



**HAL**  
open science

# Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing

Johannes Nokkala, Rodrigo Martínez-Peña, Gian Luca Giorgi, Valentina Parigi, Miguel C Soriano, Roberta Zambrini

## ► To cite this version:

Johannes Nokkala, Rodrigo Martínez-Peña, Gian Luca Giorgi, Valentina Parigi, Miguel C Soriano, et al.. Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing. *Communications Physics*, 2021, 4 (1), 10.1038/s42005-021-00556-w . hal-03186332

**HAL Id: hal-03186332**





**<https://hal.sorbonne-universite.fr/hal-03186332>**

Submitted on 31 Mar 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing

Johannes Nokkala <sup>1</sup>✉, Rodrigo Martínez-Peña<sup>1</sup>, Gian Luca Giorgi<sup>1</sup>, Valentina Parigi <sup>2</sup>, Miguel C. Soriano <sup>1</sup> & Roberta Zambrini <sup>1</sup>✉

Quantum reservoir computing aims at harnessing the rich dynamics of quantum systems for machine-learning purposes. It can be used for online time series processing while having a remarkably low training cost. Here, we establish the potential of continuous-variable Gaussian states of linear dynamical systems for quantum reservoir computing. We prove that Gaussian resources are enough for universal reservoir computing. We find that encoding the input into Gaussian states is both a source and a means to tune the nonlinearity of the overall input-output map. We further show that the full potential of the proposed model can be reached by encoding to quantum fluctuations, such as squeezed vacuum, instead of classical fields or thermal fluctuations. Our results introduce a research paradigm for reservoir computing harnessing quantum systems and engineered Gaussian quantum states.

<sup>1</sup>Instituto de Física Interdisciplinar y Sistemas Complejos (IFISC, UIB-CSIC), Campus Universitat de les Illes Balears, Palma de Mallorca, Spain. <sup>2</sup>Laboratoire Kastler Brossel, Sorbonne Université, CNRS, ENS-Université PSL, Collège de France, 4 place Jussieu, Paris F-75252, France. ✉email: [johannes@ifisc.uib-csic.es](mailto:johannes@ifisc.uib-csic.es); [roberta@ifisc.uib-csic.es](mailto:roberta@ifisc.uib-csic.es)

Machine learning (ML) covers a wide range of algorithms and modeling tools with automated data-processing capabilities based on experience<sup>1,2</sup>. ML, with the prominent example of neural networks, has proven successful for tackling practical processing tasks that are unsuitable for conventional computer algorithms<sup>3–7</sup>. With the deployment of ML algorithms, their limitations and inefficiencies when running on top of conventional computing hardware arise both in terms of power consumption and computing speed<sup>8</sup>. The demand for an increased efficiency is currently fueling the field of unconventional computing, which aims at developing hardware and algorithms that go beyond the traditional von Neumann architectures<sup>9–11</sup>. Recent extensions of neural networks and other ML techniques based on quantum systems<sup>12–16</sup> aim to offer and identify novel capabilities<sup>17–23</sup>. In this context, reservoir computing (RC) is a machine-learning paradigm that is amenable to unconventional hardware-based approaches in the classical domain, e.g., in photonics<sup>24–30</sup> and spintronics<sup>31,32</sup>.

RC exploits the dynamics of a nonlinear system—the reservoir—for information processing of time-dependent inputs<sup>33,34</sup>. RC has its roots in the discovery that in recurrent neural networks, i.e., neural networks with an internal state, it is sufficient to only train the connections leading to the final output layer, avoiding optimization difficulties well-known in neural networks, without any apparent loss in computational power<sup>35,36</sup>. In fact, restricting the training to only the final layer does not necessarily restrict the scope of tasks that can be done; indeed, universality has been shown for distinct classes of RC such as so-called liquid state machines<sup>37</sup> and echo-state networks<sup>38</sup>. Unlike feed-forward neural networks, said to be universal when they can approximate any continuous function<sup>39,40</sup>, RC realizes transformations between sequences of data and is said to be universal when it can approximate any so called fading memory function<sup>37,38</sup>, which can be thought of as a continuous function of a finite number of past inputs. In practice, reservoir computers have achieved state-of-the-art performance in tasks such as continuous speech recognition<sup>41</sup> and nonlinear time series prediction<sup>42</sup> thanks to their intrinsic memory<sup>43</sup>.

RC has the potential to be extended to the quantum regime<sup>22,44–55</sup>. Indeed, since the reservoir requires neither fine tuning nor precise engineering such extensions are good candidates for NISQ (noisy intermediate-scale quantum) technologies<sup>56</sup>. Continuous-variable quantum systems amenable to an optical implementation would have the additional benefit of intrinsic resilience to decoherence even at room temperature and allow to easily read-out a portion of the optical signal for output processing, including (direct or indirect) measurements. Previous proposals for neural networks realized with continuous-variable quantum systems have either used non-Gaussian gates<sup>19</sup> or nonlinear oscillators<sup>52</sup> to go beyond linear transformations of the modes. Such nonlinearity is required, e.g., for universal quantum computing in continuous-variable systems<sup>57</sup>, i.e., for the ability to reproduce any unitary transformation<sup>57,58</sup>, however, non-Gaussian gates in particular are an unsolved engineering challenge so far<sup>59–62</sup>.

Here, we put forward the framework for RC with Gaussian states of continuous-variable quantum systems, in bosonic reservoirs given by random harmonic networks. The input is encoded into single-mode Gaussian states injected into the network by periodic state resets, whereas the output is a trained function of the network observables. It is worth noticing that the encoding is in general nonlinear, providing nonlinearity for the overall input-output map. The linear network then provides an analog system—not based on gates or circuits—where different encodings may be exploited. We explore the utility of the method spanning across classical and quantum states and our results are

applicable to several physical reservoirs. As elaborated in the Discussion, the restriction to Gaussian dynamics of linear oscillators brings the model within reach of state-of-the-art optical<sup>63–66</sup> or superconducting and optomechanical platforms<sup>67–70</sup>, however, one might expect it to have limited information processing capabilities. We find that even universal RC, i.e., approximation of any fading memory functions, is possible. Versatility is provided by the ability to tune information processing from fully linear to highly nonlinear with the input encoding. Therefore, even a fixed network can adapt to the requirements of the task at hand. Finally, we reveal that encoding to field fluctuations, or covariances, instead of in field intensities, or first moments, can significantly increase the system's information processing capacity (IPC)<sup>71</sup>, a task-independent quantifier of RC capability. However, encoding to classical (thermal) fluctuations does not provide any nonlinearity. To solve the problem, we take a step towards exploiting the quantumness of the system and propose to harness the quantum fluctuations provided by squeezed vacuum instead. Indeed, this achieves simultaneously increased capacity, nonlinearity and universal RC.

## Results

**The model.** We consider a linear network of  $N$  interacting quantum harmonic oscillators as detailed in Methods. The scheme for using this network for RC is shown in Fig. 1. The input sequence  $\mathbf{s} = \{\dots, \mathbf{s}_{k-1}, \mathbf{s}_k, \mathbf{s}_{k+1}, \dots\}$ , where  $\mathbf{s}_k \in \mathbb{R}^n$  represents each input vector and  $k \in \mathbb{Z}$ , is injected into the network by resetting at each timestep  $k$  the state of one of the oscillators, called ancilla (A), accordingly. As often done in RC, each input vector can be mapped to a single scalar in the ancilla through a function of the scalar product  $\mathbf{v}^\top \cdot \mathbf{s}_k$  where  $\mathbf{v} \in \mathbb{R}^n$ . The rest of the network acts as the reservoir (R), and output is taken to be a function  $h$  of the reservoir observables before each input.

To express the dynamics, let  $\mathbf{x}^\top = \{q_1, p_1, q_2, p_2, \dots\}$  be the vector of network momentum and position operators and let  $\mathbf{x}_k$  be the form of this vector at timestep  $k$ , after input  $\mathbf{s}_k$  has been processed. We may take the ancilla to be the  $N$ th oscillator without a loss of generality. Let the time between inputs be  $\Delta t$ . Now operator vector  $\mathbf{x}_k$  is related to  $\mathbf{x}_{k-1}$  by

$$\mathbf{x}_k = \mathbf{S}(\Delta t)(P_R \mathbf{x}_{k-1} \oplus \mathbf{x}_k^A), \quad (1)$$

where  $P_R$  drops the ancillary operators from  $\mathbf{x}_{k-1}$  (reservoir projector, orthogonal to the ancilla vector) and  $\mathbf{x}_k^A$  is the vector of ancillary operators conditioned on input  $\mathbf{s}_k$ , while  $\mathbf{S}(\Delta t) \in \text{Sp}(2N, \mathbb{R})$  is the symplectic matrix induced by the Hamiltonian in Eq. (7) (see Methods) and time  $\Delta t$  (for more details on this formalism see, e.g., refs. 72–74). The dynamics of reservoir operators  $\mathbf{x}_k^R = P_R \mathbf{x}_k$  is conveniently described dividing  $\mathbf{S}(\Delta t)$  into blocks as

$$\mathbf{S}(\Delta t) = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}, \quad (2)$$

where  $\mathbf{A}$  is  $2(N-1) \times 2(N-1)$  and  $\mathbf{D}$  is  $2 \times 2$ . Now the formal definition of the proposed reservoir computer reads

$$\begin{cases} \mathbf{x}_k^R = \mathbf{A} \mathbf{x}_{k-1}^R + \mathbf{B} \mathbf{x}_k^A, \\ o_k = h(\mathbf{x}_k^R), \end{cases} \quad (3)$$

where  $h$  maps the reservoir operators to elements of the real output sequence  $\mathbf{o} = \{\dots, o_{k-1}, o_k, o_{k+1}, \dots\}$ .

For Gaussian states, the full dynamics of the system conditioned by the sequential input injection is entailed in the first moments vector  $\langle \mathbf{x}_k^R \rangle$  and covariance matrix  $\sigma(\mathbf{x}_k^R)$ . The values at step 0, given a sequence of previous  $m$  inputs  $\mathbf{s} = \{\mathbf{s}_{-m+1}, \dots, \mathbf{s}_{-1}, \mathbf{s}_0\}$  encoded in the corresponding ancilla vectors, is

obtained through repeated application of Eq. (3) and reads

$$\begin{cases} \sigma(\mathbf{x}_0^R) = \mathbf{A}^m \sigma(\mathbf{x}_{-m}^R) (\mathbf{A}^\top)^m \\ \quad + \sum_{j=0}^{m-1} \mathbf{A}^j \mathbf{B} \sigma(\mathbf{x}_{-j}^A) \mathbf{B}^\top (\mathbf{A}^\top)^j, \\ \langle \mathbf{x}_0^R \rangle = \mathbf{A}^m \langle \mathbf{x}_{-m}^R \rangle + \sum_{j=0}^{m-1} \mathbf{A}^j \mathbf{B} \langle \mathbf{x}_{-j}^A \rangle, \end{cases} \quad (4)$$

where  $\sigma(\mathbf{x}_{-m}^R)$  and  $\langle \mathbf{x}_{-m}^R \rangle$  are the initial conditions, i.e., the initial state of the reservoir. This is the Gaussian channel for the reservoir conditioned on the input encoded in  $\mathbf{x}^A$ . Different Gaussian states of the ancilla can be addressed, such as coherent states, squeezed vacuum or thermal states (see Fig. 1), respectively characterized by the complex displacement  $\alpha$ , squeezing degree  $r$  and phase  $\varphi$ , and thermal excitations  $n_{th}$  (see Methods, Eq. (9)). Finally, the output is taken to be either linear or polynomial in either the elements of  $\sigma(\mathbf{x}_k^R)$  or  $\langle \mathbf{x}_k^R \rangle$ .

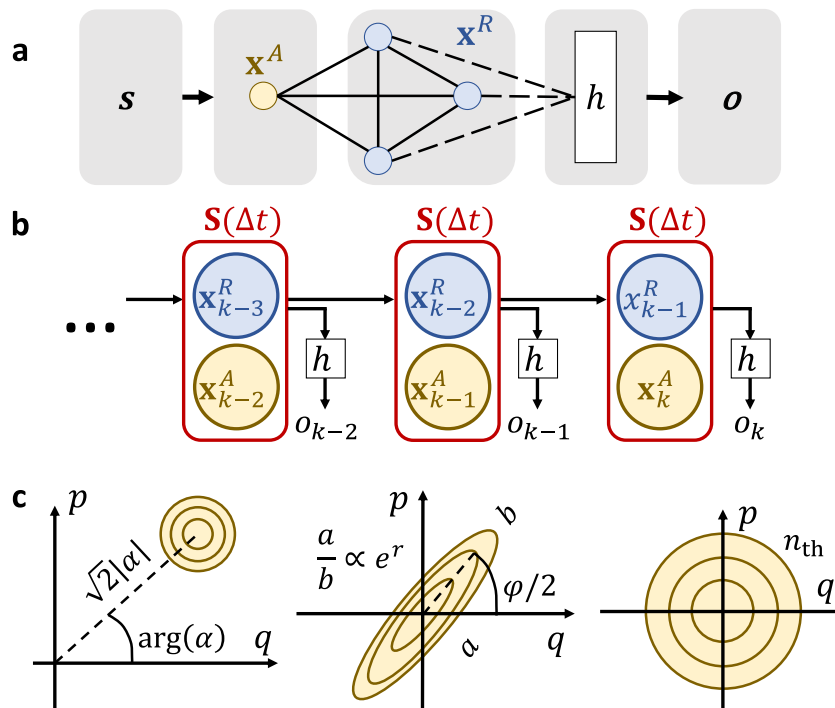
Observables can be extracted with Gaussian measurements such as homodyne or heterodyne detection. We will next show that the introduced model not only satisfies the requirements for RC, but is universal for RC even when restricting to specific input encodings.

**Universality for reservoir computing.** To begin with, we show that instances of the model defined by Eq. (3) and the dependency of  $\mathbf{x}_k^A$  on  $\mathbf{s}_k$  can be used for RC, i.e., the dynamics conditioned by the input can be used for online time series processing by

adjusting the coefficients of the polynomial defined by  $h$  to get the desired output.

As explained in Methods (Reservoir computing theory), the goal is more formally to reproduce a time-dependent function  $f(t) = F\{\dots, \mathbf{s}_{t-2}, \mathbf{s}_{t-1}, \mathbf{s}_t\}$ , associated with a given input  $\mathbf{s}$  and functional  $F$  from the space of inputs to reals. Consequently, we say that the system can be used for RC if there is a functional from the space of inputs to reals that is both a solution of Eq. (3) and sufficiently well-behaved to facilitate learning of different tasks. These two requirements are addressed by the echo-state property (ESP)<sup>43</sup> and the fading memory property (FMP)<sup>75</sup>, respectively. The associated functions are called fading memory functions. In essence, a reservoir has ESP if and only if it realizes a fixed map from the input space to reservoir state space—unchanged by the reservoir initial conditions—while FMP means that to get similar outputs it is enough to use inputs similar in recent past—which provides, e.g., robustness to small past changes in input. The two are closely related and in particular both of them imply that the reservoir state will eventually become completely determined by the input history; in other words forgetting the initial state is a necessary condition for ESP and FMP.

Looking at Eq. (4), it is readily seen that the model will become independent of the initial conditions at the limit  $m \rightarrow \infty$  of a left infinite input sequence if and only if  $\rho(\mathbf{A}) < 1$ , where  $\rho(\mathbf{A})$  is the spectral radius of matrix  $\mathbf{A}$ . Therefore,  $\rho(\mathbf{A}) < 1$  is a necessary condition for having ESP and FMP. The following lemma (proven in Supplementary Note 2) states that it is also sufficient



**Fig. 1 Reservoir computing scheme.** **a** The overall input-output map. The input sequence  $\mathbf{s}$  is mapped to a sequence of ancillary single-mode Gaussian states. These states are injected one by one into a suitable fixed quantum harmonic oscillator network by sequentially resetting the state of the oscillator chosen as the ancilla,  $\mathbf{x}^A$ . The rest of the network—taken to be the reservoir—has operators  $\mathbf{x}^R$ . Network dynamics maps the ancillary states into reservoir states, which are mapped to elements of the output sequence  $\mathbf{o}$  by a trained function  $h$  of reservoir observables. Only the readout is trained whereas the interactions between the network oscillators remains fixed, which is indicated by dashed and solid lines, respectively. **b** The corresponding circuit. The reservoir interacts with each ancillary state through a symplectic matrix  $\mathbf{S}(\Delta t)$  induced by the network Hamiltonian  $H$  during constant interaction time  $\Delta t$ . Output ( $o_k$ ) at timestep  $k$  is extracted before each new input.  $\mathbf{x}_k^A$  are the ancillary operators conditioned on input  $\mathbf{s}_k$  and  $\mathbf{x}_k^R$  are the reservoir operators after processing this input. **c** Wigner quasiprobability distribution of ancilla encoding states in phase space of ancilla position and momentum operators  $q$  and  $p$ . Here, the contours of the distribution are indicated by dark yellow lines. Input may be encoded in coherent states using amplitude  $|\alpha|$  and phase  $\arg(\alpha)$  of displacement  $\alpha \in \mathbb{C}$ , or in squeezed states using squeezing parameter  $r$  and phase of squeezing  $\varphi$  (where  $a$  and  $b$  are the length and height of an arbitrary contour), or in thermal states using thermal excitations  $n_{th}$ .

when we introduce the mild constraint of working with uniformly bounded subsets of the full input space, i.e., there is a constant that upper bounds  $\|s_k\|$  for all  $k$  in the past.

**Lemma 1** *Suppose the input sequence  $s$  is uniformly bounded. Let ancilla parameters be continuous in input and let  $h$  be a polynomial of the elements of  $\sigma(\mathbf{x}_k^R)$  or  $\langle \mathbf{x}_k^R \rangle$ . The corresponding reservoir system has both ESP and FMP if and only if the matrix  $\mathbf{A}$  in Eq. (3) fulfills  $\rho(\mathbf{A}) < 1$ .*

This is the sought condition for RC with harmonic networks, either classical or quantum. Importantly, it allows to discriminate useful reservoirs by simple inspection of the parameters of the network through the spectral radius of  $\mathbf{A}$ .

We now turn our attention to universality. The final requirement to fulfill is separability, which means that for any pair of different time series there is an instance of the model that can tell them apart. Then the class of systems defined by Eq. (3) is universal<sup>38,76</sup> in the following sense. For any causal time-invariant fading memory functional  $F$  (see Methods, reservoir computing theory) there exists a finite set of functionals realized by our model that can be combined to approximate  $F$  up to any desired accuracy. Physically, such combinations can be realized by combining the outputs of many instances of the model with a polynomial function. Mathematically, this amounts to constructing the polynomial algebra of functionals.

The next theorem (of which we give a simplified version here and a full version in the Supplementary Note 3) summarizes our analysis of the model described in Eq. (3).

**Universality theorem (simplified)** *Given instances of reservoir systems governed by Eq. (3) with a given  $\Delta t$  and for  $\rho(\mathbf{A}) < 1$ , hence providing a family  $\mathcal{Q}$  of fading memory functionals, the polynomial algebra of these functionals has separability. This holds also for the subfamilies  $\mathcal{Q}_{\text{thermal}}$ ,  $\mathcal{Q}_{\text{squeezed}}$ , and  $\mathcal{Q}_{\text{phase}}$ , that correspond to thermal, squeezed, and phase encoding, respectively. As any causal, time-invariant fading memory functional  $F$  can be uniformly approximated by its elements, the reservoir family of Eq. (3) with the specified constraint is universal.*

We sketch the main ingredients of the proof. Since the model admits arbitrarily small values of  $\rho(\mathbf{A})$ , there are instances where  $\rho(\mathbf{A}) < 1$ ; by virtue of Lemma 1,  $\mathcal{Q}$  can therefore be constructed and it is not empty. We show that the associated algebra has separability. Since the space of inputs is taken to be uniformly bounded, we may invoke the Stone-Weierstrass Theorem<sup>77</sup> and the claim follows. Full proof and additional details are in Supplementary Note 3.

We note that unlike ESP and FMP, separability depends explicitly on the input encoding. In Supplementary Note 3 we show separability for three different encodings of the input to elements of  $\sigma(\mathbf{x}_k^A)$ : thermal ( $n_{\text{th}}$ ), squeezing strength ( $r$ ) and phase of squeezing ( $\varphi$ ). It should be pointed out that separability (and hence, universality) could be shown also for first moments encoding in a similar manner.

**Controlling performance with input encoding.** The here derived Universality Theorem guarantees that for any temporal task, there is a finite set of reservoirs and readouts that can perform it arbitrarily well when combined. Let us now assume a somewhat more practical point of view: we possess a given generic network, and we attempt to succeed in different tasks by training the output function  $h$  to minimize the squared error between output  $\mathbf{o}$  and target output. For simplicity, we will also take inputs to be sequences of real numbers, rather than sequences of vectors—we stress that universality holds for both. Beyond universality of the whole class of Gaussian reservoirs, what is the performance and versatility of a generic one?

First of all, we will address how to single out instances with good memory. As pointed out earlier, memory is provided by the dependency of the reservoir observables on the input sequence. Informally speaking, reservoirs with good memory can reproduce a wide range of functions of the input and, therefore, learn many different tasks. Furthermore, to be useful a reservoir should possess nonlinear memory, since this allows the offloading of nontrivial transformations of the input to the reservoir. Then nonlinear time series processing can be carried out while keeping the readout linear, which simplifies training and reduces the overhead from evaluating the trained function.

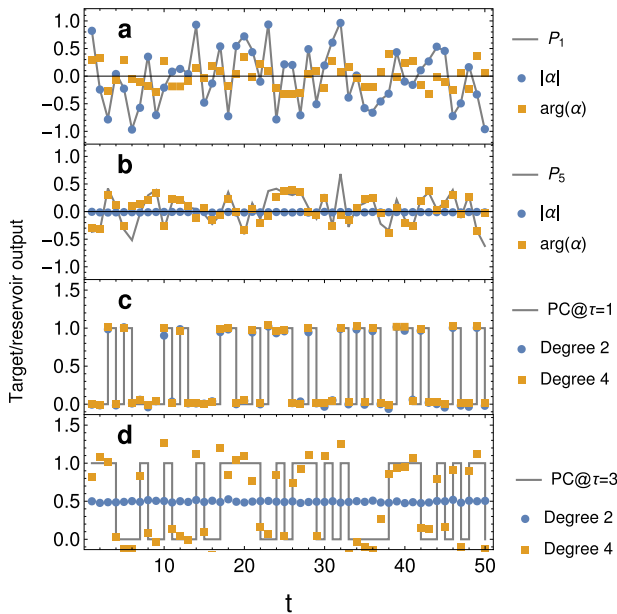
Memory is strongly connected to FMP; in fact, a recent general result concerning reservoirs processing discrete-time data is that under certain mild conditions, FMP guarantees that the total memory of a reservoir—bounded by the number of linearly independent observables used to form the output—is as large as possible<sup>71</sup>. Consequently, all instances that satisfy the spectral radius condition of Lemma 1 have maximal memory in this sense. Indeed with Lemma 1 the condition for FMP is straightforward to check. Furthermore, we find that reservoir observables seem to be independent as long as  $\mathbf{L}$  in Eq. (7) does not have special symmetries—as a matter of fact, numerical evidence suggests a highly symmetric network such as a completely connected network with uniform frequencies and weights never satisfies  $\rho(\mathbf{A}) < 1$ . Having FMP says nothing about what kind of functions the memory consists of, however.

Is there nonlinear memory? It is evident that the top equation of Eq. (3) is linear in reservoir and ancilla operators, but the encoding is not necessarily linear because of the way ancilla operators  $\mathbf{x}_k^A$  depend on the input. For single-mode Gaussian states (see Eq. (9) in Methods), it can be seen that the reservoir state is nonlinear in input when encoding to either magnitude  $r$  or phase  $\varphi$  of squeezing, or the phase of displacement  $\arg(\alpha)$ . Otherwise, that is for encoding in coherent states amplitude or thermal states average energy, it is linear (see Eq. (10) in Methods). This implies that the input encoding is the only source of nonlinear memory when the readout is linear—nonlinearity comes from pre-processing the input to ancilla states, not from an activation function, which is the conventional source of nonlinearity in RC.

The performance of Gaussian RC can be assessed considering different scenarios. For the remainder of this work we fix the network size to  $N = 8$  oscillators and form the output using the reservoir observables and a bias term; see Methods for details about the networks used in numerical experiments and their training. We consider nonlinear tasks in Fig. 2. In panels **a** and **b** we take the output function  $h$  to be a linear function of  $\langle \mathbf{x}_k^R \rangle$  and inputs  $s_k$  to be uniformly distributed in  $[-1, 1]$ , and consider two different encodings of the input into the ancilla  $\langle \mathbf{x}_k^A \rangle$ , as the amplitude and phase of coherent states. Setting  $|\alpha| \rightarrow s_k + 1$  and phase to a fixed value  $\arg(\alpha) \rightarrow 0$  leads to fully linear memory, which leads to good performance in the linear task of panel **a** only. In contrast, setting  $|\alpha| \rightarrow 1$  and encoding the input to phase as  $\arg(\alpha) \rightarrow 2\pi s_k$  leads to good performance in the nonlinear task shown in panel **b** and limited success in the linear one of panel **a**.

Nonlinearity of reservoir memory is not without limitations since  $\langle \mathbf{x}_k^R \rangle$  does not depend on products of the form  $s_k s_j \dots s_l$  where at least some of the indices are unequal, i.e., on products of inputs at different delays, as can be seen from Eq. (10). This is a direct consequence of the linearity of reservoir dynamics. When  $h$  is linear in  $\langle \mathbf{x}_k^R \rangle$  the output will also be independent of these product terms, hindering performance in any temporal task requiring them. While Universality Theorem implies the existence of a set of





**Fig. 2 Nonlinear information processing with a generic reservoir.** The targets, indicated by solid lines, are functions of the input  $\mathbf{s}$  with elements  $s_k \in [-1, 1]$ , where  $k$  indicates the timestep. Given the input the reservoir is trained to produce output that approximates the target as well as possible. In all panels the output is formed from  $\langle \mathbf{x}_k^R \rangle$ , i.e., the reservoir first moments at timestep  $k$ . In panels **a** and **b** two different encodings (in the magnitude  $|\alpha|$  and phase  $\arg(\alpha)$  of a coherent state) are compared, the trained readout function  $h$  is linear and the target are normalized Legendre polynomials  $P_d$  of degree  $d$  1 in **a** and 5 in **b** of the input. Encoding in the magnitude of displacement allows to reproduce only the linear target  $(P_1)_k = s_k$  while phase encoding has good performance with  $(P_5)_k = (15s_k - 70s_k^3 + 63s_k^5)/8$ , confirming that some nonlinear tasks are possible with linear  $h$ . In panels **c** and **d** the encoding is fixed to  $|\alpha|$  but  $h$  is a polynomial of the reservoir first moments whose degree is varied (both quadratic and quartic polynomials are considered). The target is the parity check task ( $PC@ \tau = 1$  and 3 in **c** and **d**, respectively), which requires products of the input elements at different delays  $\tau$ . Increasing the degree allows the task to be solved at increasingly long delays. In all cases a network of  $N = 8$  oscillators is used and the reservoir output is compared to target for 50 timesteps after training (see Methods for details about the used networks and their training).

reservoirs for any task, we will show that even a single generic reservoir can be sufficient when nonlinearity is introduced at the level of readout, at the cost of relatively more involved training.

To illustrate the nontrivial information processing power of a single generic reservoir, we consider the parity check task<sup>78</sup> (Fig. 2c, d), defined as

$$(PC(\tau))_k = \text{mod} \left( \sum_{l=0}^{\tau} s_{k-l}, 2 \right) \quad (5)$$

where  $s_k \in \{0, 1\}$ ; the target output is 0 if the sum of  $\tau + 1$  most recent inputs is even and 1 otherwise. It can be shown that  $(PC(\tau))_k$  coincides with a sum of products of inputs at different delays for binary input considered here. When encoding in a coherent state amplitude ( $|\alpha| \rightarrow s_k, \arg(\alpha) \rightarrow 0$ ) for readout function  $h$  polynomial in  $\langle \mathbf{x}_k^R \rangle$ , the results show that increasing the polynomial degree  $d$  of the function  $h$  allows to solve the task for higher delays  $\tau$ . In particular, we find that the reservoir can solve the well-known XOR problem for nonlinearity degrees  $d \geq 2$ , which coincides with the parity check at  $\tau = 1$ . The parity check at  $\tau = 3$  works, in turn, for  $d \geq 4$ .

**Information processing capacity.** Besides providing non-linearity, input encoding also facilitates a versatile tuning of the linear and nonlinear memory contributions. To demonstrate this, we consider how input encoding affects the degree of nonlinear functions that the reservoir can approximate, as quantified by the information processing capacity (IPC)<sup>71</sup> of the reservoir. The IPC generalizes the linear memory capacity<sup>79</sup> often considered in RC to both linear and nonlinear functions of the input. Even if its numerical evaluation is rather demanding, it has the clear advantage to provide a broad assessment of the features of RC, beyond the specificity of different tasks.

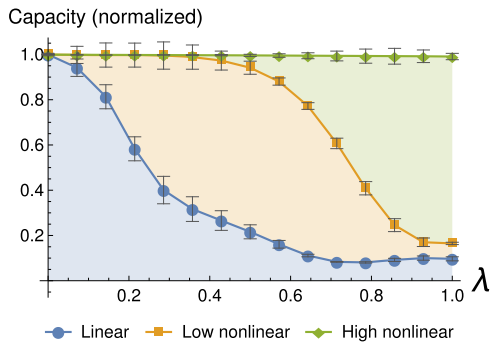
We may define the IPC as follows. Let  $X$  be a fixed reservoir,  $z$  a function of a finite number of past inputs and let  $h$  be linear in the observables of  $X$ . Suppose the reservoir is run with two sequences  $\mathbf{s}'$  and  $\mathbf{s}$  of random inputs drawn independently from some fixed probability distribution  $p(s)$ . The first sequence  $\mathbf{s}'$  is used to initialize the reservoir; observables are recorded only when the rest of the inputs  $\mathbf{s}$  are encoded. The capacity of the reservoir  $X$  to reconstruct  $z$  given  $\mathbf{s}'$  and  $\mathbf{s}$  is defined to be

$$C_{\mathbf{s}', \mathbf{s}}(X, z) = 1 - \frac{\min_h \sum_k (z_k - o_k)^2}{\sum_k z_k^2} \quad (6)$$

where the sums are over timesteps  $k$  after initialization, each  $z_k$  is induced by the function  $z$  to be reconstructed and the input, and we consider the  $h$  that minimizes the squared error in the numerator. The maximal memory mentioned earlier may be formalized in terms of capacities: under the conditions of Theorem 7 in the work by Dambre et al.<sup>71</sup>, the sum of capacities  $C_{\mathbf{s}', \mathbf{s}}(X, z)$  over different functions  $z$  is upper bounded by the number of linearly independent observables used by  $h$ , with the bound saturated if  $X$  has FMP. This also implies that finite systems have finite memory, and in particular increasing the nonlinearity of the system inevitably decreases the linear memory<sup>71,80</sup>. Importantly, infinite sequences  $\mathbf{s}'$ ,  $\mathbf{s}$  and a set of functions that form a complete orthogonal system w.r.t.  $p(s)$  are required by the theorem; shown results are numerical estimates. We consistently take  $p(s)$  to be the uniform distribution in  $[-1, 1]$ ; examples of functions  $z$  orthogonal w.r.t. this  $p(s)$  include Legendre polynomials  $P_1$  and  $P_5$  appearing in Fig. 2, as well as their delayed counterparts. Further details about the estimation of total capacity are given in Methods.

We consider the breakdown of the normalized total capacity to linear (covering functions  $z$  with degree 0 or 1), nonlinear (degrees 2–3) and highly nonlinear (degree 4 or higher) regimes in Fig. 3. We take  $h$  to be a linear function of  $\langle \mathbf{x}_k^R \rangle$  and address the capability to have Gaussian RC operating with different linear and nonlinear capabilities by varying the input encoding into a coherent ancillary state from amplitude to phase  $|\alpha| \rightarrow (1 - \lambda)(s_k + 1) + \lambda, \arg(\alpha) \rightarrow 2\pi\lambda s_k$  where  $s_k \in [-1, 1]$ ; this is a convex combination of the two encodings used in panels **a** and **b** of Fig. 2. As can be seen in Fig. 3, adjusting  $\lambda$  allows one to move from fully linear (for amplitude encoding) to highly nonlinear (for phase) information processing, which can be exploited for a versatile tuning of the reservoir to the task at hand. Remarkably, this can be done without changing neither the parameters of the Hamiltonian in Eq. (7) (that is, the reservoir system) nor the observables extracted as output in  $h$ . The earlier mentioned trade-off between linear and nonlinear memory keeps the total memory bounded, however, Lemma 1 ensures that this bound is saturated for all values of  $\lambda$ .

**From intensities to field fluctuations and RC with quantum resources.** Previously we considered coherent states for the ancilla, encoding the input to  $|\alpha|$  and  $\arg(\alpha)$ . In the limit of large amplitudes  $|\alpha| \gg 1$ , coherent states allow for a classical



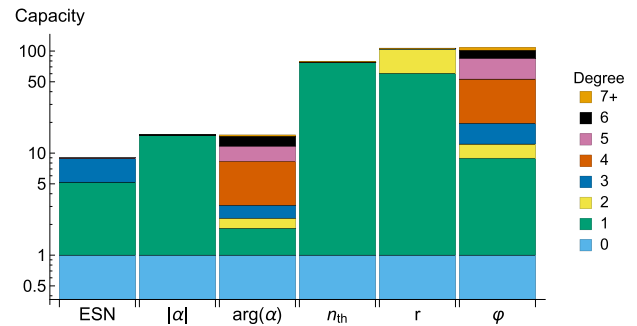
**Fig. 3 Control of nonlinearity of reservoir memory via input encoding.**

Here the input  $s_k \in [-1, 1]$  is encoded to the displacement of the ancilla according to  $|\alpha| \rightarrow (1 - \lambda)(s_k + 1) + \lambda$ ,  $\arg(\alpha) \rightarrow 2\pi\lambda s_k$ , where  $\lambda \in [0, 1]$  is a parameter controlling how much we encode to the amplitude  $|\alpha|$  or phase  $\arg(\alpha)$  of the displacement. Reservoir memory is measured using information processing capacity, which quantifies the ability of the reservoir to reconstruct functions of the input at different delays. The figure shows how the relative contributions from linear and nonlinear functions to the normalized total capacity can be controlled with  $\lambda$ . Nonlinear contributions are further divided into degrees 2 and 3 (low nonlinear) and higher (high nonlinear). For  $\lambda = 0$  the encoding is strictly to  $|\alpha|$ , leading to linear information processing, while at  $\lambda = 1$  only  $\arg(\alpha)$  depends on the input, leading to most of the capacity to come from functions of the input with degree at least 4. All results are averages over 100 random reservoirs and error bars show the standard deviation.

description of the harmonic network, with field operator expectation values corresponding, for instance, to classical laser fields<sup>81,82</sup>. This realization of RC would be impossible in the limit of vanishing fields where  $|\alpha| \rightarrow 0$  since then  $\langle \mathbf{x}_k^R \rangle$  becomes independent of input. Here, we put forward the idea of harnessing instead the fluctuations encoded in the covariance matrix  $\sigma(\mathbf{x}_k^R)$ , which also increases the amount of linearly independent reservoir observables. Even if only a subset of them can be directly monitored, the rest will play the role of hidden nodes that may be chosen to contribute as independent computational nodes, and have already been suggested to be a source of computational advantage in quantum RC in spin systems<sup>44</sup>. Here, we analyze the full IPC of the system including all observables.

In the classical regime, thermal fluctuations could be used, corresponding to thermal states (encoding to  $n_{\text{th}}$ ) for the ancilla. While representing an original approach to the best of our knowledge, it will be seen that these classical fluctuations will provide only linear memory and as such cannot be used to solve nontrivial temporal tasks without relying on external sources of nonlinearity. We propose to solve this problem by relying instead on quantum fluctuations provided by encoding the input to squeezed vacuum (encoding to  $r, \varphi$ ), i.e., by adopting a quantum approach to RC with Gaussian states. By virtue of the Universality Theorem, either thermal or squeezed states encoding can be used for universal RC.

We compare the classical and quantum approaches in Fig. 4, where we show how the capacity of the system of seven reservoir oscillators and the ancilla gets distributed for the aforementioned encodings. For comparison, we also include the case of an echo-state network (ESN, see Methods) consisting of 8 neurons, a classical reservoir computer based on a recurrent neural network<sup>43</sup>. We separate the contributions to total capacity according to degree to appreciate the linear memory capacity (degree 1) and nonlinear capacity and take  $s_k$  to be uniformly distributed in  $[-1, 1]$ . The total capacity, as displayed in Fig. 4, allows to visualize the clear advantage of the oscillators network



**Fig. 4 Histogram bars showing the information processing capacity for different input encodings.**

The capacities are further divided into linear (degree 0 or 1) and nonlinear contributions (degree 2 or higher), where the degree is given by the functions that the reservoir can reconstruct. Capacity of an echo-state network (ESN) with as many neurons (8) as there are oscillators in the harmonic network is shown for comparison. The output is a function of reservoir first moments when encoding to either magnitude  $|\alpha|$  or phase  $\arg(\alpha)$  of displacement. Three different ways to encode the input in the limit  $|\alpha| \rightarrow 0$  are shown; for them the output is a function of the elements of reservoir covariances, with the input being encoded either to thermal excitations  $n_{\text{th}}$ , squeezing strength  $r$  or angle  $\varphi$ .

over the ESN, as well as the particularities of the different input encodings.

The cases  $|\alpha|$  and  $\arg(\alpha)$  in Fig. 4 correspond to Fig. 3 for  $\lambda = 0$  and  $\lambda = 1$ , respectively. For them, we take  $h$  to be linear in  $\langle \mathbf{x}_k^R \rangle$ . The total capacity for ESN and harmonic networks with  $|\alpha|$  and  $\arg(\alpha)$  encoding differ by a factor 2 because the neurons of ESN each provide a single observable, while two quadratures are considered for reservoir oscillators. Comparing capacities, ESN provides more nonlinear memory than amplitude encoding of coherent states  $|\alpha|$ , but for phase encoding we can see significant nonlinear contributions to total capacity from higher degrees, up, e.g., to 7, in spite of the small size of the harmonic network.

Let us now reduce  $|\alpha| \rightarrow 0$ . In this case, we show in Fig. 4 that classical thermal fluctuations provided by setting  $n_{\text{th}} \rightarrow s_k + 1$  and taking  $h$  to be a linear function of  $\sigma(\mathbf{x}_k^R)$  increase the capacity significantly, as the number of available observables is proportional to the square of the number of oscillators (as explained in Methods, training of the network). For thermal encoding, the capacity does not reach the total capacity bound, limited by dependencies between some of the observables. Furthermore, there is only linear memory. As shown in Fig. 4, considering instead quantum fluctuations gives both a quadratic capacity increase and nonlinear memory. Setting  $r \rightarrow s_k + 1$ ,  $\varphi \rightarrow 0$  has a ratio of linear to nonlinear memory somewhat similar to the ESN, while setting  $r \rightarrow 1$ ,  $\varphi \rightarrow 2\pi s_k$  gives significant amounts of nonlinear memory (last bar in Fig. 4), displaying the advantage of encoding in squeezed vacuum.

## Discussion

The concept of universality in different fields and applications depends on the scope. In the context of quantum computing it refers to the ability to reproduce any unitary transformation<sup>57,58</sup>, in feed-forward neural networks it characterizes the possibility to approximate any continuous function<sup>39,40</sup>, and in RC it identifies the ability to approximate as accurately as one wants so called fading memory functions<sup>37,38</sup>.

Recently, universality of RC has been proved in the full class of classical linear systems<sup>76</sup>, in echo-state networks<sup>38</sup>, and on the quantum side, for the spins maps of the work by Chen et al.<sup>49</sup> tested with superconducting quantum computers. The choice of the quantum map governing the RC dynamics is also relevant.

For instance, at variance with the work by Chen et al.<sup>49</sup>, a previous universality demonstration with spins<sup>47</sup> applies to an input-dependent dynamical map that is not guaranteed to be convergent for all input sequences. Although in general more challenging to implement, it has been suggested that quantum reservoirs may have a more favorable scaling of computational power with size than classical reservoirs<sup>44</sup>; importantly, universality guarantees that this power may be harnessed for achieving any temporal tasks. In this work we have established that even with the limited resources offered by linear quantum optics platforms, Gaussian measurements, and a restricted class of input Gaussian states, such as thermal or squeezed vacuum, universal reservoir computing is achieved. Our work provides the foundations for RC with generic quadratic Hamiltonians in linear optics platforms, classical or quantum, with minimal resources<sup>83,84</sup>.

We find that performance is not limited by the reduced resources in place, as well as a clear improvement when information is encoded in quantum fluctuations of the field, providing superior capacity. This is displayed in the quadratic growth of IPC with system size, as opposed to linear in the case of first moments or standard ESNs. Versatility is also achieved without modifying the network reservoir, as input encoding leads to easily tunable nonlinear memory—already proven to be beneficial to tackle different computational tasks in classical RC<sup>85,86</sup>. Non-classical resources in Gaussian systems originate from squeezing and this is shown to provide nonlinear memory necessary for achieving nontrivial RC with a linear readout clearly beyond the performance achieved with thermal fluctuations, which are classical.

RC realized with physical systems is a new and rapidly developing field<sup>87</sup>, with the extension to quantum regime proposed in a handful of pioneering works<sup>44–47,49,52</sup>. Our proposal can be adapted to many classical and quantum platforms modeled by continuous variables in linear networks<sup>26,63,64,67–70,88,89</sup>. As an example, any harmonic network described by Eq. (1) can be implemented in optical parametric processes pumped by optical frequency combs, based on interacting spectral or temporal modes. The network size and structure are controlled by shaping the pump and multimode measurements, allowing in principle to create any network within a fixed optical architecture<sup>63</sup>. Lemma 1 provides a simple and mild condition to identify networks with echo-state and fading memory properties in this reconfigurable platform. Even if the full numerical analysis of the IPC here is limited to 8 nodes, larger networks have already been reported<sup>64</sup>. These linear quantum optics platforms are already well-established for measurement-based quantum computing<sup>64–66</sup>, have intrinsic resilience to decoherence even at room temperature and high potential for scalability. Our study reveals their potential as promising physical systems for quantum RC.

The proposed scheme assumes Gaussian measurements as homodyne or heterodyne detection<sup>90</sup>. A proof of principle realization of this protocol would require several copies of the experiment<sup>44,45,49</sup> at each time. Otherwise, when monitoring the system output in order to exploit RC for temporal tasks, the back-action effect of quantum measurement needs to be taken into account. Even if the system performance is robust up to some level of classical noise and the effective dissipation introduced by measurement turns out to be beneficial for the fading memory, the design of platforms including measurement protocols is a question open to further research. In optical implementations, a clear advantage is the possibility to exploit arbitrary-node manipulation via beam splitters operations and feedback loops, as in recent experiments<sup>64,89</sup>. Measurements may also be a potential source of computational power introducing nonlinearity<sup>52</sup>. As examples, even for Gaussian states, non-Gaussian measurements allow to

reproduce any unitary transformation<sup>91</sup> or to make a problem intractable for classical computers, as in Boson sampling<sup>92</sup>.

The most direct advantage when considering RC with quantum systems is the possibility to access a larger Hilbert space<sup>44</sup>. With the minimal resources of a Gaussian model, this leads to a quadratic advantage. An interesting perspective is to explore a potential superior performance based on non-Gaussian resources<sup>93–95</sup>, to achieve exponential scaling of the total processing capacity in the quantum regime and a genuine quantum advantage in terms of computational complexity<sup>91,96</sup>. A further possibility is to embed the (quantum) reservoir within the quantum system to be processed in order to explore quantum RC in quantum tasks, providing a quantum-quantum machine learning (in the data to be analyzed and in the RC platform, respectively). Finally, the extension of the learning theory to quantum output is still missing; tasks of interests could include training the reservoir to simulate given quantum circuits, for instance. Static versions of such tasks carried out in feed-forward, as opposed to recurrent, architecture have been considered<sup>48,53,97</sup> and could be used as a starting point.

## Methods

**Linear network Hamiltonian.** We consider a network of interacting quantum harmonic oscillators acting as the reservoir for RC, with spring-like interaction strengths  $g_{ij}$ . The Hamiltonian of such a system can be conveniently described in terms of the Laplacian matrix  $\mathbf{L}$  having elements  $L_{ij} = \delta_{ij}\sum_k g_{ik} - (1 - \delta_{ij})g_{ij}$ . We adopt such units that the reduced Planck constant  $\hbar = 1$  and the Boltzmann constant  $k_B = 1$ . Arbitrary units are used for other quantities such as frequency and coupling strength. The resulting Hamiltonian is

$$H = \frac{\mathbf{p}^T \mathbf{p}}{2} + \frac{\mathbf{q}^T (\Delta_\omega^2 + \mathbf{L}) \mathbf{q}}{2}, \quad (7)$$

where  $\mathbf{p}^T = \{p_1, p_2, \dots, p_N\}$  and  $\mathbf{q}^T = \{q_1, q_2, \dots, q_N\}$  are the vectors of momentum and position operators of the  $N$  oscillators while the diagonal matrix  $\Delta_\omega$  holds the oscillator frequencies  $\omega^T = \{\omega_1, \omega_2, \dots, \omega_N\}$ .

**Reservoir computing theory.** A common way to process temporal information is to use artificial neural networks with temporal loops. In these so-called recurrent neural networks, the state of the neural network nodes depends on the input temporal signals to be processed but also on the previous states of the network nodes, providing the needed memory<sup>98</sup>. Unfortunately, such recurrent neural networks are notorious for being difficult to train<sup>99</sup>. Reservoir Computing, in turn, leads to greatly simplified and faster training, enlarges the set of useful physical systems as reservoirs, and lends itself to simultaneous execution of multiple tasks by training separate output weights for each task while keeping the rest of the network—the reservoir—fixed<sup>34</sup>.

Here, we provide an overview of Reservoir computing theory that introduces the relevant definitions and concepts in context. For proper development of the discussed material we refer the reader to<sup>38,100</sup>. We will also briefly discuss the application of the framework to quantum reservoirs.

**Reservoir computers.** We consider sequences of discrete-time data  $\mathbf{s} = \{\dots, \mathbf{s}_{i-1}, \mathbf{s}_i, \mathbf{s}_{i+1}, \dots\}$ , where  $\mathbf{s}_i \in \mathbb{R}^n$ ,  $n$  is the dimension of the input vector and  $i \in \mathbb{Z}$ . Let us call the space of input sequences  $\mathcal{U}_n$  such that  $\mathbf{s} \in \mathcal{U}_n$ . Occasionally, we will also use left and right infinite sequences defined as  $\mathcal{U}_n^- = \{\mathbf{s} = \{\dots, \mathbf{s}_{-2}, \mathbf{s}_{-1}, \mathbf{s}_0\} | \mathbf{s}_i \in \mathbb{R}^n, i \in \mathbb{Z}_-\}$  and  $\mathcal{U}_n^+ = \{\mathbf{s} = \{\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \dots\} | \mathbf{s}_i \in \mathbb{R}^n, i \in \mathbb{Z}_+\}$ , respectively. Formally, a reservoir computer may be defined by the following set of equations:

$$\begin{cases} \mathbf{x}_k = T(\mathbf{x}_{k-1}, \mathbf{s}_k) \\ o_k = h(\mathbf{x}_k), \end{cases} \quad (8)$$

where  $T$  is a recurrence relation that transforms input sequence elements  $\mathbf{s}_k$  to feature space elements  $\mathbf{x}_k$ —in general, in a way that depends on initial conditions—while  $h$  is a function from the feature space to reals. When  $T$ , a target  $\bar{o}$  and a suitable cost function describing the error between output and target are given, the reservoir is trained by adjusting  $h$  to optimize the cost function. The error should remain small also for new input that was not used in training.

The general nature of Eq. (8) makes driven dynamical systems amenable to being used as reservoirs. This has opened the door to so-called physical reservoir computers that are hardware implementations exploiting different physical substrates<sup>87</sup>. In such a scenario time series  $\mathbf{s}$ —often after suitable pre-processing—drives the dynamics given by  $T$  while  $\mathbf{x}_k$  is the reservoir state. A readout mechanism that can inspect the reservoir state should be introduced to implement function  $h$ . The appeal of physical RC lies in the possibility to offload processing of the input in feature space and memory requirements to the reservoir, while keeping the readout



mechanism simple and memoryless. In particular, this can lead to efficient computations in terms of speed and energy consumption with photonic or electronic systems<sup>28,101</sup>.

**Temporal maps and tasks.** Online time series processing—what we wish to do with the system in Eq. (8)—is mathematically described as follows. A temporal map  $M: \mathcal{U}_n \rightarrow \mathcal{U}_1$ , also called a filter, transforms elements from the space of input time series to the elements of the space of output time series. In general  $M$  is taken to be causal, meaning that  $(M[\mathbf{s}])_t$  may only depend on  $\mathbf{s}_k$  where  $k \leq t$ , i.e., inputs in the past only. When  $M$  is additionally time-invariant, roughly meaning that it does not have an internal clock,  $(M[\mathbf{s}])_t = F(\{\dots, \mathbf{s}_{t-2}, \mathbf{s}_{t-1}, \mathbf{s}_t\})$  for any  $t$  for some fixed  $F: \mathcal{U}_n \rightarrow \mathbb{R}^{38}$ . We will later refer to such  $F$  as functionals. When  $M$  is given, fixing  $\mathbf{s}$  induces a time-dependent function that we will denote by  $f$ , defined by  $f(t) = F(\{\dots, \mathbf{s}_{t-2}, \mathbf{s}_{t-1}, \mathbf{s}_t\})$ .

To process input  $\mathbf{s}$  into  $\mathbf{o}$  in an online mode requires to implement  $f(t)$ ; real-time processing is needed. We will later refer to such tasks as temporal tasks. Reservoir computing is particularly suited for this due to the memory of past inputs provided by the recursive nature of  $T$  and online processing accomplished by the readout mechanism acting at each timestep.

**Properties of useful reservoirs.** In general,  $o_k$  in Eq. (8) depends on both the past inputs and the initial conditions, but  $f(t)$  depends only on the inputs; therefore any dependency on the initial conditions should be eliminated by the driving. It may also be expected that reservoirs able to learn temporal tasks must be in some sense well-behaved when driven. These informal notions can be formalized as follows.

The echo-state property (ESP)<sup>43</sup> requires that for any reference time  $t$ ,  $\mathbf{x}_t = \mathcal{E}(\{\dots, \mathbf{s}_{t-2}, \mathbf{s}_{t-1}, \mathbf{s}_t\})$  for some function  $\mathcal{E}$ , that is to say at the limit of infinitely many inputs the reservoir state should become completely determined by the inputs, and not by the initial conditions. This has two important consequences. First, it guarantees that the reservoir always eventually converges to the same trajectory of states for a given input, which also means that initial conditions do not need to be taken into account in training. Second, it ensures that the reservoir together with a readout function can realize a temporal map. A strongly related condition called the fading memory property (FMP)<sup>75</sup> requires that for outputs to be similar, it is sufficient that the inputs are similar up to some finite number of past inputs. The formal definition can be given in terms of so-called null sequences as explained in the Supplementary Note 1. It can be shown that FMP imposes a form of continuity to the overall input-output maps that can be produced by the reservoir computer described by Eq. (8)<sup>37</sup>; the associated temporal maps (functionals) are called fading memory temporal maps (functionals).

A useful quantifier for the processing power of a single reservoir was introduced by Dambre et al.<sup>71</sup>. They showed that when the readout function  $h$  is linear in reservoir variables, the ability of a reservoir to reconstruct orthogonal functions of the input is bounded by the number of linearly independent variables used as arguments of  $h$ . The central result was that all reservoirs with FMP can saturate this bound.

Considering a class of reservoirs instead offers a complementary point of view. If the reservoirs have ESP and FMP then they can realize temporal maps. If additionally the class has separability, i.e., for any  $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{U}_n$ ,  $\mathbf{s}_1 \neq \mathbf{s}_2$ , some reservoir in the class will be driven to different states by these inputs, then universality in RC becomes possible. This can be achieved by imposing mild additional conditions on the input space and realizing an algebra of temporal maps by combining the outputs of multiple reservoirs with a polynomial function<sup>76</sup>. When these properties are met, universality then follows from the Stone-Weierstrass Theorem (Theorem 7.3.1 in the work by Dieudonne et al.<sup>77</sup>).

**The explicit forms of the covariance matrix and first moments vector.** For a single-mode Gaussian state with frequency  $\Omega$ , covariances and first moments read

$$\begin{cases} \sigma(\mathbf{x}) = (n_{\text{th}} + \frac{1}{2}) \begin{pmatrix} (y + z_{\text{cos}})\Omega^{-1} & z_{\text{sin}} \\ z_{\text{sin}} & (y - z_{\text{cos}})\Omega \end{pmatrix}, \\ \langle \mathbf{x} \rangle = \begin{pmatrix} |\alpha| \cos(\arg(\alpha)) \sqrt{2\Omega^{-1}} \\ |\alpha| \sin(\arg(\alpha)) \sqrt{2\Omega} \end{pmatrix}, \end{cases} \quad (9)$$

where  $y = \cosh(2r)$ ,  $z_{\text{cos}} = \cos(\varphi) \sinh(2r)$  and  $z_{\text{sin}} = \sin(\varphi) \sinh(2r)$ . Here,  $n_{\text{th}}$  controls the amount of thermal excitations,  $r$  and  $\varphi$  control the magnitude and phase of squeezing, respectively, and finally  $|\alpha|$  and  $\arg(\alpha)$  control the magnitude and phase of displacement, respectively. The input sequence may be encoded into any of these parameters or possibly their combination.

Suppose that  $\mathbf{s} = \{\mathbf{s}_{-m+1}, \dots, \mathbf{s}_{-1}, \mathbf{s}_0\}$  and each input  $\mathbf{s}_k$  is encoded to all degrees of freedom as  $n_{\text{th}} \mapsto n_{\text{th}}(\mathbf{s}_k)$ ,  $r \mapsto r(\mathbf{s}_k)$ ,  $\varphi \mapsto \varphi(\mathbf{s}_k)$ ,  $|\alpha| \mapsto |\alpha(\mathbf{s}_k)|$  and  $\arg(\alpha) \mapsto \arg(\alpha(\mathbf{s}_k))$ . Then from Eq. (4) it follows that

$$\begin{cases} \langle \mathbf{x}_k^{\text{re}} \rangle - \mathbf{A}^m \langle \mathbf{x}_{-m}^{\text{re}} \rangle (\mathbf{A}^T)^m \Big|_y = \sum_{i=0}^{m-1} a_i^{\text{re}} n_{\text{th}}(\mathbf{s}_k) (\cosh(2r(\mathbf{s}_k)) + (b_i^{\text{re}} \cos(\varphi(\mathbf{s}_k)) + c_i^{\text{re}} \sin(\varphi(\mathbf{s}_k))) \sinh(2r(\mathbf{s}_k))), \\ \langle \mathbf{x}_k^{\text{im}} \rangle - \mathbf{A}^m \langle \mathbf{x}_{-m}^{\text{im}} \rangle \Big|_y = \sum_{i=0}^{m-1} |a_i(\mathbf{s}_k)| (a_i^{\text{im}} \cos(\arg(\alpha(\mathbf{s}_k))) + b_i^{\text{im}} \sin(\arg(\alpha(\mathbf{s}_k)))) \end{cases} \quad (10)$$

where  $a_k^{\text{re}}, b_k^{\text{re}}, c_k^{\text{re}}, a_k^{\text{im}}, b_k^{\text{im}}, c_k^{\text{im}}$ , and  $b_k^{\text{im}}$  are constants depending on the Hamiltonian in Eq. (7) and  $\Delta t$ . That is to say the part of the observables independent of the initial conditions  $\mathbf{x}_{-m}^{\text{re}}$  are linear combinations of  $n_{\text{th}}(\mathbf{s}_k)$  and  $|\alpha(\mathbf{s}_k)|$ , while the dependency on  $r(\mathbf{s}_k)$ ,  $\varphi(\mathbf{s}_k)$  and  $\arg(\alpha(\mathbf{s}_k))$  is nonlinear. When the dynamics of the reservoir is

convergent the effect of the initial conditions vanishes at the limit  $m \rightarrow \infty$  and the corresponding terms on the L.H.S. may be omitted.

**The networks used in numerical experiments.** We have used a chain of  $N = 8$  oscillators for all results shown in Figs. 2 and 4. For simplicity, all oscillators have the same frequency  $\omega = 0.25$  and all interaction strengths are fixed to  $g = 0.1$ . The ancilla is chosen to be one of the oscillators at the ends of the chain. For the aforementioned parameter values of  $\omega$  and  $g$ , we have computed  $\rho(\mathbf{A})$  as a function of  $\Delta t$ . We have set  $\Delta t = 59.6$ , which is close to a local minimum of  $\rho(\mathbf{A})$ ; in general values of  $\Delta t$  that achieve  $\rho(\mathbf{A}) < 1$  are common and produce similar results. It should be pointed out that the choice of ancilla matters, e.g., choosing the middle oscillator in a chain of odd length seems to lead to  $\rho(\mathbf{A}) \geq 1$  for any choice of  $\Delta t$ .

The starting point for the results shown in Fig. 3 is a completely connected network of  $N = 8$  oscillators with uniform frequencies  $\omega = 0.25$  and random interaction strengths uniformly distributed in the interval  $g = [0.01, 0.19]$ . We point out that the condition  $\rho(\mathbf{A}) < 1$  is favored by interaction strengths that break the symmetry of the system. A suitable value for  $\Delta t$  is then found as follows. We consider values  $\Delta t \omega_0 = 0.01, 0.02, \dots, 29.99, 30$  and find the corresponding  $\rho(\mathbf{A})$  for each. Then we choose a random  $\Delta t$  out of all values for which  $\rho(\mathbf{A}) \leq 0.99$ . In the rare event that none of the values can be chosen, new random weights are drawn and the process is repeated. We have checked that choosing instead the  $\Delta t$  that minimizes  $\rho(\mathbf{A})$  leads to virtually the same results, confirming that reservoir memory is primarily controlled by the encoding, not the choice of  $\Delta t$ .

**Training of the network.** For all shown results, we take the initial state of the reservoir to be a thermal state and use the first  $10^5$  timesteps to eliminate its effect from the reservoir dynamics, followed by another  $M = 10^5$  timesteps during which we collect the reservoir observables used to form the output. We wish to find a readout function  $h$  that minimizes

$$SE(\bar{\mathbf{o}}, \mathbf{o}) = \sum_k (\bar{o}_k - o_k)^2, \quad (11)$$

i.e., the squared error between target output  $\bar{\mathbf{o}}$  and actual output  $\mathbf{o}$ .

In Fig. 2a, b and in Figs. 3 and 4,  $h$  is linear in reservoir observables. In this case, the collected data is arranged into  $M \times L$  matrix  $\mathbf{X}$ , where  $L$  is the number of used observables. In the case of first moments,  $L = 2(N - 1)$ , whereas in the case of covariances  $L = 2N^2 - 3N + 1$ ; the latter comes from using the diagonal and upper triangular elements of the symmetric  $2(N - 1) \times 2(N - 1)$  covariance matrix. We introduce a constant bias term by extending the matrix with a unit column so that the final dimensions of  $\mathbf{X}$  are  $M \times (L + 1)$ . Now we may write  $o_k = h(\mathbf{X}_k) = \sum_i^{N+1} W_i X_{ki}$  where  $\mathbf{X}_k$  is the  $k$ th row of  $\mathbf{X}$ ,  $X_{ki}$  its  $i$ th element and  $W_i \in \mathbb{R}$  are adjustable weights independent of  $k$ . Let  $\mathbf{W}$  be the column vector of the weights. Now  $\mathbf{X}\mathbf{W} = \mathbf{o}^T$ . To minimize (11), we set

$$\mathbf{W} = \mathbf{X}^+ \bar{\mathbf{o}}^T \quad (12)$$

where  $\mathbf{X}^+$  is the Moore–Penrose inverse<sup>102,103</sup> of  $\mathbf{X}$ . When  $\mathbf{X}$  has linearly independent columns—meaning that the reservoir observables are linearly independent— $\mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ .

In Fig. 2c, d,  $h$  is taken to be polynomial in reservoir observables. In this case, the training proceeds otherwise as above except that before finding  $\mathbf{X}^+$  we expand  $\mathbf{X}$  with all possible products of different reservoir observables up to a desired degree, increasing the number of columns. Powers of the same observable are not included since they are not required by the parity check task.

**The used echo-state network.** An echo-state network (ESN) is used for some of the results shown in Fig. 4. For a comparison with a harmonic network of eight oscillators (one of them the ancilla) and a bias term we set the size of the ESN to  $N = 8$  neurons, all of which are used to form the output, and include a bias term.

The ESN has a state vector  $\mathbf{x}_k \in \mathbb{R}^N$  with dynamics given by  $\dot{\mathbf{x}}_k = \tanh(\beta \mathbf{W} \mathbf{x}_{k-1} + \mathbf{w} s_k)$  where  $\mathbf{W}$  is a random  $N \times N$  matrix,  $\mathbf{w}$  a random vector of length  $N$ ,  $\beta$ , and  $\mathbf{w}$  are scalars and  $\tanh$  acts element-wise.  $\mathbf{W}$  and  $\mathbf{w}$  are created by drawing each of their elements uniformly at random from the interval  $[-1, 1]$ . Furthermore,  $\mathbf{W}$  is scaled by dividing it by its largest singular value. Parameters  $\beta$  and  $\mathbf{w}$  are used to further adjust the relative importance of the previous state  $\mathbf{x}_{k-1}$  and scalar input  $s_k$ . We use a single fixed realization of  $\mathbf{W}$  and  $\mathbf{w}$  and set  $\beta = 0.95$  and  $\mathbf{w} = 1$ . The readout function is a linear function of the elements of  $\mathbf{x}_k$  and training follows a similar procedure to the one described for the oscillator networks in Methods D. We note that the precise linear and nonlinear memory contributions of the ESN to the IPC bar in Fig. 4 depend on the choice of the parameters values for  $\beta$  and  $\mathbf{w}$ . For this manuscript, the relevant aspect is that the total IPC of the ESN is bounded to the number of neurons (8), independently of the choice of the parameter values.

**Estimation of total information processing capacity.** Information processing capacity is considered in Figs. 3 and 4. By total capacity we mean the sum of capacities over a complete orthogonal set of functions and using infinite sequences  $\mathbf{s}'$  and  $\mathbf{s}$ . Shown results are estimates of the total capacity found as follows.

All estimates are formed with input i.i.d. in  $[-1, 1]$ . One choice of functions orthogonal w.r.t. this input is described in Eq. (12) of the work by Dambre et al.<sup>71</sup>, which we also use. More precisely, the considered orthogonality is defined in terms of the scalar product in the Hilbert space of fading memory functions given in Definition 5 of the work by Dambre et al.<sup>71</sup>—it should be stressed that in general, changing the input changes, which functions are orthogonal. Since  $\sigma(\mathbf{x}^R)$  and  $(\mathbf{x}^R)$  can only depend on products of the inputs at the same delay, we only consider the corresponding subset of functions. They are of the form  $(P_d^i)_k = P_d(s_{k-\tau})$  where  $P_d$  is the normalized Legendre polynomial of degree  $d$  and  $\tau \in \mathbb{N}$  is a delay. In Fig. 4, an estimate for the total capacity of an echo-state network is also shown, for which we consider the full set of functions.

For each considered function, we compute the capacity given by Eq. (6) by finding the weights of the optimal  $h$  as described previously in the Methods. We use finite input sequences, which in general can lead to an overestimation of the total capacity. As explained in Sec. 3.2 of the Supplementary Material of the work by Dambre et al.<sup>71</sup>, the effect of this can be reduced by fixing a threshold value and setting to 0 any capacity at or below the value. We use the same method.

In practice, only a finite number of degrees  $d$  and delays  $\tau$  can be considered for the numerical estimates, which can lead to an underestimation. We have found the following approach useful when searching for capacities larger than the threshold value. We fix a maximum degree  $d$  (for all results we have used 9) and for each degree we order the functions according to delay and find the capacity of  $N/2$  (rounded to an integer) functions at a time, until none of the  $N/2$  functions in the batch contribute to total capacity. All total capacities except the one for thermal encoding—where we have verified that some of the observables are in fact linearly dependent or almost—are very close to the theoretical maximum.

A different approach is used for the echo-state network, which we briefly describe in the following. We still fix the maximum degree as 9. For a fixed degree  $d$  we consider a sequence of delays  $\{\tau_1, \tau_2, \dots, \tau_d\}$  where the sequence is non-descending to avoid counting the same function multiple times. Then we form the product  $\prod_{\tau_i} P_{m(\tau_i)}(s_{k-\tau_i})$  over distinct delays of the sequence where  $m(\tau_i)$  is the multiplicity of  $\tau_i$  in the sequence. The lexical order of non-descending sequences of delays allows us to order the functions, which is exploited to generate each function just once. Furthermore, we have found that functions that contribute to total capacity seem to have a tendency to be early in the ordering, which makes it faster to get close to saturating the theoretical bound.

## Data availability

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

Received: 12 January 2021; Accepted: 15 February 2021;

Published online: 17 March 2021

## References

- Jordan, M. I. & Mitchell, T. M. Machine learning: trends, perspectives, and prospects. *Science* **349**, 255–260 (2015).
- Alpaydin, E. *Introduction to Machine Learning* (MIT press, 2020).
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. in *Advances in Neural Information Processing Systems*, (eds. Michael I. Jordan, Yann LeCun & Sara A. Solla) 1097–1105 (MIT Press, One Rogers Street Cambridge, MA, 2012).
- Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* **29**, 82–97 (2012).
- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
- Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
- Vinyals, O. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).
- Xu, X. et al. Scaling for edge inference of deep neural networks. *Nat. Electron.* **1**, 216–222 (2018).
- Linn, E., Rosezin, R., Tappertzhofen, S., Böttger, U. & Waser, R. Beyond von neumann-logic operations in passive crossbar arrays alongside memory operations. *Nanotechnology* **23**, 305205 (2012).
- Merolla, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).
- Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. Electron.* **1**, 22–29 (2018).
- Schuld, M., Sinayskiy, I. & Petruccione, F. The quest for a quantum neural network. *Quantum Inf. Process.* **13**, 2567–2586 (2014).
- Witteck, P. *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014).
- Dunjko, V. & Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Rep. Prog. Phys.* **81**, 074001 (2018).
- Steinbrecher, G. R., Olson, J. P., Englund, D. & Carolan, J. Quantum optical neural networks. *NPJ Quantum Inf.* **5**, 1–9 (2019).
- Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195–202 (2017).
- Torrontegui, E. & García-Ripoll, J. J. Unitary quantum perceptron as efficient universal approximator. *EPL (Europhysics Letters)* **125**, 30004 (2019).
- Beer, K. et al. Training deep quantum neural networks. *Nat. Commun.* **11**, 1–6 (2020).
- Killoran, N. et al. Continuous-variable quantum neural networks. *Phys. Rev. Res.* **1**, 033063 (2019).
- Amin, M. H., Andriyash, E., Rolfe, J., Kulchitsky, B. & Melko, R. Quantum boltzmann machine. *Phys. Rev. X* **8**, 021050 (2018).
- Adhikary, S., Dangwal, S. & Bhowmik, D. Supervised learning with a quantum classifier using multi-level systems. *Quantum Inf. Process.* **19**, 89 (2020).
- Liu, J. et al. An echo state network architecture based on quantum logic gate and its optimization. *Neurocomputing* **371**, 100–107 (2020).
- Shao, C. A quantum model of feed-forward neural networks with unitary learning algorithms. *Quantum Inf. Process.* **19**, 102 (2020).
- Brunner, D., Soriano, M. C., Mirasso, C. R. & Fischer, I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* **4**, 1364 (2013).
- Duport, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. *Opt. Express* **20**, 22783–22795 (2012).
- Vandoorne, K. et al. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.* **5**, 3541 (2014).
- Larger, L. et al. High-speed photonic reservoir computing using a time-delay-based architecture: million words per second classification. *Phys. Rev. X* **7**, 011015 (2017).
- Van der Sande, G., Brunner, D. & Soriano, M. C. Advances in photonic reservoir computing. *Nanophotonics* **6**, 561–576 (2017).
- Brunner, D., Soriano, M. C. & Van der Sande, G. *Photonic Reservoir Computing: Optical Recurrent Neural Networks* (Walter de Gruyter GmbH & Co KG, 2019).
- Dong, J., Rafayelyan, M., Krzakala, F. & Gigan, S. Optical reservoir computing using multiple light scattering for chaotic systems prediction. *IEEE J. Sel. Top. Quantum Electron.* **26**, 1–12 (2019).
- Torrejon, J. et al. Neuromorphic computing with nanoscale spintronic oscillators. *Nature* **547**, 428 (2017).
- Nakane, R., Tanaka, G. & Hirose, A. Reservoir computing with spin waves excited in a garnet film. *IEEE Access* **6**, 4462–4469 (2018).
- Verstraeten, D., Schrauwen, B., d’Haene, M. & Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Netw.* **20**, 391–403 (2007).
- Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
- Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
- Jaeger, H. & Haas, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
- Maass, W. & Markram, H. On the computational power of circuits of spiking neurons. *J. Comput. Syst. Sci.* **69**, 593–616 (2004).
- Grigoryeva, L. & Ortega, J.-P. Echo state networks are universal. *Neural Netw.* **108**, 495–508 (2018).
- Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **2**, 303–314 (1989).
- Hornik, K. et al. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
- Triefenbach, F., Demuyneck, K. & Martens, J.-P. Large vocabulary continuous speech recognition with reservoir-based acoustic models. *IEEE Signal Process. Lett.* **21**, 311–315 (2014).
- Pathak, J., Hunt, B., Girvan, M., Lu, Z. & Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102 (2018).
- Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks—with an erratum note. *German National Research Center for Information Technology GMD Technical Report*. Vol. 148, 13 (German National Research Center for Information Technology, 2001).
- Fujii, K. & Nakajima, K. Harnessing disordered-ensemble quantum dynamics for machine learning. *Phys. Rev. Appl.* **8**, 024030 (2017).
- Negoro, M., Mitarai, K., Fujii, K., Nakajima, K. & Kitagawa, M. Machine learning with controllable quantum dynamics of a nuclear spin ensemble in a solid. Preprint at <https://arxiv.org/abs/1806.10910> (2018).
- Nakajima, K., Fujii, K., Negoro, M., Mitarai, K. & Kitagawa, M. Boosting computational power through spatial multiplexing in quantum reservoir computing. *Phys. Rev. Appl.* **11**, 034021 (2019).

47. Chen, J. & Nurdin, H. I. Learning nonlinear input–output maps with dissipative quantum systems. *Quantum Inf. Process.* **18**, 198 (2019).
48. Ghosh, S., Opala, A., Matuszewski, M., Paterek, T. & Liew, T. C. Quantum reservoir processing. *NPJ Quantum Inf.* **5**, 35 (2019).
49. Chen, J., Nurdin, H. I. & Yamamoto, N. Temporal information processing on noisy quantum computers. *Phys. Rev. Appl.* **14**, 024065 (2020).
50. Kutvonen, A., Fujii, K. & Sagawa, T. Optimizing a quantum reservoir computer for time series prediction. *Sci. Reports* **10**, 14687 (2020).
51. Marcucci, G., Pierangeli, D., Pinkse, P. W., Malik, M. & Conti, C. Programming multi-level quantum gates in disordered computing reservoirs via machine learning. *Opt. Express* **28**, 14018–14027 (2020).
52. Govia, L. C. G., Ribeill, G. J., Rowlands, G. E., Krovi, H. K. & Ohki, T. A. Quantum reservoir computing with a single nonlinear oscillator. *Phys. Rev. Res.* **3**, 013077 (2021).
53. Ghosh, S., Krisnanda, T., Paterek, T. & Liew, T. C. H. Universal quantum reservoir computing. Preprint at <https://arxiv.org/abs/2003.09569> (2020).
54. Ghosh, S., Opala, A., Matuszewski, M., Paterek, T. & Liew, T. C. Reconstructing quantum states with quantum reservoir networks. *IEEE Transactions on Neural Networks and Learning Systems* (IEEE, 2020).
55. Martínez-Peña, R., Nokkala, J., Giorgi, G., Zambrini, R. & Soriano, M. C. Information processing capacity of spin-based quantum reservoir computing systems. *Cognit. Comput.* 1–12 (2020) <https://link.springer.com/article/10.1007/s12559-020-09772-y>.
56. Preskill, J. Quantum computing in the nisq era and beyond. *Quantum* **2**, 79 (2018).
57. Lloyd, S. & Braunstein, S. L. in *Quantum Information with Continuous Variables*, 9–17 (Springer, 1999).
58. Nielsen, M. A. & Chung, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
59. Gottesman, D., Kitaev, A. & Preskill, J. Encoding a qubit in an oscillator. *Phys. Rev. A* **64**, 012310 (2001).
60. Yukawa, M. et al. Emulating quantum cubic nonlinearity. *Phys. Rev. A* **88**, 053816 (2013).
61. Arzani, F., Treps, N. & Ferrini, G. Polynomial approximation of non-gaussian unitaries by counting one photon at a time. *Phys. Rev. A* **95**, 052352 (2017).
62. Sabapathy, K. K. & Weedbrook, C. On states as resource units for universal quantum computation with photonic architectures. *Phys. Rev. A* **97**, 062315 (2018).
63. Nokkala, J. et al. Reconfigurable optical implementation of quantum complex networks. *N. J. Phys.* **20**, 053024 (2018).
64. Cai, Y. et al. Multimode entanglement in reconfigurable graph states using optical frequency combs. *Nat. Commun.* **8**, 15645 (2017).
65. Chen, M., Menicucci, N. C. & Pfister, O. Experimental realization of multipartite entanglement of 60 modes of a quantum optical frequency comb. *Phys. Rev. Lett.* **112**, 120505 (2014).
66. Asavanant, W. et al. Generation of time-domain-multiplexed two-dimensional cluster state. *Science* **366**, 373–376 (2019).
67. Sandbo Chang, C. W. et al. Generating multimode entangled microwaves with a superconducting parametric cavity. *Phys. Rev. Appl.* **10**, 044019 (2018).
68. Lenzini, F. et al. Integrated photonic platform for quantum information with continuous variables. *Sci. Adv.* **4**, eaat9331 (2018).
69. Nielsen, W. H. P., Tsaturyan, Y., Möller, C. B., Polzik, E. S. & Schliesser, A. Multimode optomechanical system in the quantum regime. *Proc. Natl Acad. Sci.* **114**, 62–66 (2017).
70. Kollár, A. J., Fitzpatrick, M. & Houck, A. A. Hyperbolic lattices in circuit quantum electrodynamics. *Nature* **571**, 45–50 (2019).
71. Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. *Sci. Rep.* **2**, 514 (2012).
72. Ferraro, A., Olivares, S. & Paris, M. G. *Gaussian States in Quantum Information*. Napoli Series on physics and Astrophysics (Bibliopolis, 2005).
73. Adesso, G., Ragy, S. & Lee, A. R. Continuous variable quantum information: Gaussian states and beyond. *Open Syst. Inf. Dyn.* **21**, 1440001 (2014).
74. Nokkala, J., Maniscalco, S. & Piilo, J. Non-markovianity over ensemble averages in quantum complex networks. *Open Syst. Inf. Dyn.* **24**, 1740018 (2017).
75. Boyd, S. & Chua, L. Fading memory and the problem of approximating nonlinear operators with volterra series. *IEEE Trans. Circuits Syst.* **32**, 1150–1161 (1985).
76. Grigoryeva, L. & Ortega, J.-P. Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. *J Mach Learn Res* **19**, 892–931 (2018).
77. Dieudonné, J. *Foundations of Modern Analysis* (Read Books Ltd, 2011).
78. Bertschinger, N. & Natschläger, T. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Comput.* **16**, 1413–1436 (2004).
79. Jaeger, H. Short term memory in echo state networks. gmd-report 152. In *GMD-German National Research Institute for Computer Science* (Citeseer, 2002).
80. Ganguli, S., Huh, D. & Sompolinsky, H. Memory traces in dynamical systems. *Proc. Natl Acad. Sci.* **105**, 18970–18975 (2008).
81. Schrödinger, E. Der stetige Übergang von der mikro-zur makromechanik. *Naturwissenschaften* **14**, 664–666 (1926).
82. Glauber, R. J. Coherent and incoherent states of the radiation field. *Phys. Rev.* **131**, 2766 (1963).
83. Weedbrook, C. et al. Gaussian quantum information. *Rev. Modern Phys.* **84**, 621 (2012).
84. Lami, L. et al. Gaussian quantum resource theories. *Phys. Rev. A* **98**, 022335 (2018).
85. Soriano, M. C., Brunner, D., Escalona-Morán, M., Mirasso, C. R. & Fischer, I. Minimal approach to neuro-inspired information processing. *Front. Comput. Neurosci.* **9**, 68 (2015).
86. Araujo, F. A. et al. Role of non-linear data processing on speech recognition task in the framework of reservoir computing. *Sci. Rep.* **10**, 328 (2020).
87. Tanaka, G. et al. Recent advances in physical reservoir computing: a review. *Neural Netw.* **115**, 100–123 (2019).
88. Masada, G. et al. Continuous-variable entanglement on a chip. *Nat. Photon.* **9**, 316–319 (2015).
89. Takeda, S., Takase, K. & Furusawa, A. On-demand photonic entanglement synthesizer. *Sci. Adv.* **5**, eaaw4530 (2019).
90. Serafini, A. *Quantum Continuous Variables: A Primer of Theoretical Methods* (CRC press, 2017).
91. Menicucci, N. C. et al. Universal quantum computation with continuous-variable cluster states. *Phys. Rev. Lett.* **97**, 110501 (2006).
92. Hamilton, C. S. et al. Gaussian boson sampling. *Phys. Rev. Lett.* **119**, 170501 (2017).
93. Lvovsky, A. I. et al. Production and applications of non-Gaussian quantum states of light. Preprint at <https://arxiv.org/abs/2006.16985> (2020).
94. Albarelli, F., Genoni, M. G., Paris, M. G. A. & Ferraro, A. Resource theory of quantum non-gaussianity and wigner negativity. *Phys. Rev. A* **98**, 052350 (2018).
95. Takagi, R. & Zhuang, Q. Convex resource theory of non-gaussianity. *Phys. Rev. A* **97**, 062337 (2018).
96. Gu, M., Weedbrook, C., Menicucci, N. C., Ralph, T. C. & van Loock, P. Quantum computing with continuous-variable clusters. *Phys. Rev. A* **79**, 062318 (2009).
97. Nielsen, M. A. & Chuang, I. L. Programmable quantum gate arrays. *Phys. Rev. Lett.* **79**, 321 (1997).
98. Elman, J. L. Finding structure in time. *Cogn. Sci.* **14**, 179–211 (1990).
99. Pascanu, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 1310–1318 (JMLR, Inc. and Microtome Publishing (United States), 2013).
100. Konkoli, Z. On reservoir computing: from mathematical foundations to unconventional applications. In *Advances in Unconventional Computing*, 573–607 (Springer, 2017).
101. Canaday, D., Griffith, A. & Gauthier, D. J. Rapid time series prediction with a hardware-based reservoir computer. *Chaos* **28**, 123119 (2018).
102. Jaeger, H. *Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and The “echo state network” approach*, Vol. 5 (GMD-Forschungszentrum Informationstechnik Bonn, 2002).
103. Lukoševičius, M. in *Neural networks: Tricks of the Trade*, 659–686 (Springer, 2012).

## Acknowledgements

We acknowledge the Spanish State Research Agency, through the QUARESC project (PID2019-109094GB-C21 and -C22 /AEI/10.13039/501100011033) and through the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D (MDM-2017-0711). We also acknowledge funding by CAIB through the QUAREC project (PRD2018/47). The work of MCS has been supported by MICINN/AEI/FEDER and the University of the Balearic Islands through a “Ramon y Cajal” Fellowship (RYC-2015-18140). V.P. acknowledges financial support from the European Research Council under the Consolidator Grant COQCoN (Grant No. 820079).

## Author contributions

R.Z. and M.C.S. defined the research topic and supervised the research. J.N. performed the numerical simulations. R.M.P., J.N., and G.L.G. contributed to the theoretical aspects of the paper (demonstrations of Lemma 1 and Universality theorem). V.P. assessed the potential experimental implementation. All authors contributed to interpreting the results and writing the article.

**Competing interests**

The authors declare no competing interests.

**Additional information**

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s42005-021-00556-w>.

**Correspondence** and requests for materials should be addressed to J.N. or R.Z.

**Reprints and permission information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021