# Physics-Enforced Reservoir Computing for Forecasting Spatiotemporal Systems

**Dima Tretiak**[1] DTRETIAK@UW.EDU
**Anastasia Bizyaeva**[2] ANASTASIAB@CORNELL.EDU
**J. Nathan Kutz**[3,4] KUTZ@UW.EDU
**Steven L. Brunton**[1] SBRUNTON@UW.EDU
[1] *Department of Mechanical Engineering, University of Washington*
[2] *Sibley School of Mechanical and Aerospace Engineering, Cornell University*
[3] *Department of Applied Mathematics, University of Washington*
[4] *Department of Electrical and Computer Engineering, University of Washington*

**Editors:** N. Ozay, L. Balzano, D. Panagou, A. Abate

## Abstract

We present a new approach for incorporating hard physical constraints into *reservoir computing* (RC). The goal of this work is to increase the reliability, trustworthiness, and generalizability of RC by guaranteeing adherence to known physical laws, particularly while simulating high dimensional systems such as spatiotemporal fluid flows. A reservoir is commonly implemented as a single-layer recurrent neural network in which only the linear output layer is trained and all other parameters are randomly initialized and fixed. Therefore, training a reservoir only involves solving a least squares problem for the weights of the final layer, posing an excellent opportunity to analytically enforce hard constraints. We show that physical constraints, such as conservation laws and boundary conditions, can be imposed in the training procedure and can be guaranteed to hold for forecasting. We introduce *physics enforced reservoir computing* (PERC) in which the RC training procedure is augmented with a linear homogeneous constraint defined by a linear operator. We then demonstrate this method by enforcing conserved quantities in the Kuramoto–Sivashinsky system and zero-divergence constraints (mass conservation) in the Kolmogorov flow. In both cases, we enforce these constraints to near machine precision. We provide our code online here.

**Keywords:** Reservoir Computing, Constrained Optimization, Physical Constraints, Fluid Dynamics, Forecasting

## 1. Introduction

High-dimensional dynamical systems with complex spatiotemporal features are prevalent across many fields, such as numerical weather prediction, fluid dynamics, neuroscience, traffic management, and economics. Simulated forecasts provide key insights needed for design, optimization, and control of these systems. While numerical integration offers many benefits, including accuracy and robustness, it is limited when there is incomplete knowledge of the physics. An increasing abundance of data from real-world systems therefore motivates the development and use of data-driven methods for time-series forecasting. Among these methods, *reservoir computing* (RC) (Jaeger and Haas, 2004; Maass et al., 2002) has emerged as a powerful technique that yields accurate forecasts of dynamic time-series with minimal training requirements (Vlachas et al., 2020; Lu et al., 2018; Griffith et al., 2019). In this paper, we extend the RC method beyond its classical black-box setting and present a novel approach for incorporating physical constraints into RC training.

Recurrent neural network (RNN) reservoirs are known for their forecasting performance and their ability to reconstruct consistent statistics of chaotic attractors (Lu et al., 2018). In the RC context, training the model entails solving a least-squares minimization problem. In contrast with more widely used nonlinear optimization methods, RC training avoids exploding or vanishing gradients and other pitfalls of backpropagation. The current leading method for high-dimensional RC, known as *localization* (Pathak et al., 2018a), involves training multiple reservoirs in parallel on data from distinct but overlapping regions of the spatial domain. The method has been used to model the 1D Kuramoto-Sivashinsky equation (Pathak et al., 2018a; Vlachas et al., 2020) and several climatological systems (Arcomano et al., 2020; Penny et al., 2022). The other leading RC method for high-dimensional systems involves data compression (Pandey and Schumacher, 2020; Doan et al., 2021b). Importantly, across all of these applications, the RC method is agnostic of the physics of the underlying process that is being forecasted.

Incorporating physical knowledge into machine learning (ML) has been shown to increase data efficiency, provide robustness to noise, and improve generalization of models (Ling et al., 2016; Loiseau and Brunton, 2018; Karniadakis et al., 2021). Physics may be incorporated through hard or soft constraints. Soft constraints often take the form of a "physics-informed" penalty in the loss function (Raissi et al., 2019; Wang et al., 2021), while hard constraints typically require changes to the ML architecture or optimization algorithm, resulting in near-exact enforcement of the constraints (Loiseau and Brunton, 2018; Finzi et al., 2021; Otto et al., 2024), but at the cost of a less flexible implementation. Examples include energy-conserving Hamiltonian and Lagrangian neural networks (Greydanus et al., 2019; Lutter et al., 2019; Zhong et al., 2020; Cranmer et al., 2020; Zhong and Leonard, 2020), symmetry enforcement methods (Loiseau and Brunton, 2018; Ahmadi and El Khadir, 2020; Finzi et al., 2021; Satorras et al., 2021; Yang et al., 2024; Otto et al., 2024), and mass conserving neural networks for turbulence (Mohan et al., 2023; Tretiak et al., 2022).

Several recent efforts have implemented physical knowledge into RC. Approaches include fine-tuning imperfect model predictions with RC (Pathak et al., 2018b), and using feedback control from a physical model to adjust the reservoir connectivity matrix (Perrusquía and Guo, 2024). Physics-informed echo state networks (PI-ESNs) (Doan et al., 2020, 2021a; Mammedov et al., 2022) implement a physics-based loss term in the RC objective as a soft constraint in order to promote physically-consistent forecasts. The loss is then tuned by first fitting the training data via ridge regression, and then using this as an initial guess for an iterative L-BFGS-B (Byrd et al., 1995) optimization to fit the physical constraints. These works show that accounting for physical constraints in RC improves forecasting performance across several benchmark systems.

The present work explores enforcing physics in RC through hard constraints, which enables conservation laws to be enforced to machine precision in forecasts; see Fig. 1 for an overview schematic. Thus, the physics are guaranteed to be enforced in the forecast regardless of noise, amount of training samples, or out-of-distribution testing data as proven in Section 3. The primary contributions of *physics-enforced reservoir computing* (PERC) are:

1. We formulate a constrained optimization problem to embed physical laws as linear, homogeneous constraints during RC training.
2. We develop an analytic solution in Section 3 and numerically show that the computational implementation adds minimal overhead and doesn't increase training time.
3. We show that conservation laws are enforced to machine precision in forecasts on two spatiotemporal systems (Kuramoto–Sivashinsky in Section 4.1 and Kolmogorov flow in Section 4.2).

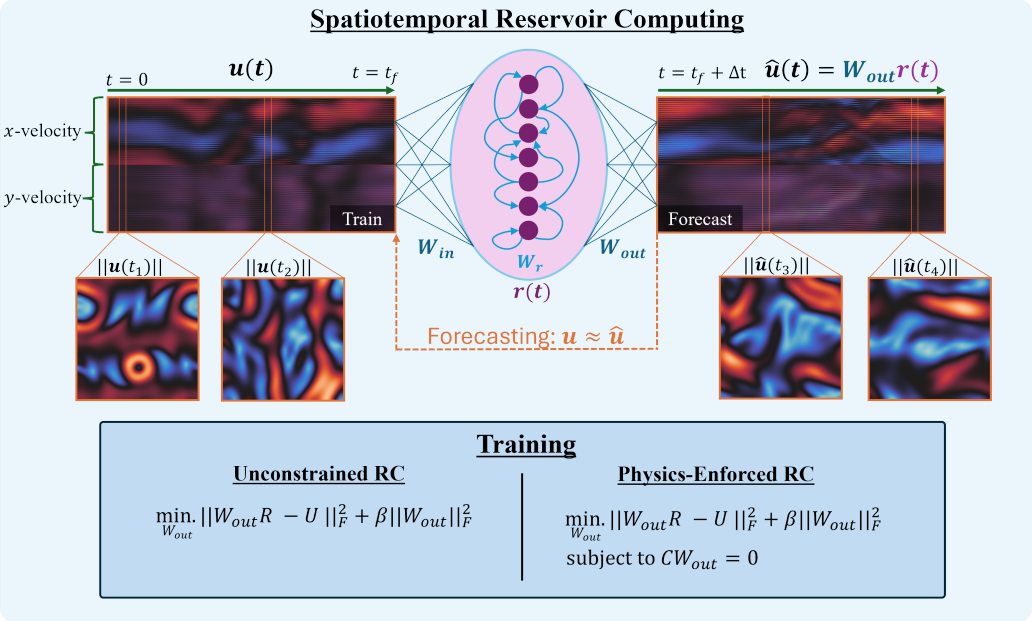


Figure 1: Physics-Enforced Reservoir Computing (PERC) architecture applied to 2D Kolmogorov flow. Flow snapshots $\mathbf{u}(t)$ are projected into a recurrent reservoir $\mathbf{r}(t)$, evolved by sparse dynamics $W_r$. A linear readout $W_{\text{out}}$ trained via regularized regression to map reservoir states to flow trajectories. Future forecasts $\hat{\mathbf{u}}(t)$ are generated from projected reservoir states, optionally enforcing physical constraints during training.

## 2. Reservoir Computing (RC) Background

A *reservoir* is a driven dynamical system satisfying an "echo state" or "fading memory" property (Nakajima and Fischer, 2021). The RC technique is used to train a reservoir system to accomplish a variety of machine learning tasks, leveraging the intrinsic input-output properties of the system and bypassing the need for gradient-based optimization. In this paper, we focus on the problem of forecasting a time series $\mathbf{u}(t) \in \mathbb{R}^{N_u}$ provided data for $t \in \{0, \ldots, t_f\}$. During training, the dynamics of the reservoir state $\mathbf{r}(t) \in \mathbb{R}^{N_r}$ are forced by the input data $\mathbf{u}(t)$:

$$\mathbf{r}(t+1) = F(\mathbf{r}(t), \mathbf{u}(t)), \quad t = 0, \ldots, t_f \tag{1}$$

where $F : \mathbb{R}^{N_r} \times \mathbb{R}^{N_u} \to \mathbb{R}^{N_r}$ is a nonlinear, continuously differentiable function. The "echo state" or "fading memory" property of the reservoir (1) is required for forecasting. This property is guaranteed when a given reservoir map is sufficiently smooth and satisfies a uniform contraction condition (Jaeger, 2007; Yildiz et al., 2012). When these conditions are satisfied, the reservoir system (1) reaches *generalized synchronization* with its driving input, meaning that trajectories of the coupled system converge to a manifold

$$\mathcal{M}_{GS} = \{(\mathbf{r}, \mathbf{u}) \text{ s.t. } \mathbf{r} = G(\mathbf{u})\} \tag{2}$$

where $G : \mathbb{R}^{N_u} \to \mathbb{R}^{N_r}$ is a continuous function (Jaeger, 2007; Platt et al., 2021a; Ohkubo and Inubushi, 2024). Typically, $N_r > N_u$ and the reservoir equations (1) can be interpreted as lifting the low-dimensional driving signal to a higher-dimensional state space. The map $G$ in (2) is conjectured to be locally invertible (Lu and Bassett, 2020). The training step in the RC approach then involves

| Model | $N_r$ | $\alpha$ | $\beta$ | $\sigma_{W_{\text{in}}}$ | $\sigma$ | $\lambda_{\max}$ |
|---|---|---|---|---|---|---|
| Unconstrained RC (KS) | 5,000 | 0.534 | $10^{-7}$ | 0.088 | 1.915 | 0.757 |
| PERC (KS) | 5,000 | 0.9 | $10^{-7}$ | 0.0439 | 0.857 | 0.429 |
| Both Models (Kol. Flow) | 20,000 | 0.6 | $10^{-7}$ | 1.6 | 0.084 | 0.8 |

Table 1: Hyperparameters (as determined via random search) used for the to unconstrained (UCRC) and physics-enforced (PERC) reservoir models for both example systems

learning the local structure of $G^{-1} : \mathbb{R}^{N_r} \to \mathbb{R}^{N_u}$ from the relationship between the input time-series $U = \begin{bmatrix}\mathbf{u}(0) & \dots & \mathbf{u}(t_f)\end{bmatrix} \in \mathbb{R}^{N_u \times N_t}$ and the output response time-series of the reservoir state $R = \begin{bmatrix}\mathbf{r}(1) & \dots & \mathbf{r}(t_f+1)\end{bmatrix} \in \mathbb{R}^{N_r \times N_t}$, assuming the coupled system is close to the manifold $\mathcal{M}_{GS}$. It is known empirically that for sufficiently high-dimensional "well-behaved" reservoirs, the local map $G^{-1}$ is approximately linear. The training step in RC is then equivalent to finding a matrix $W_{\text{out}} \in \mathbb{R}^{N_u \times N_r}$ that approximately solves $U = W_{\text{out}}R$. Forecasting is then performed by simulating a closed-loop autonomous reservoir system

$$\mathbf{r}(t+1) = F(\mathbf{r}(t), W_{\text{out}}\mathbf{r}(t)), \quad t = t_f + 1, \dots \tag{3}$$

with the predictions made from linear readouts of the reservoir state with the learned map,

$$\hat{\mathbf{u}}(t+1) = W_{\text{out}}\mathbf{r}(t+1). \tag{4}$$

Commonly used RC architectures include the echo state network (ESN) (Jaeger and Haas, 2004), liquid state machine (Maass et al., 2002), and physical reservoir systems such as photonic devices (Tanaka et al., 2019). In our numerical studies we use the ESN formulation, which adapts a single hidden layer "Elman-style" RNN with a linear output (Elman, 1990). ESN equations read

$$\mathbf{r}(t+1) = (1-\alpha)\mathbf{r}(t) + \alpha f\left(W_{\text{r}}\mathbf{r}(t) + W_{\text{in}}\mathbf{u}(t) + \sigma\mathbf{1}\right) \tag{5}$$

where $f$ is a nonlinear activation function, $\alpha$ is the relative timescale between the ESN dynamics and its driving signal, and $\sigma$ is the strength of the bias vector. $W_{\text{in}}$ and $W_{\text{r}}$ are initialized randomly and fixed for the remainder of the problem. To control the strength of the input forcing, we draw $W_{\text{in}} \sim \mathcal{U}(-\sigma_{W_{\text{in}}}, \sigma_{W_{\text{in}}})$. Furthermore, the reservoir matrix is sampled $W_{\text{r}} \sim \mathcal{N}(0,1)$ with a matrix density of $\rho_{W_{\text{r}}} = 0.01$, then normalized to have spectral radius $\lambda_{\max}$. Although we choose $\gamma = 0$, we still include it in this discussion to stay consistent with the RC literature. Following results from (Platt et al., 2021b), the data is not normalized. A schematic of the training and forecasting a process with an ESN reservoir and a 2D Kolmogorov flow is shown in Figure 1 and hyperparameters are detailed in Table 1.

### 2.1. Unconstrained Ridge Regression

Training an RC first typically involves "spinning up" the reservoir for some amount of time $\gamma$ in order for the coupled system to converge to the generalized synchronization manifold (2). Therefore, the training data should involve samples of $u$ from times $t \in [-(\gamma + N_t + 1), 0]$[1] and the forecast begins at $t = 0$. While not strictly necessarily, spinup helps circumvent the initial synchronization

1. $u(N_t + 1)$ produces the first reservoir state after spinup. Since pairs are formed of reservoir state and next state of $u$, $u(N_t + 1)$ is not used in the least squares optimization.

| **Constraint Type** | **Example Field(s) / Systems** | **What is $U$?** | **What is $C$?** |
|---|---|---|---|
| Zero divergence of a vector field | Incompressible fluids, magnetic fields | Fluid/magnetic vector field | Finite difference divergence stencil |
| Conserved quantities | Atmospheric chemistry (Liu et al., 2024) | Chemical element reaction rates | Matrix containing stoichiometric ratios |
| Boundary conditions | Fluids (no-slip, periodic), heat transfer (insulated BC), Elasticity (fixed boundaries) | Fluid velocities/heat fluxes/structural displacements | Selection matrix that chooses relevant boundary points |
| Flux conditions | Circuits (Kirchhoff's Law), pipe flow (mass conservation) | Electrical currents/pipe flows in a network | Connectivity matrix of network nodes |
| Symmetries | Lie group symmetries (e.g., self-similarity, translational invariance) (Otto et al., 2024) | State space respecting a given spatial symmetry | Linear symmetry operator |

Table 2: Examples of systems where linear, homogeneous constraints of the form $CU = 0$ are applicable.

transient that may be present if the reservoir is not sufficiently contractive. After spinup, the $U$ and $R$ matrices are formed and Tikhonov-regularized least-squares (Tikhonov et al., 1977), also known as ridge regression with regularization strength $\beta$, is applied to learn a linear mapping between $u$ and $r$. Hence, training an RC requires solving the following optimization problem:

$$\min_{W_{\text{out}}} \left( \|W_{\text{out}} R - U\|_F^2 + \beta \|W_{\text{out}}\|_F^2 \right). \tag{6}$$

The problem is convex and admits a known analytical solution for the optimal readout matrix $W_{\text{out}}^*$

$$W_{\text{out}}^* = U R^T (R R^T + \beta I_{N_r})^{-1}. \tag{7}$$

## 3. PERC: Physics-Enforced Reservoir Computing

While (7) will find a global optimal solution, there are no guarantees that the learned mapping will enforce any physical constraints. However, known constraints can be directly enforced in the RC training optimization. We consider the problem of prescribing homogeneous, linear constraints that are described by a user-defined matrix $C \in \mathbb{R}^{N_c \times N_u}$ such that $C\hat{\mathbf{u}}(t) = 0$ for all $t$. Examples of such constraints include derivatives, integrals, or boundary conditions of the data $\mathbf{u}$. Periodic boundary conditions, for instance, can be enforced by creating a $C$ that extracts and subtracts the boundaries from $\mathbf{u}$. Section 4.1 demonstrates an example where $C$ is created to perform trapezoidal integration to enforce conserved quantities in the Kuramoto–Sivashinsky equation. Likewise, Section 4.2 demonstrates using a finite difference divergence stencil for $C$ to enforce incompressiblity in a fluid flow. Table 2 provides an overview of how this class of constraint applies more broadly. To enforce linear constraints, the original ridge regression problem can now be modified:

$$\begin{aligned} &\min_{W_{\text{out}}} \|W_{\text{out}} R - U\|_F^2 + \beta \|W_{\text{out}}\|_F^2 \\ &\text{subject to } C W_{\text{out}} = 0. \end{aligned} \tag{8}$$

We refer to RC training with the constrained optimization problem (8) as *physics enforced reservoir computing* (PERC). With a change of variables $A = R^T$, $B = U^T$, $X = W_{\text{out}}^T$, (8) reads

$$\begin{aligned} &\min_{X} \|AX - B\|_F^2 + \beta \|X\|_F^2 \\ &\text{subject to } C X^T = 0. \end{aligned} \tag{9}$$

**Theorem 1 (Solution to Constrained Optimization)** *Consider (9) where $A \in \mathbb{R}^{N_t \times N_r}$, $B \in \mathbb{R}^{N_t \times N_u}$, and $C \in \mathbb{R}^{N_c \times N_u}$. Let $(\cdot)^{-T}$ denote an inverse followed by a transpose. Suppose $C$ has rank $n$ and let $k = N_u - n$. Let $C^T P = Q\tilde{R}$ be a rank-revealing QR decomposition of $C^T$ with*

$$Q = \Big[ \underbrace{Q_1}_{N_u \times n} \mid \underbrace{Q_2}_{N_u \times k} \Big] \tag{10}$$

*where $Q_2$ is the silent portion of $Q$. The solution, denoted $(\cdot)^*$, to the optimization problem is*

$$(X^*)^T = Q_2 Q_2^T B^T A (A^T A + \beta I_{N_r})^{-T}. \tag{11}$$

**Proof** The constraint implies $X^T$ is in the null space of $C$. $Q_2$ is a basis for this null space if the QR decomposition is rank-revealing. Then, there is a $Y \in \mathbb{R}^{N_r \times k}$ so that $X^T = Q_2 Y^T$; equivalently,

$$X = Y Q_2^T. \tag{12}$$

The optimization problem in (9) can now be written without the constraint

$$\min_Y h(Y) = \min_Y \left\| AYQ_2^T - B \right\|_F^2 + \beta \|YQ_2^T\|_F^2. \tag{13}$$

As with ridge regression, this problem is convex and can be solved by finding $Y^*$ such that $\nabla_Y h(Y^*) = 0$:

$$\begin{aligned}
\nabla_Y h(Y) &= \frac{\partial}{\partial Y}\left( \|AYQ_2^T - B\|_F^2 \right) + \frac{\partial}{\partial Y}\left( \beta \|YQ_2^T\|_F^2 \right) \\
&= \frac{\partial}{\partial Y}\left[ \operatorname{tr}\left( (AYQ_2^T - B)(AYQ_2^T - B)^T \right) \right] + \beta \frac{\partial}{\partial Y}\left[ \operatorname{tr}\left( (YQ_2^T)(YQ_2^T)^T \right) \right] \\
\nabla_Y g(Y^*) = 0 &\Rightarrow Y^* = 2A^T(AYQ_2^T - B)Q_2 + 2\beta Y Q_2^T Q_2.
\end{aligned}$$

Noting that the columns of $Q_2$ are orthonormal, $Q_2^T Q_2$ simplifies to the identity, and the solution to (9) is found to be similar to that of Tikhonov-regularized least squares with a change of variables.

$$(A^T A + \beta I_{N_r}) Y^* = A^T B Q_2. \tag{14}$$

$Y^*$ can thus be determined via a linear solve, and (9) is solved by finding $X^T = Q_2 (Y^*)^T$. ■

In Theorem 1 we derive a closed-form solution to the constrained optimization problem (9). In the following Corollary we show that forecasts produced by PERC provably enforce the linear constraint. This guarantee is general as it holds true for arbitrary choice of nonlinear function $F$ in the reservoir dynamics and is satisfied both during training and during forecasting.

**Corollary 2 (PERC Guarantees Constraint Enforcement)** *Consider the problem of training a closed-loop reservoir (3) with output readout $\hat{\mathbf{u}}(t)$ defined by (4). Assume $RR^T + \beta I_{N_r}$ is full rank and define $C, Q_2$ the same as Theorem 1. Then the quantity*

$$W_{out}^* = Q_2 (RR^T + \beta I_{N_r})^{-1} R U^T Q_2 \tag{15}$$

*solves the constrained optimization (8). For this choice of $W_{out}$, $C\hat{\mathbf{u}}(t) = 0$ for all $t = 0, 1, \ldots$.*

**Proof** The solution form (15) follows directly from Theorem 1 by plugging in $A = R^T$, $B = U^T$, $X = (W^*_{\text{out}})^T$. Recall that by definition of the output readout (4), $\hat{\mathbf{u}}(\mathbf{t}) = W^*_{\text{out}}\mathbf{r}(t)$. Then since $CW^*_{\text{out}} = 0$, it holds that $C\hat{\mathbf{u}}(\mathbf{t}) = CW^*_{\text{out}}\mathbf{r}(t) = 0$. ■

A summary of the PERC approach to reservoir training with the closed-form solution of Corollary 2 can be found as Algorithm 1. We state the algorithm for a general reservoir $F$ (1).

**Algorithm 1** Physics-Enforced Reservoir Computing (PERC)

**Require:** Training data $U$; Constraint matrix $C$; E equation $F(\mathbf{r}(t), \mathbf{u}(t))$; Hyperparameter $\beta$
Initialize reservoir state $\mathbf{r} \leftarrow \mathbf{0}$
**for** $i = 1$ to $\gamma$ **do**
  Let $\mathbf{u} \leftarrow U[:, i]$
  Update reservoir state: $\mathbf{r} \leftarrow F(\mathbf{r}(t), \mathbf{u}(t))$
**end for**
Trim $U$: $U \leftarrow U[:, \gamma :]$
Initialize $R \leftarrow \mathbf{0} \in \mathbb{R}^{N_r \times N_t}$
**for** $i = 1$ to $N_t - 1$ **do**
  Let $\mathbf{u} \leftarrow U[:, i]$
  Update reservoir state: $\mathbf{r} \leftarrow F(\mathbf{r}(t), \mathbf{u}(t))$
  $R[:, i] \leftarrow \mathbf{r}$
**end for**
Align $U$: $U \leftarrow U[:, 1 :]$
Compute rank of $C$: $n \leftarrow \text{rank}(C)$
Perform QR decomposition on $C^T$: $Q, \tilde{R}, P \leftarrow \text{QR}(C^T)$
Extract null-space basis: $Q_2 \leftarrow Q[:, n :]$
Solve regularized least-squares problem: $(Y^*)^T \leftarrow \text{solve}(RR^T + \beta I, RU^T Q_2)$
Compute optimized weights: $W^*_{\text{out}} \leftarrow Q_2(Y^*)^T$
**return** $W^*_{\text{out}}$

## 4. Numerical Examples

### 4.1. Imposing Integral Constraints in the Kuramoto–Sivashinsky (KS) Equation

The Kuramoto–Sivashinsky (KS) equation is a 4$^{\text{th}}$ order spatiotemporal system characterized by chaotic shock formation induced by a nonlinear wave term. The 1D form of the KS equation is

$$u_t + u_{xxxx} + u_{xx} + uu_x = 0. \tag{16}$$

To collect training data for RC, the above system was discretized into 64 spatial points and evolved through $N_t = 1000$ time steps using a fourth order exponential time-differencing (ETDRK4) scheme (Kassam and Trefethen, 2005) with periodic boundary conditions and on a domain of $(0, L = 22)$. The first 800 steps are used for training and the final 200 for testing. The following integral quantity is conserved in the KS system (Collet et al., 1993) and is used to formulate a constraint error $\epsilon$:

$$\epsilon(t) = \int_0^L u(t, x)dx = 0. \tag{17}$$

This integral can be numerically evaluated via the trapezoidal rule. Since the result of the integration is a scalar, $N_c = 1$ and the constraint matrix simplifies to a row vector $c$ of length $N_u = 64$. Hence,

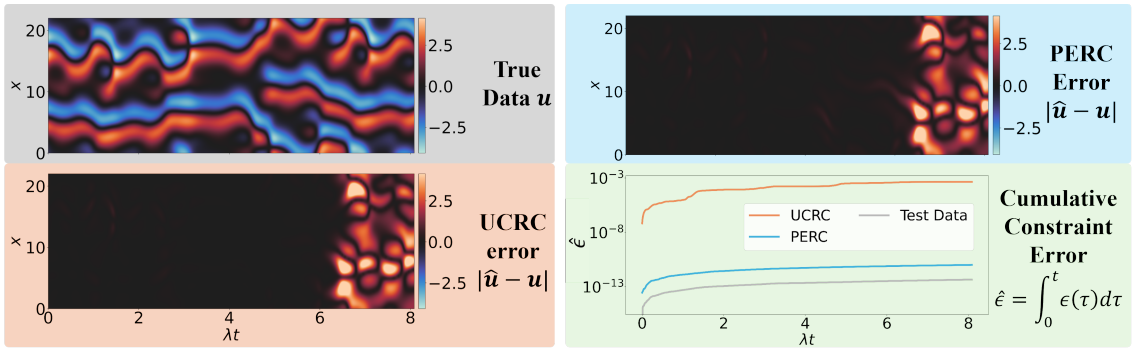


Figure 2: KS testing data (upper right panel), Unconstrained RC forecast error (lower right panel), PERC forecast error (upper right panel), and cumulative constraint enforcement error as defined by (17) (lower right panel).

the integral is approximated by the inner product $\epsilon \approx c^T u$. After implementing both unconstrained RC and PERC on the KS system with $N_r = 5,000$, a comparison of the error in enforcing the constraints is illustrated in Figure 2. As anticipated, PERC imposes the constraints to machine precision, and achieves a constraint error many order of magnitude lower than the traditional RC.

Spatiotemporal plots of the testing data, RC forecasting results, and forecast error can also be seen in Figure 2. Both methods diverge from the true trajectory just above 6 lyapunov times (where the maximum lyapunov exponent is taken to be $\lambda = 0.043$ (Edson et al., 2019)). Despite now training in a constrained space, the forecast of the PERC is unhampered in comparison to the traditional RC. Qualitatively, PERC maintains the so-called "climate replication" property of RC that make it a powerful tool for forecasting chaotic dynamics. The forecast horizon matches the performance from the single reservoir results in (Pathak et al., 2018a); however, Pathak utilizes a nonlinear readout from the linear combination of $\mathbf{r}$ and $\mathbf{r}^2$, doubling the number of trainable parameters. Due to our use a a bias, that breaks the incompatible symmetry between $\tanh$ and KS (Herteux and Räth, 2020), we only utilize a purely linear readout from $\mathbf{r}$.

### 4.2. Imposing Zero-Divergence in Kolmogorov Flow

The Navier-Stokes Equations are the governing equations of fluid flow. For an incompressible, Newtonian liquid, the mass and momentum continuity are described by (18) and (19) where $\mathbf{u}$ is the fluid velocity field, $p$ is the pressure, $\rho$ is the density, and $\nu$ is the kinematic viscosity; gravity and other body forces are neglected:

$$\nabla \cdot \mathbf{u} = 0 \tag{18}$$

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u}. \tag{19}$$

Despite already making some assumptions about the type of fluid flow, these equations still represent a set of complex, nonlinear, coupled partial differential equations (PDEs). Solving these numerically can be prohibitively expensive due to the coupling of the pressure gradient $\nabla p$ and the velocity field $\mathbf{u}$. For 2D incompressible flow, this problem can be alleviated by transforming the velocity field into a scalar field known as the stream function $\psi$ defined in terms of $x$ and $y$ velocity components $u$ and $v$, respectively: $u = \psi_y$ and $v = -\psi_x$. Lastly, the vorticity $\omega$ is defined as the

curl of the velocity field $\omega = \nabla \times \mathbf{u}$. Applying these substitutions to Equation 19 results in the following time evolution equations:

$$\omega = -\nabla^2 \psi \tag{20a}$$

$$\omega_t = -\psi_y \omega_x + \psi_x \omega_y + \nu \nabla^2 \omega \tag{20b}$$

We investigate the Kolmogorov flow, which is the Navier-Stokes equations on a doubly periodic domain with constant wavenumber body forcing (Yin et al., 2004; Dresdner et al., 2022). This flow is solved with a pseudo-spectral solver (Lagemann et al., 2025) and a fourth order implicit/explicit Runge-Kutta/Crank-Nicolson Scheme. We flatten and stack the velocity fields from a $32 \times 32$ grid to form $U$. Although this is a small grid for a fluid simulation, the size of this spatiotemporal grid ($N_u = 32 \times 32 \times 2 = 2048$) is rarely approached with RC. Due to this formulation, the flow fields should be divergence free, thus posing an excellent testbed constraint for the PERC framework. To enforce the RC output to be divergence free, the constraint definition matrix $C$ approximates the divergence operator $\nabla \cdot$ and is set to a finite difference stencil of size $1024 \times 2048$. Therefore, $C$ imposes one constraint for every point in the domain. In this case, we use a second order Taylor approximation for the derivatives. However, a spectral stencil can also be enforced given the periodic boundaries.

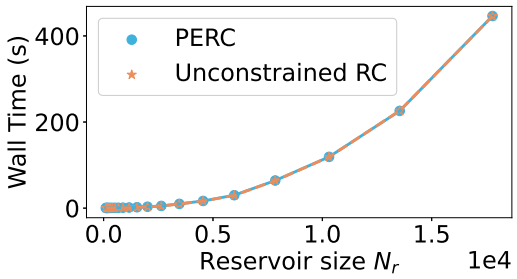


Figure 3: Empirically measured wall time for training both PERC and unconstrained RC on Kolmogorov flow across varying reservoir sizes. Each time is averaged across 5 trials.

The size of the Kolmogorov state space is over an order of magnitude larger than that of KS; therefore, the reservoir dimension was increased to $N_r = 20,000$. The resulting forecasts for both PERC and unconstrained RC methods are shown in (4). It is clear that the divergence of the PERC forecast is enforced to be zero at every point in the domain, whereas the unconstrained RC forecast fails to satisfy incompressibility.

The Kolmogorov flow is chaotic, and so both forecasts diverge from the nominal trajectory. However, neither forecast is unstable and both qualitatively continue to exhibit climate replication. We also empirically note that as $N_r$ increases, the unconstrained RC and PERC forecasts converge to each other. We infer that the extreme events, characterized by brief periods of high energy dissipation, are the primary drivers causing the RC forecast to diverge. Extreme events have been shown to be challenging to predict (Vela-Martín, 2024; Yuan and Lozano-Durán, 2024), and result in massive changes to the Kolmogorov flow state. Hence, even a small over or under prediction of an extreme event can hamper the forecast horizon substantially. We conclude that both RC models are unable to sufficiently predict extreme events. Future efforts in forecasting Kolmogorov flow should consider a focus on extreme events.

Finally, we confirm that PERC has negligible computational overhead compared to traditional RC by plotting the wall times of training both algorithms over varying reservoir sizes in Figure 3. $C$ is sparse and its size is a constant for all values of $N_r$; therefore, computing its QR factorization takes comparably minimal time. The result is a constrained algorithm with near identical runtime to the unconstrained algorithm. While the sparsity of $C$ is not guaranteed in general, it is very likely that finite difference operators will be sparse and computational time will be minimally impacted.

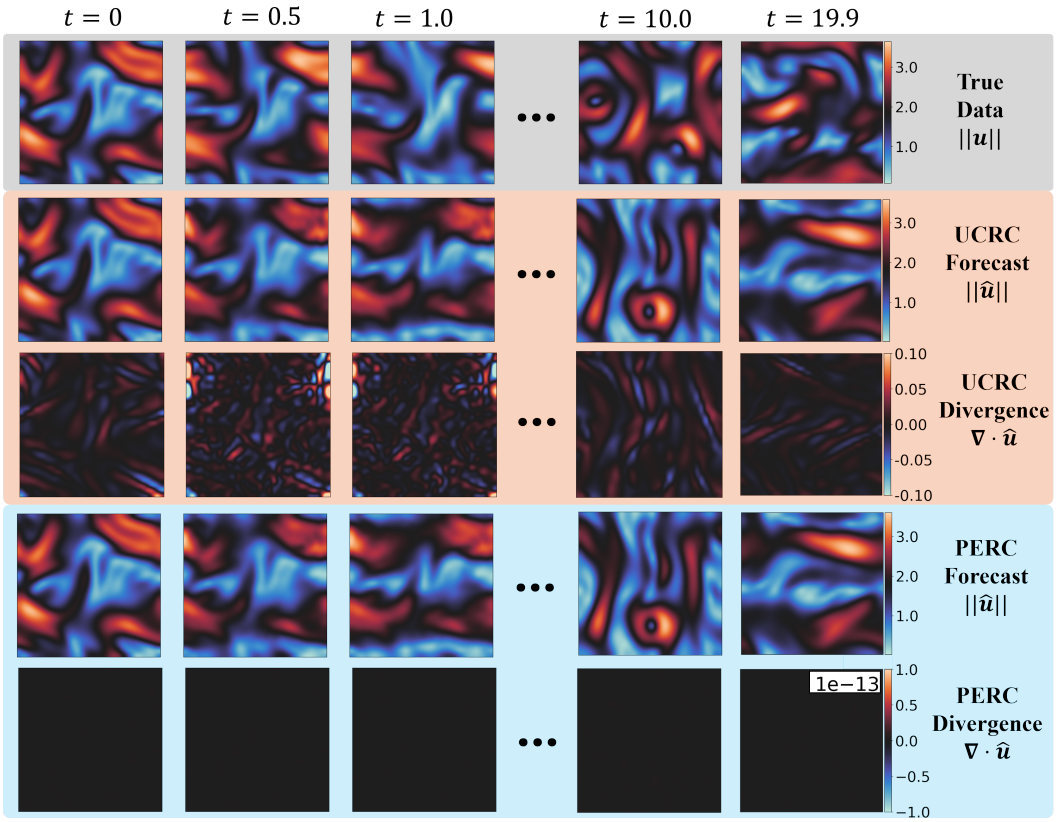


Figure 4: Unconstrained RC (UCRC) and PERC forecasts of the Kolmogorov flow fluid system stacked below tiles of the simulated testing data. Each tile is the magnitude of the velocity across a $32 \times 32$ domain. Divergence fields are computed for each of the forecasts, and the PERC is shown to locally and globally enforce zero-divergence to machine precision.

## 5. Conclusion

The proposed PERC algorithm enforces hard physical constraints of the form $C\mathbf{u}(t) = 0$ to machine precision with minimal impacts on training speed and forecast horizon. This constraint is enforced in a single-step, analytic solve. We build on soft constraint algorithms, such as PI-ESNs (Doan et al., 2020), to enable hard constraints that are guaranteed to hold in PERC forecasts. However, PERC does not allow for a generalized differential constraint and only admits linear, homogeneous constraints. Therefore, future works will focus on modifying the PERC algorithm to more readily accept nonlinear, temporal, and non-homogeneuous constraints. Furthermore, parallel localization (Pathak et al., 2018a) can be employed to enable physics-enforced large-scale RC predictions of spatiotemporal systems to expand PERC's tractability to a wider array of high dimensional problems. Preliminary results show improvement of forecast horizons for the Kolmogorov Flow while using constraints, as long as the training and testing sets come from the same trajectory. However, further robust hyperparameter tuning and model evaluation is required to draw a definitive conclusion. In future work, we will also investigate whether enforcing hard constraints with PERC may improve forecasting performance in optimized ESN architectures and in the low data limit, as has been observed with other physics based approaches.

## Acknowledgments

We acknowledge support from the National Science Foundation AI Institute in Dynamic Systems (grant no. 2112085) and the Boeing Company. We thank Sajeda Mokbel for help using her implmentation of the Kolmogorov flow in HydroGym. We also thank Dr. Heather Wilbur for the insightful discussions about constrained optimization, and Dr. Ling-Wei Kong for his advise on tuning RC hyperparameters.

## References

Amir Ali Ahmadi and Bachir El Khadir. Learning dynamical systems with side information. In *Learning for Dynamics and Control*, pages 718–727. PMLR, 2020.

Troy Arcomano, Istvan Szunyogh, Jaideep Pathak, Alexander Wikner, Brian R. Hunt, and Edward Ott. A Machine Learning-Based Global Atmospheric Forecast Model. *Geophysical Research Letters*, 47(9):e2020GL087776, 2020. ISSN 1944-8007. doi: 10.1029/2020GL087776.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, September 1995. ISSN 1064-8275. doi: 10.1137/0916069. URL https://epubs.siam.org/doi/10.1137/0916069. Publisher: Society for Industrial and Applied Mathematics.

Pierre Collet, Jean-Pierre Eckmann, Henri Epstein, and Joachim Stubbe. A global attracting set for the Kuramoto-Sivashinsky equation. *Communications in Mathematical Physics*, 152(1):203–214, February 1993. ISSN 1432-0916. doi: 10.1007/BF02097064. URL https://doi.org/10.1007/BF02097064.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Deep Differential Equations Workshop*, 2020.

N. A. K. Doan, W. Polifke, and L. Magri. Physics-informed echo state networks. *Journal of Computational Science*, 47:101237, November 2020. ISSN 1877-7503. doi: 10.1016/j.jocs.2020.101237.

N. a. K. Doan, W. Polifke, and L. Magri. Short- and long-term predictions of chaotic flows and extreme events: A physics-constrained reservoir computing approach. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 477(2253):20210135, September 2021a. doi: 10.1098/rspa.2021.0135.

Nguyen Anh Khoa Doan, Wolfgang Polifke, and Luca Magri. Auto-Encoded Reservoir Computing for Turbulence Learning. In Maciej Paszynski, Dieter Kranzlmüller, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M.A. Sloot, editors, *Computational Science – ICCS 2021*, pages 344–351, Cham, 2021b. Springer International Publishing. ISBN 978-3-030-77977-1. doi: 10.1007/978-3-030-77977-1_27.

Gideon Dresdner, Dmitrii Kochkov, Peter Norgaard, Leonardo Zepeda-Núñez, Jamie A. Smith, Michael P. Brenner, and Stephan Hoyer. Learning to correct spectral methods for simulating turbulent flows. 2022. doi: 10.48550/ARXIV.2207.00556. URL https://arxiv.org/abs/2207.00556%7D.

Russell A. Edson, J. E. Bunder, Trent W. Mattner, and A. J. Roberts. LYAPUNOV EXPONENTS OF THE KURAMOTO–SIVASHINSKY PDE. *The ANZIAM Journal*, 61(3):270–285, July 2019. ISSN 1446-1811, 1446-8735. doi: 10.1017/S1446181119000105.

Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 1551-6709. doi: 10.1207/s15516709cog1402_1.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International conference on machine learning*, pages 3318–3328. PMLR, 2021.

Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian Neural Networks, September 2019. URL http://arxiv.org/abs/1906.01563. arXiv:1906.01563.

Aaron Griffith, Andrew Pomerance, and Daniel J. Gauthier. Forecasting chaotic systems with very low connectivity reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):123108, December 2019. ISSN 1054-1500. doi: 10.1063/1.5120710.

Joschka Herteux and Christoph Räth. Breaking symmetries of the reservoir equations in echo state networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(12):123142, December 2020. ISSN 1054-1500. doi: 10.1063/5.0028993. URL https://doi.org/10.1063/5.0028993.

Herbert Jaeger. Echo state network. *Scholarpedia*, 2(9):2330, September 2007. ISSN 1941-6016. doi: 10.4249/scholarpedia.2330.

Herbert Jaeger and Harald Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, April 2004. doi: 10.1126/science.1091277.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, June 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5.

Aly-Khan Kassam and Lloyd N. Trefethen. Fourth-Order Time-Stepping for Stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, January 2005. ISSN 1064-8275. doi: 10.1137/S1064827502410633.

Christian Lagemann, Ludger Paehler, Jared Callaham, Sajeda Mokbel, Samuel Ahnert, Kai Lagemann, Esther Lagemann, Nikolaus A. Adams, and Steven L. Brunton. Hydrogym: A platform for reinforcement learning control in fluid dynamics. In *Proceedings of the 7th Annual Learning for Dynamics and Control Conference (L4DC)*, 2025. Under review.

Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

Ziming Liu, Patrick Obin Sturm, Saketh Bharadwaj, Sam J. Silva, and Max Tegmark. Interpretable conservation laws as sparse invariants. *Physical Review E*, 109(2):L023301, February 2024.

doi: 10.1103/PhysRevE.109.L023301. URL https://link.aps.org/doi/10.1103/PhysRevE.109.L023301. Publisher: American Physical Society.

J.-C. Loiseau and S. L. Brunton. Constrained sparse Galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, 2018.

Zhixin Lu and Danielle S Bassett. Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(6), 2020.

Zhixin Lu, Brian R. Hunt, and Edward Ott. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):061104, June 2018. ISSN 1054-1500. doi: 10.1063/1.5039508.

Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560, November 2002. ISSN 0899-7667, 1530-888X. doi: 10.1162/089976602760407955.

Yslam D Mammedov, Ezutah Udoncy Olugu, and Guleid A Farah. Weather forecasting based on data-driven and physics-informed reservoir computing models. *Environmental Science and Pollution Research*, pages 1–14, 2022.

Arvind T. Mohan, Nicholas Lubbers, Misha Chertkov, and Daniel Livescu. Embedding hard physical constraints in neural network coarse-graining of three-dimensional turbulence. *Physical Review Fluids*, 8(1):014604, January 2023. doi: 10.1103/PhysRevFluids.8.014604. URL https://link.aps.org/doi/10.1103/PhysRevFluids.8.014604. Publisher: American Physical Society.

Kohei Nakajima and Ingo Fischer. *Reservoir Computing*. Springer, 2021.

Akane Ohkubo and Masanobu Inubushi. Reservoir computing with generalized readout based on generalized synchronization. *Scientific reports*, 14(1):30918, 2024.

Samuel E. Otto, Nicholas Zolman, J. Nathan Kutz, and Steven L. Brunton. A Unified Framework to Enforce, Discover, and Promote Symmetry in Machine Learning, August 2024. URL http://arxiv.org/abs/2311.00212. arXiv:2311.00212.

Sandeep Pandey and Jörg Schumacher. Reservoir computing model of two-dimensional turbulent convection. *Physical Review Fluids*, 5(11):113506, November 2020. doi: 10.1103/PhysRevFluids.5.113506.

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, January 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, April 2018b. ISSN 1054-1500. doi: 10.1063/1.5028373.

S. G. Penny, T. A. Smith, T.-C. Chen, J. A. Platt, H.-Y. Lin, M. Goodliff, and H. D. I. Abarbanel. Integrating Recurrent Neural Networks With Data Assimilation for Scalable Data-Driven State Estimation. *Journal of Advances in Modeling Earth Systems*, 14(3):e2021MS002843, 2022. ISSN 1942-2466. doi: 10.1029/2021MS002843.

Adolfo Perrusquía and Weisi Guo. Reservoir computing for drone trajectory intent prediction: a physics informed approach. *IEEE Transactions on Cybernetics*, 2024.

Jason A Platt, Adrian Wong, Randall Clark, Stephen G Penny, and Henry DI Abarbanel. Forecasting using reservoir computing: The role of generalized synchronization. *arXiv preprint arXiv:2102.08930*, 2021a.

Jason A Platt, Adrian Wong, Randall Clark, Stephen G Penny, and Henry DI Abarbanel. Robust forecasting using predictive generalized synchronization in reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(12), 2021b.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent Advances in Physical Reservoir Computing: A Review. *Neural Networks*, 115:100–123, July 2019. ISSN 08936080. doi: 10.1016/j.neunet.2019.03.005.

A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. Springer Netherlands, Dordrecht, 1977. ISBN 978-90-481-4583-6 978-94-015-8480-7. doi: 10.1007/978-94-015-8480-7.

Dima Tretiak, Arvind T. Mohan, and Daniel Livescu. Physics-Constrained Generative Adversarial Networks for 3D Turbulence, December 2022. URL http://arxiv.org/abs/2212.00217. arXiv:2212.00217.

Alberto Vela-Martín. Complexity of extreme-event prediction in turbulent flows. *Physical Review Fluids*, 9(10):104603, October 2024. doi: 10.1103/PhysRevFluids.9.104603. URL https://link.aps.org/doi/10.1103/PhysRevFluids.9.104603. Publisher: American Physical Society.

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, June 2020. ISSN 0893-6080. doi: 10.1016/j.neunet.2020.02.016.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40):eabi8605, October 2021. ISSN 2375-2548. doi: 10.1126/sciadv.abi8605.

Jianke Yang, Wang Rao, Nima Dehmamy, Robin Walters, and Rose Yu. Symmetry-informed governing equation discovery. *arXiv preprint arXiv:2405.16756*, 2024.

Izzet B Yildiz, Herbert Jaeger, and Stefan J Kiebel. Re-visiting the echo state property. *Neural networks*, 35:1–9, 2012.

Z. Yin, H.J.H. Clercx, and D.C. Montgomery. An easily implemented task-based parallel scheme for the fourier pseudospectral solver applied to 2d navier–stokes turbulence. *Computers & Fluids*, 33 (4):509–520, 2004. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2003.06.003%7D.

Yuan Yuan and Adrián Lozano-Durán. Limits to extreme event forecasting in chaotic systems. *Physica D: Nonlinear Phenomena*, 467:134246, November 2024. ISSN 0167-2789. doi: 10.1016/j.physd.2024.134246. URL https://www.sciencedirect.com/science/article/pii/S0167278924001970.

Yaofeng Desmond Zhong and Naomi Leonard. Unsupervised learning of Lagrangian dynamics from images for prediction and control. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/79f56e5e3e0e999b3c139f225838d41f-Paper.pdf,ConferenceProceedingshttps://arxiv.org/pdf/2007.01926.pdf,arXiv.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning Hamiltonian dynamics with control. In *International Conference on Learning Representations (ICLR)*, 2020.