OPEN-LOOP, CLOSED-FORM: SOLVER-DRIVEN RESERVOIR COMPUTING FOR COARSE-GRID PDE FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid numerical–learning models can improve coarse-grid PDE forecasts, but training corrections through coupled rollouts is computationally expensive. We propose an open-loop solver–reservoir framework with closed-form readout training. A numerical solver initialized solely from the coarse initial state generates a base trajectory. A fixed reservoir transforms this trajectory into history-dependent features, and a linear Fourier readout, shared across forecast times, predicts the downsampled high-fidelity states. Training reduces to ridge regression independently across Fourier modes, without differentiation through the solver or reservoir. Learned predictions never enter either state update, so readout errors cannot alter subsequent features. We analyze the guarantees and limitations of this design: feature consistency for a fixed construction under a common train–test trajectory law, a prediction-risk decomposition, and an expected-risk bound using independent trajectories as the sampling units. Across five configurations of the Kuramoto–Sivashinsky, Burgers, and Navier–Stokes equations, our method reduces test nRMSE by 31–88% relative to the uncorrected solver and achieves the lowest error among the compared methods in three configurations. Measured training times, including validation-based model selection, correspond to speedups of 4.0–38.9$\times$ over solver-in-the-loop hybrids with FFNO and S4FFNO correctors under the evaluated protocols, although the best such hybrid remains more accurate in the other two configurations. Our method also outperforms the evaluated closed-loop hybrid reservoir baseline in all four shared configurations.

1 INTRODUCTION

Hybrid solver–neural models combine a numerical solver with a learned component to improve the accuracy of coarse-resolution PDE simulations (Bar-Sinai et al., 2019; Kochkov et al., 2021; Um et al., 2020). The solver advances the known physical equations, while the learned component accounts for discrepancies relative to reference trajectories. This can reduce the cost of accurate simulation, but training the hybrid model can itself be expensive. In solver-in-the-loop (SITL) learning (Um et al., 2020), each optimization iteration evaluates multiple coupled solver–network steps and propagates gradients through the rollout. Longer unrolls expose the model to the effects of its predictions on later states, while increasing computation and requiring activation storage or recomputation.

Temporal memory introduces a further consideration. Spatial coarse-graining can hide variables that influence the future resolved state, so coarse observations need not form a Markovian system. The Mori–Zwanzig formalism describes how eliminating unresolved variables produces history-dependent terms (Mori, 1965; Zwanzig, 1961), and memory-augmented neural operators can improve forecasting from low-resolution observations (Buitrago Ruiz et al., 2025). However, incorporating a memory backbone into a gradient-trained hybrid model adds temporal computations and may increase training time and activation storage, depending on the architecture and implementation. The challenge is therefore to obtain useful temporal features while keeping hybrid training inexpensive.

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.

Reservoir computing (RC) is well suited to this objective: a fixed recurrent network transforms an input history into nonlinear features, and only a linear readout is learned (Jaeger, 2001; Lukoševičius & Jaeger, 2009). With a squared loss and ridge regularization, the readout has a closed-form solution (Hoerl & Kennard, 1970). RC thus provides recurrent memory without optimizing the recurrent dynamics through backpropagation through time.

We propose a two-stage hybrid framework that uses these properties for coarse-grid PDE forecasting. A numerical solver first generates a base trajectory from the observed coarse initial state. This trajectory drives a fixed reservoir, and a shared Fourier readout maps its features to the target coarse-grid states. Training consists of forward feature construction and ridge regression, with no differentiation through the solver or reservoir. The coupling is open loop: learned predictions never enter either state update. Consequently, training and deployment use the same solver-driven feature construction, and readout errors cannot alter subsequent features. Errors in the base trajectory can still affect prediction accuracy.

Our contributions are as follows:

- We formulate an open-loop solver–reservoir hybrid for coarse-grid PDE forecasting in which the only learned component, a shared Fourier readout, is fitted in closed form.
- We analyze the structural properties and limitations of the open-loop design. Under a common trajectory law and a fixed feature construction, we establish training–deployment feature consistency, decompose prediction risk into observation uncertainty, representation error, and readout approximation error, and derive an expected-risk bound that treats independent trajectories as the sampling units.
- On the Kuramoto–Sivashinsky, Burgers, and Navier–Stokes equations, we compare with SITL hybrids (Um et al., 2020) using FFNO (Tran et al., 2023) and S4FFNO (Buitrago Ruiz et al., 2025) backbones. Our method attains the lowest reported error in three of five settings with training speedups of 4.0–38.9$\times$ under the evaluated protocols. We also compare with closed-loop hybrid RC on KS and Burgers and quantify the benefit of reservoir memory on KS.

2 Related Work

Hybrid numerical solvers. Hybrid solvers combine numerical discretizations with learned components, including learned discretizations (Bar-Sinai et al., 2019; Kochkov et al., 2021) and data-driven closures (List et al., 2022; Frezat et al., 2022). Beyond conventional SITL training (Um et al., 2020), indirect and proximal correction schemes modify how learned components interact with the governing equations or numerical integrators (Wei et al., 2025; Mezidi et al., 2026). Neural operators (Li et al., 2021) can serve as correction backbones, including the Markovian FFNO (Tran et al., 2023) and the memory-augmented S4FFNO (Buitrago Ruiz et al., 2025). In contrast to gradient-trained corrections, our framework separates solver rollout from learning and fits only a linear readout in closed form.

Memory-based modeling of unresolved dynamics. Memory effects in coarse-grained and reduced-order models have been addressed through MZ-based closures (Parish & Duraisamy, 2017; Gouasmi et al., 2017) and learned closures based on recurrent neural networks (Ma et al., 2019) or neural delay differential equations (Gupta & Lermusiaux, 2021). MemNO (Buitrago Ruiz et al., 2025) incorporates temporal memory into neural operators through a state-space memory layer. Our approach instead uses a fixed reservoir to construct a nonlinear temporal representation of the coarse solver trajectory, without explicitly approximating the MZ memory kernel or training the recurrent dynamics.

Reservoir computing and hybrid forecasting. Reservoir computing models temporal dependencies through fixed recurrent dynamics and a trainable linear readout (Jaeger, 2001; Lukoševičius & Jaeger, 2009). It has been applied to chaotic dynamical systems and spatiotemporal PDE forecasting (Pathak et al., 2018a; Vlachas et al., 2020; Chattopadhyay et al., 2020; Platt et al., 2022). Hybrid variants combine reservoirs with imperfect knowledge-based models (Pathak et al., 2018b; Wikner et al., 2020; Arcomano et al., 2022), commonly operating in closed loop by feeding hybrid predictions back into the coupled system. We refer to the Wikner-inspired closed-loop hybrid reservoir

baseline as CL–HRC. Our framework instead uses an independently generated solver trajectory to drive the reservoir in open loop, without feeding learned predictions back into either component.

3 PRELIMINARIES

Coarse-grid PDE forecasting. We consider scalar, time-dependent PDEs on the periodic domain $\Omega = (0, L)^d$:

$$\partial_t u(t, x) = \mathcal{L}[u](t, x), \qquad u(0, x) = u_0(x), \qquad (t, x) \in (0, T] \times \Omega, \tag{1}$$

where $\mathcal{L}$ is a known, possibly nonlinear spatial operator that may include fixed spatial forcing. Following the low-resolution setting of MemNO (Buitrago Ruiz et al., 2025), high-resolution reference states $u_n^H \in \mathbb{R}^{f_H^d}$ are downsampled onto an equispaced grid $\mathcal{S}$ with $f < f_H$ points per dimension. Write $P := |\mathcal{S}| = f^d$ and

$$u_n := \mathcal{P}_f(u_n^H) \in \mathbb{R}^P, \qquad t_n = n\Delta t, \qquad n = 0, \dots, N, \tag{2}$$

where $\mathcal{P}_f$ is the spatial downsampling operator and $N := N_t - 1$ for N_t snapshots including the initial condition. The training sample is $S = (Z_1, \dots, Z_M)$, with complete trajectories $Z_i = (u_0^{(i)}, \dots, u_N^{(i)})$. At deployment, only a new coarse initial state u_0 is observed; the task is to predict $u_1, \dots, u_N$ without future reference observations or high-resolution reference states. We measure spatial error using $\|v\|_{2,P} := P^{-1/2}\|v\|_2$.

Memory in coarse dynamics. Eliminating unresolved variables can make the resolved evolution depend on its history. The Mori–Zwanzig (MZ) formalism separates instantaneous resolved dynamics, a memory term, and a contribution from unresolved initial variables (Mori, 1965; Zwanzig, 1961). For nonlinear PDEs, this decomposition is expressed through observables and the Liouville operator (Chorin et al., 2002); Appendix A gives the linear case and its scope. This motivates temporal representations in our model. The history we use is generated by a solver initialized from u_0, so it supplies features rather than additional observations of unresolved variables. Section 5 formalizes the resulting information and representation limits.

4 METHOD: TWO-STAGE SOLVER–RESERVOIR LEARNING

Our predictor consists of a fixed numerical solver, a fixed reservoir, and a trainable Fourier readout. The solver supplies the reservoir inputs throughout the rollout; learned predictions do not enter either state update.

4.1 STAGE 1: NUMERICAL BASE TRAJECTORY

Let z_n be the internal solver state, $\mathcal{E}$ a deterministic lifting from the observation grid, and $\mathcal{C}$ the observation map back to $\mathcal{S}$. A fixed update $\widetilde{\Psi}$ advances the solver over one observation interval Δt:

$$z_0 = \mathcal{E}(u_0), \qquad z_n = \widetilde{\Psi}(z_{n-1}), \qquad \bar{u}_n = \mathcal{C}(z_n), \qquad n = 1, \dots, N, \tag{3}$$

with $\mathcal{C}\mathcal{E} = I$ and $\bar{u}_0 = u_0$. For a solver on the observation grid, $\mathcal{E} = \mathcal{C} = I$ and $\widetilde{\Psi} = \Psi$, giving $\bar{u}_n = \Psi(\bar{u}_{n-1})$. The internal-state formulation also covers our KS implementation, which retains a finer solver grid between observations (Appendix B). In every case, the base trajectory is determined by u_0 without access to future reference states.

4.2 STAGE 2: RESERVOIR FEATURES AND FOURIER READOUT

Local recurrent features. At position p, define the periodic stencil

$$s_{n,\mathrm{loc}}^{(p)} := \left(\bar{u}_n^{(p+\delta)}\right)_{\delta \in \{-m,\dots,m\}^d} \in \mathbb{R}^{(2m+1)^d}, \tag{4}$$

where m is the stencil radius. The actual input is $s_n^{(p)} = s_{n,\mathrm{loc}}^{(p)}$ unless position-dependent forcing requires fixed coordinates $\xi^{(p)} \in \mathbb{R}^d$, in which case

$$s_n^{(p)} = [s_{n,\mathrm{loc}}^{(p)};\, \xi^{(p)}]. \tag{5}$$

Its dimension is denoted by d_{in}. Starting from $h_0^{(p)} = 0$, a leaky echo state network updates

$$h_n^{(p)} = (I - A)h_{n-1}^{(p)} + A\tanh\Big(Wh_{n-1}^{(p)} + \sigma_{\mathrm{in}}W_{\mathrm{in}}s_n^{(p)}\Big), \tag{6}$$

where $h_n^{(p)} \in \mathbb{R}^D$ and $A = \mathrm{diag}(\alpha_1, \ldots, \alpha_D)$, with $0 < \alpha_a \leq 1$. Channel-dependent leak rates provide different response rates (Tanaka et al., 2022); their memory also depends on the recurrent coupling. The matrices $W \in \mathbb{R}^{D\times D}$ and $W_{\mathrm{in}} \in \mathbb{R}^{D\times d_{\mathrm{in}}}$ are randomly initialized and fixed. We rescale W to a prescribed spectral radius; $\sigma_{\mathrm{in}} > 0$ controls input scaling. All reservoir parameters are shared across positions and times. We use $r_n^{(p)} = \psi(h_n^{(p)})$, with

$$r_{n,a}^{(p)} = \begin{cases} h_{n,a}^{(p)}, & a \text{ odd}, \\ (h_{n,a}^{(p)})^2, & a \text{ even}, \end{cases} \qquad r_n \in \mathbb{R}^{P\times D}. \tag{7}$$

Thus $r_n = \mathcal{R}_n(u_0)$ is a deterministic, history-dependent feature field for a fixed solver and reservoir.

Shared Fourier readout. Let $\mathcal{F}$ be the unitary spatial DFT and $\widehat{r}_{n,k} := (\mathcal{F}r_n)_k \in \mathbb{C}^D$. The prediction is

$$\hat{u}_n = \mathcal{F}^{-1}\left((w_k^{\mathsf{H}}\widehat{r}_{n,k})_k\right) =: \mathcal{T}_{\boldsymbol{w}}r_n, \tag{8}$$

where $w_k \in \mathbb{C}^D$, $^{\mathsf{H}}$ denotes conjugate transpose, and $\boldsymbol{w} = (w_k)_k$ is shared over all forecast times. The readout predicts the target state directly. We impose $w_{-k} = \overline{w_k}$, with mode indices interpreted modulo the grid, to obtain real-valued predictions. Let $\mathcal{W}$ denote this constrained readout space, with $\|\boldsymbol{w}\|_{\mathcal{W}}^2 := \sum_k \|w_k\|_2^2$. Without coordinates, periodic stencils and shared weights make Stage 2 translation equivariant; equivariance of the complete predictor also requires an equivariant solver-to-observation map. Coordinates allow responses to depend on absolute position.

4.3 Closed-Form Training and Deployment

For $\lambda > 0$, we fit the readout by minimizing the mean trajectory loss with a ridge penalty:

$$\begin{aligned} \ell(\boldsymbol{w}; Z) &:= \frac{1}{N}\sum_{n=1}^{N} \|\hat{u}_n^{\boldsymbol{w}} - u_n\|_{2,P}^2, \\ \boldsymbol{w}_{S,\lambda} &:= \arg\min_{\boldsymbol{w}\in\mathcal{W}} \left\{ \frac{1}{M}\sum_{i=1}^{M} \ell(\boldsymbol{w}; Z_i) + \lambda\|\boldsymbol{w}\|_{\mathcal{W}}^2 \right\}. \end{aligned} \tag{9}$$

Parseval's identity separates this problem over Fourier modes. Define

$$x_{i,n,k} := P^{-1/2}\widehat{r}_{n,k}^{(i)}, \qquad y_{i,n,k} := P^{-1/2}(\mathcal{F}u_n^{(i)})_k.$$

The unique solution is

$$w_{S,\lambda,k} = \left(\frac{1}{MN}\sum_{i,n} x_{i,n,k}x_{i,n,k}^{\mathsf{H}} + \lambda I_D\right)^{-1}\left(\frac{1}{MN}\sum_{i,n} x_{i,n,k}\overline{y_{i,n,k}}\right). \tag{10}$$

Both sums run over all training trajectories and forecast steps. For real training fields, paired modes give conjugate weights; weights at self-conjugate modes are real. With the definitions of x and y above, a formulation using sums instead of averages requires the diagonal shift $MN\lambda I_D$. The value of an absolute ridge coefficient therefore depends on the normalization of both the Fourier transform and the regression statistics.

For each reservoir configuration, training requires forward feature construction, accumulation of regression statistics, and independent $D \times D$ linear solves, with no backpropagation through the solver or reservoir. Base trajectories can be reused across reservoirs, and regression statistics across ridge coefficients; hyperparameter selection is separate from each fit. At deployment, a new u_0 drives the same solver and reservoir updates, and the fitted readout produces forecasts without feedback.

5 Analysis of the Open-Loop Design

We analyze the finite-horizon predictor using the notation of Sections 3–4. The results identify structural properties and conditional guarantees; they do not establish an accuracy advantage over a feedback-based model. Throughout, complete trajectories $Z = (u_0, \dots, u_N)$ are sampled independently from a common distribution $\mathbb{P}$. The solver, reservoir, and all feature preprocessing are measurable and fixed independently of these samples. The solver rollout is assumed to be well-defined and finite almost surely through step N. For a randomly initialized reservoir, the statements are conditional on its fixed realization. The readout in this section acts on $r_n = \psi(h_n)$; additional unbounded input channels require separate assumptions and corresponding changes to the bounds. All proofs are in Appendix F.

Exogenous features and readout decoupling. The fixed solver rollout, including any internal solver state, is a deterministic function of u_0. Neither this rollout nor the reservoir depends on the fitted readout.

Proposition 1 (Feature consistency and readout sensitivity)**.** *For every $n = 1, \dots, N$,*

$$(\mathcal{R}_n(u_0^{\mathrm{train}}), u_n^{\mathrm{train}}) \overset{d}{=} (\mathcal{R}_n(u_0^{\mathrm{test}}), u_n^{\mathrm{test}}). \tag{11}$$

Moreover, for two readouts applied to the same initial condition,

$$\|\hat{u}_n^{\boldsymbol{w}} - \hat{u}_n^{\boldsymbol{v}}\|_{2,P} \le \sqrt{D} \max_k \|w_k - v_k\|_2, \qquad n = 1, \dots, N. \tag{12}$$

Thus readout perturbations do not alter subsequent features or acquire an autoregressive amplification factor. This does not bound error against the reference solution or prevent errors in the numerical solver. Equation (11) concerns population laws, not equality of finite-sample errors. It requires a common trajectory law; equality of coarse initial-state marginals alone is insufficient.

Information and representation limits. Let J be uniform on $\{1, \dots, N\}$ and independent of Z. Set $X = r_J$, $Y = u_J$, $m_0 = \mathbb{E}[Y \mid u_0, J]$, and $m_r = \mathbb{E}[Y \mid X]$. Define $\mathcal{E}_{\mathrm{obs}} := \mathbb{E}\|Y - m_0\|_{2,P}^2$ and $\mathcal{E}_{\mathrm{rep}} := \mathbb{E}\|m_0 - m_r\|_{2,P}^2$.

Proposition 2 (Prediction-risk decomposition)**.** *If $\mathbb{E}\|Y\|_{2,P}^2 < \infty$, the population trajectory risk satisfies*

$$\mathcal{R}(\boldsymbol{w}) := \mathbb{E}\|Y - \mathcal{T}_{\boldsymbol{w}} X\|_{2,P}^2 = \mathcal{E}_{\mathrm{obs}} + \mathcal{E}_{\mathrm{rep}} + \mathbb{E}\|m_r - \mathcal{T}_{\boldsymbol{w}} X\|_{2,P}^2. \tag{13}$$

The terms quantify uncertainty remaining after observing (u_0, J), information lost in the reservoir representation, and mismatch with the shared Fourier readout class. The first term can be nonzero even for a deterministic PDE because coarse-graining can hide relevant initial scales. It can also vanish: for example, if the observation operator is injective on the band-limited KS initial-data subspace in Appendix B.1, the initial reference field and its deterministic future are determined by u_0. A deterministic solver history adds no observations beyond u_0; its role is to construct useful features. Reservoir memory can improve this representation, but does not by itself guarantee recovery of unresolved dynamics or approximation of the MZ memory kernel.

Conditional stability and fading memory. For two reservoir input sequences, define the maximum input difference

$$\eta_n := \max_p \|s_n^{(p)} - s_n'^{(p)}\|_\infty.$$

The reservoir constants are

$$q := \max_a \left(1 - \alpha_a + \alpha_a \sum_b |W_{ab}|\right), \qquad b_{\mathrm{in}} := \sigma_{\mathrm{in}} \max_a \alpha_a \sum_b |(W_{\mathrm{in}})_{ab}|. \tag{14}$$

For coordinate-augmented inputs, s_n denotes the full input vector.

Proposition 3 (Conditional input-to-output stability)**.** *For zero-initialized reservoirs with the same fixed parameters and readout, let $C_{\boldsymbol{w}} := \max_k \|w_k\|_2$. Then*

$$\|\hat{u}_n - \hat{u}_n'\|_{2,P} \le 2\sqrt{D}\, C_{\boldsymbol{w}} b_{\mathrm{in}} \sum_{j=1}^{n} q^{n-j} \eta_j. \tag{15}$$

If $q < 1$ and $\sup_j \eta_j \le \varepsilon$, the right-hand side is at most $2\sqrt{D}\, C_{\boldsymbol{w}} b_{\mathrm{in}} \varepsilon/(1-q)$.

A perturbation confined to step j therefore has an effect bounded by a multiple of q^{n-j} when $q < 1$. This sufficient contraction condition must be distinguished from spectral-radius scaling: $\rho(W) < 1$ alone does not establish it (Yildiz et al., 2012). The result controls sensitivity to supplied solver trajectories, not stability of the solver or accuracy over arbitrary horizons.

Learning from independent trajectories. We consider the ridge estimator $\boldsymbol{w}_{S,\lambda}$ defined in (9). The following specialization of regularized learning stability (Bousquet & Elisseeff, 2002) treats each entire trajectory as one sample.

Theorem 1 (Expected-risk bound for the ridge readout)**.** *Suppose $V_u := \mathbb{E}\|Y\|_{2,P}^2 < \infty$. Under the fixed-feature assumptions above,*

$$\mathbb{E}_S \mathcal{R}(\boldsymbol{w}_{S,\lambda}) \le \inf_{\boldsymbol{w}\in\mathcal{W}} \left\{ \mathcal{R}(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_{\mathcal{W}}^2 \right\} + \frac{4DV_u}{\lambda M} \left(1 + \sqrt{\frac{D}{\lambda}} \right)^2 . \tag{16}$$

The estimation term in (16) is numerically loose: for the reservoir dimensions, trajectory counts, and ridge grid used in Section 6 and Appendix C.2, its coefficient of V_u exceeds 10^3. The bound therefore does not improve upon the zero-readout risk V_u. We use it for its structure rather than as a numerical prediction of test error. It separates regularized approximation from estimation, permits dependence within each trajectory, and requires no reservoir contraction. Here M counts independent trajectories; MN snapshots or MNP grid values cannot be counted as independent samples without additional assumptions. The bound concerns mean squared trajectory risk for a fixed reservoir and a fixed absolute regularization coefficient. It does not directly bound mean test nRMSE, validation-selected configurations, sample-dependent ridge shifts, or unseen forecast times. Appendix F.6 gives additional statements on memory, equivariance, and model selection.

6 Experiments

6.1 Datasets

We evaluate on three periodic PDE benchmarks. Each retained trajectory contains N_t snapshots: the first supplies u_0, and the remaining $N = N_t - 1$ are forecast targets. Reference data and numerical solvers are described in Appendix B.

Kuramoto–Sivashinsky (KS). We consider $\partial_t u + u\partial_x u + \partial_{xx} u + \nu \partial_{xxxx} u = 0$ on $\Omega = (0, 64)$, with $\nu = 0.1$. Following Buitrago Ruiz et al. (2025), reference trajectories are generated at $f_H = 512$ and observed at $f \in \{32, 64\}$. We retain $N_t = 26$ snapshots with $\Delta t = 0.1$ and use 2048/256/256 trajectories for training/validation/testing.

Burgers. We use the viscous Burgers dataset from PDEBench (Takamoto et al., 2022), with equation $\partial_t u + u\partial_x u = \nu \partial_{xx} u$ on the periodic domain $\Omega = (0, 1)$ and $\nu = 10^{-3}/\pi$. Here 10^{-3} is the PDEBench dataset parameter; the diffusion coefficient in our displayed PDE is that parameter divided by π. Reference states at $f_H = 1024$ are observed at $f \in \{32, 64\}$. Temporal subsampling with stride seven gives $\Delta t = 0.07$; each retained window contains $N_t = 21$ snapshots. We use 2048/256/1000 trajectories for training/validation/testing.

Navier–Stokes (NS). We consider two-dimensional incompressible flow in vorticity form, $\partial_t u + \mathbf{v} \cdot \nabla u = \nu \Delta u + F(x)$, where $\mathbf{v}$ is divergence-free and F is fixed spatial forcing. We use the dataset of Li et al. (2021) on $\Omega = (0, 1)^2$, with $\nu = 10^{-5}$, $f_H = 256$, and $f = 64$. We retain $N_t = 21$ snapshots with $\Delta t = 1.0$ and use 800/200/200 trajectories for training/validation/testing.

6.2 SITL: Gradient-Based Baseline

We implement solver-in-the-loop (SITL) learning (Um et al., 2020) with FFNO (Tran et al., 2023) or S4FFNO (Buitrago Ruiz et al., 2025) as the correction backbone. To include solvers with persistent internal states, write the coupling as

$$\begin{aligned} z_n^- &= \widetilde{\Psi}(z_{n-1}^{\mathrm{corr}}), & b_n &= \mathcal{C}(z_n^-), \\ \tilde{u}_n &= b_n + \mathcal{C}_\theta(\mathcal{B}_n), & z_n^{\mathrm{corr}} &= \mathcal{I}_{\mathrm{corr}}(z_n^-, \tilde{u}_n), \end{aligned} \tag{17}$$

with $z_0^{\mathrm{corr}} = \mathcal{E}(u_0)$ and $\tilde{u}_0 = u_0$. Here $\mathcal{I}_{\mathrm{corr}}$ inserts the corrected observation into the solver state, with $\mathcal{C}\mathcal{I}_{\mathrm{corr}}(z, v) = v$. For a solver on the observation grid, $\mathcal{I}_{\mathrm{corr}}(z, v) = v$, recovering $b_n = \Psi(\tilde{u}_{n-1})$. The FFNO corrector uses $\mathcal{B}_n = b_n$. At deployment, S4FFNO uses $\mathcal{B}_n = b_{1:n}$ for KS and Burgers, and $\mathcal{B}_n = b_{\max(1,n-H+1):n}$ for NS, with memory window $H = 4$. Training backpropagates through U consecutive solver–corrector steps, with $U = 4$ for all experiments. For a mini-batch of trajectory indices $\mathcal{I}$ and a sampled starting index n_0, we minimize

$$\mathcal{L}_{\mathrm{SITL}}(\theta) = \frac{1}{|\mathcal{I}|U} \sum_{i \in \mathcal{I}} \sum_{j=1}^{U} \left\| \tilde{u}^{(i)}_{n_0+j} - u^{(i)}_{n_0+j} \right\|^2_{2,P}. \tag{18}$$

Optimization uses Adam (Kingma & Ba, 2014), with checkpoint selection and early stopping based on full-horizon validation MSE. At inference, corrected states are fed back into the solver for all N forecast steps.

6.3 CL–HRC: Closed-Loop Hybrid Reservoir Computing

We implement a closed-loop hybrid reservoir computing (CL–HRC) baseline inspired by Wikner et al. (2020). It combines a fixed numerical solver, a fixed local reservoir, and a shared Fourier ridge readout, with hybrid predictions fed back into both the solver and reservoir during forecasting.

Training uses teacher forcing. The reference history supplies the solver input, and the preceding reference state drives the reservoir:

$$b_n^{\mathrm{TF}} = \Psi_{\mathrm{TF}}(u_{0:n-1}), \qquad h_n^{\mathrm{TF}} = \mathcal{H}(h_{n-1}^{\mathrm{TF}}, u_{n-1}), \tag{19}$$

where $\mathcal{H}$ denotes the local reservoir update and Ψ_{TF} includes any persistent internal solver state. The readout is fitted to u_n from the concatenated feature field

$$r_n^{\mathrm{CL}} = [\psi(h_n^{\mathrm{TF}}); b_n^{\mathrm{TF}}], \tag{20}$$

using the mode-wise Fourier ridge regression of Section 4.3, with input dimension $D + 1$ instead of D. Only the readout is learned.

At inference, only u_0 is observed. Starting from $\hat{u}_0 = u_0$ and $h_0 = 0$, the model forecasts recursively:

$$\begin{aligned} b_n &= \Psi_{\mathrm{CL}}(\hat{u}_{0:n-1}), \\ h_n &= \mathcal{H}(h_{n-1}, \hat{u}_{n-1}), \\ \hat{u}_n &= \mathcal{T}_{\boldsymbol{w}_{\mathrm{CL}}}\big([\psi(h_n); b_n]\big). \end{aligned} \tag{21}$$

For Burgers, $b_n = \Psi(\hat{u}_{n-1})$. For KS, the solver retains its internal 128-point spectral state: after each step, the difference between the hybrid prediction and projected solver output is injected into the resolved Fourier modes, while unresolved modes are retained.

Hyperparameters are selected using complete-horizon closed-loop validation MSE. Each candidate requires its own validation solver rollouts because its readout changes the feedback trajectory; our method can instead reuse fixed solver trajectories. To limit this model-selection cost, we evaluate CL–HRC on KS and Burgers only. CL–HRC therefore provides a closed-form hybrid baseline with autoregressive feedback on the four shared KS and Burgers configurations.

6.4 Evaluation Metric

We evaluate forecasting accuracy using the normalized root mean squared error (nRMSE). For trajectory i at forecast step n,

$$\mathrm{nRMSE}_{i,n} = \frac{\|\hat{u}_n^{(i)} - u_n^{(i)}\|_{2,P}}{\|u_n^{(i)}\|_{2,P} + \varepsilon}, \qquad \varepsilon = 10^{-12}. \tag{22}$$

The reported metric is $\mathrm{nRMSE}_{\mathrm{test}} = (M_{\mathrm{test}} N)^{-1} \sum_{i=1}^{M_{\mathrm{test}}} \sum_{n=1}^{N} \mathrm{nRMSE}_{i,n}$, excluding the initial condition. Thus, normalization and the square root are applied before averaging over trajectories and times. This metric differs from pooled relative RMSE and from the squared trajectory loss used in Theorem 1.

Table 1: Forecasting accuracy and training time on KS, Burgers, and NS. nRMSE is averaged over all test trajectories and forecast steps, excluding the initial condition. Times include validation-based model selection. Bold marks the lowest nRMSE and shortest training time within each configuration; — denotes no training.

Benchmark	Method	nRMSE ↓	Training time (s) ↓
KS ($f = 32$)	Solver-only	0.423	—
	SITL–FFNO	0.320	2537
	SITL–S4FFNO	**0.100**	5418
	CL–HRC	1.314	293
	Two-stage RC (ours)	0.127	**255**
KS ($f = 64$)	Solver-only	0.135	—
	SITL–FFNO	0.144	1971
	SITL–S4FFNO	0.049	11090
	CL–HRC	0.294	556
	Two-stage RC (ours)	**0.016**	**496**
Burgers ($f = 32$)	Solver-only	0.108	—
	SITL–FFNO	0.110	2565
	SITL–S4FFNO	0.107	8081
	CL–HRC	0.104	269
	Two-stage RC (ours)	**0.075**	**208**
Burgers ($f = 64$)	Solver-only	0.065	—
	SITL–FFNO	0.059	6061
	SITL–S4FFNO	0.061	8101
	CL–HRC	0.052	506
	Two-stage RC (ours)	**0.036**	**398**
NS ($f = 64$)	Solver-only	0.236	—
	SITL–FFNO	**0.111**	47195
	SITL–S4FFNO	0.121	73715
	Two-stage RC (ours)	0.131	**4931**

6.5 Results

Table 1 compares our two-stage RC framework with the coarse solver, the two SITL baselines, and CL–HRC. We report test nRMSE as defined in Section 6.4 and training time including validation-based model selection. CL–HRC is evaluated on KS and Burgers, as described in Section 6.3. All experiments use an NVIDIA RTX A4000 GPU. Both SITL and closed-form methods use validation MSE for model selection; test performance is reported using nRMSE.

Forecasting accuracy. Our method improves upon the coarse solver in all five configurations and achieves the lowest reported nRMSE in three: KS at $f = 64$ and Burgers at both resolutions. On KS at $f = 64$, its nRMSE of 0.016 is approximately 67% lower than the strongest alternative, SITL–S4FFNO. SITL retains an accuracy advantage on KS at $f = 32$ (0.100 versus our 0.127) and on NS (0.111 and 0.121 versus our 0.131).

Comparison with closed-loop RC. Our method outperforms CL–HRC in all four evaluated configurations. The gap is largest on KS, where CL–HRC also performs worse than the uncorrected solver at both resolutions. On Burgers, our method reduces nRMSE relative to CL–HRC by approximately 28% at $f = 32$ and 31% at $f = 64$. The accuracy gains therefore extend to a hybrid baseline that also uses reservoir memory and closed-form readout training. Because the two pipelines also differ in reservoir-driving signals and readout inputs, the gap cannot be attributed to feedback alone.

Training cost. Our method has the shortest reported training time in every configuration, with speedups of approximately 4.0–$38.9\times$ over the SITL baselines under the evaluated protocols. On NS, the speedups are $9.6\times$ over SITL–FFNO and $14.9\times$ over SITL–S4FFNO, at the cost of higher nRMSE. The two RC methods have comparable training costs on KS and Burgers, with our method requiring approximately 11–23% less time.

Ablation studies. Appendix D examines the role of reservoir memory by comparing the full reservoir with an instantaneous nonlinear feature map. Appendix E studies the effect of training-set size for the proposed readout, reporting test nRMSE and the absolute training–test MSE difference as the number of training trajectories varies from 256 to 2048.

7 DISCUSSION

Dependence on the base trajectory. Our method improves upon the uncorrected solver in all five configurations, while the best SITL baseline remains more accurate on KS at $f = 32$ and on NS. These results illustrate the capabilities and limitations of open-loop trajectory reconstruction. The readout can improve the predicted states without modifying the numerical trajectory that supplies its features. By contrast, a SITL corrector changes the state used in subsequent solver steps, which may help maintain a useful trajectory when the uncorrected solver drifts. However, the present comparisons do not isolate the effect of solver fidelity: the benchmarks also differ in dynamics, resolution, and model capacity. In particular, the NS reservoir has $D = 100$ channels, compared with $D = 1000$ for KS and Burgers. In the decomposition of Proposition 2, changes to the solver-driven representation affect $\mathcal{E}_{\mathrm{rep}}$ and the readout approximation term, while $\mathcal{E}_{\mathrm{obs}}$ is determined by the observation setting and data distribution.

Role of temporal memory. On KS at $f = 32$, removing reservoir memory increases mean test nRMSE by a factor of approximately 3.0 (Appendix D; learning curves for this setting are in Appendix E). Thus, processing the solver history provides more useful features for the shared readout than the evaluated instantaneous feature map. Because this history is a deterministic function of u_0, temporal processing supplies no additional observations of unresolved variables and cannot reduce $\mathcal{E}_{\mathrm{obs}}$; its benefit lies in the representation available to the readout.

Training cost and scalability. The measured 4.0–38.9$\times$ speedups over SITL show that forward feature construction and closed-form readout fitting can reduce training cost under the evaluated protocols. The 11–23% time reduction relative to CL–HRC is compatible with the additional candidate-dependent validation rollouts required by closed-loop coupling, but the measurements do not isolate that cost. These measurements depend on the architectures, stopping criteria, selection procedures, and implementation; they are not architecture-independent comparisons or measurements at matched accuracy. For the dense mode-wise ridge implementation, storing all regression statistics simultaneously requires $O(PD^2)$ memory, their direct accumulation requires $O(MNPD^2)$ operations, and solving the linear systems costs $O(PD^3)$ per ridge setting. These costs exclude solver rollout and feature generation. Processing modes in blocks can reduce peak statistics storage. These costs motivate low-rank approximations or parameter sharing across Fourier modes for larger spatial grids and reservoirs.

Limitations. The weighted contraction criterion in Appendix F.4 is satisfied among the selected reservoirs only for Burgers at $f = 64$, where $\rho(B) \approx 0.9991$. For the remaining configurations, this criterion does not certify uniform stability or fading memory, although the finite-horizon bound of Proposition 3 remains valid. Propositions 1– 2 and Theorem 1 do not require contraction. Theorem 1 concerns a fixed feature map and regularization choice; it does not directly characterize the configuration selected using validation data. Our experiments cover horizons of 20–25 forecast steps on periodic domains with fixed physical parameters and forcing. Generalization to longer horizons, non-periodic domains, and parameter or distribution shifts remains untested.

Conclusion. A fixed reservoir driven by a numerical base trajectory provides a practical route to learning coarse-grid PDE forecasts through closed-form regression. Across the evaluated configurations, the method consistently improves upon the uncorrected solver and offers a favorable accuracy–training-cost trade-off, while gradient-trained hybrids retain an accuracy advantage in two settings. Extending the method to broader operating conditions and developing more scalable readouts are natural directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on mathematical arguments, experimental methodology, and the interpretation of experimental results. We also used generative AI tools to edit and polish the manuscript, search for and identify related literature, and draft selected passages based on author-provided specifications. All AI-assisted content was reviewed by the authors. AI-generated feedback was treated as suggestions and independently evaluated before any changes were made, while AI-drafted passages were checked against the actual implementation, experimental protocol, and reported results. The authors take full responsibility for the final content of this work, including its text, claims, analyses, and results.

ETHICS STATEMENT

This study uses numerical PDE trajectories and publicly available benchmark data. It does not involve human participants or personal data.

REPRODUCIBILITY STATEMENT

Sections 3–4 specify the forecasting task, feature construction, and closed-form training objective. Section 6 describes the data splits, evaluation metric, and baseline protocols. Appendix B provides reference-data and numerical-solver details, and Appendices C and D describe the learned components, hyperparameter selection, and three-seed KS memory ablation. The assumptions and proofs of the analysis are given in Section 5 and Appendix F.

REFERENCES

Troy Arcomano, Istvan Szunyogh, Alexander Wikner, Jaideep Pathak, Brian R. Hunt, and Edward Ott. A hybrid approach to atmospheric modeling that combines machine learning with a physics-based numerical model. *Journal of Advances in Modeling Earth Systems*, 14(3):e2021MS002712, 2022. doi: 10.1029/2021MS002712.

Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019. doi: 10.1073/pnas.1814058116.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002. URL https://www.jmlr.org/papers/volume2/bousquet02a/bousquet02a.pdf.

Ricardo Buitrago Ruiz, Tanya Marwah, Albert Gu, and Andrej Risteski. On the benefits of memory for modeling time-dependent pdes. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 54972–55002, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/89379d5fc6eb34ff98488202fb52b9d0-Paper-Conference.pdf.

Ashesh Chattopadhyay, Pedram Hassanzadeh, and Devika Subramanian. Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics*, 27:373–389, 2020. doi: 10.5194/npg-27-373-2020.

Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman. Optimal prediction with memory. *Physica D: Nonlinear Phenomena*, 166(3–4):239–257, 2002. doi: 10.1016/S0167-2789(02)00446-3.

S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002. doi: 10.1006/jcph.2002.6995.

Hugo Frezat, Julien Le Sommer, Ronan Fablet, Guillaume Balarac, and Redouane Lguensat. A posteriori learning for quasi-geostrophic turbulence parametrization. *Journal of Advances in Modeling Earth Systems*, 14(11):e2022MS003124, 2022. doi: 10.1029/2022MS003124.

Lukas Gonon, Lyudmila Grigoryeva, and Juan-Pablo Ortega. Risk bounds for reservoir computing. *Journal of Machine Learning Research*, 21(240):1–61, 2020. URL https://www.jmlr.org/papers/v21/19-902.html.

Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001. doi: 10.1137/S003614450036757X.

Ayoub Gouasmi, Eric J. Parish, and Karthik Duraisamy. A priori estimation of memory effects in reduced-order models of nonlinear systems using the Mori–Zwanzig formalism. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170385, 2017. doi: 10.1098/rspa.2017.0385.

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*, 2022b.

Abhinav Gupta and Pierre F. J. Lermusiaux. Neural closure models for dynamical systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 477(2252):20201004, 2021. doi: 10.1098/rspa.2020.1004.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. doi: 10.1073/pnas.2101784118.

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. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Björn List, Li-Wei Chen, and Nils Thuerey. Learned turbulence modelling with differentiable fluid solvers: physics-based loss functions and optimisation horizons. *Journal of Fluid Mechanics*, 949:A25, 2022. doi: 10.1017/jfm.2022.738.

Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009. doi: 10.1016/j.cosrev.2009.03.005.

Chao Ma, Jianchun Wang, and Weinan E. Model reduction with memory and the machine learning of dynamical systems. *Communications in Computational Physics*, 25(4):947–962, 2019. doi: 10.4208/cicp.OA-2018-0269.

Abdel-Rahim Mezidi, Jordan Patracone, and Amaury Habrard. Proximal splitting methods for hybrid differentiable models. In *Proceedings of the International Conference on Machine Learning*, 2026.

Hazime Mori. Transport, collective motion, and Brownian motion. *Progress of Theoretical Physics*, 33(3):423–455, 1965. doi: 10.1143/PTP.33.423.

Eric J. Parish and Karthik Duraisamy. Non-Markovian closure models for large eddy simulations using the Mori–Zwanzig formalism. *Physical Review Fluids*, 2(1):014604, 2017. doi: 10.1103/PhysRevFluids.2.014604.

Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018a. doi: 10.1103/PhysRevLett.120.024102.

Jaideep Pathak, Alexander Wikner, Rebeckah Fussell, Sarthak Chandra, Brian R. Hunt, Michelle Girvan, and Edward Ott. Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(4):041101, 2018b. doi: 10.1063/1.5028373.

Jason A. Platt, Stephen G. Penny, Timothy A. Smith, Tse-Chun Chen, and Henry D. I. Abarbanel. A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics. *Neural Networks*, 153:530–552, 2022. doi: 10.1016/j.neunet.2022.06.025.

Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models—a review. *Monthly Weather Review*, 119(9):2206–2223, 1991. doi: 10.1175/1520-0493(1991)119⟨2206:SLISFA⟩2.0.CO;2.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. doi: 10.1137/0705041.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan 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, 2022.

Gouhei Tanaka, Takuma Matsumori, Hiroki Yoshida, and Kazuyuki Aihara. Reservoir computing with diverse timescales for prediction of multiscale dynamics. *Physical Review Research*, 4(3):L032014, 2022. doi: 10.1103/PhysRevResearch.4.L032014.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tmIiMPl4IPa.

Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Bram van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979. doi: 10.1016/0021-9991(79)90145-1.

P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020. doi: 10.1016/j.neunet.2020.02.016.

Hao Wei, Aleksandra Franz, Björn List, and Nils Thuerey. INC: An indirect neural corrector for auto-regressive hybrid PDE solvers. In *Advances in Neural Information Processing Systems*, volume 38, pp. 122167–122201, 2025. doi: 10.52202/085713-3681.

Alexander Wikner, Jaideep Pathak, Brian Hunt, Michelle Girvan, Troy Arcomano, Istvan Szunyogh, Andrew Pomerance, and Edward Ott. Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(5):053111, 2020. doi: 10.1063/5.0005541.

Izzet B. Yildiz, Herbert Jaeger, and Stefan J. Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, 2012. doi: 10.1016/j.neunet.2012.07.005. URL https://doi.org/10.1016/j.neunet.2012.07.005.

Robert Zwanzig. Memory effects in irreversible thermodynamics. *Physical Review*, 124(4):983–992, 1961. doi: 10.1103/PhysRev.124.983.

A Memory from Eliminating Unresolved Variables

The Mori–Zwanzig formalism motivates temporal representations for partially observed dynamics (Mori, 1965; Zwanzig, 1961). Here we give the linear case to distinguish this motivation from a claim that our reservoir identifies a memory kernel.

Write $u_t = u(t, \cdot)$ and $\mathcal{U} = L^2(\Omega; \mathbb{R})$. Let $\mathcal{P} : \mathcal{U} \to \mathcal{U}$ be a linear projection onto a resolved subspace and $\mathcal{Q} = I - \mathcal{P}$. This idealized projection need not equal the discrete observation map $\mathcal{P}_f$. For a linear operator $\mathcal{L}$, under the regularity needed for the following evolution operators, variation of constants in the unresolved equation gives

$$\mathcal{Q}u_t = e^{t\mathcal{Q}\mathcal{L}}\mathcal{Q}u_0 + \int_0^t e^{(t-s)\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\mathcal{P}u_s \, ds.$$

Substitution into the resolved equation yields

$$\partial_t \mathcal{P}u_t = \underbrace{\mathcal{P}\mathcal{L}\mathcal{P}u_t}_{\text{instantaneous}} + \underbrace{\int_0^t \mathcal{K}(t-s)\mathcal{P}u_s \, ds}_{\text{memory}} + \underbrace{\mathcal{P}\mathcal{L}e^{t\mathcal{Q}\mathcal{L}}\mathcal{Q}u_0}_{\text{unresolved initial state}}, \tag{23}$$

where

$$\mathcal{K}(\tau) := \mathcal{P}\mathcal{L}e^{\tau\mathcal{Q}\mathcal{L}}\mathcal{Q}\mathcal{L}\mathcal{P}. \tag{24}$$

The resolved dynamics therefore depend on both their history and unresolved initial variables; see also Buitrago Ruiz et al. (2025).

For nonlinear PDEs, an analogous decomposition is formulated on observables using the linear Liouville operator and a projection on the observable space (Chorin et al., 2002). The resulting memory term need not have the linear convolution form above. In our model, a deterministic solver history provides a temporal representation of the observed initial state. We neither identify $\mathcal{K}$ nor obtain extra observations of unresolved variables by processing this history. Proposition 2 separates the corresponding information and representation limits.

B Reference Data and Numerical Solvers

This section describes the reference data used to construct the targets and the numerical solvers that generate the base trajectories in Section 4. The latter are initialized from coarse observations through the maps $\mathcal{E}$ and $\mathcal{C}$ in (3).

B.1 Reference Data and Preprocessing

Observation grids and time windows. The target states are $u_n = \mathcal{P}_f(u_n^H)$ as defined in (2). The first snapshot of each retained window defines u_0 and the local time origin $t_0 = 0$; it need not be the first snapshot in the source file. The forecast durations $N\Delta t$ are 2.5 for KS, 1.4 for Burgers, and 20 for NS. For Burgers, the source observation spacing is 0.01, and retaining every seventh snapshot gives the spacing $\Delta t = 0.07$ used in our experiments. Observation intervals are distinct from the internal integration steps described below.

Kuramoto–Sivashinsky equation. We follow the reference-generation procedure of Buitrago Ruiz et al. (2025). Writing u_{init} for the continuous initial field of the reference simulation, we sample

$$u_{\text{init}}(x) = \sum_{j=0}^{20} c_j \sin\left(\frac{2\pi k_j x}{L} + \phi_j\right), \tag{25}$$

with $c_j \sim \text{Unif}[-0.5, 0.5]$, $k_j \sim \text{Unif}\{1, \ldots, 8\}$, and $\phi_j \sim \text{Unif}[0, 2\pi]$. Here $L = 64$, and the sum comprises 21 sinusoidal terms whose wavenumber indices lie between 1 and 8. Spatial derivatives are evaluated by a Fourier pseudo-spectral method on the 512-point reference grid. The resulting method-of-lines system is integrated with the fifth-order implicit Radau IIA method through `scipy.integrate.solve_ivp`, and output is sampled at $\Delta t = 0.1$.

Burgers equation. We use the published PDEBench trajectories rather than generating new reference solutions. In the source construction (Takamoto et al., 2022), initial fields are formed from two sinusoidal components with amplitudes sampled uniformly from $[0, 1]$, wavenumber indices from $\{1, \ldots, 4\}$, and phases from $[0, 2\pi]$. Random sign flips and optional absolute-value and spatial-window transformations diversify these fields; the latter two transformations are each applied with probability 0.1. The source solver uses a second-order finite-volume discretization, slope-limited reconstruction, upwind advective fluxes, and centered diffusive fluxes. An explicit second-order predictor–corrector update uses adaptive time steps constrained by advection and diffusion.

Navier–Stokes equation. We use the reference dataset of Li et al. (2021). Its initial vorticity is sampled from the periodic Gaussian random field

$$u_{\text{init}} \sim \mathcal{N}\Big(0,\, 7^{3/2}(-\Delta + 49I)^{-2.5}\Big)\,.$$

For consistency with the periodic vorticity–velocity relation below, the covariance is interpreted on the zero-spatial-mean subspace. The spatial forcing is

$$F(x_1, x_2) = 0.1\left[\sin\big(2\pi(x_1 + x_2)\big) + \cos\big(2\pi(x_1 + x_2)\big)\right]. \tag{26}$$

The reference solver uses a Fourier pseudo-spectral discretization on a 256×256 grid. With stream function χ, velocity is recovered from vorticity through

$$-\Delta\chi = u, \qquad \mathbf{v} = (\partial_{x_2}\chi, -\partial_{x_1}\chi). \tag{27}$$

This periodic Poisson problem requires vorticity with zero spatial mean; imposing zero mean on χ fixes its additive constant. The Poisson problem and spatial derivatives are evaluated in Fourier space; nonlinear advection is evaluated in physical space and dealiased. Time integration uses a semi-implicit Crank–Nicolson update with implicit diffusion and an explicit nonlinear term. The source integration step is 10^{-4}, and observations are recorded every unit of time.

B.2 Stage-1 Numerical Solvers

The update $\widetilde{\Psi}$ in (3) advances the numerical state over one observation interval Δt, possibly using several internal steps. All physical coefficients, forcing, and numerical parameters are fixed during readout training.

Kuramoto–Sivashinsky equation. The lifting $\mathcal{E}$ maps u_0 to a persistent 128-point solver grid by retaining the represented Fourier coefficients and setting unresolved coefficients to zero. Spatial discretization is pseudo-spectral, with the $3/2$ rule for dealiasing the quadratic term. For integer mode index k, define the physical wavenumber and linear Fourier multiplier by

$$\omega_k = \frac{2\pi k}{L}, \qquad \mu_k = \omega_k^2 - \nu\omega_k^4, \qquad L = 64. \tag{28}$$

First-order exponential time differencing (ETD1) (Cox & Matthews, 2002) integrates the linear part exactly while treating the nonlinear part explicitly. We use ten internal steps per observation interval, each of size $\Delta t/10 = 0.01$. After each interval, $\mathcal{C}$ spectrally projects the internal state onto the f-point observation grid to obtain $\bar{u}_n$. The 128-point state is retained for the next interval; it is not reinitialized from $\bar{u}_n$ or from a learned prediction.

Burgers equation. The solver evolves the state on the f-point observation grid. Nonlinear advection is discretized by a conservative finite-volume method with MUSCL reconstruction and a monotonized-central slope limiter (van Leer, 1979). We use the exact Godunov flux and second-order strong-stability-preserving Runge–Kutta (SSPRK2) time stepping (Gottlieb et al., 2001). Diffusion is integrated exactly in Fourier space and coupled to advection by Strang splitting (Strang, 1968). Adaptive internal steps enforce an advective CFL number of at most 0.45 and advance the solver to the next observation time.

Navier–Stokes equation. The solver evolves vorticity on the 64×64 observation grid. Velocity is recovered using the Fourier-space Poisson solve in (27). Semi-Lagrangian advection (Staniforth & Côté, 1991) uses iteratively computed characteristic departure points and bicubic interpolation. Adaptive internal steps limit the transport Courant number to 2.0. Diffusion and the fixed forcing in (26) are integrated in Fourier space and combined with transport by Strang splitting (Strang, 1968).

C Implementation Details of Data-Driven Components

C.1 Neural Backbone Architectures and Learning Procedure

FFNO. We use the Factorized Fourier Neural Operator (FFNO) (Tran et al., 2023), an extension of the Fourier Neural Operator (FNO) (Li et al., 2021). For KS and Burgers, the architecture stacks four FFNO layers with a hidden dimension of 128, while for NS it uses two FFNO layers with a hidden dimension of 64. Across all benchmarks, the inner dimension of the MLPs is set to four times the hidden dimension.

S4FFNO. S4FFNO (Buitrago Ruiz et al., 2025) augments FFNO with Structured State Space sequence (S4) layers (Gu et al., 2022a) to model temporal dependencies. For KS and Burgers, we use the FFNO–FFNO–S4–FFNO–FFNO configuration, i.e., a single S4 layer inserted between the second and third FFNO layers, whereas for NS we use the lighter FFNO–S4–FFNO configuration. In both cases, the FFNO layers use the same hidden dimensions as the corresponding FFNO baseline. The S4 layer relies on the diagonal S4D parameterization (Gu et al., 2022b) with a state dimension of 64. For KS and Burgers, the model has access to the full available temporal history; for NS, the memory window is limited to the last four time steps to reduce GPU memory consumption.

Learning procedure. All hybrid gradient-based baselines adopt the spatial grid encoding of Buitrago Ruiz et al. (2025). For KS and Burgers, models are trained for up to 200 epochs with a batch size of 5, using early stopping with a patience of 20 epochs on the validation loss. For NS, models are trained for up to 10 epochs with a batch size of 3, using early stopping with a patience of 5 epochs on the validation MSE. The learning rate is set to 10^{-4} for KS and NS, and 10^{-5} for Burgers.

Tuning and stability. The SITL baselines were not tuned for accuracy: we performed no systematic hyperparameter search over architectures, learning rates, batch sizes, or the unroll length U. Instead, the protocol above, together with $U = 4$ (Section 6), was chosen to obtain stable training of the coupled solver–corrector rollouts. For SITL, validation data serve for checkpoint selection and early stopping, whereas the closed-form methods also use them to select hyperparameters from the grid in Appendix C.2. The SITL results in Table 1 therefore reflect this fixed protocol and may not represent the best accuracy attainable with tuning.

C.2 Hyperparameter Selection for Closed-Form Methods

We use the same hyperparameter grid for the two-stage RC method and CL–HRC on their shared benchmarks. The reservoir dimension is fixed to $D = 1000$ for KS and Burgers and $D = 100$ for NS. The recurrent matrix W is initialized with 10% connectivity, with nonzero entries drawn from the standard Gaussian distribution, and is then rescaled to the prescribed spectral radius $\rho(W)$. The input matrix W_{in} is dense, with entries sampled independently from $\mathcal{U}(-1, 1)$. The leak rates are logarithmically spaced over $[10^{-3}, 1]$, and the input scaling is fixed to $\sigma_{\text{in}} = 1$.

For each method and spatial resolution, we select the stencil radius m, recurrent spectral radius $\rho(W)$, and ridge coefficient λ from the grid in Table 2. Each readout is fitted using only the training trajectories. Model selection minimizes the mean validation MSE over all forecast steps and validation trajectories, excluding the initial condition. CL–HRC is validated through closed-loop rollouts, whereas the two-stage method reuses fixed solver trajectories across candidate configurations.

The selected model is then evaluated on the held-out test set. Test trajectories are used neither for fitting nor for hyperparameter selection. The nRMSE values reported for model comparisons and ablations are test errors.

C.3 Training-Time Accounting

The training times in Table 1 include validation-based model selection. For the two-stage RC method, they include the Stage-1 solver rollouts that generate the base trajectories for the training and validation sets; these rollouts are performed once and reused across all candidate configurations. They further include reservoir feature construction, accumulation of the regression statistics,

Table 2: Hyperparameter grid shared by the two-stage RC method and CL–HRC.

Hyperparameter	Candidate values
Stencil radius m	$\{0, 1, 2, 3\}$
Spectral radius $\rho(W)$	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$
Ridge coefficient λ	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$

the ridge solves, and validation of every candidate configuration in Table 2. Solver rollouts on the test set are not included. For the SITL baselines, solver steps are executed within every training iteration and are therefore part of the measured time, as are the full-horizon validation rollouts used for checkpoint selection and early stopping. For CL–HRC, the measured times likewise include the teacher-forced solver steps on the training set and the candidate-specific closed-loop validation rollouts.

D Ablation: Effect of Reservoir Memory

We assess the contribution of reservoir memory on KS at $f = 32$ by comparing the full model with a memoryless variant. Setting $W = 0$ and $A = I_D$ in (6) removes both recurrent coupling and the leaky state update, giving the instantaneous nonlinear feature map

$$h_n^{(p)} = \tanh\Big(\sigma_{\text{in}} W_{\text{in}} s_n^{(p)}\Big). \tag{29}$$

The hidden state then depends only on the current solver stencil. Setting $W = 0$ alone would retain temporal memory through $(I - A)h_{n-1}^{(p)}$ whenever $A \neq I_D$.

Both variants use the same solver trajectories, feature dimension, odd/even feature augmentation, and shared Fourier ridge readout. Each variant's readout is fitted to the training data, with hyperparameters selected using the validation set. Table 3 reports held-out test nRMSE as the mean and sample standard deviation over three random seeds.

Table 3: Effect of reservoir memory on KS at $f = 32$. Entries are test nRMSE, averaged over forecast steps and test trajectories, excluding the initial condition, and reported as mean $\pm$ sample standard deviation over three random seeds. These seeds use reservoir realizations different from the single run reported in Table 1. Bold marks the lower mean error.

Variant	Test nRMSE $\downarrow$
Without memory	0.3763 ± 0.0006
Full reservoir	$\mathbf{0.1249} \pm 0.0011$

The memory-based reservoir reduces mean test nRMSE by a factor of approximately 3.0. In this setting, retaining solver history yields more effective features for the shared readout than the instantaneous feature map. Both variants remain deterministic functions of u_0; temporal processing changes the representation available to the readout, consistent with Section 5.

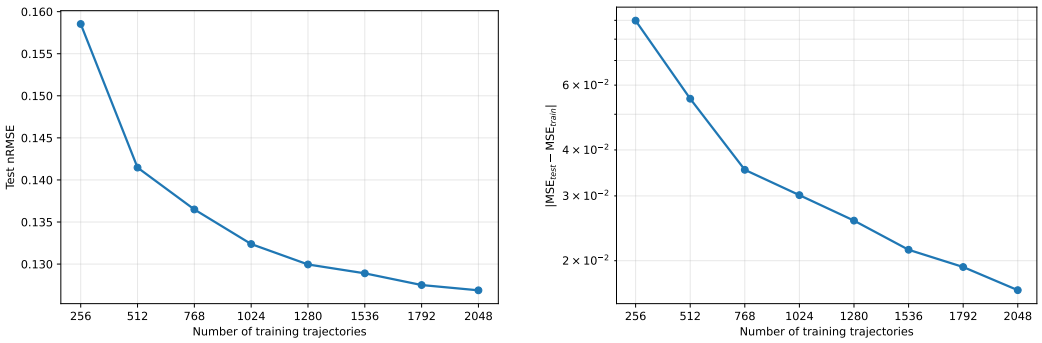
E Learning Curves on KS

Experimental protocol. We evaluate the effect of training-set size on KS at $f = 32$, using nested subsets containing the first $M \in \{256, 512, 768, 1024, 1280, 1536, 1792, 2048\}$ training trajectories. We retain the solver settings, reservoir realization, and hyperparameter settings of the $M = 2048$ configuration, and refit only the shared Fourier readout for each subset, without additional hyperparameter search. Consistent with the main model, the readout receives only the augmented reservoir features $r_n = \psi(h_n)$. All runs use the same held-out test set and forecast horizon.

Metrics. We report test nRMSE and the absolute training–test MSE difference,

$$\Delta_M := \left|\text{MSE}_{\text{test}}(M) - \text{MSE}_{\text{train}}(M)\right|. \tag{30}$$

Training MSE is evaluated on the subset used to fit the readout. Both MSEs average squared errors over trajectories, forecast steps, and spatial grid points, excluding the initial condition.



(a) Test nRMSE

(b) Absolute MSE difference

Figure 1: Learning curves on KS at $f = 32$ with fixed reservoir and hyperparameter settings. Left: test nRMSE. Right: absolute training–test MSE difference Δ_M on a logarithmic vertical axis. Each point refits the readout on M training trajectories.

Scope. Figure 1 examines how training-set size affects test accuracy and the empirical training–test discrepancy for a fixed model configuration. These curves do not directly validate the constants or convergence rate in Theorem 1, which concerns expected population MSE.

F Proofs and Additional Theoretical Results

F.1 Conventions, bounded features, and the ridge solution

All statements concern exact arithmetic and finite solver trajectories on which the numerical rollout is defined. The solver may have a persistent internal state z_n:

$$z_0 = \mathcal{E}(u_0), \qquad z_n = \widetilde{\Psi}(z_{n-1}), \qquad \bar{u}_n = \mathcal{C}(z_n).$$

The lifting, solver update, and observation maps are fixed. Consequently $\bar{u}_n = \mathcal{C}\widetilde{\Psi}^{\,n}\mathcal{E}(u_0)$ regardless of whether $\bar{u}_n$ alone is Markovian. Fixed forcing and fixed spatial coordinates are included in these maps and in the reservoir input.

For $0 < \alpha_a \leq 1$, each component of the reservoir update is a convex combination of its previous value and a value in $[-1, 1]$. Thus $h_0 = 0$ implies $h_n^{(p)} \in [-1, 1]^D$ for every finite input sequence, without any condition on $\rho(W)$ or q. The alternating linear/squared feature map obeys

$$\|r_n\|_F \leq \sqrt{PD}, \qquad \|\psi(a) - \psi(b)\|_\infty \leq 2\|a - b\|_\infty \quad (a, b \in [-1, 1]^D). \tag{31}$$

For the identity feature map, the Lipschitz constant can be replaced by one; the same bounded-feature estimate remains valid.

For the unitary DFT, Parseval's identity and the block-diagonal Fourier readout yield

$$\|\mathcal{T}_{\boldsymbol{w}} r\|_{2,P}^2 = \frac{1}{P}\sum_k |w_k^{\mathsf{H}} \widehat{r}_k|^2 \leq C_{\boldsymbol{w}}^2 \frac{\|r\|_F^2}{P}, \qquad C_{\boldsymbol{w}} = \max_k \|w_k\|_2. \tag{32}$$

In particular, $\|\hat{u}_n^{\boldsymbol{w}}\|_{2,P} \leq \sqrt{D}\, C_{\boldsymbol{w}}$. Bounded predictions need not be accurate, and this estimate does not provide stability of the numerical solver.

Parseval's identity separates the objective in (9) over Fourier modes, giving the solution in (10). The normal-equation matrices are positive definite since $\lambda > 0$. Real input and target fields give $w_{S,\lambda,-k} = \overline{w_{S,\lambda,k}}$; weights at self-conjugate modes are real. Hence the unconstrained mode-wise solutions satisfy the real-output constraint when paired modes use the same penalty. The readout space $\mathcal{W}$ is viewed as a real Hilbert space with inner product $\mathrm{Re}\sum_k w_k^{\mathsf{H}} v_k$. All convexity and gradient arguments below use this real structure.

F.2 Proof of Proposition 1

For a fixed realization of the feature construction, the map

$$G_n : (u_0, \dots, u_N) \longmapsto (\mathcal{R}_n(u_0), u_n)$$

is measurable. Training and test trajectories have the same law $\mathbb{P}$, so their images under G_n have the same pushforward law. This proves (11). Independence is needed later for statistical estimation, but equality in law of the two trajectories already suffices for this particular statement.

Neither the internal solver recursion nor the reservoir update contains $\boldsymbol{w}$. Therefore the feature field is unchanged when $\boldsymbol{w}$ is replaced by $\boldsymbol{v}$. Applying (32) to the difference of the two readouts and using (31) proves (12). An additive perturbation to a reported prediction at one time likewise leaves all later solver states and features unchanged, because the prediction is never consumed by either recursion. This is a structural absence of learned feedback, not a statement that reference prediction errors cannot grow.

Why a common coarse initial-state marginal is insufficient. For example, let $u_0 = 0$ almost surely in both populations, while $u_1 = 1$ in the training population and $u_1 = -1$ in the test population. Then the feature marginal is identical but the feature–target joint law is different. Hidden initial scales or different forcing laws can produce the same issue in partially observed dynamics. A common joint trajectory law, or an equivalent common conditional target law given the coarse initial state, rules out this mismatch.

F.3 Proof of Proposition 2

Let $\mathcal{G}_0 = \sigma(u_0, J)$ and $\mathcal{G}_r = \sigma(X)$. Since $X = \mathcal{R}_J(u_0)$, we have $\mathcal{G}_r \subseteq \mathcal{G}_0$ and, by the tower property,

$$m_r = \mathbb{E}[Y \mid \mathcal{G}_r] = \mathbb{E}[m_0 \mid \mathcal{G}_r].$$

For a fixed readout, write

$$Y - \mathcal{T}_{\boldsymbol{w}} X = (Y - m_0) + (m_0 - m_r) + (m_r - \mathcal{T}_{\boldsymbol{w}} X).$$

The first term is orthogonal in L^2 to every square-integrable $\mathcal{G}_0$-measurable variable. The second is orthogonal to every square-integrable $\mathcal{G}_r$-measurable variable. All cross terms therefore vanish after expectation in the normalized Euclidean inner product, proving (13).

Consequently every predictor measurable with respect to (u_0, J) has risk at least $\mathcal{E}_{\mathrm{obs}}$; every predictor based only on r_J has risk at least $\mathcal{E}_{\mathrm{obs}} + \mathcal{E}_{\mathrm{rep}}$. The readout in the manuscript is shared over time and receives no separate time variable. Thus $\mathcal{E}_{\mathrm{rep}}$ includes possible ambiguity between feature fields at different forecast times. If time is explicitly supplied, replace X by (r_J, J); the conditioning information has then changed. For a fitted readout independent of a fresh test trajectory, the decomposition holds conditional on its training set and then also after averaging over training sets.

F.4 Proof of Proposition 3

Let $\delta_n := \max_p \|h_n^{(p)} - h_n^{\prime(p)}\|_\infty$. Componentwise application of the 1-Lipschitz property of $\tanh$ gives

$$\begin{aligned} |h_{n,a}^{(p)} - h_{n,a}^{\prime(p)}| \le (1-\alpha_a)|h_{n-1,a}^{(p)} - h_{n-1,a}^{\prime(p)}| \\ + \alpha_a \sum_b |W_{ab}||h_{n-1,b}^{(p)} - h_{n-1,b}^{\prime(p)}| \\ + \sigma_{\mathrm{in}}\alpha_a \sum_b |(W_{\mathrm{in}})_{ab}||s_{n,b}^{(p)} - s_{n,b}^{\prime(p)}|. \end{aligned}$$

Taking maxima yields $\delta_n \le q\delta_{n-1} + b_{\mathrm{in}}\eta_n$. More generally, possibly different initial states satisfy

$$\delta_n \le q^n \delta_0 + b_{\mathrm{in}} \sum_{j=1}^{n} q^{n-j}\eta_j. \tag{33}$$

This finite-horizon inequality holds for any $q \ge 0$ (with $q^0 = 1$). For the architecture in the manuscript, $\delta_0 = 0$. Equation (31) implies $P^{-1/2}\|r_n - r_n'\|_F \le 2\sqrt{D}\,\delta_n$. Combining this with (32) proves (15). Summing a geometric series proves the uniform perturbation bound when $q < 1$.

Finite-history forgetting. If two histories agree on the last H_{hist} inputs, where $1 \le H_{\text{hist}} \le n$, the same recurrence gives $\delta_n \le q^{H_{\text{hist}}}\delta_{n-H_{\text{hist}}}$. Since both states belong to $[-1,1]^D$, for $q < 1$,

$$\|\hat{u}_n - \hat{u}'_n\|_{2,P} \le 4\sqrt{D}\, C_{\boldsymbol{w}} q^{H_{\text{hist}}}. \tag{34}$$

The constant becomes $2\sqrt{D}\, C_{\boldsymbol{w}}$ if one history is restarted from zero at time $n - H_{\text{hist}}$ and the other has a reachable state. These are upper bounds on forgetting; they do not prove that past inputs are retained or remain recoverable.

A less restrictive weighted sufficient condition. Let $B := I - A + A|W|$, where absolute value is entrywise. If a vector $v > 0$ satisfies $Bv \le q_v v$ for some $q_v < 1$, the same proof in $\|a\|_{\infty,v} := \max_j |a_j|/v_j$ gives contraction, with the corresponding weighted input constant. Such a vector exists whenever $\rho(B) < 1$: for any $\rho(B) < q_v < 1$, choose $v = (I - B/q_v)^{-1}\mathbf{1} > 0$, which obeys $Bv = q_v(v - \mathbf{1}) < q_v v$. The unweighted condition in the main text is obtained with $v = \mathbf{1}$. These are sufficient conditions, not necessary tests for the behavior of a driven reservoir (Yildiz et al., 2012).

Checking the certificate. Since $B - I = A(|W| - I)$ and A is diagonal with positive entries, a vector $v > 0$ satisfies $Bv < v$ if and only if $|W|v < v$. For a nonnegative matrix M, such a vector exists if and only if $\rho(M) < 1$: take $v = (I - M)^{-1}\mathbf{1}$ in one direction, and use $\rho(M) \le \max_j (Mv)_j/v_j$ in the other. Hence $\rho(B) < 1$ if and only if $\rho(|W|) < 1$, so whether the certificate holds does not depend on the leak rates, although its rate does. Because $\rho(W) \le \rho(|W|)$, the scaling $\rho(W) < 1$ is necessary but not sufficient. Moreover, $B \ge I - A$ entrywise gives $\rho(B) \ge 1 - \min_a \alpha_a$, so small leak rates force a certified rate close to one.

Table 4 reports these quantities for the reservoir selected in each configuration. All selected reservoirs satisfy $\rho(W) < 1$, but the certificate holds only for Burgers at $f = 64$, with $\rho(B) = 0.9991$, close to the lower bound $1 - 10^{-3}$. For this configuration, choosing $q_v = 0.9992 > \rho(B)$ gives a weighted contraction certificate with geometric factor $q_v^{20} \approx 0.984$ over the $N = 20$ forecast steps. This geometric term decays slowly over the evaluated horizon; it does not characterize the reservoir's actual memory response. For the other four configurations, Proposition 3 provides only the finite-horizon bound (15).

Table 4: Contraction diagnostics for the selected reservoirs, with $B = I - A + A|W|$.

Configuration	D	$\rho(W)$	$\rho(\|W\|)$	$\rho(B)$	$\rho(B) < 1$
KS ($f = 32$)	1000	0.9	7.054	1.6934	No
KS ($f = 64$)	1000	0.7	5.486	1.4567	No
Burgers ($f = 32$)	1000	0.3	2.351	1.0566	No
Burgers ($f = 64$)	1000	0.1	0.784	0.9991	Yes
NS ($f = 64$)	100	0.9	2.314	1.0571	No

F.5 Proof of Theorem 1

The argument adapts regularized learning stability (Bousquet & Elisseeff, 2002) to a trajectory-valued example. Reservoir risk bounds under other sampling and dependence assumptions are studied by Gonon et al. (2020); no temporal mixing assumption is required here because different trajectories are independent.

Bounded prediction operators. For a trajectory Z, define the normalized target and prediction operator

$$\mathbf{y}_Z := \frac{1}{\sqrt{NP}}(u_1, \dots, u_N), \qquad L_Z\boldsymbol{w} := \frac{1}{\sqrt{NP}}(\mathcal{T}_{\boldsymbol{w}} r_1, \dots, \mathcal{T}_{\boldsymbol{w}} r_N).$$

Let $a(Z) := \|\mathbf{y}_Z\|_2$ and $\kappa := \sqrt{D}$. Equations (31)–(32) imply

$$\|L_Z\|_{\mathcal{W}\to 2} \le \kappa, \qquad \ell(\boldsymbol{w}; Z) = \|L_Z\boldsymbol{w} - \mathbf{y}_Z\|_2^2, \qquad \mathbb{E}a(Z)^2 = V_u.$$

Here L_Z is real-linear on the conjugate-symmetric readout space. For any $\boldsymbol{w}, \boldsymbol{v}$,

$$\|\nabla \ell(\boldsymbol{w}; Z)\|_{\mathcal{W}} \le 2\kappa(\kappa\|\boldsymbol{w}\|_{\mathcal{W}} + a(Z)), \tag{35}$$

$$|\ell(\boldsymbol{w}; Z) - \ell(\boldsymbol{v}; Z)| \le \kappa\big[\kappa(\|\boldsymbol{w}\|_{\mathcal{W}} + \|\boldsymbol{v}\|_{\mathcal{W}}) + 2a(Z)\big]\|\boldsymbol{w} - \boldsymbol{v}\|_{\mathcal{W}}. \tag{36}$$

Control of the fitted norm. Comparing the ridge objective with the zero readout gives

$$\lambda\|\boldsymbol{w}_{S,\lambda}\|_{\mathcal{W}}^2 \le \frac{1}{M}\sum_{i=1}^{M} a(Z_i)^2, \qquad \mathbb{E}_S\|\boldsymbol{w}_{S,\lambda}\|_{\mathcal{W}}^2 \le \frac{V_u}{\lambda}. \tag{37}$$

No bounded-support assumption is used.

Replacing one independent trajectory. Let $S^{(i)}$ replace Z_i by an independent copy Z_i'. Write $w = \boldsymbol{w}_{S,\lambda}$, $v = \boldsymbol{w}_{S^{(i)},\lambda}$ and suppress the subscript on the readout norm. The objective $F_S = \widehat{\mathcal{R}}_S + \lambda\|\cdot\|^2$ is 2λ-strongly convex. Strong monotonicity of its gradient, together with $\nabla F_S(w) = \nabla F_{S^{(i)}}(v) = 0$, yields

$$\|w - v\| \le \frac{\|\nabla F_S(v)\|}{2\lambda} \tag{38}$$

$$\le \frac{\kappa}{\lambda M}\big(2\kappa\|v\| + a(Z_i) + a(Z_i')\big). \tag{39}$$

Apply (36) at Z_i' and then Cauchy–Schwarz in probability. By Minkowski's inequality and (37), both random factors

$$\kappa(\|w\| + \|v\|) + 2a(Z_i'), \qquad 2\kappa\|v\| + a(Z_i) + a(Z_i')$$

have L^2 norm at most $2\sqrt{V_u}(1 + \kappa/\sqrt{\lambda})$. It follows that

$$\mathbb{E}\big|\ell(w; Z_i') - \ell(v; Z_i')\big| \le \beta_{M,\lambda}, \qquad \beta_{M,\lambda} := \frac{4\kappa^2 V_u}{\lambda M}\left(1 + \frac{\kappa}{\sqrt{\lambda}}\right)^2. \tag{40}$$

Dependencies between these factors cause no problem: the argument uses their second moments, not their independence.

Generalization and the oracle inequality. Since Z_i' is independent of S, and since swapping Z_i with Z_i' leaves the joint sampling law unchanged,

$$\mathbb{E}_S\big[\mathcal{R}(\boldsymbol{w}_{S,\lambda}) - \widehat{\mathcal{R}}_S(\boldsymbol{w}_{S,\lambda})\big] = \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}\big[\ell(w; Z_i') - \ell(v; Z_i')\big] \le \beta_{M,\lambda}.$$

The absolute value of the expected gap is bounded by the same constant. For any fixed comparator $\boldsymbol{w}_\star \in \mathcal{W}$, optimality of ridge regression gives

$$\widehat{\mathcal{R}}_S(\boldsymbol{w}_{S,\lambda}) + \lambda\|\boldsymbol{w}_{S,\lambda}\|_{\mathcal{W}}^2 \le \widehat{\mathcal{R}}_S(\boldsymbol{w}_\star) + \lambda\|\boldsymbol{w}_\star\|_{\mathcal{W}}^2.$$

Taking expectations, dropping the nonnegative fitted penalty, and using the preceding expected-gap bound proves (16) after taking the infimum over comparators and substituting $\kappa^2 = D$. In conjunction with Proposition 2, this also gives

$$\mathbb{E}_S\mathcal{R}(\boldsymbol{w}_{S,\lambda}) \le \mathcal{E}_{\mathrm{obs}} + \mathcal{E}_{\mathrm{rep}} + \inf_{\boldsymbol{w}\in\mathcal{W}}\big\{\mathbb{E}\|m_r - \mathcal{T}_{\boldsymbol{w}}X\|_{2,P}^2 + \lambda\|\boldsymbol{w}\|_{\mathcal{W}}^2\big\} + \beta_{M,\lambda}.$$

The bound is conservative. It proves no ordering of risk relative to SITL, FFNO, or other model classes.

F.6 Additional consequences and scope

High-probability bound under bounded trajectory energy. If additionally $a(Z) \le Y_\star$ almost surely, define

$$B_\lambda := Y_\star^2(1 + \sqrt{D/\lambda})^2, \qquad \beta_\star := \frac{4DY_\star^2}{\lambda M}(1 + \sqrt{D/\lambda})^2.$$

The proof above then gives uniform replacement stability $|\ell(\boldsymbol{w}_{S,\lambda}; Z) - \ell(\boldsymbol{w}_{S^{(i)},\lambda}; Z)| \le \beta_\star$ for every admissible Z, and loss bounded by B_λ. For $g(S) = \mathcal{R}(\boldsymbol{w}_{S,\lambda}) - \widehat{\mathcal{R}}_S(\boldsymbol{w}_{S,\lambda})$, replacement changes g by at most $2\beta_\star + B_\lambda/M$. The expected-gap estimate and the bounded-difference inequality imply, with probability at least $1-\delta$,

$$\mathcal{R}(\boldsymbol{w}_{S,\lambda}) \le \widehat{\mathcal{R}}_S(\boldsymbol{w}_{S,\lambda}) + \beta_\star + (2M\beta_\star + B_\lambda)\sqrt{\frac{\log(1/\delta)}{2M}}. \tag{41}$$

The additional almost-sure bound is not implied by Gaussian initial conditions or by the largest energy observed in a finite dataset. Theorem 1 does not require this additional assumption.

Finite validation-based model selection. Condition on the training data and on K fitted candidate models constructed without access to an independent validation set of M_{val} trajectories. Suppose each candidate's trajectory loss lies in $[0, B_{\mathrm{val}}]$ almost surely. Hoeffding's inequality and a union bound give simultaneous deviation at most $B_{\mathrm{val}}\sqrt{\log(2K/\delta)/(2M_{\mathrm{val}})}$. Therefore the candidate $\hat{k}$ minimizing validation loss satisfies

$$\mathcal{R}(\hat{k}) \le \min_{k \le K} \mathcal{R}(k) + 2B_{\mathrm{val}}\sqrt{\frac{\log(2K/\delta)}{2M_{\mathrm{val}}}} \tag{42}$$

with conditional probability at least $1-\delta$. The candidate set must be fixed before inspecting this validation set; the statement does not cover unrestricted adaptive reuse of validation data. The bounded-loss premise also needs justification separately from the second-moment premise of the main theorem. Here selection uses the same squared trajectory loss as R, which matches the validation-MSE selection used for the closed-form methods in Section 6; the bounded-loss premise, however, is not verified there.

Translation equivariance. Let τ_a denote a periodic grid translation. Assume the complete solver-to-observation map is translation equivariant, no absolute coordinates or fixed inhomogeneous inputs are appended, and the reservoir initialization is spatially constant. If $u_0' = \tau_a u_0$, then $s_n'^{(p)} = s_n^{(p-a)}$; induction in the shared reservoir update gives $r_n'^{(p)} = r_n^{(p-a)}$. In Fourier space a translation multiplies all channels of mode k by the same phase, which commutes with w_k^{H}. Thus $\hat{u}_n' = \tau_a \hat{u}_n$. Weight sharing alone does not establish equivariance of the full pipeline: the solver and any forcing must also respect the translation. With fixed position-dependent forcing or fixed coordinates, general translations of u_0 alone need not be symmetries. The feature-to-output Fourier map remains equivariant, and simultaneous translation of all spatial input fields preserves the reservoir equivariance.

Leak rates and actual memory response. In the uncoupled special case $W = 0$, the recurrence can be solved componentwise:

$$h_{n,a}^{(p)} = \alpha_a \sum_{j=1}^{n} (1-\alpha_a)^{n-j} \tanh\Big(\sigma_{\mathrm{in}}(W_{\mathrm{in}} s_j^{(p)})_a\Big). \tag{43}$$

For $0 < \alpha_a < 1$, its exponential decay scale is $-1/\log(1-\alpha_a)$ in observation steps; $\alpha_a = 1$ retains only the current input. For $W \neq 0$, the channels interact and these decay scales are no longer exact. In particular, letting

$$D_n^{(p)} := \mathrm{diag}\Big(1 - \tanh^2(W h_{n-1}^{(p)} + \sigma_{\mathrm{in}} W_{\mathrm{in}} s_n^{(p)})\Big),$$
$$J_n^{(p)} := I - A + A D_n^{(p)} W, \qquad K_n^{(p)} := \sigma_{\mathrm{in}} A D_n^{(p)} W_{\mathrm{in}},$$

the local input sensitivity is

$$\frac{\partial h_n^{(p)}}{\partial s_j^{(p)}} = J_n^{(p)} \cdots J_{j+1}^{(p)} K_j^{(p)}, \qquad j \le n, \tag{44}$$

with an empty product equal to the identity. These Jacobian products can diagnose local memory along sampled trajectories; they do not replace a global finite-perturbation proof.

What the analysis does not establish. No statement proves that an arbitrary fixed finite reservoir is a universal approximator, that the solver-driven history identifies an MZ closure, that all experimental reservoirs satisfy a contraction certificate, or that forecast accuracy extends beyond the training horizon. An empirical check of q or a justified alternative contraction certificate is needed before invoking the uniform stability and fading-memory conclusions for a specific reservoir. Table 4 reports such a check for the selected reservoirs. The nonfeedback, bounded-feature, and expected-risk results do not require such a certificate.