by minimizing the forecast mean squared error,

$$\mathcal{L} = \frac{1}{FC}\|\widehat{Y}_t - Y_t\|_F^2, \tag{7}$$

averaged over the batch. At inference, the model evolves from the supplied history without future observations. Optimization settings are specified in Appendix A.

4 THEORETICAL ANALYSIS

The analysis explains how the ADR operators act on a predictive function and how perturbations of its initial condition affect the forecast decoder. We present the main theoretical results here; the detailed analysis and complete proofs are provided in Appendix H.

4.1 TRANSPORT, MODAL ACTION, AND DISSIPATION

The linear operators determine the propagation and smoothing built into the latent evolution. On mode k, their combined generator is $M_k = -\mathrm{i}\omega_k A - \omega_k^2 B$. The skew-Hermitian transport part preserves energy, while the negative Hermitian diffusion part dissipates it.

Proposition 1 (Full action on nonzero modes and linear dissipation)**.** *Let $\mathcal{A}z = -A\partial_s z + B\partial_{ss}z$ and $b = \lambda_{\min}(B) > 0$. For every $k > 0$, the complex block $M_k = -\mathrm{i}\omega_k A - \omega_k^2 B$, $\omega_k = 2\pi k$, is invertible. Consequently, $\mathcal{A}$ has real rank $2KC_z$ and its kernel consists exactly of the constant functions. Its continuous-time evolution satisfies*

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\frac{1}{2}\|z\|_{L^2}^2 = -\int_0^1 (\partial_s z)^\top B(\partial_s z)\,\mathrm{d}s \le -b\|\partial_s z\|_{L^2}^2. \tag{8}$$

Writing $\bar{z} = \int_0^1 z(s)\,\mathrm{d}s$, the mean is constant and $\|z(\tau) - \bar{z}\|_{L^2} \le e^{-4\pi^2 b\tau}\|z(0) - \bar{z}\|_{L^2}$ for $\tau \ge 0$.

4.2 FROM LATENT PERTURBATIONS TO FORECAST SENSITIVITY

The global decoder connects the latent bound to the predicted observable. Fix all learned parameters and let $\mathcal{S}_\tau : V_K \to V_K$ denote the exact continuous-time flow of the finite-mode generator with sampled reaction. This differs from the numerical flow $\Phi_{\Delta\tau}$ used in training. Let $\mathcal{D} : V_K \to \mathbb{R}^C$ denote the complete residual decoder in Equation 6, including its spectral projection.

Corollary 2 (Conditional decoder sensitivity)**.** *Under the assumptions of Proposition 5, suppose $\|\mathcal{D}(z) - \mathcal{D}(v)\|_2 \le L_D\|z - v\|_{L^2}$ for all $z, v \in V_K$. Then, for any $z_0, v_0 \in V_K$ and $\tau \ge 0$,*

$$\|\mathcal{D}(\mathcal{S}_\tau z_0) - \mathcal{D}(\mathcal{S}_\tau v_0)\|_2^2 \le L_D^2 e^{2\ell\tau}\|z_0 - v_0\|_{L^2}^2. \tag{9}$$

For histories $X, \widetilde{X}$, let $z_X, z_{\widetilde{X}}$ be their encoded functions and $a_X, a_{\widetilde{X}}$ their last observed anchor values. The continuous-flow forecasts $\widehat{y}_\tau^{\mathrm{ct}}(X) = a_X + \mathcal{D}(\mathcal{S}_\tau z_X)$ satisfy

$$\|\widehat{y}_\tau^{\mathrm{ct}}(X) - \widehat{y}_\tau^{\mathrm{ct}}(\widetilde{X})\|_2 \le \|a_X - a_{\widetilde{X}}\|_2 + L_D e^{\ell\tau}\|z_X - z_{\widetilde{X}}\|_{L^2}. \tag{10}$$

A finite-weight SiLU network admits a finite, generally positive ℓ; training does not enforce $\ell \le 0$. Diffusion does not damp the DC mode, and reaction need not preserve the mean. The proof uses the exact discrete sampling inner product, so it does not assume that $3/2$ padding exactly dealiases the nonpolynomial SiLU reaction.

The finite-weight spectral–SiLU decoder has a finite global L_D; Appendix H.6 gives its normalization in the function norm. Neither L_D nor ℓ is constrained to make the bound small. The result describes how an initial representation perturbation propagates through one fixed learned model.

Representation and numerical qualifications. The continuous energy and perturbation results do not guarantee stability of an RK4 rollout: an overly large explicit step can amplify a dissipative linear mode. The precise linear RK4 criterion and its nonlinear qualifications are given in Appendix H.7.

5 EXPERIMENTS

Model configuration. The primary comparisons use a matched dense latent ODE and a direct GRU. Table 1 specifies the two continuous-time models. The two models share encoder and decoder architectures, paired initialization, the forecast-MSE objective, and training rules, and are trained independently. The dense RHS width of 18 minimizes the raw parameter-count gap to the ADR generator. This comparison does not match computation or effective rank.

Table 1: Model configurations for the sensor experiments. C is the number of variables at the anchor. All MLP hidden layers use SiLU. Complete-model parameter counts below refer to the cylinder tasks ($M = 1, C = 2$).

Component	NEURALPDE	Matched ODE
History encoder	One-layer GRU, 64 units; linear projection $64 \to 136$	
Latent representation	$C_z = 8$, $K = 8$; $q = 136$ real Fourier coordinates	
Evolution law	ADR on a periodic function	Dense autonomous ODE
RHS network	Pointwise $8 \to 64 \to 64 \to 8$	$136 \to 18 \to 18 \to 136$
Global decoder	32 spectral features; MLP $32 \to 64 \to C$	
Integrator	RK4 via `torchdiffeq.odeint`, step size 5×10^{-2}	
RHS parameters	5,384	5,392
Complete-model parameters	34,322	34,330

Evaluation protocol. The main comparisons complete 80 epochs; earlier OISST direct-GRU references with shorter training are identified in separate table blocks. The checkpoint minimizes validation error over trained epochs 1–80. Tables give the mean and sample standard deviation over seeds 7, 17, and 27, using $(\text{mean} \pm \text{SD}) \times 10^n$ notation. These describe training variability, not uncertainty across physical recordings. Validation also informs the learning-rate scheduler, and the cylinder and oscillator settings have been inspected during model development. The disk–annulus representation experiment uses the separate protocol in Section 5.4. Relative error reductions use the named baseline's unrounded seed-mean sMSE as the denominator. Appendix A gives the protocol; Table 2 includes validation persistence references.

5.1 FORECASTING FROM SPARSE OBSERVATIONS OF BURGERS DYNAMICS

We first evaluate on public numerical solutions from PDEBench (Takamoto et al., 2022). The task predicts one sensor from $H = 20$ historical samples at $M \in \{1, 2, 4, 8\}$ sensors, with a forecast horizon of $F = 20$. We split the 10,000 trajectories into 8,000 training, 1,000 validation, and 1,000 test trajectories before extracting windows. All models receive the same observations, without physical coordinates or equation parameters. The comparison includes the matched ODE, direct GRU, TSMixer, and the small official-architecture TimeXer configuration. Appendix F gives the data, capacities, and training settings.

NEURALPDE has the lowest mean test sMSE at each sensor count (Figure 12). Relative to the matched ODE, its mean error is lower by 14.35%, 7.35%, 10.63%, and 13.73% for $M = 1, 2, 4, 8$, respectively. The corresponding reductions relative to the direct GRU are 10.55%, 7.29%, 27.66%, and 2.99%. The last difference is small and favors NEURALPDE in only one of three paired seeds.

5.2 CYLINDER-WAKE FORECASTING

We predict measured velocities from $H = 40$ historical frames. Five probes are evaluated separately at $F = 20$; probe 0 is also evaluated at $F = 40$ and $F = 80$, with each horizon trained separately.

At $F = 40$ and $F = 80$, NEURALPDE reduces mean sMSE relative to the matched ODE by 15.25% and 52.32%, respectively, with lower error in all three paired seeds at both horizons. Relative to the direct GRU, its mean error is 1.96% lower at $F = 40$ and 1.01% higher at $F = 80$. The

Table 2: Cylinder probe-0 sMSE (↓), $H = 40$. Values are mean ± sample SD over seeds 7/17/27; neural models train for 80 epochs. Bold marks the lowest neural mean per column.

Model	$F = 20$	$F = 40$	$F = 80$
NEURALPDE	$(\mathbf{2.777}\pm0.027)\times10^{-1}$	$(\mathbf{3.298}\pm0.035)\times10^{-1}$	$(4.2423\pm0.0756)\times10^{-1}$
ODE (GRU)	$(2.784\pm0.006)\times10^{-1}$	$(3.891\pm0.330)\times10^{-1}$	$(8.8984\pm0.0689)\times10^{-1}$
Direct GRU	$(2.809\pm0.037)\times10^{-1}$	$(3.364\pm0.048)\times10^{-1}$	$(\mathbf{4.2000}\pm0.0327)\times10^{-1}$
TimeXer	$(3.429\pm0.006)\times10^{-1}$	$(4.110\pm0.037)\times10^{-1}$	$(5.3258\pm0.0180)\times10^{-1}$
TSMixer	$(3.744\pm0.039)\times10^{-1}$	$(4.203\pm0.017)\times10^{-1}$	$(4.9529\pm0.0281)\times10^{-1}$
NLinear	$(4.028\pm0.065)\times10^{-1}$	$(4.743\pm0.031)\times10^{-1}$	$(5.9079\pm0.0123)\times10^{-1}$

five-probe comparison at $F = 20$ is closer: NEURALPDE has lower mean error than the matched ODE at three of five probes, with relative changes between a 0.55% increase and a 1.75% reduction. Appendix B reports the remaining probes, encoder variants, and per-seed long-horizon results. The larger PDE–ODE gap is therefore specific to the longer forecasts at probe 0 rather than uniform across probe locations and horizons. A width-136 ODE supplement at $F = 80$ gives mean sMSE 0.88852, close to 0.88984 for the width-18 control; the PDE remains 52.25% lower (Appendix E.3). Increasing RHS width alone does not remove the gap under this training protocol. Figure 4 shows the decoded velocity trajectories and component forecasts for a fixed validation window.

5.3 ELECTRONIC-OSCILLATOR FORECASTING

The oscillator task uses $H = 52$, $F = 26$, and nested sensor sets $M \in \{1, 2, 4, 8\}$. Repetitions 1 and 2 supply training and validation data. Every network topology and coupling strength occurs in training.

Table 3: Oscillator sMSE (↓), $H = 52, F = 26$. Values are mean ± sample SD over seeds 7/17/27; neural models train for 80 epochs. Bold marks the lowest primary mean per column;

Model	$M = 1$	$M = 2$	$M = 4$	$M = 8$
Primary comparison				
NEURALPDE	$(1.0206\pm0.1072)\times10^{-3}$	$(\mathbf{9.059}\pm0.699)\times10^{-4}$	$(\mathbf{8.8067}\pm0.4452)\times10^{-4}$	$(7.690\pm0.893)\times10^{-4}$
ODE (GRU)	$(\mathbf{1.0101}\pm0.0935)\times10^{-3}$	$(1.0289\pm0.0402)\times10^{-3}$	$(8.9350\pm1.3118)\times10^{-4}$	$(\mathbf{7.520}\pm0.532)\times10^{-4}$
Direct GRU	$(1.0320\pm0.0904)\times10^{-3}$	$(9.378\pm1.158)\times10^{-4}$	$(9.4004\pm0.7115)\times10^{-4}$	$(8.059\pm1.533)\times10^{-4}$
TimeXer	$(1.2665\pm0.0692)\times10^{-2}$	$(1.3266\pm0.0382)\times10^{-2}$	$(1.3179\pm0.0401)\times10^{-2}$	$(1.3239\pm0.0400)\times10^{-2}$
TSMixer	$(2.8213\pm0.1716)\times10^{-3}$	$(2.8492\pm0.1247)\times10^{-3}$	$(2.7896\pm0.1173)\times10^{-3}$	$(2.4886\pm0.0636)\times10^{-3}$
NLinear	$(7.2921\pm0.0475)\times10^{-3}$	$(7.2921\pm0.0475)\times10^{-3}$	$(7.2921\pm0.0475)\times10^{-3}$	$(7.2921\pm0.0475)\times10^{-3}$

The PDE–ODE comparison depends on sensor count. NEURALPDE has 11.96% and 1.44% lower mean error at $M = 2$ and $M = 4$, with lower error in three and two paired seeds, respectively. At $M = 1$ and $M = 8$, its mean error is 1.04% and 2.27% higher than the matched ODE's. It has lower mean error than the direct GRU at all four sensor counts, by 1.10–6.32%, and lower mean error than the tested TimeXer, TSMixer, and NLinear configurations. Appendix C gives the coupling breakdown and periodic and linear controls.

Encoders and ADR components. The cylinder advantage over the matched ODE at $F = 40$ and $F = 80$ persists with GRU, MLP, and RNN encoders (Appendix E); the $F = 20$ results remain mixed across probes. On cylinder $F = 80$, removing advection, diffusion, or both and retraining increases mean error by 97.15%, 72.27%, and 109.88%, respectively. On oscillator $M = 4$, removing diffusion lowers mean error by 3.15%, but improves only one of three seeds. The Burgers encoder extension also favors the PDE over its paired ODE for both MLP and RNN at every sensor count on the previously accessed test trajectories (Appendix F.4).

Sea surface temperature. The OISST experiments cover 7- and 30-day forecasts with $M \in \{1, 2, 4, 8\}$ sensors (Appendix D). The PDE and matched ODE have similar mean errors at seven days, with relative differences below 2%. At 30 days, the ODE has lower mean error for $M = 1, 2$, while the PDE has lower mean error for $M = 4, 8$. The largest reduction is 43.83% at $M = 8$, favoring the PDE in all three seeds; at $M = 2$, its mean error is instead 50.01% higher. The PDE

has lower mean error than the tested TimeXer, TSMixer, and NLinear configurations at every 30-day setting.

5.4 Representation on the disk–annulus task

Following the disk–annulus illustration in Figure 9 of Dupont et al. (2019), we examine whether a small latent function can provide the additional representation capacity discussed in the introduction. Inputs in an inner disk have target -1, and inputs in a surrounding annulus have target $+1$. A fixed angular wedge is held out. We compare a two-dimensional NODE, a three-dimensional ANODE, and NeuralPDE with three real Fourier coefficients, using SiLU activations and a learned affine decoder in each. This illustration uses seed 17, 20 epochs of Huber-loss training, and the final checkpoint. Appendix G describes our adaptation of the ANODE experiment and reports the full protocol.

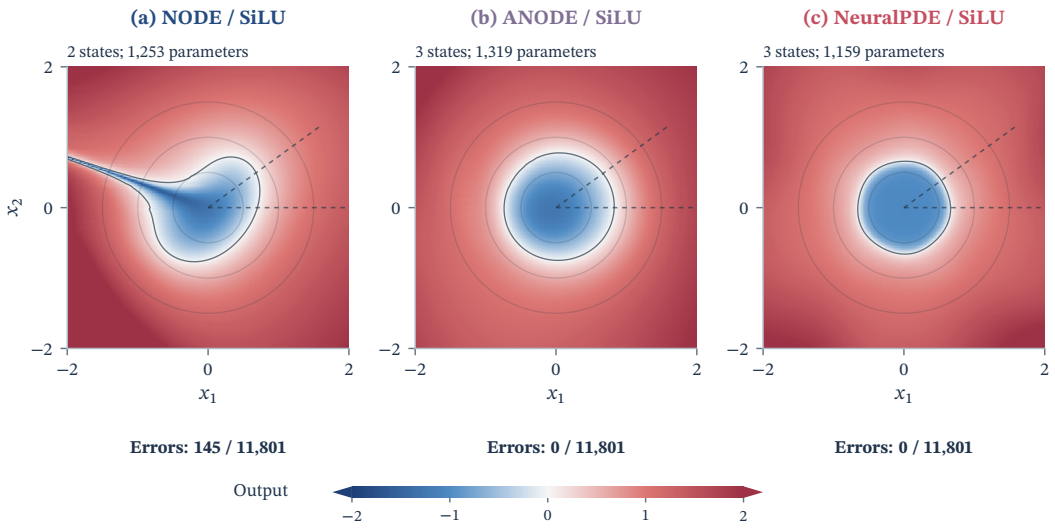


Figure 3: Learned output on the disk–annulus input plane, with SiLU in all three models. Blue and red correspond to negative and positive outputs; targets are -1 in the inner disk and $+1$ in the annulus. Circles mark the labeled regions, dashed rays bound the held-out wedge, and the thin contour marks zero output. Error counts cover the 11,801 labeled points of a fixed 201×201 grid.

The NODE output contains a narrow negative channel through the annulus, whereas ANODE and NeuralPDE separate the two labeled regions on the evaluated grid (Figure 3). Their grid sign-error counts are 145, 0, and 0, respectively. All three classify the 284 held-out sampled points correctly, so the distinction appears in the denser grid evaluation. A unique planar flow preserves the enclosing relationship, which obstructs separation of the complete regions by an affine decoder (Dupont et al., 2019). ANODE and the three-coefficient latent function can use an additional state direction to achieve separation, while their complete states still obey uniqueness. The finite-grid result illustrates this capacity; it does not prove zero error throughout the continuous domain. Figures 5–7 show how the states and latent functions evolve in this example.

6 Discussion and Conclusion

NeuralPDE This finite representation does not necessarily correspond to a physical PDE or guarantee that it retains every historical distinction needed for prediction.

On the held-out Burgers trajectories, NeuralPDE has the lowest mean error among the tested models at all four sensor counts, with a small and seed-dependent advantage over the GRU at $M = 8$. The disk–annulus example also shows that a three-coefficient latent function can form separable representations, like ANODE, without violating complete-state uniqueness. The empirical benefits depend on the task and comparator. The direct GRU has lower mean error on cylinder $F = 80$ test set. On that task, the advantage over the matched dense ODE persists across three encoders. Retraining after removing derivative terms increases error, supporting the value of this structure in that setting.

The broader comparisons show where the same structure is less helpful. The short-horizon cylinder probes and oscillator sensor counts do not uniformly favor the PDE over the matched ODE. The OISST comparison also changes with sensor count: at 30 days, the PDE is worse for $M = 1, 2$ and better for $M = 4, 8$. Removing diffusion lowers mean oscillator error, with improvement in only one seed. These outcomes support a useful, task-dependent inductive bias without establishing a universal advantage for the complete ADR model. Measurements of trained mode use, interventions on learned components, nonlinear and numerical stability checks, and evaluation under new physical conditions are needed to explain how this latent structure supports prediction in practice.

AI USE STATEMENT

Generative AI tools assisted with implementation and testing, methodological feedback, literature discovery, result interpretation, and manuscript drafting and editing. No AI tool is an author. The human authors retain responsibility for verifying the equations, citations, code, numerical results, originality, and complete disclosure of AI use before submission.

REPRODUCIBILITY STATEMENT

We are committed to reproducible research and provide the following resources and details.

Codebase and Implementation. The anonymized implementation is available at `https://anonymous.4open.science/r/NeuralPDE-review-EE53/`. Model configurations, solver settings, random seeds, package versions, and optimization details are documented in the repository and appendix.

Data and Evaluation. Data sources, preprocessing, splits, and train-only normalization are documented. The repository also provides per-seed metrics and evaluation configurations for reproducing the reported results.

Computing Environment and Reproduction Pipeline. Environment dependencies, optimization settings, and numerical solver configurations are documented in the repository and appendix. The released code and per-seed JSON metrics are intended to reproduce the reported training and evaluation pipeline.

REFERENCES

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Si-An Chen, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Boyin Huang, Chunying Liu, Viva Banzon, Eric Freeman, Garrett Graham, Bill Hankins, Tom Smith, and Huai-Min Zhang. Improvements of the daily optimum interpolation sea surface temperature (doisst) version 2.1. *Journal of Climate*, 34(8):2923–2939, 2021.

Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning space-time continuous latent neural pdes from partially observed states. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 26372–26395. Curran Associates, Inc., 2023.

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

Shiyi Qi, Zenglin Xu, Yiduo Li, Liangjian Wen, Qingsong Wen, Qifan Wang, and Yuan Qi. Pdetime: Rethinking long-term multivariate time series forecasting from the perspective of partial differential equations, 2024.

Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Jessica Shang and Jonathan Tu. Particle image velocimetry (PIV) data of flow past a cylinder, June 2026. Dataset.

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

Jonathan H Tu, Clarence W Rowley, J Nathan Kutz, and Jessica K Shang. Spectral analysis of fluid flows using sub-nyquist-rate piv data. *Experiments in Fluids*, 55(9):1805, 2014.

V. P. Vera-Ávila, R. Sevilla-Escoboza, A. A. Lozano-Sánchez, R. R. Rivera-Durón, and J. M. Buldú. Experimental datasets of networks of nonlinear oscillators: Structure and dynamics during the path to synchronization. Zenodo, October 2019. Dataset, version 1.

VP Vera-Ávila, Ricardo Sevilla-Escoboza, AA Lozano-Sánchez, RR Rivera-Durón, and Javier M Buldú. Experimental datasets of networks of nonlinear oscillators: Structure and dynamics during the path to synchronization. *Data in brief*, 28:105012, 2020.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, volume 37, pp. 469–498. Curran Associates, Inc., 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

A Experimental Protocol and Implementation

The experiments address three questions: how the latent evolution law affects prediction when the surrounding architecture is controlled; how the resulting forecasts compare with direct predictors; and how encoder and ADR-component choices affect prediction. The data-specific results are summarized in Section 5, with supplementary results in Appendices B, C, and D. The public Burgers and disk–annulus protocols are given in Appendices F and G. The matched comparison does not isolate the causal contribution of individual ADR terms.

A.1 Matched Neural ODE control

The matched control replaces the ADR RHS with

$$f_\phi(u) = W_3\,\sigma(W_2\,\sigma(W_1 u + b_1) + b_2) + b_3, \tag{11}$$

where the hidden width is selected to minimize the absolute parameter-count difference from the ADR dynamics. In the default configuration $C_z = 8$, $K = 8$, and $q = 136$. The ADR RHS has 5,384 trainable parameters and the dense RHS has 5,392. On the cylinder-wake task the complete models contain 34,322 and 34,330 parameters, respectively. Encoder and decoder parameters are initialized identically for a paired seed; only RHS initialization and functional form differ. This matches the raw RHS parameter budget, not effective rank, FLOPs, or runtime (Section 4).

A.2 Datasets and leakage controls

Experimental cylinder wake. We use 8,000 planar particle-image-velocimetry frames of flow past a finite circular cylinder at Reynolds number 413 and 20 Hz (Shang & Tu, 2026; Tu et al., 2014). Each frame contains streamwise and cross-stream velocity on a 135×80 grid. The forecasting models never see the field. Five fixed probes are selected greedily using only the first 70% of frames, ranking temporal u/v fluctuation energy with a minimum grid-index separation of 12. Each probe is a separate two-variable task and its coordinate and identity are withheld. The split is chronological 70/15/15% by target time; history may extend into the preceding segment, but targets cannot cross a split boundary. Normalization and probe selection use training frames only. Training windows use stride four; validation and test use every forecast origin. A frozen training-only Ridge regression (Hoerl & Kennard, 1970) audit verifies at least 25% MSE skill over persistence for every selected probe at $F = 20$. We use $H = 40$ and $F \in \{20, 40, 80\}$, corresponding to 2 s of history and 1–4 s forecasts. The 20-step task covers all five probes; the 40- and 80-step tasks cover only reference probe zero. Each horizon is trained separately.

Nonlinear electronic oscillator networks. The coupled-sensor protocol was specified before validation. It uses measured voltages from 28 physical Rössler-like electronic oscillators, connected under 20 network configurations and 101 coupling strengths from incoherence to synchronization (Vera-Ávila et al., 2020; 2019). Each condition provides three recordings of 30,000 samples, each containing all 28 node voltages. Complete recordings, rather than overlapping windows, define the split: repetitions 1/2/3 are assigned to train/validation/final test. We retain every temporal sample, use $H = 52$ (approximately two dominant periods) and $F = 26$ (approximately one period), and take 17 forecast origins per network–coupling trajectory, yielding 34,340 windows per split. Node zero, which has degree seven in every topology, is the anchor. Inputs use nested $M \in \{1, 2, 4, 8\}$ prefixes consisting of the anchor followed by its direct neighbors sorted by node index. Connectivity determines this frozen selection but is not supplied to any model; neither node, network, nor coupling identity is an input. A common scalar normalization is fitted using only the anchor in the training repetition. All preprocessing choices and coupling bins were frozen from training-only diagnostics before validation training; those diagnostics estimate a median dominant period of 25.77 samples. Repetition 3 was inspected in earlier model development.

NOAA OISST. NOAA OISST v2.1 combines satellite and in-situ measurements into a daily sea surface temperature analysis (Huang et al., 2021). We use a 1,094-frame Gulf Stream subset. A nested prefix of $M \in \{1, 2, 4, 8\}$ sensors is selected once by training-period temporal variance subject to grid separation. Inputs contain only ordered, individually normalized SST values; latitude, longitude, the grid, and the unobserved field are withheld. The target is the future of slot-zero anchor SST. We

use chronological 70/15/15% splits by target time, allow history from earlier segments, and exclude windows crossing missing dates. The comparisons use $(H, F) = (14, 7)$ and $(60, 30)$, with complete 80-epoch PDE, ODE, TimeXer, TSMixer, and NLinear runs. Earlier direct-GRU references are reported separately with their actual training budgets.

A.3 Models, training policies, and metrics

The comparison includes NeuralPDE, the raw-RHS-parameter-matched Neural ODE, a direct GRU, official-architecture TimeXer, TSMixer, NLinear, and persistence. The two continuous-time models use $C_z = 8$, $K = 8$, a 64-unit encoder, 32 spectral decoder features, and fixed-step RK4 with maximum latent step 0.05. The latent models and direct GRU use AdamW with initial learning rate 10^{-3} and weight decay 10^{-4}. TimeXer, TSMixer, and NLinear use Adam with learning rate 10^{-4} and zero weight decay. Gradient clipping is set to 1 for all of these models. The shared validation-driven scheduler is `ReduceLROnPlateau`, with factor 0.5 and patience 3. The forecasting models minimize the prediction MSE in Equation 7. The forecasting comparisons in this appendix use seeds 7, 17, and 27.

Training budget and model selection. All core neural comparisons complete 80 epochs with early stopping disabled. The validation-driven scheduler remains active. Checkpoints minimize validation forecast sMSE over trained epochs 1–80; the epoch-zero evaluation is excluded. Shared rules need not produce identical learning-rate trajectories, runtime, or optimization quality. The OISST direct-GRU references retain their earlier stopping policy and complete 20–45 epochs; Table 8 lists the per-seed budgets. All reported official NLinear runs complete 80 epochs.

Metrics and evidence scope. Standardized MSE (sMSE) uses the frozen training scale; it is not normalized by persistence error or validation-target energy. The artifact key is `normalized_mse`. Tables report means and sample standard deviations over three training seeds. These measure training variability, not uncertainty over independent physical recordings.

A.4 History encoder and function coordinates

A one-layer GRU reads the ordered sensor features together with the deterministic normalized position increment defined below. A linear projection of its final hidden state produces $u_0 = E_\eta(X_t) \in \mathbb{R}^q$. These coordinates specify a real multichannel function on the auxiliary periodic domain $\mathbb{T} = \mathbb{R}/\mathbb{Z}$:

$$z_0(s) = \sum_{k=-K}^{K} c_k e^{2\pi i k s}, \qquad c_{-k} = \overline{c_k}, \quad c_0 \in \mathbb{R}^{C_z}, \qquad q = C_z(2K+1). \tag{12}$$

Here c_k are continuous Fourier amplitudes. The solver stores their scaled orthonormal-FFT coefficients; Appendix H.1.1 derives the codec and function-norm relation. Conjugate symmetry makes the function real. The GRU initializes all coefficients jointly, without assigning a sensor identity or history age to an auxiliary location. The encoding is not assumed to be invertible or to preserve every distinction in history.

A.5 Time scales, initialization, and numerical implementation

All three datasets use latent query times $\tau_j = 0.05j$, $j = 1, \ldots, F$, and RK4 internal step size 0.05. For cylinder wake this numerical interval equals the physical sampling interval of $1/20$ seconds. For electronic oscillators and OISST, 0.05 is a chosen latent-time scale; an OISST observation still represents one day. The history encoder appends one deterministic normalized position increment to the MC sensor features: zero at the first history step and $1/(H-1)$ at each subsequent step. Thus the GRU input width is $MC + 1$. These increments are independent of the forecast-time scale; no physical history timestamps are passed in these regularly sampled experiments.

The base Fourier grid has $N = 2K + 1 = 17$ points and the padded reaction grid has $n_p = 26$. The implementation uses orthonormal FFTs. Stored coefficients $\widehat{z}_k$ and continuous Fourier amplitudes c_k satisfy $\widehat{z}_k = \sqrt{N} c_k$; padding and truncation use the reciprocal factors $\sqrt{n_p/N}$ and $\sqrt{N/n_p}$. Appendix H.1.1 derives the corresponding function norm. The reaction is an $8 \to 64 \to 64 \to 8$ SiLU network. Initially, $A = 0$, $L = 0.03I$, and $\epsilon = 10^{-4}$, so $B = 10^{-3}I$.

For the latent models, the trainer zero-initializes the final residual output layer and adds the last observed anchor value after decoding. This makes the untrained predictor equal persistence. Predictions are computed from the observed history alone, and gradients from the forecast loss train the encoder, dynamics, and decoder jointly.

A.6 PARAMETER TOTALS AND PROVENANCE

The real latent dimension is $q = 8(2 \cdot 8 + 1) = 136$. For one input channel, the complete NEURALPDE/matched-ODE parameter counts are 34,065/34,073 at $M = 1$ and 35,409/35,417 at $M = 8$. The eight-parameter difference is constant because only the encoder input width changes with M. For the two-channel cylinder task the corresponding totals are 34,322/34,330.

Training checkpoints and validation artifacts store the encoder, dynamics, and decoder counts; latent dimension; RHS class; RK4 step; optimization configuration; data and split fingerprints; sensor order; normalization values; best epoch; and full training history. For paired continuous-time runs, a shared-protocol hash is computed after excluding only the model-specific RHS class and parameter count. Aggregation is permitted only when hashes match.

B CYLINDER-WAKE FORECASTING

B.1 FIVE SEPARATE PROBE TASKS

Tables 2 and 4 together report all five training-selected probes at $H = 40$, $F = 20$. Each model is trained separately for each probe. The PDE has lower mean error than the ODE at probes 0, 2, and 4, and higher mean error at probes 1 and 3. The paired-seed win counts are 1/3, 2/3, 3/3, 1/3, and 2/3. The small mean differences and changing rankings do not support a uniform short-horizon advantage.

Table 4: Additional cylinder sMSE (↓), $H = 40$, 80 epochs; mean ± sample SD over seeds 7/17/27. Primary probe-0 results are in Table 2. Bold compares primary models in (a) and each PDE–ODE encoder pair in (b,c), per column. Encoder variants differ in capacity; GRU remains the primary encoder.

(a) Primary models: remaining probes at $F = 20$

Model	Probe 1	Probe 2	Probe 3	Probe 4
NEURALPDE	$(3.698\pm0.074)\times10^{-1}$	$(1.0133\pm0.0006)\times10^{0}$	$(1.2693\pm0.0006)\times10^{0}$	$(3.981\pm0.082)\times10^{-1}$
ODE (GRU)	$(3.677\pm0.044)\times10^{-1}$	$(1.0141\pm0.0004)\times10^{0}$	$(1.2685\pm0.0011)\times10^{0}$	$(4.052\pm0.093)\times10^{-1}$
Direct GRU	$(\mathbf{3.649}\pm0.015)\times10^{-1}$	$(1.0326\pm0.0003)\times10^{0}$	$(1.2951\pm0.0075)\times10^{0}$	$(\mathbf{3.845}\pm0.063)\times10^{-1}$
TimeXer	$(4.385\pm0.029)\times10^{-1}$	$(1.0444\pm0.0009)\times10^{0}$	$(1.2815\pm0.0049)\times10^{0}$	$(4.622\pm0.033)\times10^{-1}$
TSMixer	$(5.336\pm0.235)\times10^{-1}$	$(\mathbf{1.0131}\pm0.0002)\times10^{0}$	$(\mathbf{1.2662}\pm0.0011)\times10^{0}$	$(6.388\pm0.344)\times10^{-1}$
NLinear	$(5.378\pm0.054)\times10^{-1}$	$(1.1349\pm0.0084)\times10^{0}$	$(1.3861\pm0.0082)\times10^{0}$	$(5.839\pm0.057)\times10^{-1}$

(b) Encoder sensitivity: $F = 20$, probes 0–3

Model	Probe 0	Probe 1	Probe 2	Probe 3
PDE (MLP)	$(\mathbf{2.909}\pm0.011)\times10^{-1}$	$(3.887\pm0.007)\times10^{-1}$	$(1.0562\pm0.0008)\times10^{0}$	$(1.3100\pm0.0058)\times10^{0}$
ODE (MLP)	$(2.918\pm0.008)\times10^{-1}$	$(\mathbf{3.856}\pm0.032)\times10^{-1}$	$(\mathbf{1.0560}\pm0.0014)\times10^{0}$	$(\mathbf{1.3063}\pm0.0068)\times10^{0}$
PDE (RNN)	$(\mathbf{2.736}\pm0.010)\times10^{-1}$	$(3.799\pm0.025)\times10^{-1}$	$(1.0141\pm0.0002)\times10^{0}$	$(\mathbf{1.2647}\pm0.0008)\times10^{0}$
ODE (RNN)	$(2.772\pm0.017)\times10^{-1}$	$(\mathbf{3.710}\pm0.047)\times10^{-1}$	$(\mathbf{1.0138}\pm0.0003)\times10^{0}$	$(1.2666\pm0.0015)\times10^{0}$

(c) Encoder sensitivity: remaining settings

Model	$F = 20$, probe 4	$F = 40$, probe 0	$F = 80$, probe 0
PDE (MLP)	$(\mathbf{4.063}\pm0.029)\times10^{-1}$	$(\mathbf{3.444}\pm0.011)\times10^{-1}$	$(\mathbf{4.4191}\pm0.0331)\times10^{-1}$
ODE (MLP)	$(4.149\pm0.033)\times10^{-1}$	$(3.862\pm0.036)\times10^{-1}$	$(9.0205\pm0.0398)\times10^{-1}$
PDE (RNN)	$(4.077\pm0.038)\times10^{-1}$	$(\mathbf{3.255}\pm0.005)\times10^{-1}$	$(\mathbf{4.1424}\pm0.0463)\times10^{-1}$
ODE (RNN)	$(\mathbf{4.039}\pm0.030)\times10^{-1}$	$(4.307\pm0.530)\times10^{-1}$	$(8.8926\pm0.0374)\times10^{-1}$

B.2 LONGER FORECASTS AT PROBE ZERO

Table 5 gives every paired seed at $F = 40$ and $F = 80$. The PDE has lower error than the matched ODE in all six comparisons. Its advantage over that particular ODE grows with the forecast horizon, while the direct GRU remains competitive. Appendix E reports the encoder and derivative-term studies.

Table 5: Cylinder probe-0 sMSE (↓) by seed. All runs use forecast-MSE training for 80 epochs and validation-selected checkpoints.

F	Seed	NEURALPDE	Matched ODE	Direct GRU
40	7	3.2735×10^{-1}	3.8871×10^{-1}	3.3381×10^{-1}
40	17	3.2817×10^{-1}	3.5632×10^{-1}	3.4191×10^{-1}
40	27	3.3384×10^{-1}	4.2236×10^{-1}	3.3347×10^{-1}
80	7	4.1985×10^{-1}	8.8920×10^{-1}	4.1802×10^{-1}
80	17	4.1989×10^{-1}	8.8329×10^{-1}	4.2377×10^{-1}
80	27	4.3296×10^{-1}	8.9702×10^{-1}	4.1821×10^{-1}

B.3 REMAINING EVALUATION AND DIAGNOSTICS

The cylinder test segment was inspected in earlier development; reusing it would not create a new independent holdout. The width-18 matched ODE is a parameter-budget control. Appendix E.3 separately examines a width-136 RHS at $F = 80$.

B.4 VELOCITY TRAJECTORIES IN A FIXED VALIDATION WINDOW

Figure 4 compares the same observed future with three decoded forecasts at seed 17, using the middle of the 1,121 chronologically ordered validation origins. The checkpoints are those selected by validation error after 80 epochs of forecast-MSE training. The window is fixed by its position, independently of the predictions. In this window, the ratio of predicted to observed temporal standard deviation of v is 0.008 for the matched ODE, 0.817 for NEURALPDE, and 0.892 for the GRU. The latter two retain the oscillation in v, while both miss substantial variation in u. This is a qualitative example; the aggregate comparison remains the sMSE reported above.

These paths are decoded observations of a history-conditioned latent evolution. A crossing in the u–v plane does not establish an intersection of complete latent states. This diagnostic describes the learned forecasts; it does not identify a topological cause of the matched ODE's error.

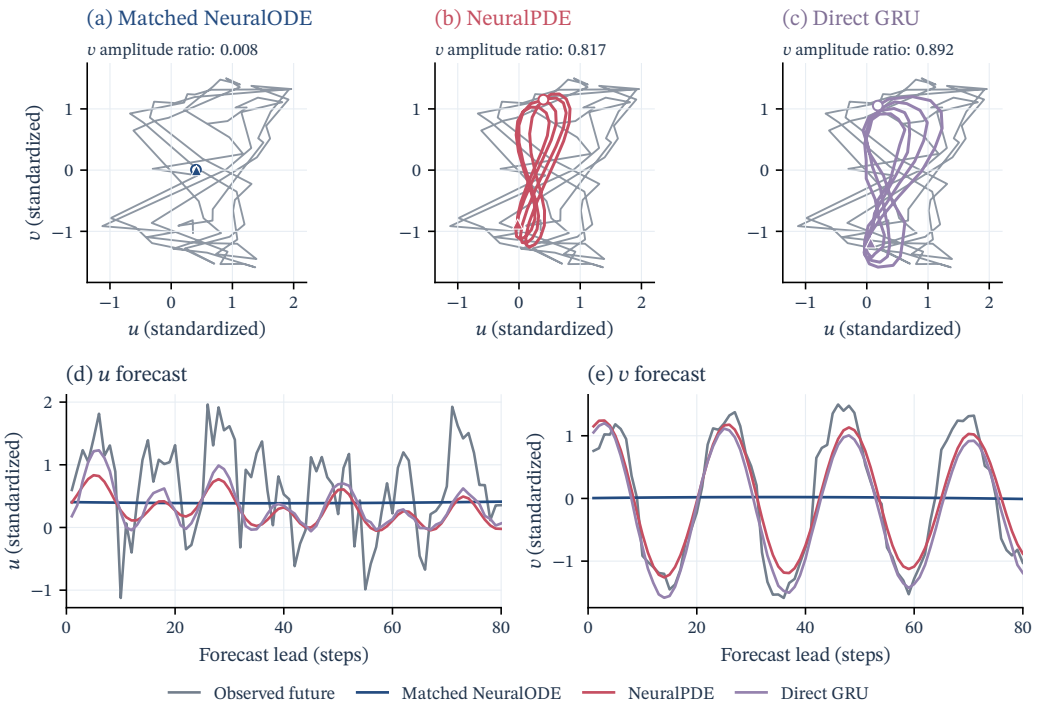


Figure 4: Cylinder velocity paths and forecasts at probe 0, $H = 40, F = 80$, seed 17, validation origin 560 (the middle of the ordered windows). (a–c) Decoded u–v paths with identical axis limits and aspect ratio. Gray curves show the same observed future; colored curves show each model's forecast. Open circles and triangles mark the first and last predicted samples. The v amplitude ratio is the temporal standard deviation of predicted v divided by that of the observed future. (d–e) The same predictions plotted against forecast lead. All values use the frozen training normalization, and curves connect the 80 forecast samples.

C ELECTRONIC-OSCILLATOR FORECASTING

C.1 VALIDATION ACROSS SENSOR COUNTS AND COUPLING STRENGTHS

The PDE–ODE comparison covers all four sensor counts with $H = 52, F = 26$ on repetition 2. Mean error favors the PDE at $M = 2, 4$ and the ODE at $M = 1, 8$ (Table 3). Table 6 stratifies these same selected checkpoints by the coupling bins defined before training.

Table 6: Oscillator sMSE (↓) by coupling strength; mean ± sample SD over seeds 7/17/27. Neural checkpoints and Ridge/LS ($\alpha = 0$) are selected on overall validation. Ridge/LS, Seasonal-26 and persistence are deterministic; the latter two are identical across M.

Model	$M = 1$	$M = 2$	$M = 4$	$M = 8$
(a) Weak coupling: indices 0–20				
NEURALPDE	$(1.9809 \pm 0.2429) \times 10^{-3}$	$(1.6090 \pm 0.1574) \times 10^{-3}$	$(1.5417 \pm 0.0297) \times 10^{-3}$	$(1.5252 \pm 0.1327) \times 10^{-3}$
ODE (GRU)	$(1.9379 \pm 0.2297) \times 10^{-3}$	$(1.8808 \pm 0.0303) \times 10^{-3}$	$(1.5922 \pm 0.2189) \times 10^{-3}$	$(1.4680 \pm 0.1020) \times 10^{-3}$
Direct GRU	$(2.0663 \pm 0.1689) \times 10^{-3}$	$(1.7179 \pm 0.2116) \times 10^{-3}$	$(1.6594 \pm 0.1646) \times 10^{-3}$	$(1.5502 \pm 0.2884) \times 10^{-3}$
NLinear	$(2.3039 \pm 0.0114) \times 10^{-2}$	$(2.3039 \pm 0.0114) \times 10^{-2}$	$(2.3039 \pm 0.0114) \times 10^{-2}$	$(2.3039 \pm 0.0114) \times 10^{-2}$
Ridge/LS	1.5959×10^{-2}	1.5818×10^{-2}	1.5641×10^{-2}	1.5636×10^{-2}
(b) Intermediate coupling: indices 21–50				
NEURALPDE	$(6.215 \pm 0.552) \times 10^{-4}$	$(5.864 \pm 0.491) \times 10^{-4}$	$(5.999 \pm 0.313) \times 10^{-4}$	$(4.299 \pm 0.544) \times 10^{-4}$
ODE (GRU)	$(6.641 \pm 0.746) \times 10^{-4}$	$(7.071 \pm 0.435) \times 10^{-4}$	$(6.281 \pm 1.065) \times 10^{-4}$	$(4.351 \pm 0.383) \times 10^{-4}$
Direct GRU	$(6.535 \pm 0.653) \times 10^{-4}$	$(6.210 \pm 0.832) \times 10^{-4}$	$(6.439 \pm 0.392) \times 10^{-4}$	$(4.669 \pm 1.027) \times 10^{-4}$
NLinear	$(2.3315 \pm 0.0274) \times 10^{-3}$	$(2.3315 \pm 0.0274) \times 10^{-3}$	$(2.3315 \pm 0.0274) \times 10^{-3}$	$(2.3315 \pm 0.0274) \times 10^{-3}$
Ridge/LS	1.8026×10^{-3}	1.7927×10^{-3}	1.7721×10^{-3}	1.7348×10^{-3}
(c) Strong coupling: indices 51–100				
NEURALPDE	$(8.566 \pm 0.853) \times 10^{-4}$	$(8.022 \pm 0.472) \times 10^{-4}$	$(7.715 \pm 0.595) \times 10^{-4}$	$(6.549 \pm 0.926) \times 10^{-4}$
ODE (GRU)	$(8.280 \pm 0.488) \times 10^{-4}$	$(8.641 \pm 0.678) \times 10^{-4}$	$(7.593 \pm 1.096) \times 10^{-4}$	$(6.413 \pm 0.427) \times 10^{-4}$
Direct GRU	$(8.246 \pm 0.742) \times 10^{-4}$	$(8.003 \pm 0.959) \times 10^{-4}$	$(8.156 \pm 0.519) \times 10^{-4}$	$(6.967 \pm 1.275) \times 10^{-4}$
NLinear	$(3.6549 \pm 0.0376) \times 10^{-3}$	$(3.6549 \pm 0.0376) \times 10^{-3}$	$(3.6549 \pm 0.0376) \times 10^{-3}$	$(3.6549 \pm 0.0376) \times 10^{-3}$
Ridge/LS	2.7912×10^{-3}	2.7743×10^{-3}	2.7242×10^{-3}	2.6426×10^{-3}

Every network topology and coupling strength occurs in training, and the validation split is used for model selection. These results do not test transfer to unseen network conditions or establish a causal role for ADR.

C.2 PERIODIC AND LINEAR VALIDATION CONTROLS

Seasonal-26 and persistence are identical across M because they use only the unchanged anchor history.

Each Ridge fit uses all 34,340 repetition-1 windows. Centering each channel at its last observation gives $52M$ features and 26 anchor residual targets; the intercept is unregularized. The expanded grid is $\{0, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000\}$. Positive penalties use float64 SVD; zero uses rank-revealing least squares with relative singular-value cutoff $\max(n, p)\epsilon_{\text{float64}}$ (7.625×10^{-12} here). Every sensor count selects zero by overall repetition-2 sMSE. The zero-penalty fits have numerical rank $51M$: each sensor's last centered feature is identically zero. Their coefficient norms are 8.2×10^4–8.6×10^4, and retained-direction condition numbers are 4.3×10^7–1.1×10^8. These diagnostics document the solution, without establishing robustness to measurement noise.

Table 7 separates the effects of changing the solver and extending the penalty grid. At the original selected penalty 10^{-3}, SVD improves sMSE by 1.16–2.67% relative to the archived float32 LSQR fit with tolerance 10^{-6}. Switching from SVD at 10^{-3} to the unregularized fit improves it by a further 6.08–7.17% relative to that SVD value. A float64 LSQR audit at tolerance 10^{-10} agrees with SVD at 10^{-3}: the maximum absolute difference in sMSE is 2.59×10^{-9} and prediction discrepancy MSE is at most 1.34×10^{-11}. Staying below an iteration cap alone would not establish this agreement.

NLinear uses the official shared, channel-independent linear layer, with 1,378 trainable parameters at every sensor count. All 12 runs complete 80 epochs. Adding auxiliary sensors does not change its anchor prediction function, which explains the nearly identical errors across M. Its centering and

Table 7: Ridge numerical and regularization-boundary audit on validation repetition 2. All fits use repetition 1; the last column is the reduction from the archived LSQR result to the selected zero-penalty result.

M	Old LSQR, $\alpha = 10^{-3}$	SVD, $\alpha = 10^{-3}$	LS, $\alpha = 0$	Reduction
1	5.7060×10^{-3}	5.6397×10^{-3}	5.2355×10^{-3}	8.25%
2	5.6788×10^{-3}	5.5772×10^{-3}	5.1947×10^{-3}	8.52%
4	5.6138×10^{-3}	5.4868×10^{-3}	5.1271×10^{-3}	8.67%
8	5.5507×10^{-3}	5.4027×10^{-3}	5.0745×10^{-3}	8.58%

addition of the last observation are part of the original model. These results and the Ridge control use the same observation protocol as the primary comparisons and do not access repetition 3.

Remaining empirical scope. The study uses one anchor per network and three training seeds. It does not test longer oscillator horizons or unseen network and coupling conditions. Additional direct baselines and retrained ADR-component ablations are reported in Appendix E. The unregularized linear control's large coefficient norms still leave its sensitivity to measurement perturbations unresolved.

D SEA SURFACE TEMPERATURE

D.1 FIXED-BUDGET COMPARISONS

Table 8 reports both horizons and all sensor counts under forecast-MSE training. TimeXer and TSMixer widths are fixed from parameter counts before training, within 1% of the GRU-encoded PDE's total parameter count (Table 10). NLinear retains its native shared linear map. Every run in the fixed-budget blocks completes 80 epochs.

Table 8: OISST sMSE (↓); mean ± sample SD over seeds 7/17/27. Bold compares only the five 80-epoch models. Direct-GRU references use earlier stopping; completed epochs are listed in seed order. Persistence is deterministic.

Model / budget	$M = 1$	$M = 2$	$M = 4$	$M = 8$
(a) $H = 14, F = 7$**; fixed budget: 80 epochs**				
NEURALPDE	$(1.702 \pm 0.033) \times 10^{-2}$	$(1.653 \pm 0.058) \times 10^{-2}$	$(\mathbf{1.398} \pm 0.011) \times 10^{-2}$	$(1.500 \pm 0.063) \times 10^{-2}$
ODE (GRU)	$(1.686 \pm 0.056) \times 10^{-2}$	$(\mathbf{1.622} \pm 0.077) \times 10^{-2}$	$(1.401 \pm 0.041) \times 10^{-2}$	$(1.512 \pm 0.067) \times 10^{-2}$
TimeXer	$(\mathbf{1.506} \pm 0.042) \times 10^{-2}$	$(1.678 \pm 0.089) \times 10^{-2}$	$(1.611 \pm 0.089) \times 10^{-2}$	$(\mathbf{1.449} \pm 0.021) \times 10^{-2}$
TSMixer	$(1.0017 \pm 0.3502) \times 10^{-1}$	$(8.171 \pm 2.244) \times 10^{-2}$	$(5.375 \pm 2.779) \times 10^{-2}$	$(1.1223 \pm 0.1121) \times 10^{-1}$
NLinear	$(2.692 \pm 0.347) \times 10^{-2}$	$(2.692 \pm 0.347) \times 10^{-2}$	$(2.692 \pm 0.347) \times 10^{-2}$	$(2.692 \pm 0.347) \times 10^{-2}$
Direct GRU	$(1.422 \pm 0.069) \times 10^{-2}$	$(1.384 \pm 0.082) \times 10^{-2}$	$(1.336 \pm 0.032) \times 10^{-2}$	$(1.544 \pm 0.056) \times 10^{-2}$
(b) $H = 60, F = 30$**; fixed budget: 80 epochs**				
NEURALPDE	$(2.764 \pm 0.019) \times 10^{-2}$	$(3.524 \pm 0.611) \times 10^{-2}$	$(\mathbf{2.582} \pm 0.446) \times 10^{-2}$	$(\mathbf{3.353} \pm 0.731) \times 10^{-2}$
ODE (GRU)	$(\mathbf{2.586} \pm 0.401) \times 10^{-2}$	$(\mathbf{2.349} \pm 0.130) \times 10^{-2}$	$(2.683 \pm 0.288) \times 10^{-2}$	$(5.969 \pm 2.835) \times 10^{-2}$
Direct GRU	$(2.855 \pm 0.398) \times 10^{-2}$	$(2.067 \pm 0.197) \times 10^{-2}$	$(2.274 \pm 0.085) \times 10^{-2}$	$(1.3271 \pm 0.0981) \times 10^{-1}$
TimeXer	$(4.085 \pm 0.136) \times 10^{-2}$	$(4.180 \pm 0.163) \times 10^{-2}$	$(4.223 \pm 0.207) \times 10^{-2}$	$(4.197 \pm 0.118) \times 10^{-2}$
TSMixer	$(5.517 \pm 0.920) \times 10^{-2}$	$(5.078 \pm 0.639) \times 10^{-2}$	$(4.554 \pm 0.683) \times 10^{-2}$	$(3.3023 \pm 0.7848) \times 10^{-1}$
NLinear	$(5.650 \pm 0.147) \times 10^{-2}$	$(5.650 \pm 0.147) \times 10^{-2}$	$(5.650 \pm 0.147) \times 10^{-2}$	$(5.650 \pm 0.147) \times 10^{-2}$

At seven days, mean differences between the PDE and matched ODE are below 2%; TimeXer has lower mean error than both at $M = 1, 8$. At 30 days, the PDE has 6.90% and 50.01% higher mean error than the ODE at $M = 1, 2$, but 3.76% and 43.83% lower mean error at $M = 4, 8$. It has lower mean error than TimeXer, TSMixer, and NLinear at all four 30-day settings. Increasing the sensor count does not consistently improve either latent model, so these results do not establish a monotonic benefit from additional observations.

D.2 EARLIER DIRECT-GRU REFERENCES

The archived direct-GRU runs use the same observations and normalization but stop after 20–45 epochs. Table 8 lists their actual budgets; they are not an equal-budget comparison with the 80-epoch blocks in the same table. Persistence is deterministic and is unchanged across M because the target anchor is fixed.

E Prediction-Only Baselines and Structural Ablations

E.1 Scope and protocol

These experiments compare history encoders, direct forecasting models, and retrained ADR ablations under the forecast-MSE objective. The sensor comparisons cover seven cylinder tasks, four oscillator sensor counts, and eight OISST settings. The ADR ablations use cylinder probe 0 at $H = 40, F = 80$ and oscillator $M = 4, H = 52, F = 26$. Results use seeds 7, 17, and 27, with means and sample standard deviations computed only when all three runs are complete. The direct-GRU OISST references occupy separate blocks because of their shorter budgets.

All other runs complete 80 epochs without early stopping. The selected checkpoint minimizes validation forecast sMSE over trained epochs 1–80; epoch 0 is excluded, and validation also informs the learning-rate scheduler. The models share observation windows, ordered sensors, split assignments, and training normalization. Latent models and the direct GRU use an anchor residual forecast. The official direct baselines retain their own normalization, initialization, and output conventions. The latent models retain RK4 with step size 0.05, $C_z = 8$, $K = 8$, and the global spectral decoder. Equal epochs do not imply equal runtime, convergence, or tuning opportunity.

E.2 Encoder sensitivity

The encoder study replaces the 64-unit GRU with a single 64-unit tanh RNN or an MLP mapping the flattened ordered history and time increment through a 64-unit hidden layer to the 136 Fourier coordinates. Each encoder is paired with the ADR generator and the matched dense ODE, retaining the decoder and training rules. Encoder changes affect both architecture and parameter count, so this is not a pure capacity comparison.

On cylinder $F = 80$, the GRU-, MLP-, and RNN-encoded PDEs reduce mean error relative to their paired ODEs by 52.32%, 51.01%, and 53.42%, respectively, with all three seeds favoring the PDE. At $F = 40$, the corresponding reductions are 15.25%, 10.83%, and 24.41%, again with all seeds favoring the PDE. The five-probe $F = 20$ results are mixed (Table 4). All these ODE controls use hidden width 18; the separate width-136 comparison is reported in Appendix E.3.

At oscillator $M = 4$, the GRU-, MLP-, and RNN-encoded PDEs reduce mean error by 1.44%, 15.42%, and 1.60% relative to their paired ODEs. The GRU and RNN comparisons favor the PDE in two seeds each, and the MLP comparison in all three. The best PDE encoder differs by task: RNN on cylinder $F = 80$ and MLP on oscillator $M = 4$. The original GRU remains the model used in the primary comparison; these encoder outcomes are not used to replace it after evaluation.

E.3 Wider dense-ODE supplement

To examine whether the cylinder $F = 80$ gap is explained by the narrow matched RHS, we increase both dense hidden layers from 18 to 136, equal to the latent-state dimension. The GRU encoder, global decoder, training data, forecast-MSE objective, solver, and optimizer settings remain fixed. The three seeds complete 80 epochs with early stopping disabled. Width 136 was fixed before these runs; no width or learning-rate search is performed for this supplement. Its larger parameter budget is reported separately from the matched comparison.

Table 9: Cylinder $F = 80$ capacity control: sMSE (↓), mean ± sample SD over three seeds. All runs train for 80 epochs; best epochs follow seed order 7/17/27. PDE width refers to the reaction network.

Model	RHS width	RHS params	Total params	Best epochs	sMSE
NeuralPDE	64	5,384	34,322	66/79/52	$(4.2423 \pm 0.0756) \times 10^{-1}$
ODE, matched	18	5,392	34,330	12/14/35	$(8.8984 \pm 0.0689) \times 10^{-1}$
ODE, wider	136	55,896	84,834	12/14/11	$(8.8852 \pm 0.0551) \times 10^{-1}$

The wider ODE reduces mean error by only 0.15% relative to the matched ODE. The PDE remains 52.25% lower in mean error, with lower error in all three paired seeds. Thus increasing dense RHS width alone does not close the gap in this experiment. This does not rule out optimization limitations

or other ODE architectures and training recipes; it does not establish a general representational impossibility for Neural ODEs.

E.4 Official direct baselines and capacities

TimeXer (Wang et al., 2024) uses the official implementation at source revision `7601190`, preserving instance normalization, native output initialization, output dropout, and the attention blocks. For multivariate targets we use its multivariate branch and select the anchor outputs. For a scalar anchor with auxiliary sensors, the anchor is reordered to the target position required by the official endogenous/exogenous interface. There is no added external anchor residual. The model uses two layers, four heads, GELU, and dropout 0.1.

TSMixer (Chen et al., 2023) uses a numerically verified PyTorch port of Google Research's basic implementation at revision `5b09c22`. It retains two residual blocks, the single linear temporal mixing layer, ReLU, the original BatchNorm axes, dropout 0.05, and a shared direct history-to-horizon head, without RevIN. Initialization and output conventions follow the upstream model.

For cylinder $F = 20$ and both OISST horizons, the two models' widths are chosen solely to match the GRU-encoded PDE's total parameter count within 1%, without training candidates or inspecting their errors. TSMixer varies its feature width while retaining two blocks. TimeXer varies model width in multiples of four from 8 to 64 and feedforward width in multiples of four between two and eight times model width. The largest actual parameter-count difference is 0.476%. Its patch length is $\gcd(4, H)$: two for $H = 14$ and four otherwise, so no historical samples are dropped.

The earlier completed cylinder $F = 40, 80$ and oscillator baselines retain their fixed configurations: TSMixer feature width 2048 and TimeXer model width 32/feedforward width 128. These six settings are not exactly parameter matched. Table 10 reports every task's actual count; the eight-parameter PDE–ODE gap is unchanged.

NLinear (Zeng et al., 2023) preserves its official channel-independent linear forecast with shared weights, last-value centering, and addition of that value to the output, at revision `0c11366`. Its $F(H + 1)$ parameters do not grow with the number of observed channels. We retain this native small model instead of adding layers to match a nonlinear model's capacity. The three official baselines use Adam at 10^{-4} with zero weight decay, the common 80-epoch budget, plateau scheduler, and clipping at 1. No baseline receives physical coordinates, network identity, or governing equations.

The GRU-encoded PDE has 20.34% lower mean cylinder $F = 80$ error and 93.32% lower oscillator $M = 4$ error than the fixed TimeXer. Its corresponding reductions relative to TSMixer are 14.35% and 68.43%. The direct GRU has lower mean error than these official direct baselines on all six settings in Tables 2 and 3. These results characterize the specified models and training recipes, not the performance limits of their families. The earlier TSMixer runs select epochs 78–80 and still improve during the last ten epochs; 80 epochs therefore do not establish convergence. NeuralPDE has also undergone more prior model selection, so the comparisons do not establish equal cumulative tuning opportunity.

E.5 Removing the derivative terms

The component study retrains three variants with the original GRU encoder: without advection, without diffusion, and with reaction alone. Removing a term removes its parameters and its contribution to the RHS; removing diffusion also removes the fixed positive diffusion floor. The reaction network, encoder, decoder, loss, and remaining training settings follow the full model. These ablations change the trainable parameter count and the optimization path. They test the effect of allowing each component to participate in training, not the effect of disabling it in a fixed trained model.

On cylinder, removing advection, diffusion, or both raises mean sMSE by 97.15%, 72.27%, and 109.88%, respectively. All three seeds worsen in each comparison, although the no-diffusion model has substantially larger seed variability. This supports the contribution of the derivative-term parameterization to cylinder prediction under the tested protocol.

On oscillators, removing advection or both derivative terms increases mean error by 11.82% and 13.12%, with all seeds worsening. Removing diffusion instead lowers mean error by 3.15%. That reduction is driven by seed 17: its error falls by 21.58%, while seeds 7 and 27 worsen by 0.35%

Table 10: Trainable parameter counts. TimeXer and TSMixer match the GRU-encoded PDE within 1% only on cylinder $F = 20$ and OISST. Other settings retain their fixed configurations. GRU denotes the direct forecaster.

Task	NEURALPDE	ODE	GRU	TimeXer	TSMixer	NLinear
Cylinder						
$F = 20$, each probe	34,322	34,330	19,944	34,292	34,324	820
$F = 40$	34,322	34,330	22,544	49,672	26,044	1,640
$F = 80$	34,322	34,330	27,744	63,792	27,684	3,280
Electronic oscillators						
$M = 1$	34,065	34,073	18,842	47,578	19,596	1,378
$M = 2$	34,257	34,265	19,034	47,578	28,206	1,378
$M = 4$	34,641	34,649	19,418	47,578	45,426	1,378
$M = 8$	35,409	35,417	20,186	47,578	79,866	1,378
OISST						
$F = 7$, $M = 1$	34,065	34,073	17,607	34,215	34,065	105
$F = 7$, $M = 2$	34,257	34,265	17,799	34,215	34,253	105
$F = 7$, $M = 4$	34,641	34,649	18,183	34,671	34,641	105
$F = 7$, $M = 8$	35,409	35,417	18,951	35,383	35,403	105
$F = 30$, $M = 1$	34,065	34,073	19,102	34,094	34,064	1,830
$F = 30$, $M = 2$	34,257	34,265	19,294	34,094	34,254	1,830
$F = 30$, $M = 4$	34,641	34,649	19,678	34,486	34,640	1,830
$F = 30$, $M = 8$	35,409	35,417	20,446	35,270	35,412	1,830
Burgers						
$M = 1$	34,065	34,073	18,452	38,740	13,710	420
$M = 2$	34,257	34,265	18,644	38,740	22,064	420
$M = 4$	34,641	34,649	19,028	38,740	38,772	420
$M = 8$	35,409	35,417	19,796	38,740	72,188	420

Table 11: Retrained ADR ablations: sMSE (↓), mean ± sample SD over seeds 7/17/27. All use the GRU encoder and prediction-only training for 80 epochs. Δ is the percentage change relative to full ADR; negative values favor the ablation.

	Cylinder: $F = 80$		Oscillator: $M = 4$	
Variant	sMSE ↓	Δ (%)	sMSE ↓	Δ (%)
Full ADR	$(4.2423 \pm 0.0756) \times 10^{-1}$	0.00	$(8.8067 \pm 0.4452) \times 10^{-4}$	0.00
Without advection	$(8.3638 \pm 0.0615) \times 10^{-1}$	+97.15	$(9.8475 \pm 0.3353) \times 10^{-4}$	+11.82
Without diffusion	$(7.3081 \pm 1.5953) \times 10^{-1}$	+72.27	$(8.5295 \pm 1.1516) \times 10^{-4}$	−3.15
Reaction only	$(8.9036 \pm 0.0291) \times 10^{-1}$	+109.88	$(9.9618 \pm 0.8685) \times 10^{-4}$	+13.12

and 13.59%. Thus the complete ADR model is not the best variant in mean oscillator error, and the benefit of diffusion depends on the task and seed.

The reaction-only model is a finite-dimensional structured ODE after Fourier discretization. It retains a pointwise shared reaction network and spectral projection, so it is not the dense matched-ODE baseline. Similar prediction errors would not establish equivalence of their parameterizations or learned dynamics. More generally, retraining ablations do not establish that a learned auxiliary operator recovers a physical governing law. That claim would require independent evidence about physical variables and dynamics, beyond forecast error on these forecasting tasks.

F PUBLIC BURGERS SPARSE-SENSOR FORECASTING

F.1 DATA AND EVALUATION PROTOCOL

We use the version-8 PDEBench file `1D_Burgers_Sols_Nu0.04.hdf5`, distributed through DaRUS under CC BY 4.0 (Takamoto et al., 2022). The data solve

$$\partial_t u + u\,\partial_x u = \frac{0.04}{\pi}\partial_{xx} u$$

on a periodic unit interval. There are 10,000 trajectories, 1,024 spatial cells, and 201 stored frames with physical spacing $\Delta t = 0.01$. The file contains one extra time coordinate without a solution frame; we use the 201 coordinates that correspond to stored frames. This is a forecasting task on public numerical solutions, with a sparse observation interface defined here.

The trajectory split uses NumPy's default random generator with seed 20260924: 8,000 training, 1,000 validation, and 1,000 test trajectories. Each trajectory contributes one window: indices 0–19 are the history, and indices 20–39 are the forecast targets. The ordered sensor bank is $[0, 512, 256, 768, 128, 640, 384, 896]$; $M = 1, 2, 4, 8$ uses its respective prefix, and the prediction target is always sensor 0. Only training trajectories are used to fit the anchor mean and standard deviation, which normalize all sensor channels. Windows from a trajectory never cross data splits. No model receives coordinates, viscosity, or other equation information.

The 60 model–sensor–seed configurations were fixed before test evaluation. Each was trained for 80 epochs, and its checkpoint was selected by validation error. The saved checkpoints were then evaluated on all 1,000 test trajectories, after reproducing their validation metrics on the evaluation machine. No training or checkpoint selection used these test errors. We report sMSE on the training normalization scale, pooling all 20 target times and all test trajectories before aggregating the three training seeds. It is not a per-trajectory relative L^2 error.

Table 12: Burgers sMSE (↓), mean ± sample SD over seeds 7/17/27, on the training-anchor scale. Panels separate validation, original test results and the post-hoc encoder study. Bold marks column minima in (a,b) and the lower mean in each encoder pair in (c). NLinear is validation-only; persistence is deterministic.

Model	$M = 1$	$M = 2$	$M = 4$	$M = 8$
(a) Validation: 80-epoch runs				
NEURALPDE	$(\mathbf{1.7149}\pm0.1285)\times10^{-2}$	$(\mathbf{1.5604}\pm0.0885)\times10^{-2}$	$(\mathbf{1.531}\pm0.179)\times10^{-3}$	$(\mathbf{4.44}\pm0.79)\times10^{-4}$
ODE (GRU)	$(2.0868\pm0.1929)\times10^{-2}$	$(1.6571\pm0.0917)\times10^{-2}$	$(1.842\pm0.296)\times10^{-3}$	$(5.40\pm0.86)\times10^{-4}$
Direct GRU	$(1.9043\pm0.0958)\times10^{-2}$	$(1.6348\pm0.0848)\times10^{-2}$	$(2.246\pm0.033)\times10^{-3}$	$(4.90\pm1.21)\times10^{-4}$
TimeXer	$(5.4674\pm0.1194)\times10^{-2}$	$(5.5362\pm0.0644)\times10^{-2}$	$(5.2649\pm0.1295)\times10^{-2}$	$(5.1803\pm0.0905)\times10^{-2}$
TSMixer	$(6.1819\pm0.3336)\times10^{-2}$	$(6.0417\pm0.1996)\times10^{-2}$	$(2.5561\pm0.2864)\times10^{-2}$	$(1.1818\pm0.3735)\times10^{-2}$
NLinear	$(5.8723\pm0.0285)\times10^{-2}$	$(5.8723\pm0.0285)\times10^{-2}$	$(5.8723\pm0.0285)\times10^{-2}$	$(5.8723\pm0.0285)\times10^{-2}$
(b) Original five-model test comparison				
NEURALPDE	$(\mathbf{1.6044}\pm0.0915)\times10^{-2}$	$(\mathbf{1.4625}\pm0.0314)\times10^{-2}$	$(\mathbf{1.492}\pm0.355)\times10^{-3}$	$(\mathbf{3.93}\pm0.54)\times10^{-4}$
Matched ODE	$(1.8731\pm0.1046)\times10^{-2}$	$(1.5786\pm0.0387)\times10^{-2}$	$(1.670\pm0.245)\times10^{-3}$	$(4.55\pm0.36)\times10^{-4}$
Direct GRU	$(1.7937\pm0.0629)\times10^{-2}$	$(1.5775\pm0.0511)\times10^{-2}$	$(2.063\pm0.181)\times10^{-3}$	$(4.05\pm0.85)\times10^{-4}$
TSMixer	$(5.0199\pm0.3417)\times10^{-2}$	$(5.1806\pm0.1382)\times10^{-2}$	$(2.3616\pm0.1658)\times10^{-2}$	$(1.1579\pm0.3701)\times10^{-2}$
TimeXer	$(4.6445\pm0.0734)\times10^{-2}$	$(4.5993\pm0.0582)\times10^{-2}$	$(4.4495\pm0.0743)\times10^{-2}$	$(4.3398\pm0.0432)\times10^{-2}$
(c) Post-hoc encoder extension: previously accessed test trajectories				
PDE (MLP)	$(\mathbf{2.4811}\pm0.0430)\times10^{-2}$	$(\mathbf{2.4697}\pm0.0326)\times10^{-2}$	$(\mathbf{2.151}\pm0.186)\times10^{-3}$	$(\mathbf{4.52}\pm0.94)\times10^{-4}$
ODE (MLP)	$(2.6373\pm0.1473)\times10^{-2}$	$(2.4993\pm0.1249)\times10^{-2}$	$(2.988\pm0.718)\times10^{-3}$	$(6.51\pm1.04)\times10^{-4}$
PDE (RNN)	$(\mathbf{1.5198}\pm0.0077)\times10^{-2}$	$(\mathbf{1.5643}\pm0.1926)\times10^{-2}$	$(\mathbf{1.974}\pm0.311)\times10^{-3}$	$(\mathbf{7.32}\pm0.72)\times10^{-4}$
ODE (RNN)	$(1.6724\pm0.0289)\times10^{-2}$	$(1.8116\pm0.0120)\times10^{-2}$	$(2.666\pm0.410)\times10^{-3}$	$(9.19\pm1.10)\times10^{-4}$

F.2 MODELS, CAPACITY, AND OPTIMIZATION

The latent models use $C_z = 8$, $K = 8$, a 64-unit GRU, 32 decoder features, the global MLP decoder, and RK4 with maximum latent step 0.05. The dense matched ODE has hidden width 18 and eight more total parameters than the PDE at each M. Physical sampling time and latent integration time remain separate. The direct GRU predicts the full horizon. These three models use AdamW with learning rate 10^{-3} and weight decay 10^{-4}.

TSMixer retains the official basic architecture described in Appendix E: two blocks, feature width 2048, ReLU, BatchNorm, dropout 0.05, and direct outputs. TimeXer uses the official architecture at source revision `7601190`, with model width 32, feedforward width 128, four heads, two layers, patch length four, and dropout 0.1. It preserves the native normalization, initialization, and output convention. It uses the same fixed small configuration as the cylinder $F = 40, 80$ and oscillator comparisons. Both official baselines use Adam with learning rate 10^{-4} and zero weight decay.

All five models minimize forecast MSE with batch size 64, gradient clipping at 1, and a validation-driven plateau scheduler with factor 0.5 and patience 3. Seeds are 7, 17, and 27; early stopping is disabled. Configuration and optimizer differences are retained, so the shared epoch budget does not imply equal computation, tuning opportunity, or convergence. TSMixer's selected checkpoints occur at epochs 77–80. The parameter counts in Table 10 describe capacity, not runtime.

Scope of the result. In the original five-model test comparison, the PDE has lower mean error than each baseline at all four sensor counts. Against the matched ODE, the paired-seed win counts are 3/3, 3/3, 2/3, and 2/3; against the GRU they are 3/3, 3/3, 3/3, and 1/3. In particular, the $M = 8$ advantage over the GRU is a small mean difference, not a uniform improvement. These results concern one viscosity, a fixed initial window, and a 20-step forecast. They do not evaluate transfer across equations or coefficients, long rollouts, or recovery of a physical governing law.

F.3 NLinear validation supplement

The additional official NLinear runs complete 80 epochs and use the same trajectory split and sparse observations. Table 12 panel (a) compares them with the validation errors of the five original models. NLinear uses 420 parameters and the same anchor-only prediction function at every M. Its mean validation error is higher than that of the PDE, matched ODE, and direct GRU at all four sensor counts. NLinear has not been evaluated on the test trajectories and appears only in the validation panel of Table 12.

F.4 Post-hoc encoder test comparison

The MLP/RNN extension contains 48 additional runs: two encoders, two evolution laws, four sensor counts, and three seeds. All complete 80 epochs under the same forecast-MSE protocol, with checkpoint selection based on validation. The saved checkpoints first reproduce their validation metrics, then predict the same 1,000 test trajectories used in the original comparison. This extension was specified after the original test results had been inspected; it is a post-hoc sensitivity study on that test set, not a fresh independent holdout. All specified variants and seeds are reported in Table 12(c), without selecting an encoder using test error.

The PDE has lower mean test error than its paired ODE at every sensor count for both encoders. Relative reductions are 5.92%, 1.18%, 28.01%, and 30.65% for MLP, and 9.13%, 13.65%, 25.95%, and 20.41% for RNN. All three seeds favor the PDE except MLP at $M = 2$, where two do. These comparisons retain the width-18 ODE; they support persistence of the observed difference across encoders without resolving RHS-capacity limitations or establishing superiority over the entire ODE family. The MLP PDE has 23,633/24,913/27,473/32,593 parameters at $M = 1/2/4/8$; the RNN PDE has 25,361/25,425/25,553/25,809. Each paired ODE has eight more.

G Disk–Annulus Representation and Latent-Function Visualizations

G.1 Fixed protocol and complete results

This experiment is inspired by the input-space response maps and angular holdout in Figure 9 of Dupont et al. (2019). We retain the nested class geometry and affine readout, and extend the comparison to the ADR latent function. The additional coordinate in ANODE allows the two classes to separate outside the input plane. Here we examine whether the Fourier coefficients of a latent function provide analogous capacity. We use SiLU throughout and the fixed protocol below, so this is an adaptation of the ANODE experiment rather than an exact reproduction of its results.

The task uses scalar target fitting on synthetic points, separately from the forecasting protocol. We sample 1,000 inner-disk points with radius in $[0, 0.5]$ and target -1, and 2,000 outer-annulus points with radius in $[1, 1.5]$ and target $+1$. Directions are normalized two-dimensional standard Gaussian draws, and radii follow $r = r_{\min} + (r_{\max} - r_{\min})\sqrt{U}$ with $U \sim \mathrm{Uniform}(0, 1)$. Data seed 20260925 is fixed. The angular sector $[0, \pi/5]$ supplies 284 held-out points; the other 2,716 points are used for training. The annular sampling rule is stated explicitly because it is not uniform in annular area.

NODE evolves (x_1, x_2) directly. ANODE and NeuralPDE use the fixed lift $(0, x_1, x_2)$ and three complete state coordinates. For ANODE, this is zero augmentation up to a coordinate permutation; for NeuralPDE, the prepended zero initializes the DC coefficient. The NODE/ANODE vector fields are two-hidden-layer, width-32 networks with explicit time input and SiLU activations. The PDE is autonomous and uses one latent channel, the DC and first Fourier mode, and a width-32 SiLU reaction network, with advection and diffusion enabled and no nonlocal operator. Its state is $(a_0, \mathrm{Re}\, a_1, \mathrm{Im}\, a_1)$. These coefficients describe a function of the auxiliary coordinate s; they do not denote three physical spatial coordinates. Every model uses a learned affine scalar decoder, without a nonlinear encoder or decoder. All three models use SiLU, but their dynamics and parameter counts differ; this is not an exact parameter-matched ablation.

All runs use seed 17, Adam with learning rate 10^{-3}, batch size 64, and 20 epochs of SmoothL1/Huber loss with threshold 1. We integrate over $[0, 1]$ with `dopri5`, absolute and relative tolerances 10^{-3}, and direct differentiation. The final epoch is used without checkpoint selection. The input–output map is evaluated on a 201×201 grid over $[-2, 2]^2$; only the 11,801 points in the disk or annulus have labels.

Table 13: Disk–annulus results with SiLU in all three models, at seed 17 and final epoch 20. Grid errors count sign mistakes on 11,801 labeled points at solver tolerance 10^{-3}. All three models have 100% sign accuracy on the 284 held-out sampled points.

Model	State dimension	Parameters	Held-out Huber	Grid errors
NODE / SiLU	2	1,253	2.8700×10^{-2}	145
ANODE / SiLU	3	1,319	1.6039×10^{-2}	0
NeuralPDE / SiLU	3	1,159	3.434×10^{-3}	0

Re-evaluating the same SiLU checkpoints with both solver tolerances set to 10^{-5} changes the NODE grid error count from 145 to 149, while ANODE and NeuralPDE remain at zero. This supports the qualitative observation across the checked tolerances. Error counts also depend on optimization and discretization; the picture alone does not identify their causal contributions or prove a topological theorem.

G.2 Complete-state evolution

Figure 5 follows the same 384 inputs, selected as 192 evenly spaced source indices from each class, through NODE, ANODE, and NeuralPDE, all using SiLU. Figure 6 instead uses eight equally spaced angles at each of the fixed radii 0.35 and 1.25. Both selections are independent of predictions. The archived trajectories use 101 integration output times and solver tolerances 10^{-5}. The gray line or plane marks the learned affine zero-output boundary.

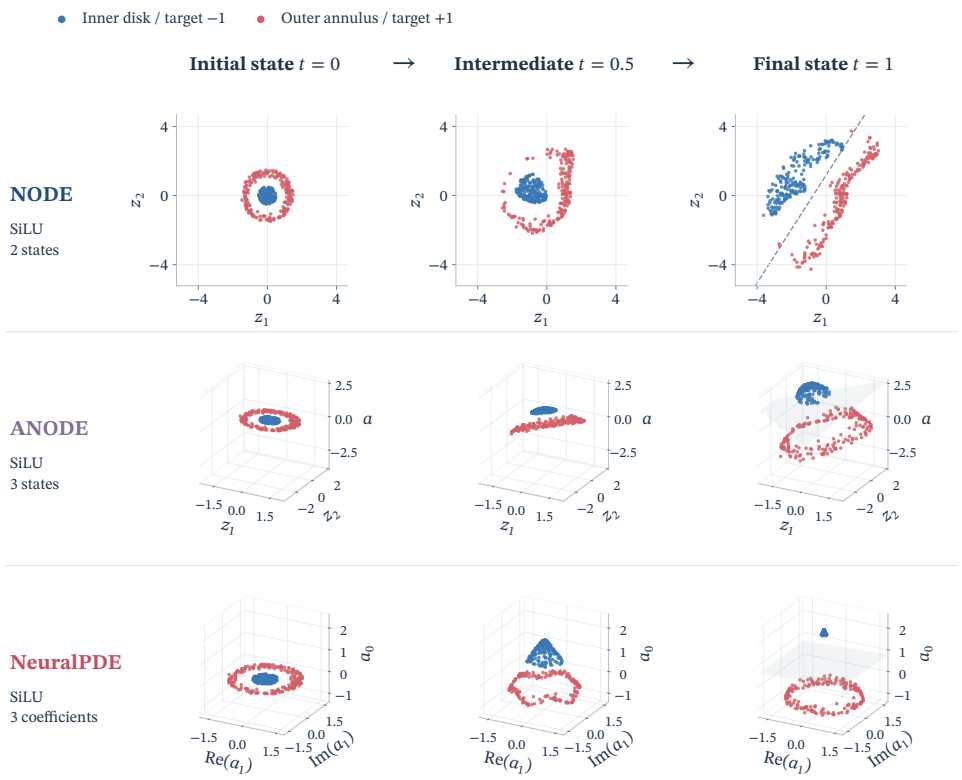


Figure 5: State evolution for a fixed set of disk and annulus inputs. Rows show NODE, ANODE, and NeuralPDE, all using SiLU; columns show $t = 0, 0.5, 1$. Blue points are from the disk and red points from the annulus. NODE uses two state coordinates; ANODE and NeuralPDE use three. The latter views are orthographic projections of complete states, without PCA. Limits are fixed within each row, with separate ranges across models. The final gray boundary marks where the affine decoder output equals zero.

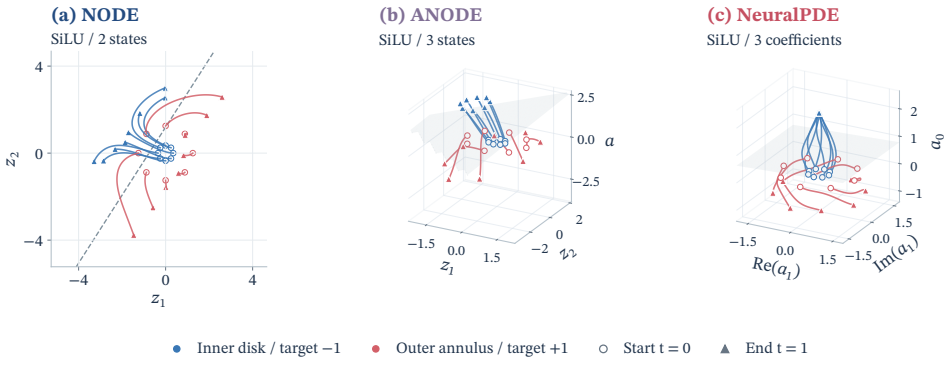


Figure 6: Trajectories for 16 geometrically fixed inputs, with the same input points in each of the three SiLU models. Hollow circles mark $t = 0$, triangles mark $t = 1$, and gray lines or planes mark zero decoder output. The three-dimensional models can separate states along an additional coordinate. Apparent crossings in a projection do not imply that complete states intersect.

In this run, the PDE's learned evolution changes both the function mean and first-harmonic amplitude while separating the displayed inputs. On the displayed disk samples, the median function mean changes from 0 to 1.271 and the median harmonic amplitude from 0.429 to 0.030. On the annulus samples, the corresponding values change from 0 to -0.462 and from 1.550 to 1.707. Thus this learned evolution retains distinct function profiles for the two classes; it does not simply smooth every input to the same constant. These are descriptive statistics for the fixed displayed samples.

G.3 Latent functions and decoder

Figure 7 makes the auxiliary coordinate explicit. It uses two inputs fixed by geometry: angle $\pi/10$, the midpoint of the held-out sector, and radii 0.35 and 1.25. In the figure, $u(s,t)$ denotes the scalar latent function $z(t,s)$, and t is latent evolution time. The orthonormal length-three Fourier representation reconstructs it as

$$u(s,t) = \frac{a_0(t) + 2\,\mathrm{Re}\{a_1(t)e^{2\pi \mathrm{i} s}\}}{\sqrt{3}}.$$

Sampling this retained series at 201 display coordinates adds no modes. The final affine outputs are approximately -1.000 and $+0.928$. Latent function amplitude is not the target label: the decoder maps the complete coefficients to the prediction. Curves can share values at individual auxiliary coordinates while their full profiles remain distinct.

G.4 Fit and solver work

Figure 8 shows all three SiLU models' recorded learning curves. The PDE attains a low final held-out Huber loss in this run, but its early learning is slower and its later forward solver work is larger. The result therefore supports representation capacity, without implying faster optimization or lower computation than ANODE.

This experiment demonstrates a representational possibility of the three-coefficient ADR system. Its finite-dimensional evolution is itself an ODE and retains complete-state uniqueness. Zero errors on the sampled grid do not establish exact separation of every point in the continuous regions, and this single seed does not establish a general performance advantage over ANODE.

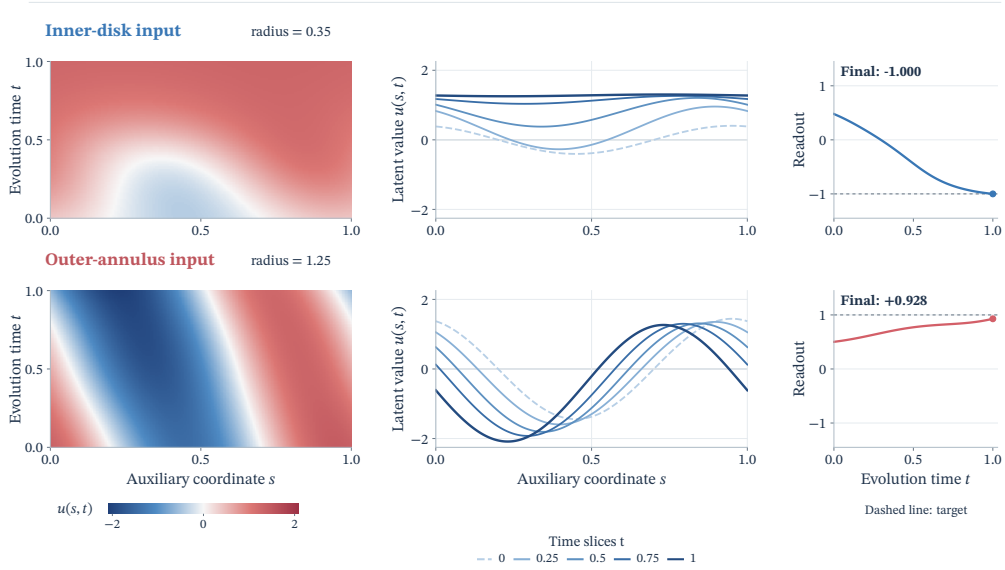


Figure 7: Latent-function evolution and decoding for two predefined inputs. Each row shows a function heatmap over auxiliary coordinate and evolution time, five function slices, and the affine decoder output along the evolution. The heatmaps and slices share an amplitude range; dashed horizontal output lines give the targets. The color of a function value does not denote its class. The function is reconstructed from three real Fourier coefficients throughout.

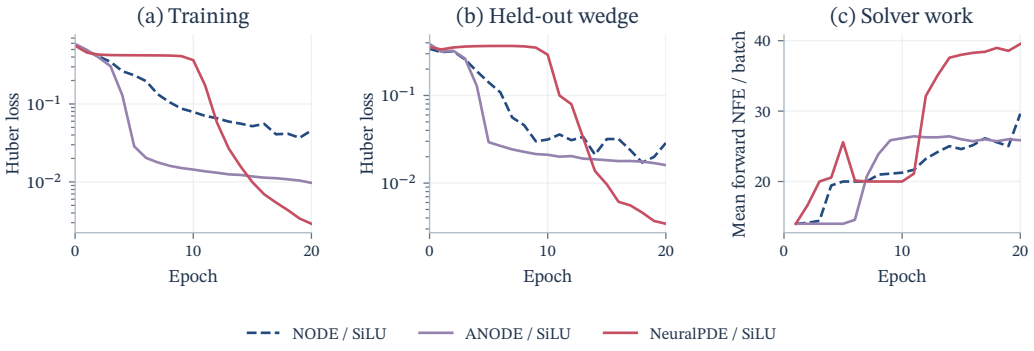


Figure 8: Training and held-out Huber loss, and forward solver work, for the three SiLU models at seed 17. Losses are evaluated after each epoch using the same checkpoint. NFE counts right-hand-side evaluations in training forward solves, averaged over batches; it excludes backward work and is not wall-clock time. There are no uncertainty bands from this single run.

H PROOFS AND NUMERICAL QUALIFICATIONS

This appendix analyzes the finite-mode, continuous-time ODE implemented by the NEURALPDE. It does not assert convergence to an infinite-dimensional PDE as $K \to \infty$, stability of a trained numerical rollout, or a statistical generalization guarantee. All parameters are fixed and finite during the analysis.

H.1 COORDINATE AND PARAMETER PROPERTIES

Proposition 3 (Coordinates, parameter budget, and dense RHS rank)**.** *The real Fourier codec is a bijection between $\mathbb{R}^q$ and V_K, where $q = C_z(2K+1)$. The two-hidden-layer dense RHS of width h has*

$$P_{\mathrm{dense}} = h^2 + 2qh + 2h + q, \qquad \operatorname{rank} J_{f_\phi}(u) \le \min(h, q). \tag{13}$$

For reaction width w, the ADR RHS has $P_{\mathrm{ADR}} = 2C_z^2 + w^2 + 2C_z w + 2w + C_z$ raw trainable parameters, independent of K. The rank bound concerns the instantaneous vector field; the exact finite-time flow of the smooth dense RHS has an invertible Jacobian.

With $C_z = K = 8$ and $w = 64$, this gives $q = 136$, $P_{\mathrm{ADR}} = 5{,}384$, and $P_{\mathrm{dense}} = 5{,}392$ for $h = 18$. Equal raw budgets do not imply equal ranks or arithmetic costs. The shared reaction MLP is evaluated at $n_p = \lceil 1.5(2K+1) \rceil = 26$ latent grid points: its linear layers require $26(w^2 + 2C_z w) = 133{,}120$ multiply–accumulates per sample per RHS call, versus $h^2 + 2qh = 5{,}220$ for the dense MLP. The comparison excludes biases, activations, FFTs, and spectral linear terms; it does not measure runtime.

H.1.1 REAL COORDINATES AND FFT NORMALIZATION

Set $N = 2K+1$. The implementation stores an orthonormal-rFFT spectrum $\widehat{z}_0 \in \mathbb{R}^{C_z}$ and $\widehat{z}_k \in \mathbb{C}^{C_z}$ for $1 \le k \le K$. Its associated trigonometric polynomial is

$$z(s) = \frac{1}{\sqrt{N}} \left[\widehat{z}_0 + \sum_{k=1}^{K} \left(\widehat{z}_k e^{2\pi \mathrm{i} k s} + \overline{\widehat{z}_k} e^{-2\pi \mathrm{i} k s} \right) \right]. \tag{14}$$

The factor $N^{-1/2}$ reconciles the stored `norm="ortho"` coefficients with the continuous Fourier amplitudes; it does not change any dimension or rank. The odd base grid has no Nyquist coefficient.

Proof of the coordinate assertion in Proposition 3. For each channel, one real DC coefficient and the real and imaginary parts of K positive coefficients give $1 + 2K$ real coordinates. Assigning the negative coefficients by conjugate symmetry yields a real polynomial in V_K. Conversely, Fourier orthogonality uniquely recovers all these coefficients from a function in V_K. These two operations are inverse linear maps, exactly the valid-spectrum restriction used by `FourierStateCodec`. The imaginary part of DC is fixed to zero, not an additional degree of freedom. Thus $\dim_{\mathbb{R}} V_K = C_z(2K+1)$. □

Parseval's identity applied to the finite sum gives

$$\|z\|_{L^2}^2 = \frac{1}{N} \left(\|\widehat{z}_0\|_2^2 + 2 \sum_{k=1}^{K} \|\widehat{z}_k\|_2^2 \right). \tag{15}$$

In the packed real coordinates u, this is $N^{-1} u^\top D u$, where D has weight one on DC entries and weight two on positive-mode real and imaginary entries. It is not the unweighted Euclidean norm of u; the two norms are equivalent, with $N^{-1}\|u\|_2^2 \le \|z\|_{L^2}^2 \le 2N^{-1}\|u\|_2^2$. Jacobian ranks are unaffected by this change of norm.

H.2 PARAMETER COUNTS, VECTOR-FIELD RANK, AND FLOW RANK

Proof of the remaining assertions in Proposition 3. The matrices in the dense network have shapes $h \times q$, $h \times h$, and $q \times h$. Their weights contribute $2qh + h^2$ parameters, while the three biases

contribute $2h + q$. With diagonal activation-derivative matrices $D_1(u)$ and $D_2(u)$, the chain rule gives

$$J_{f_\phi}(u) = W_3 D_2(u) W_2 D_1(u) W_1. \tag{16}$$

The rank of a matrix product cannot exceed any intermediate dimension, so its rank is at most $\min(h, q)$.

The ADR implementation stores two unrestricted $C_z \times C_z$ tensors: the raw advection tensor and the diffusion factor L. They contribute $2C_z^2$ parameters even though the resulting A and B are symmetric. The $C_z \to w \to w \to C_z$ reaction network contributes $2C_z w + w^2 + 2w + C_z$. Neither contribution depends on K. These are raw optimizer parameter counts, not counts of identifiable functions or distinct symmetric-matrix entries. In particular, the antisymmetric part of the raw advection tensor does not affect A, and different factors L can give the same $LL^\top$.

For completeness, SiLU is smooth with bounded derivative, as proved in Lemma 6 below. A finite-weight dense network is therefore globally Lipschitz and has a unique solution for every finite positive or negative time. Write its exact flow as Ψ_τ. Uniqueness and autonomy give $\Psi_{-\tau} \circ \Psi_\tau = \mathrm{id}$. Smooth dependence on initial conditions follows from the variational equation $\dot{J} = J_{f_\phi}(\Psi_\tau(u))J$, $J(0) = I$; differentiating the inverse identity then gives

$$D\Psi_{-\tau}(\Psi_\tau(u)) D\Psi_\tau(u) = I. \tag{17}$$

Thus $D\Psi_\tau(u)$ is invertible and has rank q. The instantaneous RHS rank bound is not a low-rank assertion about this flow, nor about a discrete RK4 map. For example, a zero vector field has a rank-zero Jacobian but its flow is the identity. □

The absence of K from P_{ADR} expresses parameter sharing across the auxiliary coordinate. It does not imply constant computation as K increases. Counting only linear-layer multiply–accumulates (MACs), the pointwise reaction costs $n_p(w^2 + 2C_z w)$ and the dense network costs $h^2 + 2qh$ per sample per RHS evaluation. Bias addition, activation evaluation, FFTs, construction and application of A/B, batching effects, and backward computation are excluded. The default numbers therefore count arithmetic operations; they are not measured FLOPs or wall-clock costs.

Table 14: Algebraic counts at $C_z = K = 8$, $q = 136$, and reaction width $w = 64$. The rank column concerns the dense RHS Jacobian or the ADR linear component, not the finite-time flow. The larger dense widths are capacity-analysis points, not additional experimental results.

Generator	Raw RHS parameters	Rank bound / linear rank
Dense width 18	5,392	≤ 18
Dense width 32	9,928	≤ 32
Dense width 64	21,768	≤ 64
Dense width 136	55,896	≤ 136
Dense width 256	135,816	≤ 136
ADR	5,384	128 (linear part)

If the optional nonlocal multiplier is enabled, the stored real and imaginary tensors add $2(K+1)C_z^2$ raw parameters. This includes the masked imaginary DC entries. The budget formula above and the automatic width-18 match describe the reported configuration with this option disabled. An arbitrary nonlocal multiplier also need not preserve the linear dissipation established next.

H.3 Invertibility and dissipation without simultaneous diagonalization

The energy identity separates the roles of the two derivative terms: transport contributes no change in total energy, and diffusion damps nonconstant variation. More specifically, $\|e^{\tau M_k}\|_2 \leq e^{-\omega_k^2 b\tau}$ for $k > 0$. Thus finer auxiliary frequencies have a stronger linear damping bound. The zero mode is preserved by the linear flow. These statements concern the operator supplied by the model, not a measured property of learned forecasts or a rank guarantee for the full nonlinear ADR Jacobian.

Proof of Proposition 1. For any $v \in \mathbb{C}^{C_z}$ and $k > 0$, symmetry of the real matrices gives

$$\mathrm{Re}(v^* M_k v) = -\omega_k^2 v^* B v \leq -\omega_k^2 b \|v\|_2^2. \tag{18}$$

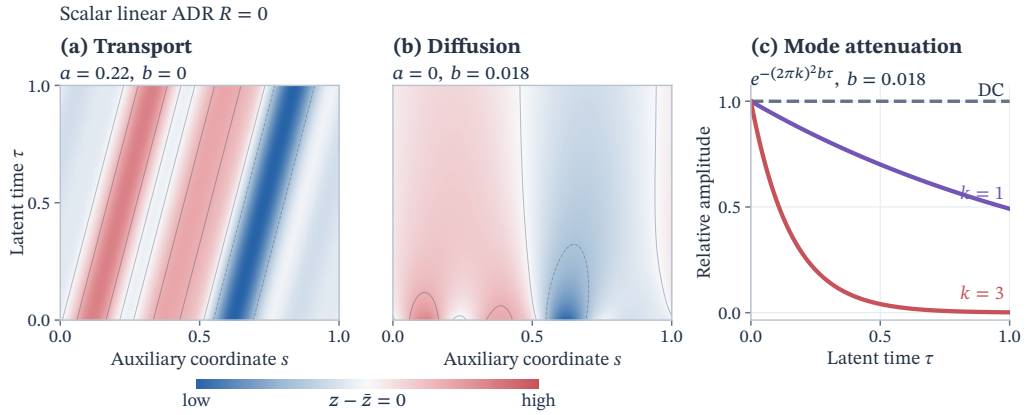


Figure 9: Analytic component illustrations with $R = 0$. The same periodic initial function evolves under (a) transport alone ($a = 0.22$, $b = 0$) and (b) diffusion alone ($a = 0$, $b = 0.018$). Color shows deviation from the conserved mean on a shared scale: transport shifts the profile, while diffusion smooths its fine variation. (c) Separate unit modal perturbations have amplitude $\exp[-(2\pi k)^2 b\tau]$ for $b = 0.018$; DC is preserved. These are analytic linear examples, not trained-model measurements.

Here $v^* A v$ is real because A is Hermitian, so its coefficient $-\mathrm{i}\omega_k$ contributes zero real part. If $M_k v = 0$, the inequality forces $v = 0$. The square matrix M_k is therefore invertible. For $y(\tau) = e^{\tau M_k} y(0)$, $\frac{\mathrm{d}}{\mathrm{d}\tau}\|y\|_2^2 \le -2\omega_k^2 b \|y\|_2^2$ by the same Hermitian-part identity. Integration gives $\|e^{\tau M_k}\|_2 \le e^{-\omega_k^2 b\tau}$.

In fact, Cauchy–Schwarz yields

$$\|M_k v\|_2 \|v\|_2 \ge |v^* M_k v| \ge -\operatorname{Re}(v^* M_k v) \ge \omega_k^2 b \|v\|_2^2, \tag{19}$$

so $\sigma_{\min}(M_k) \ge \omega_k^2 b$.

On real and imaginary coordinates, this block is represented by

$$\begin{pmatrix} -\omega_k^2 B & \omega_k A \\ -\omega_k A & -\omega_k^2 B \end{pmatrix}. \tag{20}$$

Complex invertibility is equivalent to invertibility of this real $2C_z \times 2C_z$ representation. Each of the K positive modes therefore contributes rank $2C_z$, whereas the DC block is zero. The rank of the whole block-diagonal linear generator is exactly $2KC_z$ and its kernel is exactly the C_z-dimensional space of constant functions.

For $z \in V_K$, all derivatives and integrations below involve finite trigonometric sums. Periodicity and constant symmetric A give

$$-\int_0^1 z^\top A \partial_s z \,\mathrm{d}s = -\frac{1}{2}\int_0^1 \partial_s (z^\top A z)\,\mathrm{d}s = 0. \tag{21}$$

Integration by parts similarly gives $\int_0^1 z^\top B \partial_{ss} z \,\mathrm{d}s = -\int_0^1 (\partial_s z)^\top B \partial_s z \,\mathrm{d}s$. Taking the L^2 inner product of $\dot{z} = \mathcal{A} z$ with z proves Equation 8.

The integral of each derivative term is zero, so $\bar{z}$ is conserved. For $w = z - \bar{z}$, finite-sum Fourier orthogonality gives $\|\partial_s w\|_{L^2}^2 \ge 4\pi^2 \|w\|_{L^2}^2$ because all its frequencies satisfy $|k| \ge 1$. Hence $\frac{\mathrm{d}}{\mathrm{d}\tau}\|w\|_{L^2}^2 \le -8\pi^2 b \|w\|_{L^2}^2$. Multiplication by $e^{8\pi^2 b\tau}$ and integration yield the stated decay bound after taking square roots. None of these steps diagonalizes A and B simultaneously or assumes $AB = BA$. □

Remark 4 (A limited derivative-approximation separation). Let V_K^0 be the zero-mean subspace, of real dimension $d = 2KC_z$. Transport the dense network to V_K through the linear codec. If $h < d$, then at every state z,

$$\left\| (\mathcal{A} - J_{f_\phi}(z))\big|_{V_K^0} \right\|_{L^2 \to L^2} \ge 4\pi^2 b. \tag{22}$$

Indeed, the restriction of $J_{f_\phi}(z)$ to V_K^0 has rank at most h, so it annihilates some unit vector $v \in V_K^0$. Equation 8 as a quadratic-form identity and the zero-mean inequality give $\|\mathcal{A}v\|_{L^2} \geq -\langle v, \mathcal{A}v\rangle_{L^2} \geq 4\pi^2 b$. Evaluating the difference on this v proves the claim. This separates derivatives of a specific linear generator from those of a bottleneck network. It does not separate their forecast errors after a learned encoder and decoder, prove an optimization benefit, or imply that the trained ADR model uses every available linear direction.

H.4 Nonlinear evolution and conditional sensitivity

The reaction can amplify as well as attenuate perturbations, so the full evolution requires an additional bound. We analyze the actual grid-sampled map used in the solver. If E_n evaluates a function at $n > 2K$ equally spaced points with grid inner product $n^{-1}\sum_j a_j^{\mathsf{T}} b_j$, then E_n preserves the L^2 inner product on V_K. The sampled reaction is $\mathcal{R}_{K,n} = E_n^* \mathbf{R}_\theta E_n$, where $\mathbf{R}_\theta$ applies the pointwise network to the grid. This formulation allows us to analyze the sensitivity of the implemented nonlinear map directly.

Proposition 5 (Conditional perturbation bound for the implemented reaction)**.** *Let $\mathcal{R}_{K,n_p}$ denote the actual padded-grid reaction followed by Fourier truncation, with $n_p > 2K$. Suppose the pointwise network satisfies $(a-b')^\top(R_\theta(a) - R_\theta(b')) \leq \ell\|a-b'\|_2^2$ for all $a, b' \in \mathbb{R}^{C_z}$. For two solutions of $\dot{z} = \mathcal{A}z + \mathcal{R}_{K,n_p}(z)$,*

$$\|z(\tau) - v(\tau)\|_{L^2} \leq e^{\ell\tau}\|z(0) - v(0)\|_{L^2}, \qquad \tau \geq 0. \tag{23}$$

H.5 The actual sampled reaction and a conditional nonlinear bound

Let $n = n_p > 2K$ and let $E_n : V_K \to (\mathbb{R}^{C_z})^n$ evaluate a function at $s_j = j/n$, $0 \leq j < n$. Give grid vectors the inner product $\langle a, b\rangle_n = n^{-1}\sum_{j=0}^{n-1} a_j^\top b_j$. For an integer r with $|r| \leq 2K < n$, the geometric sum $n^{-1}\sum_j e^{2\pi \mathrm{i} rj/n}$ equals one when $r = 0$ and zero otherwise. Applying this identity to products of trigonometric polynomials proves

$$\langle E_n z, E_n v\rangle_n = \langle z, v\rangle_{L^2}, \qquad E_n^* E_n = I. \tag{24}$$

Thus E_n is an isometry and its adjoint E_n^* has norm one. Define the pointwise grid map by $(\mathbf{R}_\theta(a))_j = R_\theta(a_j)$. The implemented reaction is exactly

$$\mathcal{R}_{K,n} = E_n^* \mathbf{R}_\theta E_n. \tag{25}$$

To verify the normalization, zero-padding the stored spectrum and multiplying by $\sqrt{n/N}$ makes its orthonormal inverse FFT equal Equation 14 evaluated at the n grid points. The orthonormal FFT of the grid reaction, followed by multiplication by $\sqrt{N/n}$ and truncation, stores $\sqrt{N}$ times its discrete Fourier averages. Substituting into Equation 14 gives exactly the adjoint map E_n^*. Because $n > 2K$, none of the retained positive frequencies is a padded-grid Nyquist mode.

Proof of Proposition 5. Set $w = z - v$. By the adjoint relation and the assumed one-sided bound,

$$\begin{aligned}\langle w, \mathcal{R}_{K,n}(z) - \mathcal{R}_{K,n}(v)\rangle_{L^2} &= \frac{1}{n}\sum_{j=0}^{n-1} w(s_j)^\top \big(R_\theta(z(s_j)) - R_\theta(v(s_j))\big) \\ &\leq \frac{\ell}{n}\sum_{j=0}^{n-1}\|w(s_j)\|_2^2 = \ell\|w\|_{L^2}^2. \end{aligned} \tag{26}$$

The linear energy identity therefore gives the stronger differential bound

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}\tau}\|w\|_{L^2}^2 \leq -b\|\partial_s w\|_{L^2}^2 + \ell\|w\|_{L^2}^2 \leq \ell\|w\|_{L^2}^2. \tag{27}$$

Multiplying by $e^{-2\ell\tau}$ and integrating proves Equation 23. If, additionally, $\int_0^1 w(\tau, s)\,\mathrm{d}s = 0$ throughout the interval, the zero-mean inequality improves the exponent to $\ell - 4\pi^2 b$. Equality of the two initial means alone does not ensure this additional condition: a general nonlinear reaction can immediately create different means. □

Lemma 6 (A finite Lipschitz constant for the SiLU reaction)**.** *For $\sigma(t) = t/(1 + e^{-t})$, one has $\sup_t |\sigma'(t)| \le 1 + e^{-1}$. If U_1, U_2, U_3 are the three reaction weight matrices, then*

$$L_R = (1 + e^{-1})^2 \|U_3\|_2 \|U_2\|_2 \|U_1\|_2 \tag{28}$$

is a global Lipschitz constant for R_θ and $\mathcal{R}_{K,n}$. In particular, the finite-mode ADR ODE has a unique solution for every finite positive or negative time, and Proposition 5 always holds with $\ell = L_R$.

Proof. Writing $s(t) = (1 + e^{-t})^{-1}$ gives $\sigma'(t) = s(t) + ts(t)(1 - s(t))$ and $s(t)(1 - s(t)) \le e^{-|t|}$. Hence $|\sigma'(t)| \le 1 + |t|e^{-|t|} \le 1 + e^{-1}$. The componentwise activation has the same Euclidean Lipschitz bound. Composition with the three affine layers yields Equation 28; biases cancel in differences. Applying this bound pointwise on the grid and using $\|E_n\| = \|E_n^*\| = 1$ proves the same constant for $\mathcal{R}_{K,n}$. Cauchy–Schwarz then gives the one-sided bound with $\ell = L_R$.

At fixed K, $\mathcal{A}$ is a bounded finite-dimensional linear map. Consequently the total vector field is globally Lipschitz. Local existence and uniqueness can be iterated without finite-time blow-up: if its Lipschitz constant is L_f, then $\|f(z)\| \le L_f\|z\| + \|f(0)\|$, and integration of this inequality bounds $\|z(\tau)\|$ on every finite time interval, in either time direction. This establishes the global finite-time existence assertion. It is not a claim that trajectories remain uniformly bounded as $\tau \to \infty$. □

Remark 7 (What is and is not assumed about aliasing). The identity $E_n^* E_n = I$ is exact for the retained finite-mode functions. It does not assert that $E_n^* \mathbf{R}_\theta E_n z$ equals the continuous L^2 projection of $R_\theta(z(s))$. A nonpolynomial SiLU composition can contain arbitrarily high harmonics, so $3/2$ padding provides oversampling but no general exact-dealiasing guarantee for this reaction. The proof above applies directly to the sampled nonlinear map and therefore does not require such a guarantee. A separate discretization-error or convergence analysis would have to control the unresolved nonlinear harmonics.

H.6 Decoder sensitivity and the residual anchor

Proof of Corollary 2. The decoder Lipschitz inequality and Proposition 5 give

$$\|\mathcal{D}(\mathcal{S}_\tau z_0) - \mathcal{D}(\mathcal{S}_\tau v_0)\|_2 \le L_D \|\mathcal{S}_\tau z_0 - \mathcal{S}_\tau v_0\|_{L^2} \le L_D e^{\ell\tau} \|z_0 - v_0\|_{L^2}.$$

Squaring proves Equation 9. Subtracting the two residual forecasts and applying the triangle inequality gives Equation 10. The anchor contribution vanishes when its value is held fixed. □

To express L_D in the function norm, set $N = 2K + 1$ and let u be the packed real coordinates of z. Write D_F for the diagonal matrix with weight one on DC entries and weight two on positive-mode real and imaginary entries. Equation 15 gives $\|z\|_{L^2} = N^{-1/2} \|D_F^{1/2} u\|_2$. The spectral feature map is a real linear map $g = Gu$: the DC columns of G contain $\operatorname{Re} W_{rc0}$, and positive-mode columns contain the real and imaginary parts of W_{rck} in codec order. The imaginary DC parameters have no effect. Hence

$$\|G(u - v)\|_2 \le \sqrt{N} \|G D_F^{-1/2}\|_2 \|z_u - z_v\|_{L^2}.$$

If the decoder head is $V_2\sigma(V_1 g + b_1) + b_2$, the SiLU derivative bound in Lemma 6 shows that

$$L_D = \sqrt{N}(1 + e^{-1}) \|V_2\|_2 \|V_1\|_2 \|G D_F^{-1/2}\|_2 \tag{29}$$

is a valid global Lipschitz constant. The decoder head has one SiLU activation, and its biases cancel in differences. The constant is finite for fixed finite weights; no uniform bound over fitted models or truncation levels is asserted.

For completeness, let $z_\star \in V_K$ and $a_\star \in \mathbb{R}^C$ be specified reference values, and let y_τ be a target. Denote the function-valued numerical flow by Φ_τ^h and define

$$\varepsilon_{\mathrm{num}}(\tau, z_X) = \|\Phi_\tau^h z_X - \mathcal{S}_\tau z_X\|_{L^2}, \qquad r_\tau = y_\tau - a_\star - \mathcal{D}(\mathcal{S}_\tau z_\star).$$

Inserting the two continuous-flow decoder outputs and applying the triangle inequality yields

$$\begin{aligned} &\|a_X + \mathcal{D}(\Phi_\tau^h z_X) - y_\tau\|_2 \\ &\quad \le \|a_X - a_\star\|_2 + L_D \varepsilon_{\mathrm{num}}(\tau, z_X) + L_D e^{\ell\tau} \|z_X - z_\star\|_{L^2} + \|r_\tau\|_2. \end{aligned} \tag{30}$$

The residual r_τ contains discrepancies between the reference learned forecast and the target; the decomposition does not estimate this residual. In particular, the initial-state term alone does not control total prediction error or statistical risk.

H.7 What RK4 does and does not inherit

At fixed K, the implementation evolves $q = C_z(2K+1)$ real coordinates. Parameter sharing in the PDE generator is a structural choice within this finite-dimensional system. Proposition 3 in Appendix H distinguishes raw parameter budgets, instantaneous RHS ranks, and arithmetic costs.

The `torchdiffeq` fixed-grid `RK4` implementation calls `rk4_alt_step_func`, whose stages are the four-stage $3/8$ rule with weights $(1,3,3,1)/8$ and stage times $(0,1/3,2/3,1)$. These stages are specified in the package's official fixed-grid implementation and Runge–Kutta step implementation. It has the same linear stability polynomial as classical four-stage RK4, but its nonlinear stages differ. The following statement concerns a full internal solver step on the linear problem.

Lemma 8 (Linear RK4 energy criterion)**.** *For $\dot{y} = My$, one $3/8$-rule step of length $\Delta\tau$ sends y to $p_4(\Delta\tau M)y$, with*

$$p_4(\zeta) = 1 + \zeta + \frac{\zeta^2}{2} + \frac{\zeta^3}{6} + \frac{\zeta^4}{24}. \tag{31}$$

For the linear ADR problem, the step is nonexpansive in the function L^2 norm for every initial state if and only if

$$\max_{1\le k\le K} \|p_4(\Delta\tau M_k)\|_2 \le 1. \tag{32}$$

For $A = 0$ and $B = bI$, $b > 0$, this is equivalent to $\Delta\tau(2\pi K)^2 b \le r_4$, where $r_4 = 2.785293563\ldots$ is the unique positive root of $r^3 - 4r^2 + 12r - 24 = 0$.

Proof. Let $H = \Delta\tau M$ and write s_i for the step-size-scaled stages. The $3/8$ rule gives

$$\begin{aligned} s_1 &= Hy, \\ s_2 &= H(y + s_1/3) = (H + H^2/3)y, \\ s_3 &= H(y - s_1/3 + s_2) = (H + 2H^2/3 + H^3/3)y, \\ s_4 &= H(y + s_1 - s_2 + s_3) = (H + H^2 + H^3/3 + H^4/3)y. \end{aligned}$$

Substitution into $y_+ = y + (s_1 + 3s_2 + 3s_3 + s_4)/8$ yields Equation 31. Each Fourier block evolves independently. The DC amplification is the identity, and each positive block has a single scalar Parseval weight $2/N$. Therefore the induced function norm of the whole step is $\max(1, \max_{k>0} \|p_4(\Delta\tau M_k)\|_2)$. The definition of an induced operator norm proves the equivalence in Equation 32.

In the scalar-diffusion case, set $x = \Delta\tau\omega_k^2 b \ge 0$. The polynomial is strictly positive on this axis, since

$$24p_4(-x) = (x^2 - 2x)^2 + 8(x - 3/2)^2 + 6 > 0. \tag{33}$$

Also $24(p_4(-x) - 1) = x(x^3 - 4x^2 + 12x - 24)$. The cubic factor is strictly increasing because its derivative $3x^2 - 8x + 12$ is positive for all real x; its value at zero is negative and it tends to infinity. Thus $|p_4(-x)| \le 1$ exactly for $0 \le x \le r_4$. The largest x occurs at $k = K$, giving the stated threshold. In particular, $x = 3$ gives $p_4(-3) = 11/8 > 1$, even though the corresponding exact continuous solution decays by e^{-3} over the step. □

For noncommuting A and B, the blocks M_k can be nonnormal. The spectral-radius test $\max_j |p_4(\Delta\tau\lambda_j(M_k))| \le 1$ is necessary for Equation 32, but in general is not sufficient: a matrix can have small eigenvalues while its largest singular value exceeds one. The matrix-norm test avoids this gap without imposing simultaneous diagonalization.

The nonlinear reaction is not included in this linear criterion. Its Jacobian is evaluated at different internal stage states, so replacing M by a single instantaneous full-RHS Jacobian is not generally the exact nonlinear RK4 step derivative. Query times also need not coincide with internal step endpoints; output interpolation is a separate operation. These distinctions call for checking the actual one-step perturbation map, latent norms, and predictions under step refinement and against a converged adaptive reference. Neither the continuous energy identity nor the algebraic RK4 polynomial is a substitute for those experiments.