

Tailored minimal reservoir computing: on the bidirectional connection between nonlinearities in the reservoir and in data

Davide Prosperino,¹ Haochun Ma,¹ and Christoph R  th²

¹*Ludwig-Maximilians-Universit  t M  nchen, Faculty of Physics, Geschwister-Scholl-Platz 1, 80539 Munich, Germany*

²*Deutsches Zentrum f  r Luft- und Raumfahrt (DLR), Institute of Materials Physics in Space, Linder H  he, 51147 Cologne, Germany*

(*Electronic mail: christoph.raeth@dlr.de)

(Dated: 26 March 2025)

We study how the degree of nonlinearity in the input data affects the optimal design of reservoir computers, focusing on how closely the model’s nonlinearity should align with that of the data. By reducing minimal RCs to a single tunable nonlinearity parameter, we explore how the predictive performance varies with the degree of nonlinearity in the reservoir. To provide controlled testbeds, we generalize to the fractional Halvorsen system, a novel chaotic system with fractional exponents. Our experiments reveal that the prediction performance is maximized when the reservoir’s nonlinearity matches the nonlinearity present in the data. In cases where multiple nonlinearities are present in the data, we find that the correlation dimension of the predicted signal is reconstructed correctly when the smallest nonlinearity is matched. We use this observation to propose a method for estimating the minimal nonlinearity in unknown time series by sweeping the reservoir exponent and identifying the transition to a successful reconstruction. Applying this method to both synthetic and real-world datasets, including financial time series, we demonstrate its practical viability. Finally, we transfer these insights to classical RC by augmenting traditional architectures with fractional, generalized reservoir states. This yields performance gains, particularly in resource-constrained scenarios such as physical reservoirs, where increasing reservoir size is impractical or economically unviable. Our work provides a principled route toward tailoring RCs to the intrinsic complexity of the systems they aim to model.

Reservoir computing is a powerful tool for modeling nonlinear systems, but its design often relies on heuristics. Here, we show that predictive accuracy improves when the reservoir’s nonlinearity is matched to that of the data. Using a deterministic, minimal reservoir framework and a novel chaotic system with tunable fractional exponents, we isolate this relationship and demonstrate that the smallest nonlinearity in the data plays a key role. This insight enables a method to estimate nonlinearity from time series alone, which we validate on synthetic and real-world data, including financial markets. We also show how augmenting classical reservoirs with tailored nonlinearities improves performance, especially useful in hardware-limited settings.

I. INTRODUCTION

Reservoir computing^{1–3} is a machine learning framework for modeling and predicting nonlinear dynamical systems, built on the idea of using a fixed recurrent dynamical system—the *reservoir*—and linearly combining its dynamics to create predictions. The work by Luko  evi  ius and Jaeger⁴ offers a great introduction to the theory of traditional reservoir computers (RCs).

Despite its practical success in synthetic systems^{5,6} and real world systems^{7–11}, classical reservoir computers remain somewhat heuristic: the reservoir’s weights are initialized randomly, and while empirical studies on the reservoir structure and weights have been performed^{12,13}, the optimal design of the reservoir is not well understood analytically. This random-

ness and complexity hinder a principled understanding of *why* RCs work so well, since we need to account not only for the choice of parameters, but also for the actual realization of the random numbers used in the process.

The topic of the randomness in traditional RC has been addressed in two ways: In so-called ‘next generation reservoir computing’ (NGRC)¹⁴ the reservoir is replaced by linear and nonlinear combinations of the input variables and their time lags without any weights. While this approach performs very well, also with limited training data¹⁵, the typical character of a reservoir as being a dynamical substrate with (fading) memory, which responds to some input data, is lost in this RC setup. In minimal reservoir computing (minRC)¹⁶, however, the reservoir still exists, but it is simplified by replacing the large random network by a structured block-diagonal matrix splitting the reservoir into multiple smaller sub-reservoirs, each working on a single feature. All random elements in the input layer and in the reservoir are removed, enabling a systematic analysis of RC architectures. Furthermore, the nonlinear activation at each reservoir node can be replaced by shifting the nonlinearity to the output layer: the readout operates on generalized reservoir states that include powers of the reservoir’s linear state evolution. This deterministic setup retains RC’s computational efficiency but yields a more interpretable, tractable model.

Building on this minimal RC framework, this work focuses on a fundamental question: How nonlinear should the reservoir states be in order to adequately model a given nonlinear input data? In classical or minimal RC the approach is to introduce nonlinear reservoir features to help capture nonlinear structures in the input. However, it remains unclear what degree of nonlinearity is truly needed in the reservoir to rep-

resent the nonlinear dynamics of the data. Intuitively, if the input data's dynamics are only mildly nonlinear, an overly strong nonlinearity in the reservoir might be unnecessary (or even detrimental), whereas if the data's generative process is highly nonlinear, a linear or weakly nonlinear reservoir will be insufficient to capture its behavior. We aim to formalize this intuition and determine how to tailor the reservoir's nonlinearity to the complexity of the input.

In this study, we introduce a tailored minimal RC approach to systematically investigate the matching of data and reservoir nonlinearities. In Sec. III we reduce the minimal RC model to its essence by using a single tunable nonlinearity parameter in the generalized reservoir states, and we examine the impact of this nonlinearity on prediction performance for datasets of varying complexity.

All traditional chaotic systems use integer exponents as nonlinearities with Thomas' system¹⁷ sine function being a notable exception. Here in Sec. II we introduce a fractional Halvorsen system as a novel data generator generalizing the classical Halvorsen attractor to fractional exponents in the nonlinear terms. This allows us to produce time series with a controllable degree of nonlinearity, providing an ideal testbed for our study.

We use this data to perform extensive experiments in Sec. IV measuring the prediction performance and studying the relationship between the nonlinearity in the data and the nonlinearity in the model.

Applying our findings in reverse, we find in Sec. V that we can use this framework to determine the smallest nonlinearity present in data by measuring the prediction performance over different nonlinearities in the model and noting when the prediction error minimizes. Lastly, in that section we also transfer our insights from minimal RCs to improve the performance of classical RCs by introducing fractional, generalized reservoir states.

II. DATA

Originally introduced as a model for atmospheric convection, the Lorenz system¹⁸ has become the benchmark system in the study of chaotic systems and has been extensively used in research on reservoir computers^{5,6}. However its nonlinearities consist of mixed variables, which makes it hard to control the exponent, and thus the nonlinearity. For this reason we introduce the Halvorsen system¹⁹ in our study, after performing initial studies on the Lorenz system. We introduce a modified version of the Halvorsen system, in which we can control the nonlinearity in the data more precisely.

For all our integrations of the trajectories we use the explicit Runge–Kutta method of order 5(4)²⁰ utilizing a step size of $\Delta t = 0.01$ unless stated otherwise. Our initial condition consists of a uniformly distributed random value between -20 and 20 for the Lorenz system, and due to stability reasons we use the point $(0.1, 0, 0)^T$ as initial condition for all Halvorsen realizations. In each case we discard the first 10^4 steps as transient behavior.

For all calculations involving reservoir computers we use the SCAN software package²¹.

A. Lorenz system

The Lorenz system is a set of coupled, nonlinear differential equations given by¹⁸

$$\dot{x}_1 = -\sigma x_1 + \sigma x_2 \quad (1a)$$

$$\dot{x}_2 = \rho x_1 - x_2 - x_1 x_3 \quad (1b)$$

$$\dot{x}_3 = -\beta x_3 + x_1 x_2, \quad (1c)$$

where we use the standard parametrization exhibiting chaotic behavior of $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$.

While previous studies have explored variations in the Lorenz system's nonlinear terms²², controlling the overall degree of nonlinearity across coordinates remains challenging due to the nonlinearity consisting of combinations of two variables.

B. Fractional Halvorsen system

The Halvorsen system, in contrast to the Lorenz system, is a chaotic system, which has its nonlinearities in a single variable in each dimension. Originally, the nonlinearity consists of quadratic terms, however in this work we want to introduce fractional exponents in each dimension. This allows for a control of the nonlinearity of the system by modifying the exponent in each equation, which leads to the introduction of the modified, fractional Halvorsen system given by

$$\dot{x}_1 = -ax_1 - 4x_2 - 4x_3 - x_2^{\xi_1} \quad (2a)$$

$$\dot{x}_2 = -ax_2 - 4x_3 - 4x_1 - x_3^{\xi_2} \quad (2b)$$

$$\dot{x}_3 = -ax_3 - 4x_1 - 4x_2 - x_1^{\xi_3}. \quad (2c)$$

The canonical choice displaying chaotic behavior is $a = 1.3$ and $\xi_i = 2$. However, for this study we want to explore different values of ξ_i . Here, we do not want to limit ourselves to integer exponents, but introduce the study of fractional exponents in this context.

For fractional exponents of the form $\xi_i = n_i/d$ we need to rewrite Eqs. 2 slightly and use the definition of $x^{\frac{n_i}{d}} = \sqrt[d]{x^{n_i}}$ to substitute for x^{ξ_i} . In order to prevent complex valued trajectories, we limit ourselves to even choices for n_i .

During our studies we discovered that a denominator of $d = 50$ provides a good trade-off between the accessible granularity and computational stability.

In Fig. 1 we performed a grid search over the two free parameters of the fractional Halvorsen system, in order to find the region of interests with chaotic behavior. Here, we fix all exponents to be the same with $\xi_i = \xi_1 = \xi_2 = \xi_3$. We found that for a parameter of $a = 3.98$ we observe chaotic behavior over a range of exponents ξ_i .

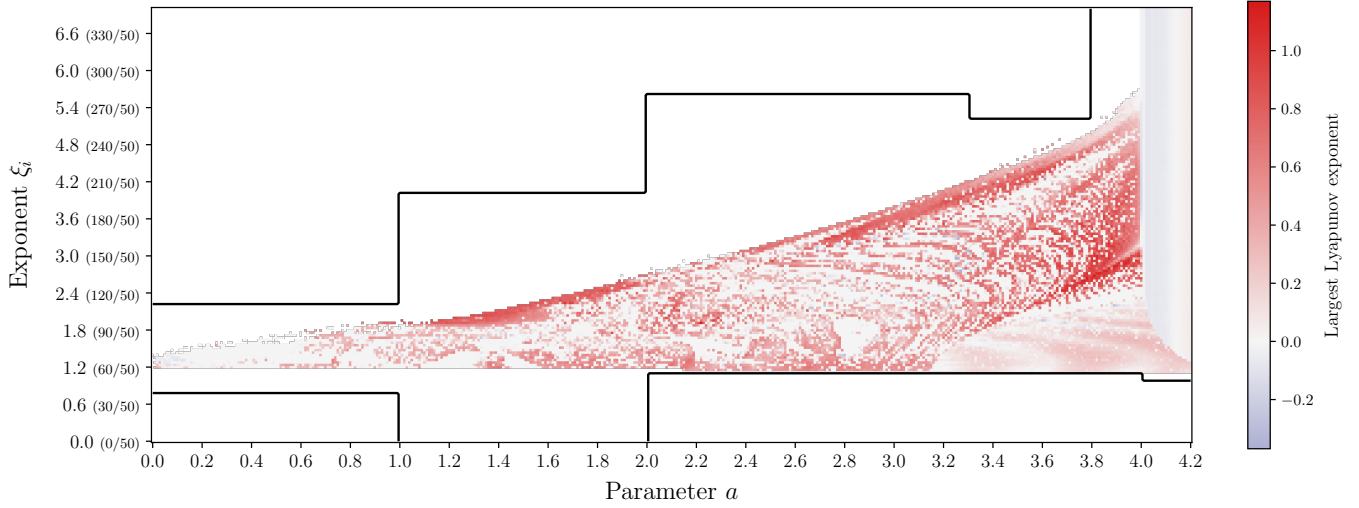


FIG. 1. The results of our grid search for the calculation of the largest Lyapunov exponent for different parameters of a and ξ_i are shown. For this plot all exponents of the fractional Halvorsen system are the same and fixed at ξ_i . White color inside the black boundary indicates a diverging trajectory for that parameter combination, while a white color outside the boundary indicates that the parameter combination has not been explored. For each parameter combination we simulate 50000 steps and discard the first 10000 steps as transient behavior. In total, we performed 44 625 experiments.

C. Thomas system

The Thomas system¹⁷ is a chaotic system, which has its nonlinearity not in exponentiation, but instead in a sine function. It is defined by

$$\dot{x}_1 = -bx_1 + \sin x_2 \quad (3a)$$

$$\dot{x}_2 = -bx_2 + \sin x_3 \quad (3b)$$

$$\dot{x}_3 = -bx_3 + \sin x_1, \quad (3c)$$

where we use a parameter of $b = 0.21$. The sine function can be defined by Taylor expansion as sum of polynomials, where the first nonlinear term appears as third order.

D. Surrogate systems

In Sec. V A we aim to identify the smallest nonlinearity in a given time series. To ensure that the observed effects indeed arise from nonlinearity in the data, we generate Fourier transform (FT) surrogates to obtain a truly linearized reference for our measure²³. If the measure from the original time series differs significantly from this linearized background, it strongly indicates that the observed effect stems from the system's nonlinearity.

Surrogating the data destroys the nonlinear features of a time series, while keeping the linear ones unaffected²⁴. We create the surrogate time series by firstly performing a Fourier transformation $\mathcal{F}$ on the original time series $\underline{x}$, separating the data into amplitudes A_k and phases ϕ_k . The linear properties are now stored in the amplitudes and the nonlinear ones in the phases. By replacing the original phases ϕ_k with uniformly distributed numbers between $[0, 2\pi]$, ϕ_k^{rand} , we destroy the nonlinear features of the original time series. The

surrogate time series $\underline{x}_s$ is then given by the inverse Fourier transformation $\mathcal{F}^{-1}$ of the original amplitudes A_k with the randomized phases ϕ_k^{rand} , sketched by

$$\underline{x}_s = \mathcal{F}^{-1} [A_k \exp(i\phi_k^{\text{rand}})] . \quad (4)$$

In order to get a robust estimate of the surrogate measures, we create multiple realizations of the surrogate time series and calculate the measure of interest across all of them and report the average with a standard deviation indicating the spread of the values.

III. RESERVOIR COMPUTING

A reservoir computer (RC) is a specialized form of recurrent neural network in which the recurrent connections remain fixed after initialization, rather than being adapted during training. The input signal is mapped into a random, high-dimensional state space, causing the randomly defined reservoir to synchronize with the input's dynamics. The resulting reservoir states now reflect the time evolution of the input in the high-dimensional space. They are then combined through a linear readout layer to produce predictions in the measured space. Such architectures have been shown to excel at forecasting chaotic systems. For a thorough overview of classical reservoir computing techniques we want to refer to Lukoševičius and Jaeger⁴.

A. Reservoir computers

In a classical reservoir computer the input time series $\underline{x}$ is mapped *randomly* into a high-dimensional space of dimen-

sional data d through the input matrix $\mathbf{W}_{\text{in}}$. Once the data is embedded, the reservoir governs the internal dynamics of the reservoir computer. The reservoir states are then linearly combined to form the prediction.

The reservoir is a randomly connected graph of d nodes represented by its adjacency matrix $\mathbf{A}$, which describes the connection of the nodes with each other. Different topologies for the reservoir $\mathbf{A}$ have been studied, showing that, generally, random networks or small-world networks work better than scale-free networks¹³. The reservoir is scaled to a target spectral radius of ρ^* to regulate the reservoir's dynamical stability and ensure it does not diverge. At each time t , we represent the reservoir by the state vector $\underline{r}(t)$, whose components reflect the activity of each node. Its evolution is given by

$$\underline{r}(t+1) = f(\mathbf{A}\underline{r}(t) + \mathbf{W}_{\text{in}}\underline{x}(t)) \quad , \quad (5)$$

and it is usually initialized with the zero vector $\underline{r}(0) = \underline{0}$. f refers to a nonlinear function and the hyperbolic tangent is the usual choice.

After a synchronization phase, sometimes referred to as warm-up phase, the reservoir state $\underline{r}$ represents the dynamic of the input data $\underline{x}$ in the high-dimensional space. The reservoir states can then be linearly combined by an output matrix $\mathbf{W}_{\text{out}}$ to reproduce the time series in the original space.

For finding the output matrix $\mathbf{W}_{\text{out}}$, the reservoir states $\underline{r}(t+1)$ with their corresponding output $\underline{x}(t+1)$ are recorded during the training process, collecting a total number of l training steps. We store the reservoir states and their corresponding outputs in the matrices $\mathbf{R}$ and $\mathbf{X}$ respectively, and perform a ridge regression²⁵ to solve the equation $\mathbf{W}_{\text{out}}\mathbf{R} = \mathbf{X}$. The solution for the output matrix is given by

$$\mathbf{W}_{\text{out}} = \mathbf{X}\mathbf{R}^T (\mathbf{R}\mathbf{R}^T + \beta \mathbf{1})^{-1} \quad , \quad (6)$$

where we have applied the mathematical trick described by Lukoševičius and Jaeger⁴, consisting of multiplying $\mathbf{R}^T$ to the right of the problem to solve, in order to make the optimization independent of the training length. Here, $\mathbf{1}$ describes the identity matrix and β the regularization parameter of the ridge regression.

For creating predictions after the training, the reservoir computer needs to be synchronized to the immediate history of the starting point of the prediction, to ensure that the reservoir state represents the current dynamics. After that, the predictions can be fed successively into the reservoir computer to reproduce the learned dynamics.

B. Minimal reservoir computers

Classical reservoir computers utilize random initializations, rendering their study challenging since we must account both for the chosen setup and for the particular realization of the random numbers. To eliminate the element of randomness in reservoir computing, we introduced minimal reservoir computers¹⁶ as architectures defined entirely without random components. This simplifies analyzing their inner workings,

as the absence of random initializations and network configurations allows for a more direct examination of the reservoir's dynamic.

Minimal reservoir computers can be seen as deterministic subsets of classical reservoir computing approaches. In the following we outline their definition but want to refer to Ref. 16 for a detailed discussion.

The input data is not embedded randomly in a high-dimensional space. Instead of creating random features from the data, as done in classical RC, for minimal RC we construct the features from the set of all subset sums of the coordinates. We use all partial sums which are creatable by the coordinates and we feed multiple copies of each feature into the reservoir. The number of copies fed into the reservoir is defined by the block size b and each copy is assigned a weight in $[0, 1]$ according to the weight vector $\underline{w}$ given by

$$\underline{w} = \left(1 \quad \sqrt{\frac{b-2}{b-1}} \quad \dots \quad \sqrt{\frac{1}{b-1}} \quad 0 \right)^T \quad . \quad (7)$$

For a three-dimensional system the input matrix $\mathbf{W}_{\text{in}}$ is constructed by

$$\mathbf{W}_{\text{in}} = \begin{pmatrix} \underline{w} & \underline{0} & \underline{0} \\ \underline{0} & \underline{w} & \underline{0} \\ \underline{0} & \underline{0} & \underline{w} \\ \underline{w} & \underline{w} & \underline{0} \\ \underline{w} & \underline{0} & \underline{w} \\ \underline{0} & \underline{w} & \underline{w} \\ \underline{w} & \underline{w} & \underline{w} \end{pmatrix} \quad , \quad (8)$$

resulting in the following feature vector being fed into the reservoir:

$$\mathbf{W}_{\text{in}}\underline{x} = (\underline{w} \odot \underline{x}_1 \quad \underline{w} \odot \underline{x}_2 \quad \dots \quad \underline{w} \odot \underline{x}_{1+2} \quad \dots \quad \underline{w} \odot \underline{x}_{1+2+3})^T \quad . \quad (9)$$

Here, $\odot$ describes the element-wise multiplication between the two vectors, and for each feature f , the vector $\underline{x}_f$ is defined as $\underline{x}_f = (x_f \quad \dots \quad x_f)$ to match the dimensionality of $\underline{w}$. The subscript indicates the coordinates out of which the features is constructed by summing over them.

Instead of using a single, big reservoir, the reservoir is constructed as several, disconnected, smaller reservoirs, which leads to a adjacency matrix in block-diagonal form. For each feature f we use a small reservoir $\mathbf{J}_f$ consisting of a matrix of ones, meaning that each node is connected to every other node. The final reservoir is then constructed by

$$\mathbf{A} = \frac{\rho^*}{b} \begin{pmatrix} \mathbf{J}_{x_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{x_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_{x_{1+2+3}} \end{pmatrix} \quad , \quad (10)$$

where the scaling factor of ρ^*/b ensures that the reservoir $\mathbf{A}$ has the spectral radius of ρ^* . The idea of block-diagonal reservoirs has been also successfully applied to classical RC architectures^{10,26}.

Unlike in classical RC architectures, in minimal RC the reservoir states are evolved purely linearly by

$$\underline{r}(t+1) = \mathbf{A}\underline{r}(t) + \mathbf{W}_{\text{in}}\underline{x}(t) \quad . \quad (11)$$

In the original definition, the nonlinearity is added *after* the evolution by extending the reservoir state to a generalized reservoir state $\tilde{r}$ containing copies of itself raised up to a maximal power of $\eta_{\max}$ given by

$$\tilde{r} = (r \ r^2 \ \dots \ r^{\eta_{\max}-1} \ r^{\eta_{\max}})^T. \quad (12)$$

The exponentiation is understood to be applied element-wise.

However, in this article we want to introduce a slightly modified setup of the generalized states containing only the linear reservoir state and a single nonlinearity η . The generalized reservoir state $\tilde{r}$ reduces to

$$\tilde{r} = (r \ r^\eta)^T. \quad (13)$$

Additionally, we want to also allow for fractional nonlinearities of the form $\eta = n/d$, where we apply the same substitution as in Sec. II B of $r^{\frac{n}{d}} = \sqrt[d]{r^n}$. Each operation is understood to be performed element-wise and again we only allow even numerators n in order to prevent complex valued reservoir states.

Utilizing this reduced definition of minimal RCs, we can study the dependence between the nonlinearity present in data and the smallest nonlinearity required to successfully predict those systems.

The training and prediction routines are identical to the classical RC's case. We train each minimal RC by performing a ridge regression of the generalized reservoir states at each time point against the corresponding output, and create predictions by iteratively inputting the previous prediction.

IV. MINIMAL REQUIRED NONLINEARITY

In this section we want to present our results and analyze the connection between nonlinearities expressed in data and the required nonlinearities for reproducing those.

A. Lorenz system

We begin our analysis with a wide grid search over *all* hyperparameters of minimal RCs in Fig. 2 for the Lorenz system. We want to emphasize at this point that each tile in Fig. 2 completely and uniquely describes a minimal reservoir computer instance. For minimal RCs, the repeated experiments for a certain setup solely average out the effect of different training data or put differently, being on different parts of the attractor. In contrast to classical RCs, where repeated experiments are required to control for the randomness in their construction in addition to different parts of the attractor being used for training.

For the results in Fig. 2 we utilize the traditional setup of minimal RC, where all integer exponents up to a maximal exponent $\eta_{\max}$ are used in the generalized reservoirs state $\tilde{r}$. We observe that the prediction is successful for a wide range of hyperparameters using a minimal data setup of only 1000 training steps. Additionally, we can report that the performance seems to increase when including higher order nonlinearities. So, including higher nonlinearities, which are not

present in the data, seems to give the minimal RC more flexibility and allow a more precise approximation. Increasing the block size b leads to instabilities for higher order terms, which is why we recommend using a relatively small block size not exceeding $b = 5$.

In our findings we also confirm that nonlinearity is required for predicting a nonlinear system, as demonstrated by the black column for $\eta_{\max} = 1$ in Fig. 2. However, *how much* nonlinearity is needed?

We analyze the transition from the first column of failing predictions ($\eta_{\max} = 1$) to the second column of successful predictions ($\eta_{\max} = 2$) in more detail. For that we apply the new fractional reservoir states for minimal RCs, meaning that the generalized reservoir states only contain two components: the linear one r and a single nonlinear one $\sqrt[d]{r^n}$. We sweep the nonlinearity of the minimal RC from $\eta = 1$ to $\eta = 4$ with a denominator of $d = 50$ in steps of two. We use a block size of $b = 3$, a regularization parameter of $\beta = 10^{-6}$, and iterate through each spectral radius from 10^{-5} up to 0.5 including 0. For each combination of RC exponent η and spectral radius ρ^* we perform twenty realizations. We train each minimal RC on 1000 points and synchronize using 100 points. The results are presented in Fig. 3, where we analyze the short and long-term prediction. The short-term prediction is measured using the forecast horizon, while we define a long-term prediction successful, if the reconstructed Lyapunov exponent and the reconstructed correlation dimension do not differ more than 0.1 from the original value.

For the short-term prediction we clearly observe a peak, when the nonlinearity in the data corresponds to the nonlinearity of the minimal RC. We find this result to be stable for multiple hyperparameters. Interestingly, this strong connection does not hold for the long-term prediction, where a reliable reconstruction of the attractor is possible even if the nonlinearity of the minimal RC exceeds the nonlinearity of the data. Here we discover the relationship to be dependent on the spectral radius and observe that, in general, a lower spectral radius allows for a bigger deviation of the exponent of minimal RC against the exponent observed in data. This implies that for an optimal short-term prediction the exponent of the minimal RC needs to exactly match the exponent of the data, while for a reliable long-term prediction a certain, small overestimation of the exponent in the data is allowed.

So far, we have only studied the Lorenz system, which contains a single order, integer nonlinearity of two. However, we aim to explore how general our results are, which is why we expand this study on the fractional Halvorsen system with a controllable and non-integer nonlinearity in the following section.

B. Fractional Halvorsen with $\xi_1 = \xi_2 = \xi_3$

In this section we want to expand on the results of the Lorenz system and study whether the peak at the exponent in the data is an oddity of the Lorenz system and the power two, or whether we can observe a more general pattern. Controlling the total nonlinearity in the Lorenz system is a nontrivial

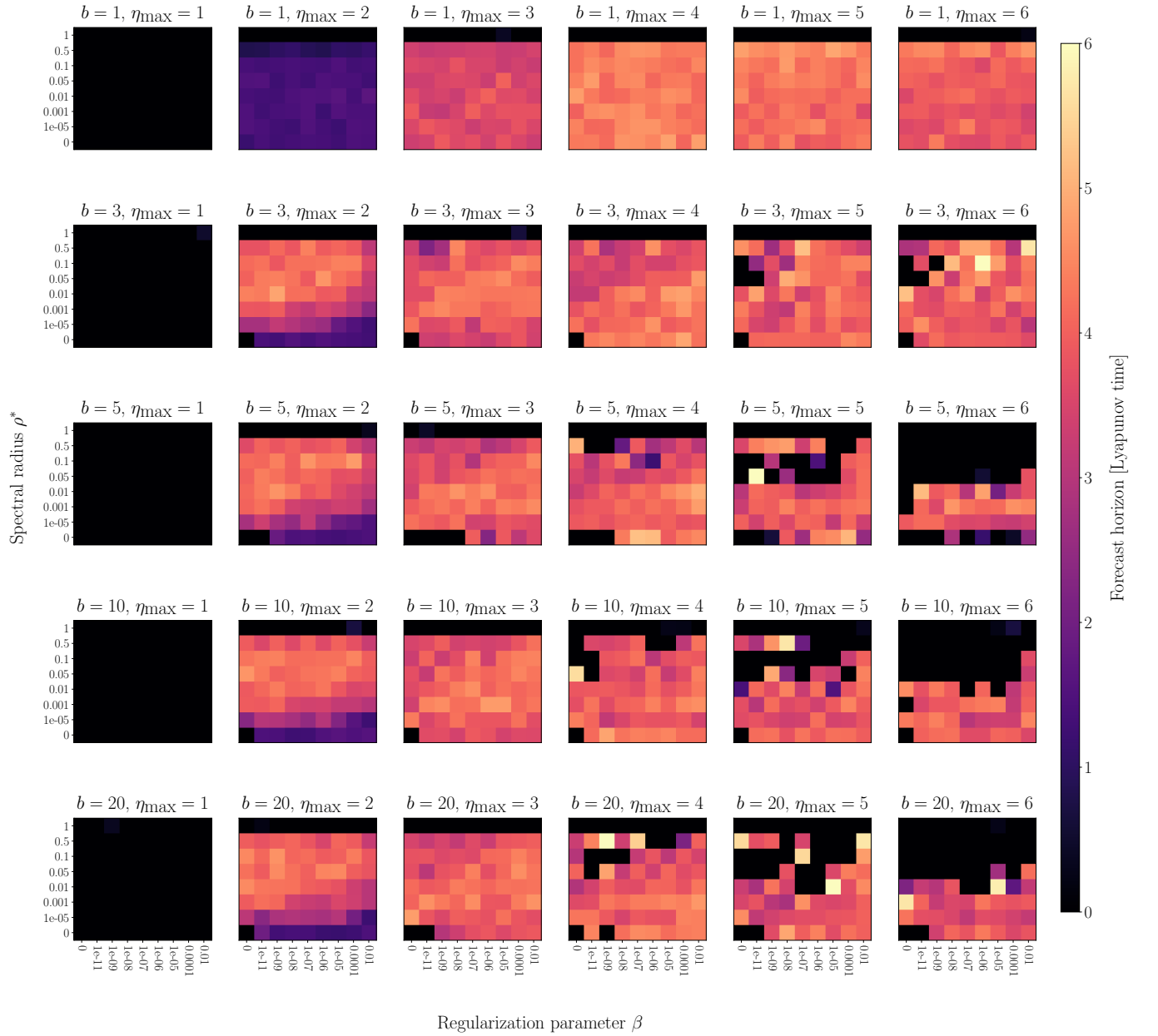


FIG. 2. The performance for different hyperparameters of minimal RCs in the classical setup containing all nonlinearities in the generalized states up to $\eta_{\max}$ predicting the Lorenz system is shown. The performance of successful runs using the forecast horizon is measured in multiple of Lyapunov times. For each realization we use 1000 data points for training, out of which 10 are used for synchronization, and a step size of $\Delta t = 0.025$. Each tile shows the average performance of at least 35 realizations and in total we performed 98 297 experiments.

task due to the asymmetric equations and mixed nonlinearity, which is why we use the fractional Halvorsen system in Eqs. 2.

For this experiment we set all exponents of the fractional Halvorsen system to the same value of $\xi_i = \xi_1 = \xi_2 = \xi_3$ and test numerators ranging from $n_i = 132$ up to including $n_i = 280$ in steps of two with a denominator of $d = 50$ using a parameter of $a = 3.98$. This corresponds to exponents ranging from $\xi_i = 2.64$ to $\xi_i = 5.6$. The exponents for the minimal RCs are ranging from $\eta = 1.32$ to $\xi_i = 5.6$ with the same denominator and step size. We use a block size of $b = 3$, a

spectral radius of $\rho^* = 10^{-3}$, and a regularization parameter of $\beta = 10^{-6}$. We train each minimal RC on 5000 points and synchronize using 1000 points and perform seven runs per parameter combination. The findings for this experiment are shown in Figs. 4 & 5.

We observe a clear peak at $\eta = \xi_i$, indicating that hitting the exact nonlinearity of the data is important for a successful prediction. Interestingly, we note that even overshooting the nonlinearity in the data will not generally improve the predictive power. Those single lines in Fig. 4 increasing after $1.5 \xi_i$ are systems with a very low Lyapunov exponent and thus eas-

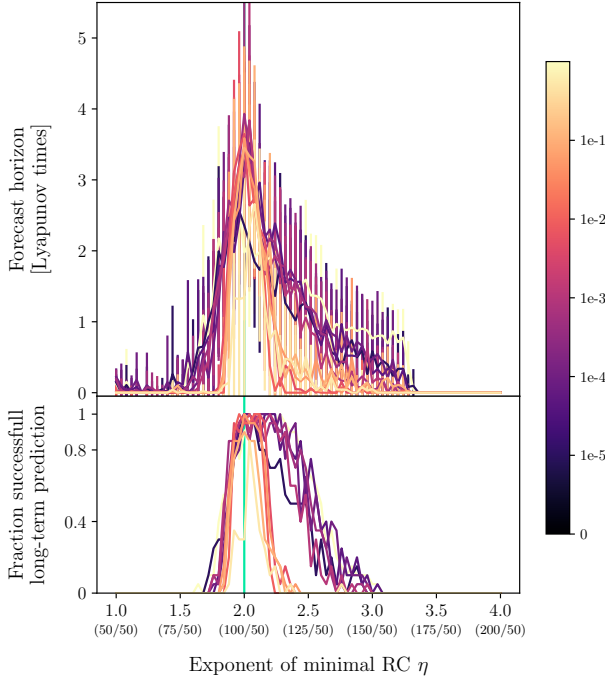


FIG. 3. The short and long-term performance of minimal RCs reproducing the Lorenz system is presented. We performed twenty experiments per parameter combination, so the upper plot reports the mean and standard deviation of those twenty runs, while the lower one shows the successful reproductions out of those. The green line shows the true nonlinearity of the Lorenz system. We show the results of 18240 experiments.

ier to predict. The bulk of interesting results lies before and around $\eta = \xi_i$. Additionally, we want to note the width of the peak in Fig. 4, which seems to be (roughly) constant using a relative x-axis. This finding could be useful when building a non-integer library in Sec. VB, indicating that for larger exponents a less precise guess, in absolute terms, is required for a reasonable prediction performance than for smaller ones.

C. Fractional Halvorsen with $\xi_1 = \xi_2 \neq \xi_3$

So far, we have only studied the fractional Halvorsen system with all equal exponents. Due to the symmetry of the equations and the equality of the exponents the data only contained a single nonlinearity. Here we want to systematically study the inclusion of two different exponents.

For this case we set $\xi_1 = \xi_2 = \xi_{1,2}$ in Eqs. 2 and differ it from ξ_3 . For $\xi_{1,2}$, ξ_3 , and η we use numerators ranging from $n = 54$ up to $n = 280$ in steps of two with a denominator of $d = 50$. This corresponds to values from $\eta_{\min} = 1.08$ to $\eta_{\max} = 5.6$ for all exponents. We use a parameter of $a = 3.98$. For each set of parameters we perform five experiments.

We show the result in Fig. 6. We order the values such that $\xi_s = \min(\xi_{1,2}, \xi_3)$ and correspondingly $\xi_l = \max(\xi_{1,2}, \xi_3)$. We do this simplification since we find that the results do not change depending on the ordering of the two exponents. For

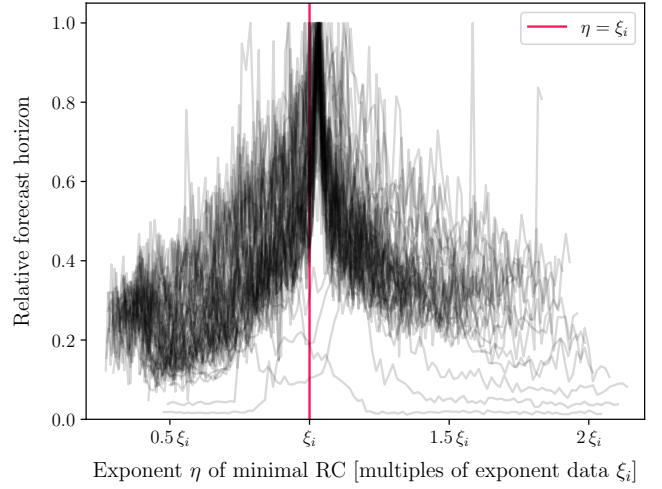


FIG. 4. The mean relative forecast horizon for different nonlinear exponents η of minimal RCs predicting the fractional Halvorsen system is shown. Each gray line represents the mean for a different ξ_i . For each parameter and exponent we perform seven runs and calculate the mean forecast horizon, which we normalize against the peak value. However, we omit the error bars in the interest of readability. We show the results of 62130 experiments.

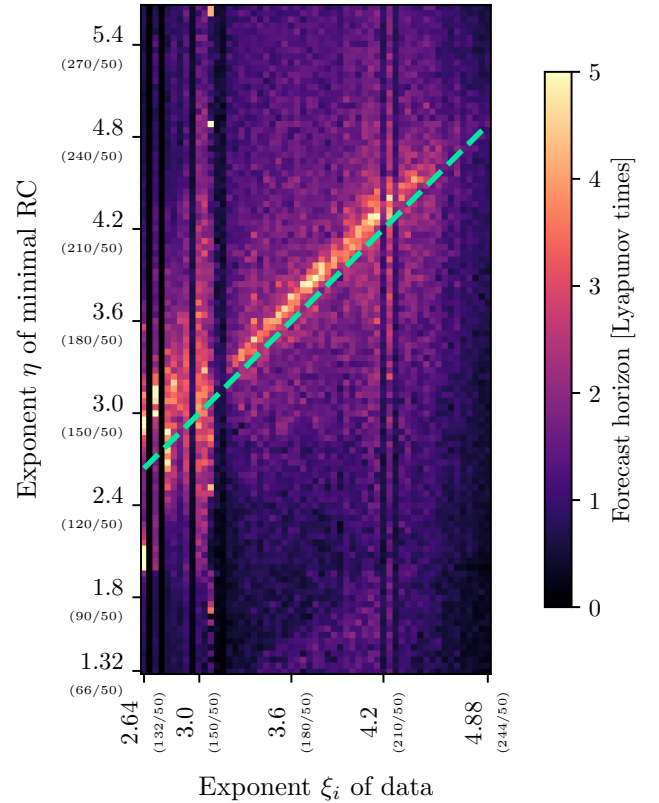


FIG. 5. This plot shows the absolute forecast horizon in for different exponents ξ_i in data and η in the model. It uses the same underlying data of the fractional Halvorsen system as setup in Fig. 4. Each tile shows the mean forecast horizon of five realizations and the green line indicates $\eta = \xi_i$.

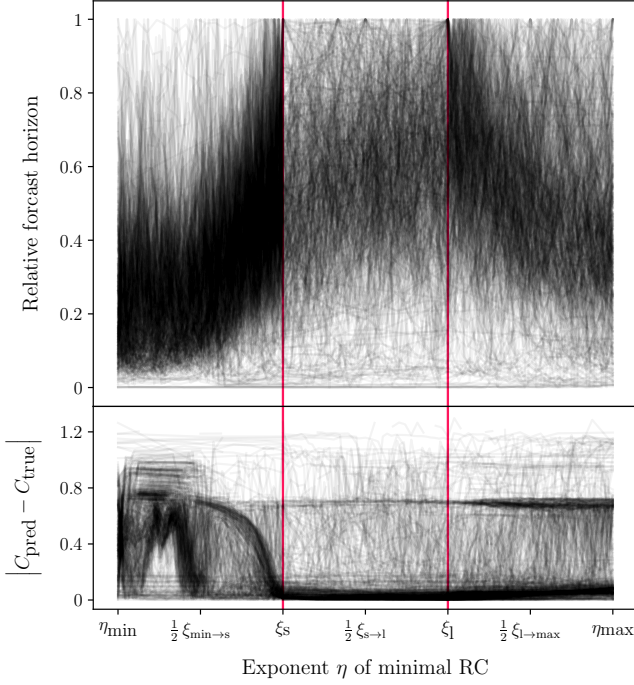


FIG. 6. The prediction performance for predicting the fractional Halvorsen system with two equal, fractional exponents is shown: $\xi_1 = \xi_2 \neq \xi_3$. We show the mean relative forecast horizon for different nonlinear exponents of minimal RCs in relative terms in upper plot and the correlation dimension error in the lower plot. We build this figure from 56031 experiments resulting in 983 trajectories.

the sake of readability, we define the relative distance between them as

$$p \xi_{a \rightarrow b} := \xi_a + p (\xi_b - \xi_a) . \quad (14)$$

While we cannot find a quantitative message in Fig. 6, we can make qualitative statements about this experiment. Looking at the interval $[\eta_{\min}, \xi_s]$ we find the previous pattern of the prediction performance increasing until we hit the first nonlinearity of the data. In the interval $[\xi_s, \xi_l]$ between the two nonlinearities of the data, there is neither a clear pattern nor consistent peaks. The last interval $[\xi_l, \eta_{\max}]$ shows again a familiar pattern of the performance decreasing when the exponent of the minimal RC gets larger than the exponent present in the data. The interesting finding of this experiment is the peaking of the prediction performance at ξ_s and ξ_l : The prediction performance is best if one of the nonlinearities is hit. Unexpectedly, the smallest one seems to be the most important one providing the most contribution to a successful prediction.

This indicates that for mixed nonlinearities we do not observe a significant improvement after including the smallest nonlinearity. It seems that including the smallest nonlinearity is more important than finding all of them. This is an important finding, since real-life data cannot be expected to contain only a single nonlinearity, and this finding hints that finding the smallest one is the most valuable one regarding the prediction performance.

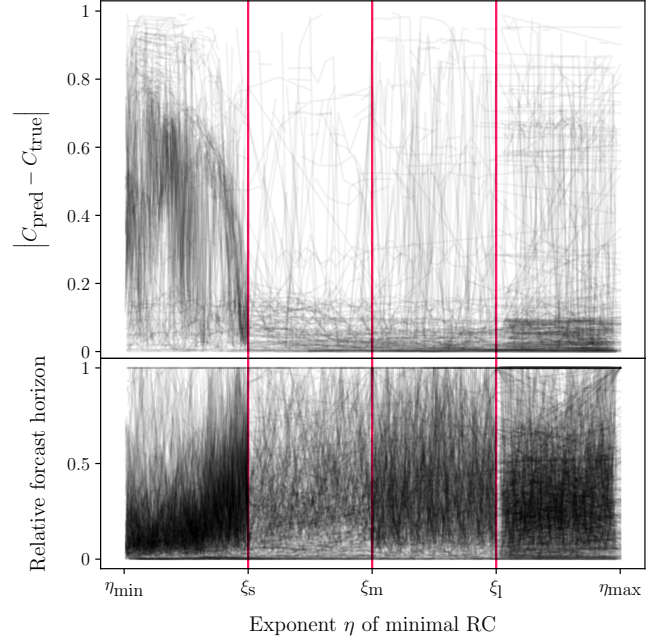


FIG. 7. The prediction performance for predicting the fractional Halvorsen system with three different fractional exponents is shown: $\xi_1 \neq \xi_2 \neq \xi_3$. We show the absolute difference between the true correlation dimension and the predicted correlation dimension in upper plot. The lower plot shows the relative forecast horizon for different nonlinear exponents of minimal RCs in relative terms. We build the 564 trajectories from 32 148 experiments.

D. Fractional Halvorsen with $\xi_1 \neq \xi_2 \neq \xi_3$

In this section we want to complete the analysis of the fractional Halvorsen by studying the case where all three exponents differ from each other. A coordinated study, as performed in previous sections, is not feasible due to the huge number of possible combinations. Instead, we randomly pick three numerators from 52 to 280 with the common denominator of 50 for the exponents ξ_i , and, if the trajectory is a valid chaotic system, we sweep the nonlinearity of the minimal RC from $\eta_{\min} = 52/50$ to $\eta_{\max} = 280/50$ in steps of two in the numerator. The exponents are ordered by magnitude and named $\xi_s < \xi_m < \xi_l$. We want to test whether our qualitative result of previous section also holds for the case of three different exponents.

The results are presented in Fig. 7, where we can confirm the pattern of Fig. 6, in which the error of the predicted correlation dimension drops to zero as soon as the nonlinearity of our estimator exceeds the smallest nonlinearity of the data. Due to the small number of trajectories, other patterns are not testable qualitatively in Fig. 7. Nevertheless, confirming the special situation of the smallest nonlinearity present in data is an important finding.

V. APPLICATIONS

For our findings we find two possible applications. A first application consists of determining the smallest nonlinearity present in data, by sweeping the nonlinearity in the minimal RC model and observing its output. For the second application we want to take the findings from minimal RC and apply them to traditional RC architectures, by extending traditional RCs with fractional nonlinearities.

A. Smallest nonlinearity in data

A possible application of this framework consists of discovering the smallest nonlinearity present in data. We have shown previously that the correlation dimension of the predicted values approaches the correlation dimension of observed data for the first time, if the exponent of the minimal RC matches the smallest exponent present in data. We can use this observation to build a test for the smallest exponent present in data.

For that we first determine the correlation dimension of the time series. We then sweep through a range of non-integer exponents for the minimal RC. For each exponent we calculate the correlation dimension of the predicted time series. We repeat the same for surrogate versions of the time series, in order to assure that the observed effect really stems from the nonlinearity of the data. When the predicted correlation dimension matches the true one and is outside the surrogates ones, we found an approximation for the smallest nonlinearity present in data. If the measure of the time series does not exceed the surrogate measure, the determination of a smallest nonlinearity failed with this method.

We test our method on some chaotic systems, with a known nonlinearity, and some financial data, with unknown nonlinearity, and present the results in Tab. I and Figs. 8f.

The chaotic systems of choice are the Lorenz system, the classical Halvorsen system with $\xi_i = 2$, and the Thomas system. We parametrize our minimal RCs with a block size of $b = 3$, a target spectral radius of $\rho^* = 0.1$, and a regularization parameter of $\beta = 10^{-6}$. We synchronize our model using 100 data points and train it on 1 000 data points. The nonlinearity exponent η is swept from $52/50$ to $280/50$ in steps of two in the numerator. Fig 8 shows the result for the Lorenz system and the Thomas system.

For the Lorenz system we observe the predicted correlation dimension to rise very quickly when the nonlinearity in the minimal RC model approaches two, the nonlinearity in the data. The value for the correlation dimension differs significantly from the surrogate background rendering our estimate in it confident. Interestingly, for the Thomas system the correlation dimension becomes a computable number, and instantly reaches the correlation dimension of the data, for an exponent of 2.92, being very close to 3. The Thomas system has its nonlinearity in the sine function, and an exponent of 3 constitutes the first nonlinear term of its Taylor approximation. We note that the predicted correlation does not stand out strongly from the linear surrogate background, as it does for the Lorenz system. A possible explanation for this is that the Thomas system

TABLE I. In the first half of this table we show the real and predicted smallest nonlinearity μ for traditional chaotic system. In the second half we present the reconstructed smallest nonlinearity of financial systems.

System	μ_{real}	μ_{recon}
Lorenz	2	1.88
Classical Halvorsen	2	1.96
Thomas	3 ^a	2.92
MSCI World Index	—	3.12
S&P 500 Index	—	1.64
STOXX Europe 600 Index	—	5.32

^a The Thomas system has a sine nonlinearity. Here we consider the first nonlinear term of the Taylor expansion of the sine function, $x^3/3!$, as smallest nonlinearity.

contains only a small degree of nonlinearity. This can be seen in its largest Lyapunov exponent of $\lambda \approx 0.01$ is barely positive (compared to $\lambda \approx 0.9$ for the Lorenz system). Additionally, the terms in the Taylor approximation of the sine function scale with the factorial of the exponent, keeping the nonlinear effect apparently small. Nevertheless, it is interesting that the observed smallest nonlinearity seems to correspond with the first nonlinear term of the Taylor expansion. We have seen that for synthetic systems we are able to coarsely determine the smallest linearity present in data making us confident to apply this method to real world data.

For the financial data we perform our test on three different stock indices: the MSCI World Index tracking publicly traded large- and mid-cap companies across the developed world; the S&P 500 Index tracking the 500 largest, publicly traded, U.S. companies; and the STOXX Europe 600 Index covering 600 publicly traded, European companies spanning from small- to large-cap. For each index we use the daily closing value starting from 1st March 2005 up to 31st January 2025 for calculating the daily return (percentage change). This results in roughly 5000 data points for each index. For each index we use the same parametrization for the minimal RCs consisting of a block size of $b = 5$, a spectral radius of $\rho^* = 0.99$, and a regularization parameter of $\beta = 10^{-6}$, and we use the first 500 steps as synchronization steps.

The results are presented in Tab. I, where we can see that the test was successful for each index and we are able to determine a minimal nonlinearity for each. Here, we want to note that predicting the stock indices was obviously unsuccessful. However, it seems the unsuccessful prediction was enough to capture the nonlinearity in the data, as we observe a peak like in Fig. 9 for every index. We find it difficult to put our numbers into context, as similar studies have not been performed yet. The observations presented here illustrate the proof-of-concept nature of the method rather than providing conclusive evidence of underlying nonlinearities in financial data. Given the well-known stochasticity of financial data²⁷, further in-depth studies are recommended to confirm the reliability of these results. Nevertheless, we find it a fascinating result to be able to measure nonlinearity from a real dataset.

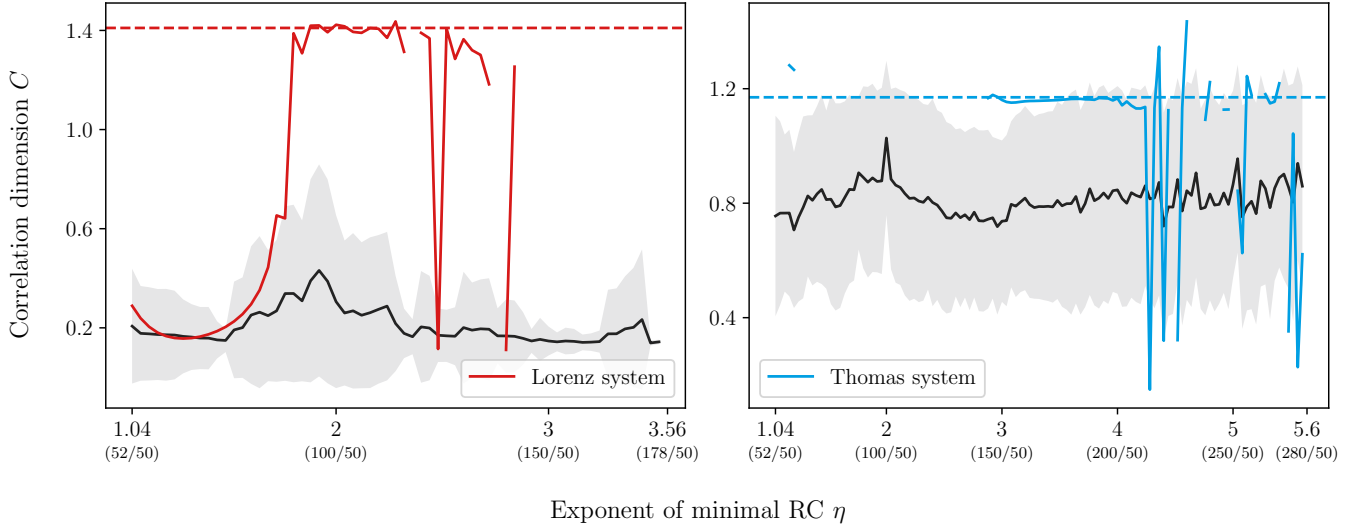


FIG. 8. The correlation dimension of the predictions using the reduced minimal RC with one linearity against the surrogate background for the Lorenz and Thomas system is shown. The black line and gray area represent the mean and one standard deviation of the correlation dimension when training the reduced minimal RC with FT surrogates. The dashed colored line shows the real correlation dimension of the data. Each plot is the result of 6375 optimizations.

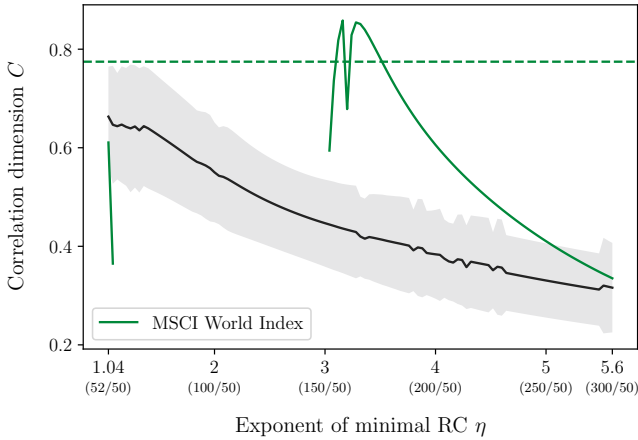


FIG. 9. The correlation dimension of the predictions of the MSCI World Index data using the reduced minimal RC model is shown. The black line with the gray area shows the mean and one standard deviation of the surrogate background. The dashed line represents the real correlation dimension of the MSCI World Index returns. This figure is a result of 6375 experiments.

B. Smart non-integer library

For the second application we want to transfer our findings discovered for minimal RCs to the traditional RC architecture. While generalizing reservoir states is not a new idea^{28,29}, we propose extending this idea with fractional powers. We have seen that minimal RCs work best, when the generalized reservoir state contains the nonlinearity present in data. For this reason we propose including fractional powers of the reservoir states in minimal RC. For determining the number of powers

to include, we revert to our results from Sec. IV B in Fig. 4: inspired by the full width at half maximum metric used a lot in optics, we calculate the full width at 75% from the peak performance in Fig. 4. While we only used a parameter of $a = 3.98$ for Fig. 4, we extend the study for Fig. 10 to include parameters of $a = 1.58$ and $a = 1.80$, in order to get a feeling for the width at lower powers. The results are shown in Fig. 10, where we see the width for 75% of the peak performance staying constant across all exponents and different parameters, indicating faintly that this result may be generalizable.

With this finding we can chose fractional exponents whose width will cover the whole space between two integers. One possible realization of these exponents can be constructed with

$$\tilde{r}_{[1,2]} = \left(\underline{r}^1 \quad \underline{r}^{\frac{54}{50}} \quad \underline{r}^{\frac{66}{50}} \quad \underline{r}^{\frac{78}{50}} \quad \underline{r}^{\frac{90}{50}} \quad \underline{r}^2 \right)^T. \quad (15)$$

Here the subscript $[1, 2]$ indicates that the fractions simply represent the spacing between the integer powers 1 and 2, and this idea can be generalized up to an arbitrary integer power. In later application we extend them to span up to an integer power of 3. We find our results to be robust against the exact choice of fractional powers. While we acknowledge this being a rather rudimentary approach with room to improvement for finding the optimal fractional powers, we find this approach to be sufficient for the scope of this work.

We want to test whether the findings for minimal RCs can be transferred to work on traditional RCs. We perform this test by predicting the Lorenz system using three different architectures: firstly, we use a RC with a dimensionality of $d = 100$. The new architecture also uses a dimensionality of $d = 100$ but generalizes the reservoir states $\underline{r}$ to $\tilde{r}$ by including fractional powers of the reservoir state up to a power of 3 with the spacing of Eq. 15. However, since including these additional

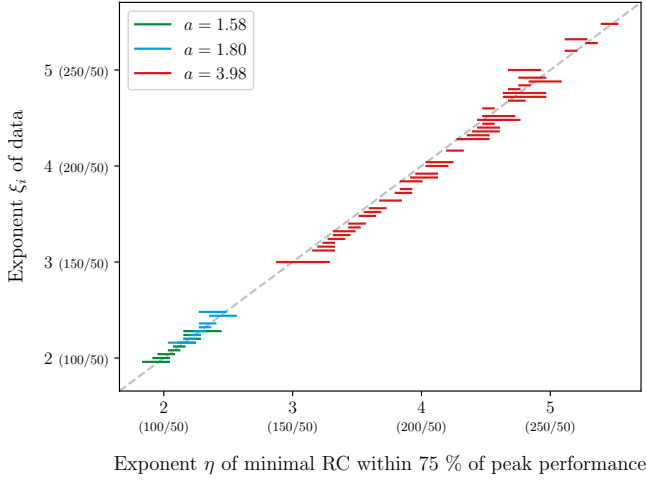


FIG. 10. For each exponent ξ_i of the fractional Halvorsen system, we show the range of exponents for minimal RCs, where the relative forecast horizon lies between 75% of their maximal value. For this plot we compiled three different parameters a of the fractional Halvorsen system. The gray line indicates the identity where $\eta = \xi_i$. We compile the results of in total 86070 separate experiments.

powers increases the size of the output matrix by a factor of 10, we need to test our proposed change against a reservoir resulting in the same sized output matrix. This results in a third RC model of with a dimensionality of $d = 1100$. For all three models we use the same hyperparameters: a spectral radius of $\rho^* = 0.2$ on a random network and a regularization parameter of $\beta = 10^{-4}$. We train all models on 4000 data points and use 1000 data points for the synchronization phase.

While we see in Fig. 11 that the small reservoir with fractional reservoir states performs worse than the large reservoir, it easily outperforms the small reservoir. Therefore, we can improve the performance of small reservoirs by generalizing their reservoir states to include fractional powers. This approach can be used to enhance the performance of physical RC implementations in situations where increasing the reservoir size is not feasible due to constraints in the hardware fabrication process, such as the limited number of neurons available on neuromorphic chips or increasing production costs associated with larger physical reservoirs³⁰.

VI. CONCLUSION

In this work, we systematically explored the relationship between the nonlinearities in input data and those introduced in RC models. Building on the minimal RC framework, we proposed a tailored architecture with a single, tunable nonlinearity parameter, allowing us to isolate and precisely control the degree of nonlinearity in the model. Using a novel fractional Halvorsen system, we generated chaotic datasets with adjustable nonlinear structure and evaluated the prediction performance across a wide range of reservoir nonlinearities.

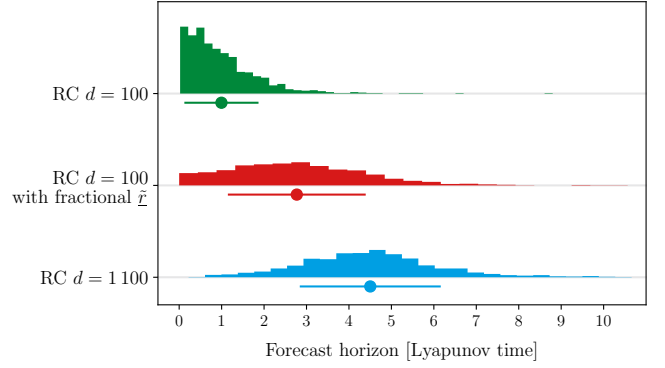


FIG. 11. We show the prediction performance for three different RCs: green represents the small reservoir, red the small reservoir with fractional reservoir states, and blue the large reservoir. For each RC we performed 1000 experiments and show the distribution of the forecast horizons, and the mean with one standard deviation below.

While we restricted ourselves to even numerators to avoid imaginary-valued reservoir states, the idea of studying imaginary reservoir states presents a compelling direction for future research.

Through extensive experiments, we found that short-term forecasting performance is maximized when the nonlinearity in the reservoir model matches the nonlinearity of the input data. In other words, the best predictions occur at a tuned “nonlinearity match” between data and model, whereas mismatches, by using a reservoir that is too linear or too nonlinear relative to the data, consistently degrade performance. This confirms our core hypothesis and directly demonstrates that an optimally tailored reservoir yields superior results.

Importantly, we observed that for systems with multiple nonlinearities, it is often the smallest nonlinearity in the data that dominates prediction performance. This insight enables us to use our framework in reverse: by sweeping through reservoir nonlinearities and observing the resulting performance, we were able to estimate the minimal nonlinearity present in a given time series. Applying this method to both synthetic and real-world financial datasets, we demonstrated its practical utility in identifying underlying nonlinear structure.

Finally, we transferred our findings to classical RC architectures and demonstrated that incorporating fractional, generalized reservoir states leads to an improvement in predictive performance. This has direct implications for physical RC platforms, where increasing the number of reservoir nodes may not be feasible due to hardware or economic constraints. By enhancing the expressiveness of the reservoir through non-integer polynomial transformations—rather than scaling the system size—we enable a more compact yet powerful representation of the input dynamics. This approach offers a structured way to increase the abilities of physical reservoirs without increasing their structural complexity, making it a viable strategy for high-performance prediction in embedded or resource-constrained environments.

Our work offers both a theoretical and practical step for-

ward in understanding and designing reservoir computers that are better aligned with the complexity of the data they aim to model.

ACKNOWLEDGMENTS

We would like to thank Allianz Global Investors for providing data and computational resources.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Appendix A: Metrics

Here we want to define the methods used for quantifying the quality of a prediction. As in similar work, we use the forecast horizon for quantifying the short-term prediction power. The so-called long-term ‘climate’ of an attractor is measured by the largest Lyapunov exponent in combination with the correlation dimension.

1. Largest Lyapunov exponent

The Lyapunov exponent can be used to formalize the concept of the sensitive dependency on initial conditions of chaotic systems. Defining the distance δ between two nearby points $\underline{x}$ and $\underline{x} + \underline{\epsilon}$, experimentally an exponential increase of this distance δ can be observed for chaotic systems modeled by

$$\delta(t) = \delta(0) \exp \lambda t . \quad (\text{A1})$$

Here λ describes the largest Lyapunov exponent and is a measure for how fast two nearby trajectories diverge.

We calculate the largest Lyapunov exponent from data using the algorithm introduced by Rosenstein, Collins, and De Luca³¹. For this, we track the evolution of initially close points in phase space by identifying nearest neighbors in the time series and measuring the average logarithmic divergence over time. The slope of this divergence, computed over a selected time interval, yields an estimate of the maximal Lyapunov exponent. To ensure valid comparisons, pairs of points are filtered to avoid temporal proximity and to allow sufficient forecast length.

Using the largest Lyapunov exponent λ , a Lyapunov time $\tau_\lambda := \lambda^{-1}$ can be defined, representing the characteristic time scale over which trajectories in phase space remain close. This quantity acts as a natural reference time scale for analyzing and contrasting the dynamics of various systems.

2. Correlation dimension

The correlation dimension is a widely used measure to estimate the fractal dimensionality of strange attractors and provides insight into the geometric complexity of a system’s long-term behavior. It is based on the idea of quantifying how the number of point pairs within a certain distance r scales with r itself. The correlation sum $\mathcal{C}(r)$ is defined as the fraction of pairs whose mutual distance is smaller than r by

$$\mathcal{C}(r) = \lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{t_1 \neq t_2} \Theta(r - \|\underline{x}(t_1) - \underline{x}(t_2)\|) . \quad (\text{A2})$$

Here Θ is the Heaviside step function returning one when the distance is smaller than r and zero when the distance is bigger than r .

Experimentally it has been discovered that for self-similar, strange attractors the power law

$$\mathcal{C}(r) \sim r^C \quad (\text{A3})$$

holds over a region of r . The scaling factor of the power law C is the correlation dimension and basically describes how densely the points fill the space as the scale r decreases.

We calculate the correlation dimension using the algorithm by Grassberger and Procaccia³² by embedding the time series in phase space and estimating how the number of point pairs within a radius r scales with r . To do this efficiently we organize the data using a binary space-partitioning data structure that organizes the points in a k -dimensional space³³. Using this structure, we compute the correlation sum $\mathcal{C}(r)$ over a range of radii r . The slope of the double logarithmic plot of $\mathcal{C}(r)$ over r yields the correlation dimension C .

3. Forecast horizon

For quantifying the short-term prediction we use the forecast horizon as measure, which describes a time for which the error between the true trajectory $\underline{x}$ and the predicted trajectory $\underline{x}_{\text{pred}}$ is smaller than a threshold $\underline{\Delta}$ in each coordinate. For each coordinate we calculate the maximal time the error is below the threshold with

$$\underline{v} = \arg \max_t \{ |x_i(t) - x_{i,\text{pred}}(t)| < \Delta_i \} . \quad (\text{A4})$$

The $\arg \max$ function is applied element-wise. We use the standard deviation σ applied element-wise as threshold $\underline{\Delta}$ with $\underline{\Delta} = \sigma(\underline{x})$.

The forecast horizon v is then defined as the minimal time across all coordinates during which the prediction error stays below the threshold by

$$v = \min \underline{v} . \quad (\text{A5})$$

In order to compare the forecast horizon across different systems, we define it in multiple of Lyapunov times as

$$v_\lambda = \frac{v}{\tau_\lambda} = v \lambda . \quad (\text{A6})$$

- ¹H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks - with an Erratum note,” Tech. Rep. (GMD Forschungszentrum Informationstechnik, 2001).
- ²W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: a new framework for neural computation based on perturbations,” *Neural Comput.* **14**, 2531–2560 (2002).
- ³H. Jaeger and H. Haas, “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication,” *Science* **304**, 78–80 (2004).
- ⁴M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Comput. Sci. Rev.* **3**, 127–149 (2009).
- ⁵J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, “Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data,” *Chaos* **27**, 121102 (2017).
- ⁶Z. Lu, B. R. Hunt, and E. Ott, “Attractor reconstruction by machine learning,” *Chaos* **28**, 061104 (2018).
- ⁷S. Shahi, C. D. Marcotte, C. J. Herndon, F. H. Fenton, Y. Shiferaw, and E. M. Cherry, “Long-Time Prediction of Arrhythmic Cardiac Action Potentials Using Recurrent Neural Networks and Reservoir Computing,” *Front. Physiol.* **12**, 734178 (2021).
- ⁸K. Brucke, S. Schmitz, D. Köglmayr, S. Baur, C. R  th, , E. Ansari, and P. Klement, “Benchmarking reservoir computing for residential energy demand forecasting,” *Energy Build.* **314**, 114236 (2024).
- ⁹J. Herteux, C. R  th, G. Martini, A. Baha, K. Koupparis, I. Lauzana, and D. Piovani, “Forecasting trends in food security with real time data,” *Commun. Earth Environ.* **5**, 611 (2024).
- ¹⁰X. Li, Q. Zhu, C. Zhao, X. Duan, B. Zhao, X. Zhang, H. Ma, J. Sun, and W. Lin, “Higher-order Granger reservoir computing: simultaneously achieving scalable complex structures inference and accurate dynamics prediction,” *Nat. Comm.* **15**, 2506 (2024).
- ¹¹M. Mijalkov, L. Storm, B. Zufiria-Gerbol  s, D. Ver  b, Z. Xu, A. Canal-Garcia, J. Sun, Y.-W. Chang, H. Zhao, E. G  mez-Ruiz, M. Passaretti, S. Garcia-Ptacek, M. Kivipelto, P. Svenningsson, H. Zetterberg, H. Jacobs, K. L  dge, D. Brunner, B. Mehlig, G. Volpe, and J. B. Pereira, “Computational memory capacity predicts aging and cognitive decline,” *Nat. Comm.* **16**, 2748 (2025).
- ¹²T. L. Carroll and L. M. Pecora, “Network structure effects in reservoir computers,” *Chaos* **29**, 083130 (2019).
- ¹³A. Haluszczynski and C. R  th, “Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing,” *Chaos* **29**, 103143 (2019).
- ¹⁴D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, “Next generation reservoir computing,” *Nat. Comm.* **12**, 5564 (2021).
- ¹⁵W. A. S. Barbosa and D. J. Gauthier, “Learning spatiotemporal chaos using next-generation reservoir computing,” *Chaos* **32**, 093137 (2022).
- ¹⁶H. Ma, D. Prosperino, and C. R  th, “A novel approach to minimal reservoir computing,” *Sci. Rep.* **13**, 12970 (2023).
- ¹⁷R. Thomas, “Deterministic chaos seen in terms of feedback circuits: analysis, synthesis, “labyrinth chaos”,” *Int. J. Bifurc. Chaos* **9**, 1889–1905 (1999).
- ¹⁸E. N. Lorenz, “Deterministic Nonperiodic Flow,” *JAS* **20**, 130–141 (1963).
- ¹⁹J. C. Sprott, *Elegant Chaos* (World Scientific Publishing, 2010).
- ²⁰J. R. Dormand and P. J. Prince, “A family of embedded Runge–Kutta formulae,” *J. Comput. Appl. Math.* **6**, 19–26 (1980).
- ²¹S. Baur, T. Nakano, D. Duncan, F. Fischbach, A. Haluszczynski, M. Klatt, D. K  glmayr, H. Ma, D. Prosperino, and C. R  th, “SCAN: A versatile implementation of reservoir computing methods,” *Comput. Phys. Commun.* (submitted).
- ²²H. Ma, A. Haluszczynski, D. Prosperino, and C. R  th, “Identifying causality drivers and deriving governing equations of nonlinear complex systems,” *Chaos* **32**, 103128 (2022).
- ²³C. R  th, M. Gliozzi, I. E. Papadakis, and W. Brinkmann, “Revisiting Algorithms for Generating Surrogate Time Series,” *Phys. Rev. Lett.* **109**, 144101.
- ²⁴J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. Doyne Farmer, “Testing for nonlinearity in time series: the method of surrogate data,” *Physica D* **58**, 77–94.
- ²⁵A. E. Hoerl and R. W. Kennard, “Ridge Regression: Applications to Nonorthogonal Problems,” *Technometrics* **12**, 69–82 (1970).
- ²⁶H. Ma, D. Prosperino, A. Haluszczynski, and C. R  th, “Efficient forecasting of chaotic systems with block-diagonal and binary reservoir computing,” *Chaos* **33**, 063130 (2023).
- ²⁷E. F. Fama, “The Behavior of Stock-Market Prices,” *J. Bus.* **38**, 34–105 (1965).
- ²⁸J. Herteux and C. R  th, “Breaking symmetries of the reservoir equations in echo state networks,” *Chaos* **30**, 123142 (2020).
- ²⁹A. Ohkubo and M. Inubushi, “Reservoir computing with generalized read-out based on generalized synchronization,” *Sci. Rep.* **14**, 30918 (2024).
- ³⁰S. Stepney, “Physical reservoir computing: a tutorial,” *Nat. Comput.* **23**, 665–685 (2024).
- ³¹M. T. Rosenstein, J. J. Collins, and C. J. De Luca, “A practical method for calculating largest Lyapunov exponents from small data sets,” *Physica D* **65**, 117–134 (1993).
- ³²P. Grassberger and I. Procaccia, “Measuring the Strangeness of Strange Attractors,” *Physica D* **9**, 189–208 (1983).
- ³³S. Maneewongvatana and D. M. Mount, “Analysis of Approximate Nearest Neighbor Searching with Clustered Point Sets,” in *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 59, edited by M. H. Goldwasser, D. S. Johnson, and C. C. McGeoch (American Mathematical Society, 2002) pp. 105–123.