Quantum Reservoir Computing Using Arrays of Rydberg Atoms

Rodrigo Araiza Bravo,[1,*] Khadijeh Najafi,[1,2,†] Xun Gao,[1] and Susanne F. Yelin[1]

[1] *Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA*

[2] *IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598, USA*

(Received 24 November 2021; revised 26 May 2022; accepted 26 July 2022; published 22 August 2022)

Quantum computing promises to speed up machine-learning algorithms. However, noisy intermediate-scale quantum (NISQ) devices pose engineering challenges to realizing quantum machine-learning (QML) advantages. Recently, a series of QML computational models inspired by the noise-tolerant dynamics of the brain has emerged as a means to circumvent the hardware limitations of NISQ devices. In this paper, we introduce a quantum version of a recurrent neural network (RNN), a well-known model for neural circuits in the brain. Our quantum RNN (qRNN) makes use of the natural Hamiltonian dynamics of an ensemble of interacting spin-1/2 particles as a means for computation. In the limit where the Hamiltonian is diagonal, the qRNN recovers the dynamics of the classical version. Beyond this limit, we observe that the quantum dynamics of the qRNN provide it with quantum computational features that can aid it in computation. To this end, we study a fixed-geometry qRNN, i.e., a quantum reservoir computer, based on arrays of Rydberg atoms and show that the Rydberg reservoir is indeed capable of replicating the learning of several cognitive tasks such as 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.

DOI: 10.1103/PRXQuantum.3.030325

I. INTRODUCTION

Quantum computing promises to enhance machine-learning algorithms. However, the implementation of these advantages often relies on either fault-tolerant quantum computers that are not yet available [1–5] or on decoherence-limited variational quantum circuits that may experience training bottlenecks [6,7]. Thus, the currently available noisy intermediate-scale quantum (NISQ) devices thwart the quantum advantages in machine-learning algorithms.

Recently, to counteract these challenges, several quantum machine-learning architectures have emerged, inspired by models for computation in the brain [8–10]. These brain-inspired algorithms are motivated by the inherent robustness of input and hardware noise in brain-like computation and by the possibility of using the analog dynamics of controllable many-body quantum systems for computation without relaying on a digital circuit architecture. Broadly speaking, these brain-inspired algorithms can be put into two categories, the first of which encompasses systems quantizing the dynamics of biological computational models at the single-neuron level. Thus, the dynamics of single qubits or of groups of qubits resemble the dynamics of a neurons in a neural circuit of interest. Examples of these include quantum memristors [11], which are electrical circuits with a history-dependent resistance, quantum versions of the biologically realistic Hodgkin-Huxley model for single neurons [12,13], and the unitary adiabatic quantum perceptron [14].

The second category of brain-inspired algorithms relies on a macroscopic resemblance between many-body quantum systems and neural circuits. In this regard, the algorithms that have received the most attention are quantum reservoir computers, which use ensembles of quantum emitters with fixed interactions to perform versatile machine-learning tasks, relying on the complexity of the unitary evolution of the system. Since these systems can couple with both classical and quantum devices, which may encode the input of the tasks, quantum reservoirs have been used for time-series prediction [15–17], entanglement measurement [18,19], quantum state preparation [20], continuous-variable computation [21], which can be made universal [22], reduction of depths in quantum circuits [23], ground-state finding [24], and long-term memory employing ergodicity-breaking dynamics [25–27]. For a comprehensive review of quantum reservoir computing, see Ref. [10].

*oaraizabravo@g.harvard.edu

†knajafi@ibm.com

In both categories, however, a thorough understanding of the potential computational advantages and their origins is slowly emerging. In this paper, we contribute to this direction by proposing a quantum extension of a well-known neural-circuit model called a recurrent neural network (RNN), of which reservoir computers are a special case [28]. Our extension uses the Hamiltonian dynamics of ensembles of two-level systems. In the limit where the Hamiltonian is diagonal, we recover the classical single-neuron dynamics that naturally encode RNNs into quantum hardware. Recently, another natural encoding of a reservoir computer has been proposed using superconducting qubits [29]. In our case, the general dynamics of the quantum RNN (qRNN) present several new features that can aid in the computation of both classical and quantum tasks. In particular, a qRNN used for simulating stochastic dynamics can exhibit speed-ups compared to classical RNNs.

To show that our scheme is experimentally realizable, we propose that arrays of Rydberg atoms can be used as qRNNs (Sec. IV). Although our Rydberg qRNNs have restricted connectivity, we are motivated to use Rydberg arrays due to recent studies with equally restricted qRNNs, which show significant computational capacity when driven near criticality [17,24]. Moreover, recent experiments using optical tweezers [30–37] have catapulted the community's interest in Rydberg arrays, as they exhibit long coherence times, controllable and scalable geometries, and increasing levels of single-atom control [38]. Additionally, Rydberg arrays can be used for novel programmable quantum simulations and universal computations [30,39–43].

We numerically implement fixed-geometry Rydberg qRNNs, i.e., Rydberg-reservoir computers, and we successfully perform cognitive tasks even when just a few atoms are available (Sec. V). The success of these tasks is explained by the physics of Rydberg atoms. For example, our Rydberg qRNNs excel at learning to multitask since they can naturally encode RNNs with inhibitory and excitatory neurons, which are vital for many cognitive tasks [44]. This encoding relies on the different types of interactions between Rydberg atoms with different principal quantum numbers [45]. Likewise, a Rydberg qRNN exhibits long-term memory due to the weak ergodicity-breaking dynamics of many-body quantum scars [35,46,47]. Lastly, we discuss possible further research directions in Sec. VI.

We remark that the notion of qRNNs has been previously coined relying on universal quantum circuits and using measurements to implement the nonlinear dynamics of a RNN [48]. Instead, what we define as a "quantum RNN" leverages the inherent unitary dynamics of ensembles of two-level systems to compute, deviating from the quantum digital circuit model for computation.

II. CLASSICAL RECURRENT NEURAL NETWORKS

We begin by reviewing an archetypal RNN consisting of N binary neurons. Each neuron is in one of two possible states $s_n(t) \in \{-1, 1\}$ and is updated from the time step t to $t+1$ following the update rule

$$\begin{aligned} s_n(t+1) &= \text{sign}\left(h_n(t)s_n(t)\right), \\ h_n(t) &\equiv -\Delta_n(t) + \sum_m J_{nm}s_m(t), \end{aligned} \tag{1}$$

where the $J_{nm} = J_{mn}$ are symmetric synaptic connections between neurons n and m. The time-dependent biases $\Delta_n(t)$ encode the inputs of the RNNs. To avoid memorization during a learning task with inputs $u_n^{\text{task}}(t)$, the RNN receives Gaussian-whitened inputs

$$\Delta_n(t) = u_n^{\text{task}}(t) + \xi_n, \tag{2}$$

where ξ_n is a zero-mean Gaussian random variable with variance σ_{in}^2, making the evolution of the RNN stochastic. In RNNs, the value of σ_{in}^2 is proportional to the value of the inputs u_n^{task} of the task.

When studying learning tasks similar to those in the mammalian cortex [44], one turns to a continuous version of the rule in Eq. (1) obtained in the case in which the time interval τ in which neurons update is small compared to J_{nm}. In this limit,

$$\tau \dot{s}_n(t) = -s_n(t) + \text{sign}\left(h_n(t)s_n(t)\right). \tag{3}$$

Thus, the RNN obeys a system of nonlinear differential equations. Note that Eq. (3) implies that $s_n \in [-1, 1]$ is a continuous and bounded variable [28].

A third way to describe a RNN is via the probability distribution $p_t(\mathbf{s})$ of observing each of the 2^N different configurations $\mathbf{s}$ at the t^{th} time step. Due to the noise in the inputs Δ_n, the dynamics of the distribution follow a Markov-chain description [28]. This description is particularly useful for analyzing the stochastic dynamics simulatable by a RNN. As we see in Sec. III A 3, this representation is useful in explaining how, relative to classical RNNs, the unitary dynamics of a qRNN can speed up stochastic process simulations.

Lastly, we describe how to use a RNN for computation. After the RNN evolves for a time t_f, a subset of M neurons is used to collect the vector $\mathbf{r}(t_f) = (s_{n_1}(t_f), \ldots, s_{n_M}(t_f), 1)$ with the last entry accommodating a bias. The other $N - M$ neurons are called *hidden neurons*. The output of the RNN is obtained via a linear transformation $\mathbf{y}^{\text{out}} = W^{\text{out}}\mathbf{r}(t_f)$, where W^{out} is a real-valued matrix. Thus, the computational complexity of the RNN comes from the nonlinear activation function in Eq. (1), which enables $\mathbf{y}^{\text{out}}$ to be a nonlinear function of the inputs.

In a learning task with a target output $\mathbf{y}^{\text{targ}}$, the RNN is trained by minimizing a loss function $\mathcal{L}(\mathbf{y}^{\text{out}}, \mathbf{y}^{\text{targ}})$ with respect to the network parameters such as W^{out}, J_{nm}, etc. subject to the task-determined inputs in Eq. (2). We choose the mean-square loss

$$\mathcal{L}(\mathbf{y}^{\text{out}}, \mathbf{y}^{\text{targ}}) = \frac{1}{N_s} \sum_{i=1}^{N_s} ||\mathbf{y}_i^{\text{targ}} - \mathbf{y}_i^{\text{out}}||^2, \tag{4}$$

where i labels the N_s different input instances. For the tasks in Sec. V, we fix the connections J_{nm} such that our qRNNs more closely resemble quantum reservoir computers.

III. QUANTUM RECURRENT NEURAL NETWORKS

A. Quantum update rule

Let us now extend the classical RNN in Eq. (1) to the quantum setting. We replace each of the N neurons with a spin-1/2 particle for which a spin measurement along the z axis yields the values $\{-1, 1\}$. Thus, each neuron n is in a normalized quantum state in the Hilbert space $\mathcal{H}_n$ with basis vectors $\{|-1\rangle_n, |1\rangle_n\}$ that are eigenstates of the Pauli Z operator $\sigma_n^z = |1\rangle\langle 1|_n - |-1\rangle\langle -1|_n$. The state of the composite system lives in the product Hilbert space $\mathcal{H} = \bigotimes_{n=1}^{N} \mathcal{H}_n$.

We choose spins interacting via the time-dependent Hamiltonian

$$H(t) = -\sum_{n=1}^{N} \Delta_n(t)\sigma_n^z + \sum_{nm} J_{nm}\sigma_n^z\sigma_m^z + \frac{\Omega(t)}{2}\sum_{n=1}^{N}\sigma_n^x, \tag{5}$$

where $\sigma_n^x = |1\rangle\langle -1|_n + |-1\rangle\langle 1|_n$ is the Pauli X operator. Indeed, the evolution under Eq. (5) encompasses the update rule in Eq. (1). To see this, note that in the classical case of Eq. (1), the RNN evolves under the rules

$$\text{If } h_n > 0, \quad s_n \text{ does not change.} \tag{1C}$$

$$\text{If } h_n < 0, \quad s_n \text{ flips.} \tag{2C}$$

Here, "C" denotes "classical." Now, consider a qRNN starting in the configuration $|s_1, s_2, \ldots, s_N\rangle$ and evolving for a time $t = 2\,\pi\Omega^{-1}$. In the limit where $\Delta_n \gg \Omega$ or $J_{nm} \gg \Omega$, each spin experiences the Hamiltonian $H_n = h_n\sigma_n^z + (\Omega/2)\sigma_n^x$, where $h_n = -\Delta_n + \sum_m J_{nm}s_m$ is the effective field generated by the rest of the spins, in which s_m denotes the measurement result of σ_m^z on the initial configuration. We then obtain the quantum update rules

$$\text{If } |h_n| \gg \Omega, \quad |s_n\rangle \text{ does not change.} \tag{1Q}$$

$$\text{If } |h_n| \ll \Omega, \quad |s_n\rangle \text{ flips.} \tag{2Q}$$

Here, "Q" denotes "quantum." Therefore, Eq. (5) can implement Eqs. (1C) and (2C) but without the use of the nonlinear activation function in Eq. (1). Nonetheless, Eq. (5) allows for more general dynamics beyond the perturbative limit for which Eqs. (1Q) and (2Q) hold. We now highlight three features arising from the quantum evolution of the qRNN: (i) the ability to compute complex functions on the input by using quantum interference; (ii) exploiting the choice of measurement basis; and (iii) efficiently achieving stochastic processes that are inaccessible to classical RNNs with no hidden neurons.

1. Quantum feature 1: Quantum interference as a means for computation

The computational power of Eq. (1) is a result of its nonlinear dynamics. For example, a RNN with linear dynamics is incapable of computing the parity function $\text{XOR}(s_1, s_2) = s_1 s_2$ between two classical binary inputs. On the other hand, quantum mechanics is a unitary theory. Yet this does not limit a qRNN to linear computation. Indeed, a qRNN can compute XOR by leveraging quantum interference, a resource fundamental to quantum computation. Thus, we can use a qRNN for complex computing tasks.

As illustrated in Fig. 1, we can compute $\text{XOR}(s_1, s_2)$ using a qRNN of three spins initially in the state $|s_1, s_2, -1\rangle$. The third spin is an outcome spin. This spin is measured to tell us information about the parity of s_1 and s_2. We let these spins evolve under the dynamics dictated by Eq. (5), choosing $\Delta_n, J_{12} = 0$ and $J_{13} = J_{23} = J \gg \Omega$. Let $\tilde{J} = J(s_1 + s_2)$. In the frame rotating at the rate $\tilde{J}$, the output spin experiences the Hamiltonian

$$H_3 = \frac{\Omega}{2}\left(e^{2i\tilde{J}\tau}|1\rangle\langle -1| + \text{h.c.}\right). \tag{6}$$

It is clear that if the spins have odd parity (i.e., $s_1 = -s_2$ so that $\tilde{J} = 0$), the output spin flips to the state $|1\rangle$ when we choose to evolve by $t = 2\,\pi\Omega^{-1}$. On the other hand, if $\tilde{J} \neq 0$, H_3 contains only fast-rotating terms and the rotating-wave approximation (RWA) allows us to neglect the evolution of the output spin [49]. Physically, the RWA can be thought of as the spin rotating along the x axis by a small amount followed by a rapid precession of the spin around the z axis. Indeed, as illustrated in Fig. 1, $J \gg t^{-1}$ amounts to averaging out the position of the spin so that the spin is along the z axis. Overall, this computation realizes the operation $|s_1, s_2, -1\rangle \rightarrow |s_1, s_2, \text{XOR}(s_1, s_2)\rangle$.

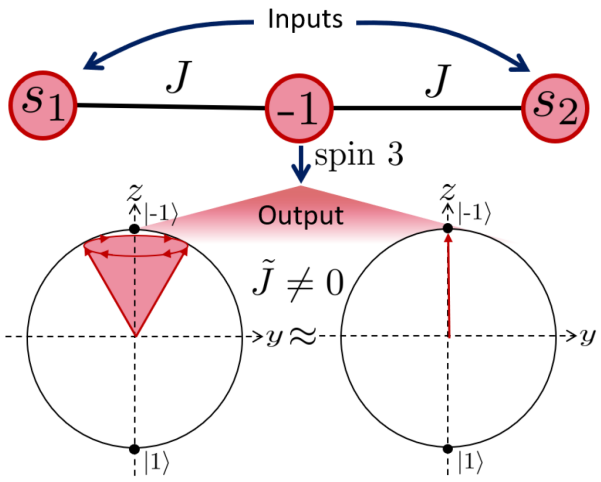

FIG. 1. The computation of the parity, $\text{XOR}(s_1, s_2)$, of two inputs s_1 and s_2 with a qRNN. Spin 3 (the output spin) experiences an effective field $\tilde{J} = J(s_1 + s_2)$ with $J \gg \Omega$. After evolving for a time $t = 2\pi\Omega^{-1}$, we measure the output spin. The measurement outcome $+1$ is obtained when $s_1 = -s_2$, since $\tilde{J} = 0$. If $s_1 = s_2$, so that $\tilde{J} \neq 0$, the inputs constructively interfere to generate a large detuning on the output such that measurement yields the outcome -1.

Note that this is a result of $s_1 + s_2$ constructively interfering to produce a large effective detuning on the output and blocking its evolution. Thus, interference serves as a means for computation in qRNNs.

2. *Quantum feature 2: Arbitrary measurement basis as a means for computation*

Equations (1Q) and (2Q) recover Eq. (1) when $t = 2\pi\Omega^{-1}$. However, $t = 2\pi\Omega^{-1}$ is not a necessary restriction. This freedom results in the ability to rotate each quantum neuron, which can be used as means for computing on a different basis. Measuring on different bases reveals the quantum correlations enhancing the performance of a qRNN relative to its classical counterpart. In this section, we show how to use the evolution of the qRNN to change the basis on which an error occurs. This freedom can detect a Z error, an error proper to quantum computation.

Consider the repetition codes $|0_L\rangle = |-_y\rangle^{\otimes 3}$ and $|1_L\rangle = |+_y\rangle^{\otimes 3}$ on qubits labeled $L_{1,2,3}$, where $|\pm_y\rangle = \frac{1}{\sqrt{2}}(|-1\rangle \pm i|1\rangle)$. Suppose that we prepare the state $|\psi\rangle = a|0_L\rangle + b|1_L\rangle$ and consequently a Z error occurs. We can detect the error by rotating all three spins $L_{1,2,3}$ using Eq. (5) with the dominant field being Ω for a time $t = \pi/2\Omega$. Note that the rotation conjugates the Z error by

$$e^{-i\pi\sigma^x/4}\sigma^z e^{i\pi\sigma^x/4} \propto \sigma^y, \tag{7}$$

where $\sigma_n^y = i|-1\rangle\langle 1| - i|1\rangle\langle -1|$ is like a bit-flip error except for a state-dependent phase. A bit-flip error can then be detected by bringing two extra spins $A_{1,2}$ and performing parity measurements of the pairings $(L_1, L2)$, and (L_2, L_3) as described in Sec. III A 1. Using Table I, the final parity of $(L_1, L2)$, and (L_2, L_3) gives the measurement results a_1 and a_2, which can be used to discern where the Z error has occurred.

TABLE I. The results of the parity measurements for the detection of a Z error. The measurement of spin A_i results in the outcome a_i. By comparing the outcomes, one can detect the location of the Z error.

a_2	a_1	
	-1	$+1$
-1	Error in L_2	Error in L_1
$+1$	Error in L_3	No error

As an example, Fig. 2 illustrates the two final states of L_3 if no error occurs (bottom left) and if a Z error occurs on L_3 (bottom right).

Detection of the Z error hinges on Eq. (7) and can be achieved by using the evolution of the qRNN to rotate the

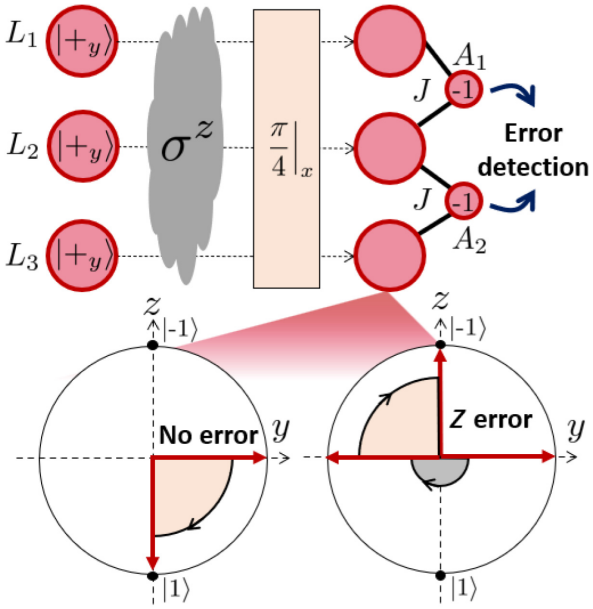

FIG. 2. The detection of a Z error on three spins $L_{1,2,3}$ using a quantum RNN. A Z error is conjugated into a bit-flip-like error using a Hamiltonian generating a rotation along the x axis, where $t = \pi/2\Omega$ and Ω is the dominant field of the Hamiltonian. The state of each of the L_i after the rotation (orange region) depends on whether a Z error occurs, as illustrated at the bottom of the figure. As exemplified here for L_3, a Z error results in a spin flipping from what we would expect in the absence of errors. To detect the Z error, a set of auxiliary qubits $A_{1,2}$ is brought in to perform a parity measurements of pairs (L_1, L_2) and (L_2, L_3). Since under no Z error the parity measurements must match, the parity measurements allow us to detect the location of the Z error as specified in Table I.

measurement basis. Note that rotation allows us to measure the error syndrome of the stabilizer state $|\psi\rangle$, bringing out the quantum correlations of the state. Thus, the native evolution of the qRNN can be used to perform quantum computational tasks. After the error is detected on spin L_i, all qubits are rotated again by $U^\dagger$ and σ_i^z can be applied to correct the error. We note that using a repetition code for error detection is a well-known technique in the quantum computing community.

The previous two quantum features show that qRNNs are naturally suited for solving important problems in machine learning and quantum computing. Recently, qRNNs have been used to compress quantum circuits [23]. However, studies on using qRNNs for error correction in circuitlike quantum computing are warranted and are left for further studies.

3. *Quantum feature 3: Stochastic processes accessible to a qRNN*

We now explore how a qRNN can be used to stochastically evolve a probability distribution faster than any classical RNN. First, we note that if we initialize a RNN according to an initial distribution $p_0(\mathbf{s})$, the dynamics in Eq. (1) dictate that for $t > 0$, the RNN obeys a distribution given by the Markov-chain dynamics

$$p_t(\mathbf{s}) = \sum_{s'} P(\mathbf{s}|\mathbf{s}')p_{t-1}(\mathbf{s}'), \tag{8}$$

where $P(\mathbf{s}|\mathbf{s}')$ is the transition probability between states $\mathbf{s}'$ and $\mathbf{s}$, the particular value of which is given by Eq. (1) [28] (for details, see Appendix A).

Given this observation, we see that a RNN can be used for the task evolving a probability distribution p_0 into $p_f = L^{\text{targ}}p_0$ by a series of stochastic transition matrices $L^{\text{out}} = P^{t_f}$. The goal is to adjust the parameters of the RNN (i.e., biases and connection weights) to simulate the stochastic matrix encoded in $L^{\text{out}} \approx L^{\text{targ}}$ in as few steps as possible. Then, one may ask if a qRNN can do this more efficiently than any RNN.

We answer this in the affirmative. It is worth noting that not all stochastic transition matrices L^{targ} are embeddable in a Markov process (for a review of classical and quantum embeddability, see Appendix A). To simulate the future behavior of a stochastic system, information about its past must be stored and thus memory is a key resource. Quantum information processing promises a memory advantage for stochastic simulation [50]. In simulating stochastic evolution with classical resources there is a trade-off between the temporal and physical resources needed [51] and it has been shown that certain stochastic evolutions, when simulated with quantum hardware, may not suffer from such a trade-off, since the evolution arising from quantum Lindbladian dynamics is far more general than classical Markovian evolution [52]. That is, there exist matrices L^{targ} that are quantum embeddable but not classically embeddable. Moreover, even if L^{targ} is embeddable, the quantum evolution can lower the number of steps needed to produce L^{targ}, since the unitary dynamics of a quantum system allow a simultaneous, continuous, and coherent update of every neuron. This separation in capabilities illustrates the computational advantages of quantizing a RNN.

Let us now give an example of a matrix L^{targ} that can be achieved exponentially faster in a qRNN. Consider the task of realizing a transformation F corresponding to a global "spin flip"

$$F_{\mathbf{s}|\mathbf{s}'} = \begin{cases} 1, & \text{if } \forall_n\, s_n \neq s'_n, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The realization of F on N neurons using a classical Markov process requires several time steps of order $\mathcal{O}(2^{N-m})$, where m is the number of hidden neurons (for details, see Sec. III.A in Ref. [52]). In other words, a classical RNN cannot produce F efficiently when all available neurons must be flipped. This is a result of Eq. (1) and the fact that the flipping of neuron n is done by ensuring that there is another neuron m in the opposite state so that $J_{nm} > 0$ dominates h_n.

On the other hand, a qRNN can perform F in a single step regardless of whether all neurons need to flip. To see this, one can consider the case of Eq. (5) with $\Omega \gg h_n$. In this case, neurons both flip simultaneously and in a single time step under a unitary U. That is, if $|\psi_0\rangle = \sum_{\mathbf{s}} \sqrt{p_0(\mathbf{s})}|\mathbf{s}\rangle$, then

$$|\psi_f\rangle = U|\psi\rangle = \sum_{\mathbf{s}} \sqrt{p_f(\mathbf{s})}|\mathbf{s}\rangle. \tag{10}$$

Figure 3 compares the classical and quantum RNNs at realizing F. While the realization of the matrix F via Eq. (5) signals a quantum advantage, we highlight that this advantage is extremely sensitive to the decoherence arising from spontaneous emission (i.e., spontaneous relaxations from $|1\rangle$ to $|-1\rangle$), a main source of noise in NISQ devices (see Appendix A). It remains an open problem whether there exist stochastic processes enabled by Eq. (5) that are robust to noise and, in the future, we hope to explore how to shield unitary stochastic processes against noise in experimentally realizable NISQ devices.

The spin-flip process F is efficiently simulated using a classical computer. However, F exemplifies the ability of the qRNN to access stochastic processes that are inaccessible to classical RNNs without hidden neurons. This implies that if a RNN is employed to simulate evolving p_0 to p_{t_f} stochastically by passing it through several linear transformations, there are instances where the qRNN requires exponentially fewer steps. Stochastic simulation,

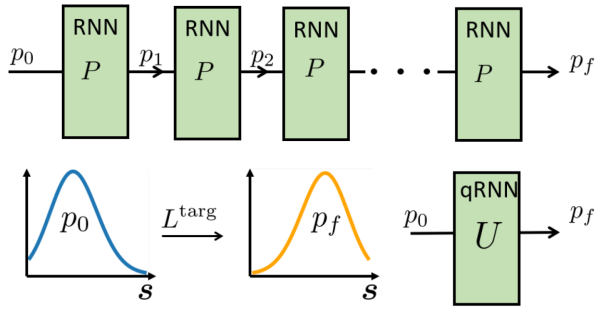

FIG. 3. The comparison of a classical and a quantum RNN to stochastically evolve a distribution p_{t_f} from an initial distribution p_0. Here, we consider $p_0(\mathbf{s}) = p_{t_f}(\mathbf{s}')$ when $s_n = -s'_n$ for all n. In this case, the RNN needs to produce a stochastic process matrix L^{targ} that flips all the spins through several time steps. The classical RNN (top) requires $\mathcal{O}(2^{N-m})$ time steps (i.e., applications of P) while using m hidden neurons. A qRNN (bottom) requires one time step and no hidden neurons.

of course, has applications in finance, biology, and ecology, among other fields. As an example, Ref. [53] has used this quantum advantage to propose a quantum circuit algorithm for stochastic process characterization and has presented applications in finance and correlated random walks. The above separation illustrates the computational advantage of quantizing a RNN.

B. qRNNs under spontaneous emission

Having seen how Eq. (5) recovers the discrete-update rule in Eq. (1), we now show that a qRNN under dissipation naturally evolves under continuous-time dynamics analogous to Eq. (3). This establishes only mathematical similarities between the evolution of NISQ devices and neural circuits, allowing us to use available quantum hardware for cognitive tasks, an idea that we explore further in Sec. V.

Consider the qRNN in Eq. (5) under spontaneous emission, where a spin relaxes from $|1\rangle$ to $|-1\rangle$ at a rate γ. To extract the dynamics of continuous variables, we focus on the dynamics of the expectation values of local Pauli operators.

The expectation value of an observable A is $\langle A\rangle = \text{Tr}(A\rho)$, where ρ is the density matrix describing the system. In particular, we focus on the expectations of the operators σ_n^x and $\sigma_n^y = i|-1\rangle\langle 1|_n - i|1\rangle\langle -1|_n$. If we start the qRNN at a state for which $\langle\sigma_n^z(0)\rangle = -1$, then (see Appendix B)

$$\dot{\langle\sigma_n^y\rangle} = -\frac{1}{\tau}\langle\sigma_n^y(t)\rangle - \frac{\Omega}{2\gamma}\sum_m J_{nm}\langle\sigma_n^x(t)\sigma_m^y(t)\rangle + \Delta_n(t)\langle\sigma_n^x(t)\rangle, \tag{11}$$

where we define the neural time scale $\tau^{-1} = \gamma/2 + \Omega^2/4\gamma$, which is different than that in Eq. (3) but bears the analogous significance of the time scale over which $\langle\sigma_n^y\rangle$ decays.

Different from Eq. (3), note that the dynamics of $\langle\sigma_n^y\rangle$ are influenced by the value of the spin along the x axis, a consequence of the nontrivial commutation relation of spin variables. The commutation relations also make Eq. (11) quadratic and therefore nonlinear. The quadratic term in Eq. (11) is analogous to the nonlinear term that gives RNNs their computational power.

In Appendix B, we also explore the dynamics of $\langle\sigma_n^x\rangle$ and show that together with $\langle\sigma_n^y\rangle$, we recover dynamics analogous to the integrate-and-fire RNN model [54], a more realistic model of neural networks in the brain than the one in Eq. (3).

IV. QUANTUM RESERVOIR COMPUTERS USING RYDBERG ATOMS: AN EXPERIMENTAL PROPOSAL

The similarities between Eq. (11) and the evolution of RNNs suggest the ability of qRNNs to emulate neurological learning. To explore neurological learning in qRNNs, we propose to fix the architecture of the qRNN coupling constants J_{nm} based on optical-tweezer arrays of Rydberg atoms.

The natural Hamiltonian of a Rydberg array closely resembles the one in Eq. (5). A Rydberg atom is a single valance-electron atom that can be coherently driven between an atomic ground state $|g\rangle$ and a highly excited state $|r\rangle$ with a much larger principal quantum number. These states can represent our $|-1\rangle$ and $|1\rangle$ neuronal states, respectively. A Rydberg atom in its excited state exhibits a large electronic dipole moment and, consequently, a collection of Rydberg atoms interacts via a $1/R^6$ van der Waals potential, where R denotes the physical distance between the two atoms. For an array of Rydberg atoms where the atoms are at fixed positions, the Hamiltonian of the system is [35]

$$H_{\text{Ryd}} = \Delta\sum_n \hat{n}_n + \frac{\Omega}{2}\sum_n \sigma_n^x + \sum_{nm}\frac{V}{R_{nm}^6}\hat{n}_n\hat{n}_m, \tag{12}$$

where $\hat{n}_n = |1\rangle\langle 1|_n$, Ω is the coherent Rabi drive coupling the $|-1\rangle$ and $|1\rangle$ states, $\Delta < 0$ is a global drive frequency mismatch to the atomic spacing of the atoms, and V is the nearest-neighbor interaction strength. Using acousto-optical deflectors (AODs) and a spatial light modulator (SLM), one can create spatially dependent light shifts, resulting in site- and time-dependent detunings $\Delta_n(t) = \Delta + \alpha(t)\Delta_n$, where $\alpha(t)$ is a time-dependent envelope. With this in mind, the Hamiltonian in Eq. (12) can be mapped to a Hamiltonian like that in Eq. (5) with $J_{nm} = V/R_{nm}^6$ since $\hat{n}_n = (\sigma_n^z + \mathbb{1}_n)/2$. In this paper, for concreteness, we compare our numerics against the experimental realization of Rydberg arrays in Ref. [35,38], where the

rates Ω, Δ_n, and V are all in units of megahertz, while the time constants are in units of microseconds. In these experiments, an off-resonance intermediate state, $|6P_{3/2}, F = 3, M_F = -3\rangle$, is used to couple $|g\rangle = |5S_{1/2}, F = 2, m_F = -2\rangle$ and $|r\rangle = |70S_{1/2}, m_J = -1/2, m_I = -3/2\rangle$ of ^{87}Rb atoms through a two-photon transition. Thus, photon scattering off the intermediate state is the dominant source of decoherence. As we show in Appendix D, we can model this with a modified spontaneous-emission process given by the jump operator

$$L^+ = \sqrt{\gamma}|g\rangle\,(\alpha\langle r| + \beta\langle g|) \tag{13}$$

instead of the typical $\sqrt{\gamma}|g\rangle\langle r|$ jump operator. In the above equation, $\gamma = 2\pi/(20\ \mu\text{s})$ and $(\alpha, \beta) = (0.05, 0.16)$ for the realistic settings that we simulate. With the full unitary and dissipative dynamics, we can think of an array of Rydberg atoms as a quantum analog of a continuous-time RNN. Figure 4 compares the architecture of a classical RNN in Fig. 4(a) and a Rydberg RNN in Figs. 4(b) and 4(c).

We note that the training of RNNs can be unstable, as it often relies on (truncated) back propagation through time or real-time recurrent learning. One way to circumvent this problem is by keeping the system parameters fixed. Instead, we focus on only training the output filter W^{out}. This easier training schedule has motivated the introduction of reservoir computers [55] and their quantum analogs [10,15–25,27]. Thus, in the following numerical experiments, we fix the position of the atoms in either a one-dimensional (1D) chain or a two-dimensional (2D) square lattice and train only W^{out} and some temporal parameters depending on the task. That is, in this paper, we implement Rydberg-reservoir computers. Logically, successful performance on the tasks presented here shows the computational ability of the qRNN sufficiently. While we include the effect of small imperfections on the positions of the atoms, we see no significant effect on the performance of the tasks after averaging our results over ten realizations of the atom positions. We leave full optimization of the qRNN for future work.

Lastly, several features of the many-body dynamics of arrays of Rydberg atoms are particularly well suited for emulating biological tasks. In Sec. V A, we show how Rydberg arrays can be used to implement inhibitory and excitatory neurons, which are vital in many biological tasks, such as multitasking [56]. The key idea behind the encoding of inhibitory neurons is to leverage positive and negative interactions between Rydberg atoms with different principal quantum numbers [45]. Additionally, in Sec. V D, we show that Rydberg arrays can store long-term memory by taking advantage of the weak ergodicity-breaking dynamics of quantum many-body scars [35,46,47].

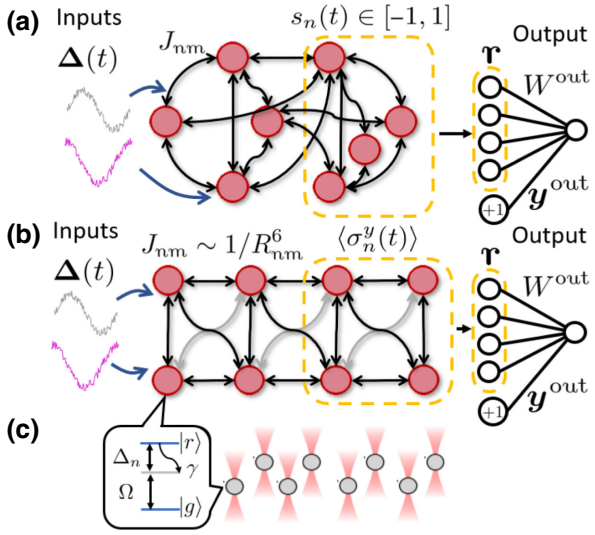

FIG. 4. The schematic picture of RNNs with classical and quantum neurons. (a) The classical RNN. The inputs are local biases and the interneural connections J_{nm} are arbitrary. A set of neurons is used for readout to produce the output $\mathbf{y}^{\text{out}} = W^{\text{out}}\mathbf{r}$. (b) A qRNN made from Rydberg atoms that restrict the connections to $J_{nm} \sim 1/R_{nm}^6$, where R_{nm} is the physical distance between atoms n and m. Here, we depict interactions between nearest and next-nearest neighbors. However, each neuron interacts with all others in the chain via $J_{nm} \sim 1/R_{nm}^6$. The local expectation values of a subset of atoms are for readout. (c) Arrays of Rydberg atoms as qRNNs. Each atom experiences a Rabi drive Ω and a local detuning Δ_n encoding the inputs of the RNNs. One of the main sources of decoherence in Rydberg atoms is spontaneous emission at a rate γ.

V. LEARNING BIOLOGICAL TASKS VIA RESERVOIR COMPUTERS

We focus on analyzing the potential of the Rydberg reservoir to learn biologically plausible tasks. In the tasks analyzed, we fix the geometry of the atoms depending on the task at hand. As a proof of principle, we focus on four simple neurological tasks that indicate good performance even with a small number of atoms. We show that a Rydberg reservoir can encode inhibitory and excitatory neurons that are vital for successful multitasking. Likewise, we show that Rydberg reservoirs can learn to decide by distinguishing properties of stimuli, have a working memory, and exhibit long-term memory enhanced by quantum many-body scars. Simulation details of each task can be found in Appendix D.

A. Multitasking

A hallmark of classical RNNs is their ability to multitask. Multitasking consists of simultaneously learning several output functions. Dale's principle defines an inhibitory neuron, indexed by n, as one with a negative sign in its

interactions with all other neurons [57]:

$$J_{nm} \leq 0 \quad \forall m. \tag{14}$$

Two Rydberg atoms with different principal quantum numbers n_Q and n'_Q and angular momentum quantum numbers that are the same can interact with a $1/r^6$ attractive potential V_{n_Q,n'_Q} [45]. Using the PYTHON package PairInteraction [58], we note that if n_Q represents the state $|r\rangle = |70S_{1/2}, m_j = -1/2, m_I = -3/2\rangle$, and n'_Q represents $|r'\rangle = |73S_{1/2}, m_j = -1/2, m_I = -3/2\rangle$, then the interaction $V_{n_Q,n_Q} = V \approx -V_{n_Q,n'_Q}$, where V is the strength between atoms with principal quantum numbers n_Q (see Appendix D). We can use this fact to encode inhibitory neurons. We restrict the concentration of n'_Q Rydberg atoms to be sparse, such that pairs of n'_Q atoms are placed as far as possible at a distance $d_{\max}$ apart in a 1D chain arrangement. We choose the field strength V so that $V/d_{\max}^6 = 10^{-2}$ and as a result we can neglect the interactions between pairs of n'_Q atoms but not the interactions between pairs $(n_Q)(n'_Q)$ and $(n_Q)(n_Q)$. This amounts to saying that if atom n is driven to n'_Q, then for all m, $J_{nm} \lesssim 0$ as in Eq. (14). By implementing this in on our reservoir, we can learn XOR, AND, and OR simultaneously for different concentrations of inhibitory neurons, as illustrated in Fig. 5(a).

Figure 5(b) shows the errors of simultaneously learning XOR, OR, and AND as a function of the system size N for a different number of inhibitory neurons in the array. The network is initialized in the state $|g\rangle^{\otimes N}$ and the network receives two binary inputs $x, y \in \{0, 1\}$ (in units of megahertz) for a time Δt (in units of microseconds) with input noise $\sigma_{\text{in}} = 0.1$. Afterward, the network is interrogated to give XOR(x, y), OR(x, y), and AND(x, y). W^{out} is trained using the loss in Eq. (4). The errors shown in Fig. 5(b) are the minimum achieved over a wide range of choices of interaction time $\Delta t \in [0, 5]$ μs. This shows that in some cases our reservoir can benefit from having a connectivity matrix J_{nm} with both positive (excitatory) and negative (inhibitory) values, analogously to the mammalian brain. For small system sizes, it seems that a ratio of 1:4 inhibitory neurons betters the learning performance, similar to the results in Ref. [56]. This is supported by the performance at four and eight neurons in Fig. 5(b). In particular, $N = 8$ neurons, two of which are inhibitory, result in a 40% decrease in the loss. Nonetheless, we observe that having no inhibitory neurons is best when dealing with $N = 6$ and 10 neurons. No inhibitory neurons are ever the worse choice. Figure 5(c)–5(e) shows the results of learning XOR, OR, and AND simultaneously using $N = 8$ and two inhibitory neurons. Note that the network is fully capable of classification errors well below the input-noise threshold σ_{in}.

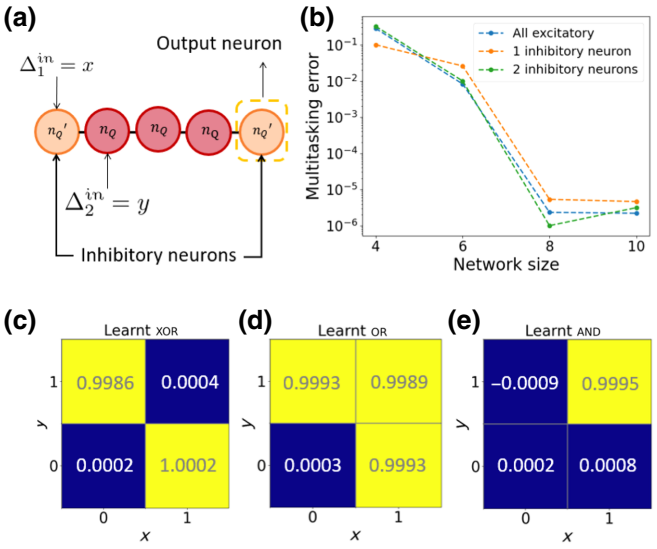

FIG. 5. The encoding of inhibitory neurons using Rydberg atoms and their use for multitasking, which consists of fixing the parameters of the qRNN and training W^{out} to produce three conflicting outputs. (a) The scheme for encoding inhibitory neurons. Rydberg atoms with different principal quantum numbers are used such that the pairs $(n_Q)(n'_Q)$ interact attractively while the pairs $(n'_Q)(n'_Q)$ and $(n_Q)(n_Q)$ interact repulsively. The network receives two binary inputs, x and y. (b) The square error for learning the functions XOR, OR, and AND on the inputs with different numbers of inhibitory neurons. Better performance is observed when one in every four neurons is inhibitory. (c)–(e) Examples of learned functions using eight neurons and two inhibitory neurons, which results in performing 40% better than without inhibitory neurons.

Lastly, while this task shows the success of the Rydberg reservoirs at approximating Boolean functions of the input, we note that one may also want to calculate different nonlinear functions of the input. We remark that our Rydberg reservoir can approximate biologically relevant nonlinear functions such as rectilinear function (ReLU) and sigmoid.

B. Decision making

One of the great successes of classical RNNs is their ability to integrate sensory stimuli to choose between two actions. Here, we present the Rydberg reservoir with a variant of the dot-motion decision-making task initially studied in monkeys, in which several inputs are analyzed to produce a scalar nonlinear function [59]. This function represents a decision. This task shows the ability of the Rydberg reservoir to produce nonlinear functions of the input and perform simple cognitive tasks, a feature of most reservoirs proposed thus far [60].

In this task, a reservoir is presented with two inputs, Δ_1^{in} and Δ_2^{in}, and the goal is to train the network to choose

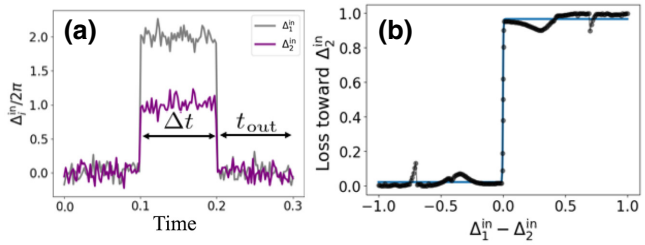

FIG. 6. The decision-making task using a Rydberg reservoir. (a) A schematic of the input stimuli as a pair of time-dependent detunings on two atoms. The stimuli are turned on for a normally distributed time Δt with standard deviation $\sigma_{in} = 0.1$. The network decides on a relaxation time t_{out} to output the decision sign $\left(\Delta_1^{in} - \Delta_2^{in}\right)$. (b) The psychometric response of the decision-making task, which maps the accuracy toward deciding that Δ_1^{in} is the largest as a function of the difference between the inputs. The simulated response (dotted) is well fitted by a sigmoid function (solid curve).

which input is the largest. That is,

$$y^{targ} = \text{sign}\left(\Delta_1^{in} - \Delta_2^{in}\right). \tag{15}$$

The stimuli, which in the case of a qRNN are local detunings on a pair of atoms, are turned on for a normally distributed time Δt with variance also $\sigma_{in} = 0.1$ and mean $\langle \Delta t \rangle = 0.1\ \mu$s [see Fig. 6(a)]. The stimuli are then turned off and the network chooses a relaxation time t_{out}, after which it "makes a decision" by approximating Eq. (15). This is known as the fixed-duration protocol, since the experimentalist fixes the stimulation period and the subject, the reservoir in this case, learns to choose a response time t_{out}.

In the brain, we expect the performance of a decision-making task to follow a sigmoidal psychometric response [44,59]. A psychometric response maps out the accuracy of a decision-making task as a function of stimuli distinguishability. As an example of a psychometric response, the reader could think about paying a routine visit to the eye doctor and having to discern the letters "b" and "p" written on the wall. If the letters are large enough, they become distinguishable; and if the letters are too small, one often fails to make out the right letter.

Classically, a decision-making task benefits from connectivity between all neurons. Since our connectivity is limited by physical constraints, a 2D square-lattice structure is chosen to prevent neurons from being isolated from the rest. Moreover, a 2D square lattice is experimentally friendly. We set up a Rydberg reservoir of 3×2 atoms with two input atoms and two different output atoms (for details, see Appendix D). The reservoir is then trained by optimizing over t_{out} and W^{out} such that the output of the reservoir approximates Eq. (15) while keeping the network parameters J_{nm}, Ω, and Δ_n^{in} fixed. We observe that $t_{out} \approx 1\ \mu$s is regularly obtained, as this is the time scale over which the information about $\Delta_{1,2}^{in}$ propagates through the network. In our case, $c_1 = \Delta_1^{in} - \Delta_2^{in}$ is a natural choice for a measure of stimuli distinguishability. Figure 6(b) shows the psychometric response of the task, which is qualitatively similar to those obtained in classical RNNs [44]. Moreover, we see in Fig. 6(b) that if $|c_1| \geq \sigma_{in}$ such that it is above the input noise level, our network achieves success more than 80% of the time. The success of this task shows the ability of the Rydberg reservoir to emulate simple cognitive tasks.

C. Parametric working memory

Our next neurological task is that of parametric working memory. One of the most important cognitive functions, working memory deals with the ability of the brain to retain and manipulate information for the later execution of a task. Here, we train a network to perform a task based on the decision-making task in Sec. V B but with two temporally separate stimuli [see Fig. 7(a)]. We use the fixed-time protocol, in which the separation between stimuli, denoted by t_{delay}, is fixed by us. The stimuli are both turned on for a time Δt and after the second input the network is left to relax for a time t_{out} before two output neurons are used to approximate Eq. (15). To avoid overfitting, we add Gaussian noise to the times Δt, t_{out} and t_{delay} with zero mean and standard deviation $\sigma_{in} = 0.1$. The network optimizes over W^{out}. Thus, the network has to retain information about Δ_1^{in} for a few "seconds" to then compare against Δ_2^{in} and make a decision.

We set a Rydberg reservoir of 3×2 atoms with two input atoms and two different output atoms (for details, see Appendix D). Figure 7(b) shows the loss of the network as a function of the total time for which the inputs are injected into the network ($\tau = 2\Delta t + t_{delay}$). We note that the loss function is high for small τ, since it takes the input neurons to correlate with the rest of the reservoir. Accordingly, in Fig. 7(b), we show that growth of the entanglement entropy of the input qubits accompanies a decrease in the loss function. For Fig. 7(c), we fix $t_{out} = 0.1$, a choice that has little effect on the performance of the reservoir.

In Fig. 7(c), we show the accuracy of the reservoir at reproducing Eq. (15) as a function of the time for which the inputs are turned on (Δt) and for different choices of t_{out}. For these plots, $t_{delay} = 0.1$ is fixed. We note that the accuracy is largely invariant to our sampled choices of t_{out}.

Lastly, in Fig. 7(d), we probe the accuracy of the reservoir as a function of t_{delay}. For these experiments, we fix $t_{out} = 0.5$ and $\Delta t = 0.15$. Importantly, we set $V = 2\pi \times 10$ MHz and $\Omega = 2\pi \times 4.2$ MHz such that $V > \Omega$ and neighboring Rydberg excitations are off resonance, putting our reservoir into the so-called *blockaded regime* [61,62]. While one initially might expect the accuracy to decrease for increasing t_{delay}, we find that this is not the case and that, instead, the accuracy oscillates persistently, reaching

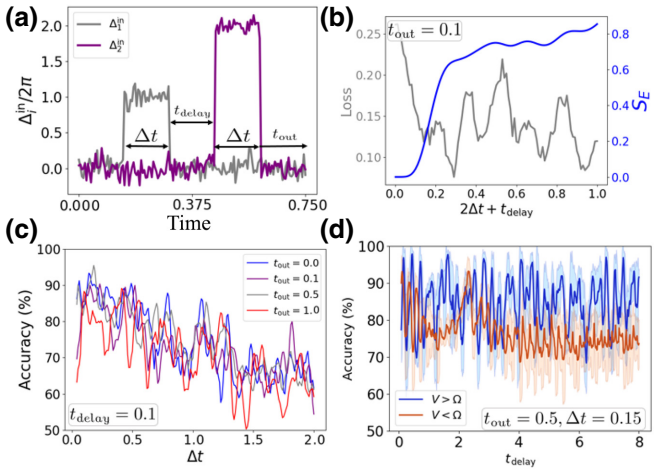

FIG. 7. The working memory of a Rydberg quantum reservoir computer. (a) A schematic of the inputs of the network, where two atoms are detuned for a time Δt but temporally separated by a time t_{delay}. Two different output neurons are used for readout at a time t_{out} after the second input is turned off. (b) The loss of the working memory task as a function of the total input time $2\Delta t + t_{\text{delay}}$ (gray). The entanglement entropy between the input qubits and the rest of the reservoir as a function of $2\Delta t + t_{\text{delay}}$ (blue). Here, the mean value of t_{out} is 0.1. The loss stays large for small input times until the input qubits start to entangle with the rest of the reservoir. (c) The accuracy as a function of the time for which the input is turned on (Δt) for four different choices of t_{out} and with fixed $t_{\text{delay}} = 0.1$. These curves show that the accuracy is largely independent of t_{out} and Δt as long as $\Delta t < 0.3$. (d) The accuracy of the working-memory task at $\Delta t = 0.15$ and $t_{\text{out}} = 0.5$ as a function of t_{delay}. The blue curve is the performance when $V > \Omega$ puts the reservoir in the Rydberg-blockaded regime, while the red curve is the performance when $V < \Omega$ puts the reservoir in the disordered regime. These plots show that when $V > \Omega$, the Rydberg reservoir can hold memory for later manipulation better than when $V < \Omega$. The shaded regions indicate error bars.

high accuracies as shown in the blue curve in Fig. 7(d). Interestingly, this behavior disappears when the coupling $V = 2\pi \times 0.1$ MHz such that $V < \Omega$, as shown in the red curve in Fig. 7(d), although the performance is statistically significant even for long t_{delay}, with an accuracy greater than 50%. We can conclude that, in the blockaded regime, the reservoir can hold information for longer periods. We can understand this dependence on V/Ω as follows. In the disordered regime, the atoms are mostly uncorrelated and are allowed to oscillate freely, with the dynamics being dominated by the drive Ω. Thus, after a short time, the inputs coming through a z field are largely irrelevant and the network is unable to hold the information about the first input. On the other hand, when $V > \Omega$, the atoms are largely correlated, since neighboring excitations of Rydberg atoms are blockaded and the dynamics are slowed down. These slow dynamics in the system allow for longer memory times. In Sec. V D, we explore the longer-term memory in the blockaded regime and show that the long-term memory in a reservoir can be stabilized due to the presence of quantum many-body scars.

D. Long-term memory via quantum many-body scars

Finally, we turn to examine the ability of a reservoir to encode long-term memory. The task consists of encoding a classical bit m in the initial state of a reservoir $|\psi_m(0)\rangle$ so that after the system is left to evolve under its inherent dynamics for a time T, local measurements of the state $|\psi_m(T)\rangle$ are used to recover m. However, m cannot be recovered from local measurements if the dynamics obey the eigenstate-thermalization hypothesis (ETH) [63]. Instead, local measurements of $|\psi_m(T)\rangle$ obey thermal statistics described by the energy spectrum of the Hamiltonian and carry no information on the initial condition $|\psi_m(0)\rangle$. Thus, reservoirs that violate the ETH are naturally suited for memory tasks, since they can locally retain information about their initial state. Indeed, this notion has begun to be studied in quantum reservoirs [25,27]. Recent experiments using quench dynamics in arrays of Rydberg atoms have revealed quantum many-body scarring behavior [35], which can be stabilized [46,47] to delay the thermalization of the system. Here, we use these results to enlarge the memory lifetime of a reservoir. The simulation details are found in Appendix D.

In the case of a kicked ring of Rydberg atoms experiencing nearest-neighbor blockade, the dynamics are captured in the so-called PXP model [35,47,64,65],

$$H(t) = H_{\text{PXP}} + \hat{N} \sum_{k\in\mathbb{Z}} \theta_k \delta(t - k\tau), \tag{16}$$

$$H_{\text{PXP}} = \Omega \sum_{n=1}^{N} P_{n-1}\sigma_n^x P_{n+1} \quad \hat{N} = \sum_n \hat{n}_n, \tag{17}$$

where $P_n = |g\rangle\langle g|_n$ projects the atom at the n^{th} site onto the ground state and we choose periodic boundary conditions to mitigate edge effects. In Eq. (16), we let $\theta_k = \pi + \epsilon_k$, where ϵ_k is a Gaussian random variable with mean ϵ and variance σ_{in}^2. That is, ϵ_k plays the role of added noise in the reservoir. For this discussion, we let $\gamma = 0$, since we know from experiments that the quantum scarring behavior is robust to the decoherence of the atom and the choice to work with the Hamiltonian evolution helps speed up the acquisition of numerical data.

We denote $\chi_\tau = \exp(-i\pi\hat{N}) \exp(-i\tau H_{\text{PXP}})$. It has been empirically observed that χ_τ approximately exchanges the Néel states $|AF\rangle = |1010\cdots\rangle$ and $|AF'\rangle = |0101\cdots\rangle$ for $\tau \approx 1.51\,\pi\Omega^{-1}$ [35]. Note that $\chi_\tau\chi_\tau = \mathbb{1}$ and so under no noise, any state $|\psi\rangle$ is recovered after a cycle of evolution of 2τ. However, the noise ϵ_k destroys the revival of all initial states except for $|AF\rangle$ and $|AF'\rangle$ (see Appendix C). This leads to many-body quantum scars stabilized by the operator $\exp(-i\pi\hat{N})$ [46,47].

Given the dynamics in Eq. (16), we propose the following scheme for encoding a binary memory $m \in \{0, 1\}$. We choose a reference state $|\psi\rangle$ and let $|\psi_0(0)\rangle = |\psi\rangle$ and $|\psi_1(0)\rangle = \chi_\tau|\psi\rangle$. Subsequently, the state $|\psi_m(0)\rangle$ is left to evolve for n cycles of duration $2\tau = 2(1.51\pi)$, after which the populations $\mathbf{r}_m(n) = (P_g(2n\tau|m), P_r(2n\tau|m))$ of the single-atom reduced density matrix are used to retrieve m. The retrieval is done by training a vector W_n^{out} on M instances of $\mathbf{r}_m(n)$ in order to minimize Eq. (4), where $\mathbf{y}^{\text{targ}} = \mathbf{m}$ is the binary vector of memories and $\mathbf{y}^{\text{out}}(n) = W_n^{\text{out}}\mathbf{r}(n)$ is the output of our network after n cycles. Figure 8 shows the encoding of the memory scheme. Note that we expect the memory to be retrievable if the system avoids thermalization.

To quantify the quality of the memory retrieval $R(n)$, we use the squared Pearson's r factor

$$R(n) = \frac{\text{cov}^2(\mathbf{m}, \mathbf{y}^{\text{out}}(n))}{\sigma^2(\mathbf{m})\sigma^2(\mathbf{y}^{\text{out}}(n))}. \tag{18}$$

Figure 9(a) shows the memory-retrieval error as a function of the number of cycles for three different choices of reference state. Figure 9(b) shows the average entanglement entropy ($\bar{S}_E$) of the leftmost atom in the ring. The saturation of $\bar{S}_E$ signals growth in the memory-retrieval error as

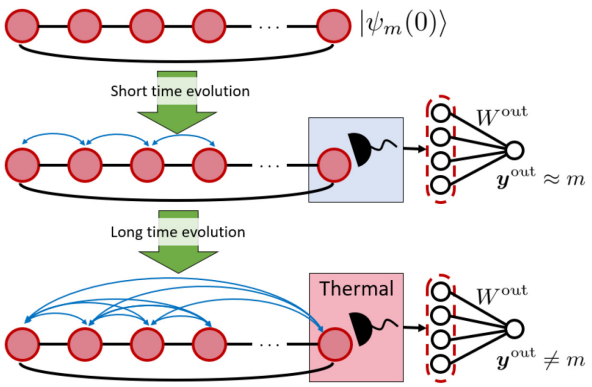

FIG. 8. A state encoding a memory m is prepared. The state evolves under its natural Hamiltonian before being interrogated via local measurements to retrieve m. If the evolution time is short, the system is still out of equilibrium and remembers its initial condition. Thus, m can be retrieved. On the other hand, after a long time, the system may thermalize and local measurements may fail to provide information about the initial state. Thus, the memory-retrieval time is upper bounded by the thermalization time of the initial state $|\psi_m(0)\rangle$ under the dynamics of the system. In the example in Sec. V D, the system is a chain of Rydberg atoms and final measurements are performed on a single atom, which is then linearly postprocessed to retrieve m. In this case, a thermal state can be observed by measuring whether the entanglement entropy of the region obeys a volume law. If the dynamics can be stabilized against thermalization, the memory can be retrieved at larger times.

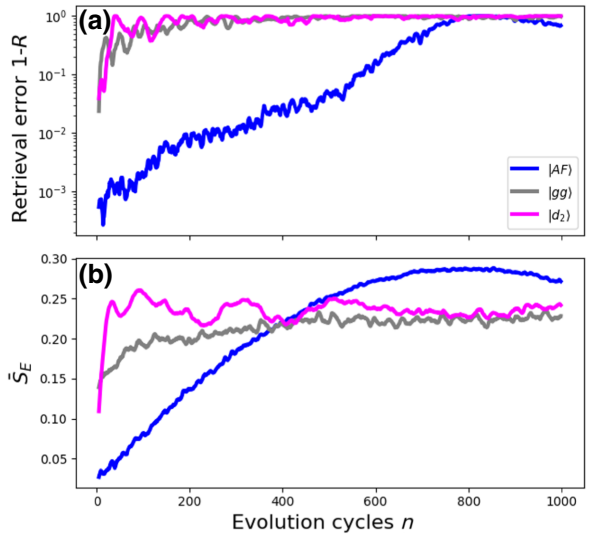

FIG. 9. The dependence of memory retrieval on different reference states. We use a ring of $N = 8$ Rydberg atoms with $\epsilon = \sigma = 0.1$ and $M = 100$ and 30 samples for the training and testing sets, respectively. The memories are sampled from a balanced Bernoulli distribution. (a) The memory-retrieval error for three different choices of reference state, $|AF\rangle = |grgrgrgr\rangle$, $|gg\rangle = |gg\ldots g\rangle$, and $|d_2\rangle = |grggggrg\rangle$. Due to the scarring behavior of $|AF\rangle$, the memory length is greatly improved. (b) The entanglement entropy of the leftmost atom, averaged over the M memory instances ($\bar{S}_E$). The saturation of $\bar{S}_E$ signals the thermalization of the system and thus a decrease in R.

the state "forgets" the initial condition. From other studies, we see that memory is retrieved at longer times due to the slow thermalization of the Néel states due to quantum many-body scars [35,46,47,64,65]. The time-crystalline nature of the reservoir using $|\psi\rangle = |AF\rangle$ signals long-time correlations and thus the reservoir can be used to encode and predict series with long-time correlations [17].

The Néel states exhibit long-term memory due to the scarring behavior of the evolution. This can be understood by analyzing the average evolution produced by a single cycle. Up to second order in ϵ_k, the state at time $2\tau n$, $\rho(n)$, evolves to the state at time $2\tau(n+1)$, $\rho(n+1)$, where (see Appendix C)

$$\begin{aligned}\rho(n+1) = \rho(n) - i\epsilon[H^+, \rho(n)] + \sigma_{\text{in}}^2 \\ \times \left(H^+\rho(n)H^+ - \frac{1}{2}\{H^+H^+, \rho(n)\}\right) \\ + \sigma_{\text{in}}^2\left(H^-\rho(n)H^- - \frac{1}{2}\{H^-H^-, \rho(n)\}\right).\end{aligned} \tag{19}$$

Here, $H^\pm = \hat{N} \pm \chi_\tau\hat{N}\chi_\tau$ are Hermitian operators. We can rewrite Eq. (19) as $\rho(n+1) = \rho(n) + \mathcal{L}_{\epsilon,\sigma}(\rho(n))$. Since $[H^+, \chi_\tau] = 0$, the operator H^+ has an emergent $\mathbb{Z}_2$ symmetry, which means that the ground states of H^+ are well

approximated by the states $|\pm\rangle = \frac{1}{\sqrt{2}}\left(|AF\rangle \pm |AF'\rangle\right)$ [47]. Note that

$$H^+|+\rangle \approx N|+\rangle, \quad H^-|+\rangle \approx 0, \tag{20}$$

$$H^+|-\rangle \approx N|+\rangle, \quad H^-|-\rangle \approx 0, \tag{21}$$

where N is the system size. We conclude that if $\rho(n) = |AF\rangle\langle AF|$, then $\rho(n+1) \approx \rho(n)$ as this state is (approximately) in the kernel of $\mathcal{L}_{\epsilon,\sigma}$. Therefore, the Néel states are suitable memory states.

Equation (19) also tells us that any density matrix in the kernel of $\mathcal{L}_{\epsilon,\sigma}$ may also serve as a memory state, since it is a steady state of the evolution. This would allow us to enlarge the number of memories accessible in a qRNN. In Appendix C, we show the existence of a large number of steady states and we present a scheme to prepare a number of them. It is worth noting, however, that these memories may have to be distinguished from one another via global measurements. The questions of how to efficiently prepare and distinguish these memory states remain, importantly, both open and key in telling us if a memory quantum advantage can be claimed in qRNNs. As it stands, the use of quantum scars signals that Rydberg-inspired RNNs may present enhanced memory since quantum scars are classically simulatable due to their low entanglement entropy. However, it is unclear whether the system can be classically simulated at late times due to the onset of the thermalization. These questions are left for future studies.

Quite recently, another proposal to enlarge the number of memories accessible in a quantum reservoir has been introduced using the emergent scale-free network dynamics of a melting discrete time crystal in an Ising chain [25]. The proposal in Ref. [25] can be seen as a generalization of the quantum reservoir presented in Eq. (16) by dropping the constraint of the Rydberg blockade. Our results, as well as those in Ref. [25], pose the possibility of having a RNN with a memory capacity that outpaces that of classical RNNs such as the Hopfield network [66].

VI. CONCLUSIONS AND OUTLOOK

In this paper, we present a quantum extension of a classical RNN on binary neurons. This implies a deep connection between controllable many-body quantum systems and brain-inspired computational models. Our qRNN facilitates the ability to employ the analog dynamics of quantum systems for computation instead of the circuit-based paradigm. We show how features of the quantum evolution of our qRNN can be used for quantum learning tasks and to speed up the simulation of stochastic dynamics. We implement a quantum reservoir using arrays of Rydberg atoms and show how Rydberg atoms analogously perform biological tasks even in the presence of a few atoms. This can be explained via the physics of the system. For example, we show how the weak ergodicity-breaking collective dynamics in Rydberg atoms can be employed for long-term memory.

While this paper takes the first step forward in connecting controllable quantum systems and neural networks from a fundamental perspective, several questions remain unanswered. First, from the first two quantum features presented in this paper, studies of how qRNNs can be used for quantum error correction in circuitlike quantum computing are warranted. Directly from this work, investigations into advantageous stochastic processes in qRNNs that are robust to decoherence are enticing. These advantages will likely emerge from the collective behavior of quantum neurons. Therefore, the field will soon require a thorough understanding of the collective dissipative dynamics of neurons in qRNNs, which would also shed light on rigorous studies of the computational power of these architectures. Guided by the fact that neural networks become universal approximators by interconnecting many neurons, one may also consider the spatial and control requirements necessary for universal brain-inspired quantum machine learning.

Given the vast number of classical computational models for the brain, there are several immediate research directions. One of these is the exploration of a systematic way to quantize more biologically realistic models of a neural circuit. A possible starting point for translating different neural circuits would be to exploit key engineering and fundamental features of different NISQ platforms. For example, recent experiments using Rydberg atoms in photonic cavities may provide us with the ability to capture neural plasticity on qRNNs by arbitrarily tuning the interneural interactions [67]. Likewise, superconducting circuits have lately been used to encode biologically realistic single-neuron models [13]. Along these explorations, it will be imperative to establish a variety of methods to analyze how quantum neural networks recover the classical protocols within certain limits, as well as the source and extent of the quantum advantages that each platform can offer.

Lastly, while our memory encoding scheme in Sec. V D offers a possibility to encode a binary memory, whether a higher number of memories can be encoded efficiently remains an important open question. In Appendix C, we offer a proposal based on the steady states of the effective dissipative evolution in the prethermalization regime introduced by the noise in the qRNN. This already shows a theoretical number of memories greater than those attainable by the vanilla Hopfield network [66]. However, distinguishing these memories, or producing Hamiltonians with the desired memory state in mind, is left for future research. It is clear, however, that memory in a quantum reservoir relies on ergodicity-breaking dynamics [25,27]. Hamiltonian engineering techniques, together with more

general driven Hamiltonians such as those in Ref. [25], may pave the way toward programmable memories in a qRNN.

ACKNOWLEDGMENTS

We thank Mikhail D. Lukin and Nishad Maskara for insightful discussions. R.A.B. acknowledges support from the National Science Foundation (NSF) Graduate Research Fellowship under Grant No. DGE1745303. X.G. acknowledges support from Harvard-MPQ Center for Quantum Optics, the Templeton Religion Trust (TRT) under Grant No. 0159, and the Army Research Office under Grant No. W911NF1910302 and the Multidisciplinary University Research Initiative (MURI) under Grant No. W911NF-20-1-0082. S.F.Y. acknowledges funding from National Science Foundation (NSF) and the Air Force Office Scientific Research (AFOSR).

APPENDIX A: PROBABILITY TRANSFORMATIONS USING qRNNs

In the case of the RNN presented in Eq. (1), using Ref. [28] we can derive that $P(\mathbf{s}|\mathbf{s}')$ in Eq. (8) is given by

$$P(\mathbf{s}|\mathbf{s}') = \prod_{i=1}^{N} \frac{1}{2}\left(1 + s_i g\left[h_i(\mathbf{s}')/\sigma_{\text{in}}^2 - 1\right]\right),$$

where $g[x] = \text{Erf}\left[z/\sqrt{2}\right]$ is the error function due to the Gaussian noise. Regarding the task in Sec. III A 3 of flipping all neurons at once, one could naively think that this can be done classically by taking the inputs $\Delta_n \to \infty$, however, since the strength σ_{in}^2 of the noise scales as the size of the inputs, one obtains $P(\mathbf{s}|\mathbf{s}') \to \prod_{i=1}^{N} \frac{1}{2}(1 + s_i/2)$, which is a completely random update, independent of the original state.

A transition matrix L obeys $L_{\mathbf{s}'|\mathbf{s}} \geq 0$ and $\sum_{\mathbf{s}'} L_{\mathbf{s}'|\mathbf{s}} = 1$. L is said to be *classically embeddable* if it can be generated by a continuous Markov process via

$$\frac{d}{dt}P(t) = K(t)P(t), \quad P(0) = \mathbb{1}, \quad P(t_f) = L, \tag{A1}$$

where K is called a generator matrix, which preserves the positive nature of P via the constraint $K_{\mathbf{s}|\mathbf{s}'} \geq 0$ for $\mathbf{s} \neq \mathbf{s}'$ and normalization via the constraint $\sum_{\mathbf{s}} K_{\mathbf{s}|\mathbf{s}'} = 0$. Applied to our setup, a classically embeddable stochastic process is one that can transform $p_{t_f} = Lp_0$ via a RNN without employing any hidden neurons (i.e., $M = N$ neurons are used for readout) and in a single step. In general, determining whether a matrix L is embeddable is an open question but any embeddable matrix must necessarily satisfy [68]

$$\prod_{\mathbf{s}} L_{\mathbf{s}|\mathbf{s}} \geq \det L \geq 0. \tag{A2}$$

From Eq. (A2), it immediately follows that the global "spin-flip" matrix F defined in Eq. (9) is not classically embeddable. That is, $\det F = 1$ and $\prod_{\mathbf{s}} F_{\mathbf{s}|\mathbf{s}} = 0$, violating Eq. (A2). Note that the impossibility of performing F without hidden neurons is quite general and is not limited to the stochastic process allowed by Eq. (1). Moreover, the number of time steps needed to achieve F using m hidden neurons is of order $\mathcal{O}(2^{N-m})$ (for details, see Sec. III.A in Ref. [52]).

Similar definitions of embeddability exist in the quantum setting. A stochastic process L is said to be *quantum embeddable* if there exists a Markovian quantum channel $\mathcal{E}$ such that

$$L_{\mathbf{s}'|\mathbf{s}} = \langle \mathbf{s}'|\mathcal{E}(|\mathbf{s}\rangle\langle\mathbf{s}|)|\mathbf{s}'\rangle. \tag{A3}$$

A Markovian quantum channel $\mathcal{E}$ is a channel arising from the time evolution under a master equation and thus $\mathcal{E}$ may include unitary and dissipative terms. Moreover, permutations such as F in Eq. (9) are quantum embeddable, since all permutations are unitary operators.

We highlight that realizing F is extremely sensitive to the decoherence arising from spontaneous emission, a main source of noise in NISQ devices. If γ is the decay rate at which spin $|1\rangle$ relaxes to $|-1\rangle$, one can show that the unitary evolution leads to the stochastic process F^{γ}, where $\det F^{\gamma} = e^{-\mathcal{O}(2^N)}$. Note that whether F^{γ} violates Eq. (A2) becomes rapidly inconclusive with increasing system size.

APPENDIX B: CONTINUOUS-TIME DYNAMICS FOR A qRNN

One successful neural-circuit model is the integrate-and-fire RNN (IF RNN). In an IF RNN, each of the N neurons is influenced by presynaptic firing rates and produces a postsynaptic firing rate as an output. Each neuron is endowed with a firing rate $s_n(t)$, where n denotes the n^{th} neuron. The presynaptic firing rates arriving at the n^{th} neuron are integrated to produce a presynaptic current $I_n(t)$. In turn, the neuron produces a firing rate s_n influenced by its current and the firings of other neurons. Additionally, each neuron can receive a temporal input stimulus $\Delta_n^{\text{in}}(t)$ that affects both the currents and the firing rates. In general, the firing rates and currents are described by nonlinear coupled differential equations of the form

$$\dot{I}_n = -\tau_I^{-1} I_n + G_n(\mathbf{s}(t), \mathbf{I}(t), J_{nm}, \Delta^{\text{in}}(t)), \tag{B1}$$

$$\dot{s}_n = -\tau_s^{-1} s_n + F_n(\mathbf{s}(t), \mathbf{I}(t), J_{nm}, \Delta^{\text{in}}(t)), \tag{B2}$$

where the $\tau_{I,r}$ are relaxation time constants for the currents and firing rates, respectively. The vector $\mathbf{s}(t)$ is defined as $\mathbf{s}(t) = (s_1(t), \ldots, s_N(t))$, with $\mathbf{I}(t)$ and $\Delta^{\text{in}}(t)$ defined analogously. The functions G and F ensure that the dynamics are nonlinear, which gives RNNs their vast computational complexity. The specific forms of G and F depend on the application and relation between the currents and the firing rates one is trying to capture by the model.

The qRNN in Sec. III B follows the Heisenberg-Langevine equations of motion

$$\dot{A} = i[H, A] + \sum_n \left(\frac{\gamma}{2}\sigma_n^+ + f_n^\dagger\right)[A, \sigma_n^-] + \sum_n [A, \sigma_n^+]\left(\frac{\gamma}{2}\sigma_n^- + f_n\right), \tag{B3}$$

for any operator A. In Eq. (B3), $\sigma_n^+ = |1\rangle\langle -1|_n$, $\sigma_n^+ = (\sigma_n^+)^\dagger$, and f_n is a Langevin noise operator with Gaussian statistics $\langle f_n(t)\rangle = 0$ and $\left\langle f_n(t) f_m^\dagger(t')\right\rangle \propto \delta_{mn}\delta(t - t')$. In the above equation, $[A, B] \equiv AB - BC$ denotes the commutator between matrices A and B.

To extract the statistics of the system, one may choose to look at the dynamics of the expectation values of two different local observables. For example, the equations of motion for expectations of the local Pauli operators $\sigma_n^x = |-1\rangle\langle 1|_n + |1\rangle\langle -1|_n$ and $\sigma_n^y = i|-1\rangle\langle 1|_n - i|1\rangle\langle -1|_n$ are given by

$$\dot{\langle\sigma_n^x\rangle} = -\frac{\gamma}{2}\langle\sigma_n^x\rangle + i\langle[H(t), \sigma_n^x]\rangle, \tag{B4}$$

$$\dot{\langle\sigma_n^y\rangle} = -\frac{\gamma}{2}\langle\sigma_n^y\rangle + i\langle[H(t), \sigma_n^y]\rangle, \tag{B5}$$

with $H(t)$ specified by Eq. (5). The expectation values are calculated in the quantum mechanical sense such that for an operator A, $\langle A\rangle = \mathrm{Tr}(A\rho)$ and terms linear in f_n cancel out. Note that the commutators in Eqs. (B4) and (B5) play the role of the functions G and F in Eqs. (B1) and (B2).

For σ_n^z, Eq. (B3) gives

$$\dot{\sigma_n^z} = -\gamma/2\sigma_n^z - \frac{\Omega}{2}\sigma_n^y + \gamma\mathbb{1}/2 - 2f_n^\dagger\sigma_n^-. \tag{B6}$$

This can be integrated out to give

$$\sigma_n^z(t) - \sigma_n^z(0) = \int_0^t dt' e^{-\gamma/2(t-t')}\left(-\frac{\Omega}{2}\sigma_n^y(t') - 2f_n^\dagger(t')\sigma_n^-(t') + \gamma/2\mathbb{1}\right). \tag{B7}$$

We choose to start the network at $\langle\sigma_n^z\rangle = -1$, for all n. We substitute this back into Eq. (B5) and we take the expectation values to eliminate terms linear in f_n. We obtain

$$\dot{\langle\sigma_n^y\rangle} = -\frac{\gamma}{2}\langle\sigma_n^y\rangle + \frac{\Omega}{2}\sum_{m=1}^N J_{nm}\int_0^t dt' e^{-\gamma(t-t')}\langle\sigma_n^x(t)\sigma_m^y(t')\rangle + \Delta_n(t)\langle\sigma_n^x\rangle - \frac{\Omega^2}{4}\int_0^t \langle\sigma_n^y(t')\rangle e^{-\gamma(t-t')}dt'. \tag{B8}$$

Similar equations can be found for $\langle\sigma_n^x\rangle$. Equation (B8) tells us that $\langle\sigma_n^y\rangle$ depends on past statistics and thus that our network has a memory time bounded by $1/\gamma$. Let J denote the matrix J_{nm}. For $\gamma t \gg 1$, we can extend the lower bound of integration to $-\infty$. Using the approximation $\int_{-\infty}^t e^{-\gamma(t-t')}f(t')dt' \approx -\gamma^{-1}f(t)$, we obtain

$$\dot{\langle\sigma_n^x\rangle} = -\frac{\gamma}{2}\langle\sigma_n^x\rangle - \Delta_n^{\text{in}}\langle\sigma_n^y\rangle - \frac{\Omega}{2\gamma}\sum_m J_{nm}\langle\sigma_n^y\sigma_m^y\rangle, \tag{B9}$$

$$\dot{\langle\sigma_n^y\rangle} = -\left(\frac{\gamma}{2} + \frac{\Omega^2}{4\gamma}\right)\langle\sigma_n^y\rangle + \Delta_n^{\text{in}}\langle\sigma_n^x\rangle - \frac{\Omega}{2\gamma}\sum_m J_{nm}\langle\sigma_n^x\sigma_m^y\rangle, \tag{B10}$$

thus leading to Eq. (11). In Eqs. (B10) and (B9), the time dependence is implied.

Let us now define $s_n(t) \equiv \langle\sigma_n^y(t)\rangle$ and $I_n(t) \equiv \langle\sigma_n^x(t)\rangle$ so that $\mathbf{s}(t) = (s_1(t), \ldots, s_N(t))$ and $\mathbf{I}(t) = (I_1(t), \ldots, I_N(t))$. We see that Eqs. (B10) and (B9) match Eqs. (B1) and (B2), where

$$G_n = -\Delta_n s_n - \frac{\Omega}{2\gamma}\sum_m J_{nm}\langle I_n I_m\rangle, \quad \tau_I^{-1} = \frac{\gamma}{2}, \tag{B11}$$

$$F_n = \Delta_n I_n - \frac{\Omega}{2\gamma}\sum_m J_{nm}\langle I_n s_m\rangle, \quad \tau_s^{-1} = \frac{\gamma}{2} + \frac{\Omega^2}{4\gamma}. \tag{B12}$$

Equations (B9) and (B10) allow us to naturally interpret $\langle\sigma_n^y\rangle$ as the firing rate of the n^{th} neuron and $\langle\sigma_n^x\rangle$ as the current. That is, the rate of the presynaptic neuron $\langle\sigma_k^y\rangle$ amounts to a current in the postsynaptic neuron $\langle\sigma_n^x\rangle$ that drives its rate $\langle\sigma_n^y\rangle$.

Equations (B9) and (B10) comprise a system of coupled quadratic differential equations, where the quadratic terms arise from the nontrivial commutation relation of the Pauli operators $[\sigma_n^\alpha, \sigma_m^\beta] = i\delta_{\alpha\beta}\epsilon_{\alpha\beta\gamma}\sigma_n^\gamma$, where $\epsilon_{\alpha\beta\gamma}$ is the Levi-Civita symbol. These quadratic terms in Eqs. (B9) and (B10) make a qRNN a powerful computational system in a similar way to how the functions G and F make a RNN a powerful computational system.

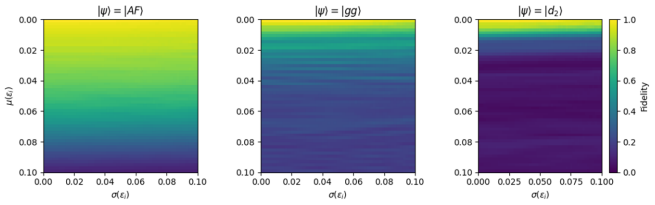

FIG. 10. Fidelities with the initial state after evolving for $n = 100$ cycles of noisy kicked dynamics. The fidelity is defined as $F \equiv |\langle\psi|\psi(2n\tau)\rangle|^2$, where $|\psi\rangle$ is the initial state. Here, we use $L = 8$ Rydberg atoms and define $|AF\rangle = |grgrgr\rangle$, $|gg\rangle = |gggggggg\rangle$, and $|d_2\rangle = |grgggggrg\rangle$. The Néel state $|AF\rangle$ is robust to the noise in the drive, since this state is invariant to decoherence up to second order in ϵ_i.

APPENDIX C: MEMORY AND QUANTUM MANY-BODY SCARS

As described in the main text and discussed more thoroughly in Ref. [47], the scarring behavior of the kicked PXP model is robust to fixed imperfections in the drive. The robustness persists even for random noise. Figure 10 exemplifies the overlap with the initial condition for a noisy kicked PXP model for different values of ϵ and σ_{in}, which is a natural extension of the model in Ref. [47]. The Néel state $|AF\rangle$ exhibits robust revivals invariant of σ_{in}^2. This fact can be explained with the effective theory presented below.

To understand the robustness of the quantum scarring behavior in the Rydberg reservoir, it is instructive to seek an effective description of the evolution of the system. Recall that a cycle is defined as two imperfect applications of χ_τ. The Hamiltonian in Eq. (16) produces the single-cycle unitary

$$U_\tau(\epsilon_1,\epsilon_2) = e^{-i\epsilon_2\hat{N}}\chi_\tau e^{-i\epsilon_1\hat{N}}\chi_\tau = e^{-i\epsilon_2\hat{N}}e^{-i\epsilon_1\chi_\tau\hat{N}\chi_\tau}, \quad \text{(C1)}$$

where we use the fact that χ_τ is both Hermitian and unitary. Using the Baker-Campbell-Hausdorf formula to second order in ϵ_i, we can rewrite Eq. (C1) as

$$U_\tau(\epsilon_1,\epsilon_2) \approx e^{-i(\epsilon_2\hat{N}+\epsilon_1\chi_\tau\hat{N}\chi_\tau)}. \quad \text{(C2)}$$

A state $\rho(n)$ evolves to $\rho(n+1) = U_\tau(\epsilon_1,\epsilon_2)\rho(n)U_\tau^\dagger(\epsilon_1,\epsilon_2)$ after a cycle. Expanding this to second order in ϵ_k and using the fact that $\langle\epsilon_k\rangle = \epsilon$ and $\langle\epsilon_k\epsilon_l\rangle = \sigma_{\text{in}}^2\delta_{kl}$, we obtain the average evolution of the state

$$\begin{aligned}\rho(n+1) - \rho(n) &= -i\varepsilon[H^+,\rho(n)] + \sigma_{\text{in}}^2\left(\hat{N}\rho(n)\hat{N} - \frac{1}{2}\{\hat{N}^2,\rho(n)\}\right)\\ &+ \sigma_{\text{in}}^2\left(\chi_\tau\hat{N}\chi_\tau\rho(n)\chi_\tau\hat{N}\chi_\tau - \frac{1}{2}\{\chi_\tau\hat{N}^2\chi_\tau,\rho(n)\}\right).\end{aligned} \quad \text{(C3)}$$

Here, $\{A,B\} = AB + BA$ denote commutators and anticommutators, respectively. We define $H^+ = \hat{N} + \chi_\tau\hat{N}\chi_\tau$. For times $T \gg 2\tau$, we can take Eq. (C3) to be a Lindbladian evolution, since the noise satisfies the Markovian properties. We can rewrite Eq. (C3) as

$$\dot{\rho} = \mathcal{L}_{\epsilon,\sigma}(\rho), \quad \text{(C4)}$$

$$\mathcal{L}_{\epsilon,\sigma}(\cdot) = -i\frac{\varepsilon}{2\tau}[H^+,\cdot] + \frac{\sigma_{\text{in}}^2}{2\tau}D^+(\cdot) + \frac{\sigma_{\text{in}}^2}{2\tau}D^-(\cdot), \quad \text{(C5)}$$

$$D^\pm(\cdot) = H^\pm H^\pm + \frac{1}{2}\{H^\pm H^\pm,\cdot\}, \quad \text{(C6)}$$

where $H^- = \hat{N} - \chi_\tau\hat{N}\chi_\tau$. For $\tau = 1.51\pi$, the Néel states are approximately simultaneous eigenstates of $\chi_\tau\hat{N}\chi_\tau$ and $\hat{N}$ with eigenvalues N for a system of size N. Thus, they are simultaneous eigenstates of $H^\pm$ and so

$$\mathcal{L}_{\epsilon,\sigma}(|AF\rangle\langle AF|) \approx 0, \quad \mathcal{L}_{\epsilon,\sigma}(|AF'\rangle\langle AF'|) \approx 0. \quad \text{(C7)}$$

Therefore, the Néel states are steady states. It is worth noting that $\mathcal{L}_{\epsilon,\sigma}$ captures the prethermal evolution. Ultimately, higher-order effects in ϵ_k take over and lead to the thermalization of the Néel states, similar to the results in Ref. [47] and as seen in Fig. 9. Nonetheless, the thermalization of the Néel states is delayed relative to other states due to Eq. (C7).

Moreover, any density matrix ρ_{SS} in the kernel of $\mathcal{L}_{\epsilon,\sigma}$ can be used as a memory state. Expressing $\mathcal{L}_{\epsilon,\sigma}$ as a superoperator on density matrices, we can look at its spectrum, which is in general complex. Figure 11 shows the number of zero eigenvalues of $\mathcal{L}_{\epsilon,\sigma}$ for different system sizes N. The number of zeros scales larger than linearly on N. Therefore, a quantum reservoir evolving under $\mathcal{L}_{\epsilon,\sigma}$ may have a larger number of memory states than a classical RNN. To prepare these states, we propose to initialize the reservoir on different string configurations $|s\rangle$ satisfying the Rydberg-blockade constraint. For example, one can have $s = rgg..g$, while $s = rrg\ldots g$ is not allowed. The system is left to evolve for some time T_{SS} to reach a steady state $\rho_{\text{SS}}(s)$, which can then be used as memory. Different initial strings can lead to different steady states, as exemplified in Fig. 12. Figure 12 shows the fidelity between $\rho_{\text{SS}}(s)$ and $\rho_{\text{SS}}(s')$ defined by the trace norm

$$F(\rho_{\text{SS}}(s),\rho_{\text{SS}}(s')) = \left(\text{Tr}\sqrt{\sqrt{\rho_{\text{SS}}(s)}\rho_{\text{SS}}(s')\sqrt{\rho_{\text{SS}}(s')}}\right)^2. \quad \text{(C8)}$$

The red arrows in Fig. 12 indicate the different memory states obtained by this scheme. It is worth noting that this scheme offers us an empirical number of memories N_m^e that scales at most as ϕ^N, where $\phi \approx 1.62$ is the golden

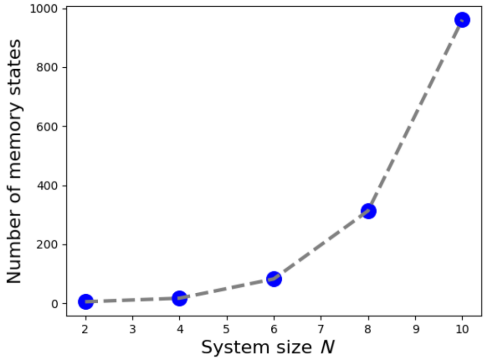

FIG. 11. The number of zero eigenvalues of the superoperator $\mathcal{L}_{\epsilon,\sigma}$ as a function of the system size. $\mathcal{L}_{\epsilon,\sigma}$ describes the effective dynamics of a Rydberg reservoir composed of kicked Rydberg atoms. The number of zeros surpasses the linear number of memories available in the Hopfield network.

ratio, since that is the number of basis states respecting the Rydberg blockade. We see that $N_m^e > N$ in all instances, a bound that is unattainable by classical RNNs such as the Hopfield network [66]. However, this scheme relies on an efficient way to recognize the different memory states through measurements, a question that we leave for future investigations.

APPENDIX D: EXPERIMENTAL VALUES, AND NUMERICAL SIMULATIONS

In this appendix, we outline the details of the experimental values used for the numerical simulation of Sec. V. First, for the simulation of Rydberg atoms we use the experimental values in Ref. [38] for concreteness (see

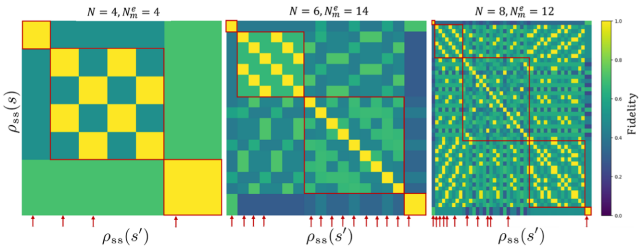

FIG. 12. Empirical memory states $\rho_{SS}(s)$ obtained from evolving the initial states $|s\rangle$, which are basis states of the Rydberg-blockaded Hilbert space. N_m^e denotes the number of memories found using this procedure. The different plots show the fidelities $F(\rho_{SS}(s), \rho_{SS}(s'))$ between different steady states. The red squares delimit the basis states with different numbers of Rydberg excitations, starting with the zero-excitation sector on the top-left square and ending with $N/2$ excitations sector on the bottom-right square. The red arrows denote initial configurations for each of the N_m^e memories found empirically. While this procedure produces a number of memory states smaller than the number of zeros of $\mathcal{L}_{\epsilon,\sigma}$, $N_m^e > N$, a bound unattainable by common classical RNNs.

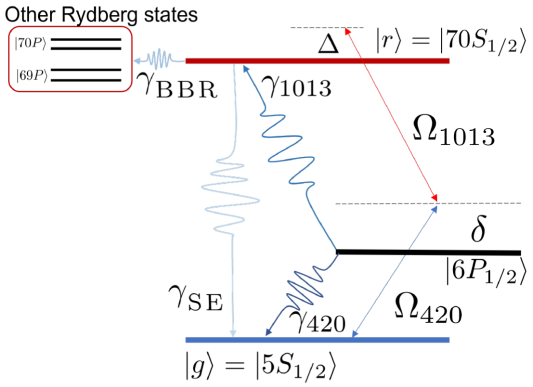

FIG. 13. A schematic of the Rydberg atoms as used in Ref. [38]. The ground state $|g\rangle = |5S_{1/2}\rangle$ and the Rydberg state $|r\rangle = |50S_{1/2}\rangle$ are coupled via a two-photon transition. An off-resonance 420-nm laser ($\Omega_{420} = 2\pi \times 160$ MHz, $\delta = 2\pi \times 1$ GHz) couples $|g\rangle$ to the intermediate $|6P_{3/2}\rangle$ state and a 1013-nm laser ($\Omega_{1013} = 2\pi \times 50$ MHz) couples the intermediate state and $|r\rangle$, creating an effective drive between $|g\rangle$ and $|r\rangle$ at rate $\Omega = \Omega_{420}\Omega_{1013}/\delta = 2\pi \times 4.2$ MHz. Four spontaneous-emission processes are at play: emission to nearby Rydberg atoms due to black-body radiation at a rate $\gamma_{BBR} = 2\pi/(250\ \mu s)$; photon scattering out of the intermediate state into the ground state at rate $\gamma_{420} = 2\pi/(20\ \mu s)$ and into the Rydberg state at rate $\gamma_{1013} = 2\pi/(150\ \mu s)$; and spontaneous emission from $|r\rangle$ to $|g\rangle$ at rate $\gamma_{SE} = 2\pi/(375\ \mu s)$. Since $\gamma_{BBR} + \gamma_{SE} + \gamma_{1013} = 2\pi/(75\ \mu s)$ is smaller than γ_{420}, the leading source of decoherence for short periods of time ($< 10\ \mu s$) is due to the γ_{420} decay.

Fig. 13). In this experimental platform, a two-photon transition couples $|g\rangle = |5S_{1/2}\rangle$ and $|r\rangle = |50S_{1/2}\rangle$ via an off-resonance state $|6P_{3/2}\rangle$. For this setup, and for short periods of simulation ($< 10\ \mu s$), the dominant source of decoherence is photon-scattering processes out of the intermediate state. Using the fact that the intermediate state is off resonance, we can adiabatically eliminate it to produce an effective decay operator (see Sec. IV.B in Ref. [69])

$$\sigma_{\text{eff}}^{-} = \frac{\sqrt{\gamma_{420}}}{2\delta} |g\rangle \left(\Omega_{420}\langle g| + \Omega_{1013}\langle r|\right), \qquad \text{(D1)}$$

which is an effective spontaneous emission from $|r\rangle$ to $|g\rangle$ accompanied by decoherence on the ground state.

We choose $\Omega = 4.2$ MHz. Additionally, a pair of $|r\rangle$ atoms interact with a strength $C_6 = 862.9$ GHz $(\mu m)^6$. We use the PairInteraction PYTHON package from Ref. [58] to determine that a pair of $|r\rangle = |70S_{1/2}\rangle$ and $|r'\rangle = |73S_{1/2}\rangle$ has a similar interaction strength of $C_6^{rr'} = -836.6$ GHz $(\mu m)^6 \approx -C_6$. We use this interaction to model the inhibitory and excitatory neurons in Sec. V A ($V_{n_Q,n_Q} = V$, $V_{n_Q,n_Q'} = -V$). We denote $V = C_6/a_0^6$, where a_0 is tuned to give us different nearest-neighbor interaction strengths.

Next, we explain and report the numerical parameters chosen for each of the biological tasks.

1. Multitasking

Our scheme to encode inhibitory and excitatory neurons relies on approximating Eq. (13). To do so, one needs the "inhibitory neurons" to be far away from each other such that they do not interact positively. For this reason, this task uses a 1D open chain of atoms separated by a distance a_0, with the inhibitory neurons being at opposite ends of the chain and in the bulk, with maximum spacing from each other. The input neurons are chosen to be the two at one end of the chain, while the output neuron is chosen to be at the opposite end of the chain. This choice is made to ensure that the input neurons interact with the whole chain before readout.

The inputs are uniformly sampled from $\{0, 2\pi\}$ MHz with added Gaussian noise $\sigma_{\text{in}} = 0.1$ and all Δt sampled from a Gaussian with average $\langle \Delta t \rangle \in [0, 5]$ (μs) and standard deviation σ_{in}. For each size of network and number of inhibitory neurons, we choose a_0 such that the separation between inhibitory neurons $d_{\max}$ results in $V/d_{\max}^2 = 10^{-2}$. For example, for the case of four neurons and two inhibitory neurons on either end, note that one needs $V/3^6 = 10^{-2}$, which amounts to choosing $V = 7.2$ MHz. Note that this value of V is of the order of magnitude of $\Omega = 4.2$ MHz and so the reservoir, in this case, is well into the nonclassical regime.

The learned parameters in the output linear map W^{out}, which in this case is a matrix in $\mathbb{R}^{3+1\times 1}$, with the last row representing a bias term. Note that the dimension of the map is so because only one neuron is measured but three functions have to be fitted.

2. Decision making

In classical RNNs, tasks such as decision making and working memory require connectivity between all neurons. Since our connectivity is limited by physical constraints, an open 2D square-lattice structure is chosen to prevent neurons from being isolated from the rest. Moreover, a 2D square lattice is experimentally friendly. In our case, we use an open 2×3 lattice with the two input neurons being at the top-left corner of the chain and two output neurons being at the bottom-right corner. Again, this architecture is chosen so that the input neurons have to interact with the rest of the system before readout. We use $V = 2\pi \times 10$ MHz for our simulations and choose $\Delta t = 2\pi/V$ as the time for which the inputs are turned on, as that is the time scale over which the input atoms entangle with the rest of the chain.

The inputs are uniformly sampled from $\{0, \pi/2, \pi, 3\pi/2, 2\pi\}$ (MHz) with added Gaussian noise $\sigma_{\text{in}} = 0.1$. In this task, the time for which the stimuli are turned on, Δt, is fixed to a mean of $\langle \Delta t \rangle = 0.1$ μs and with added Gaussian noise $\sigma_{\text{in}} = 0.1$. In this task, we optimize over the linear output map W^{out}. In this case, W^{out} is a matrix in $\mathbb{R}^{1+1,2}$. The first dimension of $1 + 1$ denotes that one function is fitted and a bias is added. The second dimension of 2 denotes that two neurons are measured. Additionally, we train the output time t_{out} after the stimuli are turned off and before the network is probed to come up with an input that is satisfied via Eq. (15). To do the optimization, we make use of the Nelder-Mead algorithm [70].

In order to compute the psychometric response plotted in Fig. 6(b), we measure the expectation values on the two output neurons and produce the vector $\mathbf{r}(\Delta_1^{\text{in}}, \Delta_2^{\text{in}}) = (\langle \sigma_{\text{out1}}^y \rangle, \langle \sigma_{\text{out2}}^y \rangle, 1)$, which depends on the inputs $\Delta_{1,2}^{\text{in}}$, as well as the temporal parameters $(\Delta t, t_{\text{out}})$. We then compute $y^{\text{out}}(\Delta_{1,2}^{\text{in}}) = W^{\text{out}}\mathbf{r}(\Delta_{1,2}^{\text{in}})$ and $(W^{\text{out}}, t_{\text{out}})$ are optimized such that $y^{\text{out}}(\Delta_{1,2}^{\text{in}}) \approx y^{\text{targ}}$ in 15. The optimization is done by generating about 40 000 different values of $\Delta_{1,2}^{\text{in}}$ of different levels of contrast $|\Delta_1^{\text{in}} - \Delta_2^{\text{in}}|$ ranging from 0 to 1 MHz. Once the optimization is done, we look at the loss toward Δ_2^{in}, which is obtained as the error in classifying Δ_2^{in} as greater than Δ_1^{in} when, indeed, $\Delta_2^{\text{in}} > \Delta_1^{\text{in}}$. The error is quantified using the mean-square loss in Eq. (4).

3. Working memory

The setup of this task is identical to the decision-making task except that the two inputs are separated by a delay time t_{delay}. The values of the interaction strength V used for Fig. 7 are $V = 2\pi \times 10$ MHz and $V = 2\pi \times 0.1$ MHz, corresponding to $V/\Omega > 1$ and $V/\Omega < 1$, respectively, the former of which sets us in the Rydberg-blockaded regime while the latter does not. In this task, the times Δt and t_{delay} are fixed up to an added Gaussian with noise $\sigma_{\text{in}} = 0.1$. In this task, we optimize over the linear output map W^{out}. In this case, W^{out} is a matrix in $\mathbb{R}^{1+1,2}$. The first dimension of $1 + 1$ denotes that one function is fitted and a bias is added. The second dimension of 2 denotes that two neurons are measured.

4. Long-term memory

Although quantum scars are known to exist in other geometries and dimensions [71], for this task we use a 1D chain of Rydberg atoms, since for this case quantum many-body scars have been experimentally observed [35,46]. Furthermore, our chain has periodic boundary conditions to avoid edge effects. Since we know that scars are robust to decoherence, we set $\gamma_{420} = 0$ so that we can evolve our states for longer periods. The number of cycles n in Fig. 9 corresponds to n evolutions under the PXP Hamiltonian for a time $2\tau = 1.51 \times \pi\Omega^{-1}$. In this case, we take $V \gg \Omega$ and renormalized $\Omega = 1$. The noisy field in Eq. (16) is sampled according to $\epsilon_k \sim N(\mu = 0.1, \sigma = 0.1)$. The input m is sampled as a fair random coin. Lastly, after each number of cycles n, the only trained parameter is $W_n^{\text{out}} \in \mathbb{R}^{1+1\times 1}$, since only one atom is probed to calculate an answer as to the input m.

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

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[3] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[4] G. H. Low, T. J. Yoder, and I. L. Chuang, Quantum inference on Bayesian networks, Phys. Rev. A **89**, 062315 (2014).

[5] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[6] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 1 (2018).

[7] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nat. Commun. **12**, 1 (2021).

[8] D. Marković and J. Grollier, Quantum neuromorphic computing, Appl. Phys. Lett. **117**, 150501 (2020).

[9] B. Kiraly, E. J. Knol, W. M. J. van Weerdenburg, H. J. Kappen, and A. A. Khajetoorians, An atomic Boltzmann machine capable of self-adaption, Nat. Nanotechnol. **16**, 414 (2021).

[10] P. Mujal, R. Martínez-Peña, J. Nokkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Opportunities in quantum reservoir computing and extreme learning machines, Preprint ArXiv:2102.11831 (2021).

[11] P. Pfeiffer, I. Egusquiza, M. Di Ventra, M. Sanz, and E. Solano, Quantum memristors, Sci. Rep. **6**, 1 (2016).

[12] T. Gonzalez-Raya, X.-H. Cheng, I. L. Egusquiza, X. Chen, M. Sanz, and E. Solano, Quantized Single-Ion-Channel Hodgkin-Huxley Model for Quantum Neurons, Phys. Rev. Appl. **12**, 014037 (2019).

[13] T. Gonzalez-Raya, E. Solano, and M. Sanz, Quantized three-ion-channel neuron model for neural action potentials, Quantum **4**, 224 (2020).

[14] E. Torrontegui and J. J. García-Ripoll, Unitary quantum perceptron as efficient universal approximator, EPL **125**, 30004 (2019).

[15] K. Fujii and K. Nakajima, Quantum reservoir computing: A reservoir approach toward quantum machine learning on near-term quantum devices, Preprint ArXiv:2011.04890 (2020).

[16] K. Nakajima, K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa, Boosting Computational Power through Spatial Multiplexing in Quantum Reservoir Computing, Phys. Rev. Appl. **11**, 034021 (2019).

[17] A. Kutvonen, K. Fujii, and T. Sagawa, Optimizing a quantum reservoir computer for time series prediction, Sci. Rep. **10**, 14687 (2020).

[18] S. Ghosh, A. Opala, M. Matuszewski, T. Paterek, and T. C. H. Liew, Quantum reservoir processing, npj Quantum Inf. **5**, 35 (2019).

[19] S. A. Khan, F. Hu, G. Angelatos, and H. E. Türeci, Physical reservoir computing using finitely-sampled quantum systems, Preprint ArXiv:2110.13849 (2021).

[20] S. Ghosh, T. Paterek, and T. C. H. Liew, Quantum Neuromorphic Platform for Quantum State Preparation, Phys. Rev. Lett. **123**, 260404 (2019).

[21] L. Govia, G. Ribeill, G. Rowlands, H. Krovi, and T. Ohki, Quantum reservoir computing with a single nonlinear oscillator, Phys. Rev. Res. **3**, 013077 (2021).

[22] 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, Commun. Phys. **4**, 1 (2021).

[23] S. Ghosh, T. Krisnanda, T. Paterek, and T. C. Liew, Realising and compressing quantum circuits with quantum reservoir computing, Commun. Phys. **4**, 1 (2021).

[24] P. Mujal, Quantum reservoir computing for speckle-disorder potentials, Preprint ArXiv:2201.11096 (2022).

[25] A. Sakurai, M. P. Estarellas, W. J. Munro, and K. Nemoto, Quantum reservoir computation utilising scale-free networks, Preprint ArXiv:2108.12131 (2021).

[26] W. Xia, J. Zou, X. Qiu, and X. Li, The reservoir learning power across quantum many-body localization transition, Front. Phys. **17**, 1 (2022).

[27] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, Dynamical Phase Transitions in Quantum Reservoir Computing, Phys. Rev. Lett. **127**, 100502 (2021).

[28] A. C. C. Coolen, in *Handbook of Biological Physics*, Vol. 4, Chapter 14 (Elsevier, North-Holland, 2001), p. 553.

[29] Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto, Natural quantum reservoir computing for temporal information processing, Sci. Rep. **12**, 1 (2022).

[30] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with Rydberg atoms, Rev. Mod. Phys. **82**, 2313 (2010).

[31] B. J. Lester, N. Luick, A. M. Kaufman, C. M. Reynolds, and C. A. Regal, Rapid Production of Uniformly Filled Arrays of Neutral Atoms, Phys. Rev. Lett. **115**, 073003 (2015).

[32] D. Barredo, S. De Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys, An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays, Science **354**, 1021 (2016).

[33] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, Science **354**, 1024 (2016).

[34] H. Labuhn, D. Barredo, S. Ravets, S. De Léséleuc, T. Macrì, T. Lahaye, and A. Browaeys, Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models, Nature **534**, 667 (2016).

[35] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, Probing many-body dynamics on a 51-atom quantum simulator, Nature **551**, 579 (2017).

[36] A. Cooper, J. P. Covey, I. S. Madjarov, S. G. Porsev, M. S. Safronova, and M. Endres, Alkaline-earth atoms in optical tweezers, Phys. Rev. X **8**, 041055 (2018).

[37] J. Wilson, S. Saskin, Y. Meng, S. Ma, R. Dilip, A. Burgers, and J. Thompson, Trapping Alkaline Earth Rydberg Atoms Optical Tweezer Arrays, Phys. Rev. Lett. **128**, 033201 (2022).

[38] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho, *et al.*, Quantum phases of matter on a 256-atom programmable quantum simulator, ArXiv:2012.12281 (2020).

[39] L. Isenhower, E. Urban, X. Zhang, A. Gill, T. Henage, T. A. Johnson, T. Walker, and M. Saffman, Demonstration of a Neutral Atom Controlled-NOT Quantum Gate, Phys. Rev. Lett. **104**, 010503 (2010).

[40] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, Quantum optimization for maximum independent set using Rydberg atom arrays, ArXiv:1808.10816 (2018).

[41] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. Wang, S. Ebadi, H. Bernien, A. Zibrov, H. Pichler, S. Choi, *et al.*, Generation and manipulation of Schrödinger cat states in Rydberg atom arrays, Science **365**, 570 (2019).

[42] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, Quantum **4**, 327 (2020).

[43] S. R. Cohen and J. D. Thompson, Quantum Computing with Circular Rydberg Atoms, PRX Quantum **2**, 030322 (2021).

[44] H. F. Song, G. R. Yang, and X.-J. Wang, Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework, PLoS Comput. Biol. **12**, 1 (2016).

[45] J. Han and T. F. Gallagher, Millimeter-wave rubidium Rydberg van der Waals spectroscopy, Phys. Rev. A **79**, 053409 (2009).

[46] D. Bluvstein, A. Omran, H. Levine, A. Keesling, G. Semeghini, S. Ebadi, T. T. Wang, A. A. Michailidis, N. Maskara, W. W. Ho, *et al.*, Controlling quantum many-body dynamics in driven Rydberg atom arrays, Science **371**, 1355 (2021).

[47] N. Maskara, A. A. Michailidis, W. W. Ho, D. Bluvstein, S. Choi, M. D. Lukin, and M. Serbyn, Discrete time-crystalline order enabled by quantum many-body scars: Entanglement steering via periodic driving, Preprint ArXiv:2102.13160 (2021).

[48] J. Bausch, Recurrent quantum neural networks, Preprint ArXiv:2006.14619 (2020).

[49] Y. Wu and X. Yang, Strong-Coupling Theory of Periodically Driven Two-Level Systems, Phys. Rev. Lett. **98**, 013601 (2007).

[50] F. Ghafari, N. Tischler, J. Thompson, M. Gu, L. K. Shalm, V. B. Verma, S. W. Nam, R. B. Patel, H. M. Wiseman, and G. J. Pryde, Dimensional quantum memory advantage in the simulation of stochastic processes, Phys. Rev. X **9**, 041013 (2019).

[51] D. H. Wolpert, A. Kolchinsky, and J. A. Owen, A space-time tradeoff for implementing a function with master equation dynamics, Nat. Commun. **10**, 1 (2019).

[52] K. Korzekwa and M. Lostaglio, Quantum advantage in simulating stochastic processes, Phys. Rev. X **11**, 021019 (2021).

[53] C. Blank, D. K. Park, and F. Petruccione, Quantum-enhanced analysis of discrete stochastic processes, npj Quantum Inf. **7**, 1 (2021).

[54] A. N. Burkitt, A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input, Biol. Cybern. **95**, 1 (2006).

[55] H. Jaeger, in *GMD—German National Research Institute for Computer Science*, GMD-Report 152, 2002.

[56] V. Capano, H. J. Herrmann, and L. de Arcangelis, Optimal percentage of inhibitory synapses in multi-task learning, Sci. Rep. **5**, 9895 (2015).

[57] J. C. Eccles, P. Fatt, and K. Koketsu, Cholinergic and inhibitory synapses in a pathway from motor-axon collaterals to motoneurones, J. Physiol. (Lond.) **126**, 524 (1954).

[58] S. Weber, C. Tresp, H. Menke, A. Urvoy, O. Firstenberg, H. P. Büchler, and S. Hofferberth, Tutorial: Calculation of Rydberg interaction potentials, J. Phys. B: At. Mol. Opt. Phys. **50**, 133001 (2017).

[59] J. D. Roitman and M. N. Shadlen, Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task, J. Neurosci. **22**, 9475 (2002).

[60] L. Govia, G. Ribeill, G. Rowlands, and T. Ohki, Nonlinear input transformations are ubiquitous in quantum reservoir computing, Neuromorphic Comput. Eng. **2**, 014008 (2022).

[61] E. Urban, T. A. Johnson, T. Henage, L. Isenhower, D. Yavuz, T. Walker, and M. Saffman, Observation of Rydberg blockade between two atoms, Nat. Phys. **5**, 110 (2009).

[62] A. Gaetan, Y. Miroshnychenko, T. Wilk, A. Chotia, M. Viteau, D. Comparat, P. Pillet, A. Browaeys, and P. Grangier, Observation of collective excitation of two individual atoms in the Rydberg blockade regime, Nat. Phys. **5**, 115 (2009).

[63] L. D'Alessio, Y. Kafri, A. Polkovnikov, and M. Rigol, From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics, Adv. Phys. **65**, 239 (2016).

[64] P. Fendley, K. Sengupta, and S. Sachdev, Competing density-wave orders in a one-dimensional hard-boson model, Phys. Rev. B **69**, 075106 (2004).

[65] I. Lesanovsky and H. Katsura, Interacting Fibonacci anyons in a Rydberg gas, Phys. Rev. A **86**, 041601 (2012).

[66] V. Folli, M. Leonetti, and G. Ruocco, On the maximum storage capacity of the Hopfield model, Front. Comput. Neurosci. **10**, 144 (2017).

[67] A. Periwal, E. S. Cooper, P. Kunkel, J. F. Wienand, E. J. Davis, and M. Schleier-Smith, Programmable interactions and emergent geometry in an atomic array, Preprint ArXiv:2106.04070 (2021).

[68] G. S. Goodman, An intrinsic time for non-stationary finite Markov chains, Z. Wahrscheinlichkeit. **16**, 165 (1970).

[69] F. Reiter and A. S. Sørensen, Effective operator formalism for open quantum systems, Phys. Rev. A **85**, 032111 (2012).

[70] J. A. Nelder and R. Mead, A simplex method for function minimization, Comput. J. **7**, 308 (1965).

[71] M. Serbyn, D. A. Abanin, and Z. Papić, Quantum many-body scars and weak breaking of ergodicity, Nat. Phys. **17**, 675 (2021).