

Gradient-free optimization of chaotic acoustics with reservoir computing

Original

Gradient-free optimization of chaotic acoustics with reservoir computing / Huhn, F.; Magri, L.. - In: PHYSICAL REVIEW FLUIDS. - ISSN 2469-990X. - 7:1(2022), pp. 1-29. [10.1103/PhysRevFluids.7.014402]

Availability:

This version is available at: 11583/2995085 since: 2024-12-23T22:40:54Z

Publisher:

American Physical Society - APS

Published

DOI:10.1103/PhysRevFluids.7.014402

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Gradient-free optimization of chaotic acoustics with reservoir computing

Francisco Huhn

University of Cambridge, Department of Engineering, United Kingdom

Luca Magri*

*Imperial College London, Aeronautics Department, United Kingdom
University of Cambridge, Department of Engineering, United Kingdom
The Alan Turing Institute, United Kingdom and
Institute of Advanced Study, TU Munich, Germany (visiting)*

(Dated: February 3, 2022)

Gradient-based optimization of chaotic acoustics is challenging for a threefold reason: (i) first-order perturbations grow exponentially in time; (ii) the statistics of the solution may have a slow convergence; (iii) and the time-averaged acoustic energy may physically have discontinuous variations, which means that the gradient does not exist for some design parameters. We develop a versatile optimization method, which finds the design parameters that minimize time-averaged acoustic cost functionals, and overcomes the three aforementioned challenges. The method is gradient-free, model-informed, and data-driven with reservoir computing based on echo state networks. First, we analyse the predictive capabilities of echo state networks in thermoacoustics both in the short- and long-time prediction of the dynamics. We find that both fully data-driven and model-informed architectures are able to learn the chaotic acoustic dynamics, both time-accurately and statistically. Informing the training with a physical reduced-order model with one acoustic mode markedly improves the accuracy and robustness of the echo state networks, while keeping the computational cost low. Echo state networks offer accurate predictions of the long-time dynamics, which would be otherwise expensive by integrating the governing equations to evaluate the time-averaged quantity to optimize. Second, we couple echo state networks with a Bayesian technique to explore the design thermoacoustic parameter space. The computational method is minimally intrusive because it requires only the initialization of the physical and hyperparameter optimizers. Third, we find the set of flame parameters that minimize the time-averaged acoustic energy of chaotic oscillations, which are caused by the positive feedback with a heat source, such as a flame in gas turbines or rocket motors. These oscillations are known as thermoacoustic oscillations. The optimal set of flame parameters is found with the same accuracy as brute-force grid search, but with a convergence rate that is more than one order of magnitude faster. This work opens up new possibilities for non-intrusive (“hands-off”) optimization of chaotic systems, in which the cost of generating data, for example from high-fidelity simulations and experiments, is high.

I. INTRODUCTION

When the heat released by a flame is sufficiently in phase with the acoustic waves of a confined environment, such as a gas turbine, thermoacoustic oscillations can arise [1]. Physically, thermoacoustic oscillations occur when the thermal power released by the flame, which is converted into acoustic energy, exceeds fluid mechanic dissipation. In gas turbines and rocket motors, thermoacoustic oscillations are unwanted because they can cause structural vibrations, fatigue, noise, and, if uncontrolled, can shake the device apart. Therefore, the objective of manufacturers is to design and operate stable devices [2–4]. The preliminary design of thermoacoustic systems is based on linear analysis, in which the growth rates of infinitesimal oscillations is computed on top of a time-independent baseline solution. If no growth rate is positive, the system is linearly stable [2, 5–7]. If the system is linearly unstable, sensitivity methods, which are based on gradient computation, have been introduced to answer the practitioners’ question “How can we change the design parameters to reduce the growth rate of infinitesimal oscillations?”. In particular, adjoint methods proved computationally cheap tools to calculate the gradients of an eigenvalue with respect to all parameters of interest [7–11] with higher-order corrections [12]. Adjoint gradients were applied to the optimization of a longitudinal combustor [13] and annular combustors [e.g., 14, 15].

Although linear analysis provides valuable information on the system’s stability and sensitivity, thermoacoustic oscillations are inherently nonlinear. First, the heat release varies nonlinearly with the acoustics, which perturb the flame dynamics [5, 16]. Second, hydrodynamic instabilities (e.g. vortex shedding), which are promoted by the

* lm547@cam.ac.uk

geometry of the combustor, can modulate the flame dynamics and, thus, the heat release rate [17]. Because of these nonlinearities, a thermoacoustic system may be stable to infinitesimal perturbations (i.e. linearly stable), but finite-amplitude perturbations can trigger sustained oscillations [18], i.e. in the bistable region of a subcritical bifurcation. These sustained oscillations can be periodic, quasi-periodic or chaotic, as shown in experiments [19–24] and numerical studies [25–27], to name a few. Among these nonlinear regimes, chaotic oscillations are the most intractable to optimization [17, 28].

Chaotic oscillations are extremely sensitive to infinitesimal perturbations [29], which results in an exponential growth of infinitesimal perturbations. In other words, the tangent space is unstable¹. Because of this, the calculation of gradients of ergodic averages, i.e. *time-averaged* quantities of interest, is intractable with traditional sensitivity methods. This roadblock motivated the development of alternative gradient-based methods, which can be grouped into six categories: (i) ensemble methods [30–32], which average the gradient over an ensemble of short time trajectories; (ii) probability density methods [33, 34], which calculate the gradient from the change in the probability density function of the chaotic attractor; (iii) unstable periodic orbits [35], which decompose the chaotic attractor into unstable periodic orbits and compute their gradients; (iv) fluctuation-dissipation-theorem methods [36–38], which compute the mean linear response of a system to small changes in external forcing; (v) shadowing methods [39–43], which average over time the difference between a baseline trajectory and its shadowing trajectory, and (vi) recent developments on linear response theory [44, 45]. In particular, shadowing methods have successfully computed first-order sensitivities of time-averaged energies in fluid mechanics [46, 47]. In thermoacoustics, shadowing-based gradients were embedded into a gradient update routine for design optimization [28], in which the time-averaged acoustic energy was minimized by computing the optimal set of flame parameters. The study highlighted three challenges in gradient-based optimization of chaotic thermoacoustics. First, thermoacoustic systems physically exhibit an abundance of bifurcations, across which the time-averaged cost functional being optimized can be discontinuous [17, 28]. Second, shadowing-based methods require a number of tangent (i.e. first-order perturbation) solutions equal to the number of positive Lyapunov exponents [41], which can bear a significant computational cost. Third, the nonlinear dynamics of thermoacoustics can be non-hyperbolic, i.e. the covariant Lyapunov basis may become defective [48], for certain design parameters [28]. This means that gradients cannot be guaranteed to exist for all thermoacoustic design parameters, which can hinder, and can even prevent, the optimization process via gradient-update. In this paper, we develop a gradient-free optimization methodology to find the optimal design parameters that minimize the time-averaged acoustic energy.

In either gradient-based and gradient-free methods, the solution must be integrated sufficiently long in time (ideally, ad infinitum), such that the quantities of interest (gradient in gradient-based, and cost functional in gradient-free methods) have converged to within a desired precision². The generation of such a time series, however, can carry a high cost. As an alternative to the integration of the governing equations, we propose the use of a data-driven technique to produce accurate predictions of the system’s dynamics to generate the required long time series. Such a task naturally falls within the category of supervised learning for time-dependent systems. In time-dependent problems, the order by which data is sorted (i.e. time) is of paramount importance. Feed-forward neural networks are a classic architecture, which works well in regression problems [49], but it does not naturally include recurrences to accurately learn the temporal correlations. To extend feed forward networks to sequential data, recurrent neural networks have an internal state, which is updated by taking into account both the current input and the previous state. Thus, the sequence by which data is fed affects the internal state, and therefore its output. Within recurrent neural networks, three architectures are highlighted: (i) Long Short-Term Memory networks (LSTM) [50], (ii) Gated Recurrent Unit networks (GRU) [51], and (iii) Echo State Networks (ESN) [52, 53]. While the three have been successfully employed to learn and predict time-dependent problems, the ESN architecture offers an advantage, which is exploited in this paper. Because its output is a linear combination of the hidden state variables, its training reduces to a least-squares problem, which is more computationally robust than repeatedly calculating gradients, as in LSTM and GRU networks. In chaotic learning, ESNs have been recently explored in multiple applications in chaotic systems, from time-accurate prediction [54, 55], to the reconstruction of hidden variables [56–58], or the calculation of ergodic quantities [59]. In particular, in Huhn and Magri [59], ESNs were employed in the prediction of the long-time average of a thermoacoustic dynamical system. Moreover, Hart *et al.* [60] proved analytically that, under certain conditions, ESNs can approximate the invariant measure of a dynamical system, which is key to the calculation of accurate statistical quantities. In this paper, we employ ESNs for the generation of sufficiently long time series, from which the time-averaged cost functional to be optimized is evaluated. Specifically, we apply this framework to predict the time-averaged thermoacoustic energy.

The objective of this paper is three-fold. First, we propose a versatile gradient-free methodology to optimize time-averaged cost functionals. The methodology requires a minimal number of user-defined parameters, which makes it a

¹ In other words, at least one Lyapunov exponent is positive.

² Both converge with $t^{-1/2}$, where t is time.

minimally-intrusive tool. We apply the methodology to a chaotic thermoacoustic system. Second, we investigate the capability of ESNs of learning thermoacoustic solutions from small data. Both short- and long-time predictions are analyzed. Third, we minimize a chaotic thermoacoustic oscillation by finding the optimal set of design parameters.

The paper is structured as follows. Section II presents the general optimization problem with a focus on the thermoacoustic system. Section III introduces the proposed gradient-free optimization method, both in general and in particular case of this paper. The method combines the tools of Section IV, which describes Bayesian optimization; and Section V, which presents both the traditional and hybrid echo state networks. Section VI A investigates the short- and long-time predictions of the ESNs in learning thermoacoustic dynamics. Section VI B applies the framework of Section III to the optimization of a chaotic thermoacoustic system. A final discussion and conclusions end the paper. We have also included a discussion of the potential cost benefit of the proposed optimization framework in Appendix C.

II. PROBLEM FORMULATION AND PHYSICAL MODELS

We consider a nonlinear dynamical system

$$\frac{d\mathbf{q}}{dt} = \mathbf{F}(\mathbf{q}, \mathbf{s}), \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^{N_d}$ is the state vector; $\mathbf{F} : \mathbb{R}^{N_d} \rightarrow \mathbb{R}^{N_d}$ is a nonlinear operator; $\mathbf{s} \in \mathbb{R}^{N_p}$ is a vector of physical (or design) parameters; and N_d is the number of degrees of freedom of the system. Given an initial condition, \mathbf{q}_0 , equation (1) can be solved to obtain a solution $\mathbf{q}(t, \mathbf{s})$. We wish to optimize the time-average of a cost functional

$$\langle \mathcal{J} \rangle(\mathbf{s}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \mathcal{J}(\mathbf{q}(t, \mathbf{s}), \mathbf{s}) dt, \quad (2)$$

where \mathcal{J} is, for example, an energy. Because we consider ergodic systems, $\langle \mathcal{J} \rangle$ does not depend on the initial condition or trajectory, but it depends only on the parameters, \mathbf{s} . The goal is to find a set of parameters, \mathbf{s}^+ , that minimize the time-averaged cost functional in Eq. 2. Mathematically, \mathbf{s}^+ is the solution of

$$\min_{\mathbf{s}} \langle \mathcal{J} \rangle, \quad (3)$$

$$\mathbf{G}(\mathbf{s}) = 0, \quad (4)$$

$$\mathbf{H}(\mathbf{s}) \geq 0, \quad (5)$$

where \mathbf{G} and \mathbf{H} are equality and inequality constraints, respectively. The inequality constraints guarantee that the physical parameters are searched in a feasible region.

A. Thermoacoustic dynamical system

We consider an acoustic resonator that consists of a tube and a heat source in it. We assume that the cut-off frequency of the acoustic resonator is sufficiently high such that only longitudinal acoustics propagate. The mean flow is assumed to have a zero Mach number with a spatially averaged temperature. The equations that govern the acoustics are the linearized momentum and energy equations

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = 0, \quad (6)$$

$$\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \zeta p - \dot{q} \delta(x - x_f) = 0, \quad (7)$$

where u , p , ζ , \dot{q} , δ and x_f are the acoustic velocity, pressure, damping, heat-release rate, Dirac delta and flame position, respectively, which are non-dimensionalized as in [28, 61]. The axial coordinate is $x \in [0, 1]$, which is non-dimensionalized by the tube length. The heat release rate is given by a modified King's law [62–65]

$$\dot{q}(t) = \beta[(1 + u(x_f, t - \tau))^{1/2} - 1], \quad (8)$$

where β and τ are the heat release intensity and time delay, respectively. The time delay models the time that the heat release takes to be perturbed by an acoustic perturbation at the base of the heat source. The solutions are

decomposed in N_g acoustic eigenfunctions of the undamped acoustic system [66], which is also known as Galerkin decomposition,

$$u(x, t) = \sum_{j=1}^{N_g} \eta_j(t) \cos(j\pi x), \quad (9)$$

$$p(x, t) = - \sum_{j=1}^{N_g} \mu_j(t) \sin(j\pi x), \quad (10)$$

which results in a system of $2N_g$ oscillators that are nonlinearly coupled by the heat source

$$\frac{d\eta_j}{dt} - j\pi\mu_j = 0, \quad (11)$$

$$\frac{d\mu_j}{dt} + j\pi\eta_j + \zeta_j\mu_j + 2\dot{q} \sin(j\pi x_f) = 0, \quad (12)$$

where $\zeta_j = c_1 j + c_2 j^{1/2}$ is the modal damping, which damps out higher-frequency oscillations according to physical scaling [67]. Despite its simplicity, the thermoacoustic model in equations (6) to (8) qualitatively captures complex nonlinear dynamics and bifurcations, as shown in [18, 28, 61, 66]. Because we wish to use numerical integrators and echo state networks, which march from time step n to $n + 1$, it is convenient to transform the time-delayed problem equation (8) into an initial value problem. To achieve this, we model the advection of a dummy variable v with velocity τ^{-1} as [28]

$$\frac{\partial v}{\partial t} + \frac{1}{\tau} \frac{\partial v}{\partial X} = 0, \quad 0 \leq X \leq 1, \quad (13)$$

$$v(X = 0, t) = u_f(t). \quad (14)$$

The time-delayed velocity is provided by the value of v at the right boundary, i.e. $u_f(t - \tau) = v(X = 1, t)$. Equation (13) is discretized using $N_c + 1$ points with a Chebyshev spectral method [68], which adds N_c degrees of freedom. Thus, these equations define a dynamical system with a state vector $\mathbf{q} = [\eta_1, \dots, \eta_{N_g}, \mu_1, \dots, \mu_{N_g}, v_1, \dots, v_{N_c}]$. This model, which is used as a proof of concept, qualitatively captures the key physics of nonlinear thermoacoustics [28]. The cost functionals that we wish to obtain and minimize are the time averages of the acoustic energy and the Rayleigh index [17]

$$E_{\text{ac}}(t) = \int_0^1 \frac{1}{2} (u^2(t) + p^2(t)) dx = \frac{1}{4} \sum_{j=1}^{N_g} (\eta_j^2(t) + \mu_j^2(t)), \quad (15)$$

$$I_{\text{Ra}}(t) = p_f(t)\dot{q}(t). \quad (16)$$

The first measures the total energy of the acoustic oscillations, while the second corresponds to the rate of input energy in the system, which is balanced out over time by the damping. (As shown in [28], the two time-averaged cost functionals are one-to-one related to each other, therefore, we focus on the optimization of the acoustic energy only.) In this work, the following parameters are fixed: $x_f = 0.2$, $c_1 = 0.1$ and $c_2 = 0.06$ [18]. Unless otherwise specified, we use 10 Galerkin modes (i.e. $N_g = 10$) and 11 Chebyshev points (i.e. $N_c = 10$), which provide a good compromise between accuracy and computational cost [28]. We solve the equations numerically in time with a 3-stage Runge-Kutta solver [69], with a time step of 0.01 time units.

III. GRADIENT-FREE DESIGN OPTIMIZATION WITH ECHO STATE NETWORKS

We introduce the proposed methodology to optimize a chaotic system with a non-intrusive approach, using echo state networks. The flowchart in Figure 1a illustrates the method. There are two optimizers, one for the physical parameters and another for the hyperparameters. The physical optimizer chooses the next point in the physical space to be evaluated. By integrating the ordinary differential equations that govern the thermoacoustic dynamics (ODEs), a short amount of data is generated, which is used to train the network. This mimics data from high-fidelity simulation or experiments, which is sparse and costly, the objective being to gain as much information as possible with a minimal number of samples. Then, the hyperparameter optimizer selects the optimal hyperparameters. With the hyperparameters tuned, the data-driven model (echo state networks in this case) runs in predictive mode for a user-defined sufficiently long time to obtain the long-time average of the physical cost functional, which is returned to the physical parameter optimizer. The only human intervention is at the start of the chain for the initialization of the

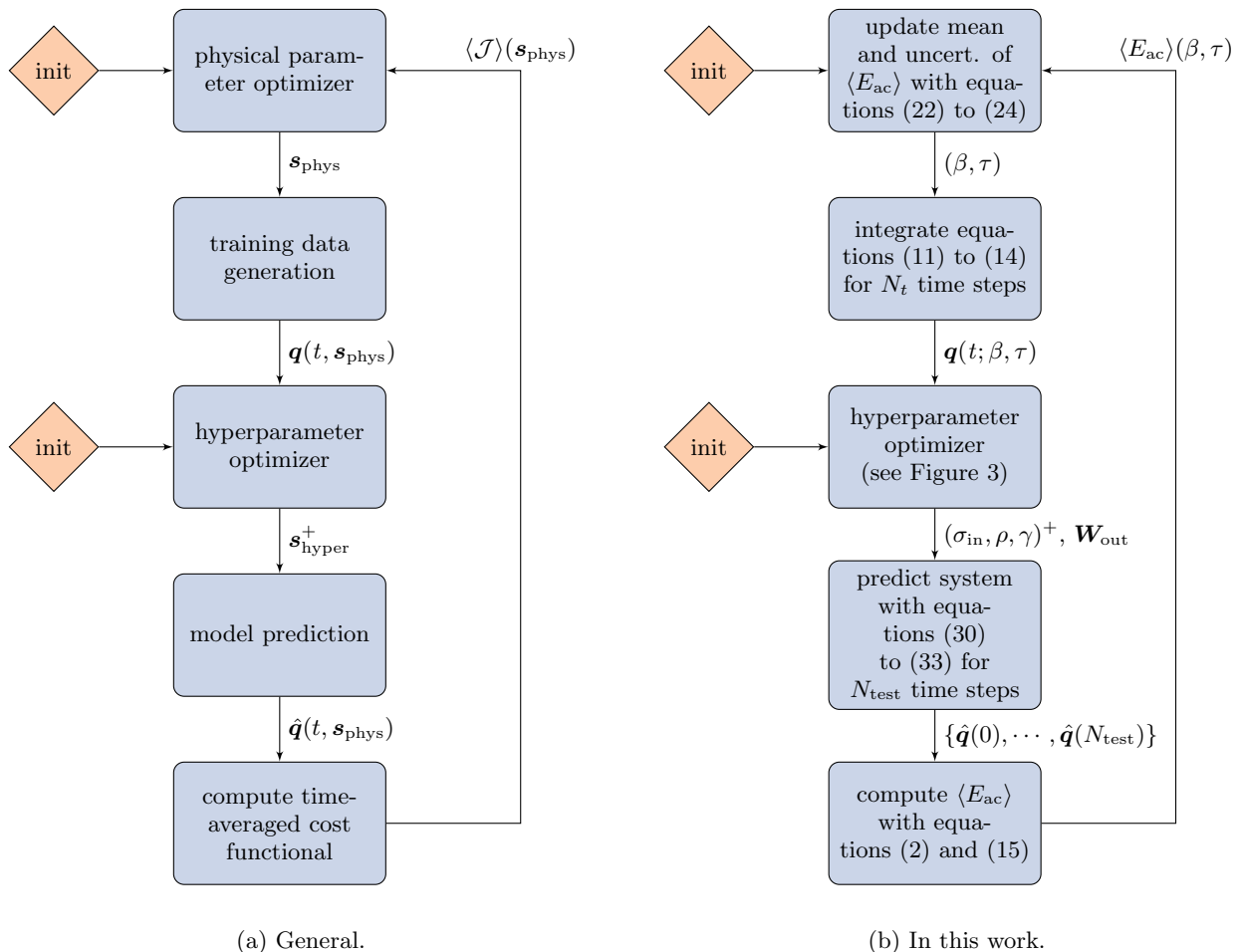


FIG. 1: Optimisation chain flowcharts. Human interaction is only required in the initialisation (**init** steps). The initialisation defines the search space, maximum number of evaluations, optimiser parameters, kernel functions, etc.

optimizers. After initialization (e.g. defining search space, maximum number of evaluations, etc.), the optimization chain runs on its own. Figure 1b depicts the chain of Figure 1a in the present work. In particular, in this work, the physical optimizer is a Bayesian optimizer using Gaussian Process Regression (GPR), with a Matérn 3/2 kernel. Because of the nature of a Gaussian Process (Section IV A), the dependent variable (in this case, the acoustic energy, E_{ac}) can be negative. Because a negative acoustic energy is unphysical, we apply the GPR to the logarithm of the acoustic energy, which means that values of the acoustic energy are modelled by a log-normal distribution. Hence, the estimates are positive and the standard deviation is additive in the exponent only. Finally, because of interpretability, we use the Lowest Confidence Bound (see Section IV B) as the acquisition function, with $\kappa = 1.960$, which corresponds to a 95% confidence interval. The ordinary differential equations are integrated using a 3-stage Runge-Kutta scheme [69]. The hyperparameters are also selected via Bayesian optimization with GPR, but with an RBF kernel and GP-Hedge acquisition function. The system is predicted with a hybrid echo state network model and the cost functional is the time-averaged acoustic energy, calculated with the prediction from the network. All these concepts are introduced in the following two sections.

IV. BAYESIAN OPTIMIZATION WITH GAUSSIAN PROCESS REGRESSION

Gaussian Process Regression offers an estimate of both the mean and standard deviation of the cost functional. This allows for a more informed choice to be made and for better control of the balance between exploration and exploitation (see Section IV B). Moreover, it is a global optimization method, which is advantageous when the cost functional is multimodal. We summarize Gaussian Process Regression in Section IV A and Bayesian Optimization in Section IV B.

A. Gaussian Process Regression

A Gaussian Process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution [70]. Here, the random variables are the values of a function on its domain. In fact, the function is deterministic, but in the context of GPs, its (unknown) values are modelled as random variables. A GP is specified by its mean function, $m(\mathbf{x})$, usually set to 0, and the covariance function, often called kernel function, $k(\mathbf{x}, \mathbf{x}')$, which are defined as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \quad (17)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \quad (18)$$

where $f(\mathbf{x})$ is the real process and \mathbb{E} is the expectation. The Gaussian Process is written as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (19)$$

where $\{(\mathbf{x}_i, f_i) | i = 1, \dots, n\}$ is a collection of n data points, from which we construct the output vector \mathbf{f} and the matrix \mathbf{X} , whose columns are the vectors \mathbf{x}_i . Similarly, we can define \mathbf{f}_* and \mathbf{X}_* for the n_* test inputs, i.e. the inputs that we wish to predict. According to the definition of a GP, prior to any observations, the joint distribution of \mathbf{f} and \mathbf{f}_* is a Gaussian distribution,

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right), \quad (20)$$

where, for any \mathbf{X}_1 and \mathbf{X}_2 , $\mathbf{K}(\mathbf{X}_1, \mathbf{X}_2)$ is an $n_1 \times n_2$ matrix of covariances evaluated for all pairs of columns (each column being one point) of \mathbf{X}_1 and \mathbf{X}_2 . To include the information from the observations f_i , this distribution is conditioned on the observations [70]

$$\mathbf{f}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*), \quad (21)$$

$$\boldsymbol{\mu}_* = \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{f}, \quad (22)$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*). \quad (23)$$

If the observations are noisy, with variance σ_n^2 , then \mathbf{K} is replaced by $\mathbf{K} + \sigma_n^2 \mathbf{I}$, where \mathbf{I} is the identity matrix.

In this work, we use two kernel functions, the Radial Basis Function,

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right), \quad (24)$$

and the Matérn 3/2 Kernel Function,

$$k(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\sqrt{3}}{l}\|\mathbf{x} - \mathbf{x}'\|\right) \exp\left(-\frac{\sqrt{3}}{l}\|\mathbf{x} - \mathbf{x}'\|\right), \quad (25)$$

where $\|\cdot\|$ is the Euclidean distance and l are tunable length scales, which control the smoothness of the function being regressed. A large l means that the covariance will be high even for relatively distant points, \mathbf{x} and \mathbf{x}' , resulting in a smoother function than that for smaller l . Because we expect the acoustic energy to be smoother in the physical space [28] than the mean squared error in the hyperparameter space [71], we use the RBF kernel for the former and the Matérn kernel for the latter.

B. Bayesian Optimization

Bayesian optimization is used in this work to select the hyperparameters, and to optimize the physical parameters such that the acoustic energy of the system is minimal.

Bayesian optimization consists of a loop of three steps:

1. Observe a point from the optimization domain;
2. Update the mean, $\boldsymbol{\mu}_*$, and uncertainty, $\boldsymbol{\Sigma}_*$;

3. Select the next point to observe by finding the minimum of the acquisition function.

The first step simply evaluates the function, f , at the given point \mathbf{x} . The second step calculates the mean and variance of the distribution of equations (22) and (23). Finally, in the third step, the new point to observe corresponds to the optimum of an acquisition function.

An acquisition function takes into account both the mean and uncertainty and, for any point \mathbf{x} , outputs a value that relates to either or a combination of the two. This provides the probability, or amount, by which \mathbf{x} can improve the current optimum. There are four common acquisition functions: Probability of Improvement (PI), Expected Improvement (EI), Lowest Confidence Bound (LCB), and GP-Hedge [72]. The first, PI, computes the probability that a candidate \mathbf{x} can improve with respect to the current optimum, $\text{prob}(f(\mathbf{x}) < f^+)$. The second, EI, is similar to PI, but weighs the probability by the potential gain, i.e. it is the expected value of the improvement, $\mathbb{E}[\max(f^+ - f(\mathbf{x}), 0)]$. Finally, LCB is based on intervals of confidence centered around the mean, $[\mu - \kappa\sigma, \mu + \kappa\sigma]$. For a given value of κ , the interval will cover a certain percentage of the outcomes. For example, with $\kappa \approx 1.960$, 95% of the outcomes will be contained in the interval, i.e. a 95% confidence interval³. When the LCB acquisition function is used, one seeks to find \mathbf{x} for which the lower bound of the confidence interval is a minimum, i.e. the \mathbf{x} with the smallest lower confidence bound. This means that κ controls the balance between *exploration* (exploring unobserved regions of the optimization space) and *exploitation* (improving an existing observation by searching close to it), with exploration being preferred when κ is large (κ multiplies the uncertainty σ) and exploitation when κ is small. Finally, GP-Hedge [72] overcomes the difficulty of knowing which acquisition function will perform best by taking the previous three acquisition functions⁴ and probabilistically picking one of the three suggestions to sample next. For each acquisition function, the better the means, μ , of the points suggested in the past, the larger the probability of being chosen. In this paper, we use the LCB because, although it does not necessarily offer the best performance among the four acquisition functions, it is simple to compute and, more importantly, it is simple to physically interpret.

V. ECHO STATE NETWORKS

Echo state networks (ESN) [52, 53] are recurrent neural networks, which are composed of a set of nodes that constitute the reservoir. The ESN receives an input signal, $\mathbf{l}(n) \in \mathbb{R}^{N_l}$, and produces an output signal, $\hat{\mathbf{y}}(n) \in \mathbb{R}^{N_y}$, where n is the discrete time variable, i.e. $t = n \Delta t$. Usually $N_l = N_y = N_d$, where N_d is the dimension of the system being predicted, such that the network can evolve on its own. The state of the reservoir is a vector, \mathbf{r} , of the states of all units, r_j , $j \in \{1, \dots, N_r\}$. The reservoir state evolves according to the nonlinear law

$$\mathbf{r}(n) = \tanh(\mathbf{W}_{\text{in}}[\mathbf{l}(n); b_{\text{in}}] + \mathbf{W}\mathbf{r}(n-1)), \quad (26)$$

where \mathbf{W}_{in} is the input matrix (i.e. $\mathbf{W}_{\text{in}}^{i,j}$ is the weight from the j -th component of the input to the i -th node) and b_{in} is the input bias, with the semicolon denoting row concatenation. Similarly, in the recurrency matrix \mathbf{W} , the component $\mathbf{W}^{i,j}$ is the weight from the j -th node to the i -th node. Therefore, \mathbf{W}_{in} and \mathbf{W} are $N_r \times N_d$ and $N_r \times N_r$ matrices. The hyperbolic tangent in equation (26) is applied entry-wise. Finally, the output is calculated by linear combination of the states of the reservoir units,

$$\hat{\mathbf{y}}(n) = \mathbf{W}_{\text{out}}[\mathbf{r}(n); b_{\text{out}}], \quad (27)$$

where \mathbf{W}_{out} is the output matrix, of size $N_d \times N_r$, and b_{out} is the (scalar) output bias.

The network is trained to produce an output $\hat{\mathbf{y}}$ that matches the target \mathbf{y} by minimizing the mean square error (MSE) [52, 53]

$$\text{MSE} = \frac{1}{N_t} \sum_{n=1}^{N_t} \frac{\|\hat{\mathbf{y}}(n) - \mathbf{y}(n)\|^2}{N_d}, \quad (28)$$

where N_t is the number of (discrete) time steps, and the norm is Euclidean in \mathbb{R}^d . In ESNs, both \mathbf{W}_{in} and \mathbf{W} are generated once and fixed. In this work, each reservoir node is connected to one input, which results in every row of \mathbf{W}_{in} having only one non-zero entry. The weight of the connections is sampled from a uniform distribution in the range $[-\sigma_{\text{in}}, \sigma_{\text{in}}]$, where σ_{in} is a scaling parameter. Hence, \mathbf{W}_{in} can be generated by sampling uniformly from the

³ For an $\alpha\%$ confidence interval, one finds κ such that $\Phi(\mu + \kappa\sigma) - \Phi(\mu - \kappa\sigma) = \alpha$, where Φ is the cumulative distribution function of the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$.

⁴ GP-Hedge can be applied to any combination of acquisition functions, not only PI, EI and LCB.

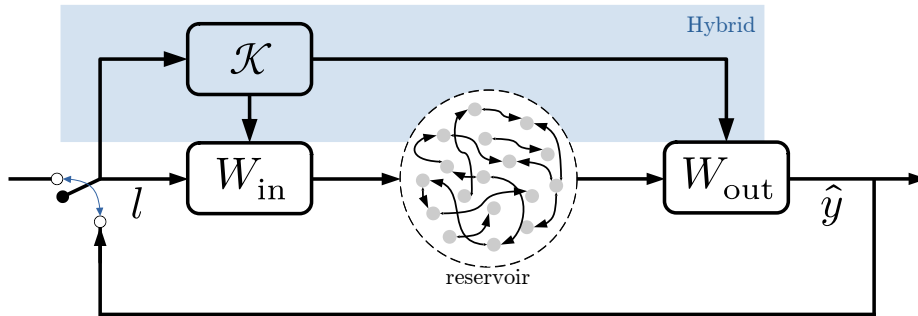


FIG. 2: Schematic of the conventional (without highlighted region) and hybrid echo state networks. The hybrid echo state network corresponds to a conventional ESN with an additional knowledge-based model (\mathcal{K} box) that feeds the reservoir and the output via the augmented \mathbf{W}_{in} and \mathbf{W}_{out} matrices. In training, the switch is horizontal, whereas in prediction, it is vertical.

range $[-1, 1]$ and scaling by σ_{in} directly in equation (26). Similarly, \mathbf{W} is generated by sampling from the uniform distribution in the range $[-1, 1]$, with each node being on average connected to $(1 - \text{sp})N_r$ other nodes, where sp is the desired sparseness. The matrix is then scaled to have a desired spectral radius, ρ , which is typically smaller than unity to satisfy the echo state property [52, 53]. A network that satisfies the echo state property “forgets” an old input after a certain time, which means that, even if starting from two different states, a network with the echo state property will converge to the same trajectory after a certain time (provided it is fed by the same input).

Because \mathbf{W}_{in} and \mathbf{W} are fixed, only the output weights (i.e. the entries of \mathbf{W}_{out}) are trained to solve the minimization problem equation (28) with ridge regression

$$\mathbf{W}_{\text{out}} = (\mathbf{R}^T \mathbf{R} + \gamma \mathbf{I})^{-1} \mathbf{R}^T \mathbf{Y}, \quad (29)$$

where γ is the user-defined Tikhonov factor, which regularises the training. The \mathbf{R} and \mathbf{Y} matrices are obtained by row-concatenating the reservoir states and output targets, i.e. the n -th row corresponds to the discrete time n . During training mode, the network is operated in open-loop, whereas in prediction mode, the output of the network is fed to its input (closed-loop), i.e.

$$\mathbf{l}(n+1) = \hat{\mathbf{y}}(n), \quad (30)$$

for the network to evolve autonomously. This corresponds to the schematic of Figure 2 with the blue highlighted region removed.

A. Hybrid Echo State Network

The hybrid echo state network (hESN) is a variant of the conventional ESN [54]. In the hESN, the capability of the conventional ESN is complemented by (possibly imperfect) physical knowledge from a dynamical system, which may be a reduced-order model. The combination of data and model knowledge achieves higher accuracy, not only in the short time prediction [54, 71], but also in the long time [59]. Figure 2 shows the architecture of an hESN. The network’s input is fed to the reservoir, as in the conventional ESN, and to the physical knowledge based system (marked \mathcal{K}). The output of the physical system, in turn, is passed to both the reservoir, via \mathbf{W}_{in} ; and the output, via \mathbf{W}_{out} . Mathematically, equation (26) and equation (27) are augmented by \mathcal{K}

$$\mathbf{r}(n) = \tanh(\mathbf{W}_{\text{in}}[\mathbf{l}(n); b_{\text{in}}; \mathcal{K}(n)] + \mathbf{W} \mathbf{r}(n-1)), \quad (31)$$

$$\hat{\mathbf{y}}(n) = \mathbf{W}_{\text{out}}[\mathbf{r}(n); b_{\text{out}}; \mathcal{K}(n)], \quad (32)$$

where the dependence on n is implicit via the input, $\mathbf{l}(n)$

$$\mathcal{K}(n) = \mathcal{K}(\mathbf{l}(n)). \quad (33)$$

Although the hESN can perform better than a conventional ESN of equal size, as shown in Section VI A, it can result in an unstable behavior. In prediction mode, the feedback of the output of \mathcal{K} , via \mathbf{W}_{out} , into its own input can create

a self-sustaining amplification that diverges to infinity (see Appendix A). In such cases, validation and test errors are undefined, violating regularity assumptions (e.g. continuity in the hyperparameter space), which are essential to many optimization algorithms. We propose ways of overcoming this issue. The first is error saturation. If the prediction error becomes greater than a threshold, the error is set to the threshold. The second is saturation, where a saturation function is applied to the output of the physical model itself, e.g. $\mathcal{K} \rightarrow \tanh(\mathcal{K})$, with the \tanh taken entry-wise (as in the conventional reservoir update equation). This can be seen as effectively increasing the reservoir by a number of units equal to the dimension of \mathcal{K} , where each of these units is connected to one entry of \mathcal{K} only and without repetition. The drawback is that, due to the saturation, the sensitivity to changes in the output of \mathcal{K} is reduced, which can impact the performance. The third is to eliminate the connection between \mathcal{K} and \mathbf{W}_{out} , with the output of \mathcal{K} feeding the reservoir only, effectively preventing unbounded growth. We tested the three suggested methods (result not shown). We found that the first option performed best for the case under investigation, which is why it is adopted in the remainder of the paper.

B. Hyperparameter selection

The traditional technique for selecting hyperparameters is manual selection, which is dependent on prior (human) knowledge and experience. However, that does not suit a non-intrusive approach, which is central to the objective of this work, as explained in Section VI B. The simplest non-intrusive technique is grid search, but it can carry high computational cost [71, 73, 74]. Furthermore, the discretization of the hyperparameter space is a delicate matter because, if it is too coarse, the optimum can be missed; whereas, if it is too fine, the computational cost becomes prohibitive. Bayesian optimization with Gaussian Process Regression has been documented to achieve good performance in hyperparameter tuning of echo state networks [71]. For example, Reinier Maat *et al.* [74] found that this technique systematically achieves similar, or lower, values of test error compared to grid search, but with fewer evaluations. In an in-depth examination of training techniques [71] (e.g. single-shot, cross validation, etc.), it was found that grid search and Bayesian optimization have similar values of validation error, with Bayesian optimization being more robust and efficient. In this paper, the hyperparameters are selected by minimizing the validation mean square error (validation MSE), using Bayesian optimization with Gaussian Process Regression (GPR). We use the implementation in the `scikit-optimize` library. The GPR uses a 3/2 Matérn Kernel [70] (see Section IV A) and the acquisition function is the GP-Hedge (see Section IV B). The initial seed points are generated using a Latin Hypercube Sampling method. To make it more amenable to optimisation, we smooth the cost functional with two modifications. First, because the values of the validation MSE cover multiple orders of magnitude (e.g. 10^{-6} to 10^3), we minimize the logarithm of the MSE. Second, we cap the error when it is larger than the threshold of 10^3 , as explained in Section V A. The saturation smooths the MSE at these points.

The minimization runs until the MSE is below a target threshold of 3×10^{-2} , which was chosen by trial and error; or the maximum number of calls, 20, has been reached. This value for the maximum number of calls is sufficiently large for the hyperparameter space to be explored, but not too large for the computation to become exceedingly expensive. We optimally tune ρ , σ_{in} , which are two hyperparameters that markedly affect the training [71]. The hyperparameter space is log-uniform, i.e. the optimization tunes the exponents of the hyperparameters,

$$\log_{10}(\sigma_{\text{in}}) \in [-2, 2], \quad (34)$$

$$\log_{10}(\rho) \in [-3, 0]. \quad (35)$$

A log-uniform allows a more efficient exploration of different scales than a linear space. In particular, ρ is related to the time scale of the dynamics, which can be of different orders of magnitude for different systems or attractors. In the optimization of Section VI B, because the attractor varies, we also include the Tikhonov factor, γ , as a hyperparameter with $\log_{10}(\gamma) \in [-11, -4]$. The process of hyperparameter selection via Bayesian optimization is schematized in Figure 3.

C. Data normalisation

Data normalization is crucial to obtaining good performance [53]. Because different components of the data vector can have vastly different ranges, a single input scaling factor, σ_{in} , can be insufficient. If the same scaling is applied to variables of different orders of magnitude, the \tanh might “ignore” one of the variables because of the saturation of values away from 0. In that case, the information from that variable would be lost.

While various normalizations exist, here we choose the “min-max” normalization, which divides the data variable by the difference between its maximum and minimum in the time period. For an unnormalized variable, \check{l} , whose

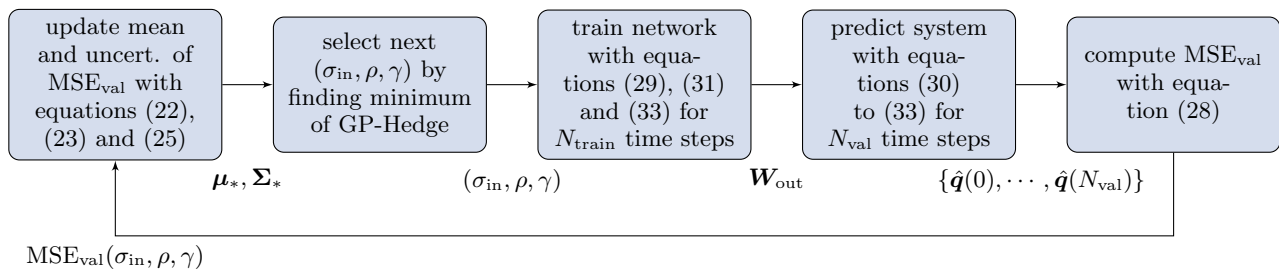


FIG. 3: Hyperparameter selection with Bayesian optimization.

time series is $\{\check{l}(0), \check{l}(1), \dots\}$, the normalised variable, l , is given by

$$l(n) = \frac{\check{l}(j)}{\max_j(\check{l}(j)) - \min_j(\check{l}(j))}, \quad n = 0, 1, \dots \quad (36)$$

This normalisation forcibly makes $l \in [-1, 1]$, which means that all the variables have the same range. With all the (normalized) variables in the same range, the use of the single scaling factor σ_{in} is justified.

VI. RESULTS

The training, validation and test lengths are $N_{\text{train}} = 5000$, $N_{\text{validation}} = 2000$ and $N_{\text{test}} = 10000$, respectively. For the chaotic attractor of Section VIA 1, this corresponds to approximately 6, 2.4 and 12 Lyapunov times⁵. To initialize the network, the reservoir state is initialized to 0 and the first 100 iterations are discarded. This ensures that the reservoir of network is not affected by the initialization. The physical model of the hESN is the same dynamical system that generates the data (equations (11) and (12)), but with one Galerkin mode only instead of 10 (i.e. $N_g = 1$). This mimics a situation in which data is available from an experiment for which a simple physical model exists. Alternatively, data can come from a high-fidelity simulation, while \mathcal{K} is a reduced-order model obtained from first principles and approximations.

The reservoir is composed of 400 and 100 nodes in the conventional and hybrid architectures, respectively. Although the reservoir size is usually chosen by heuristics, such as choosing the largest one can afford [53], or by human experience, here, for completeness, we show that these values are optimal for their respective architectures. To analyse the quality of a prediction, we use the Kullback-Leibler divergence [75]

$$D_{\text{KL}}(P||Q) = \int p(\mathbf{x}) \log \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) dV, \quad (37)$$

where P and Q are continuous distributions and p and q are the respective probability density functions. The integral is taken in the phase space of the system. In the Kullback-Leibler divergence, P refers to a “truth”, against which a “model” Q is being compared, which is suitable for the present situation, in which the truth is the data generated by the ODEs and the models are the echo state networks. If Q perfectly matches P , i.e. $p(\mathbf{x}) = q(\mathbf{x}) \forall \mathbf{x}$, then $D_{\text{KL}} = 0$, indicating that the larger D_{KL} , the worse the match. The numerical calculation of equation (37) is performed with the empirical distributions, i.e. via the histograms of P and Q .

Figure 4 shows the variation of D_{KL} with respect to the reservoir size for both the conventional and hybrid echo state network architectures. For each reservoir size, an ensemble of 10 network realizations is run, which allows the estimation of a distribution with its mean and spread. This is shown for the Galerkin modes 1, 2, 3 and 8, with the first three corresponding to the three most energetic modes and the last being representative of higher order modes. The hESN performs better on average than the ESN. In fact, there are only a few realizations of the ESN that outperform realizations of the hESN. This is not unexpected and is further explored in the next section. Furthermore, compared to the hESN, the ESN exhibits much larger variability within realizations of the same size. Analyzing separately, the hESN exhibits low variability in network realizations. The results also indicate that reservoir size has little, and possibly detrimental, impact on performance, with the optimal size being 100. This is due to a combination of: i) the

⁵ The leading Lyapunov exponent is approximately 0.12 [59], which means that the Lyapunov time is approximately 0.12^{-1} .

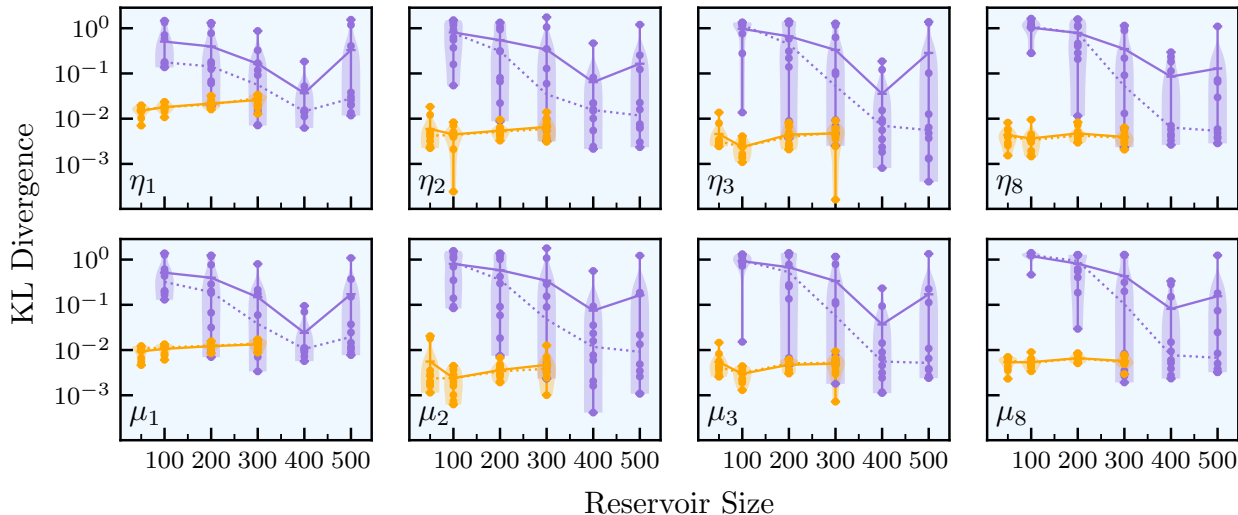


FIG. 4: Kullback-Leibler divergence, D_{KL} , versus reservoir size. The first and second rows correspond to the velocity and pressure modes, η_j and μ_j . The shaded regions correspond to the distributions of D_{KL} arising from the reservoir realizations. Each dot is a reservoir realization. The solid and dashed lines correspond to the mean and median.

physical model, which offers estimates of a good quality leaving reduced work to the network; ii) too many reservoir nodes, thus training parameters, for the small amount of data used in the training. On the one hand, at very low numbers of nodes, the network will have high error because it does not have a sufficient number of parameters to train. On the other hand, at very large number of nodes, there are too many parameters and overfitting becomes a problem. Therefore, a U-shaped curve can be expected. This shape is barely visible for hESN because there is only one point, 50, to the left of the optimum. Notwithstanding, this effect is more visible in the ESN curves, where there is an improvement as the reservoir size increases. Although both mean and median decrease up to 400, the mean increases from 400 to 500, while the median flattens or slightly decreases. This is explained by the large variability of the reservoirs of size 500. Therefore, given the similar median and comparable performances between ESN realizations of sizes 400 and 500, we select 400 nodes to keep the computational cost minimal. The realizations chosen for the following section are those closest to the median of selected sizes.

A. Short- and long-time predictions

In this section, we compare the predictive capabilities of both the conventional (ESN) and hybrid echo state networks (hESN). We fix the physical parameters $\beta = 7.0$ and $\tau = 0.2$, which correspond to a chaotic solution [59]. The physical system (equations (11) and (12)) will be referred to as the “Truth”. Information about each network, including the optimal hyperparameters, is given in Table I.

On the one hand, in short-time prediction, the objective is to time-accurately reproduce the dynamics of the system, i.e. starting from some initial condition, the objective is for the difference between the prediction and the true signals to be minimal for the largest possible time. This task is covered in Section VIA 1. On the other hand, in long-time prediction, the objective is to accurately reproduce the ergodic properties (statistics) of the system, i.e. the objective is for the difference between the true attractor (the stationary measure) and the attractor of the echo state networks to be minimal. Good performance in either task does not necessarily imply good performance in the other, as can be seen in Section VIA 2 (good long-, but poor short-time performance with the conventional ESN) and Appendix B (good short-, but poor long-time performance).

1. Short-time prediction

Figure 5 shows the time series of the first three (velocity) Galerkin modes, for the truth (data from ODE integration) and closed-loop predictions of the ESN and hESN. These modes are significantly more energetic than those of higher

order because the flame is located at $x_f = 0.2$, which markedly excites the first modes [28]. All three modes oscillate in a non-periodic manner, with the peak frequency increasing with the mode number. Figure 5 also shows the time series of the two cost functionals, acoustic energy and Rayleigh index. The Rayleigh index oscillates substantially more than the acoustic energy because the time derivative of the acoustic energy is equal to the sum of the Rayleigh index and the dissipation from damping [17]. Time-wise, the hESN is able to time-accurately predict these modes for the whole time span shown, whereas the ESN deviates from the truth signal at $t \approx 10$.

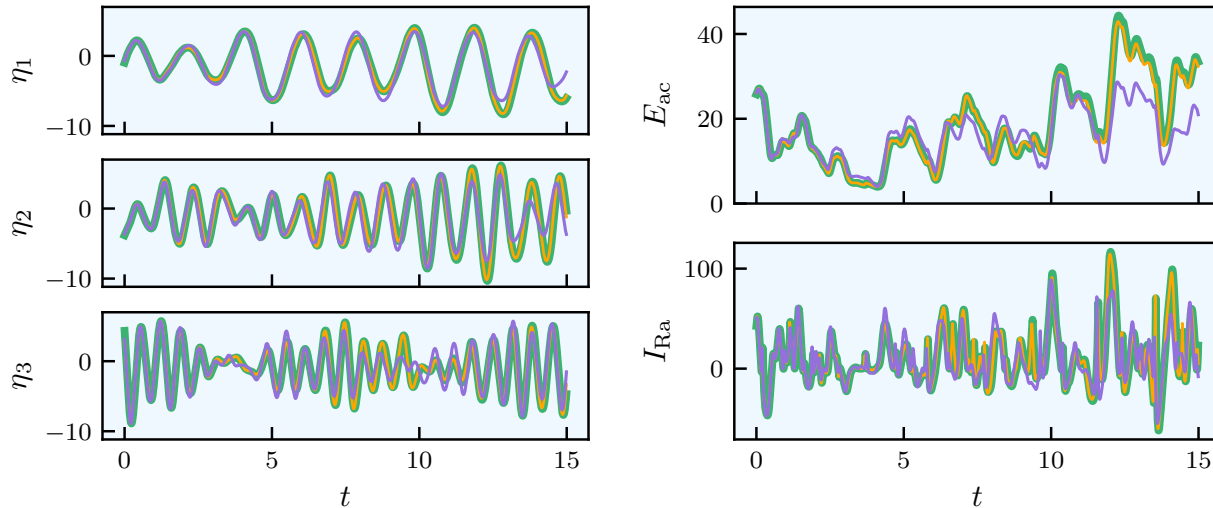


FIG. 5: Short-time prediction. Time series of the first three (velocity) Galerkin modes, η_1 , η_2 and η_3 ; acoustic energy, E_{ac} ; Rayleigh index, I_{Ra} .

As a global metric, we compute the normalized root mean square error,

$$\text{NRMSE}(n) = \left(\frac{\|\hat{\mathbf{y}}(n) - \mathbf{y}(n)\|^2}{N^{-1} \sum_{j=1}^N \|\mathbf{y}(j)\|^2} \right)^{1/2}, \quad (38)$$

which is shown in Figure 6. The hESN performs better than the ESN. Given a threshold, the predictability horizon is defined as the time at which the first crossing of the threshold occurs [54, 55]. With a threshold of 0.5, the hESN achieves a predictability horizon of 42.1 time units (5.1 Lyapunov times, compared to 7.5 time units (0.9 Lyapunov times) of the ESN. The NRMSE is, however, sensitive to normalization, and cannot discriminate between a time-inaccurate prediction that has similar dynamics and another that has completely different dynamics. An alternative visualization of the short-time behavior is given in Figure 7, which shows the time evolution of the acoustic pressure in an $x-t$ diagram. The truth panel shows that the flow is unsteady, featuring non-periodic acoustic waves propagating inside the domain. Although non-periodic, there appears to be a dominant frequency, showing up as roughly five waves in each 10 time units long window, which corresponds to an approximate period of 2 time units. This is related to the first acoustic eigenfunction. The third panel, corresponding to the ESN error, shows that the predictability horizon of the ESN is relatively short, which corroborates the findings of the NRMSE of Figure 6. However, the dynamics of the ESN are qualitative similar to those of the truth. This can be important, because, as shown in [59], an ESN can display inaccurate short-time prediction, but can have accurate long-time dynamics. The findings from the pressure map agree with those of the NRMSE not only for the ESN, but also for the hESN. The pressure plot of the hESN indicates that it only starts to exhibit significant error at $t \approx 45$, which is similar to the predictability horizon of 42.1 time units found with the NRMSE. This result further shows that hESN is capable of time-accurate

	N_r	\mathbf{b}_{in}	\mathbf{b}_{out}	sp	ρ	σ_{in}
ESN	400	1	1	99%	0.00176	12.2
hESN	100	0	0	97%	0.25760	0.02825

TABLE I: Characteristics of the ESN and hESN. sp is the sparseness (i.e. fraction of 0 entries) of \mathbf{W} . The Tikhonov factor is $\gamma = 10^{-9}$.

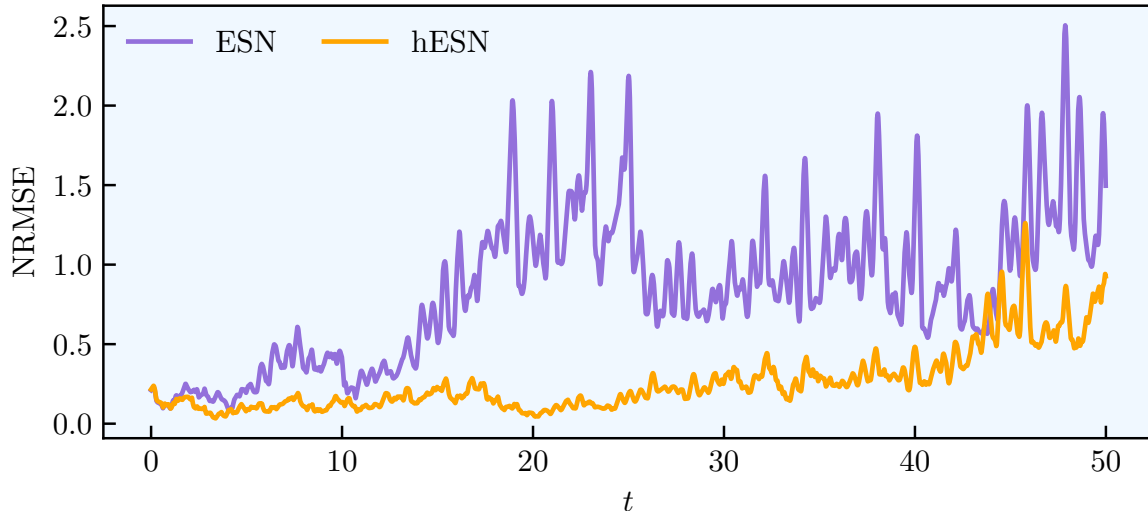


FIG. 6: Short-time prediction. Normalized Root Mean Square Error of the acoustic modes

prediction. We remark that such conclusions do not apply in general to the classes of conventional and hybrid echo state networks (i.e. an ESN need not perform worse than an hESN). Increasing the reservoir size of the ESN could yield satisfactory short-time prediction as well. However, the inclusion of model knowledge significantly improves the performance of an ESN for the same reservoir size.

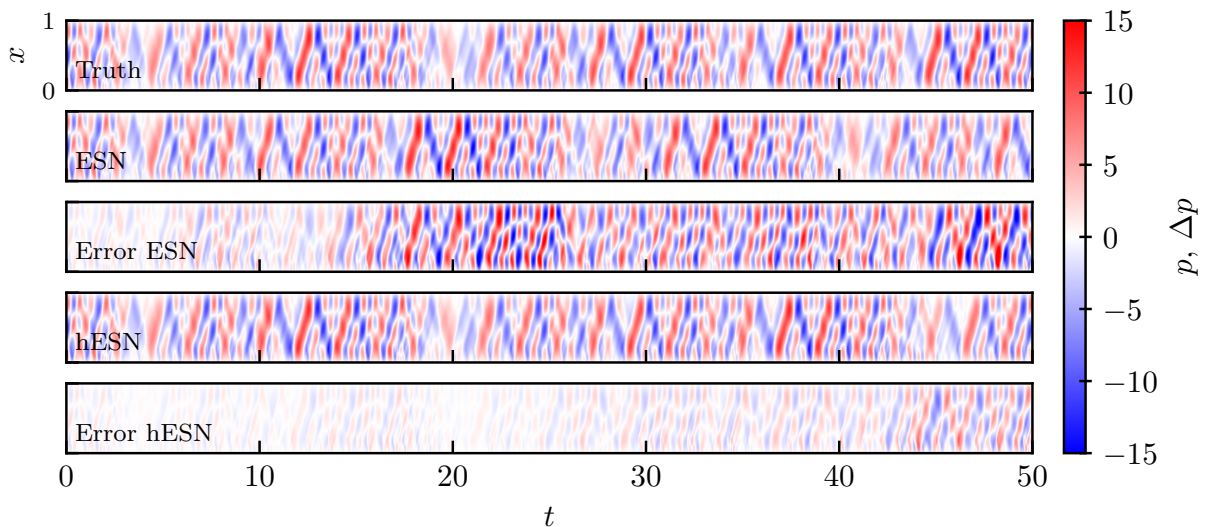


FIG. 7: Short-time prediction. Acoustic pressure. The error is defined as $\Delta p(t, x) = \hat{p}(t, x) - p(t, x)$, where \hat{p} is the prediction and p is the true pressure field.

2. Long-time prediction

In this section, we focus the ergodic (i.e. long-time) prediction, which is key to this paper. As previously mentioned, inaccurate short-time (i.e. time-accurate) prediction does not necessarily imply inaccurate long-time prediction [59]. (Conversely, as shown in appendix B, accurate short-time prediction does not necessarily imply accurate long-time prediction either.) We analyse the predictive capability of the ESN and hESN in the long-time with metrics that are

naturally defined in the statistically stationary regime.

First, we compute the frequency spectra (Figure 8). The spectra are continuous, which is consistent with the underlying signal being chaotic. The spectra match satisfactorily, with the largest error appearing at higher frequencies $f \gtrsim 2$, which have a negligible importance because the power of the signal is concentrated at the lower frequencies. For lower frequencies (inset of Figure 8), there is a favourable agreement between the two types of networks and the truth. In particular, both ESN and hESN match the dominant acoustic frequency and the peak of the true signal, which are close to the first natural acoustic eigenmode, $f = 0.5$. This is consistent with the wave number in the $x-t$ plot of Figure 7. Analysis of the spectra of the other state variables suggest similar conclusions (result not shown). We can conclude that echo state networks, both conventional and hybrid, reproduce the physical system satisfactorily in the time domain.

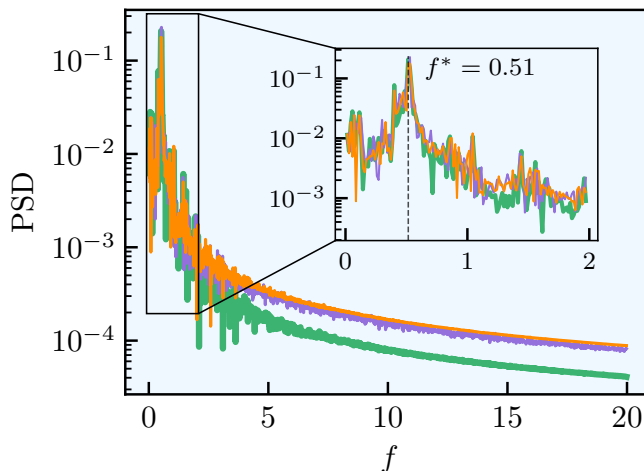


FIG. 8: Long-time prediction. Frequency spectrum of the acoustic velocity mode, $\eta_1(t)$, for Truth, ESN and hESN.

Second, we compute the probability density functions (PDFs) of the chaotic attractor because the long-time behavior of a system and its statistics depend on the invariant measure of its attractor. We compute two-dimensional joint PDFs of the Galerkin modes, i.e. (η_1, μ_1) , (η_2, μ_2) , etc.; and the one-dimensional PDFs of the individual state variables. These are performed for modes 1, 2, 3 and 8 (Figure 9), the first three being the most energetic and the last being representative of the higher modes. The PDFs are obtained via Kernel Density Estimation [76, 77]. Both networks perform well, with their PDFs matching those of the truth relatively well. However, although the ESN shows a good agreement with the truth, it is less accurate than the hESN. The difference in performance is more evident close to the modes of η_1 and η_2 , where the ESN over- and underpredicts the values of the peaks. This indicates that the invariant measure of the attractor of the dynamical system is well captured by both the ESN and hESN, with the latter being more accurate. Therefore, both ESN and hESN predict the long-time statistics of the physical system, with training from relatively small data.

B. Design optimization

For the physical optimizer, we define the cost functional, $\langle E_{ac} \rangle$; the optimization space, $\beta \in [7.5, 10.0]$, $\tau \in [0.1, 0.3]$; the acquisition function, LCB with $\kappa = 1.960$; the covariance function, RBF; the number of initial seed points, 4, which should be sufficient to properly initialize the GP; and the maximum number of evaluations (12, including the seed points), which we find to be a good compromise between efficacy and efficiency⁶. Similarly, for the hyperparameter optimizer, we define the cost functional, validation MSE; optimisation space, $\log_{10}(\sigma_{in}) \in [-2, 2]$, $\log_{10}(\rho) \in [-3, 0]$, $\log_{10}(\gamma) \in [-11, -4]$; the acquisition function, GP-Hedge; the covariance function, Matérn 3/2; the number of initial seed points, 5; and the maximum number of evaluations, 25, which include the seed points. Here, we include the Tikhonov factor as a hyperparameter. The reason is that the physical optimization evaluates attractors that can vary widely. In this case, adjusting the Tikhonov factor becomes important because it controls the relative importance

⁶ Efficacy here relates to whether the goal is achieved or not, whereas efficiency relates to how costly achieving the goal was.

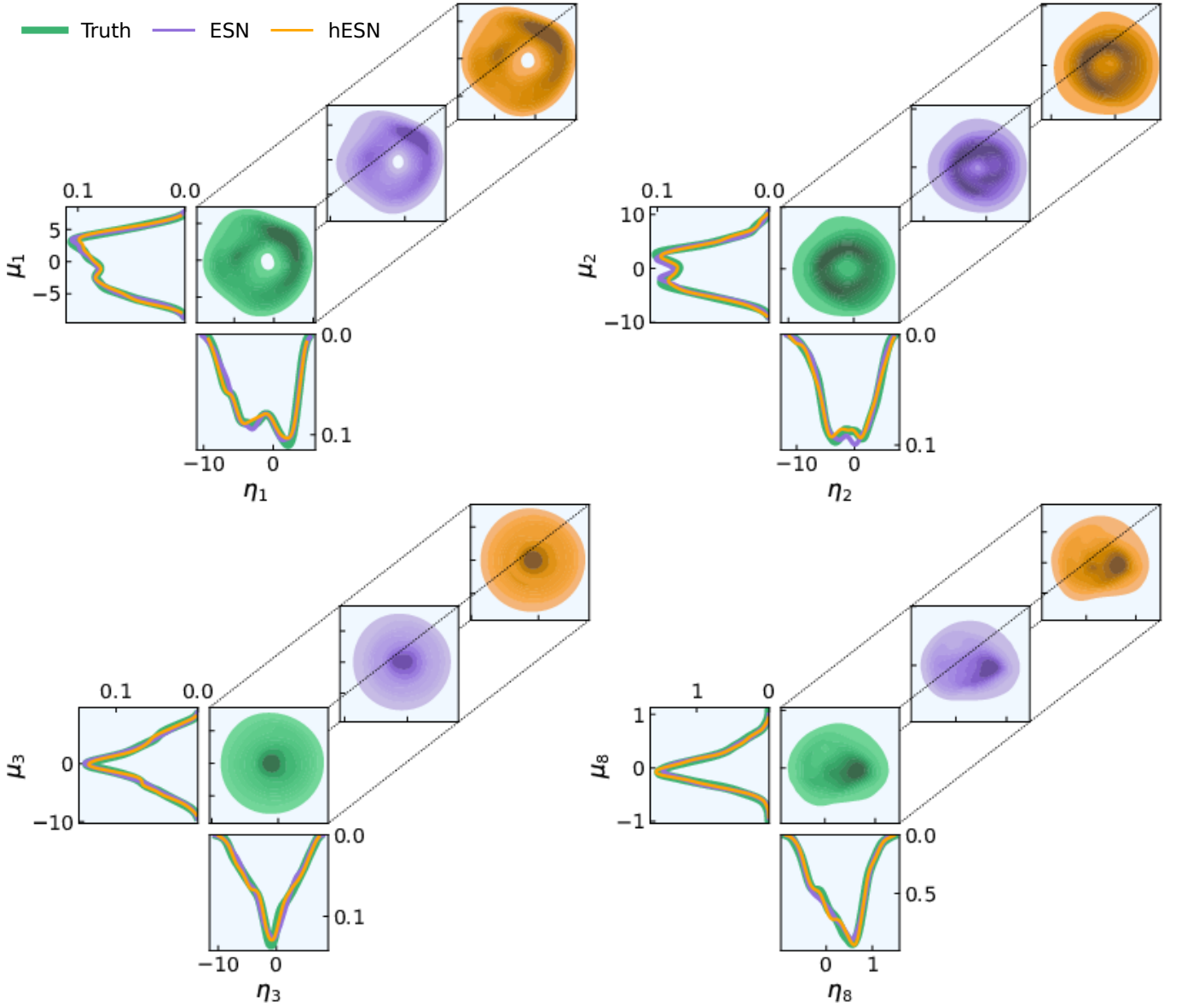


FIG. 9: Long-time prediction. Probability Density Functions of the acoustic modes 1, 2, 3 and 8. The two-dimensional joint PDFs correspond to the velocity and pressure variables of the same acoustic mode pair, (η_j, μ_j) . The one-dimensional PDFs are the marginalizations of the two-dimensional joint PDFs.

of the MSE and the norm of \mathbf{W}_{out} in the training problem, two terms that can vary substantially depending on the attractor. This is in contrast with Section VIA, where only one attractor was learned and predicted. To save on computational cost, the hyperparameter optimization stops when the error is below the threshold of 3×10^{-2} (Section VB). The chain then runs on its own.

First, the physical optimizer randomly generates seed points. For each of these points, $\mathbf{s}_{\text{phys}} = (\beta, \tau)$, data, $\{\mathbf{q}(0), \mathbf{q}(1), \dots\}^7$, is generated by integrating equations (8), (11) and (12). Then, the hyperparameter optimizer selects the optimal hyperparameters, $\mathbf{s}_{\text{hyper}}^+ = (\sigma_{\text{in}}^+, \rho^+, \gamma^+)$, using Bayesian optimization. With the optimal hyperparameters (and the corresponding optimal \mathbf{W}_{out}), the network is run in closed-loop, generating a long time series, $\{\hat{\mathbf{y}}(0), \hat{\mathbf{y}}(1), \dots\}$, from which the time-averaged acoustic energy, $\langle E_{\text{ac}} \rangle$, is computed and returned to the physical optimizer. After the seed points have been evaluated, the physical optimizer selects the next point by finding the

⁷ Whereas \mathbf{q} is defined in equation (1) as continuous in time, here, \mathbf{q} is the numerical solution, which is only defined at discrete times $0, 1, \dots$. Thus, slightly abusing notation, we write $\mathbf{q}(t = n \Delta t)$ as $\mathbf{q}(n)$, where n is a discrete time.

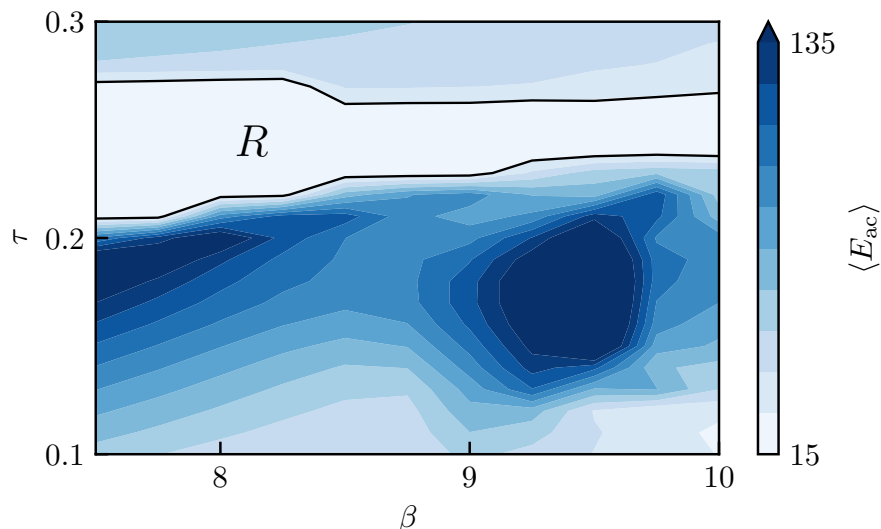


FIG. 10: Time-averaged acoustic energy, $\langle E_{ac} \rangle$, versus the flame parameters, β and τ , obtained with a brute-force grid search. Benchmark solution.

optimum of the acquisition function, $\mathbf{s}_{\text{phys}} = (\beta, \tau)$, which is then evaluated in the same manner as the seed points. The optimisation stops when 12 points (including seed points) have been evaluated.

For comparison, the true cost functional is physically shown in Figure 10. This chart is generated by integrating the ODEs on a grid of 11 values of β and 21 of τ . There is a large region of high acoustic energy, which can be divided into two sub-regions, each centered around a local maximum, one of which is on the boundary of the domain ($\beta = 7.5$ and $\tau \approx 0.18$). Above this, there exists a nearly horizontal strip spanning the whole range of β , marked R in Figure 10, which is where the optimum from the optimisation is likely to be found. Physically, as noted in Huhn and Magri [28], the time-averaged acoustic energy may be discontinuous at certain flame parameters because the attractor is structurally unstable. The global minimum found with the grid of Figure 10 is $\langle E_{ac} \rangle (\beta \approx 8.25, \tau \approx 0.27) = 15.04$.

Figure 11 shows the results of the physical optimisation. The three columns correspond to the mean of the GPR, standard deviation of the GPR, and the acquisition function LCB. The i -th row (starting at 0) corresponds to the state of the optimisation after $n_{\text{seed}} + i$ evaluations, where n_{seed} is the number of initial seed points (4 here) used to seed the optimization. It shows the previously evaluated points, with the most recent being encircled. The minimum of the acquisition function, and therefore the next point to be evaluated, is marked with a cross in the third column. Each row in Figure 11 corresponds to a row of Figure 12, which contains the time series of the acoustic energy, the phase plot μ_1 vs η_1 and the frequency spectrum of η_1 of the newly evaluated point. For the purpose of comparison, we include the true signals.

Initially (row $i = 0$), with 4 seed points, the GPR indicates a two-dimensional dependence on both β and τ , with clear regions of similar acoustic energy around each of the seed points, especially the two at the extremes of β . This is because these two points correspond to low and high values of acoustic energy. The combination of a larger distance in β than in τ between these two points, and the fact that the other two points, which have low and high values of τ , have similar $\langle E_{ac} \rangle$, leads the fit of the GPR to place a larger weight on β than τ . In terms of dynamics, the optimum of the seed points is a chaotic attractor, as shown in the first row of Figure 12.

The agreement between truth and hESN is favorable. Furthermore, the time series remains below the dashed line, which corresponds to the previous optimum, showing that this design is the optimum of the seed points. As expected, the uncertainty is lower in regions centred around evaluated points. The dependence on τ is substantially reduced after the evaluation of the first selected point ($i = 1$), a design that is close to the current optimum, not only in distance in the design space, but also in time-averaged acoustic energy (Figure 12). The perceived small dependence on τ and relatively strong dependence on β , in conjunction with low estimated uncertainty, makes the acquisition function discard approximately three quarters of the optimization domain, corresponding approximately to the upper three quarters of the range of β (Figure 11, $i = 1$, third column). With uncertainty low almost everywhere, its highest values are found close to the corners of the design space, where the distance from the sampled points is maximal. Moreover, given that the GPR indicates positive dependence of the cost functional on β and slightly negative dependence on τ , the minimum of the acquisition function is naturally found at the upper left corner, where β is minimal and τ maximal. This point ($i = 2$), in contrast with the estimate prior to its evaluation, is in a region of moderately-high acoustic

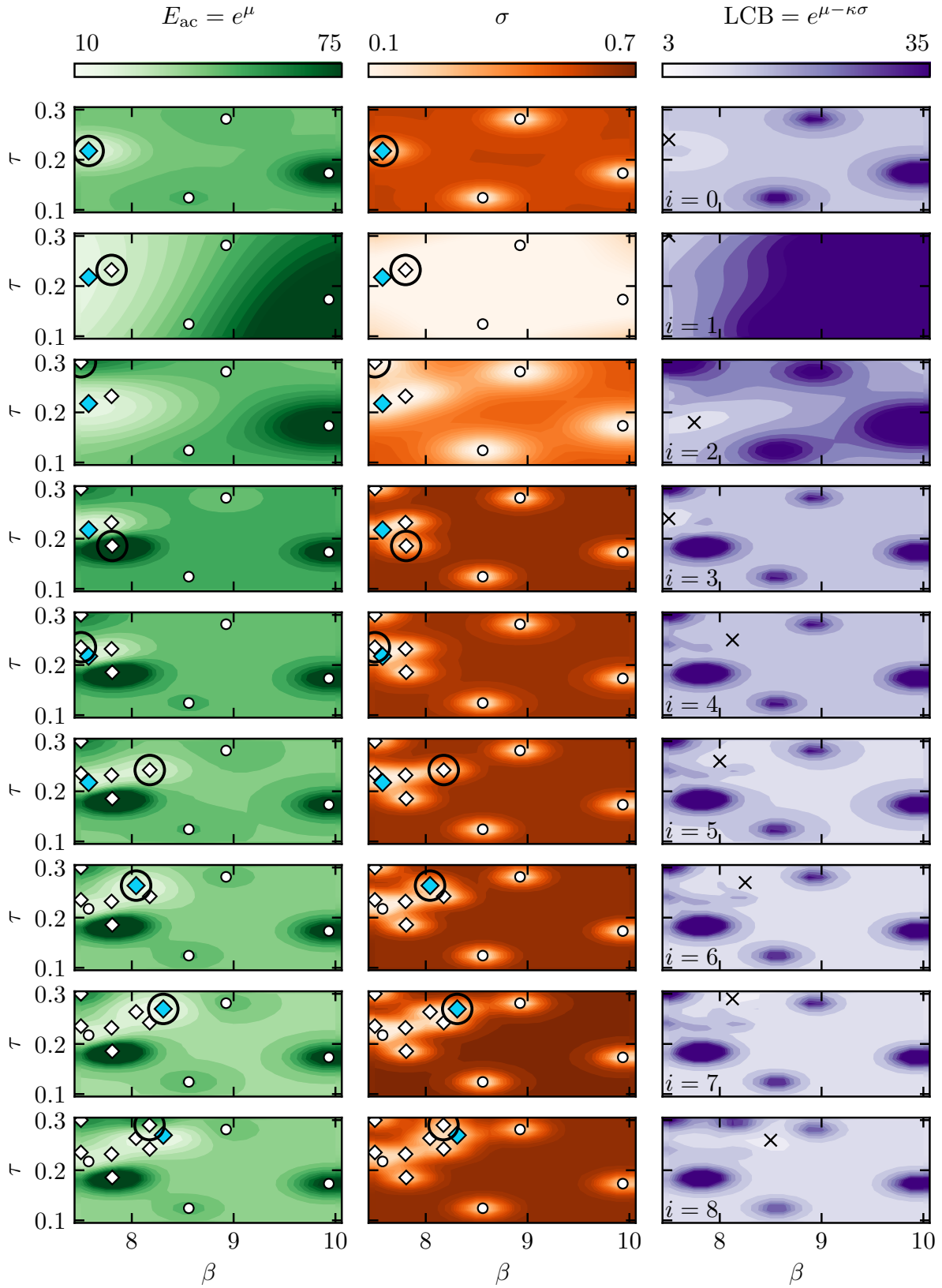


FIG. 11: Optimisation history. The first and second columns show the mean and standard deviation of the GPR. Seed points and previously evaluated points are marked with white circles and diamonds, respectively. The current optimum is blue and the last evaluated point is encircled. The third column shows the acquisition function, LCB, the minimum of which is the next point to be evaluated (cross).

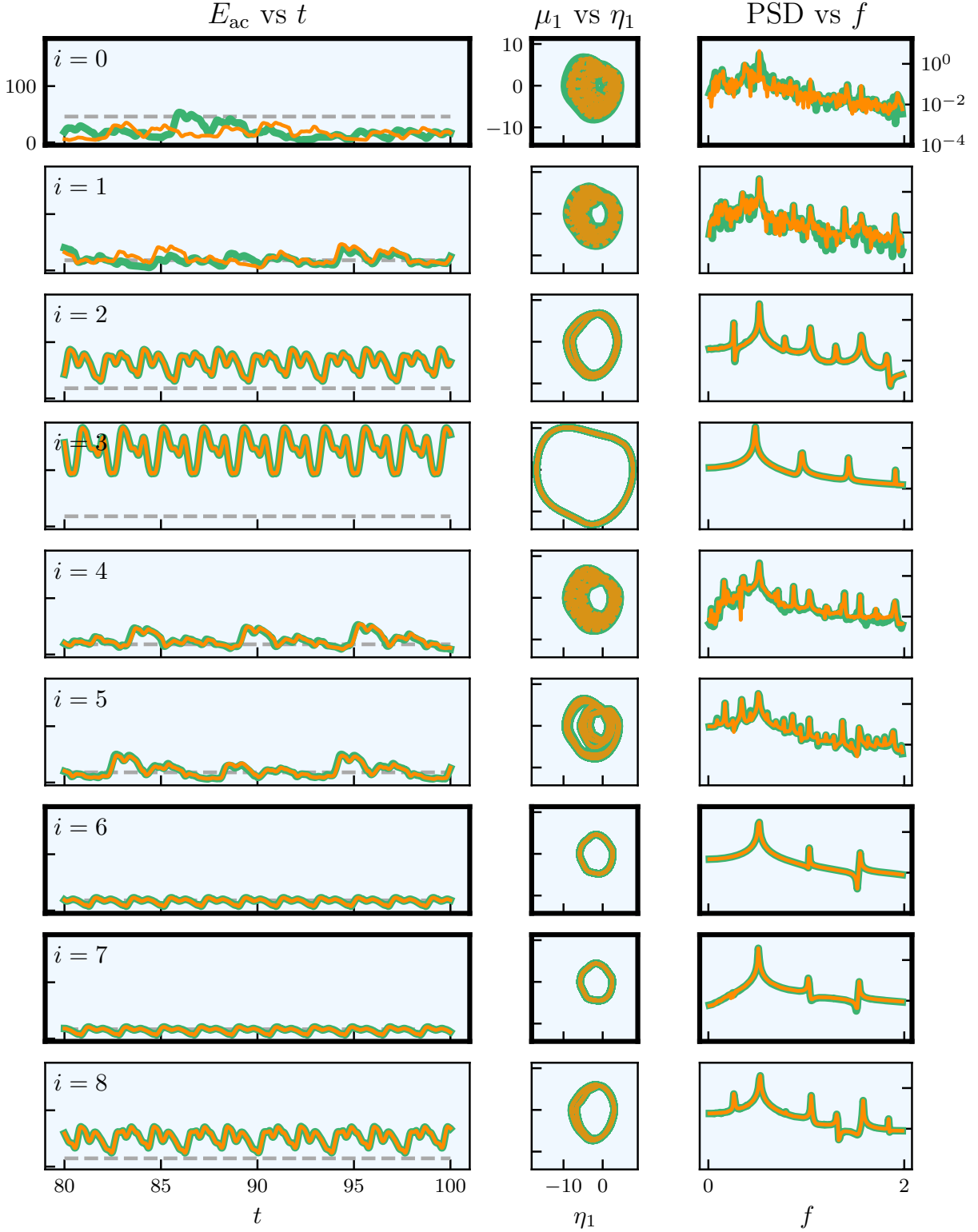


FIG. 12: Evaluated points during optimisation. The columns correspond to the last fifth of the time series of the acoustic energy; the phase plot of the first acoustic mode (μ_1 vs η_1); and the frequency spectrum of the first acoustic velocity mode, η_1 . Rows with thick spines correspond to a new optimum. The time-averaged acoustic energy of the current optimum is shown for reference as a horizontal dashed grey line.

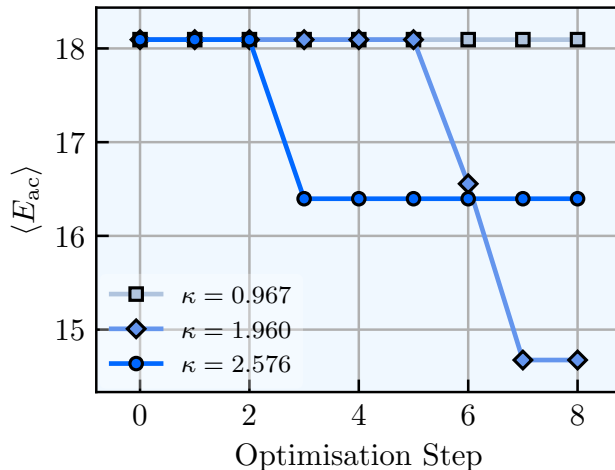


FIG. 13: Time-averaged acoustic energy, $\langle E_{ac} \rangle$, versus number of points evaluated, for three values κ : 0.967, 1.960 and 2.576; corresponding to 67, 95 and 99% confidence intervals.

energy (Figure 10). This can be verified in Figure 12, which shows that this combination of β and τ corresponds to a limit cycle whose instantaneous acoustic energy never drops below the time-averaged acoustic energy of the current optimal design. The new data updates the GPR, which now exhibits moderate uncertainty throughout the domain, except relatively close to previously evaluated points ($i = 2$ row, second column). While the dependence on β remains, the dependence on τ increases and is no longer monotonic. With moderately-high uncertainty away from the points, and with low mean in a small region only around the current optimum, the acquisition function selects a point where the two regions (high uncertainty, low mean) “meet”. The new design point, however, has very high acoustic energy (row $i = 3$ of Figure 12), as it belongs to the area surrounding the leftmost of the two peaks of high acoustic energy (Figure 10). This newly acquired information strongly contrasts with the prior estimate of the GPR. Naturally, this is because of the sharp variation of acoustic energy at the lower edge of the region R . A small variation of parameters from the current optimum resulted in high variation of the cost functional. As such, uncertainty is now high everywhere, except close to previously sampled points. At $i = 4$, the acquisition function chooses a point close to the current optimum, which is a clear evidence of exploitation. This design results in chaotic dynamics, similarly to the current optimum, but it does not produce lower time-averaged acoustic energy. Thus, it is not a new optimum. With three points (and the left boundary) enclosing the current optimum, there is little advantage in continuing exploiting. Switching to exploration, the new point ($i = 5$) is relatively far from the current optimum. Once again, it is at the intersection of low mean and high uncertainty, since that combination minimises the acquisition function. While the new design does not improve the optimum, it does provide crucial new information. Because its acoustic energy is low, the region of low mean expands with the updated GPR. This new expansion provides space to exploit. Thus, a new design ($i = 6$) relatively close to the previous is selected. This new design offers lower time-averaged acoustic energy than the current optimum, i.e. the optimization found a new optimum. The new optimal design represents an improvement of 8.4% with respect to the previous optimum. In the GPR, not only has a new optimum been found, but the region of low mean is now larger. Thus, at $i = 7$, the most recently selected design further expands the spread of points into higher β and higher τ . Similarly to the last design, a new optimum is found, this one offering a further 11.4% reduction of acoustic energy. Finally, the last design ($i = 8$), despite being close to the previous two optima, does not further improve the cost functional. It is likely that this design is above the upper boundary of the region R in Figure 10. Had the optimisation continue to run, it is possible the optimum could be slightly improved. However, the maximum number of evaluations was reached. Furthermore, the optimum of the optimization, found with 12 design evaluations (4 seed points plus 8 selected points), $\langle E_{ac} \rangle$ ($\beta \approx 8.31, \tau \approx 0.27$) = 14.68, is slightly better than that found with a brute-force grid search (Figure 10), $\langle E_{ac} \rangle$ ($\beta \approx 8.25, \tau \approx 0.27$) = 15.04, which needed 231 evaluations. A larger number of evaluations would likely have been a relatively poor trade-off between design improvement (i.e. decrease in cost functional) and computational cost.

Figure 13 shows the convergence of the optimisation procedure, i.e. the current optimum versus the number of points evaluated for three values of κ .

The largest value of κ , 2.576, favours exploration the most. This results in quickly, in the second optimisation step, finding a design that improves on the best seed point. However, because of its tendency to explore, it does not try to exploit and locally improve its current optimum as much. Hence why there is a large spread of points for

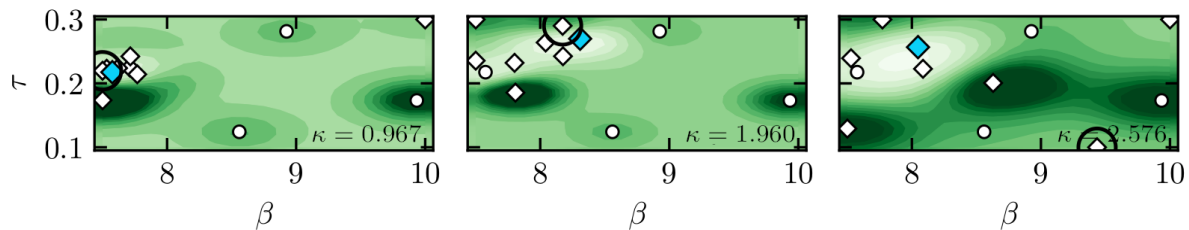


FIG. 14: Final state of optimisation with three values of κ : 0.967, 1.960 and 2.576; corresponding to 67, 95 and 99% confidence intervals. This shows the effect of κ in the balance between exploration vs exploitation.

$\kappa = 2.576$ in Figure 14, which shows the last state of the optimisation for the same three values of κ of Figure 13. In contrast, $\kappa = 0.967$, the lowest of the three, will seek to mostly exploit. The various designs concentrated in a small region are evidence of this. Unsurprisingly, this does not produce a better design than the optimal seed point. Finally, $\kappa = 1.960$, used in the optimisation of Figures 11 and 12, navigates between these two lines, exploration and exploitation, unsuccessfully trying to exploit initially, finding a new optimum with exploration and subsequently improving the recent optimum by exploiting its surrounding region. In conclusion, the larger κ is, the larger the spread of points, which shows the influence that this parameter has on the balance between exploration and exploitation. In this optimization problem, the initially chosen value of $\kappa = 1.960$ seems to offer the best performance among the three.

VII. CONCLUSIONS

Gradient-based design optimization of chaotic acoustics is notably challenging for a threefold reason. First, first-order perturbations grow exponentially in time, which makes the computation of the gradients with respect to the design parameters ill-posed. Second, the statistics of the solution may have a slow convergence, which makes the time integration of the equations computationally expensive. Third, chaotic acoustic systems may have discontinuous variations of the time-averaged energy [28], which means that the gradient may not exist for all design parameters. In this paper, we develop an optimization method to find the design parameters that minimize time-averaged acoustic cost functionals. The method is gradient-free, with Bayesian sampling; model-informed, with a reduced-order acoustic model; and data-driven, with reservoir computing.

First, we analyse the predictive capabilities of reservoir computing based on echo state networks. Both fully data-driven and model-informed architectures are considered. In the short-time prediction, model-informed networks can time-accurately predict the chaotic pressure oscillations beyond the predictability time (the Lyapunov time). For the same reservoir size, informing the training with a cheap model extends the prediction from ~ 1 Lyapunov time to ~ 5 Lyapunov times. In the long-time prediction, we show that echo state networks accurately reproduce the statistics of chaotic acoustic attractors. The hyperparameters are automatically tuned by using Bayesian optimization, which provides a consistently good performance across different architectures, reservoir sizes and data. With accurate predictions at a lower computational cost, the long-time series are generated to obtain the time-averaged acoustic energy that is being optimized. Second, we couple echo state networks with a Bayesian technique based on Gaussian Processes to explore the design parameter space. The computational method is minimally intrusive because it requires only the initialization of the physical and hyperparameter optimizers; e.g., a factor to balance the exploration versus exploitation of the sampling; and the reservoir size. Third, we apply the computational method to the minimization of the time-averaged acoustic energy of chaotic oscillations. We focus on the acoustics that is excited by a heat source, which is relevant to thermoacoustic oscillations in propulsion and power generation. The design parameters that are changed during the optimization are the flame intensity and time delay. Nonetheless, the method can tackle other physical parameters. Starting from five random designs with energetic chaotic oscillations, we find an optimal set of parameters in 8 iterations. This optimum is practically equal to the optimum found by brute-force grid search, which needs 231 evaluations. The thermoacoustic system shows a variety of solutions and bifurcations during the optimization update (e.g., limit cycles, strange attractors), which are accurately learnt by the echo state networks. This is because the echo state network learns the physical temporal correlations of the acoustic modes through the sparse recurrent dynamics of the reservoir.

This work opens up new possibilities for the optimization of chaotic systems, in which the cost of generating data, for example from high-fidelity simulations and experiments, is high.

DECLARATION OF INTERESTS

The authors report no conflict of interest.

ACKNOWLEDGEMENTS

F. Huhn is supported by Fundação para a Ciência e Tecnologia under the Research Studentship No. SFRH/BD/134617/2017. L. Magri gratefully acknowledges financial support from the ERC Starting Grant PhyCo 949388, and TUM Institute for Advanced Study (German Excellence Initiative and the EU 7th Framework Programme n. 291763). The authors thank A. Racca for insightful discussions.

Appendix A: Divergence of hybrid echo state network

Unlike conventional echo state networks, whose output is bounded (though the bound can be very far from the attractor), due to the feedback from the output of \mathcal{K} to its own input via \mathbf{W}_{out} , hybrid echo state networks can diverge to infinity. This can be seen in Figure 15. However, it should be noted that this behavior is not necessarily a function of the physical parameters or hyperparameters only. A certain fixed combination of hyperparameters can result in both divergence and non-divergence depending on the physical parameters. Similarly, for fixed physical parameters, changing the hyperparameters can result in divergence or not. This is not an issue with the training method (ridge regression), but the complex combination of training and validation time series; realizations of \mathbf{W}_{in} and \mathbf{W} ; values of σ_{in} and ρ ; Tikhonov factor γ ; model \mathcal{K} and its numerical scheme; etc. Furthermore, \mathcal{K} can be stable (i.e. its solution is stable), but the hybrid echo state network using it can be unstable. In fact, that is the case of Figure 15, where, if \mathcal{K} evolved on its own, a limit cycle arises. It is the (linear) transformation due to the output weights \mathbf{W}_{out} that changes the output of \mathcal{K} , which is then fed back to \mathcal{K} itself, that can make the whole system unstable.

A very succinct example, where we omit the reservoir nodes for simplicity, is

$$\mathcal{K}(y) = (1 - \lambda)y, \quad (\text{A1})$$

where $0 < \lambda < 1$ is a physical parameter. If left to evolve on its own, i.e.

$$y(n+1) = \mathcal{K}(y(n)) = (1 - \lambda)y(n), \quad (\text{A2})$$

y will converge to 0. However, in the hybrid echo state network framework, we have instead

$$y(n+1) = \mathcal{K}(W_{\text{out}}y(n)) = W_{\text{out}}(1 - \lambda)y(n). \quad (\text{A3})$$

For y to converge to 0, we must have

$$W_{\text{out}} < \frac{1}{1 - \lambda}. \quad (\text{A4})$$

If W_{out} does not verify this condition, the system will diverge, despite \mathcal{K} being stable.

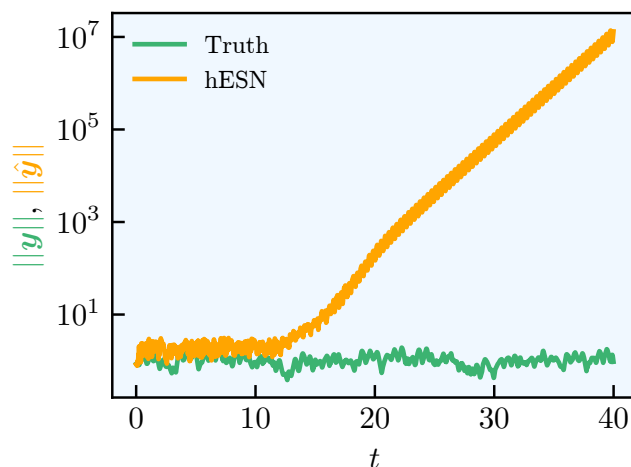


FIG. 15: Divergence of the prediction of a hybrid echo state network. This is obtained with the configuration detailed in Section VB and $\sigma_{\text{in}} = 4.23$ and $\rho = 0.9$. Other combinations of hyperparameters and physical parameters can result in similar behavior.

Appendix B: Accurate short-time, inaccurate long-time prediction

Here, we use a short example based on the Lorenz system [29] to show that accurate short-time prediction does not necessarily mean accurate long-time prediction. The Lorenz system is a three-dimensional system,

$$\frac{dx_L}{dt} = \sigma_L(y_L - x_L), \quad (\text{B1})$$

$$\frac{dy_L}{dt} = x_L(\rho_L - z_L) - y_L, \quad (\text{B2})$$

$$\frac{dz_L}{dt} = x_L y_L - \beta_L z_L, \quad (\text{B3})$$

where σ_L , ρ_L and β_L are parameters, often equal to 10, 28, $8/3$, which is a combination that produces chaotic motion. This system is numerically integrated with time step 0.01 to generate training, validation and test data. For this example, we use an echo state network with 100 nodes with no biases. The network is trained and validated on datasets of length 500, using Bayesian optimization. Figure 16 shows the NRMSE and (long-time) phase plot for the Lorenz system. The NRMSE remains below the threshold of 0.2 until $t \approx 2.87$, which corresponds to approximately 2.6 Lyapunov times (leading Lyapunov exponent of approximately 0.9). Thus, the ESN predicts the system relatively well in the short time. However, the phase plot shows a completely different behavior between prediction and data in the long-time. In this case in particular, the network has no biases (i.e. $b_{\text{in}} = b_{\text{out}} = 0$), in which case the reservoir evolves according to

$$\mathbf{r}(n) = \tanh(\widetilde{\mathbf{W}}\mathbf{r}(n-1)), \quad (\text{B4})$$

where $\widetilde{\mathbf{W}} = \mathbf{W} + \mathbf{W}_{\text{in}}\mathbf{W}_{\text{out}}$ [59]. This means that taking some reservoir state $\mathbf{r}(n-1)$, and flipping its sign, i.e. $\mathbf{r}'(n-1) = -\mathbf{r}(n-1)$, one gets $\mathbf{r}'(n) = -\mathbf{r}(n)$. Thus, either the ESN admits two attractors symmetric of each other, or admits one symmetric attractor, which is the case here. In conclusion, accurate short time prediction does not necessarily imply accurate long-time prediction.

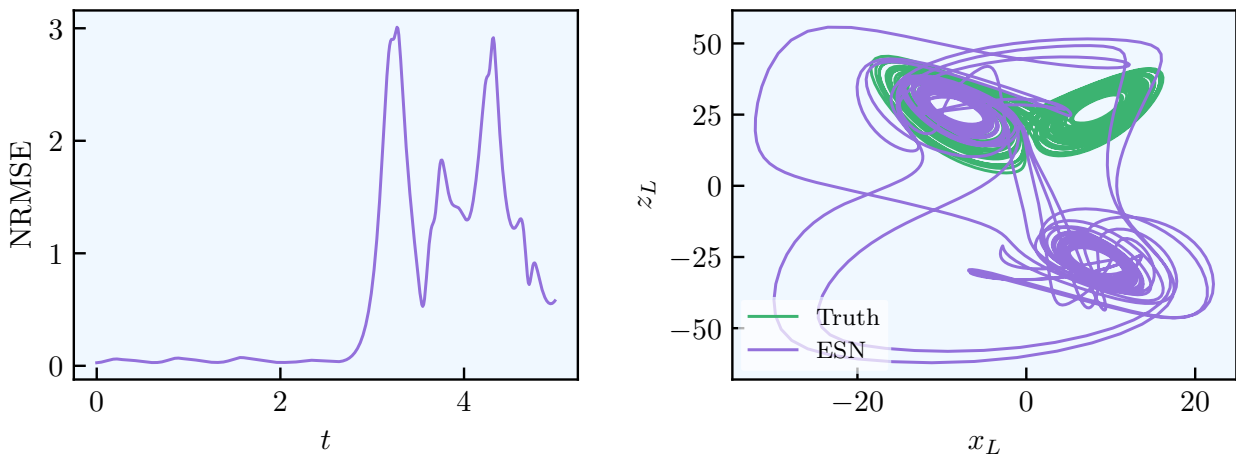


FIG. 16: Lorenz system. Time series of NRMSE and phase plot z_L vs x_L .

Appendix C: Computational cost of the method

The optimization framework in this paper (i.e. the “chain”) was demonstrated on a relatively low-dimensional system. In this particular case, the echo state network (and everything it involves) could have been foregone and Bayesian optimization applied directly to the result of the (longer run) ODEs. However, the application in this work is a proof of concept and not an example of computational gains. These are meant to be achieved in larger-scale systems, such as high-fidelity simulations. The cost of the method is

$$N_{\text{opt}}(N_{\text{train}}C_{\text{ODE}} + C_{\text{train}} + N_{\text{test}}C_{\text{ESN}}), \quad (\text{C1})$$

where N_{opt} is the number of optimization steps in the physical domain, N_{train} is the number of training timesteps, C_{ODE} is the cost per timestep of solving the ODEs, C_{train} is the cost of training the network (including the hyperparameters), N_{test} is the number of test timesteps and C_{ESN} is the cost of per timestep of the closed-loop ESN. On the other hand, applying Bayesian optimization directly to the ODEs instead would have a cost of

$$N_{\text{opt}}N_{\text{test}}C_{\text{ODE}}. \quad (\text{C2})$$

For there to be a cost benefit

$$N_{\text{opt}}(N_{\text{train}}C_{\text{ODE}} + C_{\text{train}} + N_{\text{test}}C_{\text{ESN}}) < N_{\text{opt}}N_{\text{test}}C_{\text{ODE}} \quad (\text{C3})$$

must be verified. Thus, each term must be analyzed. The most expensive operation is training the network, C_{train} , which scales with $O(N_x^3)$ due to the matrix inversion. If the number of nodes, N_x , is proportional to the dimension of the dynamical system, N_d , then this cost becomes $O(N_d^3)$. This operation only happens once (per hyperparameter combination), though. Once it is performed, the largest cost is the closed-loop ESN simulation at $O(N_x N_d) \sim O(N_d^2)$ per timestep. On the other hand, the cost of a timestep in a numerical simulation (ODE) can scale with $O(N_d^2)$ or $O(N_d^3)$, depending on the numerical scheme. This would put it at a similar scaling to the networks. Therefore, it would seem hard for equation (C3) to be verified. However, the goal is not to apply the technique directly to a high-fidelity simulation, but to a lower resolution of its results. While a certain number of grid points may be needed for an accurate simulation, after obtaining the results, only a subset of these points is required for the accurate computation of the cost functional. In other words, not all points are needed. Thus, the full state vector need not be computed/predicted. For example, if there is a downsample of 10 to 1 in every direction of a 3D high-fidelity simulation, there is a 1000-fold reduction in N_d , 1000^2 in output cost and 1000^3 in training cost; compared to predicting the full state vector of a high-fidelity simulation. In such a case, the training data generation via a high-fidelity simulation would be the most expensive step, as we assume in the paper, which would also mean that the RHS of equation (C3) would be much larger than the LHS. Additionally, as remarked before, this approach also suits an experimental framework where running the experiment for a sufficiently long time might be expensive.

-
- [1] L. Rayleigh, *Nature* **18**, 319 (1878).
 - [2] T. C. Lieuwen and V. Yang, *Combustion Instabilities in Gas Turbine Engines: Operational Experience, Fundamental Mechanisms, and Modeling* (American Institute of Aeronautics and Astronautics, Inc., 2005).
 - [3] F. E. C. Culick, *Unsteady motions in combustion chambers for propulsion systems* (RTO AGARDograph AG-AVT-039, North Atlantic Treaty Organization, 2006).
 - [4] A. P. Dowling and Y. Mahmoudi, *Proceedings of the Combustion Institute* **35**, 65 (2015).
 - [5] A. P. Dowling, *Journal of Fluid Mechanics* **346**, 271 (1997).
 - [6] M. P. Juniper and R. Sujith, *Annual Review of Fluid Mechanics* **50**, 661 (2018).
 - [7] L. Magri, *Applied Mechanics Reviews*, under review (2018).
 - [8] L. Magri and M. P. Juniper, *Journal of Fluid Mechanics* **719**, 183–202 (2013).
 - [9] L. Magri and M. P. Juniper, *Journal of Fluid Mechanics* **752**, 237–265 (2014).
 - [10] A. Orchini and M. P. Juniper, *Combustion and Flame* **165**, 97 (2015).
 - [11] G. A. Mensah and J. P. Moeck, *Journal of Engineering for Gas Turbines and Power* **139**, 061501 (2017).
 - [12] G. A. Mensah, L. Magri, A. Orchini, and J. P. Moeck, *Journal of Engineering for Gas Turbines and Power*, doi:10.1115/1.4041007 10.1115/1.4041007 (2018).
 - [13] J. G. Aguilar and M. P. Juniper, *Phys. Rev. Fluids* **5**, 083902 (2020).
 - [14] J. G. Aguilar and M. P. Juniper, in *Turbo Expo: Power for Land, Sea, and Air*, Vol. 51050 (American Society of Mechanical Engineers, 2018) p. V04AT04A054.
 - [15] F. Schaefer, L. Magri, and W. Polifke, in *Turbo Expo: Power for Land, Sea, and Air* (American Society of Mechanical Engineers, 2021).
 - [16] A. P. Dowling, *Journal of Fluid Mechanics* **394**, 51 (1999).
 - [17] F. Huhn and L. Magri, *Nonlinear Dynamics* **100**, 1641 (2020).
 - [18] P. Subramanian, S. Mariappan, R. I. Sujith, and P. Wahi, *International Journal of Spray and Combustion Dynamics* **2**, 325 (2011).
 - [19] L. Kabiraj, R. I. Sujith, and P. Wahi, *Journal of Engineering for Gas Turbines and Power* **134**, 31502 (2011).
 - [20] H. Gotoda, H. Nikimoto, T. Miyano, and S. Tachibana, *Chaos* **21**, 013124 (2011).
 - [21] H. Gotoda, T. Ikawa, K. Maki, and T. Miyano, *Chaos* **22**, 033106 (2012).
 - [22] L. Kabiraj, A. Saurabh, P. Wahi, and R. I. Sujith, *Chaos* **22**, 0 (2012).
 - [23] V. Nair, G. Thampi, and R. I. Sujith, *Journal of Fluid Mechanics* **756**, 470 (2014).
 - [24] V. Nair and R. I. Sujith, *Proceedings of the Combustion Institute*, in press **35**, 3193 (2015).
 - [25] K. Kashinath, I. C. Waugh, and M. P. Juniper, *Journal of Fluid Mechanics* **761**, 399 (2014).

- [26] I. C. Waugh, K. Kashinath, and M. P. Juniper, *Journal of Fluid Mechanics* **759**, 1 (2014).
- [27] A. Orchini, S. J. Illingworth, and M. P. Juniper, *Journal of Fluid Mechanics* **775**, 387 (2015).
- [28] F. Huhn and L. Magri, *Journal of Fluid Mechanics* **882**, A24 (2020).
- [29] E. N. Lorenz, *Journal of the Atmospheric Sciences* **20**, 130 (1963).
- [30] D. J. Lea, M. R. Allen, and T. W. Haine, *Tellus A: Dynamic Meteorology and Oceanography* **52**, 523 (2000).
- [31] D. J. Lea, T. W. N. Haine, M. R. Allen, and J. A. Hansen, *Quarterly Journal of the Royal Meteorological Society* **128**, 2587 (2002).
- [32] G. L. Eyink, T. W. Haine, and D. J. Lea, *Nonlinearity* **17**, 1867 (2004).
- [33] J. Thurn, *Quarterly Journal of the Royal Meteorological Society* **131**, 73 (2005).
- [34] P. J. Blonigan and Q. Wang, *Journal of Computational Physics* **270**, 660 (2014).
- [35] D. Lasagna, *SIAM Journal on Applied Dynamical Systems* **17**, 547 (2018).
- [36] C. E. Leith, *Journal of Atmospheric Sciences* **32**, 2022 (1975).
- [37] R. V. Abramov and A. J. Majda, *Nonlinearity* **20**, 2793 (2007).
- [38] R. V. Abramov and A. J. Majda, *J Nonlinear Sci* **18**, 303 (2008).
- [39] Q. Wang, *Journal of Computational Physics* **235**, 1 (2013).
- [40] Q. Wang, R. Hu, and P. Blonigan, *Journal of Computational Physics* **267**, 210 (2014).
- [41] A. Ni and Q. Wang, *Journal of Computational Physics* **347**, 56 (2017).
- [42] P. J. Blonigan and Q. Wang, *Journal of Computational Physics* **354**, 447 (2018).
- [43] A. Ni and C. Talnikar, *Journal of Computational Physics* **395**, 690 (2019).
- [44] N. Chandramoorthy and Q. Wang, arXiv preprint arXiv:2002.04117 (2020).
- [45] A. Ni, arXiv preprint arXiv:2009.00595 (2020).
- [46] P. J. Blonigan, *Journal of Computational Physics* **348**, 803 (2017).
- [47] A. Ni, *Journal of Fluid Mechanics* **863**, 644–669 (2019).
- [48] D. Ruelle, *Nonlinearity* **22**, 855 (2009).
- [49] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) <http://www.deeplearningbook.org>.
- [50] S. Hochreiter and J. Schmidhuber, *Neural Computation* **9**, 1735 (1997).
- [51] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation (2014), arXiv:1406.1078 [cs.CL].
- [52] H. Jaeger and H. Haas, *Science* **304**, 78 (2004).
- [53] M. Lukoševičius, *A practical guide to applying echo state networks* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012) pp. 659–686.
- [54] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **28**, 041101 (2018).
- [55] N. A. K. Doan, W. Polifke, and L. Magri, in *Computational Science – ICCS 2019* (Springer International Publishing, Cham, 2019) pp. 192–198.
- [56] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **27**, 041102 (2017), <https://doi.org/10.1063/1.4979665>.
- [57] N. A. K. Doan, W. Polifke, and L. Magri, in *Computational Science – ICCS 2020* (Springer International Publishing, Cham, 2020) pp. 117–123.
- [58] A. Racca and L. Magri, Automatic-differentiated physics-informed echo state network (api-esn) (2021), arXiv:2101.00002 [cs.LG].
- [59] F. Huhn and L. Magri, in *Computational Science – ICCS 2020* (Springer International Publishing, Cham, 2020) pp. 124–132.
- [60] A. G. Hart, J. L. Hook, and J. H. P. Dawes, Echo state networks trained by tikhonov least squares are $l_2(\mu)$ approximators of ergodic dynamical systems (2021), arXiv:2005.06967 [cs.LG].
- [61] M. P. Juniper, *Journal of Fluid Mechanics* **667**, 272 (2011).
- [62] L. V. King, *Proceedings of the Royal Society of ...* **214**, 373 (1914).
- [63] M. A. Heckl, *Journal of Sound and Vibration* **124**, 117 (1988).
- [64] W. Polifke, A. Poncet, C. O. Paschereit, and K. Döbbeling, *Journal of Sound and Vibration* **245**, 483 (2001).
- [65] A. Orchini, G. Rigas, and M. P. Juniper, *Journal of Fluid Mechanics* **805**, 523 (2016).
- [66] K. Balasubramanian and R. I. Sujith, *Journal of Fluid Mechanics* **594**, 29 (2008).
- [67] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, 2nd ed. (Pergamon Press, 1987).
- [68] L. N. Trefethen, *Spectral methods in MATLAB*, Vol. 10 (Siam, 2000).
- [69] C. A. Kennedy, M. H. Carpenter, and R. Lewis, *Applied Numerical Mathematics* **35**, 177 (2000).
- [70] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, Massachusetts, USA, 2006).
- [71] A. Racca and L. Magri, arXiv e-prints , arXiv:2103.03174 (2021).
- [72] M. Hoffman, E. Brochu, and N. de Freitas, in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11 (AUAI Press, Arlington, Virginia, USA, 2011) p. 327–336.
- [73] J. Yperman and T. Becker, arXiv e-prints , arXiv:1611.05193 (2016).
- [74] J. Reinier Maat, N. Gianniotis, and P. Protopapas, arXiv e-prints , arXiv:1903.05071 (2019).
- [75] S. Kullback and R. A. Leibler, *The Annals of Mathematical Statistics* **22**, 79 (1951).
- [76] D. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization* (John Wiley & Sons, New York, Chicester, 1992).

- [77] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, *Nature Methods* **17**, 261 (2020).