Hybridizing Traditional and Next-Generation Reservoir Computing to Accurately and Efficiently Forecast Dynamical Systems

R. Chepuri*
Department of Physics, University of Maryland, College Park, MD 20742

D. Amzalag*
University of Chicago, Chicago, IL 60637

T. Antonsen
Department of Physics, University of Maryland, College Park, MD 20742
Department of Electrical and Computer Engineering,
University of Maryland, College Park, MD 20742 and
Institute for Research in Electronics and Applied Physics (IREAP), College Park, MD 20742

M. Girvan†
Department of Physics, University of Maryland, College Park, MD 20742
Institute for Physical Science and Technology (IPST), College Park, MD 20742
Institute for Research in Electronics and Applied Physics (IREAP), College Park, MD 20742 and
Santa Fe Institute, Santa Fe, NM 87501

(Dated: March 29, 2024)

Reservoir computers (RCs) are powerful machine learning architectures for time series prediction. Recently, next generation reservoir computers (NGRCs) have been introduced, offering distinct advantages over RCs, such as reduced computational expense and lower data requirements. However, NGRCs have their own practical difficulties distinct from those of RCs, including sensitivity to sampling time and type of nonlinearities in the data. Here, we introduce a hybrid RC-NGRC approach for time series forecasting of complex and chaotic dynamical systems. We show that our hybrid approach can produce accurate short term predictions and capture the long term statistics of dynamical systems in situations where the RC and NGRC components alone are insufficient. The advantage of the hybrid RC-NGRC approach is most pronounced when both components are limited in their prediction capabilities, e.g. for a small RC and a large sampling time in the training data. Under these conditions, we show for several chaotic systems that the hybrid RC-NGRC method with a small reservoir ($N \approx 100$) can achieve prediction performance rivaling that of a pure RC with a much larger reservoir ($N \approx 1000$), illustrating that the hybrid approach offers significant gains in computational efficiency over traditional RCs while simultaneously addressing some of the limitations of NGRCs.

Predicting the behavior of a dynamical system over time poses a significant challenge, especially when dealing with chaotic or complex systems. Reservoir computing, a type of machine learning framework, has emerged as a promising solution for this task. It offers advantages over deep learning methods, particularly in terms of computational efficiency. However, harder prediction tasks generally require larger, more computationally expensive reservoir computers (RCs) containing numerous artificial neurons. To tackle this issue, researchers have introduced next-generation reservoir computers (NGRCs), which boast even greater computational efficiency. While NGRCs have shown remarkable performance across various scenarios, they sometimes struggle with tasks that traditional RCs handle easily. In this study, we propose a novel hybrid approach that leverages the strengths of both RCs and NGRCs. By combining a small, computationally efficient RC with an NGRC, our hybrid model is able to achieve the performance and flexibility of a large RC while still preserving a substantial portion of the efficiency advantages of an NGRC.

I. INTRODUCTION

Reservoir computing has emerged as a powerful machine learning architecture for forecasting dynamical systems [1–6]. In a reservoir computer (RC), a high-dimensional nonlinear system called the reservoir is used to learn the flow of a dynamical system, and subsequently make a forecast. RCs have been shown not only to achieve impressive short term forecast accuracy, notably in the difficult case of chaotic systems, but also to reproduce the statistical properties (i.e., capture the "climate") of the true system in the long term [5]. Though RCs are relatively effective, several drawbacks have been

* These authors contributed equally to this work
† girvan@umd.edu

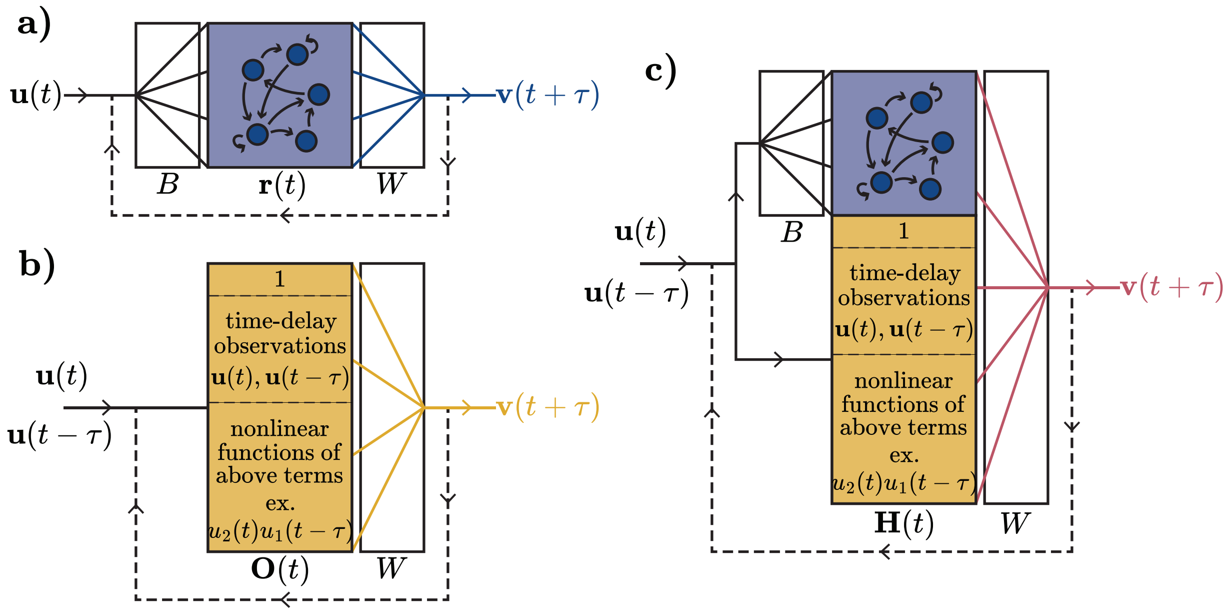

FIG. 1. **a)** Reservoir computer (RC) schematic. Time series observations $\mathbf{u}(t)$ are fed into a high-dimensional reservoir with state $\mathbf{r}(t)$ via an input matrix B, then an output matrix W is trained to predict the next data point in the series (i.e., at time $t+\tau$). Predictions at times $t > t_{\text{train}}$ are made by switching to autonomous mode in which outputs of the reservoir are repeatedly fed back in as input (dashed line). **b)** Next-generation reservoir computers (NGRCs) replace the reservoir with a nonlinear representation vector $\mathbf{O}(t)$ that is constructed using time-delayed observations. **c)** Our hybrid RC-NGRC prediction approach uses a hybrid representation vector $\mathbf{H}(t)$ that is the concatenation of a reservoir state with a NGRC representation vector in order to produce a prediction.

noted, such as the need to tune many hyperparameters for optimal performance [6].

An alternative to an RC, dubbed a next-generation reservoir computer (NGRC), has been introduced that avoids the use of a reservoir entirely [7]. Equivalent to a nonlinear vector autoregression machine [8], an NGRC replaces the reservoir with a representation vector that includes nonlinear functions of time-delayed observations of the dynamical system. NGRCs have been shown to be capable of forecasting several prototypical chaotic systems at greatly reduced computational cost compared to RCs, and are in fact mathematically equivalent to a variant of RCs with linear reservoir nodes and nonlinear readout [9]. However, implementations of NGRCs in practice show substantially different challenges from traditional RC implementations. For example, the presence of specific nonlinearities related to the true system in the NGRC representation vector has been shown to be essential for forecasting certain multistable dynamical systems [10]. We also find that NGRCs struggle to forecast prototypical chaotic systems when training data is sampled with a large time step (Section IV B 2).

In this paper, we introduce a hybrid RC-NGRC approach that leverages the strengths of both RCs and NGRCs. This approach uses a hybrid representation vector that is the concatenation of the reservoir state with an NGRC representation vector, echoing previous work hybridizing an RC with a knowledge-based model [11] or hybridizing an RC with another machine-learning-based prediction scheme [12]. We find that for some forecasting tasks that particular RC and NGRC implementations struggle with, our hybrid RC-NGRC approach can substantially outperform them at both short term accuracy and long term climate replication. In particular, we find that the hybrid approach most strongly outperforms RC and NGRC predictions when the reservoir size is small and the training data is sampled with a large time step. We show that, in cases where the NGRC is limited, a hybrid RC-NGRC approach using a small reservoir component reaches the performance level of a large traditional RC, while offering much greater computational efficiency.

In Section II, we review the use of RCs and NGRCs for time series prediction. In Section III, we present the details of our hybrid RC-NGRC approach. We show results applying this approach to forecasting the prototypical Lorenz system and other chaotic dynamical systems in Section IV. We conclude in Section V that a hybrid RC-NGRC approach is particularly useful in situations where computational resources are limited and standalone NGRCs struggle.

II. BACKGROUND: TRADITIONAL AND NEXT-GENERATION RESERVOIR COMPUTING

Suppose we have discrete time series data $\{\mathbf{u}(\tau), \mathbf{u}(2\tau), \ldots\}$ sampled at regular time steps from the trajectory $\mathbf{u}(t) \in \mathbb{R}^d$ of a d-dimensional dynamical system. Using the first n_{train} data points as training data, the goal of RC and NGRC forecasting is to produce a predicted trajectory $\mathbf{v}(t)$ for time $t > t_{\text{train}}$ (where $t_{\text{train}} = n_{\text{train}}\tau$) that is a good match to $\mathbf{u}(t)$. In addition to seeking a high quality short term forecast with $\mathbf{v}(t) \approx \mathbf{u}(t)$ for as long as possible after t_{train}, we also seek to replicate the system's climate, meaning that the long term statistical features of $\mathbf{v}$ match those of $\mathbf{u}$.

A. Reservoir computers (RCs)

A typical RC (Figure 1a)) uses a random artificial neural network with recurrent links of fixed weights as a reservoir. To define such a reservoir, we initialize a random directed unweighted network of N nodes with a Poisson degree distribution having average degree $\langle k \rangle$. We multiply the nonzero elements of the resulting Boolean adjacency matrix $\tilde{A} \in \mathbb{R}^{N\times N}$ by link weights chosen from the uniform distribution on $[-w, w]$ to form the matrix A, and then we rescale A to have spectral radius $\rho \lessapprox 1$.

At all time steps of the training data, the state of the reservoir $\mathbf{r}(t) \in \mathbb{R}^N$ depends on the input it receives, $\mathbf{u}(t)$, and its state at the previous time step, $\mathbf{r}(t-\tau)$, through the relationship

$$\mathbf{r}(t) = (1-\alpha)\mathbf{r}(t-\tau) + \alpha f(A\mathbf{r}(t-\tau) + B\mathbf{u}(t) + c), \quad (1)$$

where f is the hyperbolic tangent function (applied element-wise), and we choose the entries of the input matrix $B \in \mathbb{R}^{N\times d}$ from the uniform distribution on $[-\sigma, \sigma]$; α is a leakage parameter that controls the timescale of the reservoir's response to its input.

To train the RC, we first synchronize the reservoir by initializing it in the zero state ($\mathbf{r}(0) = 0$) and feeding in the first n_{warmup} data points (up to time $t_{\text{warmup}} = n_{\text{warmup}}\tau$) according to Equation 1. We then feed in the remaining $n_{\text{fit}} = n_{\text{train}} - n_{\text{warmup}}$ training data points (spanning a time $t_{\text{fit}} = n_{\text{fit}}\tau$), and train a readout matrix $W \in \mathbb{R}^{d\times N}$ to make a one-step-ahead prediction for this data:

$$W\mathbf{r}(t) \approx \mathbf{u}(t+\tau), \quad t_{\text{warmup}} < t < t_{\text{train}}. \quad (2)$$

We fit W using ridge regression (linear regression with Tikhonov regularization), which minimizes the quantity

$$\sum_{t_{\text{warmup}}<t<t_{\text{train}}} \left(\|W\mathbf{r}(t) - \mathbf{u}(t+\tau)\|^2 \right) + \beta \operatorname{Tr}\left(WW^T\right). \quad (3)$$

Here, β is the regularization hyperparameter that penalizes large entries of W to prevent overfitting.

We note that in practice, we use the input noise technique of adding weak Gaussian noise (standard deviation $\gamma \ll 1$) to $\mathbf{u}$ before feeding it in to the reservoir via Equation 1, but using noiseless data to fit W, as this has been shown to promote climate stability of autonomous predictions [13] (see the Supplementary Materials).

After training the readout matrix W, we switch the reservoir to autonomous mode (Figure 1a)) in which the output of the reservoir $W\mathbf{r}(t)$ is repeatedly fed back in as input:

$$\mathbf{v}(t) = W\mathbf{r}(t-\tau) \quad (4)$$
$$\mathbf{r}(t) = (1-\alpha)\mathbf{r}(t-\tau) + \alpha f(A\mathbf{r}(t-\tau) + B\mathbf{v}(t) + \mathbf{c}) \quad (5)$$

for $t = t_{\text{train}} + \tau, t_{\text{train}} + 2\tau, \ldots$. The autonomous forecast for the trajectory of the dynamical system is then $\mathbf{v}(t)$.

Reservoir computing offers a state-of-the-art method for time series forecasting of dynamical systems, capable of both short term forecast accuracy and long term climate replication. Compared to deep neural networks, RCs have dramatically reduced training time because only the output weights are fit [14]. However, training and/or simulating a large reservoir can still be computationally expensive in some cases: computational costs of fitting an output matrix scale approximately as $\mathcal{O}(N^3)$ in typical RC implementations, and well-performing RCs may require large N on the order of 1000 or more. Furthermore, there are many hyperparameters (listed in Table I) that must be tuned, making RCs nontrival to implement in many cases.

B. Next-Generation Reservoir Computers (NGRCs)

In contrast to RCs, NGRCs utilize a representation vector constructed directly from the training data $\mathbf{u}(t)$ in order to make predictions (Figure 1b)). First, we must specify the number k of current and time-delayed observations in the representation vector, and the number s of time steps between successive time-delayed observations. In our studies we focus on $k = 2$ and $s = 1$. Then, at each time step of the training data we construct the representation vector $\mathbf{O}(t)$ as follows:

- We concatenate the current and time-delayed observations to form a linear representation vector:

$$\mathbf{O}_{\text{lin}}(t) = \mathbf{u}(t) \oplus \mathbf{u}(t-s\tau) \oplus \ldots \oplus \mathbf{u}(t-(k-1)s\tau), \quad (6)$$

  where $\oplus$ represents vector concatenation. As in the RC case, in practice we add weak Gaussian noise with standard deviation $\gamma \ll 1$ to $\mathbf{u}$ before forming $\mathbf{O}_{\text{lin}}$ (see the Supplementary Materials).

- We form a nonlinear representation vector $\mathbf{O}_{\text{nonlin}}(t)$ consisting of nonlinear functions of the elements of $\mathbf{O}_{\text{lin}}(t)$. Here, we choose to form $\mathbf{O}_{\text{nonlin}}(t)$ by listing all unique quadratic monomials of the linear terms, e.g. $u_2(t)u_1(t-s\tau)$.

- We form the full representation vector by concatenating a constant element (taken to be 1), the linear representation vector, and the nonlinear representation vector:

$$\mathbf{O}(t) = 1 \oplus \mathbf{O}_{\text{lin}}(t) \oplus \mathbf{O}_{\text{nonlin}}(t) \quad (7)$$

Note that $\mathbf{O}(t)$ is defined for $s(k-1)\tau < t \le t_{\text{train}}$; as such, $s(k-1)\tau$ is the effective warm up time of the NGRC. At each time step, there are $1+dk+dk(dk+1)/2$ elements of the representation vector $\mathbf{O}(t)$, where d is the dimension of $\mathbf{u}$ (e.g. 28 elements for $k=2, d=3$).

Next, $\mathbf{O}(t)$ plays an analogous role to $\mathbf{r}(t)$ in an RC. We fit a readout matrix W using ridge regression (see Equation 3) to satisfy

$$W\mathbf{O}(t) \approx \mathbf{u}(t+\tau), \quad s(k-1)\tau < t < t_{\text{train}}. \tag{8}$$

Then, we use autonomous mode to make a prediction for $t = t_{\text{train}} + \tau, t_{\text{train}} + 2\tau, \ldots$:

- We make a one-step prediction: $\mathbf{v}(t) = W\mathbf{O}(t-\tau)$.
- We construct $\mathbf{O}(t)$ according to Equation 7. In doing so, we draw time-delayed terms from t_{train} and before from $\mathbf{u}$, and draw those from after t_{train} from $\mathbf{v}$.

The NGRC forecast for the trajectory of the dynamical system is then $\mathbf{v}(t)$.

NGRCs have strong predictive ability and have some important advantages: compared to RCs, NGRCs are more computationally efficient due to having many fewer terms in their representation vector, require less hyperparameter tuning, and have a very small effective warm up time of $s(k-1)\tau$ [7]. However, NGRCs have their own drawbacks: performance can be very dependent on the choice of nonlinear functions used to construct $\mathbf{O}_{\text{nonlin}}$, in some cases showing poor performance if the specific nonlinearities of the true dynamical system are not reflected in the representation vector [10]. We will also show later in Section IV B 2 that NGRCs struggle when the training data are sampled sparsely from the true system, i.e. the time step τ is large.

III. METHODS: HYBRID RC-NGRC FORECASTING APPROACH

We now introduce the central innovation of our paper: a hybrid RC-NGRC scheme for forecasting dynamical systems. In the hybrid RC-NGRC scheme, we utilize both a small reservoir and an NGRC representation vector to make a forecast (Figure 1c)). First, we initialize a reservoir just as in Section II A. In practice we usually use a lightweight reservoir with a small number of nodes $N \le 100$. Then, we use training data to construct both a reservoir state $\mathbf{r}(t)$ and an NGRC representation vector $\mathbf{O}(t)$ for all possible time steps, as specified in Equations 1 and 6, 7. We form a hybrid representation vector

$$\mathbf{H}(t) = \mathbf{r}(t) \oplus \mathbf{O}(t) \tag{9}$$

at each time step, where $\oplus$ represents concatenation of vectors. Then we fit W using ridge regression (see Equation 3) to best satisfy

$$W\mathbf{H}(t) \approx \mathbf{u}(t+\tau), \quad t_{\text{warmup}} < t < t_{\text{train}}. \tag{10}$$

TABLE I. Details of the training data and hyperparameters of the RCs and NGRCs used to make forecasts. The hybrid RC-NGRC uses the same hyperparameters. These values are used throughout, except where otherwise noted.

Training data	Time step	$\tau = 0.06$
	# training data points	$n_{\text{train}} = 10,000$
	Noise standard deviation	$\gamma = 1 \times 10^{-3}$
Reservoir	Number of nodes	$N = 100$
	Average degree	$\langle k \rangle = 10$
	Link weight scaling	$w = 1$
	Spectral radius	$\rho = 0.99$
	Leakage rate	$\alpha = 1$
	Bias	$c = 0$
	Input matrix scaling	$\sigma = 1$
	Warm-up time steps	$n_{\text{warmup}} = 1000$
	Regularization parameter	$\beta = 1 \times 10^{-3}$
NGRC	Number of current and time-delayed observations	$k = 2$
	Spacing of time-delayed observations	$s = 1$
	Regularization parameter	$\beta = 1 \times 10^{-3}$

After training the readout matrix W, we produce an autonomous prediction $\mathbf{v}$ by iterating the equations below:

$$\mathbf{v}(t) = W\mathbf{H}(t-\tau) \tag{11}$$
$$\mathbf{r}(t) = (1-\alpha)\mathbf{r}(t-\tau) + \alpha f(A\mathbf{r}(t-\tau) + B\mathbf{v}(t) + \mathbf{c}) \tag{12}$$
$$\mathbf{O}(t) = 1 \oplus \mathbf{O}_{\text{lin}}(t) \oplus \mathbf{O}_{\text{nonlin}}(t) \tag{13}$$
$$\mathbf{H}(t) = \mathbf{r}(t) \oplus \mathbf{O}(t). \tag{14}$$

(see Section II B for more detail on constructing $\mathbf{O}$).

In practice, to create hybrid RC-NGRC, RC, and NGRC predictions of a given chaotic dynamical system with an attractor, we use the following procedure. We sample a random initial condition from the attractor, integrate the system forward using the fourth order Runge-Kutta method with an integration time step $\tau_{\text{int}} \ll \tau$, and subsample with time step τ to obtain the time series data $\mathbf{u}(t)$. We normalize $\mathbf{u}$ so that each component of the training data (first n_{train} data points) has mean 0 and standard deviation 1. We generate a random realization of the reservoir as described in Section II A, using the hyperparameters given in Table I. Then, we construct autonomous predictions of length t_{predict} using the RC, NGRC, and hybrid RC-NGRC (Sections II A, II B, and III), using the input noise technique (see Supplementary Materials). For multiple trials, we repeat the whole procedure above, so that different trials have both different initial conditions of the trajectory and different random reservoir realizations.

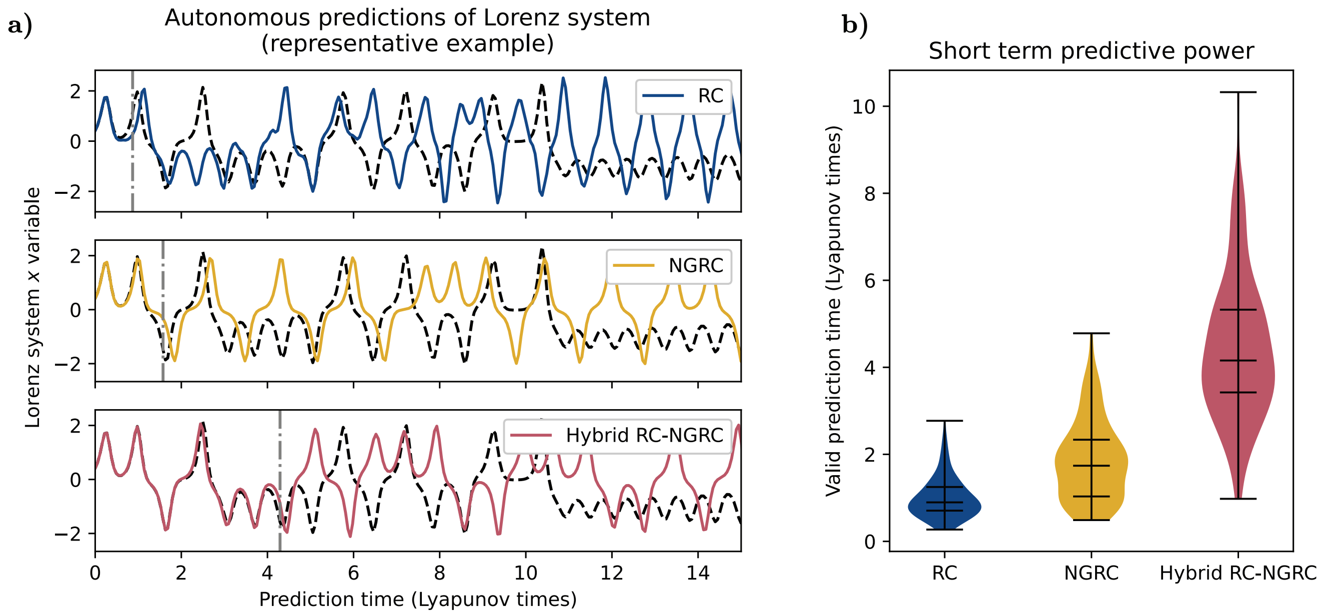

FIG. 2. **a)** Representative examples of RC, NGRC, and hybrid RC-NGRC autonomous predictions of the Lorenz system (x component shown), where RC and NGRC performance is limited by small reservoir size and large data time step. Valid prediction time (VPT) indicated by the vertical dashed line. **b)** Distributions of VPTs for RC, NGRC, and RC-NGRC predictions, where each trial is done on new initial conditions using a new reservoir realization. The hybrid RC-NGRC shows substantially stronger short term predictive power than either the RC or NGRC alone. Horizontal lines: quartiles (100 trials).

IV. RESULTS

A. Forecasting the Lorenz system with a small reservoir and large time step

We now evaluate the hybrid RC-NGRC approach on the task of predicting the Lorenz system, a prototypical chaotic dynamical system governed by the equations

$$\dot{x} = 10(y - x),\ \dot{y} = x(28 - z) - y,\ \dot{z} = xy - 8z/3 \quad (15)$$

[15]. We compare to the standalone RC and NGRC components, with hyperparameters as listed in Table I. Although both RC and NGRC approaches are capable of forecasting the Lorenz system under ideal conditions, here we impose additional constraints on the prediction methods. We limit the size of the reservoir, imitating a scenario in which large reservoirs are not desirable due to computational constraints, and we use training data that is sampled from the Lorenz system at a large time step, imitating a scenario in which observing the state of the dynamical system can only be done sparsely. These constraints make the task of forecasting the Lorenz system formidable for both RCs and NGRCs.

1. *Short term forecast quality*

To evaluate the quality of a prediction in the short term, we use valid prediction time (VPT) as a metric, defined as the time at which the root mean square error of the normalized prediction exceeds a threshold, here chosen to be $\kappa = 0.9$:

$$t_{\mathrm{VPT}} = \min\left\{ t \,:\, \frac{\|\mathbf{v}(t) - \mathbf{u}(t)\|}{\sqrt{\langle \|\mathbf{u}(t)\|^2 \rangle}} > \kappa \right\} - t_{\mathrm{train}}. \quad (16)$$

In chaotic systems such as the Lorenz system, errors grow approximately as $\exp(\Lambda_{\max} t)$ where $\Lambda_{\max}$ is the maximal Lyapunov exponent. Thus, the Lyapunov time $t_{\mathrm{lyap}} = \Lambda_{\max}^{-1}$ is a natural timescale for evaluating the quality of forecast of a chaotic dynamical system, and we report t_{VPT} in units of t_{lyap}.

Representative examples of predictions $\mathbf{v}(t)$ of the Lorenz system by RC, NGRC, and hybrid RC-NGRC prediction schemes are shown in Figure 2a) with VPTs marked (only the x coordinates of the predictions are shown). In this example, the VPT of the hybrid RC-NGRC forecast is much longer than for the RC or NGRC alone. Recording VPTs over many trials with different Lorenz system initial conditions and different random reservoirs yields a distribution of VPTs shown in Figure 2b). We find that hybrid RC-NGRC has much better short term predictive power than either the RC or NGRC, as evidenced by the longer median valid prediction times (VPTs) (hybrid: $4.16 t_{\mathrm{lyap}}$, RC: $0.90 t_{\mathrm{lyap}}$, NGRC: $1.74 t_{\mathrm{lyap}}$). We observe similar results without input noise (see Supplementary Materials).

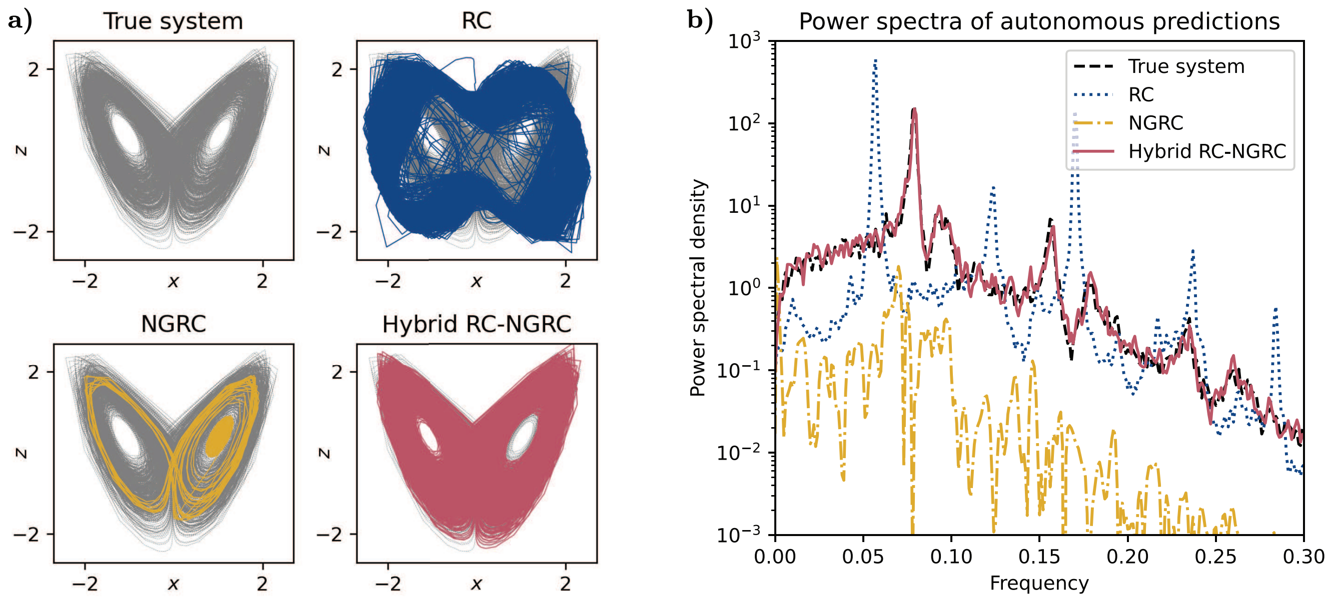

FIG. 3. **a)** Representative examples of long-term phase space trajectories of the RC, NGRC, and hybrid RC-NGRC autonomous predictions (predictions extended from Figure 2a)). Though RC and NGRC reconstruct the Lorenz attractor in some trials, only the hybrid RC-NGRC prediction reliably reconstructs the attractor of the true system across trials. **b)** Power spectra of z component of autonomous predictions for the different methods. Only the hybrid RC-NGRC prediction reliably reproduces the spectrum of the Lorenz system.

2. *Long term climate replication*

We also find that the hybrid RC-NGRC more accurately reproduce the climate (long-term statistical properties) of the Lorenz system when compared to either the RC or NGRC alone. Two-dimensional projections of the phase space trajectories of the representative predictions from Figure 2a) are plotted in Figure 3a). In this example, the RC forecast completely fails to recreate the butterfly-shaped attractor. The NGRC prediction initially tracks the true attractor, but then gets trapped orbiting inward toward one of the unstable fixed points of the true system. Although in some trials the RC and NGRC predictions can recreate the true attractor, they often fail in a similar manner as in 3a). The climate reproduction of especially the NGRC is even worse when not using the input noise technique (see the Supplementary Materials). In contrast, the hybrid RC-NGRC approach robustly succeeds at accurately reconstructing the butterfly-shaped strange attractor of the Lorenz system.

We also examine the power spectral densities of the different methods' forecasts to determine whether they recover the climate of the Lorenz system. Figure 3b) shows the power spectra of the z components of the same representative predictions, found using Welch's method [16]. The power spectrum of the hybrid RC-NGRC prediction matches that of the Lorenz system nearly perfectly, suggesting the prediction captures the statistical properties of the Lorenz system. In contrast, the spectra of the RC and NGRC predictions fail to capture the features of the true system's spectrum.

Our results on the Lorenz system suggest that in cases when both the RC and NGRC are limited, e.g. for a small reservoir and large sampling time step in the training data, the hybrid RC-NGRC method offers a significant improvement over either the RC or NGRC alone.

B. Under what conditions is the hybrid RC-NGRC approach most advantageous?

Here, we relax the constraints from the previous section that the reservoir be small and the training data be sampled with large time step. We find that although the hybrid RC-NGRC still achieves good predictive performance, it loses its relative advantage over RC and/or NGRC approaches. This occurs as we enter a regime where either the RC or NGRC perform very well, thus eliminating the need for a hybrid approach. Our results suggest that the greatest utility of the hybrid RC-NGRC comes when the NGRC is limited (e.g. because the training data is sampled at a large time step) and computational efficiency is a priority (making small reservoirs highly advantageous).

1. *Small reservoir*

The predictive power of an RC is predicated on having a high-dimensional reservoir with enough fitting parame-

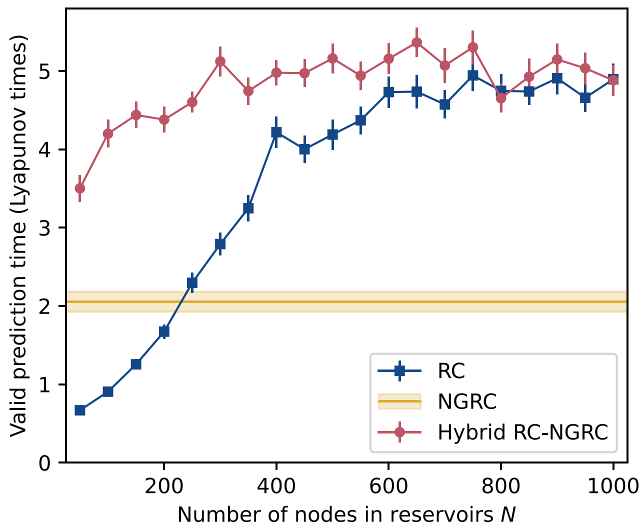

FIG. 4. Mean valid prediction times for the Lorenz system versus number of nodes in the reservoir. Although RC performance is poor at small N, and NGRC performance is modest due to using a large timestep ($\tau = 0.06$), the hybrid RC-NGRC performs well throughout, specifically providing a substantial advantage over both RC and NGRC at small N. Note that the hybrid RC-NGRC approach with reservoir size $N = 100$ approximately matches that of a pure RC with $N = 500$. Timestep $\tau = 0.06$. Error bars and band: standard error of the mean (64 trials).

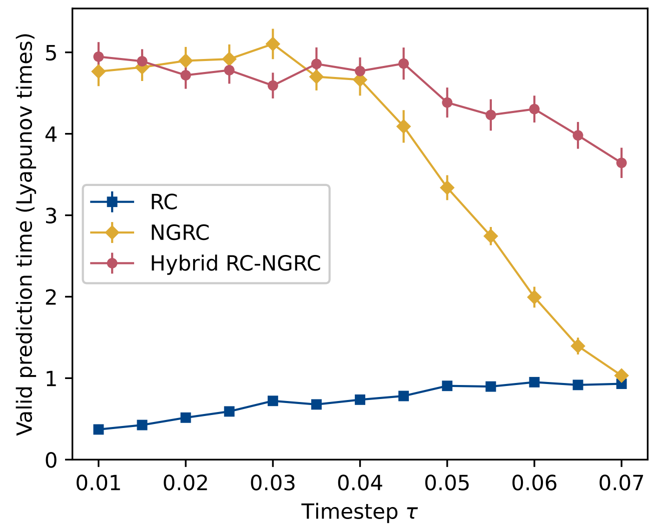

FIG. 5. Mean valid prediction times for the Lorenz system versus time step size τ in the training data. As time step is adjusted, the number of training data points n_{train} is kept constant. The hybrid RC-NGRC shows the greatest advantage in predictive power over the RC or NGRC alone when using a large time step. Reservoir size $N = 100$. Error bars: standard error of the mean (64 trials).

ters to accurately capture the behavior of the dynamical system. In practice, as the number of reservoir nodes N is increased, the forecasting skill of the RC increases, until some saturation point, as reflected in Figure 4. However, Figure 4 also shows that the hybrid RC-NGRC forecasts achieve a similar mean valid prediction time using a much smaller reservoir (compare hybrids with 100 nodes to RCs with 500 nodes). This is true even though the NGRC itself is poorly performing (due to large time step). Hybridizing an RC with even a poorly performing NGRC enables strong predictive ability even with a very small reservoir.

2. *Large time step/sparsely sampled training data*

While the NGRC approach has been shown to work in a range of cases, its most accurate predictions are achieved when the specific nonlinearities of the underlying system appear in the NGRC represenation vector [10]. In this case, during training the NGRC can learn weights that make the autonomous mode imitate a numerical integrator of the true system. However, just as numerical integration methods can fail if the integration time step is too large, the NGRC can also fail if the time step is too large. At large time steps, single time step increments of the training data are not as well approximated by the difference equation versions of the true differential equations governing the system. In Figure 5, we plot the valid prediction times for the NGRC as as a function of time step length τ. Note that as τ is varied, the number of training data time steps n_{train} is kept constant, so for larger τ, $t_{\text{train}} = n_{\text{train}}\tau$ is greater. We see that the valid prediction time of NGRC predictions decrease as τ is increased. We see a modest increase in the VPT for the RC as τ is increased, because the number of training data time steps is kept constant, the information content in the training signal initially increases as τ increases from a small value.

Compared with the NGRC, the drop off of the hybrid RC-NGRC's valid prediction time as time step is increased is much less dramatic. Although the RC alone shows weak predictive power of about one or less Lyapunov times at all time steps, at large time steps the hybrid's valid prediction time is much greater than either the RC or NGRC alone. When using data with a large time step, hybridizing even a poorly performing RC with an NGRC can drastically improve the prediction performance.

C. How performance depends on the size of the training dataset

NGRCs are touted as requiring shorter training datasets than RCs, owing to their very short effective warm up time of $s(k-1)\tau$ (just a single time step for

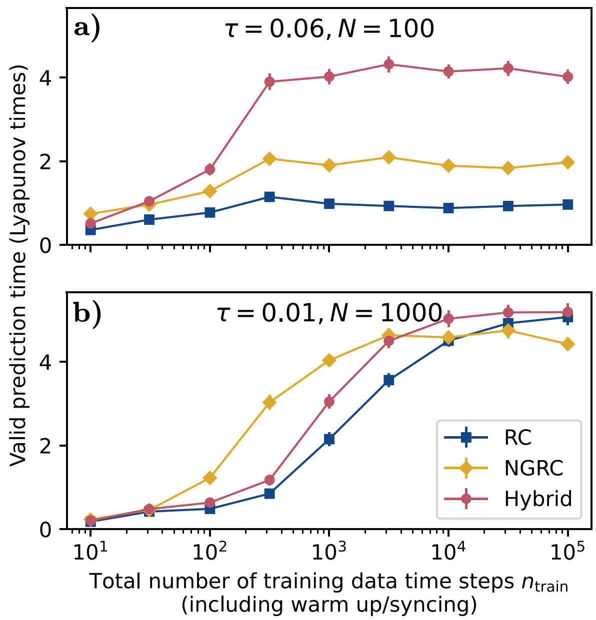

FIG. 6. Mean valid prediction times for the Lorenz system versus number of training data points, in scenarios where **a)** RC and NGRC struggle, and **b)** RC and NGRC perform well with enough training data. Error bars (where visible): standard error of the mean (64 trials).

$k = 2$, $s = 1$) [7]. In contrast, RCs must have enough warm up time t_{warmup} at the beginning of training to synchronize the reservoir to the input data. How does a hybrid RC-NGRC, which also requires a warm up time for its reservoir, compare when training on a limited number of time steps?

To investigate this question, we vary the total number of time steps n_{train} of Lorenz system training data supplied to the RC, NGRC, and hybrid RC-NGRC, and plot the mean VPTs of each model. We perform this analysis a) under the constraints of small reservoir and large time step from Section IV A ($\tau = 0.06$, $N = 100$), and b) in an easier scenario with large reservoir and small time step ($\tau = 0.01$, $N = 1000$). In all trials, to choose a warm up time for the RC, we first initialize two copies of the reservoir that are identical except for their initial reservoir states $\mathbf{r}^{(1)}(0) \neq \mathbf{r}^{(2)}(0)$, and feed the same input data into both reservoirs. We take the inverse slope of a linear fit of $\ln\left(|\mathbf{r}^{(1)}(t) - \mathbf{r}^{(2)}(t)|\right)$ vs. t as the empirical characteristic time t_{sync} for reservoir synchronization, then use a warm up time $t_{\text{warmup}} = 10t_{\text{sync}}$ (capped to a maximum of $t_{\text{train}}/4$). After t_{warmup} the synchronization error in the reservoir state will be on the order of $e^{-10} \approx 5 \times 10^{-5}$. In practice, we find $t_{\text{warmup}} \approx 20\tau$ for these hyperparameters. (Note that if we used smaller leakage rates α, we would expect the warm up time to be longer.) For the hybrid RC-NGRC, we use the same t_{warmup} as for RC. For the NGRC, the effective warm up length is always only $s(k-1)\tau$.

In the scenario of Section IV A where both RC and NGRC struggle due to small reservoir and large time step (Figure 6a)), we find that the hybrid RC-NGRC maintains it relative advantage over NGRCs and RCs over a wide range of training data amounts. Despite the NGRC having a much shorter warm up time of 1 time step versus the hybrid's ~ 20, using the hybrid RC-NGRC yields improved performance even when using training data amounts down to ~ 100 steps.

However, when a well-performing NGRC is available, the hybrid RC-NGRC loses out to NGRC in the low training data regime. In Figure 6b), we plot VPT versus training data amount in the scenario where both RC and NGRC perform well due to large reservoir and small time step. The hybrid displays only marginal performance benefit over the standalone RC or NGRC when using large amounts of training data, consistent with the results of Section IV B. However, at lower training data amounts the data efficiency of the NGRC is evident, performing much better than the RC and hybrid for $n_{\text{train}} \sim 10^2$ to $\sim 10^3$.

In summary, the hybrid RC-NGRC approach can offer substantial improvements in predictive performance when training on a limited amount of data, but is not beneficial if a well-performing NGRC is available.

D. Hybrid RC-NGRC test results on other chaotic systems

We now test the ability of the hybrid RC-NGRC approach to forecast a few other prototypical chaotic systems [18], focusing as before on scenarios where both RC and NGRC performance are limited due to small reservoir size and large data sampling time step. We find that across several systems, hybridizing the weakly performing RCs and NGRCs yields substantial benefit.

Despite the Rössler system [19] having a relatively simple flow with only one quadratic nonlinear term in its governing differential equations, we find that NGRCs struggle somewhat with prediction in the case of a large time step $\tau = 0.1$ as shown in Figure 7a). A small RC with a $N = 200$ node reservoir performs even worse, providing almost no predictive power on its own. However, hybridizing even this very poorly performing RC with the NGRC yields substantial improvements in short term accuracy over both the RC and NGRC alone. In fact, the hybrid RC-NGRC performance is exceeds the performance of a much larger RC with $N = 1000$ nodes.

The double-scroll electronic circuit introduced in reference [17] has governing differential equations that contain hyperbolic sine terms not represented in the NGRC representation vector. Reference [7] showed that NGRCs with cubic terms in the representation vector are nonetheless well-suited to forecast this system. However, without the insight to change the terms of the representation vector, NGRCs with quadratic terms in the representation vector have a difficult time predicting this system. This difficulty is heightened by a large time step of

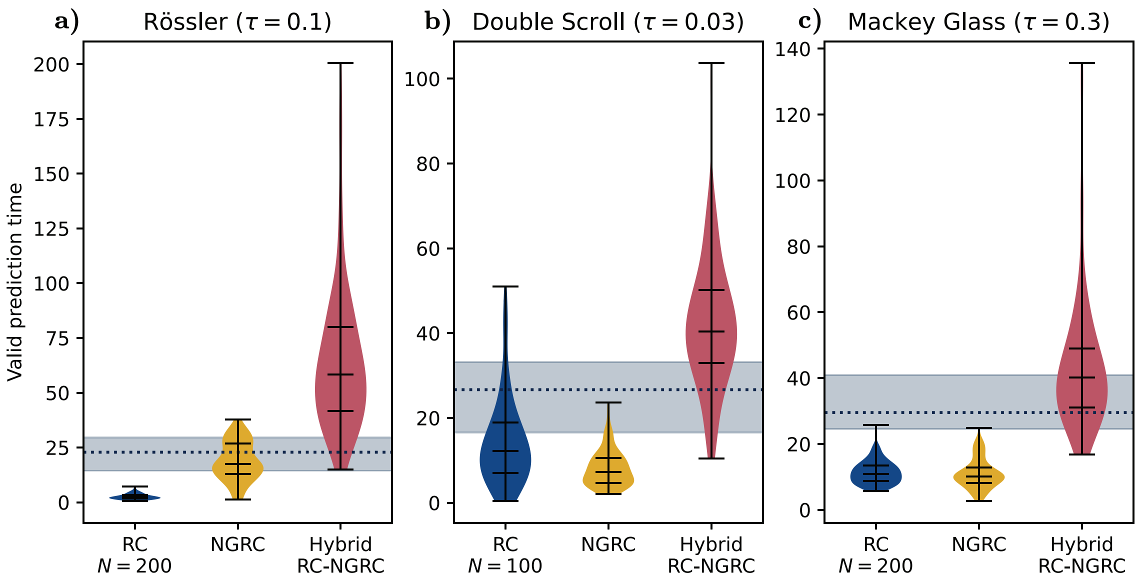

FIG. 7. Distributions of valid prediction times of RC, NGRC, and RC-NGRC predictions on the **a)** Rössler, **b)** Double Scroll [17], and **c)** Mackey-Glass systems. For the Mackey-Glass system, use we time-delayed observations spaced $s = 6$ time steps apart to create the NGRC and hybrid representation vectors, approximately matching the time-delay term of the true system. Dotted blue line and band: median and interquartile range of VPTs from a large RC with $N = 1000$ nodes. Horizontal lines in violins: quartiles (100 trials).

$\tau = 0.03$, as shown in Figure 7b). However, hybridizing to a small RC with $N = 100$ nodes yields performance far surpassing the performance of the RC or NGRC itself, and even surpassing the performance of a RC with the same hyperparameters but with $N = 1000$ nodes.

The Mackey-Glass system [20] presents another distinct challenge, as the governing differential equations include a time-delay term: the flow of the system at time t depends not just on $\mathbf{u}(t)$ but also $\mathbf{u}(t - T)$, where here $T = 2$. Large RCs are capable of forecasting the Mackey-Glass system, though Figure 7c) shows that a small RC of $N = 200$ nodes performs only modestly. An NGRC using only an observation delayed by a single time step ($s = 1$) produces very poor forecasts of the Mackey-Glass system, as it lacks information about the time-delay term of the true system. Extending the look back time to $s = 6$ to approximate time time-delay term of the true system, as we have done in 7c), yields moderate results. However, the hybrid RC-NGRC outperforms both, and approximately matches the short term predictive ability of a RC of $N = 1000$ nodes.

We emphasize that blind application of the hybrid RC-NGRC approach to new dynamical systems may not in general give better performance than RC or NGRC alone. The hybrid approach may be particularly beneficial when the NGRC is limited, e.g. by a large sampling time step, and when one desires the computational efficiency of a small RC which by itself does not offer strong performance. However, the meaning of small reservoir and large time step may differ for different systems. Additionally, to avoid numerical divergence and achieve climate replication of a new system, the input noise strength, regularization, and other hyperparameters may need to be tuned to the specific dynamical system in question.

V. CONCLUSION

We have introduced a hybrid RC-NGRC method for time series forecasting of chaotic dynamical systems. The hybrid method can make predictions that are both accurate in the short term and capture the system climate in the long term, even when the RC and NGRC components alone cannot. In our studies, we see that the hybrid RC-NGRC method holds the greatest advantage over its components when using a small reservoir and sparsely sampled training data. For other cases in which the NGRC is limited due to reasons beside large time step (e.g., when the NGRC representation vector does not capture the important nonlinearities), we expect our hybrid method to offer similar benefits at a lower computational cost than a large RC. In summary, we believe that the hybrid RC-NGRC scheme is an important step toward significantly reducing the computational load of reservoir computing, as achieved by an NGRC, while still maintaining the robustness of a traditional RC.

VI. SUPPLEMENTARY MATERIAL

Supplementary material about the technique of adding input noise can be found in a separate file.

VII. ACKNOWLEDGEMENTS

We would like to thank Andrew Pomerance, Declan Norton, and Brian Hunt for their useful discussions about this manuscript. DA's contribution to this research was supported by NSF award PHY-2150399. MG and RC's contributions were supported by ONR award number N000142212656.

VIII. AUTHOR DECLARATIONS

All authors have no conflicts to disclose.

Ravi Chepuri: writing - original draft (lead); implementation/methodology (equal); visualization (equal), data analysis (supporting) **Dael Amzalag:** data analysis (lead); visualization (equal); implementation/methodology (equal) **Thomas Antonsen:** conceptualization and design (supporting); supervision (supporting) **Michelle Girvan:** conceptualization and design (lead); supervision (lead); editing (lead).

IX. DATA AVAILABILITY STATEMENT

The data in this study can be generated by running the publicly available code (see the code availability statement).

X. CODE AVAILABILITY STATEMENT

All code is available under an MIT License on Github (`https://github.com/ravi-chepuri/hybrid_RC_NGRC`).

[1] H. Jaeger, in *GMD - Ger. Natl. Res. Inst. Comput. Sci.*, Vol. 148 (2001).
[2] W. Maass, T. Natschläger, and H. Markram, Neural Computation **14**, 2531 (2002).
[3] M. Lukoševičius and H. Jaeger, Computer Science Review **3**, 127 (2009).
[4] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Phys. Rev. Lett. **120**, 024102 (2018).
[5] Z. Lu, B. R. Hunt, and E. Ott, Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 061104 (2018).
[6] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Neural Networks **115**, 100 (2019).
[7] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, Nat Commun **12**, 5564 (2021).
[8] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis* (Springer, Berlin, Heidelberg, 2005).
[9] E. Bollt, Chaos **31**, 013108 (2021).
[10] Y. Zhang and S. P. Cornelius, Phys. Rev. Res. **5**, 033213 (2023).
[11] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, Chaos **28**, 041101 (2018).
[12] F. Köster, D. Patel, A. Wikner, L. Jaurigue, and K. Lüdge, Chaos: An Interdisciplinary Journal of Nonlinear Science **33**, 073109 (2023).
[13] A. Wikner, J. Harvey, M. Girvan, B. R. Hunt, A. Pomerance, T. Antonsen, and E. Ott, Neural Networks **170**, 94 (2024).
[14] P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, Neural Networks **126**, 191 (2020).
[15] E. N. Lorenz, J. Atmospheric Sci. **20**, 130 (1963).
[16] P. Welch, IEEE Trans. Audio Electroacoustics **15**, 70 (1967).
[17] A. Chang, J. C. Bienfang, G. M. Hall, J. R. Gardner, and D. J. Gauthier, Chaos: An Interdisciplinary Journal of Nonlinear Science **8**, 782 (1998).
[18] W. Gilpin, Phys. Rev. Res. **5**, 043252 (2023).
[19] O. E. Rössler, Physics Letters A **57**, 397 (1976).
[20] M. C. Mackey and L. Glass, Science **197**, 287 (1977).

Supplementary Material: Hybridizing Traditional and Next-Generation Reservoir Computing to Accurately and Efficiently Forecast Dynamical Systems

R. Chepuri,* D. Amzalag,* T. Antonsen, and M. Girvan†

(Dated: March 29, 2024)

* These authors contributed equally to this work

† girvan@umd.edu

I. EFFECTS OF THE INPUT NOISE TECHNIQUE

In the main text, we add small-amplitude noise to the training data when feeding it into the RC, NGRC, and hybrid RC-NGRC (we still use noiseless data as training targets when fitting the output matrix). This input noise technique has been shown to promote climate stability of forecasts by mapping small perturbations off the attractor during autonomous prediction back onto the attractor [1]. Here, we report that the central results of the paper still hold when using noiseless input data, though the input noise technique does confer some useful benefits for climate replication.

We find that reproducing all short term performance results in the main manuscript with no input noise (noise standard deviation $\gamma = 0$) yields qualitatively similar results. For example, the plot of VPT for predicting the Lorenz system vs. training data time step in Figure S1a) shows the same qualitative behavior as Figure 5 of the main text (reshown in Figure S1b) for ease of comparison). The key result that the hybrid RC-NGRC has better short term predictive performance than RC or NGRC alone, specifically at large time steps, is unchanged.

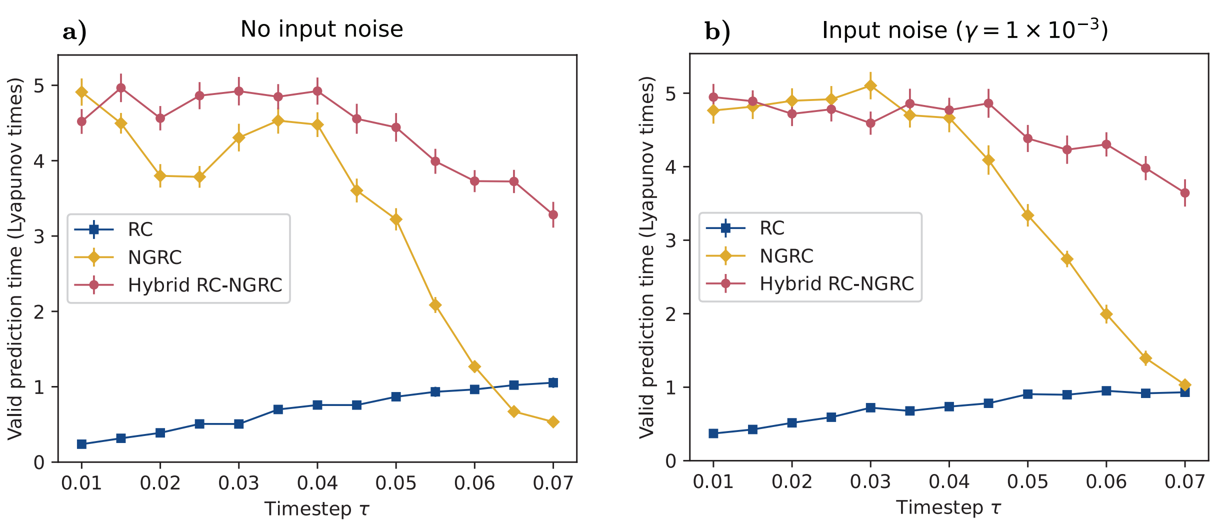

FIG. S1. Mean valid prediction times for the Lorenz system versus time step size τ in the training data, with **a)** no input noise used and **b)** input noise standard deviation $\gamma = 1 \times 10^{-3}$ used. Regardless of noise, the hybrid RC-NGRC shows the greatest advantage in predictive power over the RC or NGRC alone when using a large time step. Reservoir size $N = 100$. Error bars: standard error of the mean (64 trials). Figure b) is repeated from Figure 5 of the main text.

However, without the input noise technique, the climate replication abilities of the autonomous predictions are much worse, especially those of the NGRC. Figure S2a) shows a representative example of an NGRC autonomous prediction with no noise (time step $\tau = 0.06$). The NGRC prediction quickly limits to a fixed point, failing completely to capture the climate; similar failures are observed in most trials across many initial conditions. The failure to capture climate is much more severe than when using the input noise technique as shown in Figure S2b), where the NGRC predictions track the true attractor for a longer time. If it occurs quickly enough, the sudden convergence of NGRC predictions to a fixed point can harm short term predictive power, for example contributing to a decreased mean VPT at small time steps in Figure S1. Besidses the NGRC predictions, the hybrid RC-NGRC predictions also sometimes fail to capture the Lorenz attractor, in some trials limiting to a fixed point or limit cycle, in contrast with the main text where the hybrid RC-NGRC with input noise always captured the Lorenz system climate.

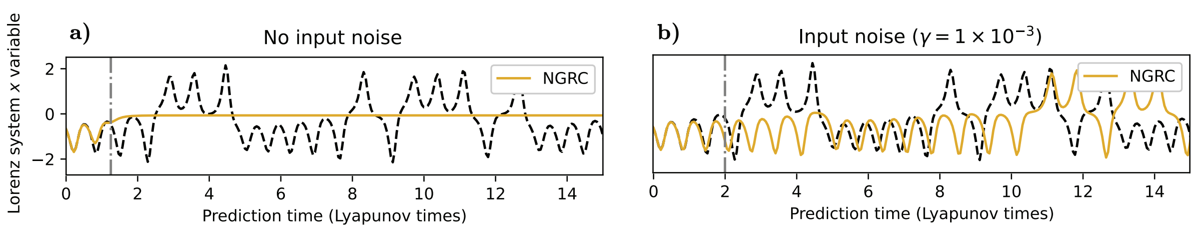

FIG. S2. Representative examples of autonomous NGRC predictions of the Lorenz system with large time step $\tau = 0.06$ with **a)** no input noise used and **b)** input noise standard deviation $\gamma = 1 \times 10^{-3}$ used (shown: x component).

Ref. [1] demonstrated that the input noise technique can stabilize autonomous predictions by preventing them from diverging numerically. Our findings suggest that in NGRCs, input noise is also helpful to prevent predictions from converging to a fixed point, especially when using relatively large time steps ($\tau = 0.06$).

Outside of the benefits for climate replication, the input noise technique is also slightly beneficial for short term predictive performance of RC and NGRC. Figure S3 shows that both RC and NGRC have optimal nonzero noise strengths for short term predictive performance. However, the hybrid RC-NGRC does not have as clear a trend, so long as the noise strength is not too large as to destroy predictive performance.

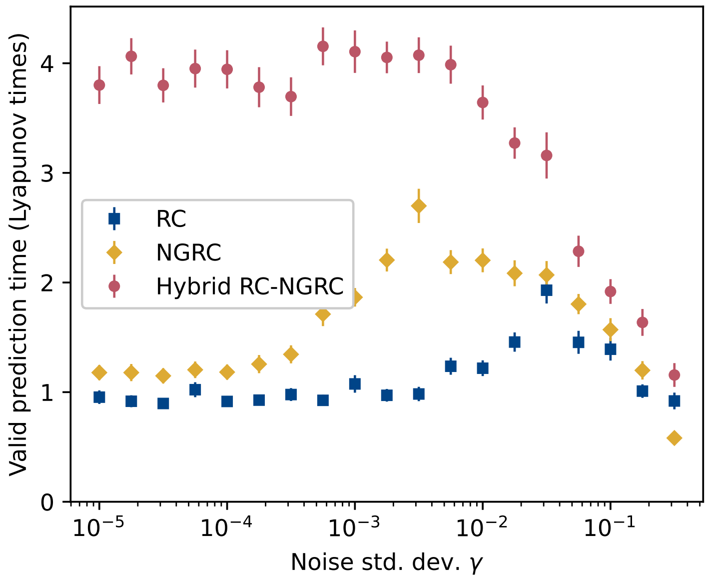

FIG. S3. Mean valid prediction times for the Lorenz system versus input noise standard deviation γ. RC and NGRC have an optimal noise level for short term prediction, but the trend is not as clear for the hybrid RC-NGRC. Error bars: standard error of the mean (64 trials).

We caution that we have not investigated the joint effect of modifying both the input noise strength and the Tikhnonov regularization parameter. The input noise technique is in effect an alternative regularization scheme for fitting the output matrix [1], and as such the interplay of Tikhonov regression and the input noise technique may have nontrivial effects on the climate stabilization of autonomous forecasts. Results using different values of the Tikhonov regression parameter β and the noise variance γ may yield different results than shown here.

Note that the input noise technique is distinct from simply adding observational noise to the training data, as the train targets are still taken to be non-noisy values. The input noise technique is also distinct from using dynamical noise in the underlying system.

[1] A. Wikner, J. Harvey, M. Girvan, B. R. Hunt, A. Pomerance, T. Antonsen, and E. Ott, Neural Networks **170**, 94 (2024).