Gaussian states provide universal and versatile quantum reservoir computing

Johannes Nokkala,1 Rodrigo Martínez-Peña,1 Gian Luca Giorgi,1 Valentina Parigi,2 Miguel C. Soriano,1 and Roberta Zambrini1

1 Instituto de Física Interdisciplinar y Sistemas Complejos (IFISC, UIB-CSIC), Campus Universitat de les Illes Balears E-07122, Palma de Mallorca, Spain
2 Laboratoire Kastler Brossel, Sorbonne Université, CNRS, ENS-PSL Research University, Collège de France: CC74 4 Place Jussieu, 75252 Paris, France

(Dated: June 11, 2020)

We establish the potential of continuous-variable Gaussian states in performing reservoir computing with linear dynamical systems in classical and quantum regimes. Reservoir computing is a machine learning approach to time series processing. It exploits the computational power, high-dimensional state space and memory of generic complex systems to achieve its goal, giving it considerable engineering freedom compared to conventional computing or recurrent neural networks. We prove that universal reservoir computing can be achieved without nonlinear terms in the Hamiltonian or non-Gaussian resources. We find that encoding the input time series into Gaussian states is both a source and a means to tune the nonlinearity of the overall input-output map. We further show that reservoir computing can in principle be powered by quantum fluctuations, such as squeezed vacuum, instead of classical intense fields. Our results introduce a new research paradigm for quantum reservoir computing and the engineering of Gaussian quantum states, pushing both fields into a new direction.

I. INTRODUCTION

Machine learning (ML) covers a wide range of algorithms and modelling tools with automated data-processing capabilities based on experience. ML, with the prominent example of neural networks, has proven successful for tackling practical processing tasks that are unsuitable for conventional computer algorithms. 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. 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 architecture. Recent extensions of neural networks and other ML techniques based on quantum systems aim to offer and identify novel capabilities. 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 and spintronics, and has the potential to be extended to the quantum regime.

Reservoir computing exploits the dynamics of a nonlinear system—the reservoir—for information processing of time dependent inputs. 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 without any apparent loss in computational power. In practice, reservoir computers have achieved state-of-the-art performance in tasks such as continuous speech recognition and nonlinear time series prediction thanks to their intrinsic memory.

Here, we put forward the framework for quantum reservoir computing (QRC) with continuous variables, in bosonic reservoirs given by harmonic networks with Gaussian states. This proposal could be implemented in tailored multimode optical parametric processes that already realize versatile and large entanglement networks in several experimental platforms. Multi-mode quantum states of few modes have also been recently implemented in superconducting and optomechanical platforms. An optical implementation would have 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. In any platform, an advantage of RC is that the reservoir Hamiltonian can have even random parameters not needing fine tuning. Therefore in the quantum regime these systems are well suited for NISQ (noisy intermediate-scale quantum) technologies. The general theoretical framework we introduce here explores the utility of the method spanning across classical and quantum states and is applicable to several physical reservoirs and any temporal task.

The restriction to Gaussian dynamics brings the model within reach of state-of-the-art experimental platforms, but one might expect it to have very modest information processing capabilities. Indeed, previous proposals for recurrent neural networks realized with continuous variable quantum systems have demonstrated a gate set universality for quantum computing and non-Gaussian gates as a source of nonlinearity, a crucial resource for nontrivial information processing. While in principle powerful, the required gates represent a formidable and so far unsolved engineering challenge. Surprisingly, we find that

---

\* johannes@ifisc.uib-csic.es
universal time series processing, i.e. universal RC, is possible with just Gaussian resources. In stark contrast to gate-set universal quantum computing\cite{Harrow2009}, universal time series processing of RC refers to the capability to approximate as accurately as one wants functions of time-varying input signals\cite{Viola1999,Viola2000}. The here demonstrated universality of RC based on linear reservoirs and with Gaussian states is valid both with classical and quantum resources.

Universal time series processing can be achieved in Gaussian QRC by combining the outputs from a finite number of different networks (which can be thought of as a large network with many components) with a polynomial readout function. We further show that even in the case of a readout function which is linear in the observables of a single component network, Gaussian QRC can provide an overall non-linear input-output map by exploiting the input encoding of Gaussian quantum states. Interestingly, this also provides a mean to tune the information processing from fully linear to strongly nonlinear even when keeping the network fixed, enabling high versatility and a further advantage to this proposal of QRC. The generality of the performance analysis is achieved considering the information processing capacity (IPC)\cite{Mandel2005} allowing for a task-independent assessment of time series processing and used for the first time for a quantum system.

A step forward towards the exploitation of the quantumness of the system is here accomplished by designing a RC scheme that fully works with quantum fluctuations. Indeed, we demonstrate that working in the limit of vanishing amplitudes leads to no loss in computational power: universal time series processing is achieved in Gaussian QRC by encoding the input into squeezed vacuum. Different computation approaches are known to be able to operate with classical\cite{Kwee2008,Kwee2009,Rudolph2009} and/or quantum advantage\cite{Viola2000,Viola2005}. A prominent example also realized in linear optics is boson sampling, classically hard but not universal\cite{Harrow2009,Vinci2011}. Linear optics also enables universal quantum computing when provided with single-photon detection or states\cite{Lanyon2010,Klimov2011}. The prospects and resources required for a quantum advantage is outside the scope of the present work—they are indeed major open questions in quantum ML and, to a lesser extent, quantum computing. The main goals of this work are (i) the demonstration that QRC operating with Gaussian resources and even with quantum fluctuations is universal for time series processing and (ii) to establish the versatility of this platform beyond specific tasks through the IPC.

II. RESULTS

In the following, after introducing QRC with Gaussian states (Sect. II A), we demonstrate that this novel QRC approach possesses universal approximation properties (Sect. II B). We then show numerical evidence of its performance and versatility enabled by input encoding in different quantum states, as one can control the degree of nonlinearity of the reservoir by tuning the input (Sections II C and II D). Finally, we illustrate that the computational power of this approach still holds in the limit of vanishing amplitudes, using squeezed vacuum (Sect. II E).

A. The model

We consider a network of interacting quantum harmonic oscillators acting as the reservoir for QRC, with spring-like interaction strengths $g_{ij}$. The Hamiltonian of such a system can be conveniently described in terms of the Laplacian matrix $L$ 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

$$H = \frac{p^T P}{2} + \frac{q^T (\Delta_\omega^2 + L) q}{2}$$

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

The scheme for using this network for reservoir computing is shown in Fig. 1. The input sequence $s = \{s_{k-1}, s_k, s_{k+1}, \ldots\}$, where $s_k \in \mathbb{R}^n$ 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. 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 $x^T = \{q_1, p_1, q_2, p_2, \ldots\}$ be the vector of network operators and let $x_k$ be the form of this vector at timestep $k$, after input $s_k$ 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 $x_k$ is related to $x_{k-1}$ by

$$x_k = S(\Delta t) \left( P_R x_{k-1} \oplus x^A_k \right)$$

where $P_R$ drops the ancillary operators from $x_{k-1}$ (reservoir projector, orthogonal to the ancilla vector) and $x^A_k$ is the vector of ancillary operators conditioned on input $s_k$, while $S(\Delta t) \in \text{Sp}(2N, \mathbb{R})$ is the symplectic matrix induced by the Hamiltonian in Eq. (1) and time $\Delta t$ (see, e.g.,\cite{Shillcock2018}). The dynamics of reservoir operators $x^R_k = P_R x_k$ is conveniently described dividing $S(\Delta t)$ into blocks as

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

where $A$ is $2(N-1) \times 2(N-1)$ and $D$ is $2 \times 2$. Now the formal definition of the proposed reservoir computer
FIG. 1. Reservoir computing scheme. a The overall input-output map. The input sequence \( s \) 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, \( x^A \). The rest of the network—taken to be the reservoir—has operators \( x^R \). Network dynamics maps the ancillary states into reservoir states, which are mapped to elements of the output sequence \( o \) by a trained function \( h \) of reservoir observables. b The corresponding circuit. The reservoir interacts with each ancillary state through a symplectic matrix \( S(\Delta t) \) induced by the network Hamiltonian \( H \) during constant interaction time \( \Delta t \). Output \((o_k)\) at timestep \( k \) is extracted before each new input. \( x^A_k \) are the ancillary operators conditioned on input \( s_k \) and \( x^R_k \) are the reservoir operators after processing this input. c Wigner quasiprobability distribution of ancilla encoding states in phase space. Input may be encoded in coherent states using amplitude \( |\alpha| \) and phase \( \arg(\alpha) \), or in squeezed states using squeezing parameter \( \tau \) and phase of squeezing \( \varphi \), or in thermal states using thermal excitations \( n_{th} \).

reads

\[
\begin{align*}
\dot{x}^R_k &= Ax^R_{k-1} + Bx^R_k, \\
o_k &= h(x^R_k),
\end{align*}
\]

where \( h \) maps the reservoir operators to elements of the real output sequence \( o = \{s_0, o_{k-1}, o_k, o_{k+1}, \ldots\} \).

For Gaussian states, the full dynamics of the system conditioned by the sequential input injection is entailed in the first moments vector \((x^R_k)^T\) and covariance matrix \(\sigma(x^R_k)\). The values at step 0, given a sequence of previous \( m \) inputs \( s = \{s_{m+1}, \ldots, s_{1}, s_0\} \) encoded in the corresponding ancilla vectors, is obtained through repeated application of Eqs. (4) and reads

\[
\begin{align*}
\sigma(x^R_0) &= A^m\sigma(x^R_{-m})(A^T)^m \\
\langle x^R_0 \rangle &= A^m\langle x^R_{-m} \rangle + \sum_{j=0}^{m-1} A^jB\sigma(x^A_{-j})B^T(A^T)^j,
\end{align*}
\]

where \(\sigma(x^R_{-m})\) and \(\langle x^R_{-m} \rangle\) 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 \(x^A\). 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 exponential displacement \(\alpha\), squeezing degree \(\kappa\) and phase \(\varphi\), and thermal excitations \(n_{th}\) (see Methods [IVB, Eqs. (6)]). Unless stated otherwise, we assume that the input is encoded into the ancilla by setting each of these parameters to some fixed continuous function of the input. Finally, the output is taken to be either linear or polynomial in either the elements of \(\sigma(x^R_k)\) or \(\langle x^R_k \rangle\). We will next show that the introduced model not only satisfies the requirements for reservoir computing, but notably that it is even universal for reservoir computing.

B. Universality for reservoir computing

To begin with, we show that instances of the model defined by Eqs. (1) and the dependency of \(x^R_k\) on \(s_k\) can be used for reservoir computing, 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 [IV A], the goal is more formally to reproduce a time-dependent function \(f(t) = F[\ldots, s_{m-2}, s_{m-1}, s_m]\), associated with given input \(s\) and functional \(F\) from the space of inputs to reals. Consequently, we say that the system can be used for reservoir computing if there is a functional from the space of inputs to reals that is both a solution of Eqs. (4) and sufficiently well-behaved to facilitate learning of different tasks. These two requirements are addressed by the echo state property (ESP) and the fading memory property (FMP) respectively. 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 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 Eqs. (6), 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 \(\rho(A) < 1\), where \(\rho(A)\) is the spectral radius of matrix \(A\). Therefore, \(\rho(A) < 1\) is a necessary condition for having ESP and FMP. The following lemma (proven in Supplementary Information) states that it is also sufficient when we introduce the mild constraint of working with uniformly bounded subsets of the full input space—briefly, this means that 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 \( \sigma(x^R_k) \) or \( \langle x^R_k \rangle \). The corresponding reservoir system has both ESP and FMP if and only if the matrix \( A \) in Eqs. (4) fulfills \( \rho(A) < 1 \).

This is the sought condition for reservoir computing 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 \( A \).

We now turn our attention to RC 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 Eqs. (4) is universal in the following sense. Essentially, for any element \( F \) in a class of fading memory functionals that will be given shortly, 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 full version in the Supplementary Information) summarizes our analysis of the model described in Eqs. (4).

Universality Theorem (simplified). Suppose the input sequence \( s \) is uniformly bounded. Fix \( \Delta t \) and consider the corresponding instances of the reservoir system Eqs. (4) for which \( \rho(A) < 1 \). By virtue of Lemma 1 they all have ESP and FMP, and therefore are associated with a family of fading memory functionals. Let this family be called \( Q \). Let \( A(Q) \) be a polynomial algebra of them and the constant functional. Since \( A(Q) \) has separability, any causal, time-invariant fading memory functional \( F \) can be uniformly approximated by its elements, that is to say it is universal. In particular, universality holds for the subfamilies \( Q_{\text{thermal}} \), \( Q_{\text{squeezed}} \), and \( Q_{\text{phase}} \), that correspond to thermal, squeezed and phase encoding respectively.

We sketch the main ingredients of the proof. Since the model admits arbitrarily small values of \( \rho(A) \), there are instances where \( \rho(A) < 1 \); therefore \( Q \) is not empty. We show that the associated algebra \( A(Q) \) has separability. Since the space of inputs is taken to be uniformly bounded, we may invoke the Stone-Weierstrass Theorem and the claim follows. Full proof and additional details are in Supplementary Information.

We note that unlike ESP and FMP, separability depends explicitly on the input encoding. In Supplementary Information we show separability for three different encodings of the input to elements of \( \sigma(x^R_k) \): thermal \((n_{th})\), squeezing \((r)\) 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.

C. Controlling performance with input encoding

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 \( o \) and target output. For simplicity, we will also take inputs to be sequences of real numbers, rather than sequences of vectors.

First of all, we might ask 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. 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 \( L \) 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 \( \rho(A) < 1 \). Having FMP says nothing about what kind of functions the memory consists of, however.

Is there nonlinear memory? It is evident that the first of Eqs. (4) is linear in reservoir and ancilla operators, but the encoding is not necessarily linear because of the way ancilla operators \( x^d_k \) depend on the input. For single-mode Gaussian states (see Eqs. (7) in Methods B), 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 Eqs. (10) in Methods B).
FIG. 2. **Nonlinear information processing with a generic reservoir.** In panels a and b the targets are \((P_1)_k = s_k\) and \((P_3)_k = (15s_k - 70s_k^3 + 63s_k^5)/8\) while \(h\) is a trained linear function of \((x_1^R)\). Encoding the input to either magnitude of displacement \(|\alpha|\) or phase \(\arg(\alpha)\) of the coherent state is compared. The former is able to reproduce only the linear target \(P_1\) while the latter has good performance with \(P_3\), confirming that some nonlinear tasks are possible with linear \(h\). In panels c and d we fix the encoding to \(|\alpha|\) and consider the parity check task \((PC@\tau = 1, 3)\) which requires products of the input at different delays \(\tau\). These terms can be introduced by a polynomial \(h\) (degree= 2, 4); increasing its degree allows the task to be reproduced 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 time steps after training (see Appendices C and D for additional details).

The performance of Gaussian QRC 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 \(2(N - 1)\) observables and a bias term; see Appendices C and D for details. We consider nonlinear tasks in Fig. 2. In panels a and b we take the output function \(h\) to be a linear function of \((x_1^R)\) and inputs \(s_k\) to be uniformly distributed in \([-1, 1]\), and consider two different encodings of the input into the ancilla \((x_1^R)\); 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 a.

Nonlinearity of reservoir memory is not without limita-

D. Information processing capacity

Besides providing nonlinearity, input encoding also facilitates its versatile tuning. 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)\(^{20}\) of the reservoir. The IPC generalizes the linear memory capacity\(^{20}\) often considered in reservoir computing 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 reservoir computing.

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 \(s'\) and \(s\) of random inputs drawn independently from some fixed probability distribution \(p(s)\). The first sequence \(s'\) is used to initialize the reservoir; observables are recorded only when the rest of the inputs \(s\) are encoded. The capacity of the reservoir \(X\) to reconstruct \(z\) given \(s'\) and \(s\) is defined to be

\[
C_{s',s}(X, z) = 1 - \frac{\min_h \sum_k (z_k - o_k)^2}{\sum_k z_k^2}
\]
FIG. 3. Control of nonlinearity of reservoir memory via input encoding. Here we set $|\alpha| \to (1 - \lambda)(s_k + 1) + \lambda$, $\arg(\alpha) \to 2\pi \lambda$ where the input is $s_k \in [-1, 1]$. 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 to 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.

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 [20], the sum of capacities $C_{s', s}(X, z)$ 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. Importantly, infinite sequences $s'$, $s$ 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. $p(s)$ include Legendre polynomials $P_1$ and $P_2$, appearing in Fig. 2, as well as their delayed counterparts. Further details are given in Methods E.

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 $\langle x_k^{|R} \rangle$ and address the capability to have Gaussian QRC operating with different linear and non-linear capabilities by varying the input encoding into a coherent ancillary state from amplitude to phase $|\alpha| \to (1 - \lambda)(s_k + 1) + \lambda$, $\arg(\alpha) \to 2\pi \lambda$ 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 to tune the reservoir to the task at hand. Remarkably, this can be done without changing neither the parameters of the Hamiltonian [1] (that is, the reservoir system) nor the observables extracted as output in $h$. Lemma 1 ensures that full capacity is maintained for all values of $\lambda$.

E. From classical to quantum RC

In the previous section we considered coherent states for the ancilla, with input encoding into $|\alpha|$ and $\arg(\alpha)$. In the limit of large amplitudes $|\alpha| \gg 1$, coherent states allow for a description of the harmonic network equivalent to the classical one, with field operator expectation values corresponding, for instance, to classical laser fields $|\alpha|^\frac{1}{2}$.

In the limit $|\alpha| \to 0$, on the other hand, $\langle x_k^{|R} \rangle$ cannot be used for reservoir computing. While the classical approach encoding, for instance in optical signals, is not applicable for vanishing fields, we may ask if similar information processing can be done harnessing quantum fluctuations. We propose to deal with elements of the covariance matrix $\sigma(\langle x_k^{|R} \rangle)$ and squeezed (encoding to $r$, $\varphi$) or, as a further benchmark, thermal (encoding to $n_{th}$) states for the ancilla. Since we rely on non-classical features like quantum fluctuations when encoding to squeeze, it represents a non-classical proposal, i.e. a quantum approach to reservoir computing with Gaussian states.

We compare the classical and quantum approaches in Fig. 4 where we show how the capacity of the reservoir breaks down for each of the aforementioned encodings. We also include the case of an echo state network (ESN, see Methods E) of same output size, a classical reservoir computer based on a recurrent neural network [13] for comparison. We separate the contributions to total capacity according to degree to appreciate linear and nonlinear reservoir information capacity and take $s_k$ to be uniformly distributed in $[-1, 1]$. Cases $|\alpha|$ and $\arg(\alpha)$ are as in Fig. 2 and correspond to Fig. 3 for $\lambda = 0$ and $\lambda = 1$, respectively. For them we take $h$ to be linear in $\langle x_k^{|R} \rangle$. For phase encoding of coherent states we can see significant contributions to total capacity from higher degrees, with degrees 7 and higher also playing a small role, in spite of the small size of the network.

Let now $|\alpha| \to 0$. We find that the linear memory found in the classical approach can be recovered in the quantum one by setting $n_{th} \to s_k + 1$ and taking $h$ to be a linear function of $\sigma(\langle x_k^{|R} \rangle)$; here we use just the elements in the first row of the covariance matrix to have the same number of observables as before. Interestingly, also the nonlinear memory found for $\arg(\alpha)$ in the classical approach is reproduced in the quantum one by setting $r \to 1$, $\varphi \to 2\pi s_k$. We may also set $r \to s_k + 1$, $\varphi \to 0$, which gives a somewhat more even split to linear and
nonlinear memory in the quantum approach. Furthermore, by virtue of Universality Theorem the cases \( n_t \), \( r \) and \( \varphi \) can all be used for universal reservoir computing.

In conclusion, we find not only that is possible to perform reservoir computing by exploiting quantum fluctuations in the case of vanishing first moments, but also that reservoir computing with quantum fluctuations can be just as powerful.

III. DISCUSSION

Reservoir computing realized with physical systems is a new and rapidly developing field,\(^{20}\) with the extension to quantum systems even more so. Aside from a handful of pioneering works with spins,\(^{33,36,38}\) the potential and viability of quantum reservoir computing is still largely unknown. In our contribution, we propose a novel approach based on continuous variable systems, which despite using only a linear system and Gaussian states is universal for time series processing both with classical and quantum resources. In Lemma 1 we present a simple and mild condition for systems of the general form given by Eqs. (1) to facilitate reservoir computing. We remark that Lemma 1 applies for all Hamiltonians quadratic in position and momentum operators. This implies that it applies to any optical platform with linear optics and squeezing processes. It is also not dependent on input, at variance with spin-based reservoirs where there is a condition on the input-dependent dynamical map,\(^{35}\) meaning in particular that a fixed reservoir of spins may be amenable for reservoir computing for certain input sequences but not for others. While Universality Theorem was derived starting from the Hamiltonian of Eq. (1), it implies that any extended family resulting for example from the inclusion of some non-Gaussian states is likewise universal. As non-Gaussian quantum features have been found to be beneficial for quantum machine learning purposes before,\(^{39}\) they may be expected to be useful in the present approach as well; this will be explored in future works. It should be stressed that the framework provides considerable engineering freedom; any non-Gaussian operations are potentially useful.

An advantage of the proposed model is easily tunable nonlinear memory, which we have characterized in detail by using the information processing capacity measure. This is an important property since nonlinearity has already proven to be beneficial to tackle different computational tasks in classical reservoir computing.\(^{50,51}\) We have observed that the elements of the reservoir covariance matrix or first moments vector cannot depend on products of the input at different delays, however; how to introduce such nonlinearities in the reservoir state is an important avenue of further research. We have circumvented this issue by making use of nonlinear readouts in a similar manner as it is currently being used in certain photonic implementations of reservoir computing.\(^{29}\)

The main questions tackled when harnessing quantum systems for computation or information processing purposes are usually about universality and advantage over classical alternatives. In the context of time series processing in reservoir computing framework, universality has previously been shown with spins,\(^{32}\) and now with minimal resources in the case of continuous variable systems. Indeed, in principle just Gaussian states, linear dynamics and Gaussian measurements are enough for this purpose also in the quantum regime where input is encoded in quantum fluctuations using squeezed vacuum. We consider the possibility of a quantum advantage to be open when it comes to time series processing, as no formal and realistic analysis has been carried out. The minimal resources considered here lend themselves to efficient classical simulation\(^{22}\) but, as we have argued, have been used to lay the theoretical foundation for further studies exploring the benefits of non-Gaussian states or operations.

Importantly, our results pave the way for new experimental investigations of quantum machine learning in reconfigurable optical implementations of quantum complex networks.\(^{57}\) Such platforms are currently in development\(^{50,52}\) mainly for the generation of nonclassical states used as a resource in measurement based quantum computing and here we point them out as potential candidates for quantum reservoir computing; their intrinsic resilience to decoherence and high potential for scalability make them very promising indeed.

While it is convenient that the reservoir Hamiltonian is time independent and can have random parameters, injection of input and extraction of output are challenges to be addressed for experimental realizations. Indeed continuous processing of input is still a challenge also for spin-based implementations of quantum reservoir computing.\(^{24,53}\) Experimental realization of a qua-
tum reservoir in NMR for machine learning was recently reported based on an ensemble of reservoirs in order to overcome the problem of measurement back-action. While certainly a hurdle to be overcome, measurements may also be a potential source of computational power. We speculate that the erasure of information due to a measurement could enhance the fading memory of the system if needed. Another possibility is that measurements can be a source of nonlinearity. Indeed, this enables universal quantum computation even for Gaussian states if any single-mode non-Gaussian measurements are available and could have a similar impact for universal time series processing.

Perhaps one of the most interesting future research avenues would be to consider processing quantum signals with a recurrent quantum neural network in the framework of reservoir computing. This could also be expected to shift the overhead from input preprocessing in favor of quantum reservoirs. While a quantum reservoir can quite naturally process an input sequence of quantum states, the corresponding learning theory is at the moment missing; tasks of interests could include training the reservoir to simulate given quantum circuits or training reservoir observables to reveal nontrivial information about a sequence of quantum states. Static versions of such tasks carried out in feedforward, as opposed to recurrent, architecture have been considered and could be used as a starting point.

IV. METHODS

A. 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 input to the neural network nodes depends on the temporal signals to be processed but also on the previous states of the network nodes, providing the needed memory. Unfortunately, such recurrent neural networks are notorious for being difficult to train. 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.

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. We will also briefly discuss the application of the framework to quantum reservoirs.

Reservoir computers. We consider sequences of discrete-time data \( s = \{ \ldots, s_{t-1}, s_t, s_{t+1}, \ldots \} \), where \( s_t \in \mathbb{R}^n \), \( n \) is the dimension of the input vector and \( t \in \mathbb{Z} \). Let us call the space of input sequences \( \mathcal{U}_n \) such that \( s \in \mathcal{U}_n \). Occasionally, we will also use left and right infinite sequences defined as \( \mathcal{U}_n^- = \{ s = \{ s_{t-1}, s_{t-2}, \ldots, s_0, s_t \} | s_t \in \mathbb{R}^n, s_i \in \mathbb{Z} \} \) and \( \mathcal{U}_n^+ = \{ s = \{ s_0, s_1, s_2, \ldots | s_i \in \mathbb{R}^n, i \in \mathbb{Z} \} \), respectively. Formally, a reservoir computer may be defined by the following set of equations:

\[
\begin{align*}
    x_k &= T(x_{k-1}, s_k) \\
    o_k &= h(x_k),
\end{align*}
\]

where \( T \) is a recurrence relation that transforms input sequence elements \( s_k \) to feature space elements \( x_k \)—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 \( \mathbf{s} \) 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 wasn’t used in training.

The general nature of Eqs. 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. In such a scenario time series \( s \)—often after suitable pre-processing—drives the dynamics given by \( T \) while \( x_k \) is the reservoir state. A readout mechanism that can inspect the reservoir state should be introduced to implement function \( h \). The appeal of physical reservoir computing 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.

Temporal maps and tasks. Online time series processing—what we wish to do with the system in Eqs. 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 \((M[s])_t\) may only depend on \( s_k \) 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, \((M[s])_t = F(s_{t-1}, s_{t-2}, \ldots, s_{t-n})\) for any \( t \) for some fixed \( F : \mathcal{U}_n \rightarrow \mathbb{R}^m \). We will later refer to such \( F \) as functionals. When \( M \) is given, fixing \( s \) induces a time-dependent function that we will denote by \( f \), defined by \( f(t) = F(s_{t-1}, s_{t-2}, \ldots, s_{t-n}) \).

To process input \( s \) into \( o \) 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 time step.

Properties of useful reservoirs. In general, \( o_k \) in Eqs. 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)\textsuperscript{15} requires that for any reference time $t$, $x_{t} = E(\ldots,s_{t-2},s_{t-1},s_{t})$ for some function $E$, that is to say at the limit of infinitely many inputs the reservoir state should become completely determined by the inputs. 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 second. 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)\textsuperscript{22} 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 Information. It can be shown that FMP imposes a form of continuity to the overall input-output maps that form of continuity to the overall input-output maps that are realized by reservoirs. It can be shown that ESP and FMP, with computational power constrained by bounds introduced in\textsuperscript{60} Moreover, a reservoir class can be shown to be universal much like before. We also note that for quantum reservoirs, ensemble measurements have been proposed to recover the output without disturbing the reservoir state.\textsuperscript{13,15,16}

B. The explicit forms of the covariance matrix and first moments vector

For a single mode Gaussian state with frequency $\Omega$, they read

$$\begin{align*}
\langle x | \rangle &= (n_{th} + \frac{1}{2}) \left( (y + z_{\cos}) \Omega^{-1} z_{\sin} \right), \\
\langle x^{a} | \rangle &= \left( |a| \cos (\arg(a)) \sqrt{2\Omega^{-1}} \right),
\end{align*}$$

where $y = \cos(2r)$, $z_{\cos} = \cos (\varphi) \sinh (2r)$ and $z_{\sin} = \sin (\varphi) \sinh (2r)$. Here, $n_{th}$ controls the amount of thermal excitations, $r$ and $\varphi$ control the magnitude and phase of squeezing, respectively, and finally $|a|$ and $\arg(a)$ 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 $s = \{s_{-m+1}, \ldots, s_{-1}, s_{0}\}$ and each input $s_{k}$ is encoded to all degrees of freedom as $n_{th} \rightarrow n_{th}(s_{k})$, $r \rightarrow r(s_{k})$, $\varphi \rightarrow \varphi(s_{k})$, $|a| \rightarrow |a(s_{k})|$, and $\arg(a) \rightarrow \arg(a(s_{k}))$. Then from Eqs. (1) it follows that

$$\begin{align*}
\left[ \sigma(x_{R}) - A^{m} \sigma(x_{R}^{m}) (A^{\top})^{m} \right]_{i,j} &= \sum_{k=0}^{m-1} a_{ij}^{k} a_{k}^{ij} n_{th}(s_{k}) (\cosh (2r(s_{k})) + b_{ij}^{k} \cos (\varphi(s_{k})) + c_{ij}^{k} \sin (\varphi(s_{k}))) \sinh (2r(s_{k}))), \\
\left[ \langle x_{R} | \rangle - A^{m} \langle x_{R}^{m} | \rangle \right]_{i} &= \sum_{k=0}^{m-1} |a(s_{k})| (a_{k}^{i} \cos (\arg(a(s_{k}))) + b_{k}^{i} \sin (\arg(a(s_{k}))))
\end{align*}$$

where $a_{ij}^{k}$, $b_{ij}^{k}$, $c_{ij}^{k}$, $a_{k}^{i}$, and $b_{k}^{i}$ are constants depending on the Hamiltonian in Eqs. (1) and $\Delta t$. That is to say the part of the observables independent of the initial conditions $x_{R}^{m}$ are linear combinations of $n_{th}(s_{k})$ and $|a(s_{k})|$, while the dependency on $r(s_{k})$, $\varphi(s_{k})$ and $\arg(a(s_{k}))$ 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.

C. The networks used in numerical experiments

We have used a chain of $N = 8$ oscillators for all results shown in Fig. 2 and Fig. 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. We have set $\Delta t = 0.6$ which is close to a local minimum of $\rho(A)$; in general many values of $\Delta t$ can achieve $\rho(A) < 1$.
and can therefore be expected to 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 uniform interaction strengths $g = 0.1$. Here the choice of the ancilla does not matter because the network is completely symmetric. For each realization all interaction strengths are independently scaled by a random factor uniformly distributed in the closed interval $[0.1, 1.9]$ to break the symmetry; we point out that otherwise it seems impossible to achieve $\rho(\mathbf{A}) < 1$. A suitable value for $\Delta t$ is then found as follows. We consider a large number of values and find the corresponding $\rho(\mathbf{A})$ for each. If the smallest found spectral radius satisfies $\rho(\mathbf{A}) < 0.99$—which is the case—we choose the corresponding $\Delta t$ and proceed; otherwise we draw new random weights and try again.

D. 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 $2(N - 1)$ reservoir observables used to form the output. Here $N - 1$ is just the number of reservoir oscillators and the factor of 2 accounts for using all first moments, or alternatively a row of the covariance matrix. We wish to find a readout function $h$ that minimizes

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

i.e. the squared error between target output $\hat{o}$ 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 $M \times 2(N - 1)$ matrix $\mathbf{X}$. We introduce a constant bias term by extending the matrix with a unit column so that the final dimensions of $\mathbf{X}$ are $M \times (2N - 1)$. Now we may write $o_k = h(\mathbf{X}_k) = \sum_{i}^{N} W_i x_{ki}$ 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 $\mathbf{XW} = \mathbf{o}^T$. To minimize (11), we set

$$\mathbf{W} = \mathbf{X}^+ \hat{o}^T$$

where $\mathbf{X}^+$ is the Moore-Penrose inverse of $\mathbf{X}$, and $\mathbf{W}$ 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.

E. The used echo state network

An echo state network (ESN) is used for some of the results shown in Fig. 4. We use 14 observables and a constant bias term to form the output with the oscillator networks. To have a fair comparison with an 8 nodes harmonic network we set the size of the ESN to $N = 14$ nodes (see Methods D), 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 $\mathbf{x}_k = \tanh(\beta \mathbf{Wx}_{k-1} + \mathbf{i} \mathbf{w} s_k)$ where $\mathbf{W}$ is a random $N \times N$ matrix, $\mathbf{w}$ a random vector of length $N$, $\beta$ and $\mathbf{i}$ 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 $\mathbf{i}$ 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 $\mathbf{i} = 1$. The readout function is a linear function of the elements of $\mathbf{x}_k$ and training is done as with the oscillator networks.

F. 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 system of functions and using infinite sequences $s'$ 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 [20], 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 [60]—it should be stressed that in general, changing the input changes which functions are orthogonal. Since $\sigma(\mathbf{x}^R)$ and $\langle \mathbf{x}^R \rangle$ can only depend on products of the input at the same delay, we only consider the corresponding subset of functions. They are of the form $(P_d^T)_{k} = P_d(s_{k-\tau})$ 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. (7) by finding the optimal $h$ as described in Methods D. We use finite input sequences, which in general can lead to an overestimation of total capacity. As explained in the Supplementary Material of [60], 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.

Obviously only a finite number of degrees \( d \) and delays \( \tau \) can be considered for the numerical estimates, which can lead to an underestimation. In practice we have found the following approach useful when searching for capacities larger than the threshold value. We fix a maximum degree; for all results we have used 9. For each degree at most this, 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. The viability of this approach can be verified by observing that all found total capacities are very close to the theoretical bound which we have normalized to 1 in the figures.

A different approach is used for the echo state network which we will briefly describe. We still fix the maximum degree as 9. For a fixed degree \( d \) we consider a sequence of delays \( \{\tau_1, \tau_2, \ldots, \tau_d\} \) where the sequence is non-descending to avoid counting the same function multiple times. Then we form the product \( \prod_{\tau_i} P_n(\tau_i) \) 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**

Data is available from the corresponding author at reasonable request.
Verstraeten, D., Schrauwen, B., dHaene, M. & Govia, L., Ribeill, G., Rowlands, G., Krovi, H. & Ohki, Dong, J., Rafayelyan, M., Krzakala, F. & Gigan, S. Optic reservoir computing using multiple light scattering for chaotic systems prediction. IEEE Journal of Selected Topics in Quantum Electronics 26, 1–12 (2019).

Torrejon, J. et al. Neuromorphic computing with nanoscale spintronics oscillators. Nature 547, 428 (2017).

Triefenbach, F., Demuynck, K. & Martens, J.-P. Large vocabulary continuous speech recognition with reservoir-based acoustic models. IEEE Signal Processing Letters 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. Physical review letters 120, 024102 (2018).

Jaeger, H. The echo state approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148, 13 (2001).

Nokkala, J. et al. Reconfigurable optical implementation of quantum complex networks. New Journal of Physics 20, 053024 (2018).

Cai, Y. et al. Multimode entanglement in reconfigurable graph states using optical frequency combs. Nat. Commun. 8, 15645 (2017).

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).

Asavanant, W. et al. Generation of time-domain-multiplexed two-dimensional cluster state. Science 366, 373–376 (2019).

Sandbo Chang, C. W. et al. Generating multimode entangled microwaves with a superconducting parametric cavity. Phys. Rev. Applied 10, 044019 (2018).

Bienfait, A. et al. Magnetic resonance with squeezed microwaves. Phys. Rev. X 7, 041011 (2017).

Nielsen, W. H. P., Tsaturyan, Y., Møller, C. B., Polzik, E. S. & Schliesser, A. Multimode optomechanical system in the quantum regime. Proceedings of the National Academy of Sciences 114, 62–66 (2017).

Kollár, A. J., Fitzpatrick, M. & Houck, A. A. Hyperbolic lattices in circuit quantum electrodynamics. Nature 571, 45–50 (2019).

Preskill, J. Quantum computing in the nisq era and beyond. Quantum 2, 79 (2018).

Maass, W. & Markram, H. On the computational power of circuits of spiking neurons. Journal of computer and system sciences 69, 593–616 (2004).

Grigoryeva, L. & Ortega, J.-P. Echo state networks are universal. Neural Networks 108, 495–508 (2018).

Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. Scientific reports 2, 514 (2012).

Nielsen, M. A. & Chung, I. L. Quantum Computation and Quantum Information (Cambridge University Press, 2000).

Boixo, S. et al. Characterizing quantum supremacy in near-term devices. Nature 560, 595–600 (2018).

Arute, F. et al. Quantum supremacy using a programmable superconducting processor. Nature 574, 505–510 (2019).

Aaronson, S. & Arkhipov, A. The computational complexity of linear optics. In Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, 333–342 (Association for Computing Machinery, New York, NY, USA, 2011).

Crespi, A. et al. Integrated multimode interferometers with arbitrary designs for photonic boson sampling. Nature Photonics 7, 545–549 (2013).

Brod, D. J. et al. Photonic implementation of boson sampling: a review. Advanced Photonics 1, 1–14 (2019).

Knill, E., Laflamme, R. & Milburn, G. J. A scheme for efficient quantum computation with linear optics. Nature 409, 46–52 (2001).

Menicucci, N. C. et al. Universal quantum computation with continuous-variable cluster states. Phys. Rev. Lett.
[97] 110501 (2006).
[69] Ferraro, A., Olivares, S. & Paris, M. G. Gaussian states in continuous variable quantum information. arXiv preprint quant-ph/0503237 (2005).
[70] Adesso, G., Ragy, S. & Lee, A. R. Continuous variable quantum information: Gaussian states and beyond. Open Systems & Information Dynamics 21, 1440001 (2014).
[71] Nokkala, J., Maniscalco, S. & Piilo, J. Non-markovianity over ensemble averages in quantum complex networks. Open Systems & Information Dynamics 24, 1740018 (2017).
[72] Boyd, S. & Chua, L. Fading memory and the problem of approximating nonlinear operators with volterra series. IEEE Transactions on circuits and systems 32, 1150–1161 (1985).
[73] Grigoryeva, L. & Ortega, J.-P. Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. The Journal of Machine Learning Research 19, 892–931 (2018).
[74] Dieudonné, J. Foundations of modern analysis (Read Books Ltd, 2011).
[75] Bertschinger, N. & Natschlager, T. Real-time computation at the edge of chaos in recurrent neural networks. Neural computation 16, 1413–1436 (2004).
[76] Jaeger, H. Short term memory in echo state networks. gmd-report 152. In GMD-German National Research Institute for Computer Science (2002) (Citeseer, 2002).
[77] Schrödinger, E. Der stetige Übergang von der mikrozur makromechanik. Naturwissenschaften 14, 664–666 (1926).
[78] Glauber, R. J. Coherent and incoherent states of the radiation field. Physical Review 131, 2766 (1963).
[79] Tanaka, G. et al. Recent advances in physical reservoir computing: A review. Neural Networks (2019).
[80] Soriano, M. C., Brunner, D., Escalona-Morán, M., Mirasso, C. R. & Fischer, I. Minimal approach to neuro-inspired information processing. Frontiers in computational neuroscience 9, 68 (2015).
[81] Araujo, F. A. et al. Role of non-linear data processing on speech recognition task in the framework of reservoir computing. Scientific Reports 10, 328 (2020).
[82] Bartlett, S. D., Sanders, B. C., Braunstein, S. L. & Nemoto, K. Efficient classical simulation of continuous variable quantum information processes. Physical Review Letters 88, 097904 (2002).
[83] Nielsen, M. A. & Chuang, I. L. Programmable quantum gate arrays. Physical Review Letters 79, 321 (1997).
[84] Elman, J. L. Finding structure in time. Cognitive science 14, 179–211 (1990).
[85] Pascuau, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks. In International conference on machine learning, 1310–1318 (2013).