Quantum Reservoir Computing via Raman-Driven Dynamics of a Single Trapped Ion

Jia-Yang Gao
Centre for Quantum Technologies, National University of Singapore, Singapore
Email: e0941514@u.nus.edu

***Abstract*—Reservoir computing leverages the rich nonlinear dynamics of a physical system to transform input streams into high-dimensional features, enabling simple ridge-regression readout training. While many physical platforms have been explored, the use of a single trapped-ion vibrational mode under monochromatic driving remains largely unexplored. Such a system can exhibit rich dynamical behavior—including nonlinear resonances, resonance-cell (localization-like) structures, and chaos-like dynamics—depending on the drive strength and detuning, suggesting potential for reservoir computing. Here, we present a reservoir-computing framework based on the vibrational dynamics of a single trapped ion under Raman-type laser driving. Using numerical simulations, we evaluate the framework on standard benchmarks, including time-series tasks, NARMA, the homogeneous SU(2) Yang–Mills field equation, Mackey–Glass, and Lorenz-63 prediction. Our results show that this system can serve as an effective physical reservoir, achieving strong performance across several benchmarks using only a linear ridge-regression readout. These findings demonstrate a minimal, experimentally accessible route to quantum reservoir computing using a single trapped-ion motional mode.**

***Index Terms*—reservoir computing, quantum dynamics, Lindblad master equation, chaotic time-series prediction**

I. Introduction

Reservoir computing (RC) is a learning paradigm for temporal machine-learning tasks in which a fixed, high-dimensional dynamical system (the reservoir) transforms an input time series into rich features, and only a linear readout is trained [1]–[13]. This separation between physics as a feature map and a simple trained readout makes RC attractive when training internal dynamics is costly or impractical.

Quantum reservoir computing (QRC) extends this idea to quantum dynamics by using an input-driven quantum system as the reservoir [14]–[43]. A key motivation is that quantum systems naturally provide large state spaces and complex transient responses, while maintaining a learning rule that only trains a linear readout. The measured observables provide true nodes, while the unobserved degrees of freedom act as hidden nodes that can enrich the effective feature map [15].

Various schemes for developing quantum reservoirs have been proposed across different platforms, such as Rydberg atoms [18], and photonic platform [23]. In Ref. [18], the authors propose a fixed-geometry quantum recurrent neural network (quantum reservoir) implemented with interacting spin-1/2 ensembles in optical-tweezer Rydberg arrays and show it can learn cognitive-style temporal tasks including multitasking, decision making, and long-term memory by taking advantage of several key features of this platform such as interatomic species interactions and quantum many-body scars. As another example, Ref. [23] proposes a scalable photonic architecture for *real-time* QRC using a physical ensemble of identical optical pulses recirculating in a closed-loop (fiber) setup, with the reservoir continuously driven by inputs and read out via homodyne detection to estimate output observables without external buffering.

In this work we target a trapped-ion–motivated quantum reservoir in which the dynamical substrate is the motional mode of a single ion, modeled as a quantum harmonic oscillator (QHO) driven by an effective traveling-wave potential [44], [45]. Concretely, two far-detuned laser beams (pump and Stokes), taken to be plane-polarized along the quantization axis, generate a Raman-type interaction whose interference term produces a spatially and temporally periodic potential. Neglecting constant light-shift terms, the resulting effective motional Hamiltonian along the weak trap axis takes the form

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{m\omega^2\hat{x}^2}{2} + \frac{\varepsilon}{k}\cos\big(k\hat{x} - \Omega t\big), \tag{1}$$

where $\Omega = \omega_p - \omega_s$ is the pump–stokes frequency difference and $k = (\mathbf{k}_p - \mathbf{k}_s)\cdot\mathbf{e}_x$ is the wave-vector projection along the motional axis.

Because the Hamiltonian is periodic in time, the dynamics admits a Floquet (quasienergy) description [44], [45]. Expanding the state in the harmonic-oscillator basis and Fourier harmonics yields a tight-binding–like eigenproblem on a two-dimensional lattice whose sites are labeled by the oscillator level and Fourier index. The level-dependent hopping amplitudes inherit an oscillatory dependence on the action and can become near-zero at specific actions, producing dynamical barriers that partition Hilbert space into resonance cells: Floquet states are typically extended within a cell but strongly suppressed from crossing between cells. As the drive is increased, the corresponding classical phase space develops large chaotic regions, and the associated Floquet states spread over these chaotic seas, providing a controllable route to richer nonlinear dynamics well-suited for reservoir computing.

Building on these ideas, we develop a simulation framework that implements an input-driven (optionally dissipative) QHO reservoir. We extract virtual nodes via time multiplexing and increases feature diversity via spatial multiplexing [15], [46]; and trains a ridge-regression readout for standard RC benchmarks, including timer tasks, NARMA, homogeneous

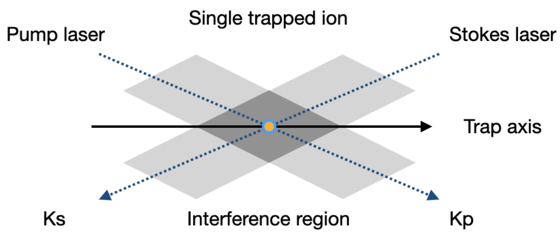


Fig. 1: Schematic of the single-trapped-ion reservoir and Raman beam geometry. A single ion is confined in a harmonic trap; the motional mode along the weak axis x serves as the reservoir degree of freedom. Two far-detuned laser beams (pump p and Stokes s), linearly polarized along the quantization axis, generate an effective Raman interaction whose interference produces a traveling-wave optical potential along x. The resulting drive has beat frequency $\Omega = \omega_p - \omega_s$ and wave-vector projection $k = (\mathbf{k}_p - \mathbf{k}_s) \cdot \mathbf{e}_x$, leading to an effective modulation $\propto \cos(k\hat{x} - \Omega t + \phi)$ after neglecting constant light-shift terms.

SU(2) Yang–Mills field equation, Mackey–Glass, and Lorenz-63 prediction.

The remainder of the paper is organized as follows. Section II reviews the driven-oscillator physics and summarizes the Raman implementation underlying the effective traveling-wave potential. Section III presents the reservoir construction, input encoding, and linear readout training. Section IV reports benchmark tasks and evaluation metrics. Section V concludes and discusses outlook.

II. Background: Driven Oscillator as a Physics-Motivated Reservoir

Our reservoir model is based on a single trapped-ion motional mode subject to a traveling-wave optical potential generated by a far-detuned Raman configuration. The resulting driven-oscillator Hamiltonian is well established in the ion-trap literature; here we summarize only the ingredients needed for reservoir computing (see Refs. [44], [47] for the full derivation).

A. Raman implementation and effective optical potential

We consider the motional mode along the weak confinement axis x, with bare Hamiltonian

$$\hat{H}_0 = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2. \tag{2}$$

Two optical fields (pump p and stokes s) are detuned by a detuning Δ (GHz level) from an electronic excited manifold. In the far-detuned regime, the excited manifold is only virtually occupied and can be adiabatically eliminated, producing an effective interaction within the long-lived ground-state manifold. Under the conditions used in Ref. [44], the dominant contribution relevant to the motional dynamics is a position-dependent scalar light shift proportional to the local intensity.

For linearly polarized fields along the quantization axis, the interference of the two beams generates a beat-note at $\Omega = \omega_p - \omega_s$ and a spatial modulation set by the wave-vector difference. Dropping the constant terms (which only contribute an overall phase), the motional degree of freedom is described by the driven-oscillator Hamiltonian

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{m\omega^2\hat{x}^2}{2} + \frac{\varepsilon}{k}\cos(k\hat{x} - \Omega t), \tag{3}$$

where

$$k = (\mathbf{k}_p - \mathbf{k}_s) \cdot \mathbf{e}_x, \tag{4}$$

$$\varepsilon = \frac{A\pi\epsilon_0 k}{2k_0^3\Delta}\,|E^{(p)}E^{(s)*}|. \tag{5}$$

Here A is the Einstein A coefficient for the relevant optical transition, k_0 is the resonant wave number, and $E^{(p)}, E^{(s)}$ are the complex field amplitudes. The form (3) is the starting point for our reservoir construction.

B. Dimensionless formulation

For analysis and simulation we work with dimensionless variables

$$\tau = \omega t, \qquad \xi = kx, \qquad \ell = \frac{I}{\hbar}, \tag{6}$$

and scaled Hamiltonians

$$\mathcal{H}_0 = \frac{H_0}{\hbar\omega}, \qquad \mathcal{H}_{\text{int}} = \frac{H_{\text{int}}}{\hbar\omega}, \qquad \mathcal{H} = \frac{H}{\hbar\omega}. \tag{7}$$

This introduces the dimensionless control parameters

$$\epsilon = \frac{\varepsilon k}{m\omega^2}, \qquad \eta^2 = \frac{\hbar k^2}{2m\omega}, \qquad \mu = \frac{\Omega}{\omega}, \qquad \delta = N - \mu. \tag{8}$$

Here η is the Lamb–Dicke confinement parameter (playing the role of an effective $\hbar$ in the dimensionless dynamics), ϵ sets the drive strength, and μ controls proximity to the Nth resonance via δ.

C. Motivation for reservoir computing

Equation (3) provides a compact, physics-grounded reservoir substrate with experimentally meaningful knobs: (i) ϵ tunes the strength of the nonlinear drive-induced mixing in the motional basis; (ii) μ controls the degree of resonant energy exchange (and hence memory timescales); and (iii) η sets the effective quantumness through the motional length scale. In our RC implementation, the reservoir state is represented in a truncated motional Fock space (finite energy in experiment), and standard RC readouts are trained on observables derived from the resulting density operator dynamics.

III. Reservoir Model and Readout

A. Truncated oscillator and precomputed nonlinear operators

We model each reservoir block as a single quantum harmonic oscillator truncated to N Fock states $\{|n\rangle\}_{n=0}^{N-1}$. The annihilation operator is

$$a = \sum_{n=0}^{N-2} \sqrt{n+1}\,|n\rangle\langle n+1|, \tag{9}$$

with $a^\dagger$ its adjoint and number operator $n = a^\dagger a$. Working in dimensionless time $\tau = \omega t$, we define the dimensionless position phase

$$\xi = kx = \eta\,(a + a^\dagger), \tag{10}$$

where η is the Lamb–Dicke parameter

$$\eta^2 = \frac{\hbar k^2}{2m\omega}. \tag{11}$$

The dimensionless bare oscillator Hamiltonian is

$$H_0' = \frac{H_0}{\hbar\omega} = n + \frac{1}{2}. \tag{12}$$

To evaluate the monochromatic drive efficiently, we precompute the Hermitian operators $\cos\xi$ and $\sin\xi$ once per choice of (N, η) by diagonalizing ξ: if $\xi = V\,\mathrm{diag}(\lambda)\,V^\dagger$, then

$$\cos\xi = V\,\mathrm{diag}(\cos\lambda)\,V^\dagger, \qquad \sin\xi = V\,\mathrm{diag}(\sin\lambda)\,V^\dagger. \tag{13}$$

These operators are reused at every integration substep.

B. Dimensionless Hamiltonian and input encoding

The system is driven by a traveling-wave potential in the dimensionless form

$$H'(\tau) = H_0' + A(\tau)\cos(\xi - \mu\tau), \tag{14}$$

where $\mu = \Omega/\omega$. Using $\cos(\xi - \mu\tau) = \cos\xi\cos(\mu\tau) + \sin\xi\sin(\mu\tau)$, our implementation evaluates

$$H'(\tau) = H_0' + A(\tau)\,\cos\xi\,\cos(\mu\tau) + A(\tau)\,\sin\xi\,\sin(\mu\tau). \tag{15}$$

A scalar discrete-time input s_k is encoded by modulating the *dimensionless* force parameter ϵ via

$$\epsilon_{\mathrm{eff}}(k) = \epsilon_0 + \epsilon_{\mathrm{scale}}\,s_k. \tag{16}$$

Within the kth discrete reservoir step, $\epsilon_{\mathrm{eff}}(k)$ is held constant and the drive amplitude is set as

$$A(\tau) \equiv A_k = \frac{\epsilon_{\mathrm{eff}}(k)}{2\eta^2}. \tag{17}$$

C. Open-system evolution and discretization

We evolve the density matrix $\rho(\tau)$ under the dimensionless Lindblad master equation [48], [49]:

$$\frac{d\rho}{d\tau} = -i[H'(\tau), \rho] + \tilde{\kappa}\,\mathcal{D}[a]\rho, \tag{18}$$

where $\tilde{\kappa} = \kappa/\omega$ and $\mathcal{D}[L]\rho = L\rho L^\dagger - \frac{1}{2}(L^\dagger L\rho + \rho L^\dagger L)$. Equation (18) is integrated using a fixed-step fourth-order Runge–Kutta (RK4) method. Each discrete reservoir step has duration T^τ_{step} and is subdivided into steps per step t_s RK4 substeps of size

$$\Delta\tau = \frac{T^\tau_{\mathrm{step}}}{t_s}. \tag{19}$$

A key implementation detail is that the internal drive phase is *not* reset between discrete steps. Instead, we maintain a persistent internal time τ_0 so that for input s_k the state is propagated from $\tau = \tau_0$ to $\tau = \tau_0 + T^\tau_{\mathrm{step}}$, after which we update $\tau_0 \leftarrow \tau_0 + T^\tau_{\mathrm{step}}$. This ensures that the phase $\mu\tau$ in $\cos(\xi - \mu\tau)$ is continuous across the full input sequence.

To reduce numerical drift, after each measurement sample we apply a lightweight projection:

$$\rho \leftarrow \frac{\rho + \rho^\dagger}{2}, \qquad \rho \leftarrow \frac{\rho}{\mathrm{Tr}(\rho)}. \tag{20}$$

D. Virtual nodes and observable feature map

Within each discrete step we sample number of virtual nodes V by recording expectation values at approximately evenly spaced substeps (implemented by sampling every $\lfloor t_s/V \rfloor$ RK4 updates, capped at V samples) [15]. Let $\{\hat{O}_\alpha\}_{\alpha=1}^M$ denote a fixed set of observables. In our implementation,

$$\{\hat{O}_\alpha\} = \left\{n,\ x,\ p,\ x^2,\ p^2,\ \tfrac{1}{2}(xp + px),\ \cos\xi,\ \sin\xi\right\}, \tag{21}$$

with quadratures

$$x = \frac{a + a^\dagger}{\sqrt{2}}, \qquad p = \frac{i\,(a^\dagger - a)}{\sqrt{2}}. \tag{22}$$

At the vth virtual node of step k, we compute signed features

$$f_{\alpha,v}(k) = \mathrm{Re}\,\mathrm{Tr}\Big[\rho(\tau_{k,v})\,\hat{O}_\alpha\Big], \tag{23}$$

and concatenate all $V \times M$ values into a block feature vector $f^{(b)}(k) \in \mathbb{R}^{VM}$.

E. Multi-block ensemble and full feature vector

To increase feature diversity, we use an ensemble of B blocks that share $(N, \eta, \mu, \tilde{\kappa}, \epsilon_0, T^\tau_{\mathrm{step}}, t_s, V)$ but differ in their input coupling scales [46]. Specifically, block b uses

$$\epsilon^{(b)}_{\mathrm{scale}} = s_b\,m_b, \tag{24}$$

where $\{s_b\}$ and $\{m_b\}$ are fixed scale grids; by default we use a 3×3 grid, giving $B = 9$. The full reservoir feature vector at time step k is

$$x_k = \Big[\,f^{(1)}(k) \parallel \cdots \parallel f^{(B)}(k) \parallel 1\,\Big] \in \mathbb{R}^{BVM+1}, \tag{25}$$

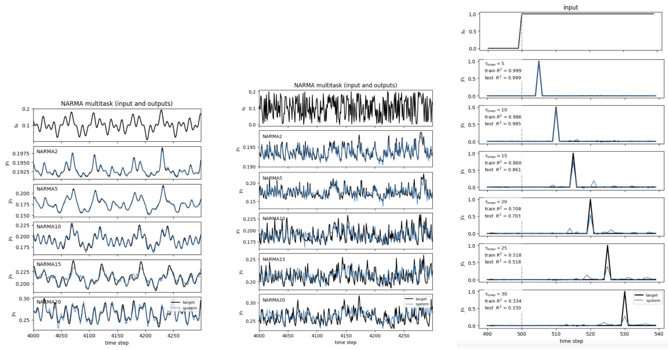
where the final entry is a constant bias feature.

F. Ridge-regression readout

Given training features $X \in \mathbb{R}^{T\times D}$ (rows $x_k^\top$) and targets $Y \in \mathbb{R}^{T\times d}$, we train a linear readout by Tikhonov-regularized least squares [50],

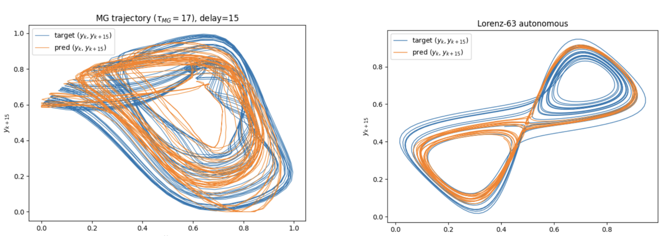
$$W = (X^\top X + \lambda I)^{-1} X^\top Y, \tag{26}$$

and predict $\hat{Y} = XW$. In our implementation we standardize (zero-mean, unit-variance) the non-bias features and the targets using training-set statistics before fitting, and undo the target standardization after prediction.



(a) NARMA, sine-product input (b) NARMA, random input (c) Timer task

Fig. 2: Benchmark performance on nonlinear temporal processing and memory. (a,b) Multitask NARMA evaluation: a single scalar input stream $s_k \in [0, 0.2]$ drives the reservoir and a shared ridge readout predicts five targets $\{y_k^{(n)}\}_{n\in\{2,5,10,15,20\}}$ simultaneously. The plots show a representative segment from the test split (black: target; colored: prediction) for sine-product and random inputs. (c) Timer task: the input switches from 0 to 1 at the cue time k_0, and the readout is trained to produce a single output spike at $k_0 + \tau_{\text{timer}}$.



(a) MG: delayed-state trajectory (y_k, y_{k+15}). (b) L63: delayed-state trajectory (y_k, y_{k+15}) (one normalized coordinate).

Fig. 3: Autonomous prediction (closed-loop rollouts). A one-step ridge readout is trained under teacher forcing ($s_k = y_k$, target y_{k+1}) and then rolled out by feeding back $\hat{y}_k$ as the next input (clipped to $[0, 1]$), initialized from the last teacher-forced input for continuity. MG: $(\alpha, \beta, \gamma, \tau_{\text{MG}}) = (0.2, 10, 0.1, 17)$, Euler with $\Delta t = 0.1$, subsample $\times 10$, discard 2000, min–max to $[0, 1]$. L63: $(\sigma, \rho, \beta) = (10, 28, 8/3)$, RK4 with $\Delta t = 0.01$, discard 2000, per-coordinate min–max to $[0, 1]$ (default: normalized x). In both cases, $T_{\text{train}} = 10{,}000$ and $T_{\text{test}} = 2000$, with small uniform noise added to teacher-forcing inputs.

IV. Benchmarks and Metrics

A. Benchmark tasks and evaluation protocol

We evaluate the reservoir on benchmarks [51] targeting (i) delayed response/timing, (ii) nonlinear temporal processing, and (iii) autonomous prediction of chaotic dynamics. At each discrete time step k the reservoir outputs a feature vector $x_k \in \mathbb{R}^D$ and a linear readout produces

$$\hat{y}_k = w^\top x_k + b, \tag{27}$$

or its vector-valued extension $\hat{\mathbf{y}}_k = W^\top x_k + \mathbf{b}$ for multi-output tasks. Only the readout parameters are trained (ridge regression); reservoir dynamics are not trained.

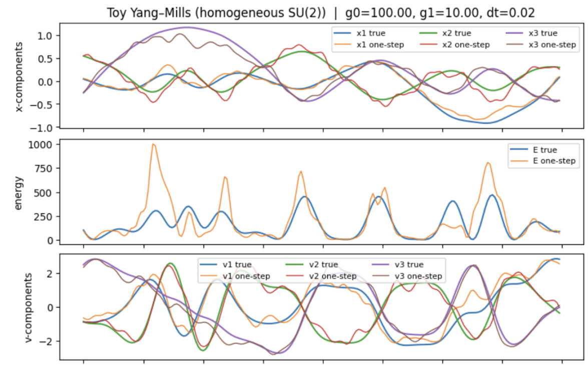


Fig. 4: Toy homogeneous SU(2) Yang–Mills field task (one-step prediction). A 6D macro-state $z_n = (x_1, x_2, x_3, v_1, v_2, v_3)^\top$ is generated by RK4 integration with $\Delta t = 0.02$ for $T_{\text{macro}} = 800$, discarding the first 200 steps and using $n = 200 \ldots 599$ for training and $n = 600 \ldots 799$ for testing. Each z_n is serialized into six scalar ticks to drive the reservoir; features are read after the sixth tick and a ridge readout learns $z_n \mapsto z_{n+1}$ with $\lambda = 10^{-2}$. A coupling quench $g(t)$ switches from g_0 to g_1 at t_q, with $(g_0, g_1, t_q) = (100, 10, 5.0)$ and $u_{\max} = 0.2$. Test performance is reported via per-component NMSE and corr^2 (Table II).

1) Timer task: The timer task tests whether the reservoir can emit a localized response after a prescribed delay following a cue. We use a step input of length $T = 800$ with cue at $k_0 = 500$:

$$s_k = \begin{cases} 0, & k < k_0, \\ 1, & k \geq k_0, \end{cases} \tag{28}$$

and the target is a single spike at $k = k_0 + \tau_{\text{timer}}$:

$$y_k = \begin{cases} 1, & k = k_0 + \tau_{\text{timer}}, \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

Following the multi-trial protocol used in our implementation, for each value of τ_{timer} we reset the reservoir for each trial, discard the first 400 steps as washout, and use the remaining window

$$\mathcal{K}_{\text{timer}} = \{400, \ldots, 799\} \tag{30}$$

to construct the readout training and test sets. We use separate training and test trials for this task, with the post-washout states and targets concatenated across trials for fitting and evaluation, respectively. We sweep $\tau_{\text{timer}} \in \{5, 10, 15, 20, 25, 30\}$ and report corr$^2(y, \hat{y})$ on the aggregated held-out test set. For Fig. 2c, the plotted prediction for each delay is taken from the test trial, indicating progressively degraded recall as the required delay increases.

Task	Random input NMSE	Sine-product input NMSE
NARMA2	9.2139×10^{-2}	2.0387×10^{-2}
NARMA5	2.0239×10^{-1}	9.2072×10^{-3}
NARMA10	4.9532×10^{-1}	4.6332×10^{-2}
NARMA15	5.5707×10^{-1}	1.9956×10^{-1}
NARMA20	4.8804×10^{-1}	1.4198×10^{-1}

TABLE I: Test NMSE on $\mathcal{K}_{\rm test} = \{4000, \ldots, 4999\}$ for the multitask NARMA benchmark under the two input types in Fig. 2.

2) NARMA multitask benchmark: We evaluate nonlinear temporal processing using a multitask NARMA benchmark [52], [53]. A single scalar input sequence $\{u_k\}_{k=0}^{T-1}$ with $u_k \in [0, 0.2]$ drives the reservoir, and a vector readout predicts five targets simultaneously,

$$\hat{\mathbf{y}}_k = W^\top x_k + \mathbf{b}, \qquad \mathbf{y}_k = \big(y_k^{(2)}, y_k^{(5)}, y_k^{(10)}, y_k^{(15)}, y_k^{(20)}\big)^\top. \tag{31}$$

a) Input generation.: We consider two input types (Fig. 2): *(i) random input* drawn i.i.d. from a continuous uniform distribution

$$u_k \overset{\text{i.i.d.}}{\sim} \mathrm{Uniform}(0, 0.2), \tag{32}$$

and *(ii) sine-product input*

$$s_k = \sin\!\left(\frac{2\pi\alpha k}{T_0}\right) \sin\!\left(\frac{2\pi\beta k}{T_0}\right) \sin\!\left(\frac{2\pi\gamma k}{T_0}\right), \tag{33}$$

$$u_k = 0.1\,(s_k + 1), \qquad u_k \in [0, 0.2], \tag{34}$$

with constants

$$\alpha = 2.11, \qquad \beta = 3.73, \qquad \gamma = 4.11, \qquad T_0 = 100. \tag{35}$$

b) Target generation.: Targets are generated from the same input stream $\{u_k\}$. For NARMA2 we use

$$y_{k+1}^{(2)} = 0.4\, y_k^{(2)} + 0.4\, y_k^{(2)}\, y_{k-1}^{(2)} + 0.6\, u_k^3 + 0.1, \tag{36}$$

and for NARMAn with $n \in \{5, 10, 15, 20\}$ we use

$$y_{k+1}^{(n)} = 0.3\, y_k^{(n)} + 0.05\, y_k^{(n)} \sum_{j=0}^{n-1} y_{k-j}^{(n)} + 1.5\, u_{k-n+1}\, u_k + 0.1, \tag{37}$$

with zero initialization (and out-of-range indices treated as zero, matching the implementation).

c) Train/test split and metric.: Each run uses $T = 5000$ steps. We discard an initial prefix of 1000 steps and split the remainder as

$$\mathcal{K}_{\rm train} = \{1000, \ldots, 3999\}, \qquad \mathcal{K}_{\rm test} = \{4000, \ldots, 4999\}. \tag{38}$$

The readout is trained on $\mathcal{K}_{\rm train}$ and performance is reported as NMSE on $\mathcal{K}_{\rm test}$ for each n.

d) Results.: Table I summarizes the test NMSEs for the two input types.

3) Toy homogeneous SU(2) Yang–Mills one-step benchmark:

a) From Yang–Mills to a finite-dimensional toy model [54]–[56].: We start from the SU(2) Yang–Mills action in $(3+1)$ dimensions,

$$S_{\rm YM} = -\frac{1}{4}\int d^4x\, F^a_{\mu\nu} F^{a\,\mu\nu},$$
$$F^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g\,\epsilon^{abc} A^b_\mu A^c_\nu, \tag{39}$$

where $a = 1, 2, 3$ is the adjoint (color) index, $\mu, \nu = 0, 1, 2, 3$, and ϵ^{abc} are the SU(2) structure constants.

We impose *temporal gauge* $A_0^a = 0$ and a *homogeneous diagonal ansatz*

$$A_i^a(t) = \delta_i^a\, x_i(t), \qquad i, a \in \{1, 2, 3\}, \tag{40}$$

so all spatial derivatives vanish ($\partial_i A^a_\mu = 0$) and the gauge field is fully characterized by three real functions $x_i(t)$.

Under (40), the "electric" components reduce to

$$F^a_{0i} = \dot{A}^a_i = \delta^a_i\, \dot{x}_i, \tag{41}$$

and the spatial components become purely non-Abelian,

$$F^a_{ij} = g\,\epsilon^{abc} A^b_i A^c_j = g\,\epsilon^{aij}\, x_i x_j, \tag{42}$$

so that the only nonzero independent entries are $F^3_{12} = g\, x_1 x_2$, $F^1_{23} = g\, x_2 x_3$, and $F^2_{31} = g\, x_3 x_1$.

Substituting (41)–(42) into the Yang–Mills Lagrangian density and integrating over space (equivalently, working per unit volume), we obtain an effective mechanical Lagrangian

$$L_{\rm eff} = \frac{1}{2}\sum_{i=1}^{3} \dot{x}_i^2 - \frac{1}{2} g^2 \Big(x_1^2 x_2^2 + x_2^2 x_3^2 + x_3^2 x_1^2\Big). \tag{43}$$

The Euler–Lagrange equations then give

$$\ddot{x}_1 = -g^2\, x_1 \big(x_2^2 + x_3^2\big),$$
$$\ddot{x}_2 = -g^2\, x_2 \big(x_3^2 + x_1^2\big),$$
$$\ddot{x}_3 = -g^2\, x_3 \big(x_1^2 + x_2^2\big), \tag{44}$$

which is the toy homogeneous SU(2) Yang–Mills system used in this benchmark. For constant g, the corresponding conserved energy is

$$E = \frac{1}{2}\sum_{i=1}^{3} v_i^2 + \frac{1}{2} g^2 \Big(x_1^2 x_2^2 + x_2^2 x_3^2 + x_3^2 x_1^2\Big), \qquad v_i = \dot{x}_i. \tag{45}$$

In our numerical experiments we optionally employ a coupling quench $g \to g(t)$ (piecewise constant), which acts as an external drive and therefore need not conserve (45).

b) State definition and data generation.: We define the 6D state

$$z(t) = \big(x_1(t), x_2(t), x_3(t), v_1(t), v_2(t), v_3(t)\big)^\top \in \mathbb{R}^6, \tag{46}$$

and generate a discrete trajectory $\{z_n\}_{n=0}^{T_{\rm macro}}$ by integrating (44) with step size Δt (RK4 in the implementation).

c) Coupling protocol.: To introduce non-autonomous dynamics we use a piecewise quench,

$$g(t) = \begin{cases} g_0, & t < t_q, \\ g_1, & t \geq t_q, \end{cases} \tag{47}$$

with parameters specified below.

Component	NMSE	corr^2
x_1	4.3832×10^{-2}	0.9730
x_2	1.2412×10^{-1}	0.9484
x_3	6.3085×10^{-2}	0.9616
v_1	8.6543×10^{-2}	0.9343
v_2	3.8249×10^{-2}	0.9645
v_3	2.7901×10^{-2}	0.9780

TABLE II: Toy homogeneous SU(2) Yang–Mills one-step prediction performance on $\mathcal{N}_{\mathrm{test}}$ (teacher forcing).

d) Scalar-input serialization.: Because the reservoir is driven by a scalar input stream, each macro-state z_n is serialized into six successive scalar "ticks",

$$s_{6n+m} = \phi_m(z_{n,m}), \qquad m = 0, \ldots, 5, \tag{48}$$

where ϕ_m applies a per-component scaling and a smooth saturation to enforce the input range $s_k \in [-u_{\max}, u_{\max}]$. Concretely, we use

$$\phi_m(z) = u_{\max} \tanh\left(\frac{\alpha_m z}{u_{\max}}\right), \tag{49}$$

with $\alpha_m > 0$ chosen from the teacher-forced trajectory so that each component approximately spans the available input range (e.g., $\alpha_m = u_{\max}/\max_n |z_{n,m}|$ up to a small numerical offset). The macro feature vector x_n is taken after the sixth tick (i.e., after consuming $s_{6n}, \ldots, s_{6n+5}$).

e) One-step readout.: We learn the one-step map $z_n \mapsto z_{n+1}$ using a ridge-regularized linear readout applied to the reservoir feature vector x_n,

$$\hat{z}_{n+1} = W^\top x_n + b. \tag{50}$$

f) Train/test split and metric.: We use $T_{\mathrm{macro}} = 800$ and $\Delta t = 0.02$, discard the first 200 macro steps, and split

$$\mathcal{N}_{\mathrm{train}} = \{200, \ldots, 599\}, \qquad \mathcal{N}_{\mathrm{test}} = \{600, \ldots, 799\}. \tag{51}$$

Performance is reported on $\mathcal{N}_{\mathrm{test}}$ for each component of z under *one-step* (teacher-forced) prediction, using NMSE and corr^2. Table II summarizes the results for the run reported here.

g) Parameters.: Unless stated otherwise, we use $g_0 = 100$, $g_1 = 10$, $t_q = 5.0$, input range $u_{\max} = 0.2$, and ridge regularization $\lambda = 10^{-2}$.

4) Autonomous prediction: Mackey–Glass and Lorenz-63: We evaluate autonomous (closed-loop) prediction on two chaotic benchmarks: Mackey–Glass [57] (MG) and Lorenz-63 [58] (L63). Both use the same two-phase protocol: train a one-step-ahead readout under teacher forcing, then roll out in closed loop.

a) Teacher forcing and closed-loop rollout.: Given a scalar ground-truth series $\{y_k\}$, teacher forcing uses

$$s_k = y_k, \qquad \text{target: } y_{k+1}, \tag{52}$$

and trains a one-step predictor

$$\hat{y}_{k+1} = w^\top x_k + b, \tag{53}$$

where x_k is the reservoir feature computed from input s_k and the current reservoir state. In the autonomous phase, predictions are fed back as inputs,

$$\hat{y}_{k+1} = w^\top x_k + b, \qquad s_{k+1} = \mathrm{clip}(\hat{y}_{k+1}, 0, 1), \tag{54}$$

and we initialize the rollout for continuity using the last teacher-forced input, $s_0 \leftarrow s_{T_{\mathrm{train}}-1}$.

b) Mackey–Glass (MG).: MG is generated from the standard Mackey–Glass delay-differential equation

$$\frac{dx(t)}{dt} = \alpha \frac{x(t-\tau_{\mathrm{MG}})}{1 + x(t-\tau_{\mathrm{MG}})^\beta} - \gamma x(t), \tag{55}$$

with parameters $(\alpha, \beta, \gamma, \tau_{\mathrm{MG}}) = (0.2, 10, 0.1, 17)$. In our implementation we discretize (55) by Euler integration with step size $\Delta t = 0.1$, subsample by a factor of 10, discard an initial transient of 2000 samples, and then apply min–max scaling to $[0, 1]$. We use $T_{\mathrm{train}} = 10{,}000$ teacher-forcing steps and a closed-loop rollout of $T_{\mathrm{test}} = 2000$ steps. During teacher forcing, the input is perturbed by small uniform noise,

$$s_k = \mathrm{clip}(y_k + \xi_k, 0, 1), \qquad \xi_k \sim \mathrm{Uniform}(-10^{-4}, 10^{-4}). \tag{56}$$

We visualize the delayed-state trajectory (y_k, y_{k+15}) (Fig. 3).

c) Lorenz-63 (L63).: Lorenz-63 is defined by

$$\begin{aligned} \dot{x}(t) &= \sigma\big(y(t) - x(t)\big), \\ \dot{y}(t) &= x(t)\big(\rho - z(t)\big) - y(t), \\ \dot{z}(t) &= x(t)y(t) - \beta z(t), \end{aligned} \tag{57}$$

with $(\sigma, \rho, \beta) = (10, 28, 8/3)$. We integrate (57) with RK4 using step size $\Delta t = 0.01$, discard an initial transient of 2000 steps, and rescale each coordinate independently to $[0, 1]$ by min–max normalization over the retained window. We then form a scalar series by selecting one coordinate (default: the scaled x component) and apply the same protocol as MG with $T_{\mathrm{train}} = 10{,}000$ and $T_{\mathrm{test}} = 2000$. Teacher-forcing noise is added as in (56), predictions are clipped to $[0, 1]$ during rollout, and visualized via the delayed-state trajectory (y_k, y_{k+15}) (Fig. 3).

B. Metrics

We report normalized mean-squared error (NMSE) and the squared Pearson correlation coefficient corr^2.

a) Normalized mean-squared error (NMSE).: For targets $\{y_k\}$ and predictions $\{\hat{y}_k\}$ over an evaluation set $\mathcal{K}$,

$$\mathrm{NMSE} = \frac{\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}(y_k - \hat{y}_k)^2}{\mathrm{Var}_{k\in\mathcal{K}}(y_k)}. \tag{58}$$

b) Correlation-squared.:

$$\mathrm{corr}^2(y, \hat{y}) = \left(\frac{\mathrm{Cov}(y, \hat{y})}{\sqrt{\mathrm{Var}(y)\,\mathrm{Var}(\hat{y})}}\right)^2. \tag{59}$$

C. Hyperparameters and implementation settings

All benchmarks use the same quantum reservoir architecture (a 3×3 array of reservoir blocks) and the same numerical integration and readout training procedures as described above. Unless otherwise stated, we use 9 blocks indexed by $b \in \{1, \ldots, 9\}$ with block-dependent input scaling factors

$$s_b \in \{0.7,\, 1.0,\, 1.3\}, \qquad m_b \in \{0.8,\, 1.0,\, 1.2\}, \tag{60}$$

and the effective input coupling in block b is $\varepsilon_{\rm eff}^{(b)}(k) = \varepsilon_0 + (s_b m_b)\, s_k$. Each block employs V virtual nodes and a truncated Hilbert space of N Fock states.

a) Hyperparameters.: Unless otherwise stated, we use the same reservoir and readout settings across all tasks:

$$\begin{aligned} &N = 10, \quad \eta = 0.45, \quad \mu = 3.99, \quad \tilde{\kappa} = 0.01, \quad \varepsilon_0 = 20.0, \\ &T_{\rm step}^{(\tau)} = 5.0, \quad N_{\rm RK} = 200, \quad \lambda = 10^{-2}, \end{aligned} \tag{61}$$

where $T_{\rm step}^{(\tau)}$ is the reservoir step length in dimensionless time τ and $N_{\rm RK}$ is the number of RK4 substeps per reservoir step (so $\Delta\tau = T_{\rm step}^{(\tau)}/N_{\rm RK}$). The number of virtual nodes per block is $V = 10$ for the timer, NARMA, Mackey–Glass, and the toy Yang–Mills task, and $V = 50$ for Lorenz-63. For the toy Yang–Mills benchmark we use the same settings except $\varepsilon_0 = 0.5$. In all cases, the ridge readout is trained on standardized (zero-mean, unit-variance) reservoir features and targets using training-set statistics.

V. Conclusion

In this work we proposed and numerically validated a minimal trapped-ion–motivated quantum reservoir in which the computational substrate is the vibrational mode of a single ion driven by a Raman-generated traveling-wave optical potential. By encoding a scalar input stream into the effective drive strength and evolving the resulting (optionally dissipative) dynamics under a Lindblad master equation, the system naturally produces a rich, high-dimensional feature map without training the internal dynamics.

Our reservoir implementation combines (i) time multiplexing via virtual nodes sampled within each discrete step and (ii) spatial multiplexing via an ensemble of blocks with varied input scalings, and it uses a compact observable set—including number, quadratures, second moments, and the nonlinear operators $\cos\xi$ and $\sin\xi$—as measured features. A key practical detail is enforcing continuous drive phase across the full input sequence (i.e., not resetting the internal phase between steps), which preserves the intended physics of the monochromatic drive and stabilizes the discrete-time reservoir mapping. A simple ridge-regression readout trained on these features is sufficient to solve standard temporal benchmarks.

Across representative tasks—timer responses, multitask NARMA, homogeneous SU(2) Yang–Mills field equation and autonomous chaotic prediction on Mackey–Glass and Lorenz-63—the driven trapped-ion oscillator acts as an effective reservoir and achieves good performance using only a linear readout, supporting the central claim that a single motional mode can serve as a practical QRC platform. Looking forward, this framework points to an experimentally accessible route to QRC in existing trapped-ion systems: the reservoir knobs (drive strength, detuning ratio μ, effective quantumness η, and dissipation $\tilde{\kappa}$) are directly tunable, and the model naturally connects computational performance to familiar dynamical regimes.

Future work will focus on (i) mapping the required observables to concrete measurement protocols (e.g., internal-state–assisted readout of motional moments and nonlinear functions), (ii) systematically optimizing hyperparameters versus task complexity, and (iii) developing physics-guided diagnostics—such as Floquet and resonance-cell structure—to better predict and control when the reservoir offers the best balance of memory and nonlinearity for real-time learning.

Code availability

The code used to generate the datasets and reproduce all experimental results is publicly available at `https://github.com/GIAYANGGAO/qrc_with_single_trapped_ion`.

Acknowledgments

This research is supported by the National Research Foundation Singapore. The author used ChatGPT (OpenAI) [59] to assist with English grammar and language polishing in parts of this manuscript. The author reviewed and edited all AI-assisted text and takes full responsibility for the final content of the paper.

References

[1] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *science*, vol. 304, no. 5667, pp. 78–80, 2004.

[2] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[3] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer science review*, vol. 3, no. 3, pp. 127–149, 2009.

[4] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, "Next generation reservoir computing," *Nature communications*, vol. 12, no. 1, p. 5564, 2021.

[5] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, 2019.

[6] 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," *Nature communications*, vol. 2, no. 1, p. 468, 2011.

[7] 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," *Physical review letters*, vol. 120, no. 2, p. 024102, 2018.

[8] X. Chen, F. A. Araujo, M. Riou, J. Torrejon, D. Ravelosona, W. Kang, W. Zhao, J. Grollier, and D. Querlioz, "Forecasting the outcome of spintronic experiments with neural ordinary differential equations," *Nature communications*, vol. 13, no. 1, p. 1016, 2022.

[9] F. Abreu Araujo, M. Riou, J. Torrejon, S. Tsunegi, D. Querlioz, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, *et al.*, "Role of non-linear data processing on speech recognition task in the framework of reservoir computing," *Scientific reports*, vol. 10, no. 1, p. 328, 2020.

[10] G. Dion, S. Mejaouri, and J. Sylvestre, "Reservoir computing with a single delay-coupled non-linear mechanical oscillator," *Journal of Applied Physics*, vol. 124, no. 15, 2018.

[11] P. J. Ehlers, H. I. Nurdin, and D. Soh, "Improving the performance of echo state networks through state feedback," *Neural Networks*, vol. 184, p. 107101, 2025.

[12] K. Nakajima, "Physical reservoir computing—an introductory perspective," *Japanese Journal of Applied Physics*, vol. 59, no. 6, p. 060501, 2020.

[13] K. Nakajima, "Reservoir computing: Theory, physical implementations, and applications," *IEICE Technical Report; IEICE Tech. Rep.*, vol. 118, no. 220, pp. 149–154, 2018.

[14] J. Chen, H. I. Nurdin, and N. Yamamoto, "Temporal information processing on noisy quantum computers," *Physical Review Applied*, vol. 14, no. 2, p. 024065, 2020.

[15] K. Fujii and K. Nakajima, "Harnessing disordered-ensemble quantum dynamics for machine learning," *Physical Review Applied*, vol. 8, no. 2, p. 024030, 2017.

[16] J. Dudas, B. Carles, E. Plouet, F. A. Mizrahi, J. Grollier, and D. Marković, "Quantum reservoir computing implementation on coherently coupled quantum oscillators," *npj Quantum Information*, vol. 9, no. 1, p. 64, 2023.

[17] L. Govia, G. Ribeill, G. Rowlands, H. Krovi, and T. Ohki, "Quantum reservoir computing with a single nonlinear oscillator," *Physical Review Research*, vol. 3, no. 1, p. 013077, 2021.

[18] R. A. Bravo, K. Najafi, X. Gao, and S. F. Yelin, "Quantum reservoir computing using arrays of rydberg atoms," *PRX Quantum*, vol. 3, no. 3, p. 030325, 2022.

[19] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, "Dynamical phase transitions in quantum reservoir computing," *Physical Review Letters*, vol. 127, no. 10, p. 100502, 2021.

[20] P. Mujal, R. Martínez-Peña, G. L. Giorgi, M. C. Soriano, and R. Zambrini, "Time-series quantum reservoir computing with weak and projective measurements," *npj Quantum Information*, vol. 9, no. 1, p. 16, 2023.

[21] D. Fry, A. Deshmukh, S. Y.-C. Chen, V. Rastunkov, and V. Markov, "Optimizing quantum noise-induced reservoir computing for nonlinear and chaotic time series prediction," *Scientific Reports*, vol. 13, no. 1, p. 19326, 2023.

[22] W. Kalfus, G. Ribeill, G. Rowlands, H. Krovi, T. Ohki, and L. Govia, "Hilbert space as a computational resource in reservoir computing," *Physical Review Research*, vol. 4, no. 3, p. 033007, 2022.

[23] J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, "Scalable photonic platform for real-time quantum reservoir computing," *Physical Review Applied*, vol. 20, no. 1, p. 014051, 2023.

[24] C. Zhu, P. J. Ehlers, H. I. Nurdin, and D. Soh, "Practical few-atom quantum reservoir computing," *Physical Review Research*, vol. 7, no. 2, p. 023290, 2025.

[25] J. Chen and H. I. Nurdin, "Correction to: Learning nonlinear input–output maps with dissipative quantum systems," *Quantum Information Processing*, vol. 18, no. 11, p. 354, 2019.

[26] T. Hülser, F. Köster, K. Lüdge, and L. Jaurigue, "Deriving task specific performance from the information processing capacity of a reservoir computer," *Nanophotonics*, vol. 12, no. 5, pp. 937–947, 2023.

[27] J. Nokkala, R. Martínez-Peña, G. L. Giorgi, V. Parigi, M. C. Soriano, and R. Zambrini, "Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing," *Communications Physics*, vol. 4, no. 1, p. 53, 2021.

[28] W. Xia, J. Zou, X. Qiu, and X. Li, "The reservoir learning power across quantum many-body localization transition," *Frontiers of Physics*, vol. 17, no. 3, p. 33506, 2022.

[29] T. Yasuda, Y. Suzuki, T. Kubota, K. Nakajima, Q. Gao, W. Zhang, S. Shimono, H. I. Nurdin, and N. Yamamoto, "Quantum reservoir computing with repeated measurements on superconducting devices," *arXiv preprint arXiv:2310.06706*, 2023.

[30] D. Marković, N. Leroux, M. Riou, F. Abreu Araujo, J. Torrejon, D. Querlioz, A. Fukushima, S. Yuasa, J. Trastoy, P. Bortolotti, *et al.*, "Reservoir computing with the frequency, phase, and amplitude of spin-torque nano-oscillators," *Applied Physics Letters*, vol. 114, no. 1, 2019.

[31] G. Angelatos, S. A. Khan, and H. E. Türeci, "Reservoir computing approach to quantum state measurement," *Physical Review X*, vol. 11, no. 4, p. 041062, 2021.

[32] S. Ghosh, A. Opala, M. Matuszewski, T. Paterek, and T. C. Liew, "Quantum reservoir processing," *npj Quantum Information*, vol. 5, no. 1, p. 35, 2019.

[33] S. Ghosh, T. Krisnanda, T. Paterek, and T. C. Liew, "Realising and compressing quantum circuits with quantum reservoir computing," *Communications Physics*, vol. 4, no. 1, p. 105, 2021.

[34] L. Domingo Colomer, G. Carlo, F. Borondo, *et al.*, "Optimal quantum reservoir computing for the noisy intermediate-scale quantum era," *Physical Review E-Statistical, Nonlinear, Biological, and Soft Matter Physics*, 2022.

[35] K. Kobayashi, K. Fujii, and N. Yamamoto, "Feedback-driven quantum reservoir computing for time-series analysis," *PRX quantum*, vol. 5, no. 4, p. 040325, 2024.

[36] O. Ahmed, F. Tennie, and L. Magri, "Optimal training of finitely sampled quantum reservoir computers for forecasting of chaotic dynamics," *Quantum Machine Intelligence*, vol. 7, no. 1, pp. 1–16, 2025.

[37] S. A. Khan, F. Hu, G. Angelatos, and H. E. Türeci, "Physical reservoir computing using finitely-sampled quantum systems," *arXiv preprint arXiv:2110.13849*, 2021.

[38] A. Palacios, R. Martínez-Peña, M. C. Soriano, G. L. Giorgi, and R. Zambrini, "Role of coherence in many-body quantum reservoir computing," *Communications Physics*, vol. 7, no. 1, p. 369, 2024.

[39] C. Zhu, P. J. Ehlers, H. I. Nurdin, and D. Soh, "Minimalistic and scalable quantum reservoir computing enhanced with feedback," *npj Quantum Information*, 2025.

[40] S. Y.-C. Chen, D. Fry, A. Deshmukh, V. Rastunkov, and C. Stefanski, "Reservoir computing via quantum recurrent neural networks," *arXiv preprint arXiv:2211.02612*, 2022.

[41] S. Y.-C. Chen, "Efficient quantum recurrent reinforcement learning via quantum reservoir computing," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13186–13190, IEEE, 2024.

[42] K. Kobayashi and Y. Motome, "Edge of many-body quantum chaos in quantum reservoir computing," *arXiv preprint arXiv:2506.17547*, 2025.

[43] A. Sakurai, M. P. Estarellas, W. J. Munro, and K. Nemoto, "Quantum extreme reservoir computation utilizing scale-free networks," *Physical Review Applied*, vol. 17, no. 6, p. 064044, 2022.

[44] G. Berman, D. James, R. Hughes, M. Gulley, M. Holzscheiter, and G. López, "Dynamical stability and quantum chaos of ions in a linear trap," *Physical Review A*, vol. 61, no. 2, p. 023403, 2000.

[45] G. Berman, A. Bishop, D. James, R. Hughes, and D. Kamenev, "Dynamical stability of an ion in a linear trap as a solid-state problem of electron localization," *Physical Review A*, vol. 64, no. 5, p. 053406, 2001.

[46] K. Nakajima, K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa, "Boosting computational power through spatial multiplexing in quantum reservoir computing," *Physical Review Applied*, vol. 11, no. 3, p. 034021, 2019.

[47] B. Shore, "The theory of coherent atomic excitation," 1991.

[48] H. Carmichael, *An open systems approach to quantum optics: lectures presented at the Université Libre de Bruxelles October 28 to November 4, 1991*. Springer, 1993.

[49] J. R. Johansson, P. D. Nation, and F. Nori, "Qutip: An open-source python framework for the dynamics of open quantum systems," *Computer physics communications*, vol. 183, no. 8, pp. 1760–1772, 2012.

[50] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method.," *Sov Dok*, vol. 4, pp. 1035–1038, 1963.

[51] C. Wringe, M. Trefzer, and S. Stepney, "Reservoir computing benchmarks: a review, a taxonomy, some best practices," *arXiv preprint arXiv:2405.06561*, 2024.

[52] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: unifying the algorithms and accelerating convergence," *IEEE transactions on neural networks*, vol. 11, no. 3, pp. 697–709, 2000.

[53] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural networks*, vol. 20, no. 3, pp. 391–403, 2007.

[54] T. Akutagawa, K. Hashimoto, T. Sasaki, and R. Watanabe, "Out-of-time-order correlator in coupled harmonic oscillators," *Journal of High Energy Physics*, vol. 2020, no. 8, pp. 1–27, 2020.

[55] S. Matinyan, G. Savvidi, and N. Ter-Arutyunyan-Savvidi, "Stochasticity of classical yang-mills mechanics and its elimination by using the higgs mechanism," *JETP Lett.(Engl. Transl.);(United States)*, vol. 34, no. 11, 1981.

[56] S. Guo, "Operator growth in su (2) yang-mills theory," *arXiv preprint arXiv:2208.13362*, 2022.

[57] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[58] E. N. Lorenz, "Deterministic nonperiodic flow 1," in *Universality in Chaos, 2nd edition*, pp. 367–378, Routledge, 2017.

[59] OpenAI, "Introducing ChatGPT." https://openai.com/index/chatgpt/, 2022. Published: 2022-11-30.