# Stability analysis of chaotic systems from data

**Georgios Margazoglou** · **Luca Magri**

**Abstract** The prediction of the temporal dynamics of chaotic systems is challenging because infinitesimal perturbations grow exponentially. The analysis of the dynamics of infinitesimal perturbations is the subject of stability analysis. In stability analysis, we linearize the equations of the dynamical system around a reference point and compute the properties of the tangent space (i.e. the Jacobian). The main goal of this paper is to propose a method that infers the Jacobian, thus, the stability properties, from observables (data). First, we propose the echo state network (ESN) with the Recycle validation as a tool to accurately infer the chaotic dynamics from data. Second, we mathematically derive the Jacobian of the echo state network, which provides the evolution of infinitesimal perturbations. Third, we analyse the stability properties of the Jacobian inferred from the ESN and compare them with the benchmark results obtained by linearizing the equations. The ESN correctly infers the nonlinear solution and its tangent space with negligible numerical errors. In detail, we compute from data only (i) the long-term statistics of the chaotic state; (ii) the covariant Lyapunov vectors; (iii) the Lyapunov spectrum; (iv) the finite-time Lyapunov exponents; (v) and the angles between the stable, neutral, and unstable splittings of the tangent space (the degree of hyperbolicity of the attractor). This work opens up new opportunities for the computation of stability properties of nonlinear systems from data, instead of equations.

G. Margazoglou (✉)· L. Magri
Aeronautics Department, Imperial College London, South Kensington Campus, London SW7 2AZ, UK
e-mail: g.margazoglou@imperial.ac.uk

L. Magri
The Alan Turing Institute, 96 Euston Road, NW1 2DB London, UK
e-mail: l.magri@imperial.ac.uk

## 1 Introduction

Chaotic behaviour has been observed and extensively studied in diverse scientific fields, initially in meteorology [1] and later in physics [2,3], chemistry, biology and engineering [4] to name a few. Chaos appears from deterministic nonlinear equations in the form of sensitivity to initial conditions, aperiodic behaviour, and short predictability. A successful mathematical tool for the analysis of chaos is provided by stability analysis. By applying infinitesimal perturbations to a system's trajectory, we can classify its stability along different

directions and compute the properties of its linear tangent space.

Stability analysis relies on the linearization of the dynamical equations, which requires the Jacobian of the system. The key quantities that characterize chaotic dynamics, and many other related physical properties, such as dynamical entropies and fractal dimensions, are the Lyapunov Exponents (LEs) [5,6], which are the eigenvalues of the Oseledets matrix [7]. There are several numerical methods to extract the LEs based on the Gram–Schmidt orthogonalization procedure [6,8,9]. The relevant eigenvectors are the corresponding Lyapunov vectors that constitute a coordinate dependent orthogonal basis of the linear tangent space. Instead, an intrinsic and norm-independent basis, which is also time invariant and covariant with the dynamics is given by the covariant Lyapunov vectors (CLVs). Crucially, CLVs are able to provide information on the local structure of chaotic attractors [10]. This viewpoint allows the study of an attractor's topology with the occurrence of critical transitions [11–14], paving the way for CLVs to be considered as precursors to such phenomena.

The previous exposition is traditionally related to model-based approaches, as it relies on the knowledge of a system's dynamical equations. However, studying the stability properties of observed data, where equations are not necessarily known, is hard; there are few approaches, e.g. [15,16], relying on the delayed coordinates attractor reconstruction by Takens [17]. The recent breakthrough of data-driven (model-free) approaches poses the reasonable question: Can we use the rich knowledge of dynamical systems theory for model-free approaches? Indeed, although at early steps, the use of advanced machine learning (ML) techniques for complex systems has shown promising potential in applications ranging from weather and climate prediction and classification [18–20] to fluid flows prediction and optimization [21–23], among others. The overarching goal of this work is to propose a machine learning approach to accurately learn and infer the ergodic properties of prototypical chaotic attractors, and in particular to extract LEs and CLVs from data.

The recurrent neural networks (RNNs) constitute a promising type of ML to address chaotic behaviour. Thanks to their architecture, the RNNs are suitable for processing sequential data, typically encountered in speech and language recognition, or time-series prediction [24]. In particular, they are proven to be universal approximators [25,26] and are able to capture long-term temporal patterns, i.e. they possess memory. A key piece of their architecture is that they maintain a hidden state that evolves dynamically, effectively allowing the RNNs to be treated as dynamical systems, and in particular as discrete neural differential equations [27]. Thus, RNNs lend themselves to being analysed with dynamical systems theory, allowing the study of stability properties from the dynamics they have learned. By exploiting this here, we derive the RNN's Jacobian and infer the linear dynamics from data.

Recently, there have been significant advancements in employing RNNs to learn chaotic dynamics [28–36], where two core objectives are studied: (1) the time-accurate prediction of chaotic fluctuations and maximization of the prediction horizon and (2) accurately learning the ergodic properties of chaotic attractors. The first objective has been addressed by one of the co-authors in [34–36] for several prototypical chaotic dynamical systems using the same RNN architecture as the present work. Here we address the second objective by extending the recent works [29,30,32], where the LEs of the Lorenz 63 [1] and the one-dimensional Kuramoto–Sivashinsky equation [37] were retrieved from trained RNNs.

In this work, we employ a specific architecture of the RNN, a type of reservoir computer, the echo state network (ESN) [38] and train it with a diverse set of four prototypical chaotic attractors. The objective of this paper is twofold; first the accurate learning and inference of the ergodic properties of the chaotic attractors by the ESN. This is accomplished by thoroughly comparing the long-term statistics of (i) degrees of freedom, (ii) LEs, (iii) finite-time LEs, and (iv) angles of the CLVs. Second, by comparing the distribution of (i) finite-time LEs and (ii) angles of CLVs on the topology of the attractor, providing a strict test of the ESN's capability to accurately learn intrinsic chaotic properties.

The paper is organized as follows: Section 2 presents the necessary tools for our study. In particular, Sect. 2.1 provides a brief introduction to the relevant concepts and quantities from dynamical systems, such as LEs and CLVs. Then, Sect. 2.2 describes the architecture of the ESN, while Sect. 2.3 its validation strategies. Section 3 presents our main results, which are divided into two subsections; Sect. 3.1 devoted in low-dimensional systems, namely the Lorenz 63 [1] and Rössler [39] attractors; and Sect. 3.2 showing results on the Charney-DeVore [40] and the Lorenz 96 [41]

attractors. Finally, we summarize our results and provide future perspectives in the conclusions in Sect. 4. The appendix A presents the two algorithms to extract the LEs and CLVs from the ESN. Appendix B provides further tests on the robustness of the methodology.

## 2 Background

In the following two subsections, we summarize the key theory that underpins the stability of chaotic systems (Sect. 2.1) and reservoir computers (Sect. 2.2).

### 2.1 Stability of chaotic systems

We consider a state $\boldsymbol{x}(t) \in \mathbb{R}^D$ with $D$ degrees of freedom, which is governed by a set of nonlinear ordinary differential equations

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = f(\boldsymbol{x}), \tag{1}$$

where $f(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^D$ is a smooth nonlinear function. Equation (1) defines an autonomous dynamical system. Hence, the dynamical system exists in a phase space of dimension $D$, equipped with a suitable metric, and is associated with a certain measure $\mu$ that we assume to be preserved (invariant). To investigate the stability of the dynamical system (1), we perturb the state by first-order perturbations as

$$\boldsymbol{x} + \boldsymbol{u}, \quad \boldsymbol{x} \sim \mathcal{O}(1), \quad \boldsymbol{u} \sim \mathcal{O}(\epsilon), \quad \epsilon \to 0. \tag{2}$$

By substituting decomposition (2) into (1) and collecting the first-order terms $\sim \mathcal{O}(\epsilon)$, we obtain the governing equation for the first-order perturbations (i.e. linear dynamics)

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{J}(\boldsymbol{x}(t))\boldsymbol{u}, \tag{3}$$

where $J_{ij} = \frac{\partial f_i(x)}{\partial x_j} \in \mathbb{R}^{D \times D}$ are the components of the Jacobian, $\boldsymbol{J}(\boldsymbol{x}(t))$, which is in general time-dependent. The perturbations $\boldsymbol{u}$ evolve on the linear tangent space at each point $\boldsymbol{x}(t)$. The goal of stability analysis is to compute the growth rate of infinitesimal perturbations,

which is achieved by computing the Lyapunov exponents and a basis of the tangent space with the covariant Lyapunov Vectors. To do so, we numerically time-march $K \leq D$ tangent vectors, $\boldsymbol{u}_i \in \mathbb{R}^D$, as columns of the matrix $\boldsymbol{U} \in \mathbb{R}^{D \times K}$, $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K]$

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{J}(\boldsymbol{x}(t))\boldsymbol{U}. \tag{4}$$

Geometrically, Eq. (4) describes the tangent space around the state $\boldsymbol{x}(t)$. Starting from $\boldsymbol{x}(t = t_0) = \boldsymbol{x}_0$ and $\boldsymbol{U}(t_0) = \mathbb{I}$, Eqs. (1) and (4) are numerically solved with a time integrator. As explained in the subsequent paragraphs, in a chaotic system, almost all nearby trajectories diverge exponentially fast with an average rate equal to the leading Lyapunov exponent. Hence, the tangent vectors align exponentially fast with the leading Lyapunov vector, $\boldsymbol{u}_1$. ('Almost all' means that the set of perturbations that do not grow with the largest Lyapunov exponents has a zero measure). To circumvent this numerical issue, it is necessary to periodically orthonormalize the tangent space basis during time evolution, using a QR-decomposition of $\boldsymbol{U}$, as $\boldsymbol{U}(t) = \boldsymbol{Q}(t)\boldsymbol{R}(t, \Delta t)$ (see [8,9]) and updating the columns of $\boldsymbol{U}$ with the columns of $\boldsymbol{Q}$, i.e. $\boldsymbol{U} \leftarrow \boldsymbol{Q}$. The matrix $\boldsymbol{R}(t, \Delta t) \in \mathbb{R}^{K \times K}$ is upper-triangular and its diagonal elements $[\boldsymbol{R}]_{i,i}$ are the local growth rates over a time span $\Delta t$ of the (now) orthonormal vectors $\boldsymbol{U}$, which are also known as backward Gram–Schmidt vectors (GSVs) [10,42]. The Lyapunov spectrum is given by[1]

$$\lambda_i = \lim_{T \to \infty} \frac{1}{T} \int_{t_0}^{T} \ln[\boldsymbol{R}(t, \Delta t)]_{i,i} \mathrm{d}t. \tag{5}$$

The algorithm 1 in the appendix A is a pseudocode for the calculation of the LEs for the ESN following [29,32]. The sign of the Lyapunov exponents indicates the type of the attractor. If the leading exponent is negative, $\lambda_1 < 0$, the attractor is a fixed point. If $\lambda_1 = 0$, and the remaining exponents are negative, the attractor is a periodic orbit. If at least a Lyapunov exponent

---

[1] The Oseledets' theorem [6,7,10] establishes the existence of Lyapunov exponents (LEs) for a generic set of orbits under fairly general assumptions. In particular, the Oseledets' theorem enables the extension of Lyapunov stability analysis to any trajectory of a dynamical system defined on a Riemannian manifold of dimension $N$ and equipped with a suitable metric, including fixed points and periodic orbits.

is positive, $\lambda_1 > 0$, the attractor is chaotic. In chaotic systems, the Lyapunov time $\tau_\lambda = \frac{1}{\lambda_1}$ defines a characteristic timescale for two nearby orbits to separate, which gives a scale of the system's predictability horizon [43].

The GSVs, $U$, constitute a norm-dependent orthonormal basis, which is not time-reversible, due to the frequent orthogonalizations via the QR decomposition. Instead, the covariant Lyapunov vectors (CLVs) $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K]$ (each CLV $\mathbf{v}_i \in \mathbb{R}^D$ is a column of $V$) form a norm-independent and time-invariant basis of the tangent space, which is covariant with the dynamics. The latter features of the CLVs, which are not possessed by the GSVs, allow us to examine individual expanding and contracting directions of a given dynamical system, thus providing an intrinsic geometrical interpretation of the attractor [6,10], as well as a hierarchical decomposition of spatiotemporal chaos, thanks to their generic localization in physical space [42]. Each bounded nonzero CLV, i.e. $0 < ||\mathbf{v}_i|| < \infty$, satisfies the following equation:

$$\frac{d\mathbf{v}_i}{dt} = J(x(t))\mathbf{v}_i - \lambda_i \mathbf{v}_i, \tag{6}$$

which shows that the CLV is evolved by the tangent dynamics $J(x(t))\mathbf{v}_i$, while the extra term $-\lambda_i \mathbf{v}_i$ guarantees that its norm is bounded [44]. The name "covariant" means that the $i$th CLV at time $t_1$, $\mathbf{v}_i(x(t_1))$, maps at $\mathbf{v}_i(x(t_2))$ at time $t_2$, and vice versa. Mathematically, if $M(t, \Delta t) = \exp(\int_t^{t+\Delta t} J(x, \tau)d\tau)$ is the system's tangent evolution operator (which contains a path-ordered exponential), covariant means $M(t, \Delta t)\mathbf{v}_i(t) = \mathbf{v}_i(t + \Delta t)$; time-invariance of CLVs naturally arises from the previous expression, as $M(t, -\Delta t)\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t)$. If the Lyapunov spectrum is non-degenerate (such as for the cases considered here), each CLV $\mathbf{v}_i$ is associated with the Lyapunov exponent $\lambda_i$ and is uniquely defined (up to a phase).

An important subclass of chaotic systems are uniformly hyperbolic systems, which have a uniform splitting between expanding and contracting directions, i.e. there are no tangencies between the unstable, neutral, and stable subspaces [45] that form the tangent space. Because of their simple geometrical structure, many theoretical tools have been developed in recent years. Hyperbolic systems have structurally stable dynamics and linear response, meaning that their statistics vary smoothly with parameter variations [46]. In prac-

tice, violations of hyperbolicity are commonly reported in the literature [44,47,48], whereas true hyperbolic systems are rare [49]. Thanks to the chaotic hypothesis [5,50,51], high-dimensional chaotic systems can be practically treated as hyperbolic systems, i.e. using techniques developed for hyperbolic systems, regardless of hyperbolicity violations. This is because many convenient statistical properties of uniformly hyperbolic systems, such as ergodicity, existence of physical invariant measures, exponential mixing and well-defined time averages with large deviation laws [52, 53], can be found in the macroscopic scale dynamics of certain large non-uniformly hyperbolic systems [46].

An application of CLVs is to assess the degree of hyperbolicity of the underlying chaotic dynamics. The tangent space of hyperbolic systems, at each point $x$, can be directly decomposed into three invariant subspaces, $E_x^U \oplus E_x^N \oplus E_x^S$. Here $E_x^U$ is the unstable subspace composed by the CLVs associated with positive LEs, $E_x^N$ is the neutral subspace spanned by the CLVs associated with the zero LEs, and $E_x^S$ is the stable subspace spanned by the CLVs associated with negative LEs. In hyperbolic systems, the distribution of angles between subspaces is bounded away from zero. In Sect. 3, we will study in detail the angles $\theta_{U,N}$, $\theta_{U,S}$, and $\theta_{N,S}$ between pairs of the subspaces, and compare the ability of the ESN to accurately learn both the long-term statistics, and the phase space finite-time variability of the angles. Because the GSVs are mutually orthogonal, they cannot assess the degree of hyperbolicity of the attractor. Moreover, CLVs are key to the optimization of chaotic acoustic oscillations [44], as well as in reduced-order modelling [54]; they can reveal two uncoupled subspaces of the tangent space, one that comprises the physical modes carrying the relevant information of the trajectory, and another composed of strongly decaying spurious modes [10]. Two recent attempts to extract CLVs from data-driven approaches, which do not employ a neural network, can be found in [55,56].

We explain the algorithm we employ to compute the CLVs; for further details, we refer the interested reader to [10,42,44]. The GSVs are generated by numerically solving Eqs. (1) and (4) simultaneously and performing a QR-decomposition every $m$ timesteps. In this way, after a time-lapse $\Delta t$, the GSVs at time $t + \Delta t$ are given by:

$$M(t, \Delta t)U(t) = U(t + \Delta t)R(t, \Delta t). \tag{7}$$

We can define the CLVs $V(t)$ in terms of the GSVs as

$$V(t) = U(t)C(t), \tag{8}$$

where $C$ is an upper triangular matrix that contains the CLV expansion coefficients, $[C(t)]_{ji} = c^{j,i}(t)$, for $j \leq i$. Hence, the objective is to calculate $C(t)$. Because the CLVs have by choice a unit norm, each column of the matrix $C$ has to be normalized independently, i.e. $\sum_{j=1}^{i}(c_{j,i}(t))^2 = 1, \forall i$.

We start by writing the evolution equation of the CLVs as

$$M(t, \Delta t)V(t) = V(t + \Delta t)\mathbf{D}(t, \Delta t). \tag{9}$$

We can re-write Eq. (9) via Eq. (8)

$$
\begin{aligned}
U(t + \Delta t)&C(t + \Delta t)\mathbf{D}(t, \Delta t) \\
&= M(t, \Delta t)U(t)C(t) \\
&= U(t + \Delta t)R(t, \Delta t)C(t), 
\end{aligned} \tag{10}
$$

and solve with respect to $C(t)$

$$C(t) = R^{-1}(t, \Delta t)C(t + \Delta t)\mathbf{D}(t, \Delta t). \tag{11}$$

This equation is evolved backwards in time starting from the end of the forward-in-time simulation. We employ the `solve_triangular` routine of scipy [57] to invert $R(t, \Delta t)$ and solve with respect to $C(t)$. The $C$ and $\mathbf{D}$ matrices are initialized to the identity matrix $\mathbb{I}$. We leave a sufficient spin-up and spin-down transient time at the beginning and end of our total time window, before we compute the CLVs via Eq. (8), to ensure that they are converged. The algorithm 2 in the appendix A is a pseudocode for the calculation of the CLVs.

To estimate the expansions and contractions of the tangent space on finite-time intervals of length $\Delta t = t_2 - t_1$, we compute the finite-time Lyapunov exponents (FTLEs) as $\Lambda_i = \frac{1}{\Delta t} \ln[R]_{i,i}$. Hence, $\lambda_i$ is the long-time average of $\Lambda_i$. The FTLE $\Lambda_1$ physically quantifies the exponential growth rate of a vector $u_1$ during the time interval $\Delta t$; therefore, $\Lambda_2$ quantifies the exponential growth rate of the vector $u_2$ that is orthogonal to $u_1$ by construction. Hence, as the GSVs form an orthogonal basis, looking at individual FTLEs for $\Lambda_i$, $i \geq 2$, lacks a physical meaning. Instead, the sum of

the first $n$ FTLEs is a growth rate in $\Delta t$ for a typical $n$-dimensional volume $\mathrm{Vol}_n$ in the tangent space [9,58]

$$
\begin{aligned}
\sum_{i=1}^{n} \Lambda_i &= \frac{1}{\Delta t} \sum_{i=1}^{n} \ln[R(t, \Delta t)]_{i,i} = \frac{1}{\Delta t} \ln \prod_{i=1}^{n} [R(t, \Delta t)]_{i,i} \\
&= \frac{1}{\Delta t} \ln \mathrm{Vol}_n(\Delta t).
\end{aligned} \tag{12}
$$

Accordingly, the diagonal matrix $\mathbf{D}(t, \Delta t)$ contains the CLV local growth factors of $\gamma_i(t, \Delta t) = ||M(t, \Delta t)v_i(t)||$, i.e. $[\mathbf{D}(t, \Delta t)]_{i,j} = \delta_{i,j}\gamma_i(t, \Delta t)$. We can extract the finite-time covariant Lyapunov exponents (FTCLEs) from the logarithm of these growth factors for a time interval $\Delta t$

$$\Lambda_i^c = \frac{1}{\Delta t} \ln[\mathbf{D}]_{ii}. \tag{13}$$

Each FTCLE quantifies a finite-time exponential expansion or contraction rate along a covariant direction given by $v_i$. Hence, each individual FTCLE has a physical interpretation, in contrast to the FTLEs, as explained before. On the other hand, now the sums of FTCLEs lack a physical meaning [58]. The long-time average of the FTCLEs is equal to the Lyapunov exponents, $\lambda_i = \lim_{T \to \infty} \frac{1}{T} \int_{t_0}^{T} \Lambda_i^c(t)\mathrm{d}t$.

## 2.2 Echo state network

The solution of a dynamical system is a time series. From a data analysis point of view, a time series is a sequentially ordered set of values, in which the order is provided by time. In a discrete setting, time can be thought of as an ordering index. For sequential data, and hence, time series, recurrent neural networks (RNNs) are designed to infer the temporal dynamics through their internal hidden state. However, training RNNs, such as long short-term memory (LSTM) [59] networks and gated recurrent units (GRUs) [60], requires back-propagation through time, which can be a demanding computational task due to the long-lasting time dependencies of the hidden states [61]. This issue is overcome by echo state networks (ESNs) [38,62], a RNN that is a type of reservoir computer, of which the recurrent weights of the hidden state (commonly named "reservoir") are randomly assigned and possess low connectivity. Therefore, only the hidden-to-output weights are trained leading to a simple quadratic optimization problem, which does not require backpropagation (see Fig. 1a for a graphical representation). The

reservoir acts as a memory of the observed state history. ESNs have demonstrated accurate inference of chaotic dynamics, such as in [28–36,63].

An echo state network maps the state from time index $t_i$ to index $t_{i+1}$ as follows (with a slight abuse of notation, the discrete time is denoted $t_i$). The evolution equations of the reservoir state and output are governed, respectively, by [36,38]

$$r(t_{i+1}) = \tanh\left([\hat{y}_{in}(t_i); b_{in}]^T \mathbf{W}_{in} + r(t_i)^T \mathbf{W}\right),$$
(14)

$$y_p(t_{i+1}) = [r(t_{i+1}); 1]^T \mathbf{W}_{out};$$
(15)

where at any discrete time $t_i$ the input vector, $\mathbf{y}_{in}(t_i) \in \mathbb{R}^{N_y}$, is mapped into the reservoir state $r \in \mathbb{R}^{N_r}$, by the input matrix, $\mathbf{W}_{in} \in \mathbb{R}^{(N_y+1)\times N_r}$, where $N_r \gg N_y$. The updated reservoir state $r(t_{i+1})$ is calculated at each time iteration as a function of the current input $\hat{y}_{in}(t_i)$ and its previous value $r(t_i)$ via Eq. (14) and then is involved in the calculation of the predicted output, $\mathbf{y}_p(t_{i+1}) \in \mathbb{R}^{N_y}$ via Eq. (15). Here, ($\hat{}$) indicates normalization by the maximum-minus-minimum range of $\mathbf{y}_{in}$ in training set component-wise, ($^T$) indicates matrix transposition, (;) indicates array concatenation, $\mathbf{W} \in \mathbb{R}^{N_r \times N_r}$ is the state matrix, $b_{in}$ is the input bias and $\mathbf{W}_{out} \in \mathbb{R}^{(N_r+1)\times N_y}$ is the output matrix. In our applications, the dimension of the input and output vectors is equal to the dimension of the physical system of Eq. (1), i.e. $N_y \equiv D$.

The matrices $\mathbf{W}_{in}$ and $\mathbf{W}$ are (pseudo)randomly generated and fixed, whilst the weights of the output matrix, $\mathbf{W}_{out}$, are the only trainable elements of the network. The input matrix, $\mathbf{W}_{in}$, has only one element different from zero per row, which is sampled from a uniform distribution in $[-\sigma_{in}, \sigma_{in}]$, where $\sigma_{in}$ is the input scaling. The state matrix, $\mathbf{W}$, is an Erdös–Renyi matrix with average connectivity $d$, in which each neuron (each row of $\mathbf{W}$) has on average only $d$ connections (i.e. nonzero elements), which are obtained by sampling from a uniform distribution in $[-1, 1]$. The echo state property enforces the independence of the reservoir state on the initial conditions, which is satisfied by rescaling $\mathbf{W}$ by a multiplication factor, such that the absolute value of the largest eigenvalue [38], i.e. the spectral radius, is smaller than unity. Following [29,36,63,64], we add a bias in the input and output layers to break the inherent symmetry of the basic ESN architecture. Specifically, the input bias, $b_{in}$ is a hyperparameter, selected in order to have the same order of
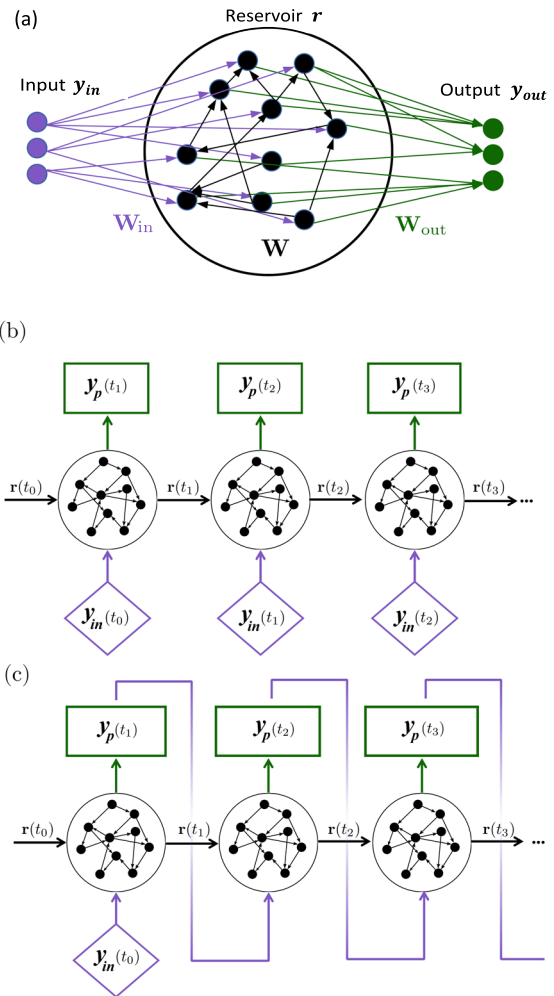


**Fig. 1** **a** Schematic representation of the echo state network. **b** Open-loop and **c** closed-loop configurations

magnitude as the normalized inputs, $\hat{y}_{in}$. Differently, the output bias is determined by training the weights of the output matrix, $\mathbf{W}_{out}$.

In Fig. 1b and c, we present the two types of configurations with which the ESN can run, i.e. in open loop or closed loop, respectively. Running in open-loop is necessary for the training stage, as the input data is fed at each step, allowing for the calculation of the reservoir time series $r(t_i)$, $t_i \in [0, T_{train}]$, which need to be stored. There is an initial transient time window, the "washout interval", where the output $\mathbf{y}_p(t_i)$ is not computed. This allows for the reservoir state to satisfy the echo state property, i.e. making it independent of the arbitrarily chosen initial condition, $r(t_0) = 0$, while also synchronizing it with respect to the current state of the system.

The training of the output matrix, $\mathbf{W}_{\text{out}}$, is performed after the washout interval and involves the minimization of the mean square error between the outputs and the data over the training set

$$\text{MSE} = \frac{1}{N_{\text{tr}} N_y} \sum_{i=0}^{N_{\text{tr}}} ||\mathbf{y}_{\text{p}}(t_i) - \mathbf{y}_{\text{in}}(t_i)||^2, \qquad (16)$$

where $||\cdot||$ is the $L_2$ norm, $N_{\text{tr}}+1$ is the total number of data in the training set, and $\mathbf{y}_{\text{in}}$ the input data on which the ESN is trained. Training the ESN is performed by solving with respect to $\mathbf{W}_{\text{out}}$ via ridge regression of

$$(\mathbf{R}\mathbf{R}^T + \beta\mathbb{I})\mathbf{W}_{\text{out}} = \mathbf{R}\mathbf{Y}_{\text{d}}^T, \qquad (17)$$

where $\mathbf{R} \in \mathbb{R}^{(N_r+1)\times N_{\text{tr}}}$ and $\mathbf{Y}_{\text{d}} \in \mathbb{R}^{N_y \times N_{\text{tr}}}$ are the horizontal concatenation of the reservoir states with bias, $[\boldsymbol{r}(t_i); 1]$, $t_i \in [0, T_{\text{train}}]$ and of the output data, respectively; $\mathbb{I}$ is the identity matrix and $\beta$ is the Tikhonov regularization parameter [65].

On the other hand, in the closed-loop configuration (Fig. 1c) the output $\mathbf{y}_{\text{p}}$ at time step $t_i$ is used as an input at time step $t_{i+1}$, in a recurrent manner, allowing for the autonomous temporal evolution of the network. The closed-loop configuration is used for validation (i.e. hyperparameter tuning, see Sect. 2.3) and testing, but not for training. For our purposes, we independently train $N_{\text{ESN}} \in [5, 10]$ networks, of which we take the ensemble average to increase the statistical accuracy of the prediction and evaluate its uncertainty. We start with $N_{\text{ESN}} = 10$ trained networks, but during post-processing we may discard any network that shows spurious temporal evolution. The $N_{\text{ESN}}$ networks are statistically independent thanks to: (1) initializing the random matrices $\mathbf{W}_{\text{in}}$ and $\mathbf{W}$ with different seeds, and (2) training each network with chaotic time series staring from different initial points on the attractor.

### 2.2.1 Jacobian of the ESN

In this subsection, we mathematically derive the Jacobian of the echo state network. Equations (14)–(15) are a discrete map [2,32],

$$\boldsymbol{r}(t_{i+1}) = f(\mathbf{y}(t_i), \boldsymbol{r}(t_i))$$
$$= \tanh\left([\hat{\mathbf{y}}(t_i); b_{\text{in}}]^T \mathbf{W}_{\text{in}} + \boldsymbol{r}(t_i)^T \mathbf{W}\right),$$
$$\mathbf{y}(t_{i+1}) = [\boldsymbol{r}(t_{i+1}); 1]^T \mathbf{W}_{\text{out}},$$

and the continuous-time formulae derived for the Lyapunov exponents and CLVs in Sect. 2.1 can be adapted for a discrete-time system. The Jacobian of the ESN reservoir is the total derivative of the hidden state dynamics at a single timestep [29]

$$\mathbf{J}(\boldsymbol{r}(t_{i+1})) = \frac{d\boldsymbol{r}(t_{i+1})}{d\boldsymbol{r}(t_i)} = \frac{df(\mathbf{y}(t_i), \boldsymbol{r}(t_i))}{d\boldsymbol{r}(t_i)}$$
$$= \frac{\partial f(\mathbf{y}(t_i), \boldsymbol{r}(t_i))}{\partial \mathbf{y}(t_i)} \frac{\partial \mathbf{y}(t_i)}{\partial \boldsymbol{r}(t_i)} + \frac{\partial f(\mathbf{y}(t_i), \boldsymbol{r}(t_i))}{\partial \boldsymbol{r}(t_i)}$$
$$= (1 - \tanh^2[\cdot])\mathbf{W}_{\text{in}}^T \mathbf{W}_{\text{out}}^T + (1 - \tanh^2[\cdot])\mathbf{W}^T$$
$$= (1 - \boldsymbol{r}(t_{i+1})^2)\left(\mathbf{W}_{\text{in}}^T \mathbf{W}_{\text{out}}^T + \mathbf{W}^T\right), \qquad (18)$$

where from Eq. (14) $\boldsymbol{r}(t_{i+1})^2 = \tanh^2([\hat{\mathbf{y}}_{\text{in}}(t_i); b_{\text{in}}]^T \mathbf{W}_{\text{in}} + \boldsymbol{r}(t_i)^T \mathbf{W})$ is the updated squared hidden state at timestep $t_{i+1}$. The Jacobian of the ESN is cheap to calculate as the expression $\left(\mathbf{W}_{\text{in}}^T \mathbf{W}_{\text{out}}^T + \mathbf{W}^T\right)$ is a constant matrix, which is fixed after the training of $\mathbf{W}_{\text{out}}$. The only time-varying component is the hidden state. The Jacobian $\mathbf{J} \in \mathbb{R}^{N_r \times N_r}$ is used for the extraction of the Lyapunov spectrum and the CLVs of a trained ESN. We time-march $D$ Lyapunov vectors $\boldsymbol{u}_i \in \mathbb{R}^{N_r}$ and periodically perform QR decompositions, where $\boldsymbol{Q} \in \mathbb{R}^{N_r \times D}$, and $\boldsymbol{R} \in \mathbb{R}^{D \times D}$. The same CLV algorithm described in Sect. 2.1 is employed to extract $D$ covariant Lyapunov vectors $\mathbf{v}_i \in \mathbb{R}^{N_r}$ from a trained ESN. The pseudocode is given in algorithm 2.

### 2.3 Validation

The dataset is split into three subsets, which are the training, validation, and testing subsets in a time-ordered fashion. During training, the ESN runs in open-loop, while during validation and testing, the ESN runs in closed-loop and the prediction at each step becomes the input for the next step. After training the ESN, its validation is necessary for the determination of the hyperparameters. The objective is to compute the hyperparameters that minimize the logarithm of the MSE (16). The logarithm of the MSE is preferred because the error varies by orders of magnitude for different hyperparameters, as explained in [36]. In general, instead of Eq. (16), other types of error functions can be used for the hyperparameter tuning, such as the maximization of the prediction horizon [29,34,36] or the minimization of the kinetic energy differences [64]. Here the input scaling, $\sigma_{\text{in}}$, the spectral radius, $\rho$, and the Tikhonov parameter, $\beta$, are the ESN hyperparameters that are being tuned [38,64]. In order to select

**Table 1** Echo state networks' hyperparameters

| Parameter | Name | Value |
|---|---|---|
| $\rho$ | Spectral radius | [0.1, 1] |
| $\sigma_{\text{in}}$ | Input scaling | [0.1, 5] |
| $\beta$ | Tikhonov parameter | $\{10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}\}$ |
| $d$ | Connectivity | 3 |
| $b_{\text{in}}$ | Input bias | 1 |
| $\sigma_n$ | Noise (training) | $0.0005\sigma_u$ |

Multiple values indicate that the parameter is optimized within the range

the optimal hyperparameters, $\sigma_{\text{in}}$ and $\rho$, we employ a Bayesian optimization, which is a strategy for finding the extrema of objective functions that are expensive to evaluate [64,66]. Within the optimal $[\sigma_{\text{in}}, \rho]$, we perform a grid search to select $\beta$ [64]. In particular, $[\sigma_{\text{in}}, \rho]$ are searched in the hyperparameter space $[0.1, 5] \times [0.1, 1]$ in logarithmic scale, while for $\beta$ we test $\{10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}\}$. The Bayesian optimization starts from a grid of $6 \times 6$ points in the $[\sigma_{\text{in}}, \rho]$ domain, and then, it selects five additional points through the `gp-hedge` algorithm [66]. We set $b_{\text{in}} = 1$, $d = 3$ and add Gaussian noise with zero mean and standard deviation, $\sigma_n = 0.0005\sigma_y$, where $\sigma_y$ is the standard deviation of the data component-wise, to the training and validation data. Adding noise to the data improves the performance of ESNs in chaotic dynamics by alleviating overfitting [32]. A summary of the hyperparameters is shown in Table 1.

One of the most commonly used validation strategy for RNNs is the single shot validation (SSV) [67], in which the data are split into a training set, followed by a single small validation set; see Fig. 2a. As the ESN now runs in closed loop, the size of the validation set is limited by the chaotic nature of the signal. In particular, at the beginning of the validation set, the input $\mathbf{y}(t_0)$ of the ESN is initialized to the target value. However, chaos causes the predicted trajectory to quickly diverge from the target trajectory in a few Lyapunov times $\tau_\lambda$. The validation interval is therefore small and not representative of the full training set, which causes poor performance in the test set [64]. An improvement to the performance with cheap computations is achieved by the the recycle validation (RV), which was recently proposed by [64]. In the RV, the network is trained only once on the entire training dataset (in open loop), and validation is performed on multiple intervals already used for training (but now in closed loop); see Fig. 2b. In
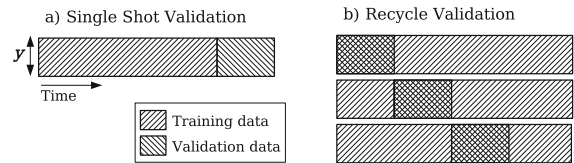


**Fig. 2** Schematic representation of the **a** single shot, and **b** recycling validation strategies. Here, **y** represents the degrees of freedom of the data. Three sequential validation intervals are shown for the Recycle Validation [64]

this work, we use the chaotic recycle validation (RVC), where the validation interval simply shifts as a small multiple of the first Lyapunov exponent, $N_{\text{val}} = 3\lambda_1$.

## 3 Results

In this section, we present the numerical results, which include a thorough comparison between the statistics produced by the autonomous temporal evolution of the ESN and the target dynamical system. The selected observables are the statistics of the degrees of freedom, the Lyapunov exponents, the angles between the CLVs or subspaces composed of CLVs, and the finite-time covariant Lyapunov exponents. We separate our analysis into two subsections, which contain two low-dimensional systems and then two higher-dimensional systems.

### 3.1 Low-dimensional chaotic systems

As a first case, we consider two low-dimensional dynamical systems that exhibit chaotic behaviour: Lorenz 63 (L63) [1] and Rössler [39] attractors. The Lorenz 63 system is a reduced-order model of atmospheric convection for a single thin layer of fluid that is

heated uniformly from below and cooled from above, which is defined by:

$$\frac{dx_1}{dt} = \sigma(x_2 - x_1)$$

$$\frac{dx_2}{dt} = x_1(\rho - x_3) - x_2$$

$$\frac{dx_3}{dt} = x_1 x_2 - \beta x_3. \tag{19}$$

We chose the parameters $[\sigma, \beta, \rho] = [10, 8/3, 28]$ to ensure a chaotic behaviour. The Rössler attractor, which models equilibrium in chemical reactions, is governed by:

$$\frac{dx_1}{dt} = -(x_2 + x_3)$$

$$\frac{dx_2}{dt} = x_1 + ax_2$$

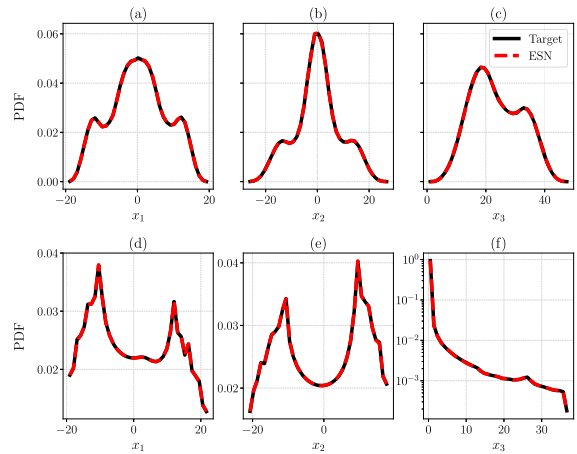$$\frac{dx_3}{dt} = b + x_3(x_1 - c), \tag{20}$$



**Fig. 3** Comparison of the target (straight black line) and ESN (red dashed line) probability density functions (PDF) of the three degrees of freedom, $x_1$, $x_2$, and $x_3$ of the Lorenz 63 system (19) **a**–**c** and the Rössler system (20) **d**–**f**. (Color figure online)

We choose the parameters $[a, b, c] = [0.1, 0.1, 14]$ to ensure a chaotic behaviour.

To generate the target set, we evolve the dynamical systems forward in time with a fourth-order Runge–Kutta (RK4) integrator and a timestep $dt = 0.005$ for both L63 and Rössler, which is sufficiently small for a good temporal resolution. (We tested slightly larger/smaller timesteps with no significant differences. Results not shown.) We perform a QR decomposition every $m = 1$ timesteps for L63 and every $m = 5$ timesteps for Rössler. For all systems, we generate a training set of size $1000\tau_\lambda$ and a test set of size $4000\tau_\lambda$, for the CLV statistics to converge, where $\tau_\lambda = 1/\lambda_1$ is the Lyapunov time, which is the inverse of the maximal Lyapunov exponent $\lambda_1$.

First, we test whether the ESN correctly learns the chaotic attractor from a statistical point of view, i.e. whether the ESN correctly learns the long-term statistics of the degrees of freedom when it evolves in the closed-loop (autonomous) mode. By estimating the probability density function (PDF) of the degrees of freedom of the ESNs, as a normalized histogram, and comparing it with the corresponding PDF of the target set, we extract information on the invariant measure of the considered chaotic system. This is shown in Fig. 3 for L63 and Rössler attractors, in which the black lines show the target statistics and the red dashed lines show the ESN statistics. In Figs. 3, 5 and 7, and Table 2, we have used $N_{ESN}$ ESNs trained on $N_{ESN}$ independent target systems, starting from different initial conditions,

and averaged among the estimated observables, where $N_{ESN} = 6$, and $N_{ESN} = 8$ for Rössler and L63, respectively. We perform the ensemble calculation to quantify the uncertainty of the predictions and the robustness of the ESN for different initializations.

Second, we test whether the ESN correctly learns the Lyapunov spectrum. Table 2 shows the ESN predictions on the Lyapunov exponents for the L63 and Rössler attractors, which are compared with the target exponents. The leading exponent is accurately predicted with a 0.2% error in the L63 and 1.5% error in the Rössler system. In chaotic systems, there exists a neutral Lyapunov exponent, which is associated with the direction of $\frac{dx}{dt}$. In these cases, the neutral Lyapunov exponents are $\lambda_2 = 0$ for both systems, which are correctly inferred by the ESN within a $\mathcal{O}(10^{-5})$ error, or less. For the smallest, and negative exponent, which is generally harder to extract because it is highly damped, the relative error is about 0.6% for L63 and 2.1% for Rössler. Therefore, the ESNs can accurately capture the tangent dynamics of a low-dimensional chaotic attractor.

Third, we investigate the angles between the CLVs. We assess whether the ESNs learn the long-term statistics of these quantities, but also whether, they correctly infer the distribution and fluctuations of those observables in the phase space. In other words, whether the ESNs learn the geometrical structure of the attractor and its tangent space.
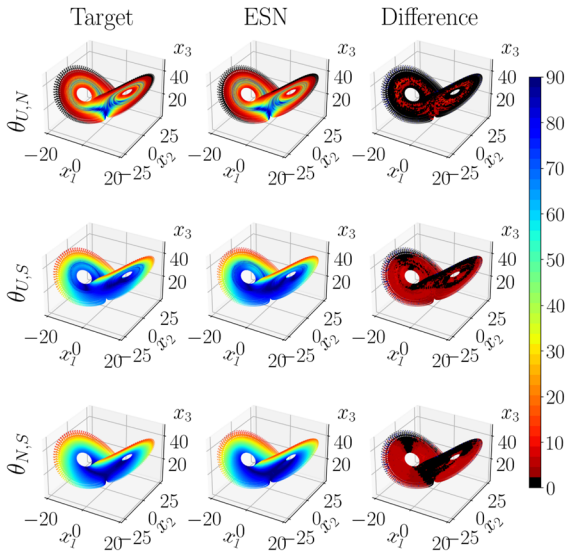
**Fig. 4** Comparison of Target (left column), ESN (middle column), and their statistical mean absolute difference (right column), for a $300\tau_\lambda$ trajectory of the Lorenz 63 system (19) in the test set, coloured by the CLV principal angles (in deg). First row: $\theta_{U,N}$, second row: $\theta_{U,S}$, and third row: $\theta_{N,S}$
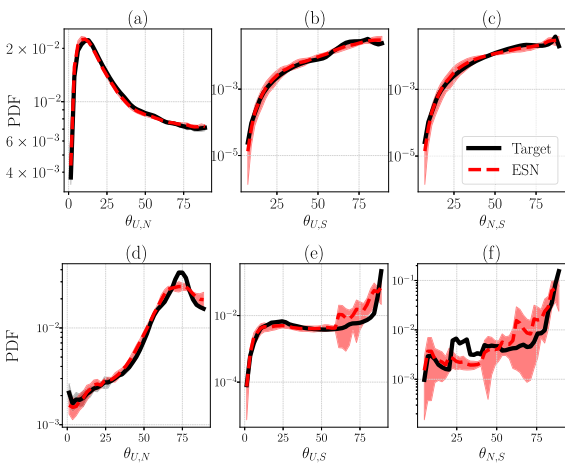


**Fig. 5** Comparison of the target (straight black line) and ESN (red dashed line) probability density functions (PDF) of the three principal angles between the covariant Lyapunov vectors, where $U$ refers to unstable, $N$ to neutral and $S$ to stable CLVs. The top row (**a**–**c**) is for Lorenz 63 (19) and the bottom row (**d**–**f**) for Rössler (20). All $y$-axes are in logarithmic scale and the $x$-axis is in degrees. The shaded region indicates the error bars derived by the standard deviation. (Color figure online)

In Fig. 4, we present an analysis of the distribution of principal angles between the CLVs,

$$\theta_{a,b} = \frac{180°}{\pi} \cos^{-1}(|\mathbf{v}_a \cdot \mathbf{v}_b|), \qquad (21)$$

$\theta_{a,b} \in [0°, 90°]$, on the topology of the L63 attractor. The attractor is well reproduced by a selected ESN (middle column), compared to the target (left column). The size of both trajectories is $300\tau_\lambda$. In this case, there are three principal angles between the CLVs; $\theta_{U,N}$ is the angle between the unstable and neutral CLV; $\theta_{U,S}$ is the angle between the unstable and stable CLV; $\theta_{N,S}$ is the angle between the neutral and stable CLV. The colouring of the attractor is associated with the measured $\theta_{a,b}$. The black and dark red colours identify small angles, i.e. regions of the attractor where near-tangencies between the CLVs occur. Possible tangencies between CLVs or invariant manifolds composed of CLVs (as will be discussed later for higher dimensional chaotic systems) are of significant importance, as they signify that the attractor is non-hyperbolic [10] (see Sect. 2.1). The right column is the mean absolute difference between the target and the ESN. The $x$, $y$, $z$ domain is discretized with 50 bins in each direction; then, the mean $\theta_{a,b}$ is calculated from each of the three-dimensional bins for the $300\tau_\lambda$ long trajectory. Finally, the absolute difference between ESN and target is calculated for each bin. The plots follow the same colour scheme as the colourbar, with black and dark red colours indicating $< 2°$ differences with a maximum of $\sim 10°$. Figure 4 shows that the ESN is able to accurately learn the dynamics of the tangent linear space of the attractor.

In Fig. 5, we show the PDF of the principal angles between the three CLVs, for which there is agreement between target and ESN results in all cases for both L63 and Rössler, even for smaller angles. The nonzero count of events close to $\theta \to 0$ indicates that the two considered systems are non-hyperbolic, which is consistent with the literature [58].

Fourth, we analyse the distribution on the attractor, as well as the statistics, of the Finite-time Covariant Lyapunov Exponents, for a time-lapse of $\Delta t = m \, dt$ timestep, and assess the accuracy of the trained ESNs. For the considered low-dimensional systems there are three FTCLEs with each showing the finite-time growth rate of the corresponding Covariant Lyapunov Vectors.

In Fig. 6, we visualize the distribution of the single timestep FTCLEs, in the case of the Rössler attractor, which is well reproduced by a selected ESN (middle column), compared to the target (left column). The size of both trajectories is $300\tau_\lambda$. FTCLE 1 is the finite-time exponent for the unstable CLV, FTCLE 2 is for the neutral CLV, and FTCLE 3 is for the stable CLV. The
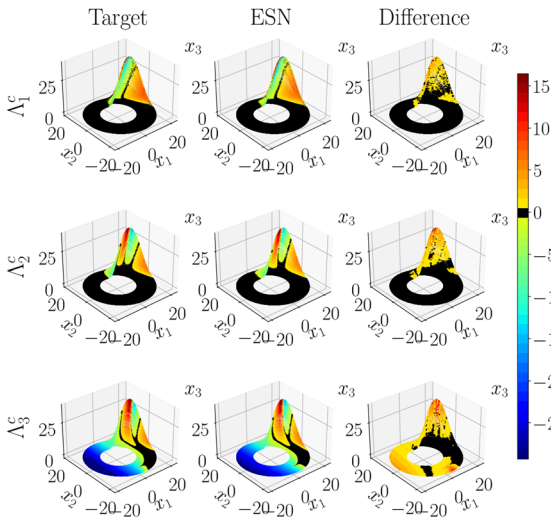
**Fig. 6** Comparison of target (left column), ESN (middle column), and their statistical mean absolute difference (right column) for a $300\tau_\lambda$ trajectory of the Rössler system (20) in the test set, coloured by the three FTCLEs. First row: FTCLE 1, second row: FTCLE 2, and third row: FTCLE 3
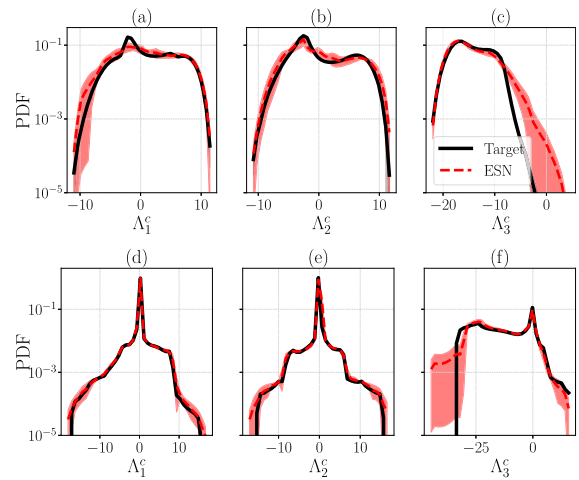


**Fig. 7** Comparison of the target (straight black line) and ESN (red dashed line) probability density functions (PDF) of the three finite-time covariant Lyapunov Exponents. The top row **a–c** is for Lorenz 63 (19) and the bottom row **d–f** for Rössler (20). All $y$-axes are in logarithmic scale. (Color figure online)

colouring is associated with the values of the FTCLEs. Large positive FTCLEs correspond to high finite-time growth rates and, thus, reduced predictability. The distribution of the leading FTCLE on the attractor is similar between the target and ESN. The second FTCLE and third FTCLE, which correspond to the neutral CLV and stable CLVs, accordingly, also show good agreement between the two. The mean difference between the target and the ESN on the attractor is plotted in the right column, in which black identifies $\Lambda_i^c \approx 0$. The right column shows that most of the small differences between the ESN and the target are located in the region of large variation of $z$.

Finally, Fig. 7 shows the PDF of the three FTCLEs. There is agreement between the ESN-inferred quantities and the target in all cases, in particular in the Rössler attractor for the most-probable statistics. The small deviation in Fig. 7a for L63 corresponds to the statistics around the peak of the first FTCLE, $\Lambda_1^c$, but the tails of the distributions are well reproduced. The mean of the $\Lambda_i^c$ distributions coincides with the LEs $\lambda_i$, which holds true for all our results. A behaviour as in Fig. 7a implies that in this case the finite-time values $\Lambda_1^c$ are less peaked around the mean value, even though their long-time average coincides with the Lyapunov exponent $\lambda_1$. Nevertheless, in Figs. 7d–f for Rössler the statistics around the peak (and beyond) are well captured.

**Table 2** Estimates of Lyapunov exponents $\lambda_i$ for the two low-dimensional systems, the Lorenz 63 and Rössler attractors

| | Lorenz 63 | | Rössler | |
|---|---|---|---|---|
| | Target | ESN | Target | ESN |
| 1 | 0.9050 | 0.9067 | 0.071 | 0.070 |
| 2 | $9\times10^{-5}$ | $-8.\times10^{-5}$ | $2\times10^{-6}$ | $1\times10^{-6}$ |
| 3 | $-14.572$ | $-14.664$ | $-13.88$ | $-14.17$ |

Comparison between the target and echo state network

We refer the interested reader to our supplementary material where the corresponding results of Figs. 4 and 6 for both attractors are shown. Also, the statistics of FTLEs, as well as their distribution on the chaotic attractors, are presented in the supplementary material.

### 3.2 Higher-dimensional chaotic systems

We follow the same analysis and approach as in Sect. 3.1 for two higher-dimensional chaotic systems, both of which are related to atmospheric physics and meteorology. The first is a reduced-order model of atmospheric blocking events by Charney and DeVore [40] (CdV), which is a six-dimensional truncation of the equations for barotropic flow with orography. We employ the formulation of [68,69], which is forced by a zonal flow profile that can be barotropically unstable.

The governing equations are:

$$\frac{dx_1}{dt} = \gamma_1^* x_3 - C(x_1 - x_1^*),$$

$$\frac{dx_2}{dt} = -(\alpha_1 x_1 - \beta_1)x_3 - Cx_2 - \delta_1 x_4 x_6,$$

$$\frac{dx_3}{dt} = (\alpha_1 x_1 - \beta_1)x_2 - \gamma_1 x_1 - Cx_3 + \delta_1 x_4 x_5,$$

$$\frac{dx_4}{dt} = \gamma_2^* x_6 - C(c_4 - x_4^*) + \epsilon(x_2 x_6 - x_3 x_5),$$

$$\frac{dx_5}{dt} = -(\alpha_2 x_1 - \beta_2)x_6 - Cx_5 - \delta_2 x_4 x_3,$$

$$\frac{dx_6}{dt} = (\alpha_2 x_1 - \beta_2)x_2 - \gamma_2 x_4 - Cx_6 + \delta_2 x_4 x_2, \quad (22)$$

where the model coefficients are

$$\alpha_m = \frac{8\sqrt{2}}{\pi}\frac{m^2}{4m^2-1}\frac{b^2+m^2-1}{b^2+m^2}, \quad \beta_m = \frac{\beta b^2}{b^2+m^2},$$

$$\delta_m = \frac{64\sqrt{2}}{15\pi}\frac{b^2-m^2+1}{b^2+m^2}, \quad \gamma_m^* = \gamma\frac{4m}{4m^2-1}\frac{\sqrt{2}b}{\pi},$$

$$\epsilon = \frac{16\sqrt{2}}{5\pi}, \quad \gamma_m = \gamma\frac{4m^3}{4m^2-1}\frac{\sqrt{2}b}{\pi(b^2+m^2)}. \quad (23)$$

Equation (22) is integrated with RK4 and $dt = 0.1$. The constants are set to $(x_1^*, x_4^*, C, \beta, \gamma, b) = (0.95, -0.76095, 0.1, 1.25, 0.2, 0.5)$, for which the CdV model generates regime transitions [68,69]. In particular, the CdV model allows for two metastable states, the so-called "zonal" state, which represents the approximately zonally symmetric jet stream in the mid-latitude atmosphere, and the "blocked" state, which refers to a diverse class of weather patterns that are a persistent deviation from the zonal state. Blocking events are known to be associated with regional extreme weather, from heatwaves in summer to cold spells in winter [70]. The dynamical properties of CLVs in connection to blocking events were recently investigated for a series of more complex atmospheric models than CdV [11,12,71], which demonstrated that CLVs are good candidates for blockings precursors, as well as a good basis for model reduction. In the previous work [34], the CdV system was used as a training model for the ESN, with the purpose of studying short-term accurate prediction of chaos, and quantifying the benefit of Physics informed echo state networks [34].

The second higher-dimensional system that we consider is the Lorenz 96 (L96) model [41], which is a system of coupled ordinary differential equations that describes the large-scale behaviour of the mid-latitude atmosphere, and the transfer of a scalar atmospheric quantity. Three characteristic processes of atmospheric systems (advection, dissipation, and external forcing) are included in the model, whose equations are

$$\frac{d\boldsymbol{x}_i}{dt} = (\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i-2})\boldsymbol{x}_{i-1} - \boldsymbol{x}_i + F \quad (24)$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_D] \in \mathbb{R}^D$. We set periodic boundary conditions, i.e. $x_1 = x_{D+1}$. In our analysis, we chose $D = 20$ degrees of freedom. The external forcing is set to $F = 8$, which ensures a chaotic evolution [32]. We integrate the system with RK4 and $dt = 0.01$. We perform a QR decomposition every $m = 5$ timesteps for CdV and every $m = 10$ timesteps for L96. Similar to the previous section, we generate a training set of size $1000\tau_\lambda$ and a test set of size $4000\tau_\lambda$.

First, Fig. 8 shows the PDF of the six degrees of freedom of CdV, and the first six from L96 (the PDFs of the rest 13 dofs have similar shape and agreement between ESN and target). We use a semilogarithmic scale to emphasize that the agreement between target (black line) and ESN (red dashed line) is accurate for the tails of the distributions, which effectively correspond to the edges of each attractor. As in Sect. 3.1 in order to evaluate uncertainty and robustness, we start with $N_{ESN} = 10$ trained networks, but during post-processing we discard any network that shows spurious temporal evolution, and perform a further averaging of the PDFs of each network's observable. Therefore, the PDFs of Fig. 8 are the outcome of averaging $N_{ESN} = 5$ and $N_{ESN} = 9$ PDFs with the same binning, for CdV and L96, respectively.

Second, Fig. 9 shows the Lyapunov exponents spectrum of (a) CdV and (b) L96 for $D = 20$ and compares the target (black squares) with the ESN prediction (red circles). The CdV model has a single positive Lyapunov exponent, with the average value of 5 ESNs resulting in $\lambda_1 = 0.0214$, and for the 5 independent target sets, $\lambda_1^{targ} = 0.0232$ with an 8% absolute error. The second Lyapunov exponent is zero (to numerical error) and corresponds to the neutral direction, with $\lambda_2 = -3 \times 10^{-5}$ for ESN, and $\lambda_2^{targ} = -7 \times 10^{-6}$ for the target. The low order of magnitude achieved by the ESN assures its ability to capture the neutral exponent. Finally, the four remaining negative exponents are well learned by the ESN, i.e. $\lambda_{3-6} = [-0.077, -0.103, -0.224, -0.234]$ and $\lambda_{3-6}^{targ} = [-0.079, -0.101, -0.218, -0.226]$ for the
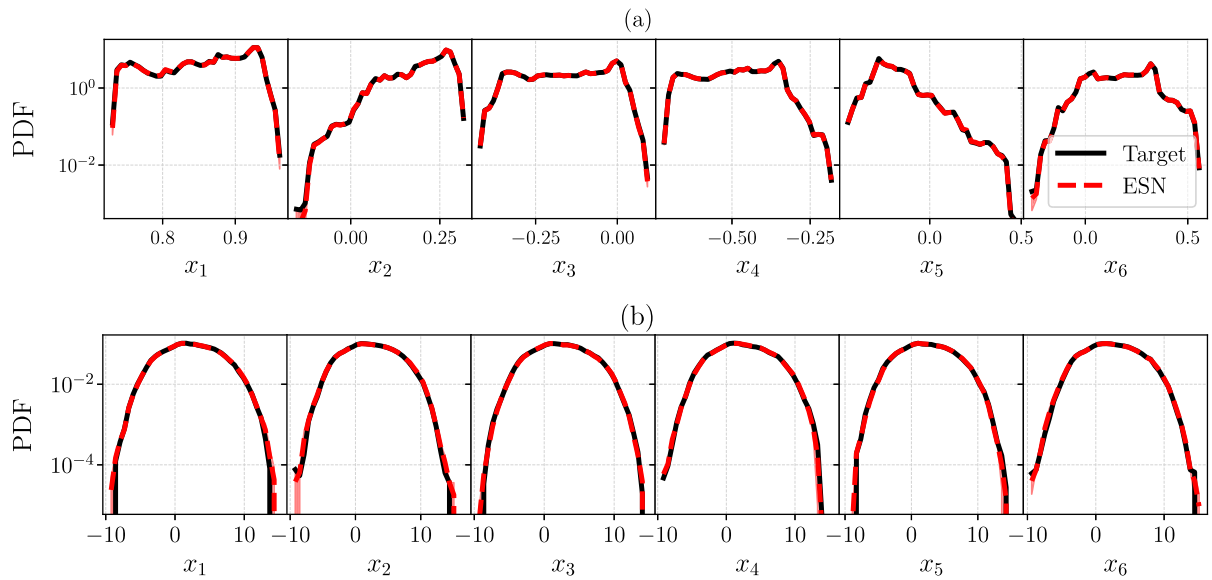
**Fig. 8** Comparison of the target (straight black line) and ESN (red dashed line) PDF of the first six degrees of freedom, $x_1, \ldots, x_6$ of the **a** Charney–DeVore (22) and **b** for Lorenz 96 (24) for $D = 20$. (Color figure online)
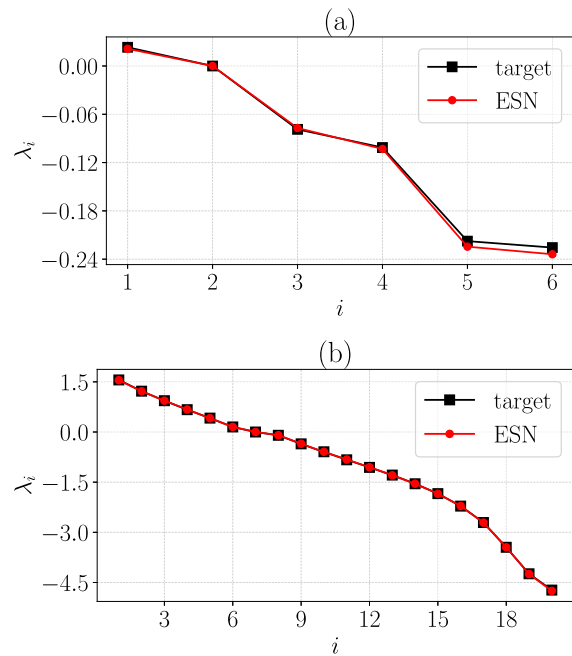


**Fig. 9** Comparison of the target (black squares) and ESN (red circles) Lyapunov spectrum for **a** Charney–DeVore (22) and **b** Lorenz 96 (24) at $D = 20$. (Color figure online)

target. Overall, excluding $\lambda_2$, the mean absolute error of the CdV Lyapunov spectrum here is 3.7%, which is negligibly small.

With respect to the L96 Lyapunov spectra in Fig. 9b, the agreement between target and ESN across all 20 exponents is good. In particular, there are 6 positive, 1 zero and 13 negative exponents. The maximal exponent predicted from the ensemble of $N_{ESN} = 9$ ESNs is equal to $\lambda_1 = 1.551$, and for the 9 independent target sets, $\lambda_1^{targ} = 1.557$, meaning a 0.4% absolute error. The rest of the positive exponents are well captured by the ESN, with $\lambda_{2-6} = [1.221, 0.936, 0.668, 0.416, 0.151]$ and $\lambda_{2-6}^{targ} = [1.217, 0.937, 0.673, 0.413, 0.152]$ for the target. The zero exponent is sufficiently small with $\lambda_7 = -10^{-4}$ for ESN, and $\lambda_7^{targ} = 4 \times 10^{-4}$ for target. Albeit more difficult to predict because of large numerical dissipation, the negative Lyapunov exponents are accurately learned by the ESN, with the smallest ones reading $\lambda_{15-20} = [-1.84, -2.22, -2.71, -3.45, -4.24, -4.73]$ and accordingly $\lambda_{15-20}^{targ} = [-1.85, -2.21, -2.71, -3.45, -4.25, -4.75]$ for the target. Those directions in tangent space decay exponentially fast and the accuracy that the ESN achieves is consistent. For L96 the mean absolute error of the Lyapunov spectrum is approximately 0.5%.

To further elaborate, the L96 is known to be an extensive system [72,73], which means that quantities such as the surface width, the entropy and the attractor dimension scale linearly with its dimensionality $D$. For the Lyapunov spectrum, this means that the pro-
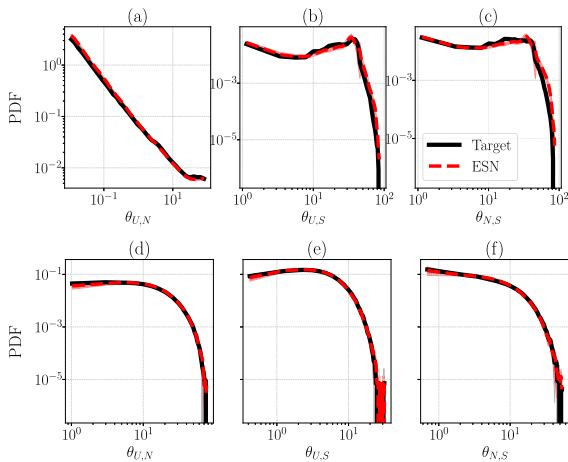
**Fig. 10** Comparison of the target (straight black line) and ESN (red dashed line) PDF of the three minimum principal angles between the three subspaces composed by the CLVs, where $U$ refers to unstable, $N$ to neutral and $S$ to stable CLVs. The top row **a**–**c** is for Charney–DeVore (22) and the bottom row **d**–**f** for Lorenz 96 (24) at $D = 20$. Both $x$ and $y$ axes are in logarithmic scale and the $x$-axis is in degrees. Only in **a** a logarithmic binning was used being denser close to $\theta_{U,N} \to 0$, while PDFs in **b**–**f** are linearly binned in $x$-axis. (Color figure online)

portion of positive to negative exponents is roughly the same ($\approx 1/2$) as $D$ changes. For this reason, our chosen $D = 20$ is sufficient for our purposes.

Third, we investigate the statistics of the principal angles, $\theta \in [0°, 90°]$, between the three subspaces that partition the invariant manifolds, which are the unstable $E_x^U$, neutral $E_x^N$ and stable $E_x^S$, spanned by the corresponding CLVs. The extraction of the principal angles between two linear subspaces requires a singular value decomposition of their matrix product $\boldsymbol{\Gamma}^{a,b} = E_x^a E_x^b$ (assuming the CLVs are ordered as stacked columns, according to their Lyapunov exponent order), because all paired products between the CLVs spanning the subspaces do not provide all the angles [10,74]. The angles are given by

$$\theta_{a,b} = \frac{180°}{\pi} \cos^{-1}\left( \text{svd}[\boldsymbol{\Gamma}^{a,b}] \right), \tag{25}$$

and we analyse the smallest singular value. Here, we use the implemented routine `scipy.linalg.sub space_angles` of the scipy package [57] in python and analyse the minimum angle in order to track homoclinic tangencies between the subspaces. This implementation is based on the algorithm presented in [75], which has improved accuracy with respect to Eq. (25) in the estimation of small angles.

In Fig. 10, we study the PDFs of the three principal angles between the linear subspaces for CdV and
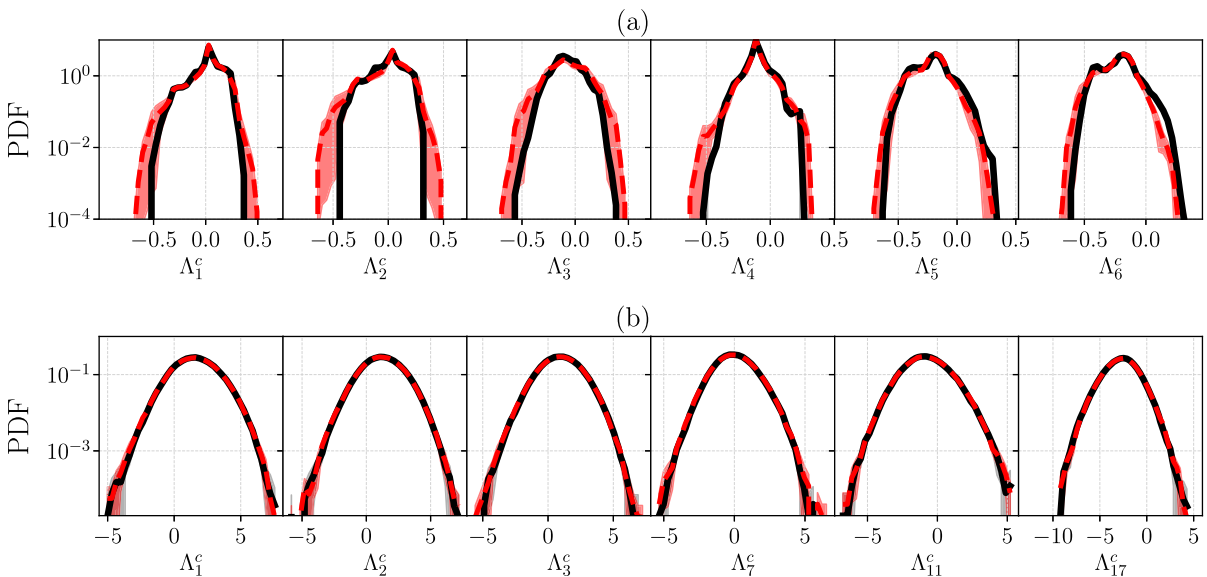


**Fig. 11** Comparison of the target (straight black line) and ESN (red dashed line) PDF of six finite-time covariant Lyapunov exponents ($\Lambda_i^c$) for **a** Charney–DeVore (22) for which $\lambda_1 > 0, \lambda_2 = 0$ and the rest are $\lambda_i < 0$, and **b** Lorenz 96 (24) at $D = 20$, where for $i = 1, 2, 3$ $\lambda_i > 0$, for $i = 7$ $\lambda_i = 0$, and for $i = 11, 17$ $\lambda_i < 0$. All $y$-axes are in logarithmic scale. (Color figure online)

L96. In CdV, the unstable and neutral subspaces are spanned only by the corresponding CLVs, while the stable subspace is spanned by the remaining four CLVs, of which $\lambda_i < 0$. In L96 with $D = 20$, the unstable subspace is spanned by the first six CLVs, the neutral subspace is spanned only by the 7th CLV, and the stable subspace is spanned by the remaining 13 CLVs. Focusing on Fig. 10a–c for CdV, we notice that this system is non-hyperbolic because the PDFs are populated close to $\theta \to 0$. Specifically for Fig. 10a, the binning is geometrically spaced and denser close to $\theta \to 0$. Interestingly, the PDF of $\theta_{U,N}$ of CdV for small angles follows a power-law $PDF(\theta) \sim \theta^{-\alpha}$ for $\theta \to 0$ and until $\approx 10°$, before it saturates. A different shape that is still highly non-hyperbolic is shown for the PDFs of $\theta_{U,S}$ and $\theta_{N,S}$ in Fig. 10b and c, in which the binning is linear and both axes in logarithmic scale. Figure 10d–f shows the same statistics in the case of L96, which is also non-hyperbolic, as there is strong frequency of tangencies, $\theta \to 0$. In all plots of Fig. 10, the agreement of the subspace angle statistics between target and ESN is good, which demonstrates that the ESN has achieved a robust and accurate learning of the ergodic properties from higher-dimensional data.

Fourth, the statistics of FTCLEs ($\Lambda_i^c$), for a time-lapse of $\Delta t = m \, dt$ timesteps, in the cases of CdV and L96 are shown in Fig. 11. All six $\Lambda_i^c$ are shown for CdV, while a representative set of six $\Lambda_i^c$ are shown for L96, such that $\lambda_i > 0$ for $k = 1, 2, 3$, $\lambda_i = 0$ for $k = 7$, and $\lambda_i < 0$ for $k = 11, 17$. For CdV, the most probable statistics are well captured by the ESN, which is in agreement with the target data. There are slight deviations at the tails of the distributions, which are still in agreement within error bars (shaded region). In the case of L96, the agreement is good for both the most probable statistics and the tails, for all FTCLEs (also those not shown). The first moment of the distributions, i.e. the mean of the FTCLEs time series, must be equal to the Lyapunov exponents, $\lambda_i = \frac{1}{T} \int_0^T \Lambda_i^c$, which indeed holds for all the cases considered here. The agreement between ESN and target sets in Fig. 11 shows that the ESN is able to accurately learn the finite-time variability of the CLV growth rates also for higher dimensional systems that are characterized by many Lyapunov exponents.

Finally, in Table 3 we show the estimated Kaplan–Yorke dimension [76] for all the considered systems and compare the outcomes of the ESN and target. This

**Table 3** Estimates of the Kaplan–Yorke dimension for all attractors, comparing between the target and echo state networks

|  | Target | ESN | % error |
| --- | --- | --- | --- |
| Lorenz 63 | 2.0621 | 2.0618 | 0.015 |
| Rossler | 2.0051 | 2.0049 | 0.01 |
| CdV | 2.294 | 2.277 | 0.74 |
| Lorenz 96 | 13.4697 | 13.4721 | 0.018 |

The error is the quantity $\frac{\text{target} - \text{ESN}}{\text{target}} \times 100\%$

dimension is an upper bound of the attractor's fractal dimension [2], which is defined as

$$D_{\text{KY}} = k + \frac{\sum_{i=1}^{k} \lambda_i}{|\lambda_{i+1}|}, \tag{26}$$

where $k$ is such that the sum of the first $k$ LEs is positive and the sum of the first $k+1$ LEs is negative. We observe a good agreement in all cases with $\leq 1\%$ error. This observation further confirms the ability of the ESN to accurately learn the properties of the chaotic attractor.

## 4 Conclusion

Stability analysis is a principled mathematical tool to quantitatively answer key questions on the behaviour of nonlinear systems: Will infinitesimal perturbations grow in time (i.e. is the system linearly unstable)? If so, what are the perturbations' growth rates (i.e. how linearly unstable is the system)? What are the directions of growth? To answer these questions, traditionally, we linearize the equations of the dynamical system around a reference point, and compute the properties of the tangent space, the dynamics of which is governed by the Jacobian. The overarching goal of this paper is to propose a method that infers the stability properties directly from data, which does not rely on the knowledge of the dynamical differential equations. We tackle chaotic systems, which have a linearized behaviour that is more general and intricate than periodic or quasi-periodic oscillations. First, we propose the echo state network with the recycle validation as a tool to accurately learn the chaotic dynamics from data. The data are provided by the integration of low- and higher- dimensional prototypical chaotic dynamical systems. These systems are qualitatively different from each other and are toy models that describe diverse physical settings, ranging from climatology and

meteorology to chemistry. Second, we mathematically derive the Jacobian of the echo state network (Eq. (18)). In contrast to other recurrent neural networks, such as long short-term memory networks or gated recurrent units, the Jacobian of the ESN is mathematically simple and computationally straightforward. Third, we analyse the stability properties inferred from the ESN and compare them with the target properties (ground truth) obtained by linearizing the equations. The ESN correctly infers quantities that characterize the chaotic dynamics and its tangent space (i) the long-term statistics of the solution, for which we compute the probability density function of each state variable; (ii) the covariant Lyapunov vectors, which are a physical basis for the tangent space that is covariant with the dynamics; (iii) the Lyapunov spectrum, which is the set of eigenvalues of the Oseledets matrix that are the perturbations' average exponential growths; (iv) the finite-time Lyapunov exponents, which are the finite-time growth along the covariant Lyapunov vectors; and (v) the angles between the stable, neutral, and unstable splittings of the tangent space, which informs about the degree of hyperbolicity of the attractor. We show that these quantities can be accurately learned from data by the ESN, with negligible numerical errors.

As mathematically and numerically shown in [44], the stability properties of fixed points (with eigenvalue analysis) and periodic solutions (with Floquet analysis) can be inferred from covariant Lyapunov analysis. Therefore, this work opens up new opportunities for the inference of stability properties from data in nonlinear systems, from simple fixed points, through periodic oscillations, to chaos.

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

## A Algorithms to compute LEs and CLVs

In this section, we present two algorithms for the computation of LEs and CLVs. Algorithm 1 is used to calculate the first $D$ LEs of an ESN, where $N_r$ is the dimensionality of the hidden state and $D$ is the dimensionality of the input state. This algorithm follows the methods described in [29,32]. Algorithm 2 computes the first $D$ CLVs for both the ESN and target chaotic systems, using the approach outlined in [42]. These algorithms are crucial for understanding the dynamics and predictability of the systems being studied.

## B Robustness

An important aspect of data-driven approaches is their ability to perform accurately under a variety of conditions. In this section, we evaluate the robustness of our approach by using smaller training sets (less data) subject to noise levels that are higher than those of Sect. 3. We also test the effect of using a loss function other than the mean square error (MSE), as defined in Eq. (16), on the accuracy of the learning. The ESN architecture follows [36], where it was trained with chaotic data from the Lorenz 63 and Lorenz 96 systems, and was robustly optimized to maximize the prediction horizon under different validation strategies.

B.1 Training with less data and higher noise intensity

It has been demonstrated that adding a small amount of Gaussian centered noise proportional to the standard deviation of the chaotic signal during training can

---

**Algorithm 1:** Algorithm to calculate the Lyapunov exponents of the echo state network

---

$U \leftarrow \texttt{random} \in \mathbb{R}^{N_r \times D}$ ;     /* Initialize $D$ GSVs */
$Q, R \leftarrow \texttt{QR}(U)$ ;     /* Othonormalize GSVs */
$U \leftarrow Q \in \mathbb{R}^{N_r \times D}$;
$N^{QR} \leftarrow (N^{test} - N^w)/m$ ;     /* Number of QR decompositions */
*Save the time series of $R$ and $Q$ for CLVs calculation*
Initialize $\tilde{R} \leftarrow \mathbf{0} \in \mathbb{R}^{D \times D \times N^{QR}}$;
Initialize $\tilde{Q} \leftarrow \mathbf{0} \in \mathbb{R}^{N_r \times D \times N^{QR}}$;
Initialize $\Lambda \leftarrow \mathbf{0} \in \mathbb{R}^{D \times N^{QR}}$ ;     /* Save the FTLEs */
$\mathbf{W}^c = \mathbf{W}_{\text{in}}^T \mathbf{W}_{\text{out}}^T + \mathbf{W}^T$ ; /* Constant matrices of Eq. (18) */
*Evolve the hidden state and GSVs simultaneously.*
*Skip a transient initial $N^w$ steps for warm-up.*
$n \leftarrow 0$ ; /* Increments the number of QR decompositions */
**for** $i = 0 : N^{test}$ **do**
   $r(t_{i+1}) \leftarrow \tanh\left([\hat{\mathbf{y}}(t_i); b_{\text{in}}]^T \mathbf{W}_{\text{in}} + r(t_i)^T \mathbf{W}\right);$
   $\mathbf{y}(t_{i+1}) \leftarrow [r(t_{i+1}); 1]^T \mathbf{W}_{\text{out}};$
   $J \leftarrow (1 - r(t_{i+1})^2)\mathbf{W}^c$ ;     /* The updated Jacobian */
   $U \leftarrow JU$ ;     /* The variational equation */
   **if** $\mod (i, m) = 0$ **then**
      $Q, R \leftarrow \texttt{QR}(U)$ ;     /* QR every $m$ steps */
      $U \leftarrow Q$;
      **if** $i > N^w$ **then**
         $\Lambda[:, n] \leftarrow \log(\text{diag}[R])/dt$ ;     /* Save the FTLEs */
         $\tilde{R}[:, :, n] \leftarrow R$ ;     /* Save $R$ */
         $\tilde{Q}[:, :, n] \leftarrow Q$ ;     /* Save $Q$ */
         $n \leftarrow n + 1$;
      **end**
   **end**
**end**
$\lambda_j \leftarrow \sum_{i=0}^{N^{QR}} \Lambda[j, i]/T^{test}$ ;     /* The $j$th Lyapunov exponent */

---

**Algorithm 2:** Algorithm to calculate the covariant Lyapunov vectors [10,42]

---

**Data**: Given $\tilde{R}$ and $\tilde{Q}$
*Set to $\mathbf{0}$ the matrices $C$, $\mathbf{D}$ and $V$*
$C \leftarrow \mathbf{0} \in \mathbb{R}^{D \times D \times N^{QR}}$ ;     /* Coordinates of CLVs in the GSV basis */
$\mathbf{D} \leftarrow \mathbf{0} \in \mathbb{R}^{D \times D \times N^{QR}}$ ;     /* Growth factors of CLVs */
$\Lambda^c \leftarrow \mathbf{0} \in \mathbb{R}^{D \times N^{QR}}$ ;     /* The $D$ FTCLEs */
$V \leftarrow \mathbf{0} \in \mathbb{R}^{N_r \times D \times N^{QR}}$ ;     /* Each column is a CLV */
*Set final time index to identity $\mathbb{I}$ for the matrices $C$ and $\mathbf{D}$*
$C[:, :, N^{QR}] \leftarrow \mathbb{I}$;
$\mathbf{D}[:, :, N^{QR}] \leftarrow \mathbb{I}$;
$V[:, :, N^{QR}] \leftarrow \tilde{Q}[:, :, N^{QR}]C[:, :, N^{QR}]$;
*Evolve backwards and solve Eq. (11)*
**for** $i = N^{QR} - 1 : 0$ **do**
   $G \leftarrow \texttt{solve\_triangular}(\tilde{R}[:, :, i]C[:, :, i + 1])$;
   *Normalize each column of $G$*
   **for** $j = 0 : G.shape[1]$ **do**
      $\mathbf{D}[:, :, i] \leftarrow \texttt{norm}(G[:, j])$;
      $C[:, :, i] \leftarrow G[:, j]/\mathbf{D}[:, :, i]$;
   **end**
   $V[:, :, i] \leftarrow \tilde{Q}[:, :, i]C[:, :, i]$;     /* Calculate CLVs */
   $\Lambda^c[:, i] \leftarrow \log(\text{diag}[\mathbf{D}[:, :, i]])/dt$ ;     /* Calculate FTCLEs */
**end**

---

improve the performance of an ESN [32,36]. Noise aids the ESN to generalize to unseen data. In Sect. 3 we add Gaussian noise with a zero mean and standard deviation, $\sigma_n = \delta\sigma_y$, where $\delta = 0.05\%$, and $\sigma_y$ is the standard deviation of the data component-wise. We consider the Lorenz 96 with $D = 10$ degrees of freedom and $F = 8$, such that the system is chaotic. We increase the noise intensity to $\delta = \{0.5\%, 5\%, 10\%\}$. We also quantify the effect of less training data by using $100\tau_\lambda$ and $500\tau_\lambda$ long time series, i.e. 1/10 and half of the $1000\tau_\lambda$ long time series that we used in Sect. 3. Figure 12 shows the effects in the Lyapunov spectrum. For 12a, where the training set is $100\tau_\lambda$ long, there is a good agreement between the target (black squares) and the ESN (coloured points) positive exponents. As

expected, a gradual deterioration appears as the noise increases. In 12b for a $500\tau_\lambda$ long training set, the agreement is good for all exponents with a smaller difference for negative exponents compared to (a). After training $N_{\text{ESN}} = 10$ statistically independent networks with chaotic time series, some might eventually evolve towards a fixed point or a periodic orbit instead (i.e. they show spurious behaviour). Here, for $100\tau_\lambda$ long training time series, no ESN evolves spuriously at 0.05% and 0.5% noise. However, at 10% noise, half of the networks show spurious evolution, and are discarded at post-processing. Instead, for $500\tau_\lambda$ long training time series, one and two out of ten evolves spuriously at 0.05% and 0.5% noise, respectively, but none at 5% and 10% noise, which ensures robustness of the network.

As a further test, in Fig. 13 we consider the minimum angles between subspaces spanned by CLVs. In 13a–c the ESNs are trained with $100\tau_\lambda$ long time series, and accordingly in 13d–f with $500\tau_\lambda$. Overall, the results are in good agreement with the target ensuring the robustness of the ESN. A slight and gradual disagreement is observed as the noise intensity increases, in particular for $\theta_{U,N}$.
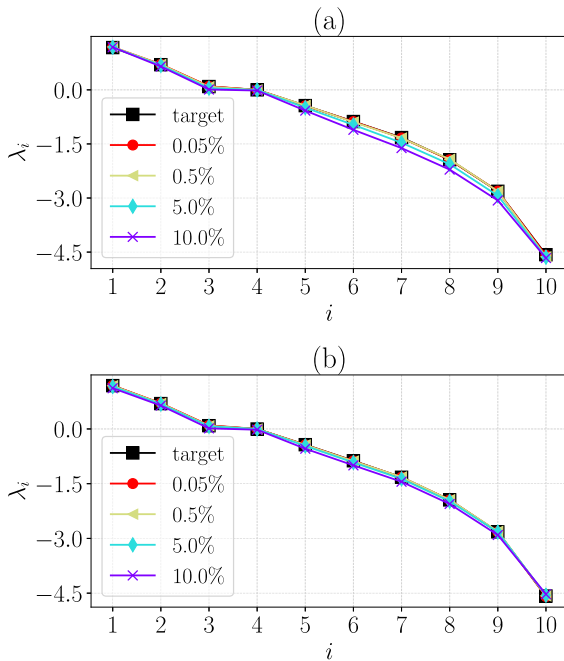
**Fig. 12** Lyapunov spectrum of Lorenz 96 trained with **a** $100\tau_\lambda$ and **b** $500\tau_\lambda$ long time series, and different noise intensity, as indicated in the legend
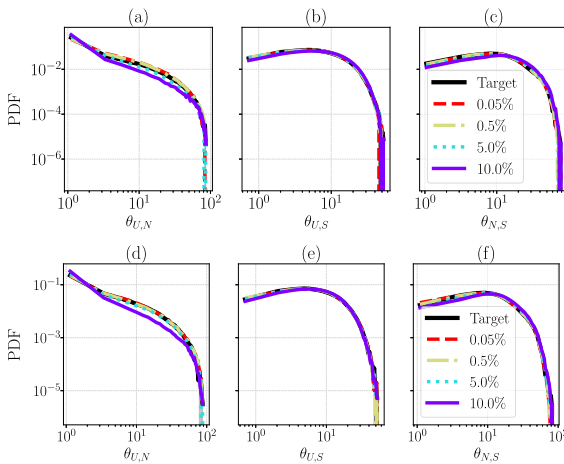


**Fig. 13** PDF of minimum angles between subspaces of CLV from Lorenz 96 trained with **a–c** $100\tau_\lambda$ and **d–f** $500\tau_\lambda$ long time series, and different noise intensity, as indicated in the legend. Both $x$ and $y$ axes are in logarithmic scale and the $x$-axis is in degrees

### B.2 Training with a different loss function

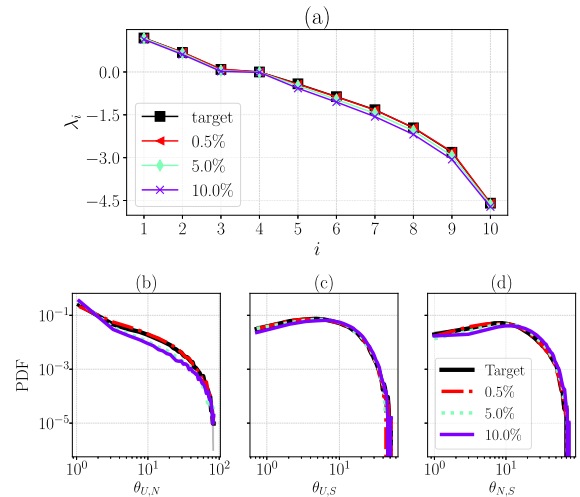The mean square error (MSE), Eq. (16), is a commonly used loss function in the ESN architecture [38]. We



**Fig. 14** Using the mean absolute error, Eq. (27), to train the ESN with $100\tau_\lambda$ long time series from the Lorenz 96, and with different noise intensity, as indicated in the legends. (**a**) Lyapunov spectrum. (**b–d**) PDF of minimum angles between subspaces of CLVs, where both $x$ and $y$ axes are in logarithmic scale and the $x$-axis is in degrees

investigate the effect of using a mean absolute error (MAE) loss function defined as

$$\text{MAE} = \frac{1}{N_{\text{tr}} N_y} \sum_{i=0}^{N_{\text{tr}}} |\mathbf{y}_{\text{p}}(t_i) - \mathbf{y}_{\text{in}}(t_i)|. \tag{27}$$

By comparing the stability properties obtained using the MSE and MAE loss functions, we can gain a better understanding of the potential impact of the choice of loss function on the performance of ESN. In Fig. 14 the results correspond to a $100\tau_\lambda$ long training set, where Eq. 27 was used as a loss function. The Lyapunov spectrum of Fig. 14a is qualitatively similar to Fig. 12a. In practice, training with MAE resulted in less stable ESNs, with increased failures during the test set. For a $100\tau_\lambda$ long training set, at $10\%\sigma_y$ noise with MAE, 80% of ESNs failed, in contrast to 50% with MSE for the same noise. Figure 14b–d are similar to Fig. 13a–c showing minor differences. We also trained the ESNs with $500\tau_\lambda$ long training sets, as in Sect. B.1. Interestingly, we obtain similar results with Figs. 12b and 13d–f, with no significant differences (result not shown).

Based on our analyses, we can conclude that the process of extracting the stability properties of an ESN is robust against higher levels of noise, smaller training sets, and the use of a MAE loss function. Our results suggest that a good practice is to use small to moderate

levels of centered Gaussian noise in the training set, a sufficiently large reservoir size, and a training trajectory of at least $100\tau_\lambda$.

## References

1. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**(2), 130 (1963). https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2

2. Ott, E.: Chaos in Dynamical Systems, 2nd edn. Cambridge University Press, Cambridge (2002). https://doi.org/10.1017/CBO9780511803260

3. Papaphilippou, Y.: Detecting chaos in particle accelerators through the frequency map analysis method. Chaos Interdiscip. J. Nonlinear Sci. **24**(2), 024412 (2014). https://doi.org/10.1063/1.4884495

4. Strogatz, S.H.: Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering. CRC Press, Boca Raton (2015). https://doi.org/10.1201/9780429492563

5. Ruelle, D.: Measures describing a turbulent flow. Ann. N. Y. Acad. Sci. **357**(1), 1 (1980). https://doi.org/10.1111/j.1749-6632.1980.tb29669.x

6. Eckmann, J.P., Ruelle, D.: Ergodic theory of chaos and strange attractors. Rev. Mod. Phys. **57**, 617 (1985). https://doi.org/10.1103/RevModPhys.57.617

7. Oseledets, V.I.: A multiplicative ergodic theorem. Characteristic Ljapunov, exponents of dynamical systems. Trudy Mosk. Mat. Obshchestva **19**, 179 (1968)

8. Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J.M.: Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: Theory. Meccanica **15**(1), 9 (1980). https://doi.org/10.1007/BF02128236

9. Shimada, I., Nagashima, T.: A numerical approach to ergodic problem of dissipative dynamical systems. Progress Theoret. Phys. **61**(6), 1605 (1979). https://doi.org/10.1143/PTP.61.1605

10. Ginelli, F., Chaté, H., Livi, R., Politi, A.: Covariant Lyapunov vectors. J. Phys. A Math. Theor. **46**(25), 254005 (2013). https://doi.org/10.1088/1751-8113/46/25/254005

11. Schubert, S., Lucarini, V.: Covariant Lyapunov vectors of a quasi-geostrophic baroclinic model: analysis of instabilities and feedbacks. Quart. J. R. Meteorol. Soc. **141**(693), 3040 (2015). https://doi.org/10.1002/qj.2588

12. Vannitsem, S., Lucarini, V.: Statistical and dynamical properties of covariant lyapunov vectors in a coupled atmosphere-ocean model–multiscale effects, geometric degeneracy, and error dynamics. J. Phys. A Math. Theor. **49**(22), 224001 (2016). https://doi.org/10.1088/1751-8113/49/22/224001

13. Sharafi, N., Timme, M., Hallerberg, S.: Critical transitions and perturbation growth directions. Phys. Rev. E **96**, 032220 (2017). https://doi.org/10.1103/PhysRevE.96.032220

14. Brugnago, E.L., Gallas, J.A.C., Beims, M.W.: Machine learning, alignment of covariant Lyapunov vectors, and predictability in Rikitake's geomagnetic dynamo model. Chaos Interdiscipl. J. Nonlinear Sci. **30**(8), 083106 (2020). https://doi.org/10.1063/5.0009765

15. Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A.: Determining Lyapunov exponents from a time series. Phys. D Nonlinear Phenom. **16**(3), 285 (1985). https://doi.org/10.1016/0167-2789(85)90011-9

16. Rosenstein, M.T., Collins, J.J., De Luca, C.J.: A practical method for calculating largest Lyapunov exponents from small data sets. Phys. D Nonlinear Phenom. **65**(1), 117 (1993). https://doi.org/10.1016/0167-2789(93)90009-P

17. Takens, F.: Detecting strange attractors in turbulence. In: Rand, D., Young, L.S. (eds.) Dynamical Systems and Turbulence, Warwick 1980, pp. 366–381. Springer, Berlin (1981). https://doi.org/10.1007/BFb0091924

18. Rasp, S., Pritchard, M.S., Gentine, P.: Deep learning to represent subgrid processes in climate models. Proc. Natl. Acad. Sci **115**(39), 9684 (2018). https://doi.org/10.1073/pnas.1810286115

19. Dueben, P.D., Bauer, P.: Challenges and design choices for global weather and climate models based on machine learning. Geosci. Model Dev. **11**(10), 3999 (2018). https://doi.org/10.5194/gmd-11-3999-2018

20. Margazoglou, G., Grafke, T., Laio, A., Lucarini, V.: Dynamical landscape and multistability of a climate model. Proc. R. Soc. A Math. Phys. Eng. Sci. **477**(2250), 20210019 (2021). https://doi.org/10.1098/rspa.2021.0019

21. Verma, S., Novati, G., Koumoutsakos, P.: Efficient collective swimming by harnessing vortices through deep reinforcement learning. Proc. Natl. Acad. Sci. **115**(23), 5849 (2018). https://doi.org/10.1073/pnas.1800923115

22. Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S.: Machine learning-accelerated computational fluid dynamics. Proc. Natl. Acad. Sci. **118**(21), e2101784118 (2021). https://doi.org/10.1073/pnas.2101784118

23. Huhn, F., Magri, L.: Gradient-free optimization of chaotic acoustics with reservoir computing. Phys. Rev. Fluids **7**, 014402 (2022). https://doi.org/10.1103/PhysRevFluids.7.014402

24. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, (2016). http://www.deeplearningbook.org

25. Schäfer, A.M., Zimmermann, H.G.: Recurrent Neural Networks Are Universal Approximators. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) Artificial Neural Networks-ICANN 2006, pp. 632–640. Springer, Berlin (2006)

26. Grigoryeva, L., Ortega, J.P.: Echo state networks are universal. Neural Netw. **108**, 495 (2018). https://doi.org/10.1016/j.neunet.2018.08.025

27. Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, J.: Neural Ordinary Differential Equations. In: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (eds), Advances in Neural Information Processing Systems. Curran Associates, Inc., vol. 31. (2018). https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf

28. Lu, Z., Hunt, B.R., Ott, E.: Attractor reconstruction by machine learning. Chaos Interdiscip. J. Nonlinear Sci. **28**(6), 061104 (2018). https://doi.org/10.1063/1.5039508

29. Pathak, J., Lu, Z., Hunt, B.R., Girvan, M., Ott, E.: Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. Chaos Interdiscip. J. Non-

linear Sci. **27**(12), 121102 (2017). https://doi.org/10.1063/1.5010300

30. Pathak, J., Hunt, B., Girvan, M., Lu, Z., Ott, E.: Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. Phys. Rev. Lett. **120**, 024102 (2018). https://doi.org/10.1103/PhysRevLett.120.024102

31. Vlachas, P.R., Byeon, W., Wan, Z.Y., Sapsis, T.P., Koumoutsakos, P.: Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. Proc. R. Soc. A Math. Phys. Eng. Sci. **474**(2213), 20170844 (2018). https://doi.org/10.1098/rspa.2017.0844

32. Vlachas, P., Pathak, J., Hunt, B., Sapsis, T., Girvan, M., Ott, E., Koumoutsakos, P.: Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. Neural Netw. **126**, 191 (2020). https://doi.org/10.1016/j.neunet.2020.02.016

33. Borra, F., Vulpiani, A., Cencini, M.: Effective models and predictability of chaotic multiscale systems via machine learning. Phys. Rev. E **102**, 052203 (2020). https://doi.org/10.1103/PhysRevE.102.052203

34. Doan, N., Polifke, W., Magri, L.: Physics-informed echo state networks. J. Comput. Sci. **47**, 101237 (2020). https://doi.org/10.1016/j.jocs.2020.101237

35. Doan, N.A.K., Polifke, W., Magri, L.: Short- and long-term predictions of chaotic flows and extreme events: a physics-constrained reservoir computing approach. Proc. R. Soc. A Math. Phys. Eng. Sci. **477**(2253), 20210135 (2021). https://doi.org/10.1098/rspa.2021.0135

36. Racca, A., Magri, L.: Robust optimization and validation of echo state networks for learning chaotic dynamics. Neural Netw. **142**, 252 (2021). https://doi.org/10.1016/j.neunet.2021.05.004

37. Kuramoto, Y.: Diffusion-induced chaos in reaction systems. Prog. Theor. Phys. Suppl. **64**, 346 (1978). https://doi.org/10.1143/PTPS.64.346

38. Lukoševičius, M.: A Practical Guide to Applying Echo State Networks, pp. 659–686. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-35289-8_36

39. Rössler, O.: An equation for continuous chaos. Phys. Lett. A **57**(5), 397 (1976). https://doi.org/10.1016/0375-9601(76)90101-8

40. Charney, J.G., DeVore, J.G.: Multiple flow equilibria in the atmosphere and blocking. J. Atmos. Sci. **36**(7), 1205 (1979). https://journals.ametsoc.org/view/journals/atsc/36/7/1520-0469_1979_036_1205_mfeita_2_0_co_2.xml?tab_body=pdf

41. Lorenz, E.: Predictability: a problem partly solved. In: Seminar on Predictability, 4-8 September 1995, vol. 1. ECMWF, vol. 1, pp. 1–18. ECMWF, Shinfield Park, Reading, (1995). https://www.ecmwf.int/node/10829

42. Ginelli, F., Poggi, P., Turchi, A., Chaté, H., Livi, R., Politi, A.: Characterizing dynamics with covariant Lyapunov vectors. Phys. Rev. Lett. **99**, 130601 (2007). https://doi.org/10.1103/PhysRevLett.99.130601

43. Boffetta, G., Cencini, M., Falcioni, M., Vulpiani, A.: Predictability: a way to characterize complexity. Phys. Rep. **356**(6), 367 (2002). https://doi.org/10.1016/S0370-1573(01)00025-4

44. Huhn, F., Magri, L.: Stability, sensitivity and optimisation of chaotic acoustic oscillations. J. Fluid Mech. **882**, A24 (2020). https://doi.org/10.1017/jfm.2019.828

45. Ruelle, D.: Differentiation of SRB states. Commun. Math. Phys. **187**(1), 227 (1997). https://doi.org/10.1007/s002200050134

46. Lucarini, V., Blender, R., Herbert, C., Ragone, F., Pascale, S., Wouters, J.: Mathematical and physical ideas for climate science. Rev. Geophys. **52**(4), 809 (2014). https://doi.org/10.1002/2013RG000446

47. Blonigan, P.J., Fernandez, P., Murman, S.M., Wang, Q., Rigas, G., Magri, L.: Toward a chaotic adjoint for LES. Center Turbul. Res. Proc. Summer Program (2017). https://doi.org/10.17863/CAM.35422

48. Wormell, C.L.: Non-hyperbolicity at large scales of a high-dimensional chaotic system. Proc. R. Soc. A Math. Phys. Eng. Sci. **478**(2261), 20210808 (2022). https://doi.org/10.1098/rspa.2021.0808

49. Kuznetsov, S.P.: Possible Occurrence of Hyperbolic Attractors, pp. 35–56. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-23666-2_2

50. Gallavotti, G., Cohen, E.G.D.: Dynamical ensembles in nonequilibrium statistical mechanics. Phys. Rev. Lett. **74**, 2694 (1995). https://doi.org/10.1103/PhysRevLett.74.2694

51. Gallavotti, G., Cohen, E.G.D.: Dynamical ensembles in stationary states. J. Stat. Phys. **80**(5), 931 (1995)

52. Lebowitz, J.L., Spohn, H.: A Gallavotti-Cohen-type symmetry in the large deviation functional for stochastic dynamics. J. Stat. Phys. **95**(1), 333 (1999). https://doi.org/10.1023/A:1004589714161

53. Lepri, S., Livi, R., Politi, A.: Energy transport in anharmonic lattices close to and far from equilibrium. Phys. D Nonlinear Phenom. **119**(1), 140 (1998). https://doi.org/10.1016/S0167-2789(98)00076-1

54. Yang, H.L., Takeuchi, K.A., Ginelli, F., Chaté, H., Radons, G.: Hyperbolicity and the effective dimension of spatially extended dissipative systems. Phys. Rev. Lett. **102**, 074102 (2009). https://doi.org/10.1103/PhysRevLett.102.074102

55. Viennet, A., Vercauteren, N., Engel, M., Faranda, D.: Guidelines for data-driven approaches to study transitions in multiscale systems: the case of lyapunov vectors (2022). https://doi.org/10.48550/ARXIV.2203.10322. arXiv:2203.10322

56. Martin, C., Sharafi, N., Hallerberg, S.: Estimating covariant Lyapunov vectors from data. Chaos Interdiscip. J. Nonlinear Sci. **32**(3), 033105 (2022). https://doi.org/10.1063/5.0078112

57. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ, Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P.: SciPy 1.0: Fundamental algorithms for scientific computing in python. SciPy 1.0 contributors. Nat. Methods **17**, 261 (2020). https://doi.org/10.1038/s41592-019-0686-2

58. Kuptsov, P.V., Kuznetsov, S.P.: Lyapunov analysis of strange pseudohyperbolic attractors: angles between tangent subspaces, local volume expansion and contraction. Regul.

Chaotic Dyn. **23**(7), 908 (2018). https://doi.org/10.1134/S1560354718070079

59. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735 (1997). https://doi.org/10.1162/neco.1997.9.8.1735

60. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734. Association for Computational Linguistics, Doha, Qatar (2014). https://doi.org/10.3115/v1/D14-1179. https://aclanthology.org/D14-1179

61. Werbos, P.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78**(10), 1550 (1990). https://doi.org/10.1109/5.58337

62. Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science **304**(5667), 78 (2004). https://doi.org/10.1126/science.1091277

63. Huhn, F., Magri, L.: Learning Ergodic Averages in Chaotic Systems. In: Krzhizhanovskaya, V.V., Závodszky, G., Lees, M.H., Dongarra, J.J., Sloot, P.M.A., Brissos, S., Teixeira, J. (eds.) Computational Science-ICCS 2020, pp. 124–132. Springer International Publishing, Cham (2020)

64. Racca, A., Magri, L.: Data-driven prediction and control of extreme events in a chaotic flow (2022). https://doi.org/10.48550/ARXIV.2204.11682. arXiv:2204.11682

65. Tikhonov, A.N., Goncharsky, A., Stepanov, V., Yagola, A.G.: Numerical methods for the solution of ill-posed problems, vol. 328. Springer Science & Business Media, London (1995). https://doi.org/10.1007/978-94-015-8480-7

66. Hoffman, M., Brochu, E., de Freitas, N.: Portfolio allocation for Bayesian optimization. In: Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, pp. 327–336. AUAI Press, Arlington, Virginia, USA, UAI'11 (2011)

67. Lukoševičius, M., Uselis, A.: Efficient Cross-Validation of Echo State Networks. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) Artificial Neural Networks and Machine Learning - ICANN 2019: Workshop and Special Sessions, pp. 121–133. Springer International Publishing, Cham (2019)

68. De Swart, H.: Physica D: Analysis of a six-component atmospheric spectral model: chaos, predictability and vacillation. Nonlinear Phenom. **36**(3), 222 (1989). https://doi.org/10.1016/0167-2789(89)90082-1

69. Crommelin, D.T., Opsteegh, J.D., Verhulst, F.: A Mechanism for Atmospheric Regime Behavior. J. Atmos. Sci. **61**(12), 1406 (2004). https://doi.org/10.1175/1520-0469(2004)061<1406:AMFARB>2.0.CO;2

70. Woollings, T., Barriopedro, D., Methven, J., Son, S.W., Martius, O., Harvey, B., Sillmann, J., Lupo, A.R., Seneviratne, S.: Blocking and its response to climate change. Curr. Clim. Chang. Rep. **4**(3), 287 (2018). https://doi.org/10.1007/s40641-018-0108-z

71. Schubert, S., Lucarini, V.: Dynamical analysis of blocking events: spatial and temporal fluctuations of covariant Lyapunov vectors. Q. J. R. Meteorol. Soc. **142**(698), 2143 (2016). https://doi.org/10.1002/qj.2808

72. Ruelle, D.: Large volume limit of the distribution of characteristic exponents in turbulence. Commun. Math. Phys. **87**(2), 287 (1982). https://doi.org/10.1007/BF01218566

73. Karimi, A., Paul, M.R.: Extensive chaos in the Lorenz-96 model. Chaos Interdiscip. J. Nonlinear Sci. **20**(4), 043105 (2010). https://doi.org/10.1063/1.3496397

74. Kuptsov, P.V., Kuznetsov, S.P.: Violation of hyperbolicity in a diffusive medium with local hyperbolic attractor. Phys. Rev. E **80**, 016205 (2009). https://doi.org/10.1103/PhysRevE.80.016205

75. Knyazev, A.V., Argentati, M.E.: Principal angles between subspaces in an A-based scalar product: Algorithms and perturbation estimates. SIAM J. Sci. Comput. **23**(6), 2008 (2002). https://doi.org/10.1137/S1064827500377332

76. Kaplan, J.L., Yorke, J.A.: Chaotic behavior of multidimensional difference equations. In: Functional Differential Equations and Approximation of Fixed Points, pp. 204–227. Springer (1979)