Higher-Order Quantum Reservoir Computing

Quoc Hoan Tran  
University of Tokyo  
k09tranhoan@gmail.com

Kohei Nakajima  
University of Tokyo  
k_nakajima@mech.t.u-tokyo.ac.jp

Abstract

Quantum reservoir computing (QRC) is an emerging paradigm for harnessing the natural dynamics of quantum systems as computational resources that can be used for temporal machine learning (ML) tasks. In the current setup, QRC is difficult to deal with high-dimensional data and has a major drawback of scalability in physical implementations. We propose higher-order QRC, a hybrid quantum-classical framework consisting of multiple but small quantum systems that are mutually communicated via classical connections like linear feedback. By utilizing the advantages of both classical and quantum techniques, our framework enables an efficient implementation to boost the scalability and performance of QRC. We demonstrate the effectiveness of our framework in emulating large-scale nonlinear dynamical systems, including complex spatiotemporal chaos, which outperforms many of the existing ML techniques in certain situations.

1 Introduction

It is postulated that quantum computers may outperform classical computers when it comes to ML tasks due to the superior ability of quantum mechanics to generate counter-intuitive patterns [1]. Quantum machine learning (QML) is an active interdisciplinary research area proposed from this motivation to improve existing ML methods through the advantages of quantum mechanics [2]. For the foreseeable future, QML algorithms are expected to run on noisy intermediate-scale quantum (NISQ) devices, which includes a few tens of qubits and supports only non-error corrected computations [3]. Existing quantum techniques are utilized to their fullest extent in ML tasks on these devices via hybrid quantum–classical methods that combine classical learning regimes with the advantages of quantum systems [4, 5].

Reservoir computing (RC) [6–9] is a framework that originated from recurrent neural networks to efficiently solve temporal ML tasks. Conventional RC consists of a randomly connected network called a reservoir and a trainable readout part for pattern analysis from output states of the reservoir. The input stream is fed into the reservoir, which functions like a nonlinear processing unit to project low-dimensional input into a high-dimensional dynamical system. Since the training in RC is simple and extremely fast, it is highly suitable and amendable for hardware implementation in a wide variety of physical systems [10–16]. Quantum reservoir computing (QRC) is a variation of RC, where the reservoir is implemented as a quantum many-body system such as a set of interacting qubits [17, 18] or a set of fermions [19, 20] driven by a Hamiltonian dynamics. The random connections in the classical reservoir are replaced by basic quantum tunneling. The input stream then drives the transition state, which is evolved through a unitary operator based on the dynamics of the system. Thereafter, quantum measurements are performed to obtain signals that can be considered as reservoir states for training [Fig. 1(a)]. Recently, the nuclear-magnetic-resonance (NMR) spin-ensemble in a molecular solid [21] and the quantum circuits on superconducting quantum processors [22] have been reported as the physical implementations of QRC. However, aspects such as the difficulty in using multidimensional inputs or the operating limitations when increasing the number of qubits remain uncertain.
In this paper, we propose a general hybrid quantum-classical framework for RC, namely, higher-order quantum reservoir computing (HQRC). We aim to utilize ensemble small quantum systems as a big reservoir and enhance its computational power through random connections between sub-systems [Fig. 1(b)]. Our scheme is called “higher-order” in terms of different quantum systems placed in an ensemble reservoir, where each system can be considered as a node in the reservoir. Multidimensional inputs can be easily fed into the system by injecting each element to each sub-system. The input of each sub-system is a linear combination of the common input streams and the signals obtained from other systems. This scheme enables to equip a massive amount of computational nodes with the controllable linear feedback, which is expected to increase the expressive power of the system.

Contribution. We provide a detailed procedure for the temporal processing in HQRC with theoretical explanations of parameter design, which are useful for evaluating a quantum reservoir (QR) as a learning system. We numerically verify that the computational power of QRC, like the memory capacity can be enhanced through higher-order settings. We further demonstrate the effectiveness of our proposal in emulating nonlinear dynamical systems, including high-dimensional spatiotemporal chaos. Interestingly, the experimental results indicate that our approach performs comparatively with classical approaches and even outperforms them in certain situations, such as in the limitation of the training data. From the physical implementation perspective, our framework also paves several opportunities for effective design of recently proposed experimental platforms for QRC [21, 22].

2 Preliminaries

We describe some standard notions that are necessary in our analysis. First, we define $\mathcal{M}_{m \times n}$ and $\mathbb{R}_{m \times n}$ as the sets of $m \times n$-dimensional matrix with complex elements and real elements, respectively. For $U \in \mathcal{M}_{m \times n}$, we denote $U^\dagger \in \mathcal{M}_{n \times m}$ as the conjugate transpose of $U$. A complex square matrix $U$ is called a unitary matrix if $UU^\dagger = U^\dagger U$ is the identity matrix, and a Hermitian matrix if $U = U^\dagger$.

Pure and mixed states. A pure state $|\psi\rangle$ of a spin is a two-dimensional complex vector spanned by the eigenstates $\{|0\rangle, |1\rangle\}$ of the Pauli operator $\sigma^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. We denote $\langle\psi| = (|\psi\rangle)^\dagger$; therefore, $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\langle 0| = \begin{bmatrix} 1 & 0 \end{bmatrix}$, and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\langle 1| = \begin{bmatrix} 0 & 1 \end{bmatrix}$. Given a quantum system $S$ comprising $N$ qubits, the Hilbert space $\mathcal{H}_S$ of $S$ is a tensor product space of two-dimensional individual spin
spaces. A pure quantum state is represented by a normalized \(2^N\)-dimensional vector \(|\Psi\rangle\). Generally, the quantum state can be considered a statistical mixture of pure states, which is described by a density matrix \(\rho \in \mathcal{M}_{2^N \times 2^N}\). For a pure state \(|\Psi\rangle\), the density matrix is defined as \(\rho = |\Psi\rangle \langle \Psi|\).

If we consider a quantum state of the system that is sampled from a set of pure states \(\{|\Psi_k\rangle\}\) with a probability distribution \(\{p_k\}\), the density matrix is given by \(\rho = \sum_k p_k |\Psi_k\rangle \langle \Psi_k|\). The density matrix \(\rho\) is Hermitian and positive semidefinite; moreover, \(\text{Tr}(\rho) = 1\), where \(\text{Tr}(X)\) denotes the trace of square matrix \(X\).

**Partial trace.** Given a composite system \(S\) of two subsystems \(S_1\) and \(S_2\), the Hilbert space of \(S\) is the tensor product \(\mathcal{H}_{S_1} \otimes \mathcal{H}_{S_2}\). From the density matrix \(\rho\) of \(S\), we can recover the marginal “reduced density matrix” for a subsystem via the partial trace operation \(\text{Tr}_1(\rho)\), which is defined as the linear extension of the mapping \(\text{Tr}_1 : A_1 \otimes A_2 \mapsto \text{Tr}(A_1)A_2\) for matrices \(A_1 \in \mathcal{H}_{S_1}, A_2 \in \mathcal{H}_{S_2}\).

**Observables and time evolution.** Observables of quantum system \(S\) correspond to Hermitian matrices \(\mathcal{O} \in \mathcal{M}_{2^N \times 2^N}\). If we perform measurements on these observables at state \(\rho\), it means that \(S\) is interacting with measuring apparatus in the presence of the surrounding environment. The outcomes of the measurement are recorded on the dial on the measuring apparatus, and the statistics of measurement outcomes is determined by the expected value of \(\mathcal{O}\) as \(\langle \mathcal{O} \rangle = \text{Tr}[\rho \mathcal{O}]\). If \(S\) is a closed system, the time evolution is generated by a Hamiltonian \(H\) via the Schrödinger equation \(i\hbar \frac{d}{dt} |\Psi(t)\rangle = H |\Psi(t)\rangle\), where \(H \in \mathcal{M}_{2^N \times 2^N}\) is the Hamiltonian matrix that defines the system dynamics. For a time \(\tau\), the evolution from \(t\) to \(t + \tau\) is given by \(|\Psi(t + \tau)\rangle = U_\tau |\Psi(t)\rangle\), or by an expression in terms of density matrix \(\rho(t + \tau) = U_\tau \rho(t) U_\tau^\dagger\), where \(U_\tau = e^{-iH\tau}\) is a unitary matrix. Most generally, the time evolution law is described by a map called a CPTP map \(\mathcal{L} : \rho \mapsto \rho'\) with the following properties: linear, trace preserving, Hermiticity preserving, and completely positive.\(^1\)

### 3 Quantum reservoir computing

We briefly explain the background of QRC. For simplicity, we consider a one-dimensional input and output case. We are given an input sequence \(u = \{u_1, \ldots, u_L\}\) and the corresponding target sequence \(\hat{y} = \{\hat{y}_1, \ldots, \hat{y}_L\}\), where \(u_k\) is a continuous variable in [0, 1]. QRC emulates a nonlinear function \(\mathcal{D}\) to produce the output \(y_k = \mathcal{D}(u, \rho^{(0)}, \{u_1\}_T)\) \((k = 1, \ldots, L)\). Here, \(\rho^{(0)}\) is the initial state of the quantum system, and \(u\) is the parameter that needs to be optimized. A temporal learning task consists of three phases: a washout phase, a training phase, and an evaluation phase. In the washout phase, the system evolves for the first \(T\) transient steps to washout the initial conditions from the dynamics. The training phase to optimize \(u\) is performed with training data \(\{(u_k)_{k=T}, \{y_k\}_{k=T}\}\), where \(1 \leq T < L_1 < L\), such that the mean-square error between \(y_k\) and \(\hat{y}_k\) over \(k = T, \ldots, L_1\) becomes minimum. The trained parameter \(u\) is used to generate outputs in the evaluation phase.

For an \(N\)-qubits system, at time \(t = (k - 1)\tau\), the input \(u_k \in [0, 1]\) is fed to the system by setting the density matrix of the first spin to \(\rho_{u_k} = (1 - u_k) |0\rangle \langle 0| + u_k |1\rangle \langle 1| \in \mathcal{M}_{2 \times 2}\). Therefore, the density matrix \(\rho \in \mathcal{M}_{2^N \times 2^N}\) of the entire system is mapped by a CPTP map

\[
\rho \mapsto \mathcal{T}_{u_k}(\rho) = \rho_{u_k} \otimes \text{Tr}_1[\rho],
\]

where \(\text{Tr}_1\) denotes a partial trace with respect to the first qubit. After the input is set, the system continues evolving itself during time interval \(\tau\). The dynamics are governed by the Schrödinger equation and the information of the input sequence encoded in the first spin spreads through the system. It follows that the state of the system before the next input \(u_{k+1}\) is \(\rho^{(k)} = e^{-iH\tau} \mathcal{T}_{u_k}(\rho^{(k-1)}) e^{iH\tau}\), where \(\rho^{(k)} = \rho(k\tau)\) is the density matrix at \(t = k\tau\). It is noted that the size of \(p_k\) grows exponentially with the number of qubits. We obtain partial information regarding \(\rho^{(k)}\) by measuring local observables on qubits. The signals for training are obtained from average values of measurement results on each qubit. More precisely, if we employ the ordered basis \(\{|\Theta_j\rangle\}\) in the operator space, then the observed signals at time \(t\) are the first \(N_{\text{out}}\) elements \(s_j(t) = \text{Tr}[\rho(t) \Theta_j]\), where the selection of observables depends on the physical implementation of the system. For example, if we consider the spin receiving the input is an ancilla qubit that we cannot perform the measurement on it, the observed operators act only on the other spins. Since this does not change our main points, for the sake of convenience, we consider a situation in which we can perform measurements on all qubits.

\(^1\)A map \(\mathcal{L}\) is completely positive if \(\mathcal{L} \otimes \mathbb{1}_2\) is positive for any extension \(\mathcal{H}_2\) of the Hilbert space \(\mathcal{H}_1\).
In practical applications, the temporal multiplexing scheme is introduced to improve the performance in extracting dynamics. Here, the signals are measured not only at time \( k\tau \) but also at each of the subdivided \( V \) time intervals during the evolution in interval \( \tau \) to construct \( V \) virtual nodes. The density matrix is then updated by

\[
\rho((k-1)\tau + \frac{1}{V}\tau) = U(\tau/V)\mathcal{T}_{uk}(\rho((k-1)\tau))U^\dagger(\tau/V),
\]

\[
\rho((k-1)\tau + \frac{v}{V}\tau) = U(\tau/V)\rho((k-1)\tau + \frac{v-1}{V}\tau)U^\dagger(\tau/V)(v = 2, \ldots, V),
\]

where \( U(\tau/V) = e^{-iH(\tau/V)} \). Therefore, we can obtain temporal signals from \( N_{\text{out}}V \) nodes, and then the learning procedure is straightforward as we parameterize the linear readout function as

\[
\rho = \sum_{n=0}^{N_{\text{out}}} w_i x_{ki},
\]

where \( x_{ki} = s_{ij}((k-1)\tau + \frac{v}{V}\tau) \) for \( i = (j-1)V + v > 0 \) (1 \leq v \leq V; 1 \leq j \leq N_{\text{out}}). Here, \( x_{k0} = 1.0 \) are introduced as constant bias terms, and \( w = [w_0 \ w_1 \ \ldots \ \ w_{N_{\text{out}}V}]^T \) represents the readout weight parameters. If we denote \( K \) as the number of time steps used in the training phase, \( w \) is optimized via the linear regression, or the Ridge regression in the matrix form

\[
\hat{\rho} = (X^\top X + \beta I)^{-1}X^\top \hat{y}.
\]

Here, \( \hat{y} = [\hat{y}_1 \ \ldots \ \hat{y}_K]^\top \) is the target sequence. \( X = (x_{ki}) \in \mathcal{R}_{K \times (N_{\text{out}}V+1)} \) is the training data matrix and \( \beta \) is the parameter serves as the positive constant shifting the diagonals introduced to avoid the problem of the near-singular moment matrix.

We employ the fully connected transverse field Ising model, which is the standard workhorse to build the QR. The Hamiltonian is given by \( H = J \sum_{i \neq j} h_{i,j} \sigma^z_i \sigma^z_j + J \sum_{j} g_{j} \sigma^z_j \), where all the spins interact with each other in \( z \)-direction and are coupled to an external magnetic field in \( z \)-direction. Here, \( \sigma^z_j \ (\gamma \in \{x, y, z\}) \) is the operator measuring the spin \( j \) along the \( \gamma \) direction, which can be described as an \( N \)-tensor product of \( 2 \times 2 \)-matrices

\[
\sigma^z = I \otimes \ldots \otimes \sigma^z \otimes \ldots \otimes I,
\]

where \( I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \), \( \sigma^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \), \( \sigma^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \), and \( \sigma^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \). \( J \) is the coupling magnitude of the Hamiltonian, while the coupling parameter \( h_{i,j} \) and the transverse field parameter \( g_{j} \) are taken uniformly from \([-1.0, 1.0]\). We select \( N_{\text{out}} = N \) observables \( \mathcal{O}_j = \sigma^z_j \) to produce the signals of readout nodes [23].

## 4 Higher-order quantum reservoir computing

### 4.1 Model

We propose an effective and practical design to enhance the computational ability of the QR. Our higher-order quantum reservoir (HQR) consists of an ensemble of \( N_q \) QRs, such that the \( l \)th system \( \mathcal{Q}_l \) has the Hamiltonian \( H_l \) with \( N_l \) qubits (which are known as true nodes) and \( V_l \) virtual nodes. We denote HQR-\( n \) as the HQR comprising \( n \) QRs. For one-dimensional input, these QR systems are driven by a common input stream, while they can receive different input streams for the multidimensional input setting in general. Here, we consider the setting of one-dimensional input for simplicity.\(^2\)

Algorithm 1 presents the temporal processing steps of the HQR. We denote \( \rho_l^{(k)} \) as the density matrix of \( \mathcal{Q}_l \) at time \( t = k\tau \), where \( \rho_l^{(0)} \) is the initialized density matrix. The reservoir states of \( \mathcal{Q}_l \) at time \( t = k\tau \) are represented by a \( V_l N_l \)-dimensional vector \( z_{kl} \), which is initialized at \( t = 0 \) as the zero vector. At \( t = (k-1)\tau \), the input \( u_{kl} \), which is injected into \( \mathcal{Q}_l \), is the linear combination between the \( u_0 \) and the linear-scaled reservoir states from other QRs (step 3). The connection coefficients vectors \( c_{lm} \in \mathcal{R}_{(V_mN_m) \times 1} \) from \( \mathcal{Q}_m \) to \( \mathcal{Q}_l \) are randomly generated and fixed, such that elements in \( c_{lm} \) are nonnegative and \( \sum_{m \neq l} \sum_{c \in c_{lm}} c = \alpha (0 \leq \alpha \leq 1) \), where \( \alpha \) is defined as the connection strength parameter. After injecting \( u_{kl} \) into \( \mathcal{Q}_l \), \( \rho_l \) is transformed by the CPTP map, \( \mathcal{T}_{ukl} \), and is then consequently evolved in each \( \tau/V \) time. The training is performed with the temporal reservoir states of the entire system, which are rewritten with bias terms as the matrix \( X \in \mathcal{R}_{K \times (\sum_{l=1}^{N_q} N_lV_l+1)} \).\(^3\)

---

\(^2\)To fed a \( M \)-dimensional input \( u \) into the HQR, we can transform \( u \) into a \( N_q \)-dimensional input via the linear transformation \( u' = W_{in}u \) where \( W_{in} \in \mathcal{R}_{N_q \times M} \) is fixed randomly.
Algorithm 1 Temporal processing of higher-order quantum reservoir

Require: The input stream \( \{u_k\} \), \( N_{qr} \) QRs \( Q_l \) with corresponding configurations \( (H_l, N_l, V_l, \rho_l^{(0)}) \), the time interval \( \tau \), and the nonnegative connection coefficient vectors \( c_{lm} \in \mathcal{R}_{(V_m, N_m) \times 1} \) \((m \neq l) \) such that \( \sum_{m \neq l} \sum_{c} c_m = \alpha \leq 1 \). \( z_{kl}^{(l)} \in \mathcal{R}_{(V_l, N_l)} \times \) are denoted as the linear scaled reservoir states of \( Q_l \) at \( t = k\tau \) by scaling \( z \to (z + 1)/2 \) such that their elements are in \([0, 1]\).

1: for input \( u_k \) do
2: \hspace{1cm} for \( l = 1, \ldots, N_{qr} \) do
3: \hspace{2cm} \( u_{kl} \leftarrow (1 - \alpha)u_k + \alpha \sum_{m \neq l} c_{ml}^{(l)} z_{(k-1)m}^{(l)} \)
4: \hspace{2cm} for \( v = 1, \ldots, V_l \) do
5: \hspace{3cm} \( \rho_l \leftarrow \mathcal{T}_{u_{kl}}(\rho_l^{(k-1)}) \) \hspace{0.5cm} \( \triangleright \) \( \mathcal{T}_{u_{kl}} \) is a CPTP map defined in Eq. (1)
6: \hspace{3cm} \( \rho_l \leftarrow e^{-\tau_{v_l}^{(l)}} \rho_l e^{\tau_{v_l}^{(l)}} \) \hspace{0.5cm} \( \triangleright \) \( Q_l \) is evolved in time interval \( \tau/V_l \)
7: \hspace{2cm} for \( j = 1, \ldots, N_l \) do
8: \hspace{3cm} \( z_{k_{l}v_{lj}} \leftarrow \text{Tr}[\rho_l \sigma_j^{(l)}] \) \hspace{0.5cm} \( \triangleright \) Measure the average spin values in the \( z \)-direction
9: \hspace{2cm} end for
10: end for
11: \( z_{kl} \leftarrow [(z_{k_{l}v_{lj}})_{v_{lj}}]^\top \in \mathcal{R}_{(V_l, N_l) \times 1} \)
12: \( \rho_l^{(k)} \leftarrow \rho_l \)
13: end for
14: \( z_k \leftarrow [(z_{k1})^\top \ldots (z_{kN_{qr}})^\top]^\top \in \mathcal{R}_{(\sum_{l=1}^{N_{qr}} V_l N_l) \times 1} \)
15: end for
16: return Reservoir temporal states: \( z = \{z_1, z_2, \ldots\} \)

4.2 Properties of higher-order quantum reservoir dynamics

In the classical regime, the reservoir is required to satisfy an asymptotic stability property termed the echo state property (ESP), which ensures that the computations are performed independently with the initial state of the reservoir [24]. ESP is a similar concept with the fading memory property, which states that the reservoir must produce the close outputs if the corresponding inputs are close in recent times [25]. We explore the asymptotic behavior and memory capacity of HQR systems.

Asymptotic stability. A “robust” HQR must produce trajectories that are robust to small perturbations to the system—that is, the computations performed by an HQR system for the same input are independent of its initial density matrix. We define this as quantum echo state property (QESP).

Definition 1 An HQR system whose reservoir dynamics are governed by Algorithm 1 is said to satisfy the quantum echo state property (QESP) when for each initial density matrix \( \rho^{(0)}, \hat{\rho}^{(0)} \), and for any input sequence \( u_L = \{u_l\}_{l=1}^{L} \) of length \( L \), it holds that \( \|\rho^{(L)} - \hat{\rho}^{(L)}\|_p \to 0 \) as \( L \to \infty \).

Here, \( \| \cdot \|_p \) denotes the Schatten p-norm for \( p \geq 1 \), defined as \( \|A\|_p = \text{Tr}[\sqrt{A^\dagger A}]^{1/p} \) for matrix \( A \).

From Eq. (1) and Algorithm 1, we can rewrite the evolution from \( t = (k-1)\tau \) to \( t = k\tau \) as a CPTP map \( \mathcal{L}_k \), such that \( \rho^{(k)} = \mathcal{L}_k(\rho^{(k-1)}) = \mathcal{L}_k \circ \mathcal{L}_{k-1} \circ \ldots \circ \mathcal{L}_1(\rho^{(0)}) \). Since a CPTP map is a contraction map, the density matrices satisfy decreasing system distinguishability—that is, \( \|\rho^{(k)} - \hat{\rho}^{(k)}\|_p \leq \|\rho^{(k-1)} - \hat{\rho}^{(k-1)}\|_p \). However, it is not trivial to theoretically explore QESP in higher-order settings [26, 27]. Instead, we consider an empirical perspective to the study of the asymptotic stability. We extend the algorithm in [28] to define the QESP index to evaluate the average deviation of observed signals generated from random initial states to reference signals starting from a fixed state [29]. The first \( T \) time-steps are discarded as the washout phase, and the QESP index is averaged over \( P \) randomly generated initial density matrices. Figure 2(a) presents the average QESP indexes along the variation of \( J \) and \( \tau \) over 10 random trials for the HQR-5 of six qubits, \( V = 1 \), and the connection strength \( \alpha = 0.5 \) for the range of dynamics in \([T, T + 1000]\) \((T = 9000 \) time steps\). The stabilizing effect is empirically confirmed by increasing \( J\tau \), which increases the contractivity of \( \mathcal{L}_k \). The empirical QESP validity is characterized by a border of a sharp stable-unstable transition.
Figure 2: (a) The QESP index, (b) the MC of the HQR-5 (α = 0.5, V = 1) according to τ with different settings of J, (c) the MC of the HQR-5 (J = 1.0, τ = 0.5) according to α with different settings of V, and (d)(e) the bifurcation diagrams with different configurations of T and inputs. For all the plots, the error bars show the standard deviations.

**Memory capacity (MC).** The property of the HQR in storing information of recent inputs to the current states is commonly measured by memory capacity MC = \( \sum_{d=0}^{\infty} \text{MF}(d) \), where \( \text{MF}(d) \) is the memory function that evaluates the capacity to reconstruct the previous \( d \) steps of the input [30]. This implies that if we set the input as a random sequence \( \{u_k\} \) in \([0, 1]\), then the forecast and target are \( y_k \) and \( \hat{y}_k = u_{k-d}, \) respectively. \( \text{MF}(d) \) is then defined as \( \text{MF}(d) = \frac{\text{cov}(x, y)}{\sigma(x)\sigma(y)} \in [0, 1] \), where \( \text{cov}(x, y) \) and \( \sigma(x) \) express the covariance and the standard deviation, respectively. A dynamical system with high MC is superior for temporal forecasting tasks that need to utilize historical patterns. Figure 2(b) depicts the MC as the functions of τ with different settings of J in the HQR-5 of six qubits, \( V = 1 \), and \( \alpha = 0.5 \). The number of time steps used in washout, training, and evaluation phases are 1000, 3000, and 1000, respectively. MC is evaluated with delay \( d = 0, \ldots, 200 \); then it is averaged over 10 random trials for each parameter setting. The connection strength \( \alpha \) measures the dominant role of the reservoir states in the feedback, which increases the ability of memorizing past patterns. As shown in Fig. 2(c), increasing \( \alpha \) while keeping the external input (\( \alpha < 1 \)) enables the extensive MC. Furthermore, increasing \( V \) leads to an increase in the MC, while the MC becomes saturated around \( V = 15 \).

As shown in Fig. 2(b), for each \( J \), there exists an optimal \( \tau \) to maximize the MC, which can be understood as a trade-off between the influence of the input and the effect of the relaxation dynamics in the QR. Between two consecutive inputs \( u_k \) and \( u_{k+1} \), the QR continues evolving for a time interval \( \tau \) as \( \xi(\tau) = e^{-iH\tau}\xi(0)e^{iH\tau} \), where we denote \( \xi(0) \) as the density matrix after setting \( u_k \). The auto-correlation function \( \langle \xi(0)\xi(\tau) \rangle \) reduces to the spin glass order parameter with the exponential decay coefficient \( \lambda = \Delta J \), where \( \Delta \) is the smallest nonzero eigengap of \( H/J \) [31] (in our Hamiltonian, changing \( J \) does not affect the value of \( \Delta \)). If \( J\tau \) is too small, the dynamics approach an identity map, therefore the signals are almost linear without prediction power. If \( J\tau \) is too large, the dynamics relax exponentially fast between inputs, thereby decreasing the effects of past inputs into the system.

We further look into the dynamics of the HQR-5 via the bifurcation diagrams where we consider \( \tau \) is the driving parameter, \( J = 1.0, V = 1 \). In Fig. 2(d), the values of four representative signals in one QR (excluding the signals from the input qubit) are overlayed for the range of dynamics in \([T, T + 1000]\). We consider the diagram at \( \alpha = 0.5 \) as an example to explore three intriguing regimes: (A) the dynamics approach fixed points for long enough transient time \( T \), (C) the dynamics exhibits two separated regions in responding to the external inputs, and (B) the transition region between (A) and (C). We note that the left part of (A) appears as a broad band even if the external input is withdrawn (\( \alpha = 1.0 \)), or setting the constant inputs without the feedback (\( \alpha = 0.0 \)) [Fig. 2(e)]. Therefore, this part remains as the effect of the initial state during the short transient time \( T \). As we increase \( T \), the region of \( \tau \) in (B) will extend to the left. While setting \( \tau \) in (B) leads to the optimal trade-off for the effect of the past input and the relaxation dynamics, which gives the maximum MC, in practical applications we should set a slightly longer \( \tau \) to avoid the effect of the initial state.
Figure 3: (a) The averaged NMSE for the NARMA tasks according to the number \( N_{qr} \) of QR systems in HQR and the connection strength \( \alpha \). (b) Typical output time series for the NARMA tasks of the HQR-5. Other parameters for (a)(b) are \( V = 20 \), \( J = 1.0 \), and \( \tau = 2.0 \).

5 Benchmarks

5.1 NARMA tasks

We first demonstrate the performance of the HQRC in the NARMA benchmark, which is commonly used for evaluating the computational capability for temporal processing with long time dependence. The NARMA system is formulated as the \( n \)th-order nonlinear dynamical system, which has the following form:

\[
y_k = \kappa y_{k-1} + \eta y_{k-1} \left( \sum_{j=0}^{n-1} y_{k-j} \right) + \gamma u_{k-n+1} u_k + \delta,
\]

where \( \kappa = 0.3 \), \( \eta = 0.05 \), \( \gamma = 1.5 \), and \( \delta = 0.1 \) in our experiments. We consider \( n = 5, 10, 15, 20 \), where the corresponding systems are NARMA5, NARMA10, NARMA15, and NARMA20, respectively. To set \( y_k \) into the stable range, we linearly scale the input \( u_k \) from \([0, 1]\) to \([0, 0.2]\). The number of time steps are set as 2000, 2000, and 2000 for the washout, the training, and the evaluation phase, respectively. The performance is evaluated using the normalized mean-squared error (NMSE) metric,

\[
\text{NMSE} = \frac{\sum_{k=4001}^{6000} (y_k - \hat{y}_k)^2}{\sum_{k=4001}^{6000} \hat{y}_k^2},
\]

where \( y_k \) and \( \hat{y}_k \) are the prediction and the target of system at time step \( k \).

We present two types of connections in HQR: the mutual connection where any two QRs are mutually connected, and the forward connection where each QR only connects with the next QR in a forward direction. Figure 3(a) illustrates the averaged NMSE over 10 random trials for the HQRs comprising \( N_{qr} = 1–5 \) QRs with five qubits, \( V = 20 \), and \( \alpha = 0.0–0.9 \). We set \( J \) and \( \tau \) in the stable region of the QESP index such as \( J = 1.0 \) and \( \tau = 2.0 \). These values are close to the stable-unstable border in Fig. 2(a) to increase the MC, but at the beginning of (C) as shown in Fig. 2(d) to avoid the effect of the initial state. Since higher-order NARMA tasks require high memory to predict, increasing \( \alpha \) will boost the performance for both types of connections. Figure 3(b) illustrates the typical outputs for the HQR-5 in the evaluation phase. The outputs with higher \( \alpha \) fit well to the targets for all tasks, even for a difficult task like NARMA20.

5.2 Emulating chaotic systems

Next, we demonstrate the performance of the HQRC on emulating chaotic systems, particularly for high-dimensional input that the normal QRC is unable to implement. In this task, the system learns the input of the next step. After learning the readout weights to fit the targets in the stage called teacher forcing stage, the external input streams are withdrawn and the output is fed into the input so that the system is closed. Thereafter, the system can automatically evolve and replicate the dynamical evolution of the target system but within a relatively short prediction horizon. We define
the normalized root mean square error $\text{NRMSE}(y_t) = \sqrt{\frac{1}{M} \sum_i (y_{ti} - \hat{y}_{ti})^2 / \hat{\sigma}_i^2}$, where $y_{ti} = (y_{ti})_i$ and $\hat{y}_{ti} = (\hat{y}_{ti})_i$ are the $M$-dimensional forecast and target, and $\hat{\sigma}_i$ is the standard deviation of the target in time of each component $i$. As referenced from Ref. [35], to evaluate the prediction performance, we compute the valid prediction time $\text{VPT} = \Lambda_{-1}^{-1} \arg\max_{t_f} \{\text{NRMSE}(y_t) \leq \varepsilon, \forall t \leq t_f \}$, which is the largest time $t_f$ (normalized with respect to $\Lambda_1$ of the chaotic system) where the NRMSE error is smaller than $\varepsilon$ ($\varepsilon = 0.5$ in our experiments). Large VPT means long prediction horizon in the performance of the model.

We employ two chaotic systems: the Lorenz attractor and the Kuramoto-Sivashinsky equation (KSE) [32, 33] with spatiotemporally chaotic solutions. The Lorenz attractor is given by three ordinary differential equations:

$$
\frac{dx}{dt} = a(y - x), \\
\frac{dy}{dt} = x(b - z) - y, \\
\frac{dz}{dt} = xy - cz,
$$

where $(a, b, c) = (10, 28, 8/3)$. The time series is obtained by the fourth-order Runge-Kutta method with step size $\Delta t = 0.01$, and the model attempts to predict $x(t + \Delta t)$ from $x(t)$. The first $K = 10^3$ steps are used for training, and $10^3$ steps are predicted iteratively. The input signals are min-max scaled to be in the range $[0, 1]$. Our HQR-5 of six qubits, with $\tau = 4.0, J = 2.0$, and $V \in \{10, 15, 20\}$, corresponds with $6 \times 5 \times V$ computational nodes. For chaos emulating tasks, the reservoir needs to learn an arbitrarily good approximation of the chaotic recurrence. The non-linear transformation is required, but adding it will degrade the MC [34], thus, we need careful considerations on this memory nonlinear trade-off of the reservoir [29]. In this context, $\alpha$ adjusts this trade-off, which is tuned for the best performance in our experiments. We adopt the framework in [35] to generate the simulation data and to implement the common ML models such as long short-term memory (LSTM) [36], gated recurrent unit (GRU) [37], and echo state network (ESN) [7]—for example, LSTM-$n$ for the LSTM with $n$ hidden units, and ESN-$n$ for the ESN with $n$ nodes, where $n \in \{80, 100, 120, 150, 500, 1000, 1500, 3000\}$. Here, we set the number of layers in the LSTM and GRU equal to one. The box plots in Fig. 4(a) demonstrate the distribution of the VPT over 100 random tests for the HQR model, and the two best results in each of the other models. HQR-5 shows superior performance, as its VPTs are highest, while other models cannot capture long time steps in the prediction as shown in the typical examples placed in the right plots of Fig. 4(a). As demonstrated in Fig. 4(b), for different $K$, the HQR are still better even with a small number of the computational nodes ($V = 15$ corresponds with 450 nodes). Here, the best results of the ESN, LSTM, and GRU models are selected to plot in Fig. 4(b).
The KSE is the partial differential equation $\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = 0$ for the scalar function $u(x, t)$ in the interval $x \in [0, L]$ with periodic boundary conditions $u(x, t) = u(x + L, t)$. The KSE is integrated on a spatial domain of $M = 64$ uniform subintervals with $\Delta t = 0.25$, $L = 22$, thereby yielding a simulation of $M$-dimensional time series $\{u_k\}$. We employ the parallel architecture in [38] to build 32 HQRs, where each HQR predicts a spatially 2-dimensional local region. We divide the input $u_k$ into 32-local groups $u_k^{(i)}$, where $u_k^{(i)}$ comprises the $i$th and $(i + 1)$th element in $u_k$.

The $i$th HQR is the HQR-10 with six qubits to receive 10-dimensional inputs to predict $u_k^{(i)}$. The input for each HQR is the concatenated vector of $u_k^{(j)}$ for $j = i - 2, i - 1, i, i + 1, i + 2$. We set $\tau = 4.0, J = 2.0, V \in \{10, 15, 20, 25\}$, and tune $\alpha$ for the best performance. The first $K = 10^4$ steps are used for training, and 400 steps are predicted iteratively. We employ the same parallel architecture for ESN-$n$, LSTM-$n$, and GRU-$n$ models, where $n \in \{80, 100, 120, 150, 500, 1000\}$ is the number of nodes or hidden units in each group. Figure 4(c) illustrates contours and error plots for the typical forecasts of HQR and ESN, which demonstrates the ability of the HQR to emulate the spatiotemporal chaos in approximately two Lyapunov time $\Lambda^{-1}$. Figure 4(d) illustrates the evolution of NRMSE averaged over 10 random tests for each model. The HQR outperforms other models even with a small number of computational nodes. This may be mainly due to the exponential numbers of degrees of freedoms behind the quantum measurements, which leads to the quantum computational supremacy region.

6 Conclusion and discussion

In this paper, we propose the HQRC framework, which offers an effective potential means for using quantum dynamics in machine learning tasks. It is suggested that QRC has higher expressive power than classical RC, even for the same number of computational nodes; however, technical scalability is a major drawback. Our framework introduces an implementation to solve this problem. Here, local operations are performed on each QR and the results of those operations are communicated in a classical manner. The experimental results on emulating nonlinear systems, including high-dimensional spatiotemporal chaos, demonstrate that our framework can boost the computational power and scalability of QRC through both quantum and classical advantages. Since only time evolution, according to the Hamiltonian as an interaction between nodes, is permitted in QRC, the design for an arbitrary nonlinear function remains as a future challenge. A complete exploration of capabilities of other quantum systems as reservoirs and determining the optimal setting for the classical communications can be a possible direction for the future research.

Broader Impact

As a positive impact in the foreseeable future, our framework can be a typical case for quantum machine learning in the NISQ era for utilizing the noise-robust property of the classical system while exploiting the quantum computational supremacy region. The flexibility of selecting the physical system as a computational resource, the low operation cost when running the hybrid quantum-classical system consisting of multiple small quantum systems, and exhibiting high computational power can enable our research to be utilized in real-world applications. Currently, this theoretical work does not present any foreseeable negative societal consequence. We believe that the experimental efforts in implementing physical quantum reservoir systems, such as the NMR ensemble system [21], the random and noisy quantum circuits on super conducting quantum processors [22], and the photonics system, can have the most benefit from our work.

Acknowledgments

We thank Y. Hasegawa, T. Van Vu, V. T. Vo, and T. Goto for insightful discussions and P. R. Vlachas for providing the detailed instructions to run the framework in Ref. [35]. We thank Y. Hasegawa and K. Inoue for providing the additional but powerful computational resources to perform our numerical experiments. This work was based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO), Q. H. T. and K. N. were supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant No. JPMXS0118067394.


References

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, Sep 2017.

[2] C. Ciliberto, M. Herbst, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Woessnig. Quantum machine learning: a classical perspective. *Proc. R. Soc. A*, 474(2209):20170551, Jan 2018.

[3] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, Aug 2018.

[4] S. D. Sarma, D.-L. Deng, and L.-M. Duan. Machine learning meets quantum physics. *Phys. Today*, 72(3):48–54, Mar 2019.

[5] G. Torlai and R. G. Melko. Machine-learning quantum states in the NISQ era. *Annu. Rev. Condens. Matter Phys.*, 11(1):325–344, Mar 2020.

[6] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.*, 14(11):2531–2560, Nov 2002.

[7] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, Apr 2004.

[8] D. Verstraeten, B. Schrauwen, M. D’Haene, and D. Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391–403, Apr 2007.

[9] M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.*, 3(3):127–149, Aug 2009.

[10] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer. Information processing using a single dynamical node as complex system. *Nat. Commun.*, 2(1):1–6, Sep 2011.

[11] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express*, 20(3):3241–3249, Jan 2012.

[12] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, K. Yakuhi, A. Fukushima, et al. Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547(7664):428–431, Jul 2017.

[13] T. Furuta, K. Fujii, K. Nakajima, S. Tsunegi, H. Kubota, Y. Suzuki, and S. Miwa. Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions. *Phys. Rev. Applied*, 10:034063, Sep 2018.

[14] K. Nakajima, H. Hauser, R. Kang, E. Guglielmino, D. G. Caldwell, and R. Pfeifer. A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm. *Front. Comput. Neurosci.*, 7:91, Jul 2013.

[15] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer. Exploiting the dynamics of soft materials for machine learning. *Soft robotics*, 5(3):339–347, Jun 2018.

[16] K. Nakajima. Physical reservoir computing—an introductory perspective. *Jpn. J. Appl. Phys.*, 59(6):060501, May 2020.

[17] K. Fuji and K. Nakajima. Harnessing disordered-ensemble quantum dynamics for machine learning. *Phys. Rev. Applied*, 8:024030, Aug 2017.

[18] K. Nakajima, K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa. Boosting computational power through spatial multiplexing in quantum reservoir computing. *Phys. Rev. Applied*, 11:034021, Mar 2019.

[19] S. Ghosh, A. Opala, M. Matuszewski, T. Paterek, and T. C. Liew. Quantum reservoir processing. *Npj Quantum Inf.*, 5(1):1–6, Apr 2019.

[20] S. Ghosh, T. Paterek, and T. C. H. Liew. Quantum neuromorphic platform for quantum state preparation. *Phys. Rev. Lett.*, 123:260404, Dec 2019.

[21] M. Negoro, K. Mitarai, K. Fujii, K. Nakajima, and M. Kitagawa. Machine learning with controllable quantum dynamics of a nuclear spin ensemble in a solid. *Preprint at arXiv:1806.10910*, 2018.
Since the Schrödinger equation is linear, it may appear that the system does not correspond to nonlinear dynamics. However, the nonlinearity is emergent from higher-order correlations or nonlinear terms, which are mixed by the linear but nonintegrable dynamics via $U_\tau$ [17].

H. Jaeger. 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(34):13, Jan 2001.

S. Boyd and L. Chua. Fading memory and the problem of approximating nonlinear operators with volterra series. *IEEE Trans. Circuits Syst.*, 32(11):1150–1161, Nov 1985.

In the particular case of the constant input without the interaction between the QR systems, we have $L_k \equiv \mathcal{L}$ for all $k$ and $\rho^{(k)} = \mathcal{L}^k(\rho^{(0)})$. The convergence property of $\rho^{(k)}$ can be explored via the spectrum of $\mathcal{L}$ [27].

L. Bruneau, A. Joye, and M. Merkli. Repeated interactions in open quantum systems. *J. Math. Phys.*, 55(7):075204, Jul 2014.

C. Gallicchio. Chasing the echo state property. *Preprint at arXiv:1811.10892*, 2018.

See Supplemental Materials for a detailed description of the QESP and MC calculations, the parameter settings for the ESN, LSTM, and GRU models, and the additional figures of our experiments.

H. Jaeger. *Short term memory in echo state networks*, volume 5. GMD-Forschungszentrum Informationstechnik, 2001.

A. Dutta, G. Aeppli, B. K. Chakrabarti, U. Divakaran, T. F. Rosenbaum, and D. Sen. *Quantum Phase Transitions in Transverse Field Spin Models*. Cambridge University Press, 2015.

G. Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations. *Acta Astronaut.*, 4(11-12):1177–1206, Nov 1977.

Y. Kuramoto. Diffusion-induced chaos in reaction systems. *Prog. Theor. Phys. Supp.*, 64:346–367, Feb 1978.

S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *Proc. Natl. Acad. Sci. U. S. A.*, 105(48):18970–18975, Nov 2008.

P. Vlachas, J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191 – 217, Jun 2020.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, Nov 1997.

K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct 2014. Association for Computational Linguistics.

J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120:024102, Jan 2018.

The maximum Lyapunov exponents of the Lorenz attractor system and the KSE in our study are 0.9056 [40] and 0.05 [41], respectively.

D. Viswanath and L. N. Trefethen. *Lyapunov Exponents from Random Fibonacci Sequences to the Lorenz Equations*. PhD thesis, USA, 1998. AAI9839919.

H. Fan, J. Jiang, C. Zhang, X. Wang, and Y.-C. Lai. Long-term prediction of chaotic systems with machine learning. *Phys. Rev. Research*, 2:012080, Mar 2020.
This supplementary material provides a detailed description of the calculations, the experiments introduced in the main text, and the additional figures. The equation, figure, and table numbers in this section are prefixed with S (e.g., Eq. (S1) or Fig. S1, Table S1), while numbers without the prefix (e.g., Eq. (1) or Fig. 1, Table 1) refer to items in the main text.
Algorithm S1 Calculate the quantum echo state property (QESP) index

Require: Higher-order quantum reservoir system $Q_l$ of $N_{qr}$ QR systems $Q_l (l = 1, \ldots, N_{qr})$, number of washout time steps $T$, number of evaluation time steps $E$, number of trials $P$. Denote $F(\rho^{(0)}, \{u_i\}_{i=1}^P) = [x_{k,1}, \ldots, x_{k,N_{total}}]^T$ as the reservoir states of the system ($N_{total} = \sum_i N_i V_i$) at time $t = k\tau$ for input sequence $\{u_i\}_{i=1}^P$ and the initial density matrix $\rho^{(0)}$.

1: Initialize the density matrices $\rho^{(0)}_l (l = 1, \ldots, N_{qr})$.
2: $\rho^{(0)}_l \leftarrow \rho^{(0)}_l \otimes \rho^{(0)}_{N_{qr}}$
3: for $p = 1, \ldots, P$ do
4: Initialize the density matrices $\sigma^{(0)}_l (l = 1, \ldots, N_{qr})$.
5: $\sigma^{(0)}_l \leftarrow \sigma^{(0)}_l \otimes \sigma^{(0)}_{N_{qr}}$
6: for $k = 1, \ldots, E$ do
7: $\delta_p(k) = \|F(\rho^{(0)}_l, \{u_{i}^{k+T}\} - F(\sigma^{(0)}_l, \{u_{i}^{k+T}\})\|$
8: end for
9: $\Delta_o = \langle\delta_p(k)\rangle_k$
10: end for
11: $\Delta \leftarrow \langle\Delta_o\rangle_p$ \quad \triangleright Average over the trials
12: return The QESP index $\Delta$

MEMORY CAPACITY

The property of a reservoir system in storing information of recent inputs to the current states is commonly measured by memory capacity $MC = \sum_{d=0}^{\infty} MF(d)$, where $MF(d)$ is the memory function that evaluates the capacity of system to reconstruct the previous $d$ steps of the input. To calculate $MF(d)$, the system learns to reproduce the previously injected random input of $d$ time steps by using the current states. This implies that if we set the input as a random sequence $u_k$ in $[0, 1]$, then the forecast and target are $y_k$ and $\hat{y}_k = u_{k-d}$, respectively. $MF(d)$ is then defined as

$$MF(d) = \frac{\text{cov}^2(y_k, \hat{y}_k)}{\sigma^2(y_k)\sigma^2(\hat{y}_k)} \in [0, 1],$$

where $\text{cov}(x, y)$ and $\sigma(x)$ express the covariance and the standard deviation, respectively. We note that a dynamical system with high memory capacity is superior for the temporal forecasting tasks that need to utilize historical patterns. In Fig. S1, $MF(d)$ is plotted as a function of $d$ for the HQR system, which contains $N_{qr}$ QR systems. Hereafter, we denote HQR-n for the HQR system consisting of $n$ QR systems. We consider the coupling magnitude $J = 1.0$ in the Hamiltonian, the number of virtual nodes $V = 1$ in each QR system, the number of QR systems $N_{qr} = 1.5$, and the connection strength $\alpha = 0.0, 0.9$ with different settings of the time interval $\tau$ to inject the input. Here, each QR has five qubits and the number of time steps used in the washout, training, and evaluation phases are 1000, 3000, and 1000, respectively. The memory function is evaluated on 10 samples of HQR with respect to the random coupling. It is evident that increasing $N_{qr}$ will increase the value of $MF(d)$ for any $\tau$, thereby indicating the relatively large value of $MF(d)$ in the region of the large delay $d$. For the HQR-5 system, the connection strength $\alpha = 0$ implies that we have a spatial multiplexing setup where all QR systems are disjoint. By introducing the feedback, for example, $\alpha = 0.9$, the values of $MF(d)$ are higher in the region of the small delay, $d$.

Figure S2 shows $MC$ as functions of $\tau$ for the HQR along with different $\alpha$, $N_{qr}$ [Fig. S2(a)], $J$ [Fig. S2(b)], and $V$ [Fig. S2(c)]. MC is calculated for delay $d = 0, 1, \ldots, 200$, and is averaged over 10 different runs with random trials of Hamiltonian coefficients and connection coefficients in the HQR. We observe from Fig. S2(a) and Fig. S2(c) that increasing $N_{qr}$ and $V$ will lead to an increase in the MC of the system, while the MC becomes saturated around $V = 15$ [Fig. S2(c)].

It is interesting to see the optimal values of $J, \tau$ in Fig. S2(b) to maximize the memory capacity. These optimal values are close to the border of stable-unstable transition for the QESP index presented in Fig. 2(a) in the main text. The theoretical reasons for this behavior can be understood as a trade-off between the influence of the input and the effect of the relaxation dynamics in the QR. Between two consecutive inputs $u_k$ and $u_{k+1}$, the QR continues evolving for a time interval $\tau$ as $\xi(\tau) = e^{-iH\tau}\xi(0)e^{iH\tau}$, where we denote $\xi(0)$ as the density matrix after setting $u_k$, and $H$ is the Hamiltonian of the QR. The auto correlation function $G(\tau) = \langle\xi(0)\xi(\tau)\rangle$ at the absolute zero temperature can be written as the ground state (the state of the lowest possible energy) average of the operator $\xi(\tau)$ as the following
FIG. S1. The memory functions \( MF(d) \) of the HQR system according to delay \( d \) with different settings \( N_{qr}, \alpha, \) and \( \tau \). The memory functions are evaluated as the average functions on 10 samples of HQR with respect to random coupling.

FIG. S2. Memory capacity shown as a function of the time interval \( \tau \) for the HQR system, which is formed by \( N_{qr} \) QR systems of five qubits. The number of time steps used in the washout, training, and evaluation phases are 1000, 3000, and 1000, respectively. The memory capacity is calculated for delay \( d = 0, 1, \ldots, 200 \), and is averaged over 10 random trials of different runs. For all the plots, the error bars indicate the standard deviations. (a) \( V = 1, J = 1.0 \), (b) \( V = 1, N_{qr} = 5 \), (c) \( J = 1.0, N_{qr} = 5 \).

FIG. S3. The signals \( \text{Tr}[\rho \sigma_i^z] \) obtained from \( i \)th qubit (excluding the signals from the input qubits) in one QR system for window length \( 5\tau \) of HQR-5 with the connection strength \( \alpha = 0.5 \) and the coupling magnitude \( J = 1.0 \).
form:

\[ G(\tau) = \langle 0 | \xi(\tau) \xi(0) | 0 \rangle = \sum_m e^{-(E_m - E_0)\tau} |\langle 0 | \xi(\tau) | m \rangle|^2, \quad (S2) \]

where \( \{ |m\} \) are the eigenstates of \( H \), which correspond with the energies \( \{E_m\} \). Here, \( E_0 \) is the lowest energy, that is, the smallest eigenvalue of \( H \). Equation (S2) implies that the time auto correlation function will decay exponentially with the dominant term \( \exp(-\tau \Delta_J) \), where \( \Delta_J = J \Delta \) is the smallest nonzero eigengap of \( H \). Here, \( \Delta \) is the smallest nonzero eigengap of \( H/J \) (in our Hamiltonian, changing \( J \) does not affect the value of \( \Delta \)). We plot in Fig. S3(a) the signals \( \text{Tr}[\rho \sigma_z^i] \) obtained from \( i \)th qubit (excluding the signals from the input qubits) in one QR system for different time windows of HQR-5 system \((\alpha = 0.5, J = 1.0) \). If \( J \tau \) is too small, the dynamics approach an identity map, therefore the signals are almost linear without prediction power. If \( J \tau \) is too large, the signals in the interval \( \tau \) are too chaotic because the dynamics relax exponentially fast between inputs, thereby decreasing the effects of past inputs into the system.

MACHINE LEARNING MODELS FOR TIME SERIES FORECASTING

We briefly explain the conventional machine learning models used in the main text for time-series forecasting. One can refer to Ref. [2] for more detailed explanations and implementations in the software framework. Here, the models are trained on the time-series \( \{\hat{\mathbf{o}}_1, \ldots, \hat{\mathbf{o}}_t\} \) of an observable \( \hat{\mathbf{o}} \in \mathbb{R}^{d_o} \). The temporal information of the observable is encoded via the internal high-dimensional state denoted by \( \mathbf{h}_t \in \mathbb{R}^{d_h} \), where \( d_h \) is the number of hidden units. Given the time series \( \{\hat{\mathbf{o}}_1, \ldots, \hat{\mathbf{o}}_t\} \), based on the hidden states, the models output the value \( \mathbf{o}_{t+1} \) as the prediction for the next target \( \hat{\mathbf{o}}_{t+1} \). This temporal processing can be modeled by the following equations:

\[ h_t = f_h^h(\hat{\mathbf{o}}_t, h_{t-1}), \quad \hat{\mathbf{o}}_{t+1} = f_h^o(h_t), \quad (S3) \]

where \( f_h^h \) is the hidden-to-hidden mapping and \( f_h^o \) is the hidden-to-output mapping.

Echo State Network (ESN)

We consider the reservoir computing framework implemented in the context of echo state network (ESN) [3]. Here, the hidden-to-hidden mapping \( f_h^h \) is given by

\[ h_t = \tanh(\mathbf{W}_{h,i} \hat{\mathbf{o}}_t + \mathbf{W}_{h,h} h_{t-1}), \quad (S4) \]

where \( \mathbf{W}_{h,i} \in \mathbb{R}^{d_h \times d_o} \) and \( \mathbf{W}_{h,h} \in \mathbb{R}^{d_h \times d_h} \) are fixed randomly. The elements of \( \mathbf{W}_{h,i} \) are generated from a uniform distribution in \([-\omega, \omega]\). Moreover, to make the system satisfy the Echo State Property, the matrix \( \mathbf{W}_{h,h} \) is often set as a large low-degree matrix with its spectral radius (absolute value of the largest eigenvalue) is in a finite range \((< 1.0)\). This condition can be satisfied by properly normalizing the elements in \( \mathbf{W}_{h,h} \). The hidden-to-output mapping is set to

\[ \mathbf{o}_{t+1} = \mathbf{W}_{o,h} h_t, \quad (S5) \]

where \( \mathbf{W}_{o,h} \in \mathbb{R}^{d_o \times d_h} \) is trained via regularized least-squared regression. In practical applications, to enable the stability of ESN in long-term forecasting, Gaussian noise sampled from \( \mathcal{N}(0, \eta_n \sigma) \) is added to the training data. Here, \( \sigma \) is the standard deviation of the data and \( \eta_n \) is the noise level. In our experiments for timeseries forecasting tasks, we set \( \eta_n \) as a tuning parameter in \( \{0, 1e - 3, 5e - 3\} \).

Long Short-Term Memory (LSTM)

The long short-term memory (LSTM) [4] was proposed to deal with the vanishing gradient problem of classical RNNs by utilizing the mechanism that allow information to be forgotten. In LSTM, the hidden-to-hidden mapping is defined by the following recurrent functions:

\[ g_t^l = \sigma_f (\mathbf{W}_f q_t + b_f), \quad g_t^i = \sigma_i (\mathbf{W}_i q_t + b_i), \quad (S6) \]

\[ \tilde{c}_t = \tanh (\mathbf{W}_c q_t + b_c), \quad c_t = g_t^l \odot c_{t-1} + g_t^i \odot \tilde{c}_t, \quad (S7) \]

\[ g_t^o = \sigma_o (\mathbf{W}_o q_t + b_o), \quad h_t = g_t^o \odot \tanh(c_t), \quad (S8) \]

where \( \sigma_f, \sigma_i, \sigma_o \) are the sigmoid functions.
where $\odot$ denotes the element-wise product, and $g_t^f, g_t^i, g_t^o \in \mathbb{R}^{d_h}$ are the forget, input, and output gates signals, respectively. $q_t = [h_{t-1}, \tilde{o}_t] \in \mathbb{R}^{d_h+d_o}$ is the concatenated vector of the hidden state $h_{t-1}$ and the observable input $\tilde{o}_t$, $c_t \in \mathbb{R}^{d_h}$ is the cell state. $W_f, W_i, W_o, W_h \in \mathbb{R}^{d_h \times (d_h+d_o)}$ are weight matrices, and $b_f, b_i, b_o, b_h \in \mathbb{R}^{d_h}$ are bias vectors. The activation functions $\sigma_f, \sigma_i, \sigma_o$ are sigmoid functions. Similar to ESN, the hidden-to-output mapping is set to $o_{t+1} = W_{o,h} h_t$, where $W_{o,h} \in \mathbb{R}^{d_h \times d_o}$. In our study, we use back-propagation through time (BPTT) algorithm to train LSTM and its variations. We referenced Ref. [2] for a more detailed explanation of BPTT and the implementation of BPTT in the software framework with hyper-parameter settings used in our study.

Gated Recurrent Unit (GRU)

The gated recurrent unit (GRU) [5] is a variation of the LSTM without output gates. GRU uses update gate and reset gate instead to decide what information should be passed to the output; thus, information from long ago can be kept in the training process without excluding information that is irrelevant to the prediction. The recurrent mappings of the GRU are given by

$$\begin{align*}
    z_t &= \sigma_g(W_z q_t + b_z), \quad r_t = \sigma_g(W_r q_t + b_r), \\
    \hat{h}_t &= \tanh(W_h p_t + b_h), \quad h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t,
\end{align*}$$

(S9)

(S10)

where $z_t \in \mathbb{R}^{d_h}$ is the update gate vector, $r_t \in \mathbb{R}^{d_h}$ is the reset gate vector, $q_t = [h_{t-1}, \tilde{o}_t] \in \mathbb{R}^{d_h+d_o}$, $p_t = [r_t \odot h_{t-1}, \tilde{o}_t] \in \mathbb{R}^{d_h+d_o}$ are concatenated vectors. The gating activation $\sigma_g$ is a sigmoid function, while $W_z, W_r, W_h \in \mathbb{R}^{d_h \times (d_h+d_o)}$ are weight matrices and $b_z, b_r, b_h \in \mathbb{R}^{d_h}$ are bias vectors. The hidden-to-output mapping is set to $o_{t+1} = W_{o,h} h_t$, where $W_{o,h} \in \mathbb{R}^{d_h \times d_o}$.

EMULATING CHAOTIC SYSTEMS

In our HQR for the tasks of emulating chaos, we employ the simple form of the Hamiltonian:

$$H = J \sum_{i \neq j} h_{i,j} \sigma_i^x \sigma_j^x + Jg \sum_j \sigma_j^z,$$

(S11)

where $g = 2.0$, and $h_{i,j}$ is distributed randomly in $[-1.0, 1.0]$.

The Lorenz attractor

The Lorenz attractor is given by three ordinary differential equations: $dx/dt = a(y-x), dy/dt = x(b-z)-y, dz/dt = xy-cz$, where $(a, b, c) = (10, 28, 8/3)$. The time series is obtained by the fourth-order Runge-Kutta method with step size $\Delta t = 0.01$, and the model attempts to predict $x(t+\Delta t)$ from $x(t)$. The first $K = 10^4$ steps are used for training, and $10^3$ steps are predicted iteratively. The input signals are min-max scaled to be in the range $[0.0,1.0]$. Our HQR comprises five QR systems (HQR-5) of six qubits, with $\tau = 4.0, J = 2.0$, and $V \in \{10, 15, 20\}$, corresponds with $6 \times 5 \times V$ computational nodes. We adopt the framework in Ref. [2] to generate the simulation data and implement the ESN, LSTM and the GRU—for example, LSTM-$n(m)$ for the LSTM with $n$ hidden units and $m$ layers (if $l > 1$), and ESN-$n$ for the ESN with $n$ nodes, where $n \in \{80, 100, 120, 150, 500, 1000, 1500, 3000\}$. We note that in the manuscript, we consider LSTM and GRU model with one layer to obtain the results in Fig. 4(a)(b). In the preparation of this supplemental material, we became aware that increasing the number of layers in LSTM and GRU leads to an improvement in the performance of these models. We present the results here with the number $m$ of layers is set in $\{1, 2, 3\}$. This modification will be addressed in the revised version of our main manuscript. We present in the accompanied software package the full experimental results and the details for the parameter settings.

We evaluate the prediction accuracy using the normalized root mean square error

$$\text{NRMSE}(y_t) = \sqrt{\frac{1}{M} \sum_i \frac{(y_{t_i} - \hat{y}_{t_i})^2}{\hat{\sigma}_i^2}},$$

(S12)
FIG. S4. (a) Box plot showing the distribution of the VPT for the HQR-5 (V=15) with different settings of $\alpha$. (b) Typical predictions of HQR-5 (V=15) with $\alpha = 0.0, 0.1, 0.5, \text{and} 0.9$.

where $y_t = (y_{ti})_i$ and $\hat{y}_t = (\hat{y}_{ti})_i$ are the forecast and target vectors at time $t$, and $\hat{\sigma}_t$ is the standard deviation of the target in time of each component $i$. To evaluate the prediction performance, we compute the valid prediction time (VPT):

$$VPT = \Lambda_1^{-1} \arg\max_{t_f} \{ \text{NRMSE}(y_t) \leq \varepsilon, \forall t \leq t_f \},$$  

which is the largest time $t_f$ (normalized with respect to $\Lambda_1$ of the chaotic system) where the NRMSE error is smaller than $\varepsilon$ ($\varepsilon = 0.5$ in our experiments).

Fig. S4 demonstrates the box plots to depict the distribution of VPT over 100 random tests for HQR-5 model with different settings of the connection strength $\alpha$. We can observe that large $\alpha$ will reduce the average value of VPT. We believe that this may be due to the trade-off of linear and nonlinear dynamics in the reservoir model [7, 8]. If we increase the connection strength $\alpha$ in the HQR model, the reservoir becomes more linear and then extensive memory capacity can be obtained. However, for effective functioning of computing in order to address the linearly inseparable problem, a nonlinear transformation of the input signal is required to introduce into reservoir dynamics. Further, by introducing the nonlinearity in general signal-driven dynamical systems, the memory capacity of the reservoir will be degraded (Jaeger conjecture [9]). If the tasks require extensive memory like NARMA tasks, the linear property must be preferred. Therefore increasing connection strength in HQR leads to an improvement in the performance in NARMA tasks. For chaotic emulating tasks with output feedback, the reservoir needs to learn an arbitrarily good approximation of the chaotic recurrence. Therefore, a non-linear transformation is required in these tasks and, thus, we need careful consideration of the memory nonlinear trade-off of the HQR model. In this context, the connection strength $\alpha$ provides a good parameter to adjust this trade-off, which depends on the learning tasks.

Figures S5–S7 depict box plots to indicate the distribution of VPT over 100 random tests for HQR, LSTM, GRU, and ESN model. Here, the number of time steps for training is set as $K = 10^3$ steps (Fig. S5), $K = 10^4$ steps (Fig. S6), and $K = 10^5$ steps (Fig. S7). We plot HQR-5 for $V=10, 15, 20$; and the three best results (with the highest average VPT) in each of the ESN, LSTM, and GRU model. As shown in Fig. S5(a), with only $K = 10^3$ time steps for training, HQR-5 shows superior performance as its VPTs are highest, while other models cannot capture long time steps in the prediction as shown in the typical examples [Fig. S5(b)]. As demonstrated in Fig. S8, increasing $K$ leads to improvements for LSTM and GRU models. Nevertheless, the HQR are still better even with a small number of the computational nodes ($V = 15$ corresponds with 450 nodes). Here, the best results of the ESN, LSTM, and GRU models are selected to plot in Fig. S8.
FIG. S5. (a) Box plot showing the distribution of the VPT over 100 random predictions. The number of time steps for training is $K = 10^3$ steps. (b) Certain typical predicted time series on the Lorenz system.

FIG. S6. (a) Box plot showing the distribution of the VPT over 100 random predictions. The number of time steps for training is $K = 10^4$ steps. (b) Certain typical predicted time series on the Lorenz system.

FIG. S7. (a) Box plot showing the distribution of the VPT over 100 random predictions. The number of time steps for training is $K = 10^5$ steps. (b) Certain typical predicted time series on the Lorenz system.
The Kuramoto-Sivashinsky equation

The Kuramoto-Sivashinsky equation (KSE) is the partial differential equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4} = 0,$$

(S14)

for the scalar function $u(x, t)$ in the interval $x \in [0, L]$ with periodic boundary conditions $u(x, t) = u(x + L, t)$. The KSE is integrated on a spatial domain of $M = 64$ uniform subintervals with $\Delta t = 0.25, L = 22$, thereby yielding a simulated data set with $M$-dimensional time series, which can be used as multi-dimensional input-target pairs. The simulation is performed using the fourth-order method for stiff PDEs [10]. We utilize the framework in Ref. [2] to simulate the solution of KSE up to $T = 6 \times 10^4$, which corresponds to $24 \times 10^4$ samples. The first $4 \times 10^4$ are excluded for initial transients, and the remaining data are divided into training and testing dataset of $10^5$ samples in each dataset. For the HQR and the ESN model, we employ the augmentation technique proposed in Ref. [11]. Here, the hidden states are augmented such that the hidden states are squared in half of the computational nodes. In order to reproduce the results in the main text, we present the details for the parameter settings in the running scripts of the accompanied software package.

- k09tranhoan@gmail.com
- k_nakajima@mech.t.u-tokyo.ac.jp

[1] A. Dutta, G. Aeppli, B. K. Chakrabarti, U. Divakaran, T. F. Rosenbaum, and D. Sen, Quantum Phase Transitions in Transverse Field Spin Models (Cambridge University Press, 2015).
[2] P. Vlachas, J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics, Neural Networks 126, 191 (2020).
[3] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science 304, 78 (2004).
[4] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Computation 9, 1735 (1997).
[5] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Association for Computational Linguistics, Doha, Qatar, 2014) pp. 1724–1734.
[6] D. Viswanath and L. N. Trefethen, Lyapunov Exponents from Random Fibonacci Sequences to the Lorenz Equations, Ph.D. thesis, USA (1998), aAI9839919.
[7] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, Information processing capacity of dynamical systems, Sci. Rep. 2, 10.1038/srep00514 (2012).
[8] M. Inubushi and K. Yoshimura, Reservoir computing beyond memory-nonlinearity trade-off, Sci. Rep. 7, 10.1038/s41598-017-10257-6 (2017).
[9] S. Ganguli, D. Huh, and H. Sompolinsky, Memory traces in dynamical systems, Proc. Natl. Acad. Sci. U. S. A. 105, 18970 (2008).
[10] A.-K. Kassam and L. N. Trefethen, Fourth-order time-stepping for stiff PDEs, SIAM J. Sci. Comput. 26, 1214 (2005).
[11] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Phys. Rev. Lett. 120, 024102 (2018).