Quantum Machine Intelligence (2025) 7:31
https://doi.org/10.1007/s42484-025-00261-9

RESEARCH ARTICLE

Optimal training of finitely sampled quantum reservoir computers for forecasting of chaotic dynamics

Osama Ahmed[1] · **Felix Tennie**[1,2] · **Luca Magri**[1,3,4]

Received: 2 September 2024 / Accepted: 12 February 2025

Abstract

In the current Noisy Intermediate Scale Quantum (NISQ) era, the presence of noise deteriorates the performance of quantum computing algorithms. Quantum reservoir computing (QRC) is a type of quantum machine learning algorithm, which, however, can benefit from different types of tuned noise. In this paper, we analyze how finite sampling noise affects the chaotic time series prediction of the gate-based QRC and recurrence-free quantum reservoir computing (RF-QRC) models. First, we examine RF-QRC and show that, even without a recurrent loop, it contains temporal information about previous reservoir states using leaky integrated neurons. This makes RF-QRC different from quantum extreme learning machines (QELM). Second, we show that finite sampling noise degrades the prediction capabilities of both QRC and RF-QRC while affecting QRC more due to the propagation of noise. Third, we optimize the training of the finite-sampled quantum reservoir computing framework using two methods: (a) singular value decomposition (SVD) applied to the data matrix containing noisy reservoir activation states and (b) data-filtering techniques to remove the high frequencies from the noisy reservoir activation states. We show that denoising reservoir activation states improves the signal-to-noise ratios with smaller training loss. Finally, we demonstrate that the training and denoising of the noisy reservoir activation signals in RF-QRC are highly parallelizable on multiple quantum processing units (QPUs) as compared to the QRC architecture with recurrent connections. The analyses are numerically showcased on prototypical chaotic dynamical systems with relevance to turbulence. This work opens opportunities for using quantum reservoir computing with finite samples for time series forecasting on near-term quantum hardware.

Keywords Quantum reservoir computing · Sampling noise · Recurrence-free quantum reservoir computing · Chaos · Turbulence

1 Introduction

Despite various noise sources affecting the performance of quantum algorithms in NISQ devices, finite sampling noise is a device-independent source of noise in various quantum machine learning (QML) algorithms. Finite sampling noise, which provides a fundamental limit to learning in different QML applications (Hu et al. 2023; Mujal et al. 2023; Tennie et al. 2025), is rooted in the foundations of quantum mechanics and will be present in future fault-tolerant quantum computers (FTQC) as well (Preskill 1998).

Obtaining information on the density operator ρ of a quantum system requires measurements of multiple copies of ρ, which is known as quantum state tomography (Nielsen and Chuang 2011). A complete quantum state tomography is exponentially hard and scales exponentially with the system size $\mathcal{O}(D^2)$, where $D = 2^n$ with n qubits. In Aaronson (2018), they showed that instead of generating a full-classical description of quantum states, it is often sufficient to directly predict many properties of the associated quantum system efficiently (shadow tomography), for example, in quantum chemistry and quantum simulations. The same analysis is

✉ Luca Magri
l.magri@imperial.ac.uk

Osama Ahmed
o.ahmed22@imperial.ac.uk

Felix Tennie
f.tennie@imperial.ac.uk

[1] Department of Aeronautics, Imperial College London, Exhibition Road, London SW7 2BX, UK

[2] Department of Engineering, City St George's, University of London, Northampton Square, London EC1V 0HB, UK

[3] The Alan Turing Institute, London NW1 2DB, UK

[4] Politecnico di Torino, DIMEAS, Corso Duca degli Abruzzi, 24, Torino 10129, Italy

Published online: 27 February 2025

extended by the approach of classical shadows (Huang et al. 2020), through which we can predict many associated properties of a quantum system from few measurements. For QML applications, one often requires access to a large number of measurement expectation values to perform classification and regression tasks (Schuld and Petruccione 2021). Therefore, in those cases, the concept of shadow tomography does not result in useful QML models.

The calculation of finite expectation values in variational quantum algorithms can also result in vanishing gradients and a mostly flat loss landscape called *barren plateaus* (McClean et al. 2018). To circumvent the issue of barren plateaus, quantum extreme learning machines (QELM) and quantum reservoir computing (QRC) (Fujii and Nakajima 2021; Pfeffer et al. 2022; Mujal et al. 2021) are promising frameworks because they do not require the calculation of gradients for loss minimization. QRC is inspired by classical reservoir computers (Jaeger 2001)—a class of recurrent neural networks (RNNs), which have proven to be excellent tools for time series forecasting (Racca and Magri 2022; Doan et al. 2019). QRC also benefits from different types of tuned noises such as amplitude and phase damping noise (Domingo et al. 2023; Fry et al. 2023a). QELM, on the other hand, does not involve recurrence and is easier to employ but has limited applications because they have no memory of past inputs and suffer from exponential concentration (Xiong et al. 2025). The most promising applications of QRC include forecasting chaotic dynamics (Ahmed et al. 2024; Fry et al. 2023a) and quantum dynamics (Sornsaeng et al. 2024) on a quantum computer.

The recurrence-free QRC (RF-QRC) (Ahmed et al. 2024) combines both QRC and QELM frameworks, in a way that does not have recurrence built in the quantum circuit, making it easier to exploit rich quantum reservoir dynamics for generating reservoir activation signals. Instead of recurrence inside the reservoir, the information about previous reservoir states is fed as a classical post-processing step with leaky integrated neurons with individual state dynamics and temporal memory (Jaeger et al. 2007). Nevertheless, it may be possible to employ both QRC and RF-QRC in classical settings as a *quantum-inspired* machine learning algorithm to improve the prediction capabilities of classical reservoir computers. For the prediction of chaotic dynamics and extreme events (Ahmed et al. 2024), this has been done by emulating the evolution of the quantum state vector, which allows us to numerically determine exact measurement expectation values. However, in order to realize any quantum advantage for the increasing number of qubits and for processing quantum data using QRC, we require the implementation of QRC (or RF-QRC) on quantum hardware with finite sampling.

Some previous proposals of QRC also consider the impact of finite sampling noise in physical implementations (Khan et al. 2021; Dudas et al. 2023). A recent framework uses weak and projective measurements in QRC to reduce noise effects (Mujal et al. 2023). Another proposal for QML applications uses variance regularization to suppress probabilistic noise in the framework of quantum neural networks (Kreplin and Roth 2024). Despite these proposals for suppressing finite sampling noise, the effect of spreading of correlations for temporal learning tasks in recurrent QML applications still needs to be explored.

In this work, we study and compare the impact of finite sampling noise on conventional QRC with recurrence and recurrence-free QRC (RF-QRC) architectures. We focus our analysis on the finite sampling noise for two reasons: (a) The physical computing time required for executing quantum circuits imposes limitations on the possible number of shots taken for a learning task, which results in lower bounds on the size of finite sampling noise. (b) Beyond that, in some cases, QRC can instead benefit from certain types of tuned noises (Fry et al. 2023a). Therefore, the motivation for this work is to analyze the impact of sampling noise in QRC as well as in RF-QRC and to present a few methods to mitigate its effects. Because of the lack of recurrence inside the reservoir (i.e., the parameterized quantum circuit), RF-QRC suppresses the propagation of correlations arising from noisy expectation values over time. RF-QRC also contains leaky integrated neurons, which introduce temporal memory and provide exponential smoothing of noisy states (lukosevicius 2012). This makes RF-QRC a promising candidate for succeeding with learning tasks on noisy NISQ devices. A feedback-driven reservoir computing approach (Kobayashi et al. 2024) also enhances the memory of QRC by providing active feedback inside the reservoir at each time step. On the other hand, RF-QRC does not require active feedback inside the reservoir, and the previous information is only introduced as a classical post-processing step.

This paper is structured as follows: In Sect. 2, we provide a brief overview of classical reservoir computing with leaky integrated neurons. In Sect. 3, we outline recurrence-free quantum reservoir computing (RF-QRC) as introduced in Ahmed et al. (2024) and compare it with leaky integrated reservoir computing without recurrence. We then extend this analysis to model uncorrelated noise in RF-QRC. Section 4 compares QRC and RF-QRC with finite sampling noise, and we present two denoising methods based on singular value decomposition (SVD) and signal smoothing techniques. These denoising methods are then applied to the three-dimensional Lorenz-63 and a nine-dimensional turbulent shear flow models (Appendix A, B). The proposal of denoising is then tested on noisy quantum hardware in Sect. 5. Finally, in Sect. 6, we conclude our findings and present prospects of future work.

2 Reservoir computing formalism

Reservoir computing is a type of recurrent neural network (RNN) that learns temporal correlations from the input data by mapping the low-dimensional input data to a high-dimensional reservoir. A particular type of classical reservoir computing, also known as echo state network (Jaeger 2001), uses randomly generated input and reservoir weight matrices $\boldsymbol{W}_{in} \in \mathbb{R}^{N_r \times N_u}$ and $\boldsymbol{W} \in \mathbb{R}^{N_r \times N_r}$ to generate reservoir activation states ($\boldsymbol{r}(t_{i+1}) \in \mathbb{R}^{N_r}$) at each time step from the input data $\boldsymbol{u}_{in}(t_i) \in \mathbb{R}^{N_u}$ (Fig. 1), as

$$\hat{\boldsymbol{r}}(t_{i+1}) = \tanh\left(\boldsymbol{W}_{in}\hat{\boldsymbol{u}}_{in}(t_{i+1}) + \boldsymbol{W}\boldsymbol{r}(t_i)\right), \tag{1}$$

$$\boldsymbol{r}(t_{i+1}) = (1-\epsilon)\,\boldsymbol{r}(t_i) + \epsilon\,\hat{\boldsymbol{r}}(t_{i+1}), \tag{2}$$

where ϵ is the user-defined leak-rate, which combines previous state information with the current time step as a linear combination—also known as leaky integrate echo state network (Jaeger et al. 2007). The tanh function is applied component-wise.

The optimal weight matrix $\boldsymbol{W}_{out}$ is obtained by linear ridge regression

$$(\boldsymbol{R}\boldsymbol{R}^T + \beta\boldsymbol{I})\,\boldsymbol{W}_{out} = \boldsymbol{R}\,\boldsymbol{U}_d^{\mathrm{T}}, \tag{3}$$

here, $\boldsymbol{R} \equiv [\boldsymbol{r}(t_1), \boldsymbol{r}(t_2), ...\boldsymbol{r}(t_{N_{tr}})] \in \mathbb{R}^{N_r \times N_{tr}}$ is a matrix of concatenated reservoir activation signals corresponding to each neuron for N_{tr} time steps of training (Fig. 2), β is the Tikhonov regularization factor, and $\boldsymbol{U}_d \equiv [\boldsymbol{u}_{in}(t_1), \boldsymbol{u}_{in}(t_2), ...\boldsymbol{u}_{in}(t_{N_{tr}})] \in \mathbb{R}^{N_r \times N_{tr}}$ is the matrix of concatenated input time series data used for training.

To predict future time series elements $\boldsymbol{u}_p(t_{i+1})$, the reservoir computer can either be run in open loop or autonomous (closed loop) configurations (Ahmed et al. 2024). The prediction, $\boldsymbol{u}_p$, is a linear combination of reservoir states

$$\boldsymbol{u}_p(t_{i+1}) = [\boldsymbol{r}(t_{i+1})]^{\mathrm{T}}\,\boldsymbol{W}_{out}. \tag{4}$$

2.1 Leaky integrated reservoir computing

The continuous-time dynamics of a dynamical system with leaky integration can be written as

$$\boldsymbol{x}(t) + \tau\frac{d\boldsymbol{x}}{dt} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{5}$$

where $\boldsymbol{x}(t)$ is the state of the dynamical system, τ is a time-constant of the system determining the rate of leakage or decay, and $\boldsymbol{F}$ is a non-linear function describing the evolution of state $\boldsymbol{x}(t)$ that is influenced by an input signal $\boldsymbol{u}(t)$. For echo state networks, the function $\boldsymbol{F}$ depends on random input and reservoir weight matrices (Jaeger 2001)

$$\tau\frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x}(t) + \tanh\left(\boldsymbol{W}_{in}\boldsymbol{u}_{in}(t) + \boldsymbol{W}\boldsymbol{x}(t)\right). \tag{6}$$

Using Euler discretization with stepsize Δt, we obtain a discrete time reservoir state update equation. In practical implementations, the input data $\boldsymbol{u}_{in}(t)$ is also sampled at discrete time steps

$$\boldsymbol{x}(t+\Delta t) = (1-\frac{\Delta t}{\tau})\,\boldsymbol{x}(t) + \frac{\Delta t}{\tau}\tanh\left(\boldsymbol{W}_{in}\boldsymbol{u}_{in}(t+\Delta t) + \boldsymbol{W}\boldsymbol{x}(t)\right). \tag{7}$$

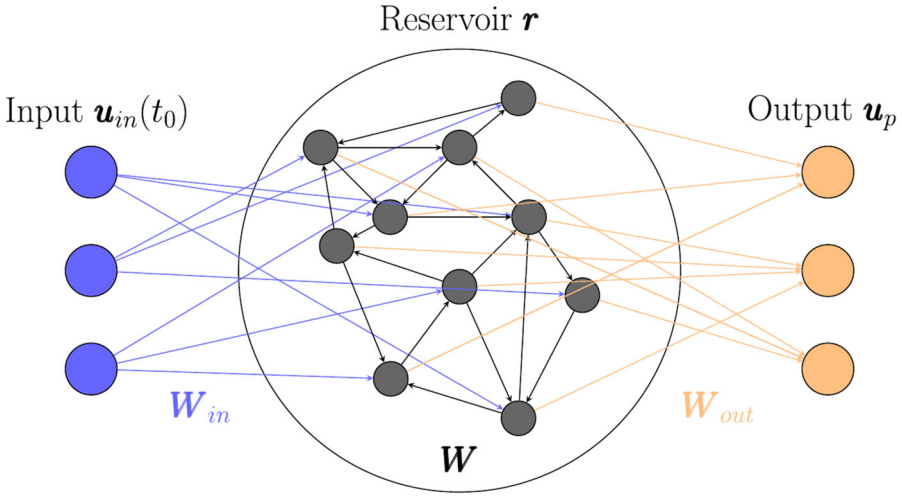

Fig. 1 Schematic representation of a reservoir computer (Jaeger 2001). The input data $\boldsymbol{u}_{in}$ is mapped to the reservoir matrix via $\boldsymbol{W}_{in}$. The reservoir neuron connections governed by $\boldsymbol{W}$ matrix allow the flow of information between neurons. The linear readout layer, with the trained output weight matrix $\boldsymbol{W}_{out}$, is used to make output predictions $\boldsymbol{u}_p$

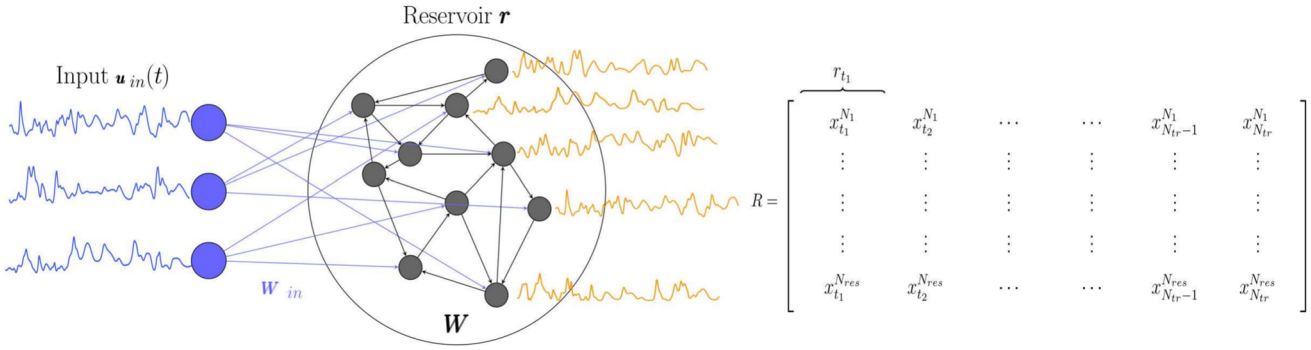

Fig. 2 Training phase in reservoir computing. Input time series $\boldsymbol{u}_{in}(t)$ is mapped to a reservoir using $\boldsymbol{W}_{in}$ matrix for CRC and suitable encoding schemes for QRC. Inside reservoir r, each neuron *echoes* with the input time series to generate a series of reservoir activation state signals. The reservoir activation state signals are concatenated in the reservoir state matrix $\boldsymbol{(R)}$, which is then used for finding optimal output weight matrix $\boldsymbol{W}_{out}$ in RC training using ridge regression

Simplifying Eq. 7 and keeping $\Delta t \equiv 1$ for discrete time steps gives the same reservoir state update equations shown previously in Eqs. 1–2. Here, $\epsilon \leq 1$ is a hyperparameter governing how much information from the previous reservoir states is retained

$$\boldsymbol{x}(t+1) = (1-\epsilon)\,\boldsymbol{x}(t) + \epsilon \tanh\left(\boldsymbol{W}_{in}\boldsymbol{u}_{in}(t+1) + \boldsymbol{W}\boldsymbol{x}(t)\right). \tag{8}$$

This type of model is also known as the leaky integrate and fire (LIF) neurons, which has several applications in neuroscience (Teeter et al. 2018; GABBIANI and COX 2010) and signal processing (Lansky and Ditlevsen 2008). The concept of leakage in reservoir computing was introduced in liquid state machines (Maass et al. 2002) and echo state networks (Jaeger et al. 2007; lukosevicius 2012). Physically, these leaky integrated neurons have individual state dynamics that make them suitable for temporal learning tasks. Leaky integration can also be considered as a digital low-pass filter or exponential smoothing (lukosevicius 2012), making the model appropriate for learning with noisy data.

3 Recurrence-free quantum reservoir computing

Quantum reservoir computing exploits the exponential size of the quantum state space to generate reservoirs for temporal information processing, with a potential for greater efficiency than classical reservoir computers (Mujal et al. 2021; Fujii and Nakajima 2021; Sornsaeng et al. 2024). However, any prospects of achieving quantum advantage require addressing noise issues for practical reservoir implementations. The combination of leaky integrate with quantum reservoir dynamics provides a pathway towards noise suppression. In this section, we briefly review gate-based standard quantum reservoir computing (Pfeffer et al. 2022) and recurrence-free quantum reservoir computing (Ahmed et al. 2024), which use the concept of leaky integrated neurons explained in Sect. 2.1. In Sect. 3.1, we give a brief overview of the parameters of the RF-QRC model employed in this work. Finally, in Sect. 3.2, we introduce a theoretical description of finite sampling noise in recurrence-free quantum reservoir computing.

In gate-based quantum reservoir computing, the quantum state vector is propagated by a $\boldsymbol{\theta}$-parametrized quantum circuit with unitary $\mathcal{U}(\boldsymbol{\theta})$ at each time step

$$|\psi(t_{i+1})\rangle = \mathcal{U}(\boldsymbol{\theta})|\psi(t_i)\rangle. \tag{9}$$

Previous proposals of gate-based QRC involve a reservoir map that depends on input data $\boldsymbol{u}_{in}$, previous reservoir states $\boldsymbol{r}$, and a random unitary parameterized by $\boldsymbol{\alpha}$. Specifically,

$$|\psi(t_{i+1})\rangle = \mathcal{U}(\tilde{\boldsymbol{\alpha}})\mathcal{U}(\boldsymbol{u}_{in}(t_{i+1}))\mathcal{U}(\boldsymbol{r}(t_i))|0\rangle^{\otimes n}, \tag{10}$$

Eq. 10 is analogous to the classical reservoir update in Eq. 1.

By contrast, the recurrence-free quantum reservoir state updating equation only evolves as a function of the input time series and a random $\boldsymbol{\alpha}$-parametrized unitary

$$|\psi(t_{i+1})\rangle = \mathcal{U}(\boldsymbol{\alpha})\mathcal{U}(\boldsymbol{u}_{in}(t_{i+1}))|0\rangle^{\otimes n}. \tag{11}$$

After each time stepping, a measurement in the computational basis $\{|k\rangle\}_{k=0}^{k=2^n}$ is performed and the result is used to form a new reservoir state vector $\boldsymbol{r}(t_{i+1})$

$$r^{(k)}(t_{i+1}) = (1-\epsilon)\,r^{(k)}(t_i) + \epsilon\,|\langle\psi(t_{i+1})|k\rangle|^2. \tag{12}$$

Equation 12 resembles the leaky integrated quantum reservoir state update with recurrence Eq. 8, but is different in that the state update in the parameterized quantum circuit only depends on input data $\boldsymbol{u}_{in}$ and not on the recurrent $\boldsymbol{r}(t_i)$ at each time step.

The quantum evolution in Eqs. 10 and 11 represents a unitary quantum feature map, in the absence of environmental or system's noise. For reservoir computing to work, the reservoir feature map should be dissipative to satisfy the "echo state" or fading memory property (Jaeger 2001). This property ensures that the reservoir is determined by the driving signal irrespective of its initial state. In this work, we have used the measurement of the quantum state in the computational basis and combined it with the information from the previous time step (with the hyperparameter ϵ), which provides dissipative reservoir states and satisfies the echo state property. More details about the network are in Sect. 3.1.

The RF-QRC may resemble a QELM because it does not have an active recurrence inside the reservoir (parameterized quantum circuit) (Mujal et al. 2021; Xiong et al. 2025). However, RF-QRC is not a QELM because it also contains information about previous reservoir states using leaky integrated neurons with hyperparameter ϵ. It has been observed that ϵ can be tuned to alter the short-term memory capacity (lukosevicius 2012). This also makes RF-QRC suitable for temporal learning tasks as discussed in Ahmed et al. (2024) for various chaotic systems. Additionally, the performance of RF-QRC depends on the chosen feature map which is analogous to the necessity of tuning classical reservoirs in classical reservoir computing. Owing to the lack of active recurrence, RF-QRC is a specific kind of quantum reservoir computer that is scalable and potentially offers a natural path to mitigate the issue of propagating noise.

3.1 Model parameters

In this section, we present the parameters for the RF-QRC model implemented in this work. The quantum reservoir state update equation shown in Eq. 11 involves unitary quantum feature maps. Each unitary evolution is parameterized by the classical input with single qubit rotation R_y gate and two-qubit entanglement CNOT gates. Common choices of the unitary map were explored in Pfeffer et al. (2023); Ahmed et al. (2024); Suzuki et al. (2022). In this work, we utilize a fully connected feature map for unitary evolution. For the second unitary encoding $V(\boldsymbol{\alpha})$, the random rotation angles $(\boldsymbol{\alpha})$ are sampled $V(\boldsymbol{\alpha}) \epsilon \mathbb{R}^n$ (n = number of qubits) from a uniform distribution interval $[0, 4\pi]$ with a predefined seed, which we keep it fixed throughout the training and prediction of a particular realization. This is similar to the classical input $\boldsymbol{W}_{in}$ and reservoir weight matrices $\boldsymbol{W}$, which are also pseudo-randomly generated and fixed for any specific realization Eqs. 1 and 2. Alternatively, $(\boldsymbol{\alpha})$ could also be treated as a hyperparameter for performance enhancement.

We applied the RF-QRC model to study a three-dimensional reduced order model of thermal convection flow also known as Lorenz-63 model (Lorenz 1995) and a nine-dimensional qualitative low-order model of turbulent shear flows, which is based on Fourier modes known as "Moehlis, Faisst, and Eckhardt (MFE)" (Moehlis et al. 2004) model. Further details about the physics of these models are presented in Appendices A and B, respectively. The time series data for both of these models are derived by a Runge–Kutta method. The time step (dt) for the numerical schemes and hyperparameters for RF-QRC are shown in Table 1. Each time series is divided into a washout, training, and testing data set. The washout phase also known as preparation time is the warmup phase in which the transients are discarded to wash out the effect of initial conditions.

The dynamics of chaotic systems can be characterized by the leading Lyapunov exponent λ_1 (Boffetta et al. 2002). This exponent underpins the average exponential rate of divergence for initially nearby trajectories in chaotic systems. The Lyapunov exponent also provides a time scale to assess the time-accurate prediction of the chaotic systems. We have rescaled our time units to the inverse of the Lyapunov exponent (λ_1), which is called the Lyapunov time (LT). Based on the discretization time step, 1 LT corresponds to 110 steps for Lorenz-63 and 245 steps for the MFE system as shown in Table 1. After specifying quantum feature maps, only tunable hyperparameters are Tikhonov regularization β and leak-rate ϵ in RF-QRC. We perform a grid search to find optimal parameters between the ranges specified in Table 1.

Table 1 Parameters for the tests on the Lorenz-63 and MFE systems with the recurrence-free quantum reservoir computing architecture

Parameters	Symbol	Lorenz-63 system	MFE system
Time step	dt	0.01s	0.25s
Leading Lyapunov exponent	λ_1	0.9056	0.0163
Lyapunov Time	LT	1 LT = 110 steps	1 LT = 245 steps
Washout steps	N_W	5 LT	2 LT
Training steps	N_{tr}	20 LT	65 LT
Tikhonov regularization	β	1×10^{-9},1×10^{-12}	1×10^{-9},1×10^{-12}
Leak rate	ϵ	[0.05, 0.3]	[0.05, 0.3]
Resevoir density	D	Fully connected	Fully connected

3.2 Quantum reservoir computing with finite samples

Before exploring the effect of finite sampling noise by applying the RF-QRC model to prototypical model systems, we first consider the effects of finite sampling noise analytically. Due to the absence of active recurrence, the finite sampling noise can generally be assumed as noise uncorrelated in time. When neglecting hardware noise, the actual reservoir state $\boldsymbol{r}_N(t)$ is related to the outcomes of S measurements $\overline{\boldsymbol{r}}_N(t)$ by[1]

$$\boldsymbol{r}_N(t) = \lim_{S\to\infty} \overline{\boldsymbol{r}}_N(t). \tag{13}$$

The effect of finite sampling noise on reservoir activation states ($\boldsymbol{r}_N$) can be modeled as a time-dependent stochastic variable $\boldsymbol{\zeta}(t)$ with an explicit constant prefactor of $1/\sqrt{S}$ accounting for the central limit theorem

$$\overline{\boldsymbol{r}}_N(t) = \boldsymbol{r}_N(t) + \frac{1}{\sqrt{S}}\boldsymbol{\zeta}(t). \tag{14}$$

Like in classical reservoir computing, in QRC, computational basis states correspond to neurons $N_1, N_2, ..., N_{res}$ where $N_{res} = 2^n$ is the dimension of the reservoir. The time varying shot noise signals may therefore be regarded as time-dependent noise functions on those neurons as illustrated in Fig. 2. The stochastic nature of individual reservoir vectors translates to a reservoir state matrix R with stochastic variability $\boldsymbol{Z}$

$$\overline{\boldsymbol{R}} = \boldsymbol{R} + \frac{1}{\sqrt{S}}\boldsymbol{Z}. \tag{15}$$

Quantum sampling noise of a single qubit follows a binomial distribution (Schuld and Petruccione 2021). For multiple qubits forming a quantum state vector with finite samples, the sampling noise becomes a multinomial distribution. The lack of recurrence in RF-QRC allows us to model this noise as uncorrelated in time. This form of noise has also been considered in Hu et al. (2023) in the context of the analysis of resolvable expressive capacity (REC).

Consequently, in Eq. 15, we can model $\boldsymbol{Z}$ as a centered multinomial stochastic process. Without loss of generality, $\boldsymbol{Z}$ can always be shifted to have zero mean ($\mathbb{E}[\boldsymbol{Z}] = 0$). Thus, the stochastic matrix can be modeled by only considering second-order moments that form the sampled covariance matrix

$$\boldsymbol{\Sigma}_{ij}(t) = Cov[\zeta_i(t), \zeta_j(t)]. \tag{16}$$

[1] We will drop the subscript from t_{i+1} for brevity.

By taking the expectation value over an infinite number of measurements, one gets the covariance matrix $\boldsymbol{V} = \mathbb{E}[\boldsymbol{\Sigma}_{ij}]$. It can be written in terms of the sampled reservoir matrix $\boldsymbol{R}$

$$\boldsymbol{V} = \text{diag}\left(\frac{1}{N_{tr}}\sum_{N_{tr}}(\overline{\boldsymbol{R}})\right) - \overline{\boldsymbol{R}}\,\overline{\boldsymbol{R}}^T. \tag{17}$$

The second term on the right-hand side of Eq. 17, also known as Gram Matrix, arises naturally in the ridge regression loss in reservoir computing Eq. 3. This type of loss function is quadratic, and cumulants up to second-order (mean and covariances) are generally assumed to be sufficient to study the effect of noise in the training of these models (Hu et al. 2023; Khan et al. 2023).

4 Noisy reservoir activation states

In this section, we analyze the impact of finite sampling noise on the prediction capabilities of QRC and RF-QRC and propose methods to reduce it. After training classical and quantum reservoirs on the MFE model, we perform autonomous closed loop predictions for both networks to compare the performances. In Fig. 3, results of the time series prediction are shown for an ideal probability distribution (assuming an infinite number of measurements, sometimes referred to as "shots") and for various different learning outcomes based on a variable number of shots, S. A minimum number of finite samples is required to improve the forecasting abilities of QRC beyond classical reservoir computers. To address the performance deterioration with a finite number of shots in QRC, we first outline the working of the reservoir computers.

The working principle of reservoir computing is to combine the generated reservoir activation signals for functional approximation of the dynamical systems (Fig. 2). This is done by minimizing the loss function over the input training data set. Reservoir signals have a large overlap with a relatively lower-dimensional manifold (Carroll 2020) of active states, which is the active space. More specifically, the singular value decomposition of common reservoir matrices reveals only a limited number of relevant singular values, and any denoising procedure should preserve the corresponding eigenspaces.

For a number of chaotic systems, the time series signals have corresponding reservoirs with low-dimensional active spaces (Carroll 2020). Often, in the classical reservoir computing framework, reservoir matrices with high-dimensional active spaces are associated with lower testing errors (Carroll and Pecora 2019). However, in QRC, we find that, although the addition of finite sampling noise in QRC reservoirs increases the dimensionality of the active space, it does not

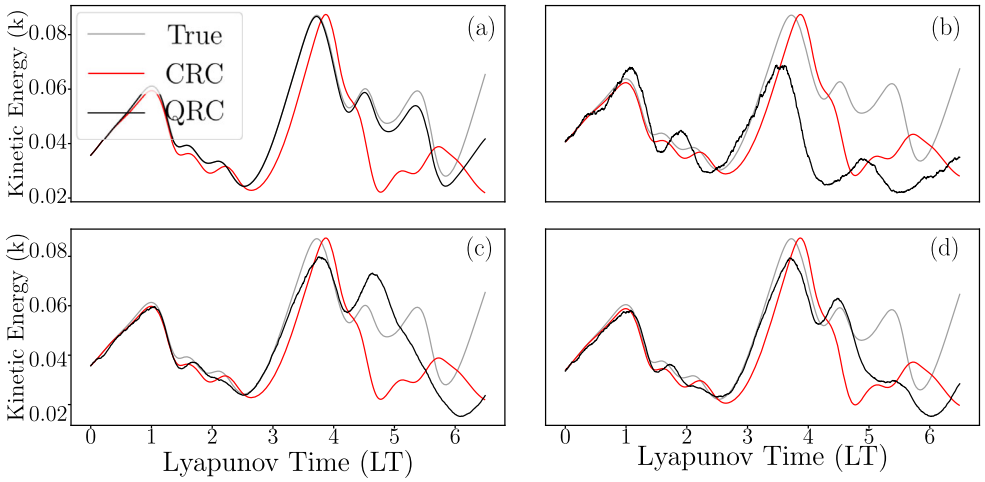

Fig. 3 Prediction of the MFE time series on unseen data set with **a** noise-free probability distribution, **b** 0.5×10^5 shots, **c** 2×10^5 shots, **d** 4×10^5 shots. Classical reservoir computing (CRC) results from the best-performing reservoir of size 1024 (hyperparameters in Appendix C) compared with best-performing quantum reservoir computing (QRC) of 10 qubits. QRC hyperparameters are detailed in Sect. 3.1

necessarily improves the learning performance. REC analysis (Hu et al. 2023) shows that, in the presence of noise, taking more activation states than a given threshold results in poor functional approximation. Therefore, in order to perform an analysis of the impact of noise on the learning performance, we consider the signal-to-noise (SNR) ratios of individual reservoir activation signals. Furthermore, we compute the accuracy of function fitting of these noisy signals, which is determined by computing the training loss.

In Fig. 4, we compare the SNR ratio for QRC and RF-QRC for a 9-dimensional chaotic shear flow model of the MFE system (Moehlis et al. 2004). Our results indicate that the presence of correlated noise in QRC results in more noisy estimates of the reservoir activation signals than RF-QRC because of the propagation of correlations in QRC. An example of the noisy and denoised reservoir activation state signals for RF-QRC, measured on a 10 qubits system, can be seen in Fig. 5. Our results indicate that for a constant number of samples, denoising reservoir activation signals results in a better fit of the training signal with lower noise variance. We now discuss the details of the two approaches utilized for suppressing noise in RF-QRC.

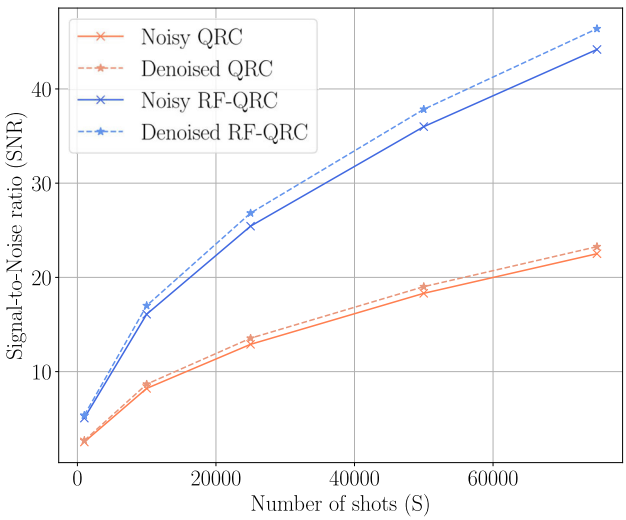

Fig. 4 Averaged signal-to-noise ratios (SNR) of the reservoir states of QRC and RF-QRC with and without denoising filters. The underlying model is an MFE times series with a reservoir size of 10 qubits ($N_{res} = 1024$)

4.1 Noise suppression using SVD

In signal processing, principal component analysis (PCA) or singular value decomposition (SVD) is a method to improve the SNR of noisy signals (Shlens 2014; Schanze 2017; Jha and Yadava 2010). The unbiased estimation of the expectation values in quantum computation is fundamentally limited by the Cramer-Rao bound (Yu et al. 2022). For an ensemble of quantum systems, governed by an input time series, the resulting expectation values form a reservoir signal with an added finite sampling noise. We found that the SNR of these noisy reservoir signals can be improved by using classical signal processing tools such as SVD. For a noisy reservoir state matrix $\overline{\boldsymbol{R}} \in \mathbb{R}^{N_r \times N_{tr}}$, the singular value decomposition is

$$\overline{\boldsymbol{R}} = \boldsymbol{U}\,\boldsymbol{S}\,\boldsymbol{V}^T, \tag{18}$$

where $\mathbf{U}$ is an orthogonal $N_r \times N_{tr}$ matrix, $\mathbf{S}$ is a diagonal $N_{tr} \times N_{tr}$ matrix with non-negative singular values, and $\mathbf{V}$ is an orthogonal $N_{tr} \times N_{tr}$ matrix. In order to maximize

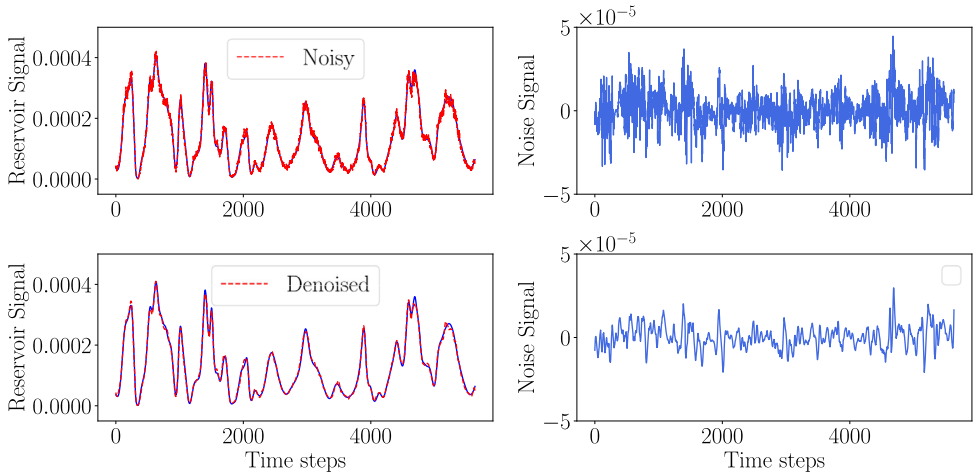

Fig. 5 MFE system trained on a 10-qubit system results in $N_{res} = 1024$ reservoir activation signals each of length $N_{tr} = 5800$ time steps. A single randomly chosen reservoir activation signal is shown with 50,000 shots at each timestep for RF-QRC are used for training. Top, visualization of the noisy reservoir activation signal (left) plotted on ideal (noise-free) signal, and the noise component—the difference between the two signals (right). Bottom, visualization of the denoised reservoir activation signal (left) plotted on ideal (noise-free) signal, and the noise component—the difference between the two signals (right)

the SNR, we derive a low-rank approximation of our noisy reservoir matrix by truncating singular values below a given threshold that are assumed to be associated with the noise component.

We apply SVD to denoise reservoir activation signals in the Lorenz-63 and MFE systems, considering two cases for each system: (a) The low-rank approximation of the noisy reservoir matrix (after SVD) is compared to the original noisy matrix, which contains more reservoir activation signals but includes noise. This approach evaluates the impact of truncating approximately half of the singular values to reduce noise. (b) The low-order reduced rank approximation after SVD is compared with the reduced noisy matrix of the same size, effectively having the same number of reservoir activation signals. This is relevant when the reduced dimensionality of the reservoir state matrix is required to reduce computational resources before post-processing it for training.

In Fig. 6, we show the results of the training error for a 7-qubits ($N_{res} = 128$) reservoir size trained on a Lorenz-63 time series with finite quantum reservoir sampling. The results indicate that truncating half of the singular values, as shown in Fig. 6a, increases the mean-squared error compared to the original noisy reservoir matrix. This suggests that while truncation removes noise, it may also discard important activation signals necessary for functional approximation. In contrast, Fig. 6b demonstrates that the low-rank approximation improves training error compared to a reduced noisy matrix of the same size, making it particularly effective for

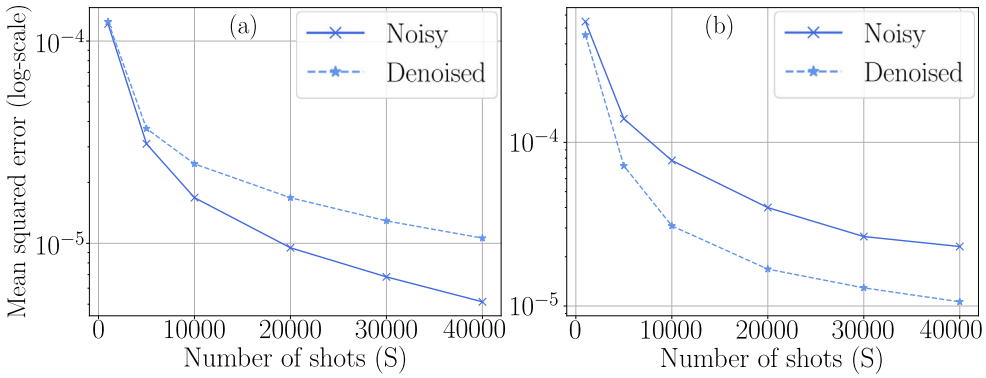

Fig. 6 Time series prediction of the Lorenz 63 model using RF-QRC. Averaged mean-squared errors of the trajectory are shown as a function of the number of measurements S for noisy and SVD denoised protocols. **a** Comparison of noisy and denoised figures with an underlying original reservoir size of $N_{res} = 128$. **b** Comparison of noisy and denoised figures with an underlying reduced reservoir size of $N_{res} = 60$

dimensionality reduction of the feature matrix before training. We have found similar results for the MFE model in Fig. 7.

Despite the noise suppression in the reduced dimensionality case, the SVD approach for quantum reservoirs is not computationally feasible. The number of reservoir activation signals (N_{res}) in quantum reservoir scales exponentially with the number of qubits. The computational complexity of SVD applied to the noisy reservoir activation signal is $\mathcal{O}(N_{tr}N_{res} \times min(N_{tr}N_{res}))$, which makes SVD on classical computers unfeasible for high-dimensional reservoirs. In Sect. 4.2, we present the denoising method based on signal filtering, which is scalable for larger reservoir sizes.

4.2 Noise suppression using signal filtering

Using SVD for denoising requires the knowledge of the complete reservoir matrix $\boldsymbol{R}$, the dimension of which scales exponentially with the number of qubits, which limits the application of SVD analysis for RF-QRC. In this section, we propose a second method of suppressing noise from the reservoir activation states by applying a denoising low-pass filter to each reservoir activation state.

We emphasize that in the case of RF-QRC and the absence of recurrence, reservoir activation states are only driven by the input time series, which are known *a priori*, as opposed to conventional quantum reservoirs with recurrence, which require the reservoir activation states from previous time step. Thus, we can employ multiple quantum systems in parallel to generate the reservoir activation states at each time step (Fig. 11). Later on, we can concatenate the expectation of the estimates for each eigenbasis to form a reservoir activation signal. This signal is the noisy estimate of our reservoir signal in which we remove the high frequencies by applying a *moving average polynomial regression* method (Schafer 2011). In principle, one could also apply a physical filter to the noisy signal estimates (Khan et al. 2023); in this work, we employ digital filtering by post-processing quantum measurements on a classical computer. Specifically, we have used a *Savitsky-Golay* (Schafer 2011) type of moving-average filter and applied it to each reservoir activation state. This type of signal filtering needs the window size and the order of the polynomial to be prescribed. On the one hand, in an open loop (training), the maximum length of the window is equal to the length of the training timeseries, and there is no constrain on the minimum length. On the other hand, in a closed loop (prediction), we need to evolve the noisy reservoir as long as the moving-average window length before denoising. Therefore, the moving average window is a hyperparameter that requires tuning. By performing a grid search for the polynomial degree in the range {2,6} and the window length in the range {10,60} with a stepsize of 10, we select a 3^{rd} polynomial for a window length of 20 and a 4^{th} order polynomial with a window length of 30 for the Lorenz-63 and MFE systems, respectively.

In Fig. 8, for a time series of the Lorenz-63 system, we present the SNR and mean-squared training error for noisy and denoised reservoir activation states using polynomial regression. By contrast to denoising based on SVD, using this method for suppressing noise always results in a lower mean-squared error for both cases of reduced and complete reservoir activation state matrix representations. The results in Fig. 8b show that denoising leads to a lower mean-squared error for various numbers of shots. The lower mean-squared error therefore gives an improved SNR ratio for the reservoir activation signals. In Fig. 9, similar results are shown for the

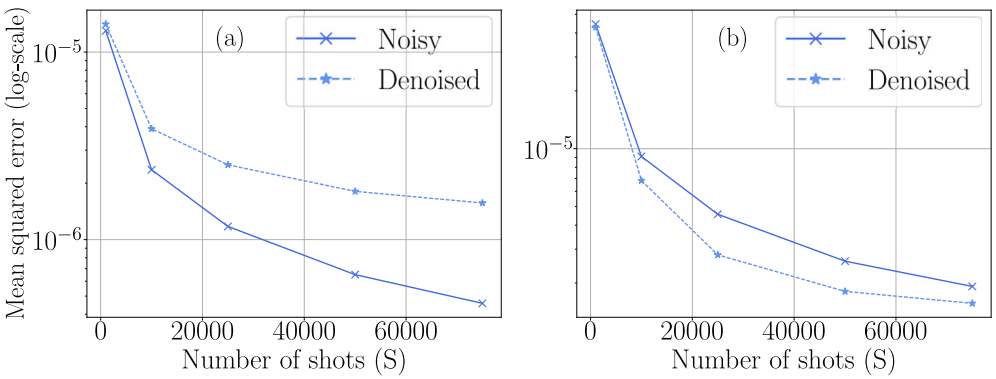

Fig. 7 Time series prediction of the MFE model using RF-QRC. Averaged mean-squared errors of the trajectory are shown as a function of the number of measurements S for noisy and SVD denoised protocols. **a** Comparison of noisy and denoised figures with an underlying original reservoir size of $N_{res} = 1024$. **b** Comparison of noisy and denoised figures with an underlying reduced reservoir size of $N_{res} = 500$

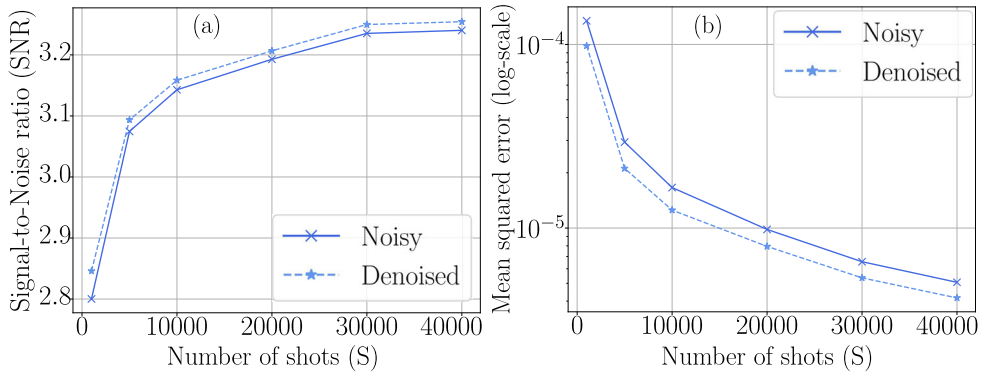

Fig. 8 Comparison of **a** signal-to-noise ratios and **b** mean-squared errors of noisy and denoised RF-QRC trained on a time series of the Lorenz 63 system. Reservoir size $N_{res} = 128$ represented by 7 qubits

MFE model, indicating that denoising reduces the training error significantly. These analyses can be extended to explore and employ more advanced classical filtering techniques. For the computational time, this method also requires the application of a denoising filter to each reservoir activation signal (N_{res}), which scales exponentially with the number of qubits. However, by comparison with SVD, denoising methods that involve individual signal smoothing are parallelizable classically for each reservoir signal, offering an efficient choice for noise suppression.

5 Hardware implementation

To demonstrate the feasibility of training RF-QRC on quantum hardware and test our proposal of denoising, we train the turbulent chaotic shear flow model on the IBM Quantum

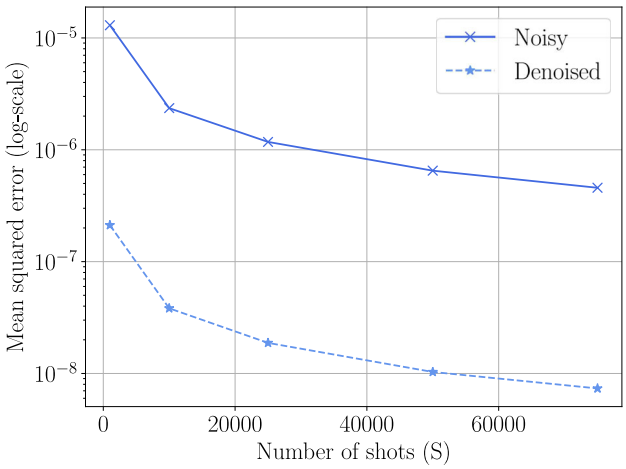

Fig. 9 Turbulent chaotic shear flow time series analysis with RF-QRC, using 10 qubit systems. Mean-squared error for noisy and denoised activation states with signal filtering according to Sect. 4.2

127-Qubit ibmq_kyoto device (IBM Quantum 2024). The qubit connectivity map of the hardware backend is shown in Fig. 10. From the 127-qubit system, we train our MFE model on a 10-qubit space that corresponds to a reservoir size of $N_{res} = 1024$.

Before proceeding with the training of RF-QRC, we simplified the ansatz from fully connected layers to linear entangling layers (Ahmed et al. 2024). This simplification is necessary because the device used has limited connectivity. Our simplified RF-QRC contains a layer of Hadamard gates applied on each qubit followed by R_y rotation gates, parameterized by the input time series signal (9 parameters for MFE). The circuit is then followed by a linear entangling layer of CNOT gates entangling all 10 qubits. This first feature map is applied twice to enrich reservoir dynamics. The second feature map differs from the first in that the rotation gates R_y are now parameterized by random rotation angles $V(\boldsymbol{\alpha})$ uniformly sampled from the interval $[0, 4\pi]$, to introduce randomization in the reservoir. These random rotation gates are sampled once and are kept constant throughout the training.

The quantum circuit is then transpiled by the $Qiskit$ runtime transpiler to achieve an optimal map of the quantum circuit to the physical qubits of the ibmq_kyoto backend. Finally, dynamic decoupling (Qiskit contributors 2023) to the idle qubits is applied to mitigate decoherence. No additional error mitigation strategies were applied.

5.1 Parallel training and denoising on multiple QPUs

The MFE input time series signal parameters are mapped on the pre-processed parameterized quantum circuit. The length of the time series signal was chosen as 1200 time steps. The training procedure for RF-QRC is illustrated in Fig. 11. Each quantum circuit is executed in parallel, independent of the other quantum circuits because of the lack of recurrence

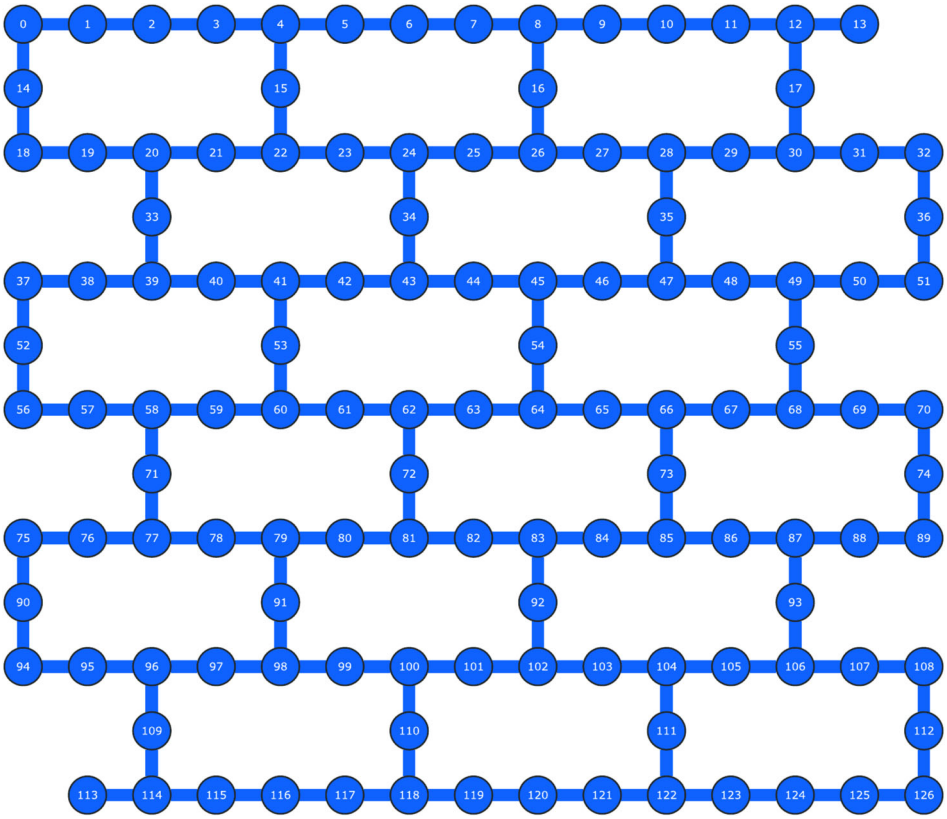

Fig. 10 Connectivity and qubit map of the ibmq_kyoto 127 qubit device. Ten of the qubits with the lowest readout assignment error are used as a reservoir for RF-QRC of the MFE model

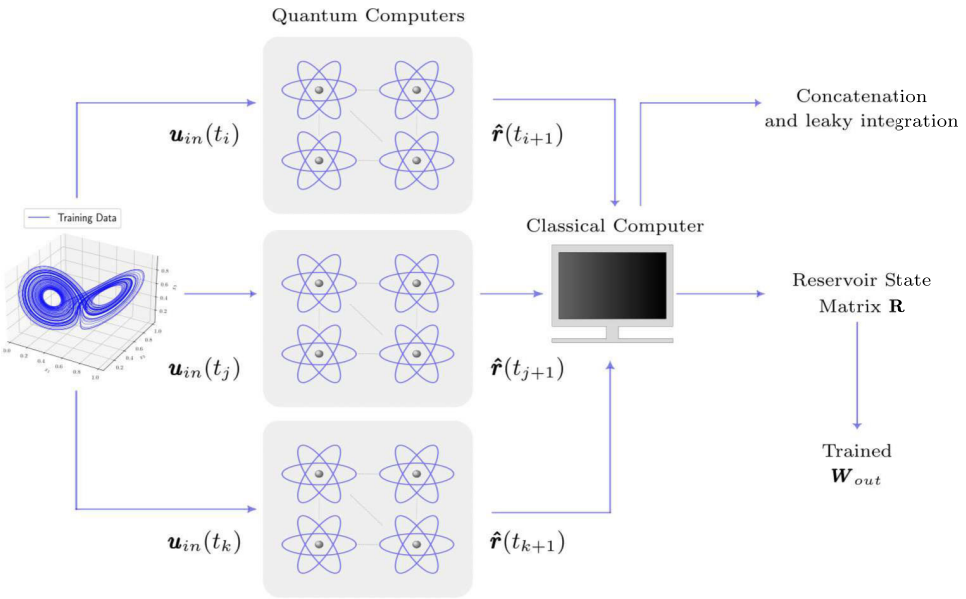

Fig. 11 Recurrence-free quantum reservoir computing parallel training. Input data $\boldsymbol{u}_{in}(t)$ is divided into smaller-length signals and passed through multiple parallel quantum computers. The obtained reservoir activation signals $\hat{\boldsymbol{r}}_{in}(t)$ associated with each input time series signal are concatenated classically where leaky integration and denoising are applied. The combined reservoir state matrix $\boldsymbol{R}$ is then used for training using ridge regression

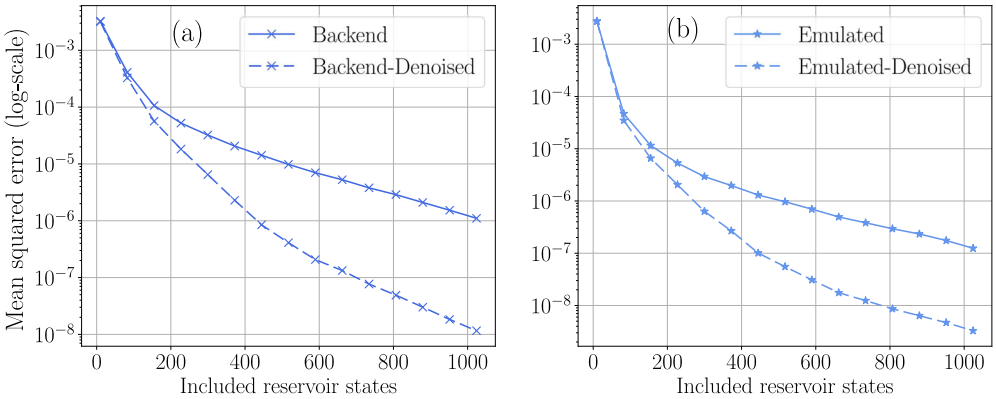

Fig. 12 Turbulent chaotic shear flow (MFE) training mean-squared error comparison with noisy and denoised states for both emulated and quantum processor-based RF-QRC

inside the circuit and the measurement in the computational basis is performed. For each data point, 10^4 shots are sampled, and the results are concatenated classically, followed by leaky integration and denoising techniques as outlined in Sect. 4.

Results in Fig. 12 indicate that denoising helps to improve the training accuracy for both the emulation of the state vector propagation (only subject shot noise) and the training on the ibmq_kyoto quantum processor (subject to shot noise *and* hardware noise) for increasing reservoir states. Therefore, even in the presence of correlated hardware noises, denoising improves the performance of the model. Other results in Fig. 13 demonstrate that hardware noise also has a significant impact on the performance, in addition to the impact of shot noise. In particular, the model trained on the quantum processor requires more reservoir states (i.e., a larger active space of the reservoir) for the same training accuracy when compared to a model with only shot noise. Similar effects are observed in both denoised models. However, denoising can be instrumental in mitigating those errors to reduce this gap between hardware and emulated reservoirs. This suggests that denoising suppresses shot noise and potentially other different types of noises as well. The averaged training loss for each configuration with all included reservoir states ($N_{res} = 1024$) is listed in Table 2.

Finally, we present the results of the reconstructed time series of kinetic energy from a trained 9-mode MFE in Fig. 14. We do so for both emulated models with shot noise and the trained RF-QRC on the backend including hardware noises. The reconstructed time series demonstrates that without denoising, the resulting fit for the trained signal suffers more from finite sampling and hardware noise. However, the training of RF-QRC coupled with denoising helps in reducing the shot noise as well as various hardware noises to have a better fit for optimal training.

6 Conclusion

Quantum reservoir computing is a promising tool for time series forecasting of chaotic signals when emulated classically with the assumption of ideal (noise-free) expectation values. To realize any quantum advantage and for real-world applications in weather and climate forecasting, a

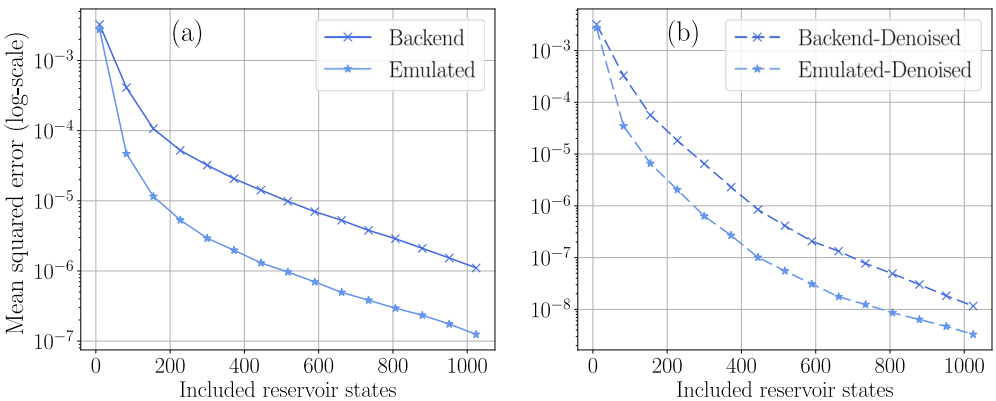

Fig. 13 Turbulent chaotic shear flow training mean-squared error comparison with emulated and quantum processor results

Table 2 Training loss comparison on emulator and backend trained networks, with and without denoising

	Noisy	Denoised
Emulator	1.24×10^{-7}	3.28×10^{-9}
Backend	1.10×10^{-6}	1.16×10^{-8}

high-dimensional reservoir and a sufficient number of qubits on quantum hardware are required. The performance of quantum hardware is, however, limited by the presence of environmental and sampling noise. In this work, we study the effect of sampling noise on chaotic and turbulent systems, which exhibits extreme events. The findings of this paper are four-fold. First, we provide a mathematical overview of RF-QRC that involves a temporal memory with leaky integrator, which also provides exponential smoothing of noisy states. This also makes the RF-QRC model scalable and suitable for temporal learning tasks. Second, we compare the effects of finite sampling noise on quantum reservoir architectures with and without recurrence. We show that the framework of RF-QRC is more resilient to sampling noise than QRC with correlated noise. Third, we propose two methods based on SVD and signal filtering to suppress noise in reservoir activation signals. Our emulated results indicate that suppressing noise improves the training accuracy as highlighted by smaller mean-squared training errors and higher signal-to-noise (SNR) ratios. The methods of denoising applied in this work are general, and the same analysis could be extended further by employing different advanced techniques for noise filtering to further improve the performance. Finally, we demonstrate our proposal by employing RF-QRC on multiple parallel QPUs on hardware backends, coupled with denoising techniques. The results indicate that denoising helps suppress finite sampling noise as well as other types of hardware noises. The proposed denoising can be extended to standard quantum reservoir computing architectures with recurrence and quantum extreme learning machines to suppress sampling and other types of noises for optimal training. This work opens up opportunities to employ quantum reservoir computing on quantum hardware for chaotic time series forecasting.

Appendix A. Three-dimensional Lorenz-63 model

One of the analyzed system, Lorenz-63 (Lorenz 1995) is a reduced order model of thermal convection flow. In this model, the fluid is heated uniformly from bottom and cooled from the top. Mathematically,

$$\frac{dx_1}{dt} = \sigma\,(x_2 - x_1), \tag{A1}$$

$$\frac{dx_2}{dt} = x_1\,(\rho - x_3) - x_2, \tag{A2}$$

$$\frac{dx_3}{dt} = x_1 x_2 - \beta x_3, \tag{A3}$$

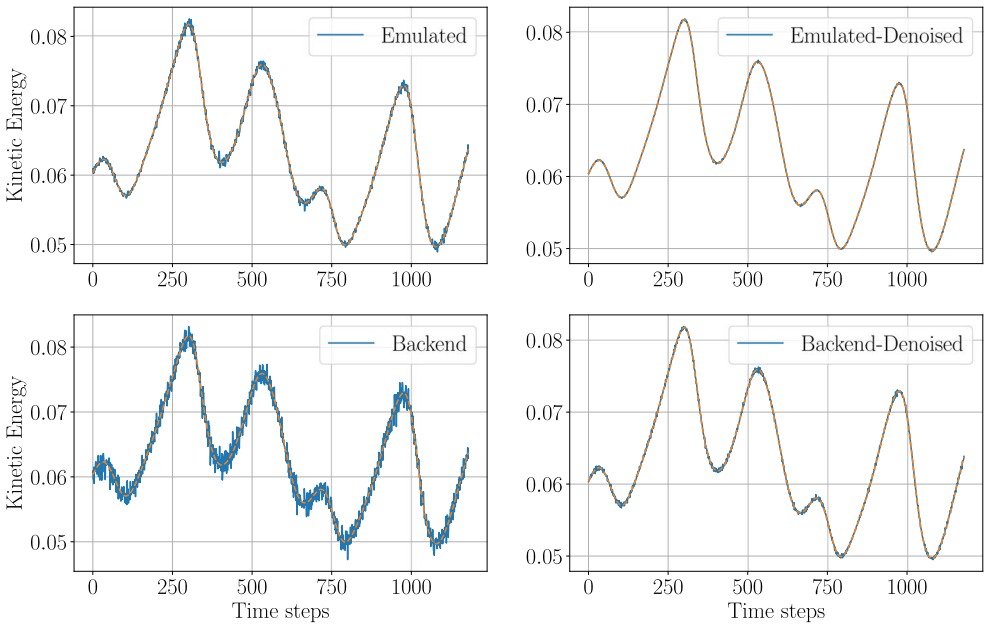

Fig. 14 Turbulent chaotic shear flow reconstructed kinetic energy comparison with emulated and backend results. The top panel presents the results of emulated noisy and denoised reservoir activation states. The bottom panel presents the results of hardware-trained noisy and denoised reservoir activation states

where σ, ρ, and β are system parameters, and we take [σ, ρ, β] = [10, 28, 8/3] to ensure chaotic behavior of the system. The largest Lyapunov exponent describes the non-linear dynamics (Racca and Magri 2022), which is $\Lambda = 0.9$ for Lorenz-63 system and $1LT = 1/0.9$ (Lorenz 1995). The time series data set is derived numerically via Runge–Kutta method and by taking a time step size $dt = 0.01$. The training time series comprises data points over a total time of 20 LT. For the training, the reservoir is evolved in an open loop to obtain reservoir states and calculate the $\boldsymbol{W}_{out}$ matrix and associated training mean-squared error.

Appendix B. Nine-dimensional turbulent chaotic shear flow model

To study turbulence, we consider a qualitative low-order model of turbulent shear flows, which is based on Fourier modes. Also known as the MFE model ('Moehlis, Faisst, and Eckhardt (MFE)' model), it is a non-linear model that captures the relaminarization and turbulent bursts (Moehlis et al. 2004). Due to the non-linear nature of this model, the MFE model has been employed to study turbulence transitions and chaos predictability (Srinivasan et al. 2019). Mathematically, the MFE model can be described by the non-dimensional Navier–Stokes equations for forced incompressible flow

$$\frac{\partial \boldsymbol{v}}{\partial t} = -(\boldsymbol{v}\,.\,\boldsymbol{\nabla})\,\boldsymbol{v} - \boldsymbol{\nabla}\,p + \frac{1}{Re}\Delta \boldsymbol{v} + \boldsymbol{F}(y), \qquad \boldsymbol{\nabla}.\boldsymbol{v} = 0, \quad \text{(B4)}$$

where $\boldsymbol{v} = (u, v, w)$ is the three-dimensional velocity vector, p is the pressure, Re is the Reynolds number, ∇ is the gradient, and Δ is the Laplacian operator. $\boldsymbol{F}(y)$ on the right-hand side is the sinusoidal body forcing term, which is $\boldsymbol{F}(y) = \sqrt{2}\pi^2/(4Re)\sin(\pi y/2)\,\boldsymbol{e_x}$. The body forcing term is applied between the plates along the x, y direction of the shear. Furthermore, we consider a three-dimensional domain of size $L_x \times L_y \times L_z = [4\pi, 2, 2\pi]$ and apply free slip boundary conditions at $y = L_y/2$ and periodic boundary conditions at $x = [0; L_x]$ and $z = [0; L_z]$. The set of PDEs can be converted into ODEs by projecting the velocities onto Fourier modes as given by Eq. B5

$$\boldsymbol{v}(\boldsymbol{x}, t) = \sum_{i=1}^{9} a_i(t)\,\hat{\boldsymbol{v}}_i\,(\boldsymbol{x}). \quad \text{(B5)}$$

These nine decompositions for the amplitudes $a_i(t)$ are substituted into Eq. B4 to yield a set of nine ordinary differential equations as in Moehlis et al. (2004). The MFE system displays a chaotic transient, which in the long term converges to a stable laminar solution. We want to predict the turbulent burst of kinetic energy and chaotic transients, which are extreme events. We solve the MFE system ODEs using an RK4 solver with $dt = 0.25$. The leading Lyapunov exponent of the MFE model (Racca and Magri 2022) is $\Lambda = 0.0163$. The length of each training time series is 65 LT. The resulting time series is used as input data $\boldsymbol{u}_{in}(t)$ for training reservoir networks.

Appendix C. Classical reservoir computing hyperparameters

The performance of classical reservoir computers critically depends on the set of chosen hyper-parameters (Racca and Magri 2021). As compared to QRC or RF-QRC with only two hyperparameters reported in Sect. 3.1, in classical reservoir, there are additional two hyperparameters known as the spectral radius ρ and input scaling σ_{in}. We, therefore, use grid search and Bayesian optimization to tune the hyperparameters (Snoek et al. 2012). We have used the scikit-learn library (Pedregosa et al. 2011) and recycle validation techniques (Racca and Magri 2021) for hyperparameter tuning in classical reservoir computers. The hyperparameter ranges for different chaotic systems are reported in Table 3.

Table 3 Parameters for the Lorenz-63 and MFE system for the classical reservoir computing architecture

Parameters	Symbol	Lorenz-63 system	MFE system
Time step	dt	0.01s	0.25s
Leading Lyapunov exponent	λ_1	0.9056	0.0163
Lyapunov Time	LT	1 LT = 110 steps	1 LT = 245 steps
Washout steps	N_W	5 LT	2 LT
Training steps	N_{tr}	20 LT	65 LT
Spectral radius	ρ	[0.1, 1]	[0.1, 1]
Input scaling	σ_{in}	[0, 1]	[0, 1]
Tikhonov regularization	β	1×10^{-6},1×10^{-9},1×10^{-12}	1×10^{-6},1×10^{-9},1×10^{-12}
Leak rate	ϵ	[0.05, 0.3]	[0.05, 0.3]
Resevoir density	D	0.1,0.6,0.9	0.1,0.6,0.9

Acknowledgements O.A. thanks Defne E. Ozan for fruitful discussions on classical reservoir computing. The authors acknowledge financial support from the UKRI New Horizon grant EP/X017249/1. L.M. is grateful for the support from the ERC Starting Grant PhyCo 949388, and F.T. acknowledges support from the UKRI AI for Net Zero grant EP/Y005619/1.

Author Contributions Osama Ahmed: data curation (lead) formal analysis (equal) investigation (equal) methodology (equal) software (lead) validation (lead) writing—original draft (equal) writing—review and editing (equal). Felix Tennie: formal analysis (equal) investigation (equal) supervision (supporting) validation (equal) writing—review and editing (equal). Luca Magri: conceptualization (lead) funding acquisition (lead) investigation (equal) methodology (equal) project administration (lead) resources (lead) supervision (lead) validation (equal) writing—original draft (supporting) writing—review and editing (equal)

Data Availability The data that support the findings of this study can be accessed online via this link.

Declarations

Conflict of Interest The authors declare no competing interests.

References

Aaronson S (2018) Shadow tomography of quantum states. In: Proceedings of the 50th annual ACM SIGACT symposium on theory of computing, pp 325–338

Ahmed O, Tennie F, Magri L (2024) Prediction of chaotic dynamics and extreme events: a recurrence-free quantum reservoir computing approach. Phys Rev Res 6(4):043082

Boffetta G, Cencini M, Falcioni M et al (2002) Predictability: a way to characterize complexity. Phys Reports 356(6):367–474

Carroll TL (2020) Dimension of reservoir computers. Chaos: An Interdisciplinary J Nonlinear Sci 30(1)

Carroll TL, Pecora LM (2019) Network structure effects in reservoir computers. Chaos: An Interdiscip J Nonlinear Sci 29(8)

Doan NAK, Polifke W, Magri L (2019) Physics-informed echo state networks for chaotic systems forecasting. In: Rodrigues JMF, Cardoso PJS, Monteiro J et al (eds) Computational Science - ICCS 2019. Springer International Publishing, Cham, pp 192–198

Domingo L, Carlo G, Borondo F (2023) Taking advantage of noise in quantum reservoir computing. Sci Reports 13(1):8790

Dudas J, Carles B, Plouet E, et al (2023) Quantum reservoir computing implementation on coherently coupled quantum oscillators. npj Quantum Inf 9(1):1–7. https://doi.org/10.1038/s41534-023-00734-4, https://www.nature.com/articles/s41534-023-00734-4, number: 1 Publisher: Nature Publishing Group

Fry D, Deshmukh A, Chen SYC, Rastunkov V, Markov V (2023) Optimizing quantum noise-induced reservoir computing for nonlinear and chaotic time series prediction. Sci Rep 13(1):19326

Fujii K, Nakajima K (2021) Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices, Springer Singapore, Singapore, pp 423–450. https://doi.org/10.1007/978-981-13-1687-6_18,

GABBIANI F, COX SJ (2010) Chapter 10 - reduced single neuron models. In: GABBIANI F, COX SJ (eds) Mathematics for Neuroscientists. Academic Press, London, pp 143–154https://doi.org/10.1016/B978-0-12-374882-9.00010-1, https://www.sciencedirect.com/science/article/pii/B9780123748829000101

Hu F, Angelatos G, Khan SA et al (2023) Tackling sampling noise in physical systems for machine learning applications: fundamental limits and eigentasks. Phys Rev X 13(4):041020

Huang HY, Kueng R, Preskill J (2020) Predicting many properties of a quantum system from very few measurements. Nat Phys 16(10):1050–1057

IBM Quantum https://quantum.ibm.com/ (2024)

Jaeger H (2001) 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

Jaeger H, Mantas, Popovici D, et al (2007) Optimization and applications of echo state networks with leaky- integrator neurons. Neural Netw 20(3):335–352. https://doi.org/10.1016/j.neunet.2007.04.016, https://www.sciencedirect.com/science/article/pii/S089360800700041X, echo State Networks and Liquid State Machines

Jha SK, Yadava R (2010) Denoising by singular value decomposition and its application to electronic nose data processing. IEEE Sensors J 11(1):35–44

Khan SA, Hu F, Angelatos G, et al (2021) Physical reservoir computing using finitely-sampled quantum systems. arXiv:2110.13849

Khan SA, Kaufman R, Mesits B, et al (2023) Practical trainable temporal post-processor for multi-state quantum measurement. arXiv:2310.18519

Kobayashi K, Fujii K, Yamamoto N (2024) Feedback-driven quantum reservoir computing for time-series analysis. arXiv:2406.15783

Kreplin DA, Roth M (2024) Reduction of finite sampling noise in quantum neural networks. Quantum 8:1385

Lansky P, Ditlevsen S (2008) A review of the methods for signal estimation in stochastic diffusion leaky integrate-and-fire neuronal models. Biol Cybernet 99(4):253–262

Lorenz E (1995) Predictability: a problem partly solved. PhD thesis, Shinfield Park, Reading

Lukosevicius M (2012) A practical guide to applying echo state networks. In: Montavon G, Orr GB, Muller KR (eds) Neural Networks: Tricks of the trade, Second Edition. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp 659–686, https://doi.org/10.1007/978-3-642-35289-8_36

Maass W, Natschläger T, Markram H (2002) Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Computation 14(11):2531–2560https://doi.org/10.1162/089976602760407955 https://direct.mit.edu/neco/article-pdf/14/11/2531/815288/089976602760407955.pdf

McClean JR, Boixo S, Smelyanskiy VN et al (2018) Barren plateaus in quantum neural network training landscapes. Nat Commun 9(1):4812

Moehlis J, Faisst H, Eckhardt B (2004) A low-dimensional model for turbulent shear flows. New J Phys 6:56–56

Mujal P, Martínez-Peña R, Nokkala J, et al (2021) Opportunities in quantum reservoir computing and extreme learning machines. https://doi.org/10.48550/arXiv.2102.11831, https://ui.adsabs.harvard.edu/abs/2021arXiv210211831M, publication Title: arXiv e-prints ADS Bibcode: 2021arXiv210211831M

Mujal P, Martínez-Peña R, Giorgi GL, et al (2023) Time-series quantum reservoir computing with weak and projective measurements. npj Quantum Inf 9(1):16. https://doi.org/10.1038/s41534-023-00682-z

Nielsen MA, Chuang IL (2011) Quantum computation and quantum information: 10th. Cambridge University Press, Anniversary

Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Pfeffer P, Heyder F, Schumacher J (2022) Hybrid quantum-classical reservoir computing of thermal convection flow. Phys Rev Res 4(3):033176

Pfeffer P, Heyder F, Schumacher J (2023) Reduced-order modeling of two-dimensional turbulent Rayleigh-Bénard flow by hybrid quantum-classical reservoir computing. Phys Rev Res 5(4):043242

Preskill J (1998) Fault-tolerant quantum computation. In: Introduction to quantum computation and information. World Sci pp 213–269

Qiskit contributors (2023) Qiskit: an open-source framework for quantum computing. https://doi.org/10.5281/zenodo.2573505

Racca A, Magri L (2021) Robust optimization and validation of echo state networks for learning chaotic dynamics. Neural Net 142:252–268

Racca A, Magri L (2022) Data-driven prediction and control of extreme events in a chaotic flow. Phys Rev Fluids 7(10):104402

Schafer RW (2011) What is a Savitzky-Golay filter?[lecture notes]. IEEE Signal Process Mag 28(4):111–117

Schanze T (2017) Removing noise in biomedical signal recordings by singular value decomposition. Current Directions Biomed Eng 3(2):253–256

Schuld M, Petruccione F (2021) Machine learning with quantum computers. Springer

Shlens J (2014) A tutorial on principal component analysis. arXiv:1404.1100

Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. Adv Neural Inf Process Syst 25

Sornsaeng A, Dangniam N, Chotibut T (2024) Quantum next generation reservoir computing: an efficient quantum algorithm for forecasting quantum dynamics. Quant Mach Intell 6(2):57

Srinivasan PA, Guastoni L, Azizpour H et al (2019) Predictions of turbulent shear flows using deep neural networks. Phys Rev Fluids 4(5):054603

Suzuki Y, Gao Q, Pradel KC et al (2022) Natural quantum reservoir computing for temporal information processing. Sci Reports 12(1):1353

Teeter C, Iyer R, Menon V et al (2018) Generalized leaky integrate-and-fire models classify multiple neuron types. Nat Commun 9(1):709

Tennie F, Laizet S, Lloyd S, Magri L (2025) Quantum computing for nonlinear differential equations and turbulence. Nat Rev Phys 1–11

Xiong W, Facelli G, Sahebi M, Agnel O, Chotibut T, Thanasilp S, Holmes Z (2025) On fundamental aspects of quantum extreme learning machines. Quant Mach Intell 7(1):20

Yu M, Liu Y, Yang P, et al (2022) Quantum fisher information measurement and verification of the quantum Cramér–Rao bound in a solid-state qubit. npj Quantum Inf 8(1):56