Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Enhanced Nonlinear Capacity of a Trapped-Ion Quantum Memristor Reservoir

Anna Frolova

frolova.as17@physics.msu.ru

Lomonosov Moscow State University

Sergey Stremoukhov

Lomonosov Moscow State University

Pavel Forsh

Lomonosov Moscow State University

Ksenia Khabarova

P.N. Lebedev Physical Institute of the Russian Academy of Sciences

Nikolay Kolachevsky

P.N. Lebedev Physical Institute of the Russian Academy of Sciences

Article

Keywords: Quantum Memristor, Quantum computing technology, Quantum machine learning, Quantum Reservoir Computing

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11404134/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Enhanced Nonlinear Capacity of a Trapped-Ion Quantum Memristor Reservoir

Anna Frolova[1,2*], Sergey Stremoukhov[1,2], Pavel Forsh[1,2], Ksenia Khabarova[2], Nikolay Kolachevsky[2,3]

[1*]Faculty of Physics, Lomonosov Moscow State University, Leninskie Gory, 1/2, Moscow, 119991, Russia.
[2]P.N. Lebedev Physical Institute, Russian Academy of Science, Leninskiy Prospect, 53, Moscow, 119991, Russia.
[3]Russian Quantum Center, , Skolkovo, Moscow, 143025, Russia.

*Corresponding author(s). E-mail(s): frolova.as17@physics.msu.ru;

Abstract

We present a numerical study of a quantum memristor (QM) reservoir based on a trapped-ion platform. The reservoir exploits nonlinear input–output mapping arising from Born-rule nonlinearity, quantum coherences, and feedback-induced fading memory to process classical time-series data.

We benchmarked the performance of the QM reservoir on monomial approximation, NARMA, Mackey–Glass, and the Santa Fe time series, standard tasks in reservoir computing and quantum reservoir computing (QRC). The QM reservoir shows a clear memory–nonlinearity trade-off. Its nonlinear capacity scales favorably: two coupled QMs reproduce monomials $\boldsymbol{x^d}$ up to degree $\boldsymbol{d = 100}$ with test MSE below $\mathbf{10^{-2}}$, while classical echo state networks (ESNs) fail beyond $\boldsymbol{d \approx 40}$ and show no improvement with increasing size up to **2000** nodes. However, for memory-dominated tasks such as NARMA10, the same two-QM extension provides only marginal gains; the insensitivity of the test MSE to the memory decay parameter confirms that, in the present architecture, nonlinear expressivity rather than dynamical memory is the main computational resource. On standard time-series benchmarks, the QM reservoir matches the accuracy of small ESNs while using a comparable number of readout weights, and it remains robust against gate noise.

Our results establish trapped-ion QMs as a competitive QRC platform with particular advantages in tasks dominated by strong nonlinearity.

Keywords: Quantum Memristor, Quantum computing technology, Quantum machine learning, Quantum Reservoir Computing

1 Introduction

Reservoir computing (RC) is a powerful paradigm for processing temporal information, where a high-dimensional dynamical system — the reservoir — nonlinearly transforms input sequences, and only a simple linear readout layer is trained to produce the desired output. Echo state networks (ESNs) [1] are the most established classical RC model; they have been successfully applied to chaotic time-series prediction, speech recognition, and nonlinear control, often outperforming fully trained recurrent neural networks in low-data regimes. The key features of a reservoir are high dimensionality, nonlinearity, and fading memory.

Quantum reservoir computing (QRC) has emerged as a natural extension of these ideas, leveraging the high-dimensional Hilbert space and intrinsic dynamics of quantum systems to process time-dependent data with reduced training overhead and potential energy savings [2–4]. In QRC, dissipation and measurement-induced nonlinearities are not merely tolerated but actively exploited as computational resources: dissipation provides fading memory, while controlled noise and feedback enhance nonlinear processing [5, 6]. Various physical platforms have been explored as quantum reservoirs, including spin networks, bosonic and oscillator systems, Rydberg atoms, and cavity-QED or few-atom architectures, often demonstrating competitive or superior performance to classical ESNs in tasks such as nonlinear and chaotic time-series prediction [3, 4, 7, 8]. Recent work has also emphasized minimal, few-site reservoirs augmented with feedback and polynomial readout, showing that very small quantum systems can already match or exceed classical reservoirs in memory and nonlinear capacity [3, 4, 8].

A particularly natural component for QRC is a quantum memristor (QM), which combines essential reservoir ingredients: a high-dimensional Hilbert space, nonlinear input–output mapping arising from Born-rule nonlinearity, and memory from feedback-controlled dissipation [9]. In recent years, QMs have been proposed for various physical platforms, including superconducting circuits with tunable oscillator schemes [9–12], quantum photonics with tunable beam splitter schemes [13–15], ion traps [16, 17], light-matter systems [18, 19], and digital quantum simulation [20–22]. Among these, trapped ions offer a particularly attractive platform due to their long coherence times, high-fidelity control, and flexible level structure, making them natural candidates for implementing quantum synaptic elements and scalable reservoirs. Moreover, the rich energy-level structure of a single ion allows multiple QMs to be defined and coupled on the same ion, significantly reducing hardware overhead [17]. QRC has already been applied to standard benchmarks such as NARMA, Mackey–Glass, and real-world financial time series, providing a common basis for comparing quantum and classical reservoirs [4, 23, 24].

In this work, we extend the proposal of Ref. [16] to a full reservoir-computing framework and conduct a numerical study of a QM reservoir based on a trapped-ion platform for classical time-series processing. Building on the idea of using measurement-controlled dissipation and feedback as computational resources, we investigate how feedback-controlled memory and quantum coherences in trapped-ion QM architectures translate into nonlinear capacity and temporal processing power. We benchmarked the

QM reservoir on monomial approximation, NARMA2 and NARMA10 tasks, Mackey–Glass, and Santa Fe time series and compare its performance with classical ESNs. By analyzing both accuracy and resource requirements, we position trapped-ion QMs as a competitive QRC platform with distinctive advantages in strongly nonlinear regimes.

2 Model

We consider a QM reservoir based on a single trapped ion with three relevant energy levels: $|g\rangle$, $|e\rangle$, and $|a\rangle$. The ion is driven by two resonant pulses, as proposed in Ref. [16] with an additional restriction for the pulses to be time-separated. The first pulse realizes a rotation between $|g\rangle$ and $|e\rangle$ with angle θ (gate U_1). The second pulse couples $|e\rangle$ and $|a\rangle$ with an angle $\phi = 2\arccos\sqrt{s}$ that depends on the state parameter $s \in [0,1]$ of the QM (gate U_2).

In the basis $\{|g\rangle, |e\rangle, |a\rangle\}$ the gates have the explicit matrix form

$$U_1(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} & 0 \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad U_2(s) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\frac{\phi(s)}{2} & -i\sin\frac{\phi(s)}{2} \\ 0 & -i\sin\frac{\phi(s)}{2} & \cos\frac{\phi(s)}{2} \end{pmatrix}, \tag{1}$$

The state parameter s is updated according to a feedback rule that involves the population of the auxiliary level $P_{|a\rangle} = |\langle a|\psi\rangle|^2$ measured immediately after U_2 is applied. In our numerical simulations, we use the exact quantum-mechanical probability $P_{|a\rangle}$; in an experimental implementation, it would need to be estimated by repeated projective measurements, which does not alter the architecture.

Two feedback rules are considered: a sliding-window average and an exponential moving average. For the sliding-window rule the update reads

$$s_{t+1} = \frac{1}{w}\sum_{k=0}^{w-1}\big(aP_{|a\rangle,t-k} + b\big), \tag{2}$$

where w is the window length, and a, b are optimisable parameters. For the exponential rule we use

$$s_{t+1} = s_t + \frac{1}{m}\big(P_{|a\rangle,t} - s_t\big), \tag{3}$$

with a memory decay parameter $m \geq 1$ (the limit $m \to \infty$ corresponds to a fixed state parameter).

Throughout the paper, "ion reservoir" denotes the three-level ion subject to the unitary gates U_1, U_2 without the feedback update and with fixed $s = 0.5$, while "QM" refers to the full system with active feedback loop.

2.1 Input encoding and feature extraction

A scalar input x_t is first mapped to a pure state in the $\{|g\rangle, |e\rangle\}$ subspace,

$$|\psi_{\text{in}}\rangle = \sqrt{f(x_t)}|g\rangle + \sqrt{1 - f(x_t)}|e\rangle, \tag{4}$$

where $f(x_t)$ is task-dependent, specified for each benchmark task below.

The encoded state then undergoes a sequence of unitary transformations:

1. **Input unitary (optional).** An optimisable random unitary U_{in} mixes the populations.
2. **Internal dynamics.** The state passes through $U_2(s_t)\,U_1(\theta)$, after which P_a is measured and s_t is updated according to Eqs. (2) or (3).
3. **Output unitary (optional).** A second optimisable random unitary U_{out} is applied, resulting in the output state $|\psi_{\text{out}}\rangle$.
4. **Measurement.** The probabilities $P_i = |\langle i|\psi_{\text{out}}\rangle|^2$ ($i = |g\rangle, |e\rangle, |a\rangle$) are obtained by a projective measurement in the computational basis. Optionally, a fixed rotation U_{meas}, which we take as a $\pi/2$ pulse in the $\{|g\rangle, |e\rangle\}$ subspace, can be inserted before the measurement to mix the populations and enrich the readout features. Furthermore, one may extract not only the probabilities but also the coherences, i.e. the real and imaginary parts of the off-diagonal density matrix elements $\rho_{ij} = \langle i|\psi_{\text{out}}\rangle\langle\psi_{\text{out}}|j\rangle$ for $i \neq j$. The feature vector $\mathbf{f}_t$ is then either the three probabilities alone or their combination with the six coherences.

The complete transformation of the input x_t into the feature vector $\mathbf{f}_t$ is therefore

$$\mathbf{f}_t = \mathcal{M}\big(U_{\text{out}}\,U_2(s_t)\,U_1(\theta)\,U_{\text{in}}|\psi_{\text{in}}\rangle\big), \tag{5}$$

with $\mathcal{M}$ denoting the measurement procedure described above.

2.2 Single-QM reservoir

The single-QM reservoir follows the steps outlined above. This basic building block provides all the essential ingredients: parameterized unitary transformations $U_{\text{in}}, U_{\text{out}}$, feedback-controlled memory, and the possibility to enhance the nonlinear mapping through U_{meas} and coherences.

2.3 Processing depth options

We consider the following possible extensions of a single-QM reservoir:

- n_{upd} **(memory updates).** The internal cycle U_2U_1 is repeated n_{upd} times for one given input, with the state parameter s updated after each repetition.
- n_{chip} **(chip runs).** The entire sequence (encoding, U_{in}, internal dynamics, U_{out}, measurement) is repeated n_{chip} times, mimicking multiple passes through the physical chip. The memory is preserved between runs, and the feature vectors from all runs are concatenated.
- **Previous-step features.** The feature vector for the current data point can be concatenated with the feature vector from the previous data point. This increases the dimensionality of the feature vector and may improve the prediction accuracy.

These parameters can be tuned to balance nonlinear expressivity against the size of the feature vector. In this work, we only tested the minimal extensions.

2.4 Multi-QM architectures

The reservoir can be expanded to several QMs that process the same input in parallel, sequentially, or in a coupled fashion.

Parallel.

Two independent single-QM channels receive the same encoded state and evolve with their own unitaries $U_{\text{in}}^{(i)}, U_{\text{out}}^{(i)}$, angles θ_i, and state parameters s_i. The two output states are measured independently, giving 2×3 (or 2×9 with coherences) features.

Sequential.

The output state of the first QM is fed as the input to the second QM. Only the final state after the second QM is measured, but both $P_{|a\rangle,1}$ and $P_{|a\rangle,2}$ are used to update the respective state parameters.

Coupled.

Two QMs are embedded in a five-level ion ($|g\rangle, |e_1\rangle, |a_1\rangle, |e_2\rangle, |a_2\rangle$) as described in Ref. [17]. The first pair of gates couples $|g\rangle$–$|e_1\rangle$ and $|e_1\rangle$–$|a_1\rangle$, yielding $P_{|a_1\rangle}$. The second pair couples $|g\rangle$–$|e_2\rangle$ and $|e_2\rangle$–$|a_2\rangle$, yielding $P_{|a_2\rangle}$. The feature vector is obtained from the final state after all four gates. In this architecture, U_{meas} acts as a $\pi/2$ pulse in the $\{|g\rangle, |e_1\rangle\}$ subspace.

2.5 Readout and training

The QM reservoir optimisable parameters are selected by random search. The gate angle θ is sampled uniformly from $[0, 2\pi]$. The feedback constants are sampled as $a \in [0.1, 2.0]$ and $b \in [0.0, 0.5]$, the sliding-window length w is chosen uniformly from the set $\{1, 3, \ldots, 19\}$, and the exponential memory parameter is sampled as $m \in [1.01, 50]$. If used, the input/output unitaries $U_{\text{in}}, U_{\text{out}}$ are generated as random unitary matrices according to the Haar measure, which is the unique uniform probability measure over the unitary group and ensures isotropic sampling of unitary transformations.

For each candidate reservoir, the full training sequence of inputs $\mathbf{x}_{\text{train}}$ of length T_{train} is processed to construct the feature matrix $\mathbf{F}$, and the linear readout weights $\mathbf{w}$ and bias b are obtained by ordinary least-squares regression on the training set of outputs $\mathbf{y}_{\text{train}}$:

$$\hat{\mathbf{y}}_{\text{train}} = \mathbf{F}\mathbf{w} + b\,\mathbf{1}, \qquad \mathbf{F} = \begin{pmatrix} \mathbf{f}_1^\top \\ \mathbf{f}_2^\top \\ \vdots \\ \mathbf{f}_{T_{\text{train}}}^\top \end{pmatrix}, \qquad \mathbf{y}_{\text{train}} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{T_{\text{train}}} \end{pmatrix}, \tag{6}$$

where $\hat{\mathbf{y}}_{\text{train}}$ is the reservoir's prediction on the training set and $\mathbf{1}$ is the vector of ones.

Then the candidate with trained readout weights is evaluated on a test set $\mathbf{y}_{\text{test}}$ of length T_{test} with its quality quantified by the mean squared error (MSE)

$$\text{MSE} = \frac{1}{T_{\text{test}}} \sum_{t=1}^{T_{\text{test}}} (y_t - \hat{y}_t)^2 , \tag{7}$$

where $\hat{y}_t$ indicate the reservoir predictions.

We evaluate $n_{\text{trials}} = 30$–50 random reservoir candidates, and additionally, if $U_{\text{in}}, U_{\text{out}}$ are used, we test $n_{\text{pairs}} = 10$ random unitary pairs per trial. The candidate with the lowest test MSE, along with its trained readout weights, is retained.

3 Results

In all experiments, the data are fed sequentially: for monomial tasks, the input x is swept from 0 to 1 in increasing order, while for time-series tasks, the data points are processed in their natural temporal order. This ensures that the dynamical memory of the reservoir evolves continuously and that the feedback mechanism can capture temporal correlations.

For each benchmark task, we specify the encoding function, the train/test split, and the feedback rule used.

3.1 Monomial approximation

Task and encoding.

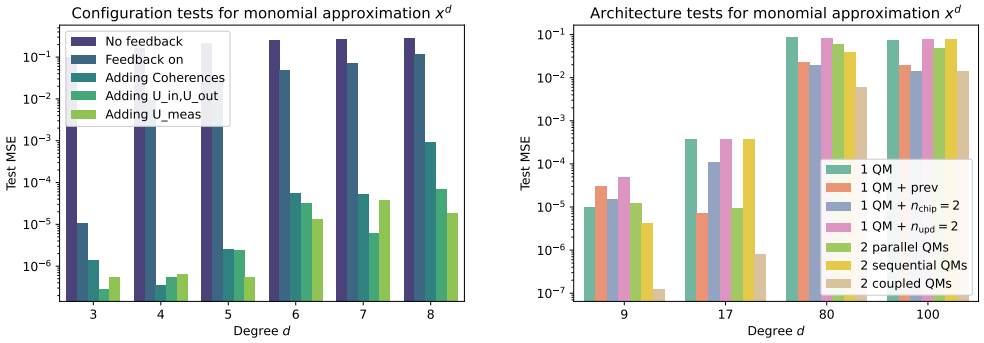
The reservoir receives a scalar input $x \in [0, 1]$ and must reconstruct the target $y = x^d$ for a given integer degree d. We use the linear encoding $f(x) = x$, i.e. $|\psi_{\text{in}}\rangle = \sqrt{x}|g\rangle + \sqrt{1-x}|e\rangle$ (cf. Eq. (4)). This choice keeps the encoding free of additional nonlinearity and allows us to directly probe the nonlinear capacity of the reservoir itself. The data set consists of 200 equally spaced points $x \in [0, 1]$; the first 90% of the data ($0 \leq x \leq 0.9$) are used for training and the remaining 10% ($x > 0.9$) for testing. Feedback is implemented with the sliding-window rule.

Nonlinearity sources.

We investigated the contribution of various nonlinearity sources in a cumulative fashion for a single QM for degrees $d = 3, \ldots, 8$ (Fig. 1a). An ion reservoir with no feedback or additional unitary transformations essentially produces a linear fit. Sliding-window feedback alone adds some nonlinear capacity, but it's not enough for $d > 3$. The inclusion of quantum coherences in the feature vector produces a dramatic drop of the MSE by several orders of magnitude for degrees 4 and above. The importance of additional unitary transformations $U_{\text{in}}, U_{\text{out}}$ for nonlinear capacity is evident for higher degrees, 6 and above, while including a fixed measurement rotation U_{meas} may provide a marginal accuracy improvement. An example of the feedback advantage is shown in Fig. 2a, where the configuration with active feedback closely follows the target x^4 curve, while the same reservoir without feedback substantially deviates from it.

Architecture tests.

To identify the most powerful reservoir configuration for the monomial task, we tested minimal processing-depth extensions and two-QM architectures, described in Secs. 2.3 and 2.4, focusing on higher degrees $d = 9, 17, 80, 100$ (Fig. 1b). We took the single QM with coherences and additional unitary transformations $U_{\text{in}}, U_{\text{out}}$, U_{meas} as the baseline, denoted as "1 QM" in the legend. Among the tested options, two coupled QMs provide consistent accuracy improvements across all degrees compared to the baseline, while additional processing depth options become relevant with the highest degrees.



(a) (b)

Fig. 1: (a) Configuration tests for monomial $y = x^d$ task. MSE gains demonstrated for including feedback, quantum coherences, unitary transformations $U_{\text{in}}, U_{\text{out}}$ and U_{meas} in a cumulative fashion with a single ion reservoir as baseline. (b) Performance of different reservoir architectures. The various processing depth options (concatenating features from the previous data point ("prev"), two chip runs ($n_{\text{chip}} = 2$) and two QM updates per data point ($n_{\text{upd}} = 2$)) and two-QM architectures are tested against the single QM performance with coherences, and $U_{\text{in}}, U_{\text{out}}, U_{\text{meas}}$ unitary transformations.

Comparison with classical models and a photonic QM.

We studied the scaling of the test MSE with the monomial degree d, comparing QM and ESN reservoirs (Fig. 2c). For each degree we also determined the minimal ESN size $N_{\text{min}} \leq 2000$ needed to match the QM accuracy (within a 10% margin). The comparison was conducted for two coupled QMs with coherences and unitary transformations $U_{\text{in}}, U_{\text{out}}$ and U_{meas}. This QM reservoir uses 25 readout weights, and 108 other optimisable parameters: U_1 gate angles θ_1, θ_2 for the two coupled QMs, feedback constants a_1, a_2, b_1, b_2, w_1, w_2 and two 5×5 complex unitaries $U_{\text{in}}, U_{\text{out}}$ with 50 real parameters each. The ESN optimizes 3 hyperparameters (spectral radius ρ, leaking rate α and an input scaling parameter) and trains N_{min} readout weights.

For low degrees ($d \lesssim 24$), classical ESNs achieve the QM performance with relatively small reservoirs $N_{\text{min}} = 6$–30. As d increases, the required ESN size grows

rapidly, $N_{\min}$ reaching 100 already at $d = 29$. For $d > 40$, none of the tested ESNs (up to $N = 2000$) can reproduce the QM accuracy. While for the QM reservoir, the performance for $d \gtrsim 50$ can be improved with a minimal extension to two chip runs ($n_{\text{chip}} = 2$), allowing its test MSE to stay below 10^{-2} up to $d = 100$ with only increasing the amount of readout weights to 50. An example of ESN and QM performances for $d = 80$ is shown in Fig. 2b. This demonstrates a practically unbounded nonlinear capacity of the trapped-ion QM reservoir.

For the specific case $d = 4$, our best result for a single QM is achieved with only feedback and coherences (without unitaries), yielding MSE $= 3.47 \times 10^{-7}$ (Fig. 2a). For comparison, the photonic QM of Ref. [25] reports an MSE of 7.1×10^{-4} for the same task. The more than three-order-of-magnitude improvement can be attributed to the higher dimensionality of our three-level system compared with the two-level photonic realization. This is consistent with the further enhancement observed when moving to five-level coupled QMs, highlighting the role of the Hilbert-space dimension in increasing nonlinear capacity.

3.2 NARMA and time-series benchmarks

Tasks and encoding.

We evaluate the reservoir on the NARMA family of nonlinear input–output mappings, specifically NARMA2, NARMA10, and a stabilised variant, denoted "NARMA10-st", as well as on one-step-ahead prediction of the Mackey–Glass and Santa Fe time series.

For NARMA2, following Ref. [25], the input is a random sequence $x_t \sim U[0, 0.5]$ and the target y_t is generated by

$$y_t = 0.4\, y_{t-1} + 0.4\, y_{t-1} y_{t-2} + 0.6\, x_{t-1}^3 + 0.1. \tag{8}$$

For the standard NARMA10 benchmark [26], the target is

$$y_t = 0.3\, y_{t-1} + 0.05\, y_{t-1} \sum_{k=0}^{9} y_{t-1-k} + 1.5\, x_{t-1} x_{t-10} + 0.1. \tag{9}$$

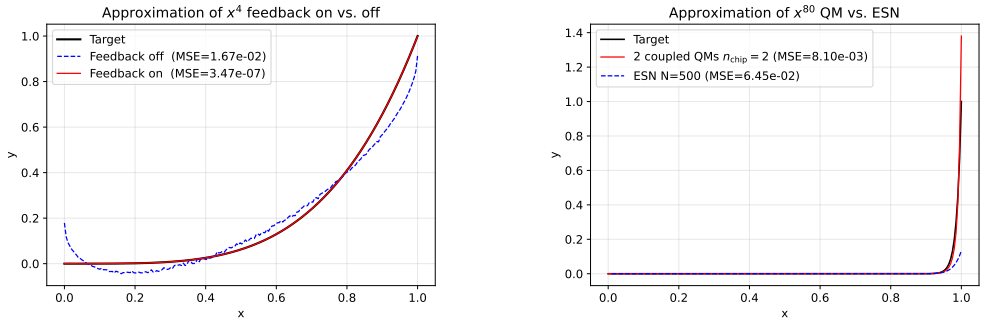
The stabilised NARMA10-st task was proposed in Ref. [27], it avoids the divergence of the original NARMA10 and is highly stable and robust against the choice of input range and initial conditions.

For NARMA tasks, we use the quadratic encoding $f(x) = x^2$, i.e. $|\psi_{\text{in}}\rangle = x_t|g\rangle + \sqrt{1 - x_t^2}|e\rangle$ (cf. Eq. (4)). Among the tested power-law encodings $f(x) = x^p$ with $p = 1, 2, 3, 4$, this choice gave the best test MSE in preliminary experiments.

For Mackey–Glass and Santa Fe time-series tasks, the input is the current value y_t and the target output is y_{t+1}, for encoding we use $f(y) = y_t$, i.e. $|\psi_{\text{in}}\rangle = \sqrt{y_t}|g\rangle + \sqrt{1 - y_t}|e\rangle$ (cf. Eq. (4)), which was found to be optimal.

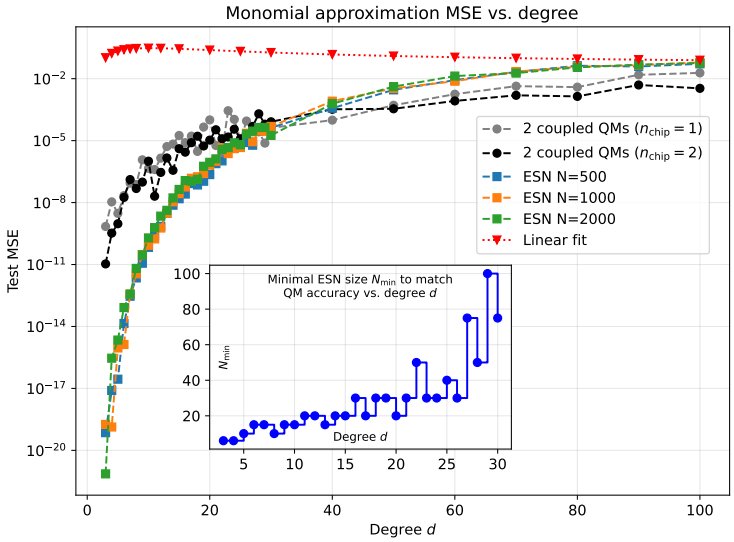
The Mackey–Glass time series is generated by the delay-differential equation

$$\frac{dy}{dt} = \frac{0.2\, y(t - \tau)}{1 + y(t - \tau)^{10}} - 0.1\, y(t), \tag{10}$$



(a)

(b)



(c)

Fig. 2: (a) Comparison of feedback and no-feedback cases for x^4 using a single QM reservoir with coherences. (b) Approximation of x^{80} by two coupled QMs with $n_{\text{chip}} = 2$ and by an ESN with $N = 500$ nodes. (c) Test MSE as a function of degree for the best QM configurations (with $n_{\text{chip}} = 1$ and $n_{\text{chip}} = 2$), ESNs of sizes $500, 1000, 2000$, and a linear fit for reference. A subplot shows minimal sizes of ESN N_{min} which match the QM accuracies for degrees $[3, 30]$.

with $\tau = 17$ and integration step $dt = 1$. After numerical integration, the resulting sequence is normalised to the interval $[0, 1]$.

The Santa Fe dataset [28] records the intensity of a far-infrared laser in a chaotic regime; we use 2000 experimental data points normalised to $[0, 1]$.

All datasets, NARMA and time series, are split into washout (20 points), training (480 points), and test (the remaining points). Feedback is implemented using the exponential smoothing rule.

Configuration tests.

To identify the optimal configuration, we tested the available reservoir options in cumulative fashion (Fig. 3a). For all tasks, introducing feedback provides a noticeable accuracy improvement. Adding coherences further improves the results significantly for Mackey–Glass and Santa Fe, whereas the effect is less pronounced for the NARMA tasks. The inclusion of unitaries $U_{\text{in}}, U_{\text{out}}$ on top of coherences is beneficial for NARMA2 and Mackey–Glass, and leaves the accuracy essentially unchanged for the other tasks. Adding U_{meas} does not improve the performance and can even slightly degrade it, presumably because it interferes with the optimisation of $U_{\text{in}}, U_{\text{out}}$.

Therefore, for all time-series benchmarks, except Santa Fe, we select the configuration with feedback, coherences, and unitaries, denoted "Adding $U_{\text{in}}, U_{\text{out}}$" in Fig. 3a. For Santa Fe, we opt for the configuration with only feedback and coherences, denoted "Adding Coherences".

For NARMA10 and NARMA10-st, the improvements beyond the feedback-only configuration are almost negligible, which indicates that these tasks demand memory rather than nonlinear capacity.

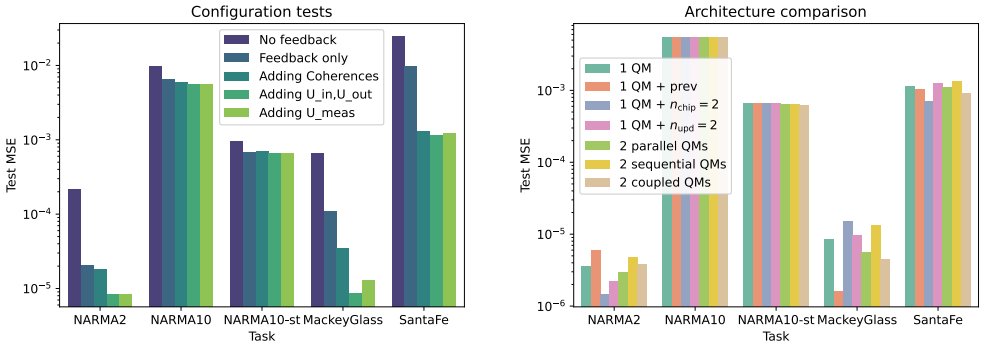
Architecture tests.

Next, we explored processing-depth extensions and two-QM architectures (Fig. 3b); in the legend, "1 QM" denotes the baseline, which is a single-QM reservoir with coherences and unitaries $U_{\text{in}}, U_{\text{out}}$. For NARMA2, the best performance is achieved with n_{chip}=2 extension. For NARMA10 and NARMA10-st, the accuracies of all configurations are almost identical to the 1 QM case, suggesting that these minimal architectural extensions do not increase the effective memory capacity of the reservoir. For Mackey–Glass, the previous-step concatenation ("1 QM + prev"), parallel and coupled QM architectures provide noticeable improvements over the baseline. For Santa Fe, the best performance is achieved with n_{chip}=2 addition.

Based on these results, for the subsequent comparison with ESNs, we select the "1 QM + n_{chip}=2" architecture for NARMA2 task, "1 QM" architecture for the NARMA10 and NARMA10-st tasks, the "1 QM + prev" architecture for Mackey–Glass, and the "1 QM + n_{chip}=2" architecture for Santa Fe. We do not select the two-QM architectures for NARMA2, Mackey–Glass and Santa Fe because they substantially increase the number of optimized parameters, while the corresponding accuracy improvement is moderate (less than an order of magnitude).

Comparison with classical models and a photonic QM.

Figure 4 shows the predicted curves for the NARMA and one-step-ahead time-series tasks obtained with the best QM configurations (denoted "QM best"), together with the no-feedback case ("QM off" in the legend) and the classical baselines: a cubic regression with memory (CM) and ESNs. For NARMA2, CM is defined as $\text{Poly}_3(x_t, x_{t-1})$, for NARMA10 and NARMA10-st we use $\text{Poly}_3(x_t, x_{t-10})$. For



(a) (b)

Fig. 3: (a) Configuration tests for NARMA and time-series benchmarks. MSE gains demonstrated for including feedback, coherences, $U_{\text{in}}, U_{\text{out}}$ and U_{meas} unitary transformations in a cumulative fashion with a single ion reservoir as baseline. (b) Performance of different QM reservoir architectures. The various processing depth options (concatenating features from the previous data point ("prev"), two chip runs ($n_{\text{chip}} = 2$) and two QM updates per data point ($n_{\text{upd}} = 2$)) and two-QM architectures are tested against the single QM performance with coherences and $U_{\text{in}}, U_{\text{out}}$.

Mackey–Glass and Santa Fe, the CM model is $\text{Poly}_3(y_{t-1}, y_{t-2})$. For ESNs, we choose the minimal reservoir sizes N_{min} that match the QM accuracies within a 10% margin for respective tasks.

Table 1 summarises the corresponding test MSE values. For NARMA2, Mackey–Glass, and Santa Fe we also include the results reported for the photonic QM [25], which operates in a two-level Hilbert space. For a fair comparison with the photonic platform, we use the single-QM configuration without architectural extensions, denoted "Trapped-ion QM" in the table. Our trapped-ion implementation uses a three-level system, providing a larger computational space and, as a result, consistently lower MSE values on these shared benchmarks.

To provide a clear resource efficiency comparison, we summarize the number of readout weights W_{ro} and other optimisable parameters Φ via random search for the QM and ESN reservoirs for each task in Table 2. The QM reservoirs use 9 readout weights, or 18 when $n_{\text{chip}} = 2$ or previous-step features are implemented. Optimisable parameters for the QM include U_1 gate angle θ, feedback constant m and two 3×3 complex unitaries $U_{\text{in}}, U_{\text{out}}$ with 18 real parameters each (unitaries are not used for Santa Fe task). The ESN optimizes three hyperparameters (spectral radius ρ, leaking rate α and an input scaling parameter) and trains N readout weights.

For NARMA10 and NARMA10-st, classical ESNs achieve the QM accuracy with very few nodes, indicating that our QM is not competitive in memory-dominated tasks. Mackey–Glass proves to be a rather simple task, where the no-feedback configuration achieves MSE$\sim 10^{-5}$. The QM reservoir shows competitive performance in Santa Fe task, which contains high level of nonlinearity in the data, with ESN needing 18 nodes to match its accuracy.

Table 1: Test MSE for the time-series and NARMA benchmarks. "QM best" denotes the best QM configuration selected in Sec. 3.2; "QM off" is the same reservoir without feedback; CM is the cubic regression with memory; ESN lists the minimal reservoir size $N_{\min}$ needed to match the QM accuracy. Photonic QM data are taken from Ref. [25].

Model	NARMA2	NARMA10	NARMA10-st	Mackey–Glass	Santa Fe
QM best	1.5×10^{-6}	5.5×10^{-3}	6.6×10^{-4}	1.6×10^{-6}	7.9×10^{-4}
QM off	2.1×10^{-4}	9.4×10^{-3}	9.5×10^{-4}	2.4×10^{-5}	2.3×10^{-2}
CM	5.2×10^{-4}	1.2×10^{-2}	1.5×10^{-3}	1.3×10^{-4}	1.1×10^{-2}
ESN	$N_{\min} = 7$	$N_{\min} = 3$	$N_{\min} = 1$	$N_{\min} = 4$	$N_{\min} = 18$
Photonic QM	2.6×10^{-5}	n/a	n/a	2.2×10^{-4}	9.2×10^{-3}
Trapped-ion QM	3.6×10^{-6}	5.5×10^{-3}	6.6×10^{-4}	8.6×10^{-6}	1.2×10^{-3}

Table 2: Resource summary for the best QM configurations and the minimal ESNs matching their accuracy. The notation "$W_{\text{ro}} + \Phi$" means W_{ro} readout weights and Φ other optimisable parameters.

Task	QM ($W_{\text{ro}} + \Phi$)	ESN ($W_{\text{ro}} + \Phi$)
NARMA2	$18 + 38$	$7 + 3$
NARMA10	$9 + 38$	$3 + 3$
NARMA10-st	$9 + 38$	$1 + 3$
Mackey–Glass	$18 + 38$	$4 + 3$
Santa Fe	$18 + 2$	$18 + 3$

Memory–nonlinearity trade-off.

To verify that the nonlinearity of the QM reservoir dominates its dynamical memory, we performed a systematic scan of the exponential smoothing parameter m over the range $m \in [1.01, 1000]$, re-optimising all other parameters for each fixed value of m, using the best respective configuration for each task. The resulting test MSE curves for NARMA and time-series tasks are shown in Fig. 5; the accuracies remain essentially stable over the entire range of m, with only shallow minima at small m for NARMA2 and Mackey–Glass.

It is important to note that the case $m = 1000$ is not equivalent to the no-feedback regime: even for large m, the state parameter s slowly drifts towards the running average of $P_{|a\rangle}$, effectively providing an adaptive bias that tracks the input statistics. The fact that this slow memory alone already yields nearly optimal accuracy indicates that the nonlinear mapping of the QM reservoir is the dominant computational resource, while the exact memory timescale is of secondary importance. Such insensitivity to the memory decay parameter is a clear experimental signature of the memory–nonlinearity trade-off widely discussed in reservoir computing theory.

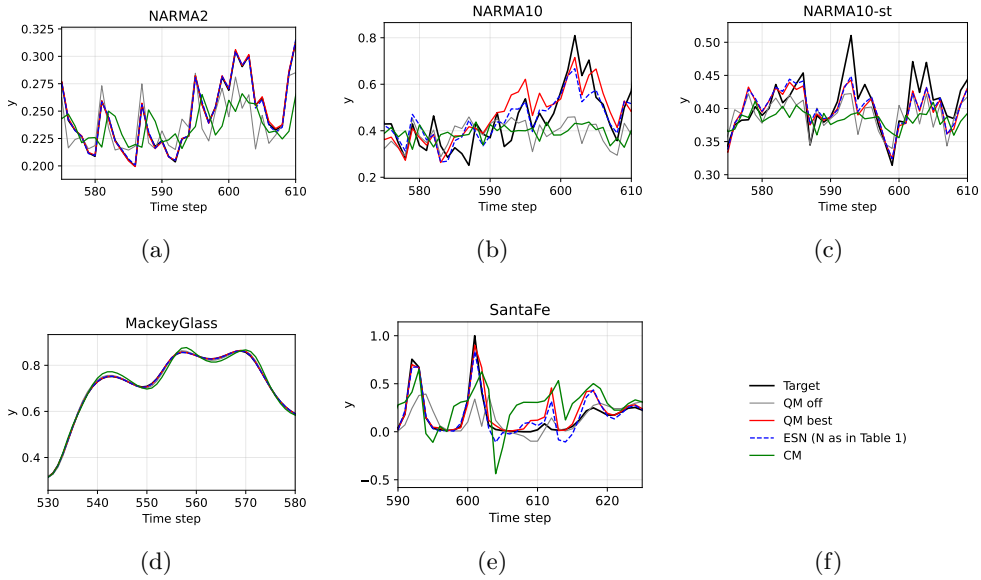


Fig. 4: Performance of different learning models on the NARMA and time-series benchmarks. The curves show the target signal, the best QM reservoir (QM best), the same reservoir without feedback (QM off), the minimal ESN matching QM accuracy (ESN), and the cubic regression with memory (CM). Detailed MSE values are provided in Table 1.

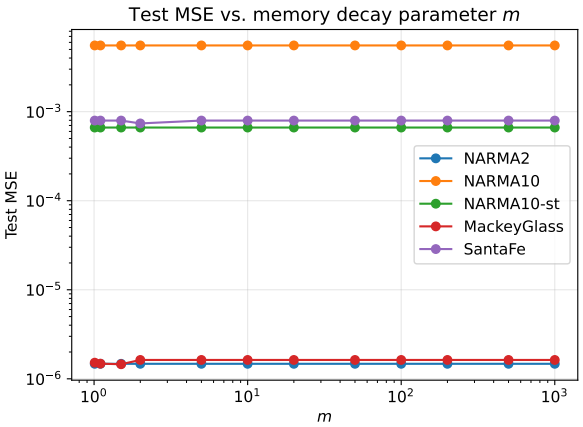
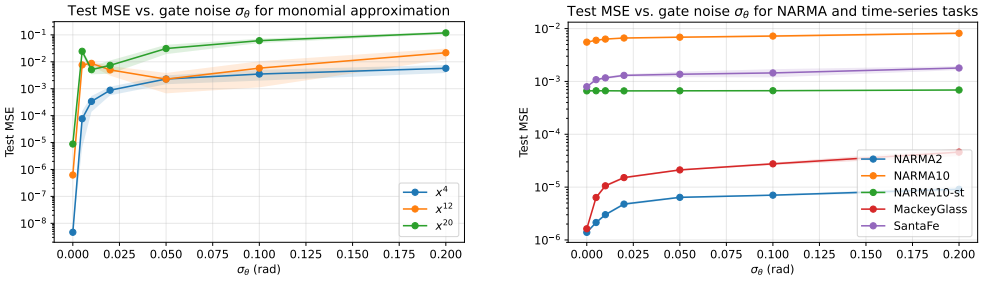


Fig. 5: Test MSE as a function of the exponential smoothing parameter m for the NARMA and time-series benchmarks.



(a) (b)

Fig. 6: Robustness to dynamic gate-angle noise for (a) monomial benchmarks x^d for $d = 4, 12, 20$; and (b) NARMA and time-series benchmarks. The curves show the mean test MSE over 20 noise realisations; shaded error bands indicate the standard deviation.

3.3 Robustness to gate noise

To assess the sensitivity of the QM reservoir to experimental imperfections, we introduced Gaussian noise into the rotation angles of the unitary gates. Specifically, at every application of a gate parameterized by an angle θ (or θ_1, θ_2 in two-QM architectures), we added an independent random offset $\delta\theta \sim \mathcal{N}(0, \sigma_\theta^2)$. This models dynamic gate fluctuations during operation. In all experiments, the QM reservoir uses the best configurations determined in the previous sections: for monomial benchmarks this is the two coupled QMs with $n_{\text{chip}} = 1$, and for NARMA and the time-series tasks we use the best respective architectures selected in Sec. 3.2.

For each noise level σ_θ in the range $[0, 0.2]$ rad, we performed 20 independent simulations. In each simulation, the QM reservoir used the best parameters found in the noise-free optimization, but the gate angles were perturbed as described above. The linear readout layer was re-trained on the noisy training features for each noise level. We report the mean test MSE and its standard deviation over the 20 realizations.

Fig. 6 summarizes the results. For the monomial benchmarks (Fig. 6a), even small gate noise leads to rapid degradation of accuracy, especially for high degrees d, indicating that the extreme nonlinearity of the reservoir relies on precise control of the gate angles. In contrast, the time-series tasks (Fig. 6b) show remarkable robustness: the test MSE remains stable over a wide range of σ_θ, with only a moderate increase at the largest noise levels for NARMA2 and Mackey–Glass. This difference likely stems from the different nature of the two computational regimes. In monomial approximation, the output must reproduce a sharply varying function, and even a small gate error is immediately reflected in the feature vector, leading to a large error for high degrees d. In contrast, time-series tasks possess an intrinsic temporal redundancy: the dynamical memory and feedback can partially compensate for random gate fluctuations, making the readout less sensitive to individual gate errors.

4 Conclusions

We presented a comprehensive numerical study of a QM reservoir implemented on a trapped-ion platform and evaluated its performance on a suite of standard reservoir-computing benchmarks.

The central finding of this work is the exceptional scalability of the nonlinear capacity of the QM reservoir. With minimal extensions of the single QM reservoir to two coupled QMs with $n_{\text{chip}} = 2$ chip passes, quantum-coherence readout, and additional unitary transformations $U_{\text{in}}, U_{\text{out}}, U_{\text{meas}}$, it was possible to maintain a test MSE below 10^{-2} for monomial targets x^d up to degree $d = 100$, while classical echo state networks with up to 2000 nodes fail beyond $d \approx 40$ and show no improvement with increasing size. We attribute this advantage to the rich functional basis generated by Born-rule nonlinearities and quantum coherences acting on a three-level (or five-level) Hilbert space, coupled with feedback-controlled memory that allows the reservoir state to track the input history. This result positions QM reservoir as a resource-efficient alternative to classical ESNs in strongly nonlinear regimes.

On NARMA tasks and standard time-series benchmarks — Mackey–Glass and Santa Fe — the QM reservoir matches the accuracy of minimal-size classical ESNs. A systematic ablation study revealed that feedback-controlled memory is essential for accurate performances on all tasks except the simpler Mackey–Glass benchmark. However, its memory capacity is limited, which is proven by how the QM reservoir's accuracy is independent of memory decay parameter m and especially with memory-dominated tasks like NARMA10 and NARMA10-st, where the minimal tested architectural expansions cannot improve the performance, meaning they do not provide a significant improvement in memory capacity.

From a resource perspective, the QM reservoir achieves competitive accuracy with a small number of trainable readout weights in tasks with high nonlinearities; while it involves more optimisable parameters than classical ESNs in tasks which require more memory capacity, the random search over a fixed number of candidate configurations (up to 300) keeps the optimization cost moderate. In addition, the ability to define multiple coupled QMs on a single ion [17] reduces the number of trapped ions required, which is a key advantage of the platform.

A robustness study showed that the QM reservoir is highly resilient to gate noise for time-series tasks, whereas monomial approximation is sensitive to gate errors, reflecting the different computational regimes: reliable processing of the sharp rise in the monomial test region requires high gate stability.

The present study is a numerical proof of principle. While an experimental implementation on trapped ions would remove the overhead of simulating the gate dynamics, it would not eliminate the need for classical optimization of the reservoir hyperparameters, and the feedback protocol still requires repeated projective measurements and re-initialization at each time step. Moreover, state preparation and unitary transformations rely on composite laser pulses with finite fidelity. The primary long-term motivation for experimental realization is therefore not efficiency at small scales, but rather the possibility of scaling to larger ion registers, where classical simulation eventually becomes costly.

Several directions remain open for future work. First, while the empirical evidence strongly suggests a scalable nonlinear capacity that outpaces classical ESNs, a theoretical characterization of the functional basis spanned by the QM readout — analogous to the information-processing capacity framework of Ref. [29] — would provide deeper insight into why the advantage persists at high degrees. Second, avenues for potential memory capacity scaling need to be investigated.

In summary, our results establish trapped-ion quantum memristors as a promising platform for QRC, particularly suitable for tasks governed by strong nonlinearity, and motivate further theoretical and experimental investigations.

Declarations

- **Funding.** Russian Science Foundation grant 24-12-00415, https://rscf.ru/project/24-12-00415/.
- **Conflict of interest.** The authors declare no conflicts of interest.
- **Data availability.** The Python code used to generate the results and figures presented in this work is available as Supplemental Material. All other data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks. GMD Report **148** (2001)

[2] Mujal, P., R. Martínez-Peña, Nokkala, J., Garcia-Beni, J., Giorgi, G.L., Soriano, M.C., Zambrini, R.: Opportunities in quantum reservoir computing and extreme learning machines. Adv. Quantum Technol. **4**(8), 2100027 (2021) https://doi.org/10.1002/qute.202100027

[3] Zhu, C., Ehlers, P.J., Nurdin, H.I., Soh, D.: Practical few-atom quantum reservoir computing. Phys. Rev. Res. **7**, 023290 (2025) https://doi.org/10.1103/PhysRevResearch.7.023290

[4] Fry, D., Deshmukh, A., Chen, S.Y.-C., Rastunkov, V., Markov, V.: Optimizing quantum noise-induced reservoir computing for nonlinear and chaotic time series prediction. Sci. Rep. **13**, 19326 (2023) https://doi.org/10.1038/s41598-023-45015-4 arXiv:2303.05488

[5] Sannia, A., R. Martínez-Peña, Soriano, M.C., Giorgi, G.L., Zambrini, R.: Dissipation as a resource for quantum reservoir computing. Quantum **8**, 1291 (2024) https://doi.org/10.22331/q-2024-03-20-1291 arXiv:2212.12074 [quant-ph]

[6] Kobayashi, K., Fujii, K., Yamamoto, N.: Feedback-driven quantum reservoir computing for time-series analysis. PRX Quantum **5**, 040325 (2024) https://doi.org/10.1103/PRXQuantum.5.040325 arXiv:2406.15783

[7] Settino, J., Salatino, L., Mariani, L., D'Amore, F., Channab, M., Bozzolo, L., Vallisa, S., P. Barillà, Policicchio, A., Gullo, N.L., Giordano, A., Mastroianni, C., Plastina, F.: Memory-augmented hybrid quantum reservoir computing. Phys. Rev. Appl. **24**, 024019 (2025) https://doi.org/10.1103/PhysRevApplied.24.024019

[8] Zhu, C., Ehlers, P.J., Nurdin, H.I., Soh, D.: Minimalistic and scalable quantum reservoir computing enhanced with feedback. npj Quantum Inf. **11** (2025) https://doi.org/10.1038/s41534-025-01144-4

[9] Pfeiffer, P., Egusquiza, I.L., Ventra, M.D., Sanz, M., Solano, E.: Quantum memristors. Sci. Rep. **6**, 29507 (2016) https://doi.org/10.1038/srep29507

[10] Salmilehto, J., Deppe, F., Ventra, M.D., Sanz, M., Solano, E.: Quantum memristors with superconducting circuits. Sci. Rep. **7**, 42044 (2017) https://doi.org/10.1038/srep42044

[11] Kumar, S., F. A. Cárdenas-López, Hegade, N.N., Chen, X., F. Albarrán-Arriagada, Solano, E., G. Alvarado Barrios: Entangled quantum memristors. Phys. Rev. A **104**, 062605 (2021) https://doi.org/10.1103/PhysRevA.104.062605

[12] Kumar, S., F. A. Cárdenas-López, Hegade, N.N., F. Albarrán-Arriagada, Solano, E., G. Alvarado Barrios: Tripartite entanglement in quantum memristors. Phys. Rev. Appl. **18**, 034004 (2022) https://doi.org/10.1103/PhysRevApplied.18.034004

[13] Sanz, M., Lamata, L., Solano, E.: Invited article: Quantum memristors in quantum photonics. APL Photonics **3**(8), 080801 (2018) https://doi.org/10.1063/1.5036596

[14] Gonzalez-Raya, T., Lukens, J.M., L. C. Céleri, Sanz, M.: Quantum memristors in frequency-entangled optical fields. Materials **13**(4), 864 (2020) https://doi.org/10.3390/ma13040864

[15] Spagnolo, M., Morris, J., Piacentini, S., Antesberger, M., Massa, F., Crespi, A., Ceccarelli, F., Osellame, R., Walther, P.: Experimental photonic quantum memristor. Nat. Photon. **16**, 318–323 (2022) https://doi.org/10.1038/s41566-022-00973-5

[16] Stremoukhov, S., Forsh, P., Khabarova, K., Kolachevsky, N.: Proposal for trapped-ion quantum memristor. Entropy **25**(8), 1134 (2023) https://doi.org/10.3390/e25081134

[17] Stremoukhov, S., Forsh, P., Frolova, A., Kovalishin, I., Rusakov, D., Shakirov, M., Khabarova, K., Kolachevsky, N.: Quantum memristors: Towards scalable quantum neuromorphic architectures with coupling. Phys. Rev. A **113**(3), 032619 (2026) https://doi.org/10.1103/PhysRevA.113.032619

[18] Norambuena, A., Torres, F., Ventra, M.D., Coto, R.: Polariton-based quantum memristors. Phys. Rev. Appl. **17**, 024056 (2022) https://doi.org/10.1103/PhysRevApplied.17.024056

[19] J.-L. Tang, G. Alvarado Barrios, Solano, E., F. Albarrán-Arriagada: Tunable non-markovianity for bosonic quantum memristors. Entropy **25**(5), 756 (2023) https://doi.org/10.3390/e25050756

[20] Sweke, R., Sanz, M., Sinayskiy, I., Petruccione, F., Solano, E.: Digital quantum simulation of many-body non-markovian dynamics. Phys. Rev. A **94**, 022317 (2016) https://doi.org/10.1103/PhysRevA.94.022317

[21] Tacchino, F., Macchiavello, C., Gerace, D., Bajoni, D.: An artificial neuron implemented on an actual quantum processor. npj Quantum Inf. **5**, 26 (2019) https://doi.org/10.1038/s41534-019-0140-4

[22] Y.-M. Guo, F. Albarrán-Arriagada, Alaeian, H., Solano, E., G. Alvarado Barrios: Quantum memristors with quantum computers. Phys. Rev. Appl. **18**, 024082 (2022) https://doi.org/10.1103/PhysRevApplied.18.024082

[23] Kutvonen, A., Fujii, K., Sagawa, T.: Optimizing a quantum reservoir computer for time series prediction. Sci. Rep. **10**, 14687 (2020) https://doi.org/10.1038/s41598-020-71673-9 arXiv:1806.07519

[24] Chen, J., Nurdin, H.I., Yamamoto, N.: Temporal information processing on noisy quantum computers. Phys. Rev. Appl. **14**, 024065 (2020) https://doi.org/10.1103/PhysRevApplied.14.024065 arXiv:2001.09498

[25] Selimović, M., Agresti, I., Siemaszko, M., Morris, J., Dakić, B., Albiero, R., Crespi, A., Ceccarelli, F., Osellame, R., Stobińska-Moretto, M., Walther, P.: Experimental neuromorphic computing based on quantum memristor. PRX Quantum **7**(3) (2026) https://doi.org/10.1103/jknv-3tx7

[26] Atiya, A.F., Parlos, A.G.: New results on recurrent network training: unifying the algorithms and accelerating convergence. IEEE Trans. Neural Netw. **11**(3), 697–709 (2000) https://doi.org/10.1109/72.846741

[27] Kubota, T., Takahashi, H., Nakajima, K.: A unifying framework for information processing in stochastically driven dynamical systems (2019)

[28] Weigend, A.S., Gershenfeld, N.A.: Time series prediction: Forecasting the future and understanding the past. In: Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley, ??? (1994)

[29] Dambre, J., Verstraeten, D., Schrauwen, B., Massar, S.: Information processing capacity of dynamical systems. Scientific Reports **2**(1), 514 (2012) https://doi.org/10.1038/srep00514

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ionreservoirgeneralized.ipynb