# DS4D-FFNO: HIDDEN-STATE RETRIEVAL FOR PDES WITH FIXED-DELAY BOUNDARY FEEDBACK

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Partial differential equations (PDEs) with fixed-delay boundary feedback arise in fields such as wave propagation, structural dynamics, electromagnetics, and flow control, where boundary feedback depends on past states. In recent years, neural operators have been widely used to solve PDEs, but most lack a dedicated mechanism for modeling fixed-delay boundary feedback. We propose DS4D-FFNO, combining factorized Fourier neural operator (FFNO) spatial mixing with DS4D, an S4D extension that models delayed dynamics by retrieving interpolated past hidden states via a delayed-feedback branch. DS4D-FFNO consists of an encoder followed by two FFNO layers, one DS4D layer, two further FFNO layers, and an output head. Compared with seven other models, DS4D-FFNO achieves the lowest error in solving each of six such PDEs, which describe wave propagation, structural vibrations, thermoelastic coupling, boundary dynamics, and reaction–diffusion processes. To assess its applicability beyond delayed systems, we compare DS4D-FFNO with S4-FFNO on six classical PDEs without physical delay and find comparable predictive accuracy. To test whether its learned delay indices contribute to the gains on delayed PDEs, we permute or randomly replace them, and both interventions increase error, indicating that learned delayed access is essential for maintaining the trained model's predictive accuracy.

## 1 INTRODUCTION

Partial differential equations (PDEs) describe how physical fields interact and evolve in fluid flows, wave propagation, and structural dynamics (Li et al., 2021, Nicaise & Pignotti, 2006, Saba et al., 2023). The need to solve PDEs repeatedly under varying initial conditions and physical parameters motivates the use of neural operators, which learn reusable mappings from problem inputs to solution fields (Kovachki et al., 2023, Lu et al., 2021, Li et al., 2021).

For time-dependent PDEs, fixed-delay boundary feedback introduces an additional difficulty beyond predicting evolution from the current state. The boundary condition at time $t$ explicitly contains a quantity evaluated at $t - \tau$. Observing the complete field at $t$ does not, in general, determine that earlier boundary quantity. The initial state must therefore include a history segment, and a predictor must retain the past information needed to evaluate the subsequent boundary feedback.

To address time-dependent physical prediction, researchers have explored iterative refinement and architectures with temporal memory (Lippe et al., 2023, Buitrago Ruiz et al., 2025, Yu et al., 2026). PDE-Refiner improves forecasting accuracy through repeated refinement of predicted fields (Lippe et al., 2023). State-space models (SSMs), including S4 and S4D, provide efficient recurrent representations of temporal dependence (Gu et al., 2022a, b). Building on this idea, Memory Neural Operator (MemNO) combines an SSM with Fourier layers to improve PDE prediction under coarse or noisy observations (Buitrago Ruiz et al., 2025). However, standard SSM updates summarize past information in the current hidden state without explicitly retaining earlier hidden states for retrieval at selected delays. How to preserve and explicitly access relevant historical representations for these PDEs remains an open modeling question.

Our approach is to give the predictor explicit access to its hidden-state history. Delay State Space Models (DSSMs) extend diagonal SSM recurrences with a branch that reads stored states at learnable delays (Qian et al., 2026). This provides a way to combine the current recurrent state with selected

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.

past representations. We combine this temporal mechanism with spatial mixing to represent delayed boundary effects across the field.

We propose DS4D-FFNO, which combines factorized Fourier neural operator (FFNO) layers with DS4D, the delayed S4D module of the DSSM framework. The network has five stages: an encoder, a first pair of FFNO layers, DS4D, a second pair of FFNO layers, and an output head. DS4D receives spatially mixed features from the first FFNO pair and stores its hidden-state history for delayed retrieval. The second FFNO pair mixes the updated features before the output head predicts the next field. The same network processes the observed history and then advances its predictions autoregressively.

Our main contributions are:

- We propose DS4D-FFNO, a neural operator designed to predict PDE dynamics with fixed-delay boundary feedback. By combining FFNO spatial mixing with retrieval of past hidden states at learnable delays, it explicitly models delayed dependencies in field evolution.
- We compare DS4D-FFNO with seven baselines, including S4-FFNO, MambaFormer, and Transolver, across six one- and two-dimensional PDE benchmarks with fixed-delay boundary feedback. Under shared data splits and evaluation protocols, DS4D-FFNO achieves the lowest mean test full-field nRMSE on all six systems. On six separate PDEs without physical delay, DS4D-FFNO and S4-FFNO achieve similar mean test full-field errors.
- We conduct ablation experiments to assess the contribution of learned delayed hidden-state access. Specifically, permuting delays across channels or randomly replacing their values in trained models increases mean prediction error on all six delayed benchmarks.

## 2 BACKGROUND

**Fixed-delay boundary-feedback PDEs.** Wave propagation (Nicaise & Pignotti, 2006, 2016), structural dynamics (Zhang & Stépán, 2016, Cheng et al., 2024), electromagnetics (Anikushyn & Pokojovy, 2019, Yao et al., 2023), and controlled gas flow (Gugat & Dick, 2011) can involve sensing, communication, actuation, or transport delays that cause boundary feedback to depend on a past system state. We model this delayed dependence with fixed-delay boundary-feedback PDEs. To express these systems in a common form, let $T > 0$ be the final time, $\Omega \subset \mathbb{R}^D$ a $D$-dimensional spatial domain, $\Gamma \subseteq \partial\Omega$ the boundary where feedback is applied, and $z(t, x)$ the field value at time $t \in (0, T]$ and position $x \in \Omega$. We write

$$\begin{aligned} \mathcal{E}_\Omega[z; a](t, x) &= 0, && (t, x) \in (0, T] \times \Omega, \\ z(0, x) &= z_0(x), && x \in \Omega, \\ \mathcal{B}_\Gamma[z](t) &= F_\Gamma((\mathcal{C}_0 z)(t), (\mathcal{C}_\tau z)(t - \tau)), && t \in (0, T], \\ (\mathcal{C}_\tau z)(s) &= \phi_\Gamma(s), && s \in (-\tau, 0]. \end{aligned} \tag{1}$$

Here $z_0$ is the initial field, $\tau > 0$ is the fixed delay, and $a$ contains the physical parameters. The operators $\mathcal{E}_\Omega$ and $\mathcal{B}_\Gamma$ define the interior PDE and boundary condition. The operators $\mathcal{C}_0$ and $\mathcal{C}_\tau$ extract the current and delayed boundary quantities, respectively. These quantities determine the feedback $F_\Gamma$, through which past boundary information influences field evolution (Figure 1a). The function $\phi_\Gamma$ gives the delayed boundary history for $s \in (-\tau, 0]$.

**State-space models and delayed hidden-state retrieval.** At discrete time step $n$, an S4D channel with state size $N$ maps a scalar input $u_n$ to a scalar output $y_n$ through a hidden state $\boldsymbol{h}_n \in \mathbb{C}^N$:

$$\begin{aligned} \boldsymbol{h}_n &= \boldsymbol{A}\boldsymbol{h}_{n-1} + \boldsymbol{B}u_n, \\ y_n &= \boldsymbol{C}\boldsymbol{h}_n. \end{aligned} \tag{2}$$

Here $\boldsymbol{A} \in \mathbb{C}^{N \times N}$ is the diagonal state matrix, $\boldsymbol{B} \in \mathbb{C}^{N \times 1}$ is the input vector, and $\boldsymbol{C} \in \mathbb{C}^{1 \times N}$ is the readout vector (Gu et al., 2022a, b). DS4D adds a branch that retrieves stored hidden states at learnable delays (Qian et al., 2026). Its recurrence is given in Equation 9.

**Architectural choices in neural PDE solvers.** Neural PDE solvers have advanced along two largely orthogonal directions: spatial representation and prediction strategy. For spatial representation, DeepONet separates input-function encoding from coordinate queries (Lu et al., 2021). Fourier

neural operators (FNOs) and FFNO use global spectral mixing, with FFNO factorizing it by spatial axis (Li et al., 2021, Tran et al., 2023). Message-passing solvers operate directly on meshes, while Transolver groups spatial tokens before applying Physics-Attention (Brandstetter et al., 2022, Wu et al., 2024). For prediction strategy, DPOT and Poseidon use cross-PDE pretraining to improve generalization (Hao et al., 2024, Herde et al., 2024). We adopt FFNO as the spatial backbone because its axis-wise factorization preserves nonlocal mixing while reducing multidimensional spectral cost.

## 3 METHOD

**Prediction task.** Our goal is to learn a neural operator that predicts future fields from observed fields. Let $z(t, x) \in \mathbb{R}^P$ denote the continuous $P$-dimensional field at position $x \in \Omega \subset \mathbb{R}^D$ and time $t \in [0, T]$. We sample time at intervals of $\Delta T > 0$, giving $t_n = n \cdot \Delta T$ for $n = 0, \ldots, \lfloor T/\Delta T \rfloor$.

We use a uniform Cartesian grid $\{x_g\}_{g=1}^M$. For a rectangular $\Omega$, the grid covers the domain and its boundary. For a non-rectangular $\Omega$, such as the disk, we sample its bounding box, retain field values inside the domain and on its boundary, and set exterior values to zero. Thus $M$ counts the full grid used for Fourier transforms. A domain mask restricts the loss and evaluation metrics to valid points. The sampled field and coordinate matrices are

$$\boldsymbol{Z}_n = \begin{bmatrix} z(t_n, x_1)^\top \\ \vdots \\ z(t_n, x_M)^\top \end{bmatrix} \in \mathbb{R}^{M \times P}, \qquad \boldsymbol{X} = \begin{bmatrix} x_1^\top \\ \vdots \\ x_M^\top \end{bmatrix} \in \mathbb{R}^{M \times D}. \tag{3}$$

For a physical delay $\tau = m\Delta T$, $\boldsymbol{Z}_{n-m}$ denotes the sampled field $m$ time steps earlier.

Given $L$ observed fields, the neural operator $\mathcal{G}$ predicts the next field:

$$\mathcal{G} : (\boldsymbol{Z}_{n-L+1}, \ldots, \boldsymbol{Z}_n) \longmapsto \widehat{\boldsymbol{Z}}_{n+1}. \tag{4}$$

Here $\widehat{\boldsymbol{Z}}_{n+1}$ is the model's predicted field on the same grid. An $H$-step forecast targets $\boldsymbol{Z}_{n+1}, \ldots, \boldsymbol{Z}_{n+H}$, with $n \geq L - 1$ and $(n + H)\Delta T \leq T$.

At forecast step $n + h$, the boundary feedback depends on the boundary quantity at step $n + h - m$. Across the $H$-step forecast, this dependence spans $[n+1-m, n+H-m]$ (Figure 1b). This delayed dependence motivates retaining past information alongside the current spatial field.

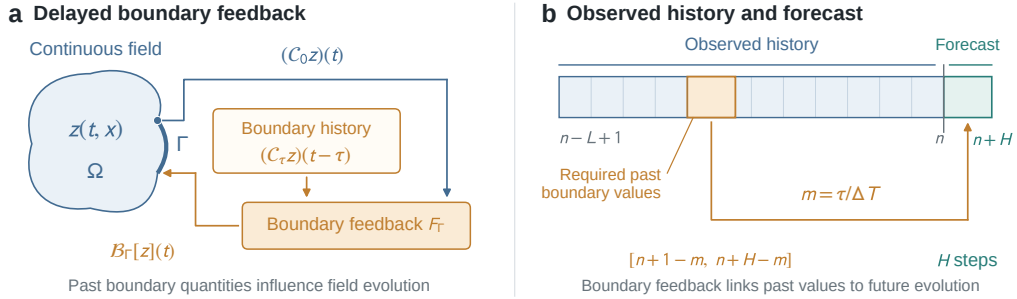


Figure 1: Delayed boundary feedback and its history dependence. **a**, Current and delayed boundary quantities jointly determine $F_\Gamma$, which acts through $\Gamma$ on the continuous field $z(t, x)$. **b**, The $L$ observed fields and $H$ forecast steps are shown together with the past boundary interval $[n + 1 - m, n + H - m]$ required by the PDE, where $m = \tau/\Delta T$.

DS4D-FFNO combines spatial mixing with a temporal module that retains and retrieves hidden-state history. It implements $\mathcal{G}$ through five stages: an encoder, a first pair of FFNO layers, DS4D, a second pair of FFNO layers, and an output head (Figure 2a). We write $\boldsymbol{v}_j^{(s)} \in \mathbb{R}^{M \times C}$ for an intermediate feature matrix at time step $j$, where $C$ is the hidden width. The feature index $s = 0, \ldots, 5$ follows individual layer outputs. Index 0 follows the encoder, indices 1–2 and 4–5 follow the four FFNO layers, and index 3 follows DS4D. All modules share parameters across time. The encoder, FFNO layers, and output head act frame-wise, while DS4D carries hidden state and history between steps (Figure 2b).

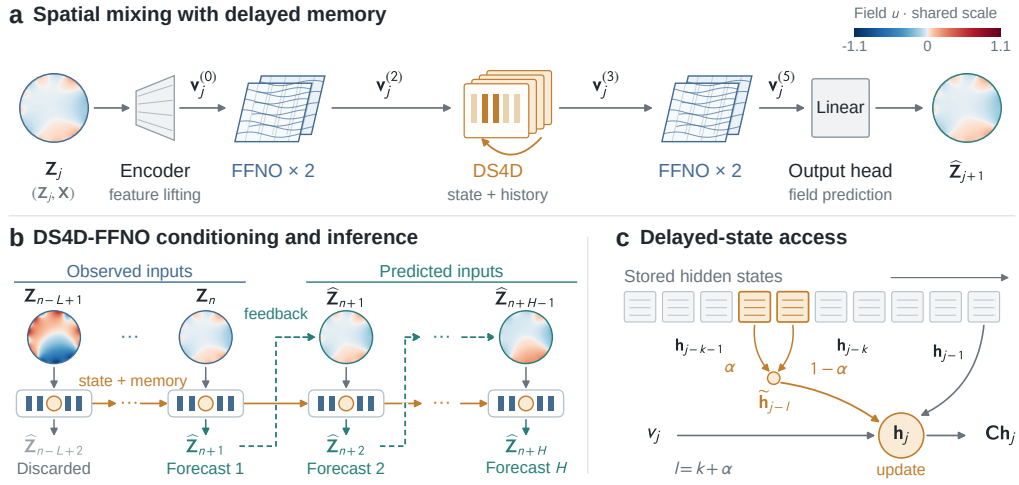


Figure 2: DS4D-FFNO architecture and inference. **a**, FFNO layers mix encoded field and coordinate features before and after DS4D, and the output head predicts the next field. **b**, DS4D-FFNO processes the $L$ observed fields sequentially while carrying its hidden state and hidden-state memory. Predictions before the forecast origin are discarded. After $\boldsymbol{Z}_n$, each predicted field is fed back as the next input until all $H$ forecast fields are generated. **c**, DS4D interpolates neighboring stored states, updates the hidden state, and forms the state readout. Field thumbnails show a Delayed linear problem test trajectory and its model predictions.

**Encoder.** The encoder concatenates the sampled field with grid coordinates and produces the initial hidden features:

$$\boldsymbol{v}_j^{(0)} = \mathrm{Linear}(\mathrm{Concat}(\boldsymbol{Z}_j, \boldsymbol{X})) \,. \tag{5}$$

**FFNO layers.** FFNO layers combine factorized spectral mixing and identity residual connections (Tran et al., 2023, Buitrago Ruiz et al., 2025):

$$\mathcal{S}(\boldsymbol{v}) = \boldsymbol{v} + \mathrm{MLP}\left(\sum_{d=1}^{D} \mathcal{F}_d^{-1}[\mathcal{R}_d \mathcal{F}_d(\boldsymbol{v})]\right). \tag{6}$$

Here $\mathcal{F}_d$ and $\mathcal{F}_d^{-1}$ are Fourier transforms along spatial axis $d$. The learned operator $\mathcal{R}_d$ mixes channels independently at each retained Fourier mode. The MLP contains two linear layers separated by GELU. We denote the four instances of this residual layer by $\mathcal{S}_1, \ldots, \mathcal{S}_4$, each with independent parameters, and obtain

$$\boldsymbol{v}_j^{(2)} = (\mathcal{S}_2 \circ \mathcal{S}_1)(\boldsymbol{v}_j^{(0)}), \qquad \boldsymbol{v}_j^{(5)} = (\mathcal{S}_4 \circ \mathcal{S}_3)(\boldsymbol{v}_j^{(3)}). \tag{7}$$

**DS4D.** Consider one grid point $g$ and feature channel $c$, with scalar input $v_j = [\boldsymbol{v}_j^{(2)}]_{g,c}$ and hidden state $\boldsymbol{h}_j$. Each channel's recurrence parameters and learnable continuous-time delay $\tau_c$ are shared across all grid points. In units of the observation interval $\Delta T$, this delay is $l = \tau_c / \Delta T \in \mathbb{R}$. Its integer and fractional parts are $k = \lfloor l \rfloor$ and $\alpha = l - k \in [0, 1)$, respectively. Linear interpolation gives the delayed hidden state (Figure 2c):

$$\widetilde{\boldsymbol{h}}_{j-l} = (1-\alpha)\boldsymbol{h}_{j-k} + \alpha \boldsymbol{h}_{j-k-1}. \tag{8}$$

The recurrent update is

$$\boldsymbol{h}_j = \boldsymbol{A}\boldsymbol{h}_{j-1} + \boldsymbol{A}^d \widetilde{\boldsymbol{h}}_{j-l} + \boldsymbol{B} v_j. \tag{9}$$

Here $\boldsymbol{A}^d \in \mathbb{C}^{N \times N}$ weights the delayed hidden state. A separate state history is stored at each grid point. Collecting the real-valued channel readouts $\boldsymbol{C}\boldsymbol{h}_j$ at all $M$ grid points gives $\boldsymbol{Y}_j \in \mathbb{R}^{M \times C}$. The DS4D layer then applies the identity residual update $\boldsymbol{v}_j^{(3)} = \boldsymbol{v}_j^{(2)} + \boldsymbol{Y}_j$. Appendix B specifies the implementation, memory length, and initialization.

**Output head.** A linear output head maps the final hidden representation to the model's predicted field:

$$\widehat{\boldsymbol{Z}}_{j+1} = \mathrm{Linear}(\boldsymbol{v}_j^{(5)}) \in \mathbb{R}^{M \times P}. \tag{10}$$

# 4 EXPERIMENTS

We first compare eight models on six PDEs with fixed-delay boundary feedback, then compare DS4D-FFNO with S4-FFNO on six separate PDEs without physical delay. Finally, we assess sensitivity to learned delay values and channel assignments on the delayed benchmarks. Appendix D.3 reports spatial resolution and parameter sensitivity analyses.

**Delayed benchmarks.** We test KdV–KdV system and Microbeam system in one dimension, and Thermoelastic system and Delayed linear problem in two dimensions. We also test the Kawahara equation and Reaction–diffusion model, whose governing equations are given in Appendix A. The appendix also provides data-generation settings for these six delayed benchmarks.

**Baselines.** On the delayed benchmarks, we compare DS4D-FFNO with S4-FFNO, FNO, FFNO, MambaFormer (Park et al., 2024), Transolver, MambaFormer–FFNO, and Transolver–FFNO. FNO and FFNO follow MemNO (Buitrago Ruiz et al., 2025) in using four frame-wise spatial blocks, and Transolver variants apply Physics-Attention frame-wise. S4-FFNO and MambaFormer–FFNO share the FFNO layout with S4D and MambaFormer temporal cores, respectively.

Within each benchmark, all models share data splits, prediction targets, training budgets, and validation-based checkpoint selection. For each equation and seed, DS4D-FFNO and S4-FFNO share initial tensors for the encoder, output head, FFNO blocks, and S4D core. These include the time scales and S4D–Lin spectrum. Table 5 reports widths, applicable SSM state sizes, and exact parameter counts.

**Evaluation.** Models train with teacher-forced one-step supervision and are evaluated autoregressively. Full-field nRMSE is the primary metric, and we also report controlled-boundary-region nRMSE for delayed benchmarks. Tables report means and sample standard deviations over model seeds 0–2. Appendix C defines the metrics, and Table 4 specifies boundary masks. Appendices A and E give training settings.

## 4.1 ONE-DIMENSIONAL EXPERIMENTS

**Setup.** We test dispersive waves and structural vibrations on $0 < x < 1$, using $M = 65$ grid points, $L = 192$ observations, and $H = 24$ forecast steps. The prescribed physical delays, $\tau = 0.128$ for KdV–KdV system and $\tau = 1.0$ for the Microbeam system, both correspond to $m = \tau/\Delta T = 128$ sampled steps. DS4D learns its hidden-state delays. All models train for 120 epochs with batch size 4. Appendix A gives coefficients, initial histories, and solvers.

**KdV–KdV system.** This nonlinear system couples surface elevation $\eta$ and horizontal velocity $\omega$, with delayed boundary-gradient feedback (Capistrano-Filho et al., 2024):

$$\begin{aligned}
&\eta_t + \omega_x + \omega_{xxx} + (\eta\omega)_x = 0,\\
&\omega_t + \eta_x + \eta_{xxx} + \omega\omega_x = 0,\\
&\eta(t,0) = \eta_x(t,0) = \eta(t,1) = 0,\\
&\omega(t,0) = \omega(t,1) = 0,\\
&\omega_x(t,1) = -\alpha\eta_x(t,1) + \beta\eta_x(t-\tau,1).
\end{aligned} \tag{11}$$

The coefficients $\alpha, \beta$ weight the current and delayed boundary gradients, respectively. Initial data prescribe $\eta(0,x)$, $\omega(0,x)$, and the history of $\eta_x(s,1)$ for $-\tau < s \le 0$. The prediction target is the two-channel field $(\eta, \omega)$.

**Microbeam system.** We use the fixed-delay form of the sixth-order microbeam model for displacement $u$ with delayed tip-velocity feedback (Saba et al., 2023):

$$\begin{aligned}
&\rho A u_{tt} + M_1 u_{xxxx} - M_2 u_{xxxxxx} = 0,\\
&u(t,0) = u_x(t,0) = u_{xx}(t,0) = 0,\\
&M_1 u_{xxx}(t,1) - M_2 u_{xxxxx}(t,1) = \beta_1 u_t(t,1) + \beta_2 u_t(t-\tau,1),\\
&M_1 u_{xx}(t,1) - M_2 u_{xxxx}(t,1) = 0,\\
&M_2 u_{xxx}(t,1) = 0.
\end{aligned} \tag{12}$$

Here $\rho A$ is mass per unit length, $M_1, M_2$ are physical bending stiffnesses, and $\beta_1, \beta_2$ weight current and delayed tip velocities. Initial data give $u(0,x)$, $u_t(0,x)$, and $u_t(s,1)$ for $-\tau < s \leq 0$. We predict displacement and velocity, $(u, u_t)$.

**Results.** DS4D-FFNO achieves the lowest mean full-field and boundary-region nRMSE among the eight models on both systems (Table 1). S4-FFNO ranks second. Compared with S4-FFNO, DS4D-FFNO reduces mean full-field error by approximately $56\%$ on KdV–KdV system and $50\%$ on the Microbeam system.

Table 1: Test nRMSE on the 1D benchmarks (mean $\pm$ sample SD, seeds 0–2). Bold/underline mark the best/second-best unrounded means. Settings and metrics are in Appendices A and C.

| Model | KdV–KdV system | | Microbeam system | |
|---|---|---|---|---|
| | Full nRMSE ↓ | Boundary nRMSE ↓ | Full nRMSE ↓ | Boundary nRMSE ↓ |
| DS4D-FFNO | $\mathbf{(1.86 \pm 2.54) \times 10^{-2}}$ | $\mathbf{(4.55 \pm 6.35) \times 10^{-2}}$ | $\mathbf{(1.72 \pm 1.93) \times 10^{-2}}$ | $\mathbf{(2.39 \pm 2.53) \times 10^{-2}}$ |
| S4-FFNO | $\underline{(4.18 \pm 6.38) \times 10^{-2}}$ | $\underline{(8.18 \pm 11.95) \times 10^{-2}}$ | $\underline{(3.41 \pm 4.69) \times 10^{-2}}$ | $\underline{(4.44 \pm 5.88) \times 10^{-2}}$ |
| FNO | $(4.39 \pm 0.70) \times 10^{-1}$ | $(9.21 \pm 0.67) \times 10^{-1}$ | $(2.47 \pm 2.47) \times 10^{0}$ | $(2.27 \pm 2.20) \times 10^{0}$ |
| FFNO | $(3.62 \pm 0.88) \times 10^{-1}$ | $(7.06 \pm 1.78) \times 10^{-1}$ | $(9.56 \pm 0.02) \times 10^{-1}$ | $(1.01 \pm 0.005) \times 10^{0}$ |
| MambaFormer | $(1.09 \pm 0.93) \times 10^{1}$ | $(1.88 \pm 1.81) \times 10^{1}$ | $(5.18 \pm 0.79) \times 10^{-1}$ | $(5.68 \pm 0.93) \times 10^{-1}$ |
| Transolver | $(2.22 \pm 0.17) \times 10^{-1}$ | $(4.05 \pm 0.46) \times 10^{-1}$ | $(9.62 \pm 0.07) \times 10^{-1}$ | $(1.00 \pm 0.01) \times 10^{0}$ |
| MambaFormer–FFNO | $(7.13 \pm 2.14) \times 10^{-1}$ | $(1.25 \pm 0.25) \times 10^{0}$ | $(1.13 \pm 0.65) \times 10^{0}$ | $(1.12 \pm 0.59) \times 10^{0}$ |
| Transolver–FFNO | $(5.58 \pm 1.49) \times 10^{-1}$ | $(1.10 \pm 0.34) \times 10^{0}$ | $(9.64 \pm 0.13) \times 10^{-1}$ | $(1.07 \pm 0.10) \times 10^{0}$ |

DS4D-FFNO has lower mean error than S4-FFNO at every forecast step on KdV–KdV system and all but step 2 on the Microbeam system (Figure 3a,e). Figure 3b–d compares the numerical reference, DS4D-FFNO prediction, and absolute error for surface elevation $\eta$ in KdV–KdV system. Figure 3f–h shows the corresponding comparison for velocity $u_t$ in the Microbeam system. Each heatmap displays the complete 24-step forecast, with spatial position on the horizontal axis and forecast step on the vertical axis.

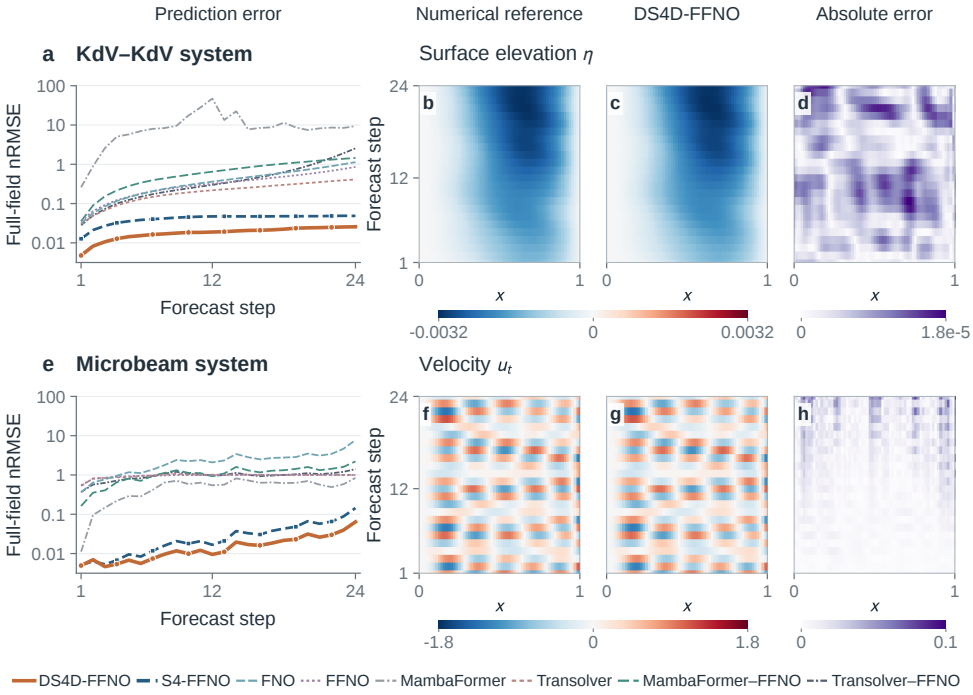


Figure 3: One-dimensional prediction results. **a,e**, mean full-field nRMSE over 24 forecast steps (three seeds, logarithmic scale). **b–d,f–h**, numerical reference, DS4D-FFNO prediction, and absolute error for KdV–KdV system surface elevation $\eta$ and Microbeam system velocity $u_t$, respectively. Reference and prediction share symmetric linear color limits within each row. Errors use separate linear scales.

On the Kawahara equation, DS4D-FFNO likewise achieves the lowest mean full-field and boundary-region nRMSE among all eight models. It reduces mean full-field error by approximately $83\%$ relative to S4-FFNO, the strongest baseline (Appendix D.1).

### 4.2 Two-Dimensional Experiments

**Setup.** We evaluate a thermoelastic strip and a disk with delayed boundary dynamics on a $64 \times 64$ grid, using $L = 96$ observations and $H = 12$ prediction steps. Both have physical delay $\tau = 0.32$, or $m = \tau/\Delta T = 64$ recorded steps. Here $\partial_\nu$ is the outward-normal derivative and $\Delta_\Gamma$ the tangential Laplace–Beltrami operator. Models train for 60 epochs with batch size 4. Appendix A specifies coefficients, initial histories, and solvers.

**Thermoelastic system.** This linear system has delayed Wentzell boundary conditions and couples vector displacement $u \in \mathbb{R}^2$ and temperature $\theta$ on a strip $\Omega$ (Boureghida & Kasri, 2026). The left and right edges are $\Gamma_0$ and $\Gamma_1$, respectively, and all fields are periodic in the transverse direction. The governing equations and boundary conditions are

$$\begin{aligned}
&u_{tt} - \Delta u + \gamma \nabla \theta + a u_t = 0 && \text{in } \Omega,\\
&\theta_t - \Delta \theta + \gamma \nabla \cdot u_t = 0 && \text{in } \Omega,\\
&u_{tt} + \partial_\nu u - \Delta_\Gamma u + \mu_0 u_t + \mu_1 u_t(t-\tau) = 0 && \text{on } \Gamma_1,\\
&\theta_t + \partial_\nu \theta - \Delta_\Gamma \theta + \kappa_0 \theta = 0 && \text{on } \Gamma_0,\\
&u = 0 && \text{on } \Gamma_0,\\
&\theta = 0 && \text{on } \Gamma_1.
\end{aligned} \tag{13}$$

The controlled right edge $\Gamma_1$ carries delayed mechanical feedback. The coefficients $\gamma$ and $a$ govern thermoelastic coupling and interior damping. The coefficients $\mu_0, \mu_1$ weight current and delayed boundary velocities, and $\kappa_0$ sets thermal boundary damping. Initial data prescribe displacement, velocity, and temperature, together with the boundary-velocity history $u_t(s,\cdot)|_{\Gamma_1}$ for $-\tau < s \le 0$. We predict both displacement components, both velocity components, and temperature.

**Delayed linear problem.** This problem defines a harmonic interior field on the unit disk $\Omega = \{x \in \mathbb{R}^2 : \|x\| < 1\}$ subject to a delayed dynamic boundary condition (Casal et al., 2011):

$$\begin{aligned}
&-\Delta u(t,x) = 0 && \text{in } \Omega,\\
&u_t(t,x) + \partial_\nu u(t,x) + b_0 u(t-\tau,x) = 0 && \text{on } \partial\Omega.
\end{aligned} \tag{14}$$

Here $b_0$ weights the delayed boundary trace. The boundary history $u(s,\cdot)|_{\partial\Omega}$ is prescribed for $-\tau < s \le 0$, with harmonic extension determining the initial interior field. The prediction target is the scalar harmonic field $u$.

**Results.** DS4D-FFNO also achieves the lowest mean full-field nRMSE on both two-dimensional benchmarks (Table 2). It reduces mean error by approximately $13\%$ relative to FFNO on Thermoelastic system and $9\%$ relative to S4-FFNO on Delayed linear problem, the strongest baselines on the respective systems. For boundary-region nRMSE, DS4D-FFNO and S4-FFNO rank first and second on Delayed linear problem. On Thermoelastic system, DS4D-FFNO and FFNO have nearly equal mean boundary errors. DS4D-FFNO has lower mean error than S4-FFNO at every forecast step on both benchmarks (Figure 4a,e). The remaining panels show representative predictions and absolute errors at the final step.

Table 2: Test nRMSE on the 2D benchmarks (mean ± sample SD, seeds 0–2). Metric definitions are in Appendix C and evaluation masks are in Table 4. †Boundary-region entries for the Thermoelastic system are exploratory.

| Model | Thermoelastic system | | Delayed linear problem | |
|---|---|---|---|---|
| | Full nRMSE ↓ | Boundary nRMSE† ↓ | Full nRMSE ↓ | Boundary nRMSE ↓ |
| DS4D-FFNO | $\mathbf{(8.55\pm0.16)\times10^{-3}}$ | $\mathbf{(1.69\pm0.18)\times10^{-2}}$ | $\mathbf{(1.53\pm0.10)\times10^{-2}}$ | $\mathbf{(1.61\pm0.10)\times10^{-2}}$ |
| S4-FFNO | $(1.03\pm0.04)\times10^{-2}$ | $(1.82\pm0.17)\times10^{-2}$ | $(1.68\pm0.16)\times10^{-2}$ | $(1.76\pm0.14)\times10^{-2}$ |
| FNO | $(4.95\pm0.59)\times10^{-2}$ | $(6.84\pm1.10)\times10^{-2}$ | $(4.63\pm0.07)\times10^{-1}$ | $(4.11\pm0.04)\times10^{-1}$ |
| FFNO | $(9.80\pm0.52)\times10^{-3}$ | $(1.69\pm0.20)\times10^{-2}$ | $(4.66\pm0.13)\times10^{-1}$ | $(4.11\pm0.10)\times10^{-1}$ |
| MambaFormer | $(1.46\pm0.45)\times10^{0}$ | $(1.46\pm0.47)\times10^{0}$ | $(1.45\pm0.14)\times10^{0}$ | $(1.30\pm0.11)\times10^{0}$ |
| Transolver | $(4.71\pm0.13)\times10^{-2}$ | $(4.74\pm0.08)\times10^{-2}$ | $(4.60\pm0.28)\times10^{-1}$ | $(4.15\pm0.19)\times10^{-1}$ |
| MambaFormer–FFNO | $(5.11\pm0.65)\times10^{-2}$ | $(6.01\pm0.86)\times10^{-2}$ | $(6.87\pm0.45)\times10^{-2}$ | $(6.55\pm0.46)\times10^{-2}$ |
| Transolver–FFNO | $(1.20\pm0.06)\times10^{-2}$ | $(2.38\pm0.09)\times10^{-2}$ | $(4.53\pm0.16)\times10^{-1}$ | $(4.02\pm0.09)\times10^{-1}$ |

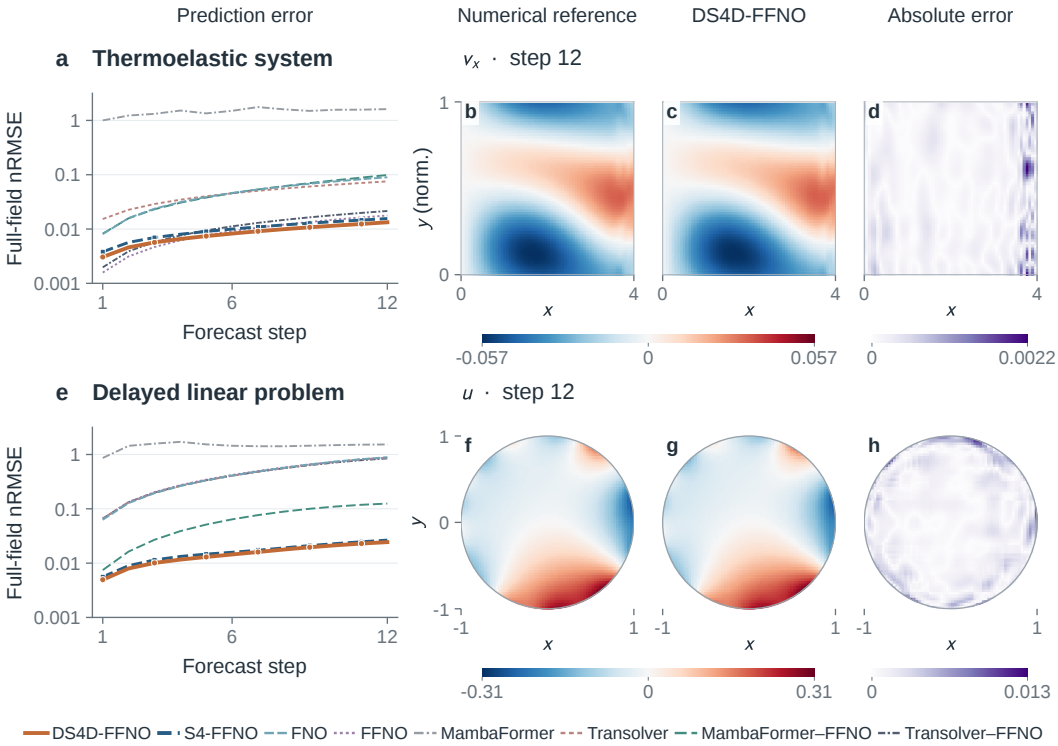


Figure 4: Two-dimensional prediction results. **a,e**, mean full-field nRMSE over 12 forecast steps (three seeds, logarithmic scale). **b–d,f–h**, numerical reference, DS4D-FFNO prediction, and absolute error at step 12 for Thermoelastic system velocity $v_x$ and Delayed linear problem field $u$, respectively.

DS4D-FFNO also achieves the lowest mean full-field and boundary-region nRMSE among all eight models on the Reaction–diffusion model. It reduces mean full-field error by approximately 16% relative to S4-FFNO, the strongest baseline (Appendix D.1).

### 4.3 PDEs Without Physical Delay

To assess predictive accuracy when physical delay is absent, we compare DS4D-FFNO with S4-FFNO on six separate PDEs. DS4D-FFNO retains its learnable delayed branch. The two architectures achieve similar mean test full-field nRMSE (Table 3). DS4D-FFNO has slightly lower error on four equations and slightly higher error on viscous Burgers and advection–diffusion. Appendix E gives the experimental settings.

Table 3: DS4D-FFNO and S4-FFNO on six PDEs without physical delay. Test full-field nRMSE is reported as mean ± sample standard deviation over three model seeds. Lower is better.

| Dimension | Benchmark | S4-FFNO | DS4D-FFNO |
|---|---|---|---|
| 1D | Viscous Burgers | $\mathbf{(1.27\pm0.44)\times10^{-3}}$ | $(1.30\pm0.34)\times10^{-3}$ |
| | Advection–diffusion | $\mathbf{(3.97\pm1.80)\times10^{-4}}$ | $(4.25\pm1.60)\times10^{-4}$ |
| | Allen–Cahn | $(1.42\pm0.67)\times10^{-3}$ | $\mathbf{(1.40\pm0.43)\times10^{-3}}$ |
| 2D | Navier–Stokes (vorticity) | $(7.73\pm1.21)\times10^{-3}$ | $\mathbf{(7.26\pm1.50)\times10^{-3}}$ |
| | Gray–Scott | $(4.12\pm0.12)\times10^{-4}$ | $\mathbf{(3.91\pm0.17)\times10^{-4}}$ |
| | Nonlinear shallow water | $(3.62\pm0.06)\times10^{-2}$ | $\mathbf{(3.58\pm0.10)\times10^{-2}}$ |

### 4.4 Ablation Experiments

We assess delay sensitivity on the six delayed benchmarks by permuting learned delays across channels or randomly replacing their values. All other parameters remain fixed, and neither intervention involves retraining.

Both interventions increase mean full-field nRMSE on all six delayed benchmarks. Channel permutation raises error to 14.4 times its original value on KdV–KdV system. Random replacement raises it to 12.7 times on Delayed linear problem (Figure 5). These results show that the trained predictor is sensitive to the learned delay values and their channel assignments. Appendix D.2 gives the intervention details, and Table 8 reports the complete six-benchmark statistics.

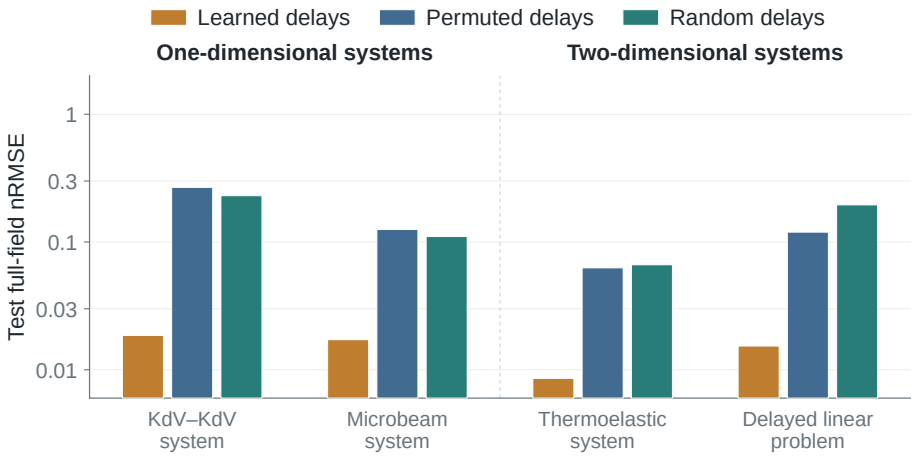


Figure 5: Delay sensitivity on four delayed PDE benchmarks. Bars show three-seed mean test full-field nRMSE for learned, channel-permuted, and randomly replaced delays on a logarithmic scale.

We also examine spatial resolution and model parameters on the Microbeam system and Delayed linear problem. The resolution study compares all eight models on four grids, with training and evaluation performed at the same resolution. DS4D-FFNO achieves the lowest mean full-field nRMSE on all tested grids. The parameter study varies DS4D-FFNO's initial delayed-branch allocation, hidden width, state size, and Fourier cutoff one at a time. Appendix D.3 reports both studies.

## 5 Conclusion

We presented DS4D-FFNO, which integrates FFNO spatial mixing with delayed hidden-state retrieval for PDEs with fixed-delay boundary feedback. DS4D retains hidden-state histories of spatially mixed features and retrieves interpolated past states at delays learned for each channel. These states enter the recurrent update, whose output is spatially mixed to predict the evolving field. Experiments show improved prediction on delayed PDEs while retaining accuracy comparable to S4-FFNO on PDEs without physical delay. Together, these findings support learning the timing of historical-state access alongside spatial transformations, providing a neural operator that couples field interactions with the temporal structure of delayed feedback.

## AI USE STATEMENT

Generative AI assisted with manuscript organization, language editing, LaTeX preparation, and writing code for running experiments and generating figures. The authors reviewed the text, code, numerical results, and citations, and take responsibility for the manuscript and documented experiments.

## REPRODUCIBILITY STATEMENT

Section 3 describes the FFNO spatial layers, delayed hidden-state updates, and autoregressive prediction in DS4D-FFNO. Appendix A gives benchmark equations, data-generation procedures, and training settings for the six delayed benchmarks. Appendix E gives the settings for the six PDEs without physical delay. Appendix B specifies DS4D initialization and memory length, and Appendix C defines the training loss and test metrics.

## REFERENCES


Andrii Anikushyn and Michael Pokojovy. Global well-posedness and exponential stability for heterogeneous anisotropic Maxwell's equations under a nonlinear boundary feedback with delay. *Journal of Mathematical Analysis and Applications*, 475(1):278–312, 2019. doi: 10.1016/j.jmaa.2019.02.042.

Remissa Boureghida and Hicham Kasri. The multidimensional thermoelastic system with variable coefficients and Wentzell boundary conditions: Uniform stability. *Reports on Mathematical Physics*, 97(3):379–405, 2026. doi: 10.1016/S0034-4877(26)00041-8.

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

Ricardo Buitrago Ruiz, Tanya Marwah, Albert Gu, and Andrej Risteski. On the benefits of memory for modeling time-dependent PDEs. In *International Conference on Learning Representations*, 2025.

Roberto de A. Capistrano-Filho, Boumediène Chentouf, Luan S. de Sousa, and Victor H. Gonzalez Martinez. Two stability results for the Kawahara equation with a time-delayed boundary control. *Zeitschrift für Angewandte Mathematik und Physik*, 74(1):16, 2023. doi: 10.1007/s00033-022-01897-4.

Roberto de A. Capistrano-Filho, Boumediène Chentouf, Victor H. Gonzalez Martinez, and Juan Ricardo Muñoz. On the boundary stabilization of the KdV–KdV system with time-dependent delay. *Nonlinear Analysis: Real World Applications*, 79:104122, 2024. doi: 10.1016/j.nonrwa.2024.104122.

Alfonso C. Casal, Jesús Ildefonso Díaz, and José M. Vegas. Finite extinction time property for a delayed linear problem on a manifold without boundary. *Discrete and Continuous Dynamical Systems*, 2011(Special):265–271, 2011. doi: 10.3934/proc.2011.2011.265.

Yi Cheng, Xin Wang, Baowei Feng, and Donal O'Regan. Semigroup well-posedness and exponential stability for the von Kármán beam equation under the combined boundary control of nonlinear delays and non-delays. *Nonlinear Analysis: Real World Applications*, 80:104143, 2024. doi: 10.1016/j.nonrwa.2024.104143.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022a.

Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35971–35983, 2022b. doi: 10.52202/068431-2607.

Martin Gugat and Markus Dick. Time-delayed boundary feedback stabilization of the isothermal Euler equations with friction. *Mathematical Control and Related Fields*, 1(4):469–491, 2011. doi: 10.3934/mcrf.2011.1.469.

Shangjiang Guo. Bifurcation in a reaction-diffusion model with nonlocal delay effect and nonlinear boundary condition. *Journal of Differential Equations*, 289:236–278, 2021. doi: 10.1016/j.jde.2021.04.021.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17616–17635. PMLR, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624, 2024. doi: 10.52202/079017-2311.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

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

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

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Serge Nicaise and Cristina Pignotti. Stability and instability results of the wave equation with a delay term in the boundary or internal feedbacks. *SIAM Journal on Control and Optimization*, 45 (5):1561–1585, 2006. doi: 10.1137/060648891.

Serge Nicaise and Cristina Pignotti. Stability of the wave equation with localized Kelvin–Voigt damping and boundary delay feedback. *Discrete and Continuous Dynamical Systems - S*, 9(3): 791–813, 2016. doi: 10.3934/dcdss.2016029.

Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. Can Mamba learn how to learn? A comparative study on in-context learning tasks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39793–39812. PMLR, 2024.

Yixiao Qian, Song Chen, Jiaxu Liu, Shengze Cai, and Chao Xu. DSSMs: State space models with explicit memory via delay differential equations. *arXiv preprint arXiv:2607.10244*, 2026. doi: 10.48550/arXiv.2607.10244.

Désiré Saba, Akram Ben Aissa, and Gilbert Bayili. Stabilization of a microbeam system with a boundary or internal distributed delay. *Annals of the University of Craiova, Mathematics and Computer Science Series*, 50(2):442–463, 2023. doi: 10.52846/ami.v50i2.1742.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized Fourier neural operators. In *International Conference on Learning Representations*, 2023.

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

C. H. Yao, H. J. Fan, Y. M. Zhao, and Y. F. Tang. Finite element approximation for Maxwell's equations with Debye memory under a nonlinear boundary feedback with delay. *Communications in Nonlinear Science and Numerical Simulation*, 119:107082, 2023. doi: 10.1016/j.cnsns.2022.107082.

Minghao Yu, Yingqi Feng, William Fairman, Bing Ouyang, Paul S. Wills, Minghan Wei, and Yufei Tang. Physics-guided transfer learning for predicting dissolved oxygen in smart aquaculture farms. *IET Cyber-Systems and Robotics*, 8(1):e70053, 2026. doi: 10.1049/csy2.70053.

Li Zhang and Gábor Stépán. Exact stability chart of an elastic beam subjected to delayed feedback. *Journal of Sound and Vibration*, 367:219–232, 2016. doi: 10.1016/j.jsv.2016.01.002.

# A Core Experimental Configuration

Unless stated otherwise, the integration step equals the recording interval $\Delta T$ (Table 4). Each benchmark requires initial fields and boundary histories over $s \in [-\tau, 0]$ for feedback during the first delay interval.

Table 4: Benchmark settings and evaluation masks.

| Dimension | Case | State | Split | $\tau$ | $\Delta T$ | Seed | Boundary |
|---|---|---|---|---|---|---|---|
| 1D | KdV–KdV system | $(\eta, \omega)$ | 384/96/96 | 0.128 | 0.001 | 20260827 | Rightmost 16 points |
| | Microbeam system | $(u, u_t)$ | 384/96/96 | 1.0 | 0.0078125 | 20260827 | Rightmost 16 points |
| | Kawahara equation | $u$ | 1536/96/96 | 0.64 | 0.005 | 20260827 | Rightmost 16 points |
| 2D | Thermoelastic system | $(u_x, u_y, v_x, v_y, \theta)$ | 192/48/64 | 0.32 | 0.005 | 20260907 | Rightmost column |
| | Delayed linear problem | $u$ | 192/48/64 | 0.32 | 0.005 | 20260906 | $0.72 \le r \le 1$ |
| | Reaction–diffusion model | $u$ | 192/48/64 | 0.32 | 0.005 | 20260907 | Topmost 16 rows |

$\tau$ is the physical delay and $\Delta T$ the recording interval. Splits list training/validation/test trajectory counts. Seeds are for data generation. Full-field masks cover the complete grid. For Delayed linear problem, they are restricted to the disk $r \le 1$, where $r$ is radial distance. Thermoelastic subscripts $x, y$ denote vector components.

## A.1 One-Dimensional Benchmarks

All three benchmarks use $x_i = i/64$, $i = 0, \ldots, 64$.

**KdV–KdV system.** For Equation (11), we set $\alpha = 0.501$ and $\beta = 0.500$. We use spatial finite differences, backward Euler for linear coupling, and explicit nonlinear terms. To start time integration, we construct initial elevation, velocity, and a compatible right-boundary gradient history. Initial fields use random low-order modes with boundary envelopes. The history adds sinusoidal perturbations and a linear trend to the initial right-boundary gradient, joining it at $s = 0$. We then adjust the velocity field and boundary grid values to satisfy the initial boundary conditions.

**Microbeam system.** For Equation (12), we set $\rho A = M_1 = 1$, $M_2 = 0.02$, $\beta_1 = 0.8$, and $\beta_2 = 0.799$. The Ritz–Galerkin discretization uses eight modes satisfying the clamped-end and two homogeneous free-tip conditions. Time integration uses the average-acceleration Newmark method. To satisfy the initial delayed shear condition, we jointly construct initial displacement, velocity, and free-tip velocity history. Specifically, we first randomly sample initial displacement and velocity in this modal space. The history adds sinusoidal perturbations that vanish at both endpoints to the initial tip velocity. Within the same modal space, we apply the smallest Euclidean correction to the displacement coefficients to satisfy the delayed shear condition.

**Kawahara equation.** The cubic Kawahara equation (Capistrano-Filho et al., 2023) on $0 < x < 1$ is

$$\begin{aligned} &u_t + au_x + bu_{xxx} - u_{xxxxx} + u^p u_x = 0, \\ &u(t,0) = u(t,1) = u_x(t,0) = u_x(t,1) = 0, \\ &u_{xx}(t,1) = \alpha u_{xx}(t,0) + \beta u_{xx}(t-\tau,0). \end{aligned} \tag{15}$$

We set $a = b = 1$, $p = 2$, $\alpha = 0.499$, and $\beta = 0.500$. We use nine-point finite differences, backward Euler for linear terms, and explicit nonlinear transport. To satisfy the delayed boundary conditions, we construct the initial field and left-boundary curvature history jointly. Specifically, we generate a candidate field using random sine modes under an envelope, then perturb its left-boundary curvature to determine the history's starting value. Using this value as the delayed curvature at $t = 0$, we correct the initial field to satisfy all five discrete boundary conditions. Linear interpolation and sinusoidal perturbations that vanish at both endpoints connect the history's starting value to the corrected initial curvature.

### A.2 Two-Dimensional Benchmarks

**Thermoelastic system.** For Equation (13), we set $\gamma = 0.20$, $a = 0.02$, $\mu_0 = 0.60$, $\mu_1 = 0.599$, and $\kappa_0 = 1$. Elastic and thermal diffusion coefficients are both one. We use a $64 \times 64$ grid with spacing $4/63$ in both directions and transverse period $256/63$, with centered spatial differences. Euler steps of $\Delta T/8$ update velocity first, then displacement and temperature using the new velocity. To generate trajectories, we construct initial displacement, velocity, and temperature, then generate a history joining the initial right-boundary velocity. Initial fields combine random spatial modes satisfying the homogeneous boundary values. At $x = 4$, the velocity history adds space–time perturbations to the initial boundary velocity. These perturbations vanish at both ends of the history interval.

**Delayed linear problem.** For Equation (14), we set $b(t) = b_0 = 4.5$. On a $64 \times 64$ grid covering $[-1, 1]^2$, fields inside the unit disk use 25 Steklov harmonic modes. Grid values outside the disk are zero. Modal damping uses implicit Euler, and delayed feedback is evaluated explicitly. To join the boundary history to the initial field, we use the same harmonic modes for both. Specifically, for each mode, we sample a coefficient history combining a random periodic term with a linear trend. The coefficients at $s = 0$ determine the initial harmonic field.

**Reaction–diffusion model.** Following the family of Guo (2021), we set the interior delay to zero and choose the following nonlinear boundary-feedback map on $\Omega = (0, 1)^2$:

$$\begin{aligned} &u_t = d\Delta u + ru\big(1 - \overline{u}(t)\big), \\ &\partial_\nu u = g_b \sin^2(\pi x) \tanh(\kappa[u(t-\tau, x, 1) - u_*]) \qquad \text{on } y = 1, \\ &\partial_\nu u = 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{on the other edges.} \end{aligned} \tag{16}$$

Here $\overline{u}$ averages all grid nodes and $\partial_\nu$ is the outward-normal derivative. We set $d = 0.0015$, $r = 0.4$, $g_b = 114$, $\kappa = 5$, and $u_* = 0.45$. We use a uniform $64 \times 64$ grid with five-point interior differences. Noncorner top-edge nodes use Neumann ghost points, while other edges retain only normal second differences and corner diffusion is zero. Time integration uses forward Euler with step size $\Delta T/4$, holding delayed flux fixed within each recording interval. To generate trajectories, we construct an initial field and a boundary history joining its top-edge values. The initial field and the history's starting profile are obtained by adding random low-order modes to $u_*$. The history linearly interpolates between its starting profile and the initial top-edge values, with sinusoidal perturbations that vanish at both temporal endpoints. Samples are recorded after a burn-in of one delay interval.

### A.3 Model and Training Settings

**Model configurations.** Tables 5 and 6 summarize model sizes, core block sequences, and attention heads.

Table 5: Model widths, state sizes, and parameter counts.

| Model | Width $C$ | State size $N$ | Trainable parameters | | | |
|---|---|---|---|---|---|---|
| | | | KdV–KdV/ Microbeam | Kawahara | Thermoelastic | Delayed linear/ Reaction–diffusion |
| DS4D-FFNO | 32/32/32 | 24 | 308,514 | 308,417 | 153,189 | 152,801 |
| S4-FFNO | 32/32/32 | 24 | 307,714 | 307,617 | 152,389 | 152,001 |
| FNO | 32/14/14 | — | 285,378 | 285,281 | 156,861 | 156,689 |
| FFNO | 32/32/32 | — | 306,114 | 306,017 | 150,789 | 150,401 |
| MambaFormer | 52/3/14 | 24 | 294,086 | 287,261 | 148,961 | 156,626 |
| Transolver | 70/32/32 | — | 308,627 | 308,416 | 149,583 | 149,195 |
| MambaFormer–FFNO | 28/24/24 | 24 | 315,436 | 315,351 | 149,129 | 148,837 |
| Transolver–FFNO | 32/29/29 | — | 324,484 | 324,387 | 152,047 | 151,695 |

The width triple $C$ is ordered as: all one-dimensional benchmarks, Thermoelastic system, and Delayed linear problem with Reaction–diffusion model. $N$ is the state size of the S4D or Mamba core. Dashes indicate no temporal SSM. Counts are from seed-0 runs, with each complex parameter counted as two real scalars.

FNO and FFNO normalize each spatial coordinate component to $[0, 1]$. Both use biased linear encoders and biased two-layer output heads with hidden width $2C$ and no activation. Their spatial blocks omit dropout, normalization, and padding.

Following MemNO (Buitrago Ruiz et al., 2025), our FFNO layers use fourfold MLP expansion and GELU. S4-FFNO and DS4D-FFNO retain 33 one-sided Fourier modes in one dimension and seven per axis in two dimensions. FNO follows spectral mixing with a width-$C$ two-layer MLP, then adds the pointwise affine branch and outer residual in order. Only two-dimensional FNO applies GELU between these additions.

Table 6: Core block sequences and attention heads. MQA denotes causal multi-query attention; its head count refers to query heads. Brackets group repeated attention–Mamba pairs. The notation omits input/output projections, normalization, and residual connections.

| Model | Core block sequence | Attention heads (1D/2D) |
|---|---|---|
| DS4D-FFNO | 2 FFNO, 1 DS4D, 2 FFNO | — |
| S4-FFNO | 2 FFNO, 1 S4D, 2 FFNO | — |
| FNO | 4 frame-wise FNO | — |
| FFNO | 4 frame-wise FFNO | — |
| MambaFormer | $\mathrm{Mamba} \to [\mathrm{MQA} \to \mathrm{Mamba}] \times 4$ | 4/1 |
| Transolver | 5 frame-wise Physics-Attention | 2/2 |
| MambaFormer–FFNO | $\mathrm{FFNO} \times 2 \to \mathrm{Mamba} \to [\mathrm{MQA} \to \mathrm{Mamba}] \times 3 \to \mathrm{FFNO} \times 2$ | 4/4 |
| Transolver–FFNO | 2 FFNO, 1 frame-wise Transolver, 2 FFNO | 2/2 |

**Baseline adapters.** Standalone MambaFormer uses affine projections to encode each flattened field as one temporal token and decode it to the grid. MambaFormer–FFNO processes each grid point's latent time sequence. Both retain input LayerNorm, causal multi-query attention, and residual ordering, including outer residuals. Mamba expansion and convolution width are both 4, with time-step projection rank $\lceil C/16 \rceil$. Dropout and positional embeddings are disabled.

Transolver (Wu et al., 2024) combines each grid point's features and coordinates into one spatial token. Standalone Transolver and Transolver–FFNO both use 32 slices per head and fourfold MLP expansion. We use the authors' implementations for irregular meshes in one dimension and structured meshes in two dimensions. Dropout, time embeddings, and reference-grid position embeddings are disabled.

**Training and inference.** During training, DS4D-FFNO, S4-FFNO, MambaFormer, and MambaFormer–FFNO causally process $L + H - 1$ ground-truth fields to predict the next field at each step (Buitrago Ruiz et al., 2025). FNO, FFNO, Transolver, and Transolver–FFNO each predict the next field from only the current ground-truth field and spatial coordinates, without retaining state between time steps. All models use the same final $H$ ground-truth fields as targets and compute loss only at these target times.

At inference, all models feed their predictions back as inputs to generate $H$ successive fields. FNO, FFNO, Transolver, and Transolver–FFNO start from the last observed field and then use the preceding prediction at each step. DS4D-FFNO and S4-FFNO retain the states obtained from the $L$ observations, with DS4D-FFNO also retaining its hidden-state history. Both MambaFormer models recompute the sequence of observations and previous predictions at each step.

**Normalization.** Channel means and sample standard deviations are computed from the $L$ observed fields of training trajectories over valid spatial points, restricted to the disk for Delayed linear problem. Standard deviations are floored at $10^{-6}$, and all splits use these statistics.

**Optimization.** AdamW uses $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, zero weight decay, and gradient norm clipping at 1.0 (Loshchilov & Hutter, 2019). The learning rate decays from $10^{-3}$ to zero by an epoch-wise cosine schedule without warmup (Loshchilov & Hutter, 2017). Two-dimensional runs accumulate four single-trajectory gradients per update. Validation at epoch 1 and every tenth epoch selects the checkpoint with the lowest autoregressive full-field nRMSE. Trajectories are reshuffled each epoch, using the same order for paired DS4D-FFNO and S4-FFNO runs.

## B Delay Initialization and Memory Length

**DS4D parameterization.** Following Qian et al. (2026), each channel $c = 1, \dots, C$ stores $N/2$ independent complex modes, with conjugate pairing giving a real readout. For mode $r = 1, \dots, N/2$, $\zeta_{cr}$ sets the continuous-time decay and frequency. After discretization, $1 - \rho_{cr}$ and $\rho_{cr} \in (0, 1)$ specify the relative magnitude shares of the previous-state and delayed-state branches, respectively. We initialize

$$\zeta_{cr}^{(0)} = -\frac{1}{2} + \mathrm{i}\pi(r-1), \qquad \rho_{cr}^{(0)} = 0.003. \tag{17}$$

Here $\mathrm{i}$ is the imaginary unit, and superscript $(0)$ denotes initialization. At each grid point, the memory stores the previous $L$ states. The $C$ channels have delays initialized independently from a uniform distribution:

$$q_c^{(0)} \sim \mathcal{U}(\epsilon, 1-\epsilon), \qquad l_c^{(0)} = 1 + (L-2)q_c^{(0)}, \qquad \epsilon = 10^{-6}. \tag{18}$$

Here $q_c^{(0)}$ specifies the channel's relative position in the delay range. The trainable logit is initialized as $a_c^{(0)} = \mathrm{logit}(q_c^{(0)})$. The hidden state and state-history buffer are updated within each sequence and reset only before a new sequence, including during evaluation. Training updates the delays $l_c$ and branch magnitude fractions $\rho_{cr}$ along with the other model parameters.

## C Error Metrics and Boundary Regions

For $N_s$ sequences, a spatial mask $M_R$, and a set of forecast steps $\mathcal{T} \subseteq \{1, \dots, H\}$, nRMSE is

$$E_{R,\mathcal{T}} = \frac{1}{N_s|\mathcal{T}|} \sum_{i=1}^{N_s} \sum_{h \in \mathcal{T}} \frac{\left\| M_R \odot (\widehat{\boldsymbol{Z}}_{n+h}^{(i)} - \boldsymbol{Z}_{n+h}^{(i)}) \right\|_2}{\left\| M_R \odot \boldsymbol{Z}_{n+h}^{(i)} \right\|_2 + 10^{-8}}. \tag{19}$$

Here $n$ is the last observed step, and $\boldsymbol{Z}$ and $\widehat{\boldsymbol{Z}}$ are true and predicted fields in physical units. Norms include all selected points and channels. Full-field scores are computed over the entire physical domain and averaged across all $H$ forecast steps. Per-step curves show the full-field error separately at each forecast step. Boundary scores use all $H$ steps with the masks in Table 4. Training uses full-field nRMSE over all $H$ teacher-forced outputs. Validation and test scores are averaged over their respective complete splits.

# D Extended Results and Ablation Details

## D.1 Additional Delayed PDE Prediction Results

Relative to S4-FFNO, DS4D-FFNO reduces mean full-field error by approximately $83\%$ on Kawahara equation and $16\%$ on Reaction–diffusion model (Table 7), with lower error at every forecast step (Figure 6).

Table 7: Supplementary test nRMSE on the Kawahara equation and Reaction–diffusion model (mean ± sample SD over three seeds, 0–2). Metric definitions are in Appendix C and evaluation masks are in Table 4.

| Model | Full nRMSE ↓ | | Boundary nRMSE ↓ | |
|---|---|---|---|---|
| | Kawahara equation | Reaction–diffusion model | Kawahara equation | Reaction–diffusion model |
| DS4D-FFNO | $\mathbf{(1.06\pm\ 0.40)\times10^{-2}}$ | $\mathbf{(1.10\pm0.03)\times10^{-2}}$ | $\mathbf{(1.55\pm\ 0.37)\times10^{-2}}$ | $\mathbf{(1.84\pm0.04)\times10^{-2}}$ |
| S4-FFNO | $(6.38\pm\ 7.99)\times10^{-2}$ | $(1.31\pm0.15)\times10^{-2}$ | $(7.24\pm\ 8.64)\times10^{-2}$ | $(2.05\pm0.14)\times10^{-2}$ |
| FNO | $(9.97\pm\ 0.01)\times10^{-1}$ | $(1.83\pm0.01)\times10^{-2}$ | $(9.98\pm\ 0.02)\times10^{-1}$ | $(3.03\pm0.01)\times10^{-2}$ |
| FFNO | $(9.98\pm\ 0.01)\times10^{-1}$ | $(1.89\pm0.01)\times10^{-2}$ | $(1.00\pm0.002)\times10^{0}$ | $(3.15\pm0.01)\times10^{-2}$ |
| MambaFormer | $(9.28\pm11.74)\times10^{-1}$ | $(2.96\pm0.36)\times10^{-1}$ | $(9.51\pm12.19)\times10^{-1}$ | $(4.36\pm0.43)\times10^{-1}$ |
| Transolver | $(9.98\pm\ 0.01)\times10^{-1}$ | $(2.17\pm0.06)\times10^{-2}$ | $(9.98\pm\ 0.01)\times10^{-1}$ | $(3.53\pm0.08)\times10^{-2}$ |
| MambaFormer–FFNO | $(1.41\pm\ 0.25)\times10^{0}$ | $(3.42\pm0.89)\times10^{-2}$ | $(2.09\pm\ 0.90)\times10^{0}$ | $(3.90\pm0.59)\times10^{-2}$ |
| Transolver–FFNO | $(9.99\pm\ 0.01)\times10^{-1}$ | $(1.90\pm0.01)\times10^{-2}$ | $(1.00\pm0.004)\times10^{0}$ | $(3.16\pm0.01)\times10^{-2}$ |

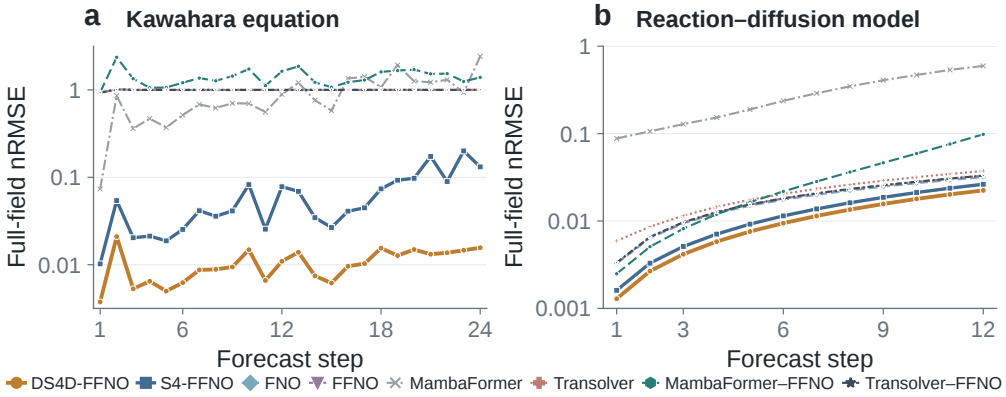


Figure 6: Prediction curves for the two additional benchmarks. **a**, Kawahara equation over all 24 forecast steps. **b**, Reaction–diffusion model over all 12 forecast steps. Curves show per-step full-field nRMSE for all eight models, averaged over three model seeds on the complete test split. Both vertical axes are logarithmic, with independently chosen limits.

## D.2 Delay-Value Interventions

Interventions modify only the channel delays in validation-selected DS4D-FFNO checkpoints, without retraining. Here $c = 1, \ldots, 32$ indexes hidden channels, and $L$ is the number of observed fields, equal to the state-memory length in these experiments. Random replacement independently samples $q_c$ from the uniform distribution $\mathcal{U}(0, 1)$ for each channel and clips it to $[10^{-6}, 1 - 10^{-6}]$. The replacement delay is $l'_c = 1 + (L - 2)q_c$, measured in observation time steps. The replacement uses random seed $20260829 + s$, where $s \in \{0, 1, 2\}$ is the model-training seed. Other parameters and test trajectories remain fixed.

Table 8: DS4D-FFNO delay interventions: test full-field nRMSE (mean ± sample SD, seeds 0–2). The intervention protocol is specified in Appendix D.2.

| Dimension | Benchmark | DS4D-FFNO | Permuted delays | Randomly replaced delays |
|---|---|---|---|---|
| 1D | KdV–KdV system | $\mathbf{(1.86 \pm 2.54) \times 10^{-2}}$ | $(2.68 \pm 0.30) \times 10^{-1}$ | $(2.30 \pm 0.08) \times 10^{-1}$ |
| | Microbeam system | $\mathbf{(1.72 \pm 1.93) \times 10^{-2}}$ | $(1.25 \pm 0.30) \times 10^{-1}$ | $(1.10 \pm 0.39) \times 10^{-1}$ |
| | Kawahara equation | $\mathbf{(1.06 \pm 0.40) \times 10^{-2}}$ | $(1.65 \pm 0.75) \times 10^{0}$ | $(1.48 \pm 0.70) \times 10^{0}$ |
| 2D | Thermoelastic system | $\mathbf{(8.55 \pm 0.16) \times 10^{-3}}$ | $(6.28 \pm 2.18) \times 10^{-2}$ | $(6.63 \pm 2.88) \times 10^{-2}$ |
| | Delayed linear problem | $\mathbf{(1.53 \pm 0.10) \times 10^{-2}}$ | $(1.19 \pm 0.43) \times 10^{-1}$ | $(1.95 \pm 0.90) \times 10^{-1}$ |
| | Reaction–diffusion model | $\mathbf{(1.10 \pm 0.03) \times 10^{-2}}$ | $(4.66 \pm 1.26) \times 10^{-2}$ | $(5.82 \pm 2.80) \times 10^{-2}$ |

### D.3 Resolution and Parameter Sensitivity

**Spatial resolution.** Figure 7 compares Microbeam system at $N_x \in \{25, 33, 65, 96\}$ and Delayed linear problem at $N_x = N_y \in \{14, 32, 64, 96\}$. At every tested resolution of both equations, DS4D-FFNO achieves the lowest mean test full-field nRMSE among the eight models, with means computed over three model seeds.

The spatial-resolution experiments in Figure 7 retain the model widths, observation and forecast lengths, physical parameters, and data-generation seeds used in the main experiments. Microbeam system evaluates the same eight-mode trajectories on each grid. Its Fourier cutoff is $m_s = 17$ for $N_x \geq 33$ and 13 for $N_x = 25$, compared with 33 in the main experiment. Delayed linear problem retains seven modes per axis and the same coefficient histories, normalizing each harmonic basis to unit RMS on the current disk grid.

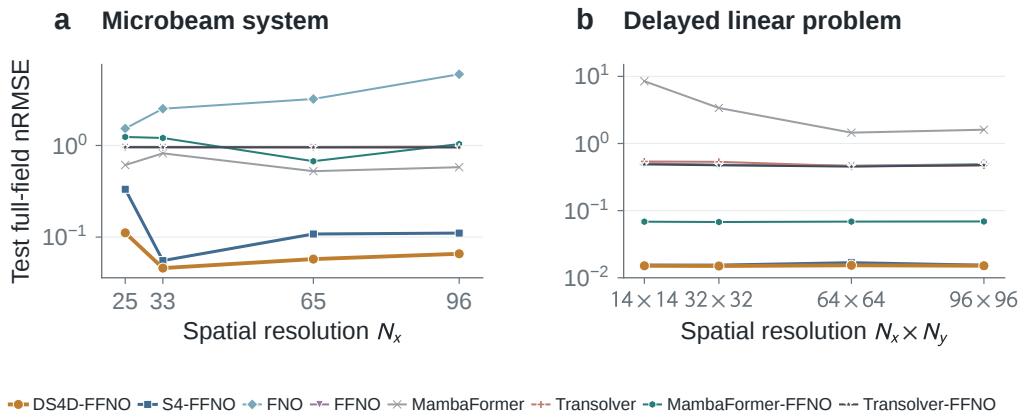


Figure 7: Spatial-resolution sensitivity. **a**, Microbeam system with a 24-step forecast horizon. **b**, Delayed linear problem with a 12-step forecast horizon. Points show test full-field nRMSE averaged over three model seeds. Both vertical axes are logarithmic. Each model is evaluated on the grid used for training.

**Model parameters.** Tables 9 and 10 vary one factor at a time with unchanged data and optimization. Reference settings reuse the main checkpoints. Other settings are trained and validation-selected separately.

Table 9: DS4D-FFNO parameter sweep on the Microbeam system.

| Factor | Value | #Params | Full-field nRMSE ↓ |
|---|---|---|---|
| Initial allocation $\rho_{cr}^{(0)}$ | $3 \times 10^{-4}$ | 308,514 | $(2.72 \pm 3.48) \times 10^{-2}$ |
| | $0.003^{\dagger}$ | 308,514 | $\mathbf{(1.72 \pm 1.93) \times 10^{-2}}$ |
| | 0.03 | 308,514 | $(5.95 \pm 5.47) \times 10^{-2}$ |
| Hidden width $C$ | 16 | 77,970 | $(9.16 \pm 6.97) \times 10^{-2}$ |
| | $32^{\dagger}$ | 308,514 | $\mathbf{(1.72 \pm 1.93) \times 10^{-2}}$ |
| | 48 | 691,634 | $(5.15 \pm 4.52) \times 10^{-2}$ |
| State size $N$ | 8 | 306,978 | $(1.17 \pm 0.56) \times 10^{0}$ |
| | $24^{\dagger}$ | 308,514 | $(1.72 \pm 1.93) \times 10^{-2}$ |
| | 48 | 310,818 | $\mathbf{(7.97 \pm 2.68) \times 10^{-3}}$ |
| Fourier coeffs. | 17 | 111,906 | $(3.64 \pm 3.00) \times 10^{-1}$ |
| | 49 | 242,978 | $(1.76 \pm 1.01) \times 10^{-1}$ |
| | $65^{\dagger}$ | 308,514 | $\mathbf{(1.72 \pm 1.93) \times 10^{-2}}$ |

† denotes the configuration used for the experiments in the main text.

Table 10: DS4D-FFNO parameter sweep on the Delayed linear problem.

| Factor | Value | #Params | Full-field nRMSE ↓ |
|---|---|---|---|
| Initial allocation $\rho_{cr}^{(0)}$ | $3 \times 10^{-4}$ | 152,801 | $(1.63 \pm 0.18) \times 10^{-2}$ |
| | $0.003^{\dagger}$ | 152,801 | $\mathbf{(1.53 \pm 0.10) \times 10^{-2}}$ |
| | 0.03 | 152,801 | $(1.72 \pm 0.31) \times 10^{-2}$ |
| Hidden width $C$ | 16 | 39,025 | $(2.21 \pm 0.93) \times 10^{-2}$ |
| | $32^{\dagger}$ | 152,801 | $(1.53 \pm 0.10) \times 10^{-2}$ |
| | 48 | 341,329 | $\mathbf{(1.09 \pm 0.18) \times 10^{-2}}$ |
| State size $N$ | 8 | 151,265 | $(1.79 \pm 0.29) \times 10^{-2}$ |
| | $24^{\dagger}$ | 152,801 | $(1.53 \pm 0.10) \times 10^{-2}$ |
| | 48 | 155,105 | $\mathbf{(1.32 \pm 0.48) \times 10^{-2}}$ |
| Spatial modes $m_s$ | 3 | 87,265 | $\mathbf{(1.24 \pm 0.30) \times 10^{-2}}$ |
| | $7^{\dagger}$ | 152,801 | $(1.53 \pm 0.10) \times 10^{-2}$ |
| | 15 | 283,873 | $(1.47 \pm 0.12) \times 10^{-2}$ |

† denotes the configuration used for the experiments in the main text.

# E Experimental Settings for PDEs Without Physical Delay

**Shared settings.** The 1D problems use $N_x = 65$, $L = 192$, $H = 24$, 33 spatial modes, a 384/96/96 train/validation/test split, and 120 training epochs. The 2D problems use a $64 \times 64$ grid, $L = 96$, $H = 12$, seven modes per axis, a 192/48/64 split, and 60 epochs.

Both architectures use $C = 32$, $N = 24$, and identical initial values for shared parameters. DS4D-FFNO retains the initialization in Appendix B. Training and evaluation follow Appendix A.

**Data construction.** All problems use periodic boundaries. Independent trajectories are recorded from $t = 0$. Only the 1D representation repeats the periodic endpoint. Data-generation seeds are 20260827 in 1D and 20260908 in 2D.

**One-dimensional problems.** We predict the scalar field $u$ on $[0, \ell_x)$:

$$\begin{aligned} \text{Burgers:} \quad & u_t + \tfrac{1}{2}\partial_x(u^2) = 0.01 u_{xx}, \\ \text{Advection–diffusion:} \quad & u_t + u_x = 0.01 u_{xx}, \\ \text{Allen–Cahn:} \quad & u_t = 0.01 u_{xx} + u - u^3. \end{aligned} \tag{20}$$

Initial fields are $u_0 = A\widetilde{S}_J + \mu$, where $\widetilde{S}_J$ is constructed by summing sine and cosine modes at frequencies $2\pi k/\ell_x$, $k = 1, \ldots, J$, with independent $\mathcal{N}(0, k^{-3})$ coefficients, then centering and

normalizing the sum to unit grid RMS. The amplitude $A$ and mean $\mu$ are sampled independently and uniformly from the ranges in Table 11.

Table 11: Initial-field and recording settings for the 1D PDEs without physical delay.

| Equation | $\ell_x$ | $J$ | $A$ range | $\mu$ range | $\Delta T$ |
|---|---|---|---|---|---|
| Burgers | 1 | 6 | $[0.45, 0.75]$ | $[-0.20, 0.20]$ | 0.001 |
| Advection–diffusion | $2\pi$ | 8 | $[0.45, 0.85]$ | $[-0.20, 0.20]$ | 0.005 |
| Allen–Cahn | $2\pi$ | 8 | $[0.35, 0.75]$ | $[-0.12, 0.12]$ | 0.005 |

Advection–diffusion uses exact Fourier propagation. Burgers and Allen–Cahn use Fourier pseudospectral derivatives with two-thirds dealiasing and fourth-order Runge–Kutta integration, with two and four steps per recording interval, respectively.

**Sampling two-dimensional initial fields.** Navier–Stokes, Gray–Scott, and nonlinear shallow water use smooth random fields $G_\sigma$ to construct their initial conditions. The respective constructions are given in Equations (22), (24), and (26). To sample $G_\sigma$, we draw independent standard normal noise at each grid point and multiply its Fourier coefficients by $\exp[-(k_x^2+k_y^2)/(2\sigma^2)]$. Here $k_x, k_y$ are integer frequency indices, and $\sigma$ controls the filter's frequency range. After the inverse Fourier transform, we subtract the field's grid mean and divide by its grid sample standard deviation to obtain $G_\sigma$. Different superscripts denote fields sampled independently using this procedure.

**Navier–Stokes.** On $[0,1)^2$, we predict vorticity $\omega$:

$$\begin{aligned} \omega_t + \boldsymbol{v}\cdot\nabla\omega = 10^{-3}\Delta\omega + 0.1\big[\sin(2\pi(x+y)) + \cos(2\pi(x+y))\big], \\ \Delta\psi = \omega, \qquad \boldsymbol{v} = (\partial_y\psi, -\partial_x\psi). \end{aligned} \tag{21}$$

The initial condition is

$$\omega_0 = G_5. \tag{22}$$

Fourier pseudospectral advection uses two-thirds dealiasing. We treat advection and forcing explicitly and diffusion with Crank–Nicolson, taking ten steps per recording interval $\Delta T = 0.02$.

**Gray–Scott.** We predict $(u, v)$ on $[0, 64)^2$:

$$u_t = 0.16\Delta u - uv^2 + 0.035(1-u), \qquad v_t = 0.08\Delta v + uv^2 - (0.035+0.062)v. \tag{23}$$

The initial fields are

$$u_0 = 1 - 0.50P + 0.015G_8^{(1)}, \qquad v_0 = 0.25P + 0.010G_8^{(2)}, \tag{24}$$

where $P = \exp[-(d_x^2+d_y^2)/(2w^2)]$ is a periodic Gaussian patch. Its center is sampled uniformly over the domain, $d_x, d_y$ are periodic distances to that center, and $w \sim \mathcal{U}(3,6)$, independently of the noise. We use a five-point Laplacian with unit grid spacing and forward Euler integration, taking two steps per recording interval $\Delta T = 1$.

**Nonlinear shallow water.** On $[0, 2\pi)^2$, depth $h$ and velocity $\boldsymbol{v} = (v_x, v_y)$ satisfy

$$\begin{aligned} h_t + \nabla\cdot(h\boldsymbol{v}) = 0, \\ \boldsymbol{v}_t + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla h = 0.002\Delta\boldsymbol{v}. \end{aligned} \tag{25}$$

Gravity and mean depth are one. Initial fields are

$$h_0 = 1 + 0.08G_4^{(1)}, \qquad v_{x,0} = 0.03G_4^{(2)}, \qquad v_{y,0} = 0.03G_4^{(3)}. \tag{26}$$

We predict $(h-1, v_x, v_y)$. We use second-order centered differences and Heun integration, taking four steps per recording interval $\Delta T = 0.02$.