ARTICLE **Communicated by Nick Hardy**

Stimulus-Driven and Spontaneous Dynamics in Excitatory-Inhibitory Recurrent Neural Networks for Sequence Representation

Alfred Rajakumar
aar653@nyu.edu
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, U.S.A.

John Rinzel
rinzel@cns.nyu.edu
Courant Institute of Mathematical Sciences and Center for Neural Science, New York University, New York, NY 10012, USA.

Zhe S. Chen
zhe.chen@nyulangone.org
Department of Psychiatry and Neuroscience Institute, New York University School of Medicine, New York, NY 10016, U.S.A.

Recurrent neural networks (RNNs) have been widely used to model sequential neural dynamics ("neural sequences") of cortical circuits in cognitive and motor tasks. Efforts to incorporate biological constraints and Dale's principle will help elucidate the neural representations and mechanisms of underlying circuits. We trained an excitatory-inhibitory RNN to learn neural sequences in a supervised manner and studied the representations and dynamic attractors of the trained network. The trained RNN was robust to trigger the sequence in response to various input signals and interpolated a time-warped input for sequence representation. Interestingly, a learned sequence can repeat periodically when the RNN evolved beyond the duration of a single sequence. The eigenspectrum of the learned recurrent connectivity matrix with growing or damping modes, together with the RNN's nonlinearity, were adequate to generate a limit cycle attractor. We further examined the stability of dynamic attractors while training the RNN to learn two sequences. Together, our results provide a general framework for understanding neural sequence representation in the excitatory-inhibitory RNN.

1 Introduction

Sequentially activated neuronal activities ("neural sequences") are universal neural dynamics that have been widely observed in neural assemblies

Neural Computation **33, 2603–2645** (2021)
https://doi.org/10.1162/neco_a_01418

of cortical and subcortical circuits (Fujisawa, Amarasingham, Harrison, & Buzsaki, 2008; Long et al., 2010; Harvey, Coen, & Tank, 2012; Buzsaki & Tingley, 2018; Adler et al., 2019; Hemberger, Shein-Idelson, Pammer, & Laurent, 2019). Sequence-based neural dynamics have been proposed as a common framework for circuit function during spatial navigation, motor learning, memory, and decision-making tasks (Sussillo, 2014). Neural sequential activity can be driven by sensory or motor input, such as in the rodent hippocampus, rodent primary motor cortex, and premotor nucleus HVC of song birds; neural sequences can be driven by intrinsic dynamics during the task period in the absence of sensory or motor input. Such sequential dynamics could potentially be implemented using feedforward or recurrent architectures (Goldman, 2009; Cannon, Kopell, Gardner, & Markowitz, 2015).

Biological networks are strongly recurrent. Therefore, nonlinear recurrent neural networks (RNNs) have been developed for modeling a wide range of neural circuits in various cognitive and motor tasks (Mante, Sussillo, Shenoy, & Newsome, 2013; Sussilo, Churchland, Kaufman, & Shenoy, 2015; Rajan, Harvey, & Tank, 2016; Barak, 2017; Goudar & Buonomano, 2018; Yang, Joglekar, Song, Newsome, & Wang, 2019; Kao, 2019; Mackwood, Naumann, & Sprekeler, 2021; Bi & Zhou, 2020; Zhang, Liu, & Chen, 2021). However, biological constraints were only recently considered in RNN models (Song, Yang, & Wang, 2016; Ingrosso & Abbott, 2019; Xue, Halassa, & Chen, 2021), whereas in other work (Murphy & Miller, 2009), biological constraints were considered only in the linear recurrent system. In general, RNNs have been treated as black boxes, and their mechanisms and high-dimensional computations are difficult to analyze (Sussillo & Barak, 2013; Ceni, Ashwin, & Livi, 2019).

In this article, we used an excitatory-inhibitory nonlinear RNN to model neural sequences. Extending previous modeling efforts (Rajan et al., 2016; Hardy & Buonomano, 2018; Orhan & Ma, 2019), we explicitly incorporated Dale's principle and excitatory-inhibitory (E/I) balance into the RNN and studied its stimulus-driven and spontaneous dynamics. Neural sequences, as a special form of dynamical attractors, can be viewed as an alternative to fixed points for storing memories (Rajan et al., 2016). Generalized Hopfield-type neural networks can store patterns in limit cycles (Deshpande & Dasgupta, 1991). However, to our best knowledge, limit cycle dynamics have not been well studied in the context of excitatory-inhibitory RNN.

Neural dynamics during online task behavior is referred to as a transient state, whereas the neural dynamics during the offline state (in the absence of stimulus input) is often described as the steady or stationary state, which can also play important roles in brain functions (such as memory replay). Based on extensive computer simulations for learning one or two neural sequences, we discovered and studied the stability of limit cycles encoded by the excitatory-inhibitory RNN while evolving in the stimulus-free spontaneous state ("steady state"). Our combined theoretical and computational

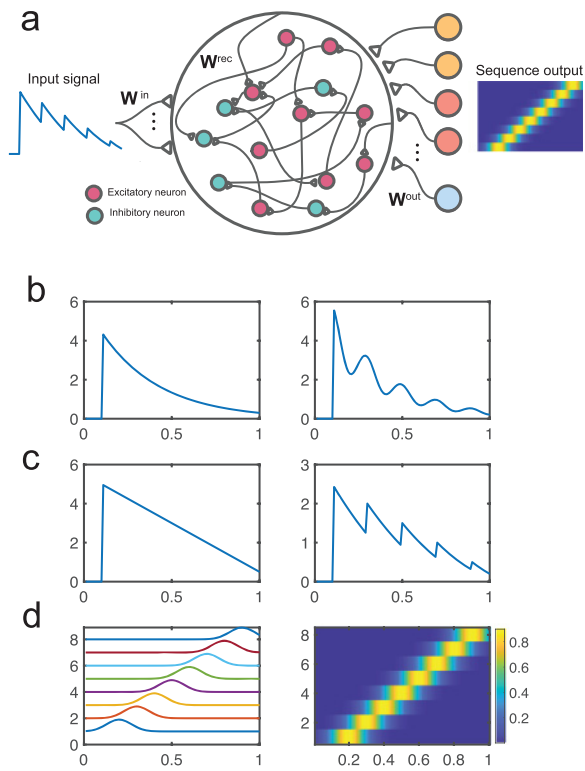

Figure 1: (a) Schematic of an excitatory-inhibitory recurrent neural network (RNN) characterized a fully recurrent connectivity matrix $\mathbf{W}^{rec}$. All neurons in the RNN receive a time-varying input signal (scaled by $\mathbf{W}^{in}$) and produce a partial sequence in the neuronal readout. The readout is assumed to be the subset of excitatory neurons (scaled by $\mathbf{W}^{out}$). (b,c) Several versions of input signals. (b) An exponentially decaying signal $6e^{-3(t-0.1)}$ (left) and modulated sinusoidal signal $6e^{-3(t-0.1)}(1 + 0.3\cos(10\pi(t - 0.1)))$ (right). (c) A linear function (left) and a modulated sawtooth signal (right). (d) Target neural sequence represented by eight readout neurons. Right: The sequential neural activation is shown as a heat map, where the values of each row are normalized between 0 and 1.

analyses provide a framework to study sequence representation in neural circuits in terms of excitatory-inhibitory balance, network connectivity, and synaptic plasticity.

2 Excitatory-Inhibitory RNN

In a basic RNN (see Figure 1a), the network receives an N_{in}-dimensional input $\mathbf{u}(t)$ and produces an N_{out}-dimensional output $\mathbf{z}(t)$. We assumed the

neural state dynamics (which can be interpreted as the latent current or activity as in a rate model), $\mathbf{x}(t)$, follows

$$\tau \dot{\mathbf{x}} = -\mathbf{x} + \mathbf{W}^{\text{rec}}\mathbf{r} + \mathbf{W}^{\text{in}}\mathbf{u} + \sigma\boldsymbol{\xi}, \tag{2.1}$$

where τ denotes the time constant; ξ denotes additive N-dimensional gaussian noise, each independently drawn from a standard normal distribution; σ^2 defines the scale of the noise variance; $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N\times N_{\text{in}}}$ denotes the matrix of connection weights from the inputs to network units and $\mathbf{W}^{\text{rec}}$ is an $N \times N$ matrix of recurrent connection weights, and $\mathbf{W}^{\text{out}}$ is an $N_{\text{out}} \times N$ matrix of connection weights from the network units to the output.

The network state of RNN is described by the N-dimensional dynamical system over time. The neuronal firing rate vector $\mathbf{r}$ is defined by a nonlinear function $\phi(\mathbf{x})$, which can be either a rectified linear function or a softplus function,

$$\mathbf{r} = \phi(\mathbf{x}) = \begin{cases} [\mathbf{x}]_+ = \max\{\mathbf{x}, \mathbf{0}\} \\ \texttt{softplus}(\mathbf{x}) = \log(1 + e^{\mathbf{x}}) \end{cases}, \tag{2.2}$$

where the rectified linear function is a piecewise linear function, whereas the softplus function is continuous and differentiable function. The rectified linear unit (ReLU) is scale invariant and favors sparse activation. Additionally, it can alleviate the saturation or vanishing gradient problem as can occur for a sigmoid activation function (Glorot, Bordes, & Bengio, 2011). For both ReLU and softplus functions, the nonlinearity is defined componentwise for the vector $\mathbf{x}$. Furthermore, the output $\mathbf{z}$ is given by

$$\mathbf{z} = \mathbf{W}^{\text{out}}\mathbf{r}. \tag{2.3}$$

In a scalar form, equation 2.1 is described by

$$\tau \frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} w_{ij}^{\text{rec}} r_j + w_i^{\text{in}} u + \sigma\xi_i, \tag{2.4}$$

and the output component z_ℓ ($\ell = 1, \ldots, N_{\text{out}}$) is given by

$$z_\ell = \sum_{i=1}^{N} w_{\ell i}^{\text{out}} r_i. \tag{2.5}$$

The RNN dynamics is nonlinear due to the function $\phi(\mathbf{x})$. Generally, we can rewrite equation 2.1 as a nonlinear mapping function G: $\dot{\mathbf{x}} = G(\mathbf{x}, u)$,

with the first-order derivative (Jacobian) defined as

$$\nabla_{\mathbf{x}} G(\mathbf{x}, u) = -\frac{1}{\tau}\mathbf{I} + \frac{1}{\tau}\mathbf{W}^{\text{rec}} diag\{\phi'(\mathbf{x})\} \tag{2.6}$$

where the derivative $\phi'(x) = \frac{e^x}{1+e^x} < 1$ is a nonnegative logistic sigmoid function in the case of softplus function, whereas the derivative of ReLU is defined by a Heaviside step function. In the scalar case, we have

$$\frac{\partial G_i(\mathbf{x}, u)}{\partial x_j} = -\frac{1}{\tau}\delta_{ij} + \frac{1}{\tau} w_{ij}^{\text{rec}} \phi'(x_j), \tag{2.7}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

In a special case when $\phi(\cdot)$ is an identity function, the firing rate dynamics is reduced to a linear dynamical system,

$$\tau \dot{\mathbf{r}} = -\mathbf{r} + \mathbf{W}^{\text{rec}}\mathbf{r} + \mathbf{W}^{\text{in}}\mathbf{u} + \sigma\boldsymbol{\xi}, \tag{2.8}$$

where the eigenvectors of $\mathbf{W}^{\text{rec}}$ determine N basis patterns of neurons, and the associated eigenvalues determine the pattern propagation (Murphy & Miller, 2009). For instance, the eigenvalues $\{\lambda_1, \ldots, \lambda_N\}$ of $\mathbf{W}^{\text{rec}}$ with positive real part correspond to patterns that can self-amplify by routing through the network (if not dominated by the leakage rate, -1). In the special case where equation 2.8 has no noise and external input, the firing rate mode decays exponentially with a time constant proportional to $\tau/(1 - \text{Re}\{\lambda_i\})$.

Biological neuronal networks consist of excitatory and inhibitory neurons. Local cortical circuits have varying degrees of connectivity and sparsity depending on the nature of the cell types: excitatory-to-excitatory (EE), inhibitory-to-excitatory (IE), excitatory-to-inhibitory (EI), and inhibitory-to-inhibitory (II) connections (Murphy & Miller, 2009). We imposed the following biological constraints on our excitatory-inhibitory RNN:

- The ratio of the number of excitatory to inhibitory neurons is around 4:1 according to Dale's principle: $\frac{N_{\text{exc}}}{N_{\text{inh}}} = \frac{4}{1}$, where $N = N_{\text{exc}} + N_{\text{inh}}$.
- The dimensionality of the input signal is 1, $N_{\text{in}} = 1$; for ease of interpretation, we assumed that $\mathbf{W}^{\text{in}} = \{w_i^{\text{in}}\}_{N\times 1}$ has only positive elements.
- $\mathbf{W}^{\text{rec}} = \{w_{ij}^{\text{rec}}\}_{N\times N}$ has EE, EI, IE, and II connections as follows: $\begin{bmatrix} \text{EE} & \text{IE} \\ \text{EI} & \text{II} \end{bmatrix}$. Both EE and EI connections were from presynaptic excitatory neurons, and therefore their weights were positive, whereas IE and II connections were from presynaptic inhibitory neurons and had negative weights. No self-connection was assumed for $\mathbf{W}^{\text{rec}}$; namely, all diagonal elements $\{w_{ii}^{\text{rec}}\}$ are zeros.

- The output sequences are direct (scaled) readout of N_{out} neurons from the excitatory neuron population. Therefore, $\mathbf{W}^{\text{out}} = \{w^{\text{out}}_{\ell i}\}_{N_{\text{out}} \times N}$ is a partial identity matrix. Here, $N_{\text{out}} = 8$, and we constrained $\mathbf{W}^{\text{out}}$ such that its upper block contained an $N_{\text{out}} \times N_{\text{out}}$ diagonal matrix, namely, $w^{\text{out}}_{\ell i} = 0, \forall \ell \neq i$.

For the ith postsynaptic neuron in the sequence output, we define its net excitatory and inhibitory currents as follows:

$$I_i^{\text{exc}} = \sum_{j \in \text{exc}} w^{\text{rec}}_{ij,\text{EE}} r_j,$$
$$I_i^{\text{inh}} = \sum_{j \in \text{inh}} w^{\text{rec}}_{ij,\text{IE}} r_j, \tag{2.9}$$

where $r_j = [x_j]_+ = \max(x_j, 0)$ denotes the neuronal firing rate of the jth presynaptic neuron, and $w^{\text{rec}}_{ij,\text{EE}}$ and $w^{\text{rec}}_{ij,\text{IE}}$ represent the EE and IE weights within $\mathbf{W}^{\text{rec}}$, respectively.

2.1 Computer Simulation Setup. For a relatively simple computational task, we assumed that the input was a one-dimensional signal (i.e., $N_{\text{in}} = 1$) and set $N = 100$. According to Dale's principle, we assumed that the EE, EI, IE, and II connections were represented as 80×80, 20×80, 80×20, and 20×20 submatrices, respectively.

We have considered different forms of input signals $u(t)$ lasting 900 ms during the 1 s task period (see Figures 1b and 1c), which could be either sinusoidal modulated on an exponential carrier (single sequence case) or sawtooth functions on a linear increasing/decreasing carrier (two sequences case). The continuous-time model was numerically implemented in discrete time by Euler's method. For each input signal, we set $dt = 10$ ms and $N_{\text{time}} = 100$, resulting in a simulation trial duration of ~ 1 s. We have also tried smaller dt (e.g., 2 ms and 5 ms) and obtained similar results. However, the simulation sample size became much larger and significantly increased the computational cost. Therefore, our default setup was $dt = 10$ ms.

The output has the same duration as the input signal. The desired readout of the RNN was a neural sequence (see Figure 1d); each neuronal activation was sparse in time and had 5% overlap between its neighbor activation. In the simulations here, we used the first eight excitatory neurons' firing rates as the readout for the sequence output (i.e., $N_{\text{out}} = 8$). The number of trials was $N_{\text{trials}} = 20$, and the input signals were generated with added independent additive gaussian noise (zero mean and variance 0.01).

2.2 RNN Training with Gradient Descent. We initialized the RNN weights as follows. The input weights $\mathbf{W}^{\text{in}}$ were drawn from a uniform

distribution $[-0.1, 0.1]$; the recurrent weights $\mathbf{W}^{\text{rec}}$ was first initialized with a gamma distribution $\texttt{gamma}(2, 0.0495)$ (mean 0.099, variance 0.0049) while ensuring all the diagonal values were zeros. Furthermore, we rescaled the recurrent weight matrix to obtain a spectral radius around 0.99 and ensured the sum of excitatory elements the same as the sum of the inhibitory elements to keep the E/I balance. Finally, the output weights $\mathbf{W}^{\text{out}}$ had a diagonal structure, with diagonal values drawn from a uniform distribution $[-0.1, 0.1]$ while keeping the nontarget neuron connections as zeros. We did not impose additional constraints on the connection weights.

The training procedure was the same as in Song et al. (2016) and briefly described here. We used the cost function

$$\begin{aligned}\mathcal{E} &= \frac{1}{N_{\text{trials}}} \sum_{n=1}^{N_{\text{trials}}} \mathcal{L}_n \\ &= \frac{1}{N_{\text{trials}} N_{\text{out}} N_{\text{time}}} \sum_{n=1}^{N_{\text{trials}}} \sum_{\ell=1}^{N_{\text{out}}} \sum_{t=1}^{N_{\text{time}}} M_{t\ell}^{\text{error}} \left[(\mathbf{z}_t)_\ell - (\mathbf{z}_t^{\text{target}})_\ell \right]^2, \end{aligned} \tag{2.10}$$

where the error mask $M_{t\ell}^{\text{error}}$ is a matrix of ones and zeros that determines whether the error in the ℓth output at time t should be taken into account. During training, we used the batch size as the same as $N_{\text{trials}} = 20$.

In addition, we imposed an L_2-norm regularization on the firing rates and the sum of L_1-norms of connection weights onto $\mathcal{E}$. We used the stochastic gradient descent (SGD) algorithm with default learning rate parameter. The gradient was computed effectively by backpropagation through time (BPTT). The vanishing gradient problem was alleviated using the regularization techniques discussed in Song et al. (2016). The RNN implementation was adapted based on the Python package (https://github.com/frsong/pycog), with modifications for task design, network architecture, and optimization. To avoid overfitting, in addition to the training trials, we also monitored the cost function of an independent validation data set. We concluded the trained network achieved convergence when the mean-squared error between the network output and the target out was sufficiently small or their correlation coefficient was sufficiently high (e.g., $R^2 > 0.95$). The standard choice of hyperparameters is shown in Table 1. However, the results reported here are not sensitive to the hyperparameter configuration.

2.3 Eigenspectrum Analysis. For a linear dynamical system (e.g., equation 2.8), the real part of the eigenvalue is responsible for the decay or growth of oscillation. Therefore, the dynamical system becomes unstable in the presence of even a single eigenvalue with a positive real component. However, for an RNN with ReLU activation, the analysis of dynamics is more complex (see appendixes A and B).

Table 1: Hyperparameters Used for RNN Training.

Parameter	Setup
L_1 weight regularization for weights	2
driven noise variance	$(0.01)^2$
input noise variance σ_{in}^2	$(0.01)^2$
gradient mini-batch size	20
validation mini-batch size	1000
unit time constant τ	50 ms
learning rate	0.01
max gradient norm	1

The stability of the dynamic system described by the excitatory-inhibitory RNN is strongly influenced by the recurrent connection matrix $\mathbf{W}^{\text{rec}}$ (Brunel, 2000; Rajan & Abbott, 2006). Since $\mathbf{W}^{\text{rec}}$ is generally asymmetric, and each column of the matrix has either all positive or all negative elements, it will produce a set of complex conjugate pairs of eigenvalues (possibly including purely real eigenvalues) from eigenvalue decomposition $\mathbf{W}^{\text{rec}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$, where the column vectors of matrix $\mathbf{U}$ are the eigenvectors and the diagonal elements of $\mathbf{\Lambda}$ denote the eigenvalues. Generally, $\mathbf{W}^{\text{rec}}$ is nonnormal (unless all submatrices are symmetric and EI and IE connections are identical); as a result, its eigenvectors are *not* orthogonal (Goldman, 2009; Murphy & Miller, 2009).

For the leakage matrix ($\mathbf{W}^{\text{rec}} - \mathbf{I}$), each eigenmode determines the corresponding response, and each mode is labeled by a complex eigenvalue, whose real part corresponds to the decay rate and imaginary part is proportional to the frequency of the oscillation. Take the delta pulse of input as an example: its response can be described by a sinusoid with an exponential envelope, where the real part of the eigenvalue λ of $\mathbf{W}^{\text{rec}}$ determines the rate of exponential growth or decay (Goldman, 2009). $\text{Re}\{\lambda\} > 1$ implies exponential growth; $\text{Re}\{\lambda\} < 1$ corresponds to exponential decay; $\text{Re}\{\lambda\} = 1$ denotes no decays (i.e., pure harmonic oscillatory component); and $\text{Re}\{\lambda\} = 0$ corresponds to decay with the intrinsic time constant. The magnitude of the imaginary part of eigenvalue $|\text{Im}\{\lambda\}|$ determines the frequency of the sinusoidal oscillation, with a greater value representing a faster oscillation frequency (see Figure 2a).

An alternative way to examine the network connectivity matrix $\mathbf{W}^{\text{rec}}$ is through the Schur decomposition, $\mathbf{W}^{\text{rec}} = \mathbf{Q}\mathbf{T}\mathbf{Q}^{-1}$, where $\mathbf{Q}$ is a unitary matrix whose columns contain the orthogonal Schur mode and $\mathbf{T}$ is an upper triangular matrix that contains the eigenvalues along the diagonal. The triangular structure of $\mathbf{T}$ can be interpreted as transforming an RNN into a feedforward neural network, and the recurrent matrix $\mathbf{W}^{\text{rec}}$ corresponds to a rotated version of the effective feedforward matrix $\mathbf{T}$, which defines self-connections and functionally feedforward connections (FFC) of the neural

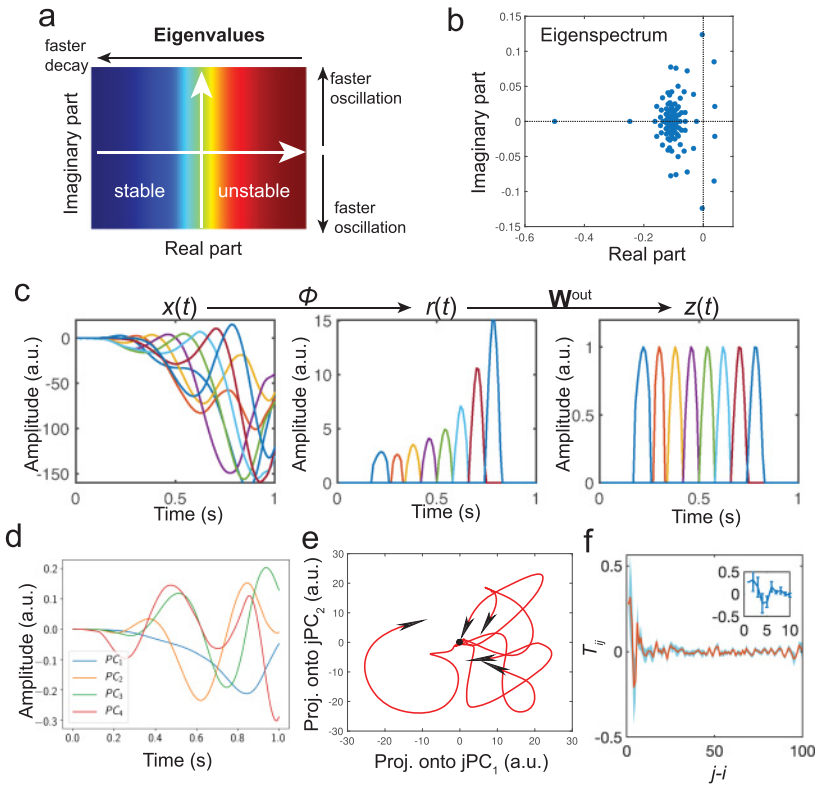

Figure 2: (a) Eigenvalues in the complex plane. (b) The eigenspectrum of $(\mathbf{W}^{\text{rec}} - \mathbf{I})/\tau$, with each dot representing the real or complex-valued eigenvalue (based on ReLU nonlinearity). (c) The neural state dynamics consisted of a set of driven harmonic oscillators aligned in time, with time-varying amplitude and frequency. Different colors represent individual components $\{x_i(t)\}$ or $\{r_i(t)\}$. The traces of $\{x_i(t)\}$ were first mapped to firing rates $\{\phi(x_i(t))\}$ and then mapped to the output sequence $\{z_\ell(t)\}$. (d) Extracted four principal components (PCs) derived from 100-dimensional $\mathbf{x}(t)$. (e) Rotation dynamics extracted from 100-dimensional $\mathbf{x}(t)$ during the trial period ([0, 1] s), with each trace representing one computer simulation trial. The origin represents the initial point, and the arrow indicates the time direction. (f) Applying the Schur decomposition to $\mathbf{W}^{\text{rec}}$ and visualizing the connection strengths of effective feedforward matrix $\mathbf{T} = \{T_{ij}\}$ (upper diagonal) with respect to the neuron index gap $j - i$. Shaded area shows SEM. The inset shows the first 10 elements.

network (Goldman, 2009). For a normal matrix, the Schur basis is equivalent to the eigenvector basis. However, unlike eigenvalue decomposition, the Schur decomposition produces the simplest (yet nonunique) orthonormal basis for a nonnormal matrix.

3 Results

3.1 Sequences Are Generated by Superposition of Driven Harmonic Oscillators. To avoid local minima, we trained multiple excitatory-inhibitory RNNs with different random seeds, hyperparameters, and initial conditions in parallel. In each condition, we trained at least five networks and selected the suboptimal solution with the smallest cost function. Depending on the learning rate parameter and the input-output setup, the excitatory-inhibitory RNN converged relatively slowly (typically $> 10^5$ epochs) to obtain a good solution.

Upon training the excitatory-inhibitory RNN, we examined the learned representation in the RNN dynamics. By examining the temporal trajectories of $\mathbf{x}$ and $\phi(\mathbf{x})$ that were associated with the readout neurons, we found that the RNN behaved approximately as a collection of driven (growing or damping) harmonic oscillators with time-varying frequencies (see Figure 2c). In the trained RNN, some readout neurons had large negative amplitudes in $x_i(t)$ because of the lack of constraint—this could be seen in the net excitatory and net inhibitory input to each target neuron (see equation 2.9). By examining the eigenspectrum plane of $(\mathbf{W}^{\text{rec}} - \mathbf{I})/\tau$ (see Figure 2b), the maximum imaginary frequency was around 0.125 radian/s (equivalently $0.125/2\pi \approx 0.02$ Hz), which produced one oscillator at the same timescale of time constant $\tau = 50$ ms. Each generalized harmonic oscillator was associated with an eigenmode that has nonzero imaginary parts regardless of whether the real part is positive, negative, or zero. The maximum peaks of the harmonic oscillators were temporally aligned to form a neural sequence. The peaks and troughs of the generalized harmonic oscillators were controlled by the time-varying excitation and inhibition levels in the RNN, and the spacing between the neighboring peaks determined the span of sequential activation.

In the case of RNN dynamics, the neural state will become unstable if all components of $x_i(t)$ are positive simultaneously (i.e., all neurons fire together). Since this phenomenon rarely happens, the RNN could reach stable dynamics when a few of the $x_i(t)$ components were positive (i.e., neurons fire sparsely). In general, it is difficult to derive mathematical solutions to high-dimensional time-varying nonlinear dynamical systems (see equation 2.1). In the case of RNN with threshold-linear (ReLU) units, the system is piecewise linear, and we can derive the analytic solutions of two and three-dimensional systems (see appendixes A and B, respectively). In these low-dimensional systems, the phase space contains regions where neuronal activation is either linear or zero. Importantly, the system is time-varying, and the stability of RNN dynamics may change from moment to moment in the phase space. However, it may be possible to construct dynamic trajectories, such as fixed points or periodic orbits in higher-dimensional systems by patching the eigenmodes across the "switch boundaries" (although we did not implement the patching procedure, only fixed points but not

periodic orbits were found in our simulations of the two- and three-dimensional systems; see appendixes A and B).

In light of principal component analysis (PCA), we found that the 100-dimensional neural state trajectory $\mathbf{x}(t)$ was usually embedded in a lower-dimensional space (see Figure 2d), with five to six components explaining more than 90% variance. Each principal component (PC) behaved like an oscillator that was intrinsically related to the presence of complex eigenvalues in $\mathbf{W}^{\text{rec}}$. To further examine the oscillator structure in the simulated neural dynamics, we also applied the polar decomposition (see appendix C) to high-dimensional $\mathbf{x}(t)$ (similar to the rotation jPCA method as shown in Churchland et al., 2012) and visualized the two-dimensional projection in the rotation plane (see Figure 2e). Interestingly, we also found rotation dynamics on the single simulation trial basis while projecting the data onto the eigenvectors associated with jPCA.

In general, it is difficult to infer any structure from the eigenspectrum of $\mathbf{W}^{\text{rec}}$ alone. We further applied the Schur decomposition to $\mathbf{W}^{\text{rec}}$ and computed the effective feedforward matrix $\mathbf{T}$, which is an upper diagonal matrix (see section 2.3). To examine the relative feedforward connection strengths between neurons, we plotted T_{ij} with respect to the neuron index difference $(j - i)$. Interestingly, we found relative large strengths between the first eight Schur modes (see Figure 2f). Furthermore, on average, each mode excited its nearby modes but inhibited modes that were farther away. Because of Dale's principle, $\mathbf{W}^{\text{rec}}$ is nonnormal; the nonnormal dynamics in RNNs may offer some benefits such as extensive information propagation and expressive transients (Ganguli, Hug, & Sompolinsky, 2008; Kerg et al., 2019; Orhan & Pitkow, 2020).

3.2 Robustness to Tested Input and Time Warping. Once the RNN was trained, we found that the output sequences are robust to qualitative modifications to the original input signal. Specifically, the same output sequences were produced in response to different input signals, with varying forms of amplitude, frequency, or waveform (see examples in Figure 3a).

We further tested the generalization or invariance of RNNs in sequence representation with respect to temporal scaling of the inputs. Specifically, we trained the RNN in a task of mapping five different inputs to five different scaled versions of output sequences. Each sequence was represented by sequential activation of five excitatory neurons. The five input signals were temporally scaled versions of each other. Similarly, the five output sequences were also temporally scaled versions of each other (see Figure 3b). We then tested whether the trained RNN can learn to interpolate or extrapolate in response to the unseen scaled version of input signals. In each test condition, we created five random realizations and computed the error bar for each output sequence duration. The results are shown in Figures 3b and 3c. Specifically, the RNN generalized very well by an approximately linear interpolation but extrapolated poorly in the case of

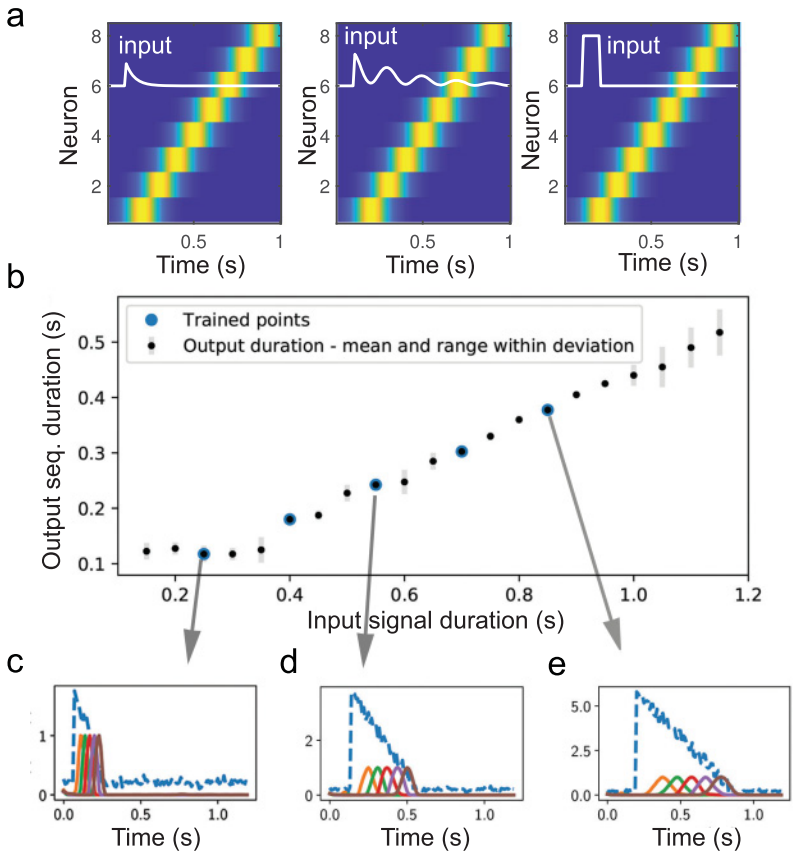

Figure 3: (a) In the testing phase, the neural sequence out of the trained excitatory-inhibitory RNN was robust to the exact waveform or amplitude of the tested input signal (white trace in the inset). The RNN was trained on one input signal and tested on the other two input signals. (b) Temporal scaling invariance of the sequences. Five input-output pairs with different durations were trained together. Each input-output pair is a temporally scaled version of each other (i.e., 25%, 40%, 55%, 70%, and 85% scaled versions of the original 1 s duration). In each case, the durations of input and output were scaled by the same scaling factor. Three scaled versions (25%, 55% and 85%) are illustrated in panels c, d, and e, respectively. The dashed blue trace represents a noisy version of $u(t)$, and the remaining solid traces represent the sequential activations of five output neurons: $z_{1-5}(t)$. We trained the excitatory-inhibitory RNN five times, each with a different random seed. The mean of the output duration for each scaling factor was shown by the black dots in panel a, and the actual output duration was highlighted with a larger blue dot. The range (mean $\pm$ SD) of the RNN output duration for each tested scaling factor is shown by a gray error bar.

short-scaled input (e.g., 0.1 and 0.2). Our finding here was also consistent with the recent results of RNN without biological constraints, in which temporal scaling of complex motor sequences was learned based on pulse inputs (Hardy, Goudar, Romero-Sosa, & Buonomano, 2018) or on time-varying sensory inputs (Goudar & Buonomano, 2018).

3.3 Limit Cycle in the Steady State. After examining the RNN dynamics during the task period, we further investigated the dynamics of trained RNNs in the steady state: for a long-duration, stimulus-free time course well beyond the duration of a trained sequence. During the steady state, the RNN dynamics was driven by spontaneous and recurrent network activity. If one or more growing complex eigenmodes are activated, the network may exhibit associated spontaneous oscillations. Interestingly, we found that the RNN sequence generator was able to produce a stable high-dimensional dynamic attractor (see Figure 4a). That is, the neural dynamics $x_i(t)$ and $x_k(t)$ $(k \neq i)$ form a constant phase shift in their activation strengths, thereby forming a limit cycle and seen as a closed orbit in two-dimensional projections of phase portrait (see Figure 4b). Letting $P_{\text{ss}}(\mathbf{x})$ denote steady-state probability distribution of $\mathbf{x}$, we define a potential function of the nonequilibrium system as follows:

$$\mathcal{U}(\mathbf{x}) = -\log P(\mathbf{x}, t \to \infty) = -\log P_{\text{ss}}. \tag{3.1}$$

Notably, the limit cycle was embedded not only in the eight sequence readout neurons (i.e., $z_1(t)$ through $z_8(t)$), but also in the remaining 92 neurons. To visualize the low-dimensional representations, we applied PCA to 100-dimensional $\mathbf{x}(t)$ during the training trial period ([0,1] s) and then projected the steady-state activations onto the dominant four-dimensional PC subspaces. Again, we observed stable limit cycles in the PC subspace.

Since the limit cycle was embedded in the high-dimensional neural state dynamics, it is difficult to derive theoretical results based on any existing mathematical tools (e.g., the Hopf bifurcation theorem). The dynamic attractors in the steady state are fully characterized by equation 2.1 and the $N \times N$ matrix $\mathbf{W}^{\text{rec}}$. To date, limited analysis of periodic behavior was obtained for the RNN, except for two-dimensional cases (Jouffroy, 2007; Bay, Lepsoy, & Magli, 2016). To understand the conditions of limit cycles in the excitatory-inhibitory RNN, we decomposed $\mathbf{W}^{\text{rec}}$ into a symmetric and a skew-symmetric component and made further low-dimensional approximation (see appendix D). Specifically, the symmetric matrix produces a spectrum with only real eigenvalues, whereas the skew-symmetric matrix produces a spectrum with only imaginary eigenvalues. Furthermore, we projected the N-dimensional $\mathbf{x}(t)$ onto a two-dimensional eigenvector space. Under some approximations and assumptions, we could numerically simulate limit cycles based on the learned weight connection matrix $\mathbf{W}^{\text{rec}}$ (Susman, Brenner, & Barak, 2019).

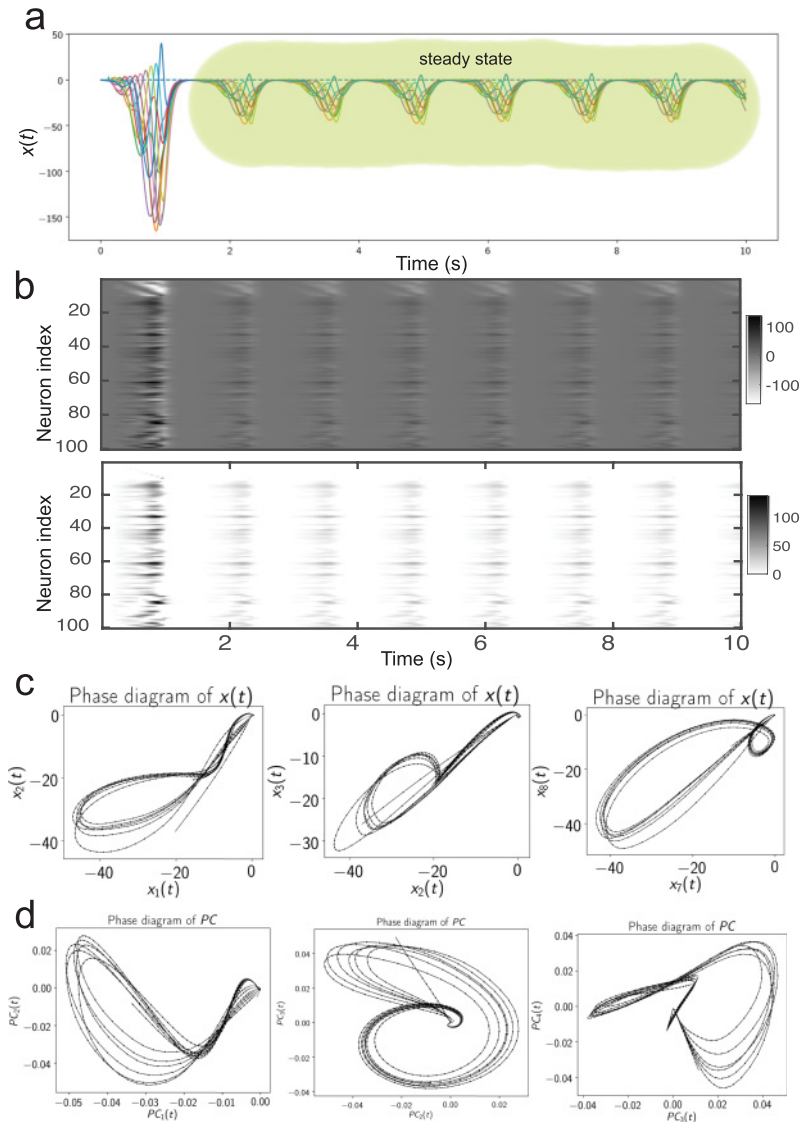

Figure 4: A trained excitatory-inhibitory RNN produced a limit cycle attractor in the steady state. (a) Output sequence shown as a heat map during $t \in [0, 10]$ s, where the shaded area indicated the steady state. (b) Heat maps of neural activity $\mathbf{x}(t)$ (top panel) and firing rate $\mathbf{r}(t)$ (bottom panel) of 100 neurons. (c) Illustration of two-dimensional phase diagrams at different subspaces of $\{x_1(t), x_2(t), x_3(t), x_7(t), x_8(t)\}$. (d) Illustration of two-dimensional phase diagrams at the vector spaces spanned by the first four dominant principal components (PCs): $\{PC_1(t), PC_2(t), PC_3(t), PC_4(t)\}$.

Limit cycle attractors were frequently observed in our various testing conditions. During training, we varied the sequence duration, the input waveform, and the number of neurons engaging in the output sequence; we frequently found a limit cycle attractor in the steady state. By observing equation 2.9, it is noted that the relative E/I balance is required for the reactivation of sequence output: too much inhibition would suppress the neuronal firing because of below-threshold activity. For each $x_i(t)$, if the level of net inhibition was greater than the level of net excitation, the ith neuron would not be able to fire even though the memory trace was preserved in the high-dimensional attractor.

3.4 Impact of $\mathbf{W}^{\text{rec}}$ on Dynamic Attractors. The stability of the limit cycle attractor can be studied by theoretical analysis or numerical (perturbation) analysis. To study how the change in $\mathbf{W}^{\text{rec}}$ affects the stability of dynamic attractors, we conducted a perturbation analysis on the connection weights. First, we scaled the recurrent weight matrix according to the functional blocks as follows, $\begin{bmatrix} a_1\mathbf{W}^{\text{rec}}_{\text{EE}} & a_2\mathbf{W}^{\text{rec}}_{\text{IE}} \\ a_3\mathbf{W}^{\text{rec}}_{\text{EI}} & a_4\mathbf{W}^{\text{rec}}_{\text{II}} \end{bmatrix}$, where $\mathbf{W}^{\text{rec}}_{\text{EE}}$ defines the excitatory-to-excitatory connectivity, $\mathbf{W}^{\text{rec}}_{\text{IE}}$ defines the inhibitory-to-excitatory connection strength, and $\mathbf{W}^{\text{rec}}_{\text{II}}$ defines the disinhibition among inhibitory neurons; $\{a_1, a_2, a_3, a_4\}$ are four positive scalars. A scaling factor greater (or smaller) than 1 implies scaling up (or down) the degree of excitation and inhibition. Specifically, we found that the limit cycle dynamics was robust with respect to EI and II scaling and was most sensitive to the EE scaling (see Figure 5b). Scaling the whole matrix together would scale up or down the eigenspectrum, thereby affecting the stability. These results suggest that the excitatory-inhibitory RNN is only stable with balanced E/I, the excitatory network by itself is unstable, and stabilized only by the feedback inhibition. Additionally, when scaling down EE connections, we found that the amplitude of the limit cycle (i.e., $x(t)$ activation amplitude) reduced with decreasing scaling factor a_1 (see Figure 5c), until converging to a fixed point, suggesting that emergence of limit cycle behavior is via a Hopf bifurcation of the network.

Next, we imposed a sparsity constraint onto the learned $\mathbf{W}^{\text{rec}}$. We randomly selected a small percentage of connection weights and set them zeros and examined the impact on the dynamic attractor. Since $\mathbf{W}^{\text{rec}}_{\text{EE}}$ was most sensitive to the dynamics, we gradually increased the percentage from 5%, to 10%, 15%, and 20%, and set randomly chosen entries of $\mathbf{W}^{\text{rec}}_{\text{EE}}$ to zeros. As a result, the neural sequential activation was reduced during both intask and steady-state periods (see Figure 5d). Notably, the limit cycle was still present even when 5% to 15% of $\mathbf{W}^{\text{rec}}_{\text{EE}}$ elements were zeros, suggesting that the dynamic attractor was robust to sparse coding. Notably, the amplitude of $\mathbf{x}(t)$ gradually decreased with increasing sparsity level. When 20% of $\mathbf{W}^{\text{rec}}_{\text{EE}}$ elements were zeros, the limit cycle converged to a fixed point.

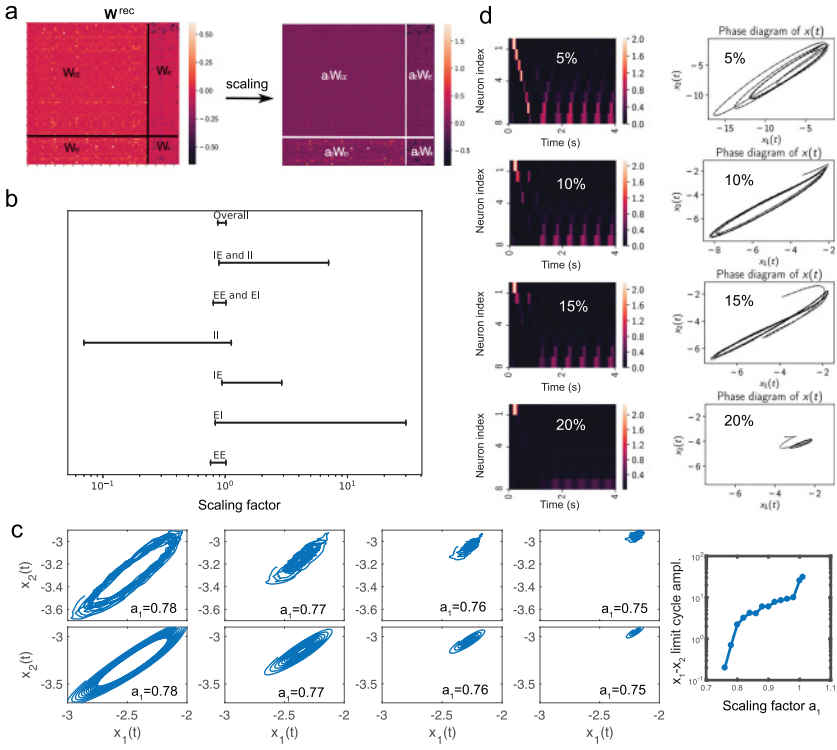

Figure 5: Robustness of recurrent connectivity matrix that preserved the limit cycle attractor in the steady state. (a) The recurrent connectivity matrix $\mathbf{W}^{\text{rec}}$ was divided into four blocks. The four submatrices $\mathbf{W}^{\text{rec}}_{\text{EE}}$, $\mathbf{W}^{\text{rec}}_{\text{IE}}$, $\mathbf{W}^{\text{rec}}_{\text{EI}}$, and $\mathbf{W}^{\text{rec}}_{\text{II}}$ were scaled with four scaling factors a_1, a_2, a_3, and a_4, respectively. We examined the range of these scaling factors so that the dynamic attractor remained stable as a limit cycle and beyond which the dynamic attractor became unstable or converged to a fixed point. (b) The range of scaling factor that the limit cycle was preserved under different conditions. EE: a_1 was varied while fixing $a_2 = a_3 = a_4 = 1$. IE: a_2 was varied while fixing $a_1 = a_3 = a_4 = 1$. EI: a_3 was varied while fixing $a_1 = a_2 = a_4 = 1$. II: a_4 was varied while fixing $a_1 = a_2 = a_3 = 1$. EE and EI: $a_1 = a_3$ were varied together while fixing $a_2 = a_4 = 1$. IE and II: $a_2 = a_4$ were varied together while fixing $a_1 = a_3 = 1$. Overall: $a_1 = a_2 = a_3 = a_4$ were varied together. (c) The amplitude of limit cycle gradually decreased with decreasing scaling factor a_1 of $\mathbf{W}^{\text{rec}}_{\text{EE}}$. Only the phase diagram of $\{x_1(t), x_2(t)\}$ is shown for illustration, but similar observations also held for other state variable pairs. Top and bottom rows correspond to the noisy and noiseless conditions, respectively. (d) The transient and steady-state responses while setting the sparsity of $\mathbf{W}^{\text{rec}}_{\text{EE}}$ to 5%, 10%, 15%, and 20%. The heat maps of $z_{1:8}(t)$ and phase portraits $x_1(t)$ versus $x_2(t)$ are shown in the left and right columns, respectively.

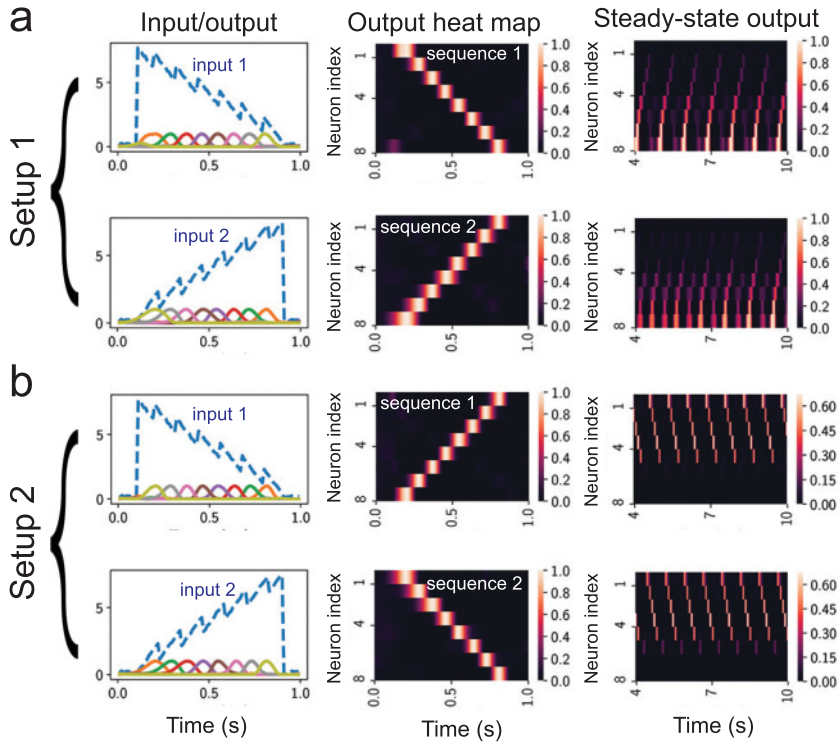

Figure 6: Computer setup for learning two neural sequences. The first column shows the two input-output patterns $\{u(t), z_{1-8}(t)\}$. Note that inputs 1 and 2, as well as output sequences 1 and 2 are mirror images of each other. The second column shows the $\{z_{1-8}(t)\}$ in the normalized heat maps. The third column shows the heat maps of $\{z_{1-8}(t)\}$ in the steady-state period. (a) Setup 1. (b) Setup 2: swapped input-output pairs of setup 1.

3.5 Extension to Learning Two Sequences. RNNs have been also used to learn multiple sequences (Rajan et al., 2016; Hardy & Buonomano, 2018). Next, we extended the RNN task from learning one sequence to two sequences. For simplicity, we used a mirror image setup such that the same amplitude or power was shared between two inputs and between two outputs (see Figure 6). In setup 1, we mapped two input signals to two equal-duration outputs, one forward-ordered and one reverse-ordered sequence, using the same readout neurons. The input signal consisted of a combination of linear input and a sawtooth component, plus additive gaussian noise. In setup 2, we reversed the input-output mapping and swapped the output sequence 1 and output sequence 2. In both setups, we used $N_{\text{trials}} = 20$ for each input (total 40 trials) and added the variability (such

as the sinusoidal frequency of the input, the timing of the trigger, and network noise) at each trial. After training the RNN, we examined the transient and steady-state responses of eight readout neurons.

During the transient response period ([0, 1] s), similar to the previously demonstrated single-sequence condition, the RNN showed robust sequence responses and was insensitive to the exact input waveform. In order to examine the effect of the input magnitude on the trained RNN, we fed DC input signals with different levels for a duration of 0.8 s (see Figure 7a) and observed the network output. Depending on the magnitude of DC input, the output sequence was a superposition of two output sequences (see Figures 7b and 7i). This result suggests that the bimodal RNN output was determined by the level of internal excitability in $\mathbf{x}(t)$, which was contributed from the DC input (see equation 2.1). At a certain level, the system reached a state that simultaneously produced two sequence outputs (see Figures 7e, 7f, and 7j). The coactivation of two sequences suggests a symmetric (or asymmetric) double-well potential landscape, and the biased activation of one of two sequences is influenced by the control DC input, suggesting a kind of biased competition. The intermediate cases appear as transients during the competition. To visualize the N-dimensional dynamics, we projected $\mathbf{x}(t)$ onto the two-dimensional jPCA subspace, and observed an ∞-shaped trajectory (see Figure 7k for level 3). The change in spiraling direction (first clockwise, then counterclockwise) of the phase portrait reflected the push-pull force in the bistable state.

Interestingly, the steady-state output with zero input showed only one of the output sequences (sequence 2 in both setups shown in Figure 6, where $\mathcal{U}(\mathbf{x}_{\text{seq2}}) < \mathcal{U}(\mathbf{x}_{\text{seq1}})$ in the potential landscape). To investigate the tolerance of the dynamic attractor to interference, we further fed the variants of input signals to the trained excitatory-inhibitory RNN at random timing during the steady-state period. It was found that the reactivation of output sequence could be triggered by a transient excitatory input, leveling up the excitability in $\mathbf{x}(t)$. Interestingly, the transient input triggered one of two sequences and then switched back to the stable state (see Figures 8a and 8b), suggesting that the trained RNN was not in a state with bistable attractors. In our computer simulations and perturbation analyses, we didn't observe the coexistence of two stable limit cycles, yet the single stable limit cycle showed a high degree of robustness to various transient input signals. We may envision two periodic orbits, one stable and the other unstable, coexisting in the system (see Figure 8d). One such example is described by the following two-dimensional system equations,

$$\frac{dx_1}{dt} = x_1(x_1^2 + x_2^2 - 1)(x_1^2 + x_2^2 - 2) - x_2,$$
$$\frac{dx_2}{dt} = x_2(x_1^2 + x_2^2 - 1)(x_1^2 + x_2^2 - 2) + x_1,$$

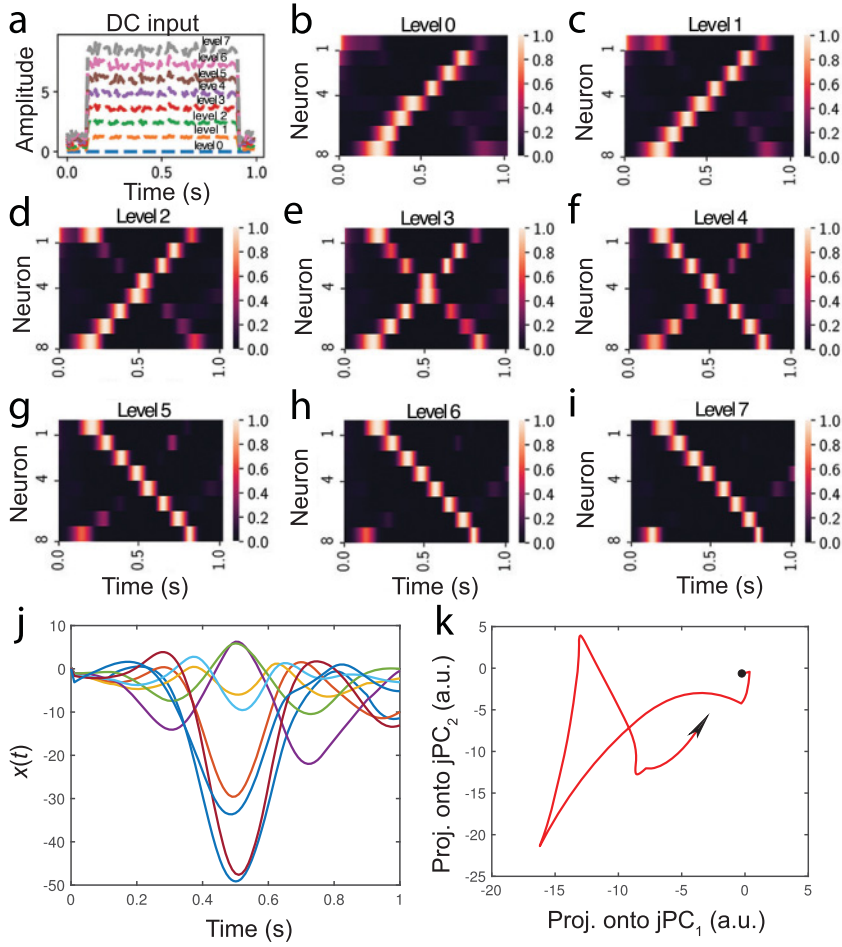

Figure 7: Readout neuronal sequences change in response to different levels of the DC input during the transient response period ([0, 1] s). DC input was sent to the trained RNN using setup 1 shown in Figure 6. (a) Illustration of eight different levels of DC inputs (the mean input $u(t) \in \{0, 1, 2, \ldots, 7\}$) for $t \in [0.1, 0.9]$ s). (b)–(i) Normalized heat maps of output sequences $\{z_1(t), z_2(t), \ldots, z_8(t)\}$ in response to varying levels of DC input. Note that the transition occurred at levels 3 and 4, where both sequences were clearly visible. (j) Activation of $\{x_1(t), x_2(t), \ldots, x_8(t)\}$ at level 3. (k) Two-dimensional projection of single-trial $\mathbf{x}(t)$ activation at level 3 onto the first two PC subspaces from jPCA. The origin represents the initial point, and the arrow indicates the time direction.

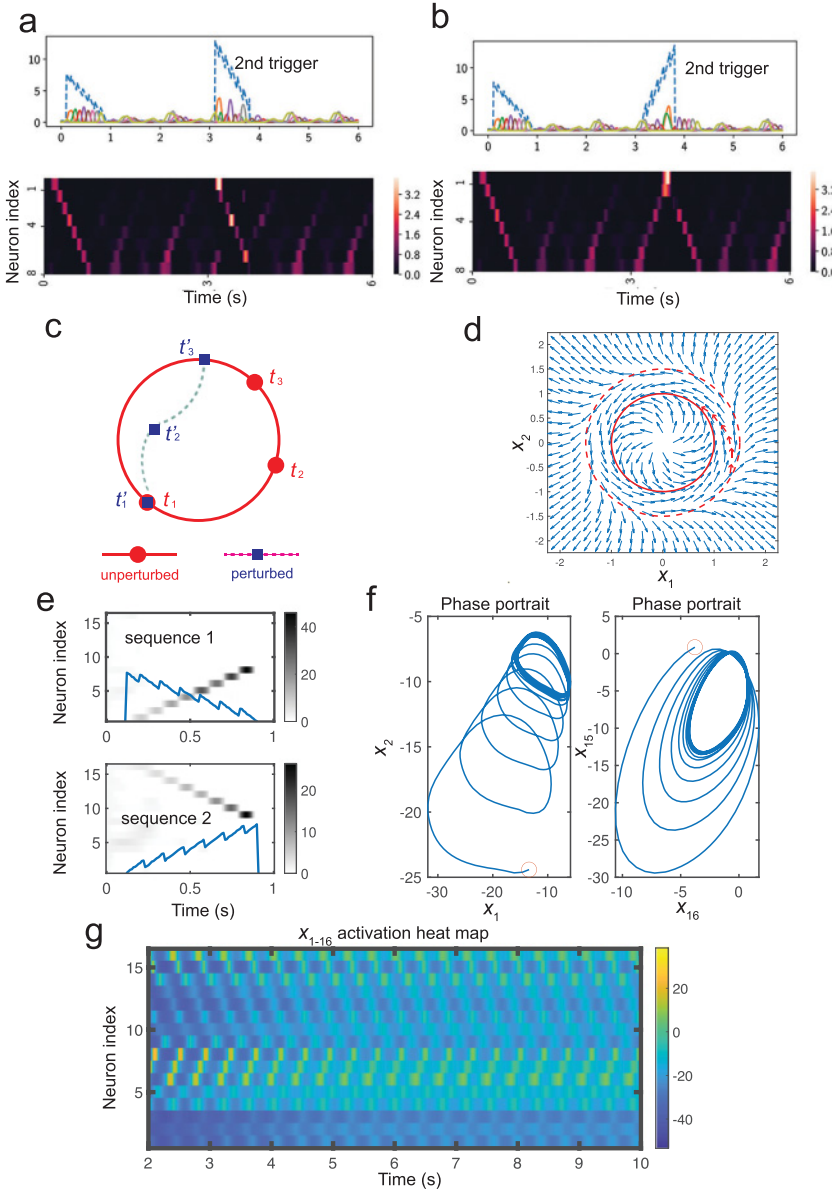

Figure 8: (a) Two test input signals were presented at two periods ([0, 1] s and [3, 4] s, respectively) for the pretrained RNN. The eight readout neurons' activations $\{r_{1-8}(t)\}$ were shown by the heat map at the bottom. Note that the second input triggered the learned forward sequence (at around 3 s) and then switched back to the stable reverse sequence in the steady state. (b) The

which contain two periodic orbits (the circle $x_1^2 + x_2^2 = 1$ is stable, but the outer circle is unstable); the trajectories in between are repelled from the unstable cycle and attracted to the stable one.

In both setups 1 and 2, the output of readout neurons (units #1–8) were orthogonal to each other for representing two sequences. Next, we considered a different setup (setup 3), in which we set $N^{\text{out}} = 16$, where units #1–8 represented the output of sequence 1, and units #9–16 represented the output of sequence 2 (see Figure 8e). Therefore, readout neurons have orthogonal representations for these two sequences. Upon learning these two sequences, we again examined the phase portraits of $\mathbf{x}(t)$ during the steady state. Interestingly, in this new setup, the RNN dynamics also exhibited a stable limit cycle attractor (see Figure 8f). This stable limit cycle attractor simultaneously represented the partial forward- and reverse-ordered sequences, with opposite rotation directions in the phase portrait (see Figures 8f and 8g). Therefore, the two orthogonal output sequences learned independently can be coactivated in the steady state, and any two-dimension combination from $\{x_1, \ldots, x_{16}\}$ would also form a periodic oscillator in the two-dimensional phase space. However, the coactivation of forward and reverse sequences still appeared as a single limit cycle (with a shorter period compared to Figures 8a and 8b).

Finally, we investigated whether the period of limit cycle attractor changed with respect to different inputs. First, using the RNN shown in Figure 6 as an example, we fed various input signals to the trained RNN and found qualitatively similar limit cycles (see Figure 9a), suggesting the invariance property of the RNN-encoded limit cycle with respect to the new input. Second, using the temporally scaled inputs (as shown in Figure 3), we fed the various scaled input signals to the trained RNN and again found

---

second input triggered a mixed representation of forward sequence and reverse sequence (at around 3.6 s), and then switched back to the stable reverse sequence. (c) Schematic illustration of perturbation of a stable limit cycle. Time trajectory $t_1 \to t_2 \to t_3$ progresses on the limit cycle of a unperturbed system, whereas a perturbed system is driven away from the limit cycle ($t_1' \to t_2' \to t_3'$) by an impulsive signal between t_1' and t_2' and then relaxes back to the limit cycle. (d) Phase portraits of two periodic orbits, one being stable (red solid line) and the other unstable (red dashed line). The arrows represent the vector field $(\dot{x}_1, \dot{x}_2)$. (e) Computer simulation setup 3: Units #1–8 represent sequence 1 activation, whereas units #9–16 represent sequence 2 activation. Heat maps show the $r(t)$ activation upon completion of RNN training. Blue traces represent the corresponding input $u(t)$. (f) Phase portraits of limit cycle attractor in the steady state with $u(t) = 0$. The phase portrait x_1 versus x_2 displays a limit cycle projection with clockwise direction, whereas the phase portrait x_{16} versus x_{15} displays a limit cycle projection with counterclockwise direction. Open circles represent the initial point. (g) Heat map of $x_{1-16}(t)$ during the steady-state period.

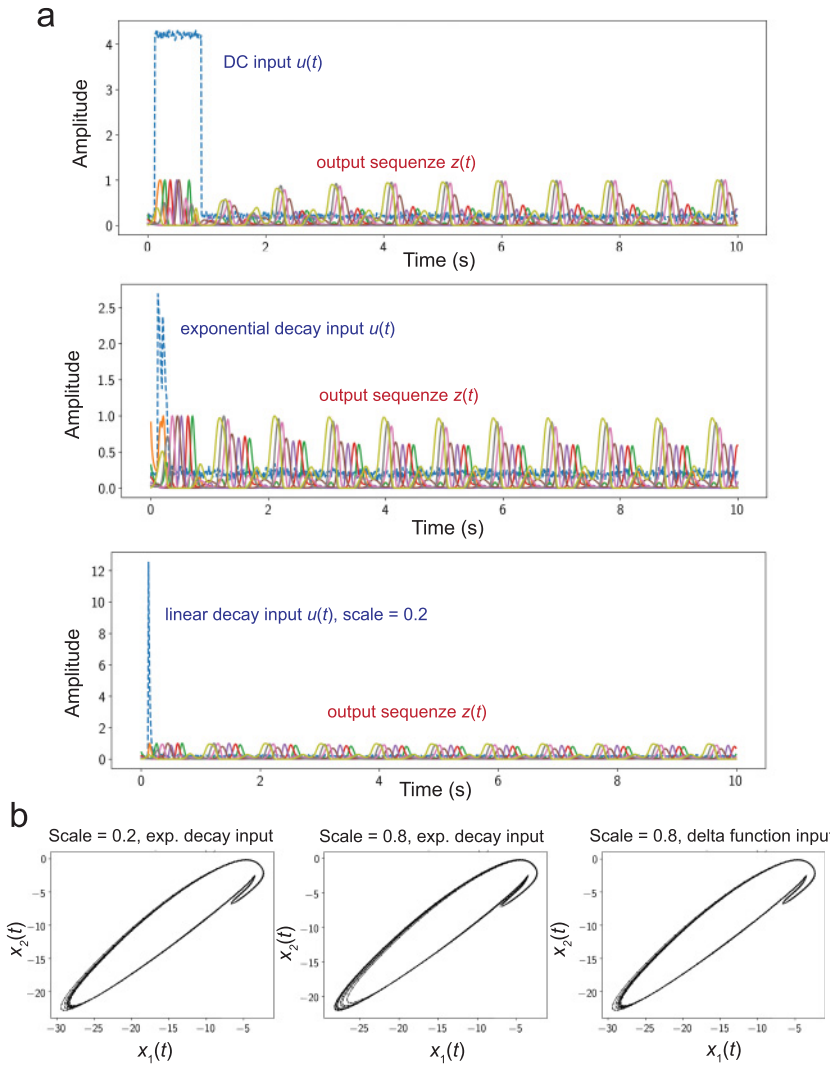

Figure 9: (a) Three examples of $u(t)$ and sequence output $\{z(t)\}$ during the task period ([0, 1] s) and steady state ([1, 10] s) for the RNN shown in Figure 6. (b) Three examples of phase diagrams of $x_1(t)$ versus $x_2(t)$ during the steady state under different task input settings for the RNN shown in Figure 3.

the limit cycle was invariant to the task input signal's period and waveform (Figure 9b). Together, these results suggest that once the excitatory-inhibitory RNN was trained, the dynamic attractor properties during the steady state were fully determined by the eigenspectrum of $\mathbf{W}^{\text{rec}}$, and the

period of limit cycle attractor was invariant to the amplitude, waveform, or frequency of the transient input signals.

4 Discussion

4.1 Relation to Other Work. RNNs are capable of generating a rich class of dynamical systems and attractors (Trischler & D'Eleuterio, 2016; Pollock & Jazayeri, 2020). Dynamics of excitatory-inhibitory RNN has been studied in the literature (Brunel, 2000; Murphy & Miller, 2009). Murphy and Miller (2009) used mean field theory to study the linear recurrent dynamics of excitatory-inhibitory neuronal populations and studied the steady-state response with balanced amplification mechanisms. However, since their neural network is linear, no limit cycle attractor can arise from the assumed linear dynamical system. To model neural sequences, Gillett, Pereira, and Brunel (2020) used mean field theory to derive a low-dimensional description of a sparsely connected linear excitatory-inhibitory rate (and spiking) network. They found that a nonlinear temporally asymmetric Hebbian learning rule can produce sparse sequences.

Most RNN modeling work for brain functions has focused on representations and dynamics during the task period (or transient state), yet has ignored the stimulus-absent steady state. It remains unclear how the steady-state dynamics are related to the task-relevant dynamics and how they further contribute to the task. In the literature, the analysis of limit cycles for nonlinear RNN dynamics has been limited to two-dimensional systems (Jourffroy, 2007; Bay et al., 2016; Trischler & D'Eleuterio, 2016). To our best knowledge, no result was available for the existence of excitatory-inhibitory RNN in a high-dimensional setting. Our computer simulation results have shown the emergence of limit cycles from the trained excitatory-inhibitory RNN while learning the sequence tasks. However, no theoretical results have been derived regarding the sufficient or necessary condition. Our computer simulations were built on supervised learning and gradient descent for learning neural sequences, which is close in spirit to prior work (Namikawa & Tani, 2009) that they used an Elman-type RNN structure to learn multiple periodic attractors and the trained RNN could embed many attractors given random initial conditions.

It is possible to show the existence of periodic orbits or limit cycles in a content-addressable memory (CAM) network with an asymmetric connection weight matrix, where the neural firing patterns are binary (Folli, Gosti, Leonetti, & Ruocco, 2018; Susman et al., 2019). However, the CAM networks in previous studies did not consider temporal dynamics.

Upon learning neural sequences, we found that the excitatory-inhibitory RNN exhibited a rotation dynamics in the population response (by jPCA). This rotation dynamics can be robustly recovered in the single-trial data. Our finding is consistent with a recent independent finding while reexamining the neuronal ensemble activity of the monkey's motor cortex

(Lebedev et al., 2019). In our case, the rotation dynamics analysis was applied to the complete population, whereas Lebedev's study was applied only to the sequence readout neurons. Notably, it was shown that rotational patterns can be easily obtained if there are temporal sequences in the population activity (Lebedev et al., 2019).

4.2 Dynamic Attractors and Memories. Synaptic efficacy is fundamental to memory storage and retrieval. The classical RNN used for associative memory is the Hopfield network, which can store the patterns via a set of fixed points. The weight connection matrix in the Hopfield network is symmetric, thereby yielding only real positive eigenvalues. However, memories stored as time-varying attractors of neural dynamics are more resilient to noise than fixed points (Susman et al., 2019). The dynamic attractor is appealing for storing memory information as it is tolerant to interference; therefore, the limit cycle mechanism can be useful to preserve time-varying persistent neuronal activity during context-dependent working memory (Mante, Sussillo, Shenoy, & Newsome, 2013; Schmitt et al., 2017).

In the excitatory-inhibitory RNN, the stability of a dynamic system is defined by the recurrent connectivity matrix, which can further affect Intrinsically generated fluctuating activity (Mastrogiuseppe & Ostojic, 2017). According to dynamical systems theory, stability about a set-point is related to the spectrum of the Jacobian matrix (i.e., local linear approximation of the nonlinear dynamics) (Sussillo & Barak, 2013). For a general Jacobian matrix, the spectrum consists of a set of complex eigenvalues. The real part of this spectrum defines the system's stability: a system is stable only if all eigenvalues have negative real parts, whereas the imaginary part of the spectrum determines the timescales of small-amplitude dynamics around the set-point, but not stability (Susman et al., 2019).

In the special case, the unconstrained RNN (i.e., without imposing Dale's principle) can be approximated and operated as a line attractor (Seung, 1996), although the approximation of line attractor dynamics by a nonlinear network may be qualitatively different from the approximation by a linear network. Additionally, strong nonlinearity can be used to make memory networks robust to perturbations of state or dynamics. Linearization around some candidate points that minimize an auxiliary scalar function $|G(\mathbf{x})|^2$ (kinematic energy) may help reveal the fixed and slow points of high-dimensional RNN dynamics (Sussillo & Barak, 2013; Kao, 2019; Zhang et al., 2021). However, efficient numerical methods to analyze limit cycle attractors of RNNs remain unexplored.

It has been found that our trained excitatory-inhibitory RNN can trigger the neural sequence based on a wide range of input signals, representing one possible mechanism of memory retrieval. This is analogous to remembering a sequence of a phone number triggered by a name, where the form of name presentation may vary. The sequence readout is also robust to noise and system perturbation. During the steady state, the steady-state RNN

dynamics displayed either forward or reverse sequences, reminding us of rodent hippocampal memory replay during the offline state (Foster & Wilson, 2006; Diba & Buzsaki, 2007).

Multistability is an important property of RNNs (Cheng, Lin, & Shih, 2006; Ceni, Ashwin, Livi, & Oostlethwaite, 2020). For a general class of N-dimensional RNN with (k-stair) piecewise linear activation functions, RNNs can have $(4k-1)^N$ equilibrium points, and $(2k)^N$ of them are locally exponentially stable (Zeng, Huang, & Zheng, 2010). Recently, it has been shown that a two-dimensional gated recurrent units (GRU) network can exhibit a rich repertoire of dynamical features that includes stable limit cycles, multistable dynamics with various topologies, Andronov-Hopf bifurcation, and homoclinic orbits (Jordan, Sokol, & Park, 2019). However, their RNN allows self-connections and does not impose excitatory-inhibitory constraints. While training the excitatory-inhibitory N-dimensional RNN to learn two sequences, we found that memory retrieval is dominated by a single memory pattern within one stable limit cycle attractor (see Figures 8a and 8b). However, reactivation of two sequence patterns can be obtained by using orthogonal neuronal representations (see Figures 8e to 8g). The topics of coexistence of bi- or multistability and the mechanism of bistable switching between two limit cycles or between a limit cycle and fixed points would require further theoretical and simulation studies.

4.3 Hebbian Plasticity versus Backpropagation. Supervised learning algorithms have been widely used to train RNNs (Werbos, 1988; Jaeger, 2001; Maass, Natschläger, & Markram, 2002; Sussilo & Abbott, 2009; Song et al., 2016). We have used the established BPTT algorithm to train the excitatory-inhibitory RNN, where synaptic modification was applied to all synaptic connections based on gradient descent. In addition, it is possible to introduce a partial synaptic weight training mechanism on the recurrent weight matrix (Rajan et al., 2016; Song et al., 2016).

The supervised learning algorithms (e.g., BPTT or FORCE) assume an error-correction mechanism (Werbos, 1988; Sussilo & Abbott, 2009). However, how error backpropagation mechanisms can be implemented in the brain remains debatable (Lillicrap et al., 2020). In contrast, Hebbian plasticity represents an alternative yet more biologically plausible rule for learning sequential activity in the brain (Fiete, Senn, Wang, & Hahnloser, 2010; Gillett et al., 2020). More important, the brain is likely to use a continual learning strategy for multiple tasks (Kirkpatrick et al., 2017; Zenke, Poole, & Ganguli, 2017; Yang et al., 2019; Duncker, Driscoll, Shenoy, Sahani, & Sussillo, 2020), such as multiple sequences. Distributing resources of synaptic weights and readout neurons in RNNs is crucial for continual learning. Recently, it has been shown that neural sequences in the mouse motor cortex could reactivate faster during the course of motor learning (Adler et al., 2019). Eigenvalue analysis of recurrent weight matrix $\mathbf{W}^{\text{rec}}$ can provide new insight into such neural plasticity (see Figure 10 for a toy example illustration).

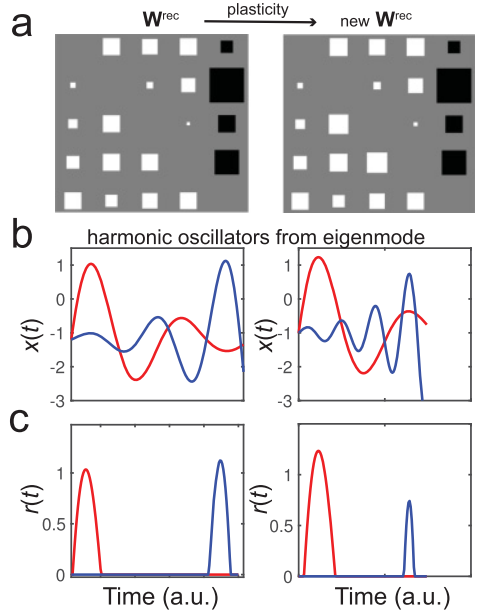

Figure 10: (a) Hinton diagram of 5×5 connection matrix $\mathbf{W}^{\text{rec}}$ for one inhibitory and four excitatory neurons. The white and black represent the positive and negative value, respectively. The size of the square is proportional to the absolute value of the connection weights. The eigenspectrum of $\mathbf{W}^{\text{rec}}$ consists of one real and two pairs of complex conjugate eigenvalues $\{-0.99, 1.22 \pm j2.42, -0.73 \pm j0.29\}$. Weight perturbation by slightly enhancing the local E-E connectivity while keeping I-E connection unchanged leads to a new eigenspectrum $\{-0.99, 1.39 \pm j2.43, -0.90 \pm j0.57\}$. (b) Two growing or decaying modes of harmonic oscillators emerge from the superposition of real and complex eigenmodes. The eigenmode in blue oscillates faster after synaptic plasticity. (c) Activation of two readout neurons (in red and blue) after ReLU transformation. After synaptic weight perturbation, the red-blue neuron sequence activation shifts earlier.

Biologically inspired Hebbian learning has been proposed for sequence learning (Fiete et al. 2010; Panda & Roy, 2017). For instance, a generalized Hebbian synaptic plasticity rule has been proposed with the following form (Koulakov, Hromadka, & Zador, 2009),

$$\dot{w}_{ij}^{\text{rec}} = \epsilon_1 r_i r_j w_{ij}^{\text{rec}} - \epsilon_2 w_{ij}^{\text{rec}}, \tag{4.1}$$

where the first term describes the generalized correlation between the presynaptic firing r_i and postsynaptic firing r_j, and the second term

describes a weight decay component; $0 < \epsilon_1 < 1$ denotes the learning rate, and ϵ_2 denotes the weight-decaying coefficient. In the discrete form, equation 4.1 is rewritten as

$$\begin{aligned} w_{ij}^{\text{rec}}(t+\Delta t) &= \epsilon_1 r_i(t) r_j(t) w_{ij}^{\text{rec}}(t)\Delta t + (1-\epsilon_2\Delta t) w_{ij}^{\text{rec}}(t) \\ &= \Big(\epsilon_1 r_i(t) r_j(t)\Delta t + 1 - \epsilon_2\Delta t\Big) w_{ij}^{\text{rec}}(t) \\ &= \chi(t) w_{ij}^{\text{rec}}(t), \end{aligned} \tag{4.2}$$

where Δt denotes the discrete time bin, and $\chi(t) = \epsilon_1 r_i(t) r_j(t)\Delta t + 1 - \epsilon_2\Delta t > 0$ is a positive time-varying scaling factor (assuming that ϵ_2 and Δt are sufficiently small). Note that this generalized Hebbian plasticity has a multiplicative form and preserves the excitatory or inhibitory sign of synaptic weights. As the multiplicative rule implies, the change in synaptic weights is proportional to the absolute value of the synaptic weight. This type of multiplicative rule can lead to a log-normal distribution of synaptic strengths (Loewenstein, Kuras, & Rumpel, 2011; Buzsaki & Mizuseki, 2014). A future research direction will focus on how heterosynatpic Hebbian/non-Hebbian plasticity influences the existing neural sequence representation or, equivalently, the eigenspectrum of recurrent weight matrix.

4.4 Limitation of Our Work. The excitatory-inhibitory RNN provides a framework to investigate the dynamics of biological neural networks. However, it still has several important limitations. First, our computer simulations could not fully capture the constraints in the biological system. For instance, the unconstrained $\mathbf{x}(t)$ showed strongly negative amplitudes in the time course, raising the issue of plausibility in terms of subthreshold synaptic activity. One possible solution is to bound the $x_i(t)$ amplitude by their net E and I contributions,

$$\tau \frac{dx_i}{dt} = -x_i + (I_i^{\text{exc}} - x_i) \sum_{j\in\text{exc}} w_{ij}^{\text{rec}} r_j + (I_i^{\text{inh}} - x_i) \sum_{j\in\text{inh}} w_{ij}^{\text{rec}} r_j + w_i^{\text{in}} u + \sigma\xi_i, \tag{4.3}$$

which, however, may make the learning unstable or slower. Second, the computational task used in our simulations was very simple, and the network size was relatively small. Generalization of our empirical observations to a more complex setting would require further investigations. Third, limited insight can be derived from a learned large weight matrix. However, it is possible to impose additional recurrent connectivity constraint onto the RNN while conducting constrained optimization; this additional constraint can potentially improve the expressivity and interpretability of RNN (Kerg et al., 2019). Fourth, it should be noted that computer simulations,

albeit useful, could not provide a rigorous mathematical proof regarding the stability or sufficient condition for limit cycle attractors. One type of mathematical analysis in the future work can study the Hopf bifurcation via AUTO (Roussel, 2019). Specifically, this would require linearization of the high-dimensional nonlinear dynamical system around the fixed points, followed by computation of complex conjugate eigenvalues. Finally, it remains unclear whether we can construct a line attractor excitatory-inhibitory RNN that produces a neural sequence by hand-tuning the network connectivity (result not shown: we have succeeded in constructing a simple RNN with a ring structure of 6 excitatory-inhibitory units to generate a periodic oscillator).

5 Conclusion

In summary, we trained an excitatory-inhibitory RNN to learn neural sequences and study its stimulus-driven and spontaneous network dynamics. Dale's principle imposes a constraint on the nonnormal recurrent weight matrix and provides a new perspective to analyze the impact of excitation and inhibition on RNN dynamics. Upon learning the output sequences, we found that dynamic attractors emerged in the trained RNN, and the limit cycle attractors in the steady state showed varying degrees of resilience to noise, interference, and recurrent connectivity. The eigenspectrum of the learned recurrent connectivity matrix with growing or damping modes Combined together with the RNN's nonlinearity, are adequate to generate a limit cycle attractor. By theoretical analyses of the approximate linear system equation, we also found conceptual links between the recurrent weight matrix and neural dynamics. Despite the task simplicity, our computer simulations provide a framework to study stimulus-driven and spontaneous dynamics of biological neural networks and further motivate future theoretical work.

Appendix A: Solving Equation for the Two-Dimensional Dynamical System

For simplicity of analysis, let us consider a rectified linear function and a noiseless condition,

$$\tau\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{W}^{\text{rec}}\phi(\mathbf{x}) + \mathbf{W}^{\text{in}}u, \tag{A.1}$$

where $\phi(\mathbf{x}) = [\mathbf{x}]_+$. It could be observed that equation A.1 is nonlinear and hence highly difficult and complicated to solve by analytical means for large dimensions of $\mathbf{x}(t)$. However, the ReLU activation results in nonlinearity. Depending on the sign of each of the components of $\mathbf{x} \in \mathbb{R}^N$, we obtain 2^N different linear equations.

For $\mathbf{x}(t) \in \mathbb{R}^2$, equation A.1 may be split into $2^2 = 4$ different conditions depending on the sign of individual component of $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$, as follows:

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} w_{11}^{\text{rec}} - 1 & w_{12}^{\text{rec}} \\ w_{21}^{\text{rec}} & w_{22}^{\text{rec}} - 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \end{bmatrix} \frac{u}{\tau}, \text{ where } x_1 > 0, x_2 > 0, \tag{A.2}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} w_{11}^{\text{rec}} - 1 & 0 \\ w_{21}^{\text{rec}} & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \end{bmatrix} \frac{u}{\tau}, \text{ where } x_1 > 0, x_2 \leq 0, \tag{A.3}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & w_{12}^{\text{rec}} \\ 0 & w_{22}^{\text{rec}} - 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \end{bmatrix} \frac{u}{\tau}, \text{ where } x_1 \leq 0, x_2 > 0, \tag{A.4}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \end{bmatrix} \frac{u}{\tau}, \text{ where } x_1 \leq 0, x_2 \leq 0. \tag{A.5}$$

Equations A.2 through A.5 may be simply rewritten as

$$\dot{\mathbf{x}} = \boldsymbol{A}\mathbf{x} + \boldsymbol{B}u, \tag{A.6}$$

where $\boldsymbol{B} = \frac{1}{\tau}\mathbf{W}^{\text{in}}$ and $\boldsymbol{A}$ is different depending on whether $x_{1,2} > 0$ or $x_{1,2} \leq 0$.

Without loss of generality, we assume that the input $u(t)$ has an exponential decaying form as follows:

$$u(t) = \begin{cases} 0, & \text{if } 0 \leq t < t_s \\ c_1 e^{-c_2(t-t_s)}, & \text{if } t \geq t_s, \text{ where } c_1 \text{ and } c_2 \text{ are two positive constants.} \end{cases} \tag{A.7}$$

A more generalized form of $u(t) = \sum c_1 \cos(\beta t + \beta_0) e^{-c_2(t-t_s)}$ is also valid for the following derivation.

To examine the dynamics of linear differential equation A.6, we apply eigenvalue decomposition to matrix $\boldsymbol{A}$,

$$\boldsymbol{AV} = \boldsymbol{VZ}, \tag{A.8}$$

where

$$\boldsymbol{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} v_{11} & v_{21} \\ v_{12} & v_{22} \end{bmatrix} \quad \text{and } \boldsymbol{Z} = \begin{bmatrix} \zeta_1 & 0 \\ 0 & \zeta_2 \end{bmatrix}.$$

Here, $\mathbf{v}_1$ and $\mathbf{v}_2$ denote two eigenvectors of $\boldsymbol{A}$, and ζ_1 and ζ_2 are their corresponding eigenvalues, respectively.

Furthermore, we assume that $\mathbf{v}_1$ and $\mathbf{v}_2$ are linearly independent of each other, and for arbitrary $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$, we have

$$\mathbf{x} = p_1\mathbf{v}_1 + p_2\mathbf{v}_2 = \boldsymbol{V}\mathbf{p}, \tag{A.9}$$

where $\mathbf{p} = \begin{bmatrix} p_1 & p_2 \end{bmatrix}^\top$.

In light of equations A.6, A.8, and A.9, we have

$$\boldsymbol{V}\dot{\mathbf{p}} = \boldsymbol{V}\boldsymbol{Z}\mathbf{p} + \boldsymbol{B}u. \tag{A.10}$$

Multiplying $\boldsymbol{V}^{-1}$ on both sides of the equation A.10 yields

$$\dot{\mathbf{p}} = \boldsymbol{Z}\mathbf{p} + \boldsymbol{V}^{-1}\boldsymbol{B}u. \tag{A.11}$$

Let $\mathbf{m} = \boldsymbol{V}^{-1}\boldsymbol{B} = \begin{bmatrix} m_1 & m_2 \end{bmatrix}^\top$; then equation A.11 is rewritten as

$$\dot{p_1} = \zeta_1 p_1 + m_1 u(t), \tag{A.12}$$

$$\dot{p_2} = \zeta_2 p_2 + m_2 u(t). \tag{A.13}$$

Note that $\zeta_1, \zeta_2, m_1, m_2$ are different depending on the algebraic signs of x_1 and x_2.

Solving linear equations A.12 and A.13 further yields

$$p_i(t) = \begin{cases} 0, \ 0 \le t < t_s \\ \frac{m_i}{c_1}\left[e^{\zeta_i(t-t_s)} - e^{-c_2(t-t_s)}\right], \ t \ge t_s \end{cases} \tag{A.14}$$

for $i = 1, 2$. From A.9, $\mathbf{x}(t)$ is rewritten as

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} p_1(t) + \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} p_2(t). \tag{A.15}$$

For each of four quadrants of $[x_1, x_2]$, the eigenvalues can be calculated as follows:

- When $x_1 > 0, x_2 > 0$, we have the determinant

$$\begin{vmatrix} \zeta - w_{11}^{\text{rec}} + 1 & -w_{12}^{\text{rec}} \\ -w_{21}^{\text{rec}} & \zeta - w_{22}^{\text{rec}} + 1 \end{vmatrix} = 0. \tag{A.16}$$

This further implies

$$\zeta^2 + (2 - w_{11}^{\text{rec}} - w_{22}^{\text{rec}})\zeta + (1 + w_{11}^{\text{rec}} w_{22}^{\text{rec}} - w_{12}^{\text{rec}} w_{21}^{\text{rec}} - w_{11}^{\text{rec}} - w_{22}^{\text{rec}}) = 0 \text{ and}$$
$$\zeta^2 + 2\zeta + (1 - w_{12}^{\text{rec}} w_{21}^{\text{rec}}) = 0 \tag{A.17}$$

because of $w_{11}^{\text{rec}} = w_{22}^{\text{rec}} = 0$. Solving equation A.17 yields two complex roots:

$$\zeta_{1,2}^{++} = -1 \pm \sqrt{w_{12}^{\text{rec}} w_{21}^{\text{rec}}}, \tag{A.18}$$

where w_{12}^{rec} has the opposite signs from w_{21}^{rec} so that $\sqrt{w_{12}^{\text{rec}} w_{21}^{\text{rec}}}$ is purely imaginary. Let $\omega = \sqrt{-w_{12}^{\text{rec}} w_{21}^{\text{rec}}}$ and $i = \sqrt{-1}$, so we have $\zeta_{1,2}^{++} = -1 \pm i\omega$.

- For all three other cases, that is, when $x_1 > 0, x_2 \leq 0$, or when $x_1 \leq 0, x_2 > 0$, or when $x_1 \leq 0, x_2 \leq 0$ in light of equation A.17, we have

$$\zeta^2 + 2\zeta + 1 = 0, \tag{A.19}$$

as either $w_{11}^{\text{rec}} = w_{21}^{\text{rec}} = 0$ and/or $w_{12}^{\text{rec}} = w_{22}^{\text{rec}} = 0$. We further derive the solutions as

$$\zeta_{1,2}^{+-} = \zeta_{1,2}^{-+} = \zeta_{1,2}^{--} = -1.$$

In summary, for a two-dimensional system equation, the solution is given as

$$\zeta_{1,2} = \begin{cases} -1 \pm \sqrt{w_{12}^{\text{rec}} w_{21}^{\text{rec}}}, & \text{if } x_1 > 0, x_2 > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{A.20}$$

Therefore, the solution can be either real or a pair of conjugate complex values.

Since $\{x_1(t), x_2(t)\}$ are potentially changing the quadrant location, they will influence the percentage of neuron activation (either 100%, 50%, or 0%) in a time-varying manner. Therefore, the superposition of the eigenmodes will change in both stimulus-driven and spontaneous network dynamics.

To help visualize the vector field in two-dimensional space, we used the softplus activation function (as a continuous approximation to ReLU) and randomized $\mathbf{W}^{\text{rec}}$ to examine the existence of fixed points or limit cycles. An example of phase portrait is shown in Figure 11a. Although a fixed point could be found, we did not find any closed orbit or periodic solution in four orthants.

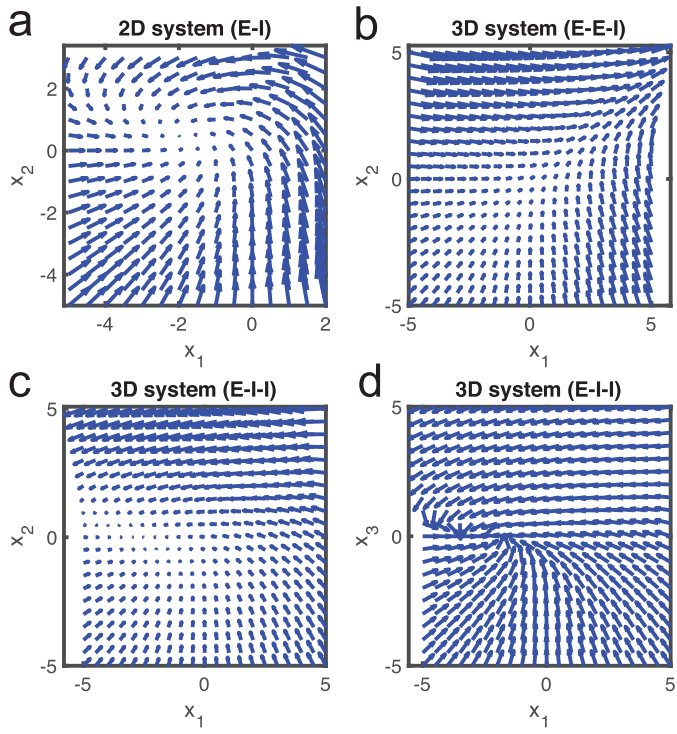

Figure 11: (a) Phase portrait of a two-dimensional (2D) dynamical system based on the softplus activation function, where $w_{12}^{\text{rec}} = -2$, $w_{21}^{\text{rec}} = 3$. A fixed point was found around $(-2, 0)$. (b–d) Phase portraits of a three-dimensional (3D) dynamical system based on the softplus activation function. Arrows represent the vector field.

Appendix B: Solving Equation for the Three-Dimensional Dynamical System

For $\mathbf{x} \in \mathbb{R}^3$, equation A.1 can be split for $2^3 = 8$ different cases depending on the algebraic sign of individual components. For notation simplicity, we omit the superscript in w_{ij}^{rec} and replace it with w_{ij} in the remaining derivation:

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} w_{11} - 1 & w_{12} & w_{13} \\ w_{21} & w_{22} - 1 & w_{23} \\ w_{31} & w_{32} & w_{33} - 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 > 0, x_2 > 0, x_3 > 0 \tag{B.1}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} w_{11}-1 & w_{12} & 0 \\ w_{21} & w_{22}-1 & 0 \\ w_{31} & w_{32} & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 > 0, x_2 > 0, x_3 \leq 0 \tag{B.2}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} w_{11}-1 & 0 & w_{13} \\ w_{21} & -1 & w_{23} \\ w_{31} & 0 & w_{33}-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 > 0, x_2 \leq 0, x_3 > 0 \tag{B.3}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & w_{12} & w_{13} \\ 0 & w_{22}-1 & w_{23} \\ 0 & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 \leq 0, x_2 > 0, x_3 > 0 \tag{B.4}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} w_{11}-1 & 0 & 0 \\ w_{21} & -1 & 0 \\ w_{31} & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 > 0, x_2 \leq 0, x_3 \leq 0 \tag{B.5}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & w_{12} & 0 \\ 0 & w_{22}-1 & 0 \\ 0 & w_{32} & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 \leq 0, x_2 > 0, x_3 \leq 0 \tag{B.6}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & 0 & w_{13} \\ 0 & -1 & w_{23} \\ 0 & 0 & w_{33}-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 \leq 0, x_2 \leq 0, x_3 > 0 \tag{B.7}$$

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_1^{\text{in}} \\ w_2^{\text{in}} \\ w_3^{\text{in}} \end{bmatrix} \frac{u}{\tau},$$

$$\text{where } x_1 \leq 0, x_2 \leq 0, x_3 \leq 0 \tag{B.8}$$

Equations B.1 through B.8 may be simply rewritten as

$$\dot{\mathbf{x}} = \boldsymbol{A}\mathbf{x} + \boldsymbol{B}u, \tag{B.9}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top$, $\boldsymbol{B} = \frac{1}{\tau}\mathbf{W}^{\text{in}}$ and the exact form of matrix $\boldsymbol{A}$ depends on whether $x_{1,2,3} > 0$ or $x_{1,2,3} \leq 0$.

Similar to the derivation shown in appendix A, applying eigenvalue decomposition to matrix $\boldsymbol{A}$ yields

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{Z}, \tag{B.10}$$

where

$$\boldsymbol{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{bmatrix} \quad \text{and } \boldsymbol{Z} = \begin{bmatrix} \zeta_1 & 0 & 0 \\ 0 & \zeta_2 & 0 \\ 0 & 0 & \zeta_3 \end{bmatrix},$$

where $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ are the eigenvectors of $\boldsymbol{A}$ and $\{\zeta_1, \zeta_2, \zeta_3\}$ are their corresponding eigenvalues, respectively.

Again, we express $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top$ as a linear combination of eigenvectors,

$$\mathbf{x} = p_1\mathbf{v}_1 + p_2\mathbf{v}_2 + p_3\mathbf{v}_3 = \boldsymbol{V}\mathbf{p}, \tag{B.11}$$

where $\mathbf{p} = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^\top$.

In light of equations B.9, B.10, and B.11, we have

$$\dot{\mathbf{p}} = \boldsymbol{Z}\mathbf{p} + \boldsymbol{V}^{-1}\boldsymbol{B}u, \tag{B.12}$$

Let $\mathbf{m} = \boldsymbol{V}^{-1}\boldsymbol{B} = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix}^\top$. We further obtain

$$\begin{cases} \dot{p_1} = \zeta_1 p_1 + m_1 u(t) \\ \dot{p_2} = \zeta_2 p_2 + m_2 u(t) \\ \dot{p_3} = \zeta_3 p_3 + m_3 u(t) \end{cases}, \tag{B.13}$$

a decoupled form of equations B.1 through B.8. Note that $\zeta_1, \zeta_2, \zeta_3$, m_1, m_2, m_3 are different depending on the algebraic sign of $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top$.

Solving the linear differential equation B.13 yields

$$p_i(t) = \begin{cases} 0, & 0 \leq t < t_s \\ \frac{m_i}{c_2}\left[e^{\zeta_i(t-t_s)} - e^{-q(t-t_s)}\right], & t \geq t_s, \end{cases} \tag{B.14}$$

and equation B.11 is rewritten as

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \end{bmatrix} p_1(t) + \begin{bmatrix} v_{21} \\ v_{22} \\ v_{23} \end{bmatrix} p_2(t) + \begin{bmatrix} v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} p_3(t). \tag{B.15}$$

Depending on the values of $\{x_1, x_2, x_3\}$, we can compute the eigenvalues as follows:

- When $x_1 > 0, x_2 > 0, x_3 > 0$, we have the following 3×3 determinant matrix:

$$\begin{vmatrix} \zeta - w_{11} + 1 & -w_{12} & -w_{13} \\ -w_{21} & \zeta - w_{22} + 1 & -w_{23} \\ -w_{31} & -w_{32} & \zeta - w_{33} + 1 \end{vmatrix} = 0. \tag{B.16}$$

  From $w_{11} = w_{22} = w_{33} = 0$, we further derive

$$\begin{aligned} &\zeta^3 + 3\zeta^2 + \zeta(3 - w_{12}w_{21} - w_{13}w_{31} - w_{23}w_{32}) \\ &\quad + (1 - w_{12}w_{23}w_{31} - w_{13}w_{21}w_{32} - w_{23}w_{32} - w_{12}w_{21} - w_{13}w_{31}) = 0. \end{aligned} \tag{B.17}$$

  The analytic solution of cubic equation B.17 is available, but this equation does not have $\zeta = -1$ as one of its roots, unlike the remaining cases discussed. The eigenvalues for other cases shown in equations B.2 to B.9 are discussed in the sequel.
- When $x_1 > 0, x_2 > 0, x_3 \le 0$, from equation B.17, we have

$$\zeta^3 + 3\zeta^2 + \zeta(3 - w_{12}w_{21}) + (1 - w_{12}w_{21}) = 0 \tag{B.18}$$

  because of $w_{13} = w_{23} = 0$, which is assumed in this case. It is noted that $\zeta = -1$ is one of the roots satisfying equation B.18. Rewriting equation B.18 in a factorized form, $(\zeta + 1)(\zeta^2 + 2\zeta + (1 - w_{12}w_{21})) = 0$, we obtain the solutions

$$\zeta = -1 \text{ or } -1 \pm \sqrt{w_{12}w_{21}}. \tag{B.19}$$

- When $x_1 > 0, x_2 \le 0, x_3 > 0$, solving equation B.17 yields

$$\zeta = -1 \text{ or } -1 \pm \sqrt{w_{13}w_{31}}. \tag{B.20}$$

- When $x_1 \le 0, x_2 > 0, x_3 > 0$, solving equation B.17 yields

$$\zeta = -1 \text{ or } -1 \pm \sqrt{w_{23}w_{32}}. \tag{B.21}$$

- For all other remaining cases, namely, when $x_1 > 0, x_2 \le 0, x_3 \le 0$, or when $x_1 \le 0, x_2 > 0, x_3 \le 0$, or when $x_1 \le 0, x_2 \le 0, x_3 > 0$, or when $x_1 \le 0, x_2 \le 0, x_3 \le 0$, we have the solutions as

$$\zeta = -1. \tag{B.22}$$

Therefore, the solutions can contain one real and a pair of conjugate complex values.

Examples of phase portraits of three-dimensional dynamical system for two selected hyperoctants are shown in Figures 11b to 11d. We have examined and simulated both E-E-I (two excitatory and one inhibitory neurons) and E-I-I (one excitatory and two inhibitory neurons) setups.

Appendix C: jPCA via Polar Decomposition

The jPCA method has been used to reveal rotational dynamics of neuronal population responses (Churchland et al., 2012). Let's assume that the data are modeled as a linear time-invariant continuous dynamical system of the form

$$\dot{\mathbf{x}} = \mathbf{M}\mathbf{x}, \tag{C.1}$$

where the linear transformation matrix $\mathbf{M}$ is constrained to be skew-symmetric (i.e., $\mathbf{M}^\top = -\mathbf{M}$). The jPCA algorithm projects high-dimensional data $\mathbf{x}(t)$ onto the eigenvectors of the $\mathbf{M}$ matrix, and these eigenvectors arise in complex conjugate pairs. Given a pair of eigenvectors $\{v_k, \bar{v}_k\}$, the kth jPCA projection plane axes are defined as $u_{k,1} = v_k + \bar{v}_k$ and $u_{k,2} = j(v_k - \bar{v}_k)$ (where $j = \sqrt{-1}$).

The solution to continuous-time differential equation C.1 is given by $\mathbf{x}(t) = e^{\mathbf{M}t}\mathbf{x}(0)$, where the family $\{e^{\mathbf{M}t}\}$ is often referred to as the semi-group generated by the linear operator $\mathbf{M}$. Since $\mathbf{M}$ is skew-symmetric, $e^{\mathbf{M}}$ is orthogonal; therefore, it can describe the rotation of the initial condition $\mathbf{x}(0)$ over time. We apply eigenvalue decomposition to the real skew-symmetric matrix $\mathbf{M}$, so that $\mathbf{M} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$, where $\mathbf{\Lambda}$ is a diagonal matrix whose entries $\{\lambda_i\}_{i=1}^N$ are a set of (zero or purely imaginary) eigenvalues. Therefore, we have $e^{\mathbf{M}t} = \mathbf{U}e^{\mathbf{\Lambda}t}\mathbf{U}^{-1}$. By taking the powers of the diagonal matrix $\mathbf{\Lambda}$, we have

$$\begin{aligned} e^{\mathbf{M}t} &= \mathbf{U}\sum_{k=0}^{\infty}\frac{(\mathbf{\Lambda}t)^k}{k!}\mathbf{U}^{-1} \\ &= \mathbf{U}\begin{pmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_N t} \end{pmatrix}\mathbf{U}^{-1} \end{aligned} \tag{C.2}$$

in which each eigenmode defines the natural frequency of oscillation.

In addition, we have

$$\mathbf{x}(t+\tau) = e^{\mathbf{M}(t+\tau)}\mathbf{x}(0) = e^{\mathbf{M}\tau}\mathbf{x}(t), \tag{C.3}$$

which implies that for any time pairs $\{t, t+\tau\}$, their activations are separated by a constant phase shift $e^{\mathbf{M}\tau}$. In a two-dimensional space, assuming that the orthogonal matrix $e^{\mathbf{M}\tau}$ has the form of $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, this orthogonal matrix corresponds to a rotation around the origin by angle θ.

Upon time discretization (assuming $dt = 1$), we have the discrete analog of equation C.1,

$$\mathbf{x}(t+1) = (\mathbf{I} + \mathbf{M})\mathbf{x}(t). \tag{C.4}$$

Note that $(\mathbf{I} + \mathbf{M})$ is a first-order Taylor's series approximation to the orthogonal transformation $\boldsymbol{Q} = e^{\mathbf{M}}$.

Alternatively, we can directly solve a discrete dynamical system of the vector autoregressive (VAR) process form $\mathbf{x}(t+1) = \boldsymbol{Q}\mathbf{x}(t)$ over the space of orthogonal $\boldsymbol{Q}$ matrices. Mathematically, we have previously shown that this is equivalent to solving the following constrained optimization problem (Nemati, Linderman, & Chen, 2014):

$$\boldsymbol{Q}^* = \arg\min_{\boldsymbol{Q}} \|\|\boldsymbol{A} - \boldsymbol{Q}\|\|_F^2, \text{ subject to } \boldsymbol{Q}\boldsymbol{Q}^\top = \mathbf{I}, \tag{C.5}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm and $\boldsymbol{A} = (\mathbf{X}_{t+1}\mathbf{X}_t^\top)(\mathbf{X}_t\mathbf{X}_t^\top)^{-1}$ represents the least square solution to the unconstrained problem $\mathbf{x}(t+1) = \boldsymbol{A}\mathbf{x}(t)$. The solution to the above constrained optimization is given by the orthogonal matrix factor of celebrated polar decomposition of matrix $\boldsymbol{A}$, namely, $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{P}$ (Higham, 1986).

From polar decomposition, if λ_k and v_k correspond to the eigenvalue and eigenvector of matrix $\mathbf{M}$, respectively, then $1 + \lambda_k$ and v_k will correspond to the approximate eigenvalue and eigenvector of matrix $\boldsymbol{Q}$. That is, the jPCA eigenvectors span the same space as the eigenvectors of the polar decomposition factor $\boldsymbol{Q}$.

Appendix D: Low-Dimensional Approximation of Limit Cycle Attractor

Given an arbitrary real-valued square matrix $\mathbf{W}_{\text{rec}}$, we can decompose it into the sum of a symmetric matrix $\tilde{\mathbf{W}}_1 = \tilde{\mathbf{W}}_1^\top$ and a skew-symmetric matrix $\tilde{\mathbf{W}}_2 = -\tilde{\mathbf{W}}_2^\top$:

$$\begin{aligned}\mathbf{W}_{\text{rec}} &= \frac{\mathbf{W}_{\text{rec}} + \mathbf{W}_{\text{rec}}^{\top}}{2} + \frac{\mathbf{W}_{\text{rec}} - \mathbf{W}_{\text{rec}}^{\top}}{2} \\ &\equiv \tilde{\mathbf{W}}_1 + \tilde{\mathbf{W}}_2. \end{aligned} \tag{D.1}$$

Applying eigenvalue decomposition to the $N \times N$ matrix $\tilde{\mathbf{W}}_1$, we use the first two dominant eigenvectors $\mathbf{u}$ and $\mathbf{v}$ to approximate $\tilde{\mathbf{W}}_1$:

$$\tilde{\mathbf{W}}_1 \approx \lambda_1 \mathbf{u}\mathbf{u}^{\top} + \lambda_2 \mathbf{v}\mathbf{v}^{\top}, \tag{D.2}$$

where $\lambda_1 > \lambda_2 > 0$ denotes the eigenvalues, and $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$, $\mathbf{u}^{\top}\mathbf{v} = \mathbf{v}^{\top}\mathbf{u} = 0$. The vectors $\mathbf{u}$ and $\mathbf{v}$ represent the dominant principal component (PC) subspaces.

We further approximate $\tilde{\mathbf{W}}_2$ with the following form,

$$\tilde{\mathbf{W}}_2 \approx \rho(\mathbf{u}\mathbf{v}^{\top} - \mathbf{v}\mathbf{u}^{\top}), \tag{D.3}$$

where ρ is a constant such that

$$\rho = \arg\min_{\lambda} \|\tilde{\mathbf{W}}_2 - \lambda(\mathbf{u}\mathbf{v}^{\top} - \mathbf{v}\mathbf{u}^{\top})\|_F^2. \tag{D.4}$$

In light of equations D.2 and D.3, we define $p_{\mathbf{u}} = \mathbf{u}^{\top}\mathbf{x}/\sqrt{N}$ and $p_{\mathbf{v}} = \mathbf{v}^{\top}\mathbf{x}/\sqrt{N}$ as two linear projections of N-dimensional $\mathbf{x}$ onto the vector spaces $\mathbf{u}$ and $\mathbf{v}$, respectively. In the steady state (i.e., $u(t) = 0$), equation A.1 is rewritten as

$$\tau\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{W}^{\text{rec}}\phi(\mathbf{x}). \tag{D.5}$$

Multiplying $\mathbf{u}^{\top}$ (or $\mathbf{v}^{\top}$) to the both sides of equation D.5 yields

$$\tau\dot{p}_{\mathbf{u}} \approx -p_{\mathbf{u}} + (\lambda_1\mathbf{u}^{\top} + \rho\mathbf{v}^{\top})\phi/\sqrt{N}, \tag{D.6}$$

$$\tau\dot{p}_{\mathbf{v}} \approx -p_{\mathbf{v}} + (\lambda_2\mathbf{v}^{\top} - \rho\mathbf{u}^{\top})\phi/\sqrt{N}. \tag{D.7}$$

Therefore, we can examine the phase portrait of the approximate dynamical system in the two-dimensional ($p_{\mathbf{u}}$-$p_{\mathbf{v}}$) space.

According to Susman et al. (2019), linear stability theory predicts that the solutions of equations D.6 and D.7 converge onto the **u**-**v** plane, so we can express $\mathbf{x} = p_{\mathbf{u}}\mathbf{u} + p_{\mathbf{v}}\mathbf{v}$. Let $q_{\mathbf{u}} = \mathbf{u}^{\top}\phi/\sqrt{N}$ and $q_{\mathbf{v}} = \mathbf{v}^{\top}\phi/\sqrt{N}$; equations D.6 and D.7 can be rewritten as

$$\tau\dot{p}_{\mathbf{u}} \approx -p_{\mathbf{u}} + \lambda_1 q_{\mathbf{u}} + \rho q_{\mathbf{v}}, \tag{D.8}$$

$$\tau\dot{p}_{\mathbf{v}} \approx -p_{\mathbf{v}} + \lambda_2 q_{\mathbf{v}} - \rho q_{\mathbf{u}}. \tag{D.9}$$

Based on an approximation of activation function ϕ, the approximate two-dimensional dynamical system equations D.8 and D.9 can be numerically

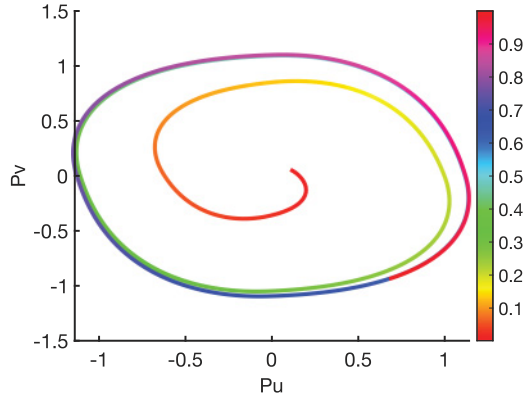

Figure 12: Numerical simulation of a stable limit cycle based on an approximate two-dimensional dynamical system in the $(p_{\mathbf{u}}, p_{\mathbf{v}})$ plane. Here, we used $\lambda_1 = 1.96$, $\lambda_2 = 1.51$, $\rho = 3$. A stable limit cycle attractor was observed for a wide range of ρ. Color represents the time direction.

simulated given arbitrary initial conditions (Susman et al., 2019). An example of stable limit cycle simulated on the $(p_{\mathbf{u}}, p_{\mathbf{v}})$ plane is shown in Figure 12.

In Figures 4c and 4d, we demonstrated that once high-dimensional $\mathbf{x}$ forms a limit cycle attractor, its lower-dimensional PCA projection also forms a limit cycle.

Acknowledgments

We thank Vishwa Goudar for valuable feedback. This work was partially supported by NIH grants R01-MH118928 (Z.S.C.) and R01-NS100065 (Z.S.C.) and an NSF grant CBET-1835000 (Z.S.C).

References

Adler, A., Zhao, R., Shin, M. E., Yasuda, R., & Gan, W. B. (2019). Somatostatin-expressing interneurons enable and maintain learning-dependent sequential activation of pyramidal neurons. *Neuron*, *102*, 202–216.

Bay, A., Lepsoy, S., & Magli, E. (2016). Stable limit cycles in recurrent neural networks. In *Proceedings of IEEE International Conference on Communications*. Piscataway, NJ: IEEE. doi:10.1109/ICComm.2016.7528305

Barak, O. (2017). Recurrent neural networks as versatile tools of neuroscience research. *Curr. Opin. Neurobiol.*, *46*, 1–6.

Bi, Z., & Zhou, C. (2020). Understanding the computation of time using neural network models. In *Proc. Natl. Acad. Sci. USA*, *117*, 10530–10540.

Brunel, N. (2000). Dynamics of networks of randomly connected excitatory and inhibitory spiking neurons. *J. Physiol. Paris*, *94*, 445–463.

Buzsaki, G., & Mizuseki, K. (2014). The log-dynamic brain: How skewed distributions affect network operations. *Nat. Rev. Neurosci.*, *15*, 264–278.

Buzsaki, G., & Tingley, D. (2018). Space and time: The hippocampus as a sequence generator. *Trends in Cognitive Sciences*, *22*, 853–869.

Cannon, J., Kopell, N., Gardner, T., & Markowitz, J. (2015) Neural sequence generation using spatiotemporal patterns of inhibition. *PLOS Comput. Biol.*, *11*, e1004581.

Ceni, A., Ashwin, P., & Livi, L. (2019). *Interpreting recurrent neural networks behavior via excitable network attractors.* arXiv:1807.10478v6.

Ceni, A., Ashwin, P., Livi, L., & Oostlethwaite, C. (2020). The echo index and multistability in input-driven recurrent neural networks. *Physica D: Nonlinear Phenomena*, *412*, 132609.

Cheng, C-Y., Lin, K-H., & Shih, C-W. (2006). Multistability in recurrent neural networks, *SIAM Journal on Applied Mathematics*, *66*, 1301–1320.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., & Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, *487*, 51–56.

Deshpande, V., & Dasgupta, C. (1991). A neural network for storing individual patterns in limit cycles. *J. Phys. A Math. Gen.*, *24*, 5105–5119.

Diba, K., & Buzsaki, G. (2007). Forward and reverse hippocampal place-cell sequences during ripples. *Nat. Neurosci.*, *10*, 1241–1242.

Duncker, L., Driscoll, L. N., Shenoy, K. V., Sahani, M., & Sussillo, D. (2020). Organizing recurrent network dynamics by task-computation to enable continual learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems*, *33*. Red Hook, NY: Curran.

Fiete, I. R., Senn, W, Wang, C. Z. H., & Hahnloser, R. H. R. (2010). Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity. *Neuron*, *65*, 563–576.

Folli, V., Gosti, G., Leonetti, M., & Ruocco, G. (2018). Effect of dilution in asymmetric recurrent neural networks. *Neural Networks*, *104*, 50–59.

Foster, D. J., & Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, *440*, 680–683.

Fujisawa, S., Amarasingham, A., Harrison, M. T., & Buzsaki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nature Neuroscience*, *11*, 823–833.

Ganguli, S., Hug, D., & Sompolinsky, H. (2008). Memory traces in dynamical systems. In *Proc. Natl. Acad. Sci. USA*, *105*, 18970–18975.

Gillett, M., Pereira, U., & Brunel, N. (2020). Characteristics of sequential activity in networks with temporally asymmetric Hebbian learning. In *Proc. Natl. Acad. Sci. USA*, *117*, 29948–29958.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, *15*, 315–323.

Goldman, M. S. (2009). Memory without feedback in a neural network. *Neuron*, *61*, 621–634.

Goudar, V., & Buonomano, D. V. (2018). Encoding sensory and motor patterns as time-invariant trajectories in recurrent neural networks. *eLife*, *7*, e31134.

Hardy, N. F., & Buonomano, D. V. (2018). Encoding time in feedforward trajectories of a recurrent neural network model. *Neural Computation*, *30*, 378–396.

Hardy, N. F., Goudar, V., Romero-Sosa, J. L., & Buonomano, D. V. (2018). A model of temporal scaling correctly predicts that motor timing improves with speed. *Nature Communications*, *9*, 4732.

Harvey, C. D., Coen, P., & Tank, D. W. (2012). Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, *484*, 62–68.

Hemberger, M., Shein-Idelson, M., Pammer, L., & Laurent, G. (2019). Reliable sequential activation of neural assemblies by single pyramidal cells in a three-layered cortex. *Neuron*, *104*, 353–369.

Higham, N. J. (1986). Computing the polar decomposition with applications. *SIAM J. Sci. Stat. Comput.*, *7*, 1160–1174.

Ingrosso, A., & Abbott, L. F. (2019). Training dynamically balanced excitatory-inhibitory networks. *PLOS One*, *14*, e0220547.

Jaeger, H. (2001). *The "echo state" approach to analyzing and training recurrent neural networks* (GMD Technical Report 148). St. Augustin: German National Research Center for Information Technology.

Jordan, I. D., Sokol, P. A., & Park, I. M. (2019). *Gated recurrent units viewed through the lens of continuous time dynamical systems*. arXiv:1906.01005.

Jouffroy, G. (2007). Design of simple limit cycles with recurrent neural networks for oscillatory control. In *Proceedings of Sixth International Conference on Machine Learning and Applications*. doi:10.1109/ICMLA.2007.99

Kao, J. C. (2019). Considerations in using recurrent neural networks to probe neural dynamics. *J. Neurophysiol*, *122*, 2504–2521.

Kerg, G., Goyette, K., Touzel, M. P., Gidel, G., Vorontsov, E., Bengo, Y., & Lajoie, G. (2019). Non-normal recurrent neural network (nnRNN): Learning long time dependencies while improving expressivity with transient dynamics. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*, *32*. Red Hook, NY: Curran.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., . . . Hadsell, R. (2017) Overcoming catastrophic forgetting in neural networks. In *Proc. Natl. Acad. Sci. USA*, *114*, 3521–3526.

Koulakov, A., Hromadka, T., & Zador, A. M. (2009). Correlated connectivity and the distribution of firing rates in the neocortex. *Journal of Neuroscience*, *29*, 3685–3694.

Lebedev, M. A., Ossadtchi, A., Mill, N. A., Urpi, N. A., Cervera, M. R., & Nicolelis, A. L. (2019). Analysis of neuronal ensemble activity reveals the pitfalls and shortcomings of rotation dynamics. *Sci. Rep.*, *9*, 18978.

Lillicrap, T. P., Stantoro, A., Marris, L., Akerman, C. J., & Hinton, G. E. (2020), Backpropagation and the brain. *Nat. Rev. Neurosci.*, *21*, 335–346.

Loewenstein, Y., Kuras, A., & Rumpel, S. (2011). Multiplicative dynamics underlie the emergence of the log-normal distribution of spine sizes in the neocortex in vivo. *Journal of Neuroscience*, *31*, 9481–9488.

Long, M. A., Jin, D. Z., & Fee, M. S. (2010). Support for a synaptic chain model of neuronal sequence generation. *Nature*, *468*, 394–399.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*, 2531–2560.

Mackwood, O. H., Naumann, L. B., & Sprekeler, H. (2021). *Learning excitatory-inhibitory neuronal assemblies in recurrent networks*. *eLife*, *10*, e59715. https://doi.org/10.7554/eLife.59715

Mante, V., Sussillo, D., Shenoy, K., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, *503*, 78–84.

Mastrogiuseppe, F., & Ostojic, S. (2017). Intrinsically-generated fluctuating activity in excitatory-inhibitory networks. *PLOS Comput. Biol.*, *13*(4), e1005498.

Murphy, B. K., & Miller, K. D. (2009). Balanced amplification: A new mechanism of selective amplification of neural activity patterns. *Neuron*, *61*, 635–648.

Namikawa, J., & Tani, J. (2009). Building recurrent neural networks to implement multiple attractor dynamics using the gradient descent method. *Advances in Artificial Neural Systems*, *2009*, 846040.

Nemati, S., Linderman, S. W., & Chen, Z. (2014). *A probabilistic modeling approach for uncovering neural population rotational dynamics*. COSYNE abstract.

Orhan, A. E., & Ma, W. J. (2019). A diverse range of factors affect the nature of neural representations underlying short-term memory. *Nat. Neurosci.*, *22*, 275–283.

Orhan, A. E., & Pitkow, X. (2020). Improved memory in recurrent neural networks with sequential non-normal dynamics. In *Proc. Int. Conf. Learning Representations*. ICLR.

Panda, P., & Roy. K. (2017). Learning to generate sequences with combination of Hebbian and non-Hebbian plasticity in recurrent spiking neural networks. *Frontiers in Neursoscience*, December 12.

Pollock, E., & Jazayeri, M. (2020). Engineering recurrent neural networks from task-relevant manifolds and dynamics. *PLOS Computational Biology*, *16*, e1008128.

Rajan, K., & Abbott, L. F. (2006). Eigenvalue spectra of random matrices for neural networks. *Physical Review Letters*, *97*, 188104.

Rajan, K., Harvey, C. D., & Tank, D. W. (2016). Recurrent network models of sequence generation and memory. *Neuron*, *90*, 128–142.

Roussel, M. R. (2019). Bifurcation analysis with AUTO. *Nonlinear dynamics*. Morgan & Claypool.

Schmitt, L., Wimmer, R. D., Nakajima, M., Happ, M., Mofakham, S., & Halassa, M. M. (2017). Thalamic amplification of cortical connectivity sustains attentional control. *Nature*, *545*, 219–223.

Seung, H. S. (1996). How the brain keeps the eyes still. In *Proc. Natl. Acad. Sci. USA*, *93*, 13339–13344.

Song, H. F., Yang, G. R., & Wang, X. J. (2016). Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework. *PLOS Comput. Biol.*, *12*, e1004792.

Susman, L., Brenner, N., & Barak, O. (2019). Stable memory with unstable synapses. *Nature Communications*, *10*, 4441.

Sussillo, D. (2014). Neural circuits as computational dynamical systems. *Curr. Opin. Neurobiol.*, *25*, 156–163.

Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*, 544–557.

Sussillo, D., & Barak. O. (2013). Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, *25*, 626–649.

Sussillo, D., Churchland, M. M., Kaufman, M. T., & Shenoy, K. V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, *18*, 1025–1033.

Trischler, A. P., & D'Eleuterio, G. M. T. (2016). Synthesis of recurrent neural networks for dynamical system stimulation. *Neural Networks*, *80*, 67–78.

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, *1*, 339–356.

Xue, X., Halassa, M. M., & Chen, Z. (2021). *Spiking recurrent neural networks represent task-relevant neural sequences in rule-dependent computation.* bioRxiv preprint. https://www.biorxiv.org/conten

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., & Wang, X. J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, *22*, 297–306.

Zeng, Z., Huang, T., & Zheng, W. X. (2010). Multistability of recurrent neural networks with time-varying delays and the piecewise linear activation function. *IEEE Trans. Neural Networks*, *21*, 1371–1377.

Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of International Conference on Machine Learning* (pp. 3978–3995). PMLR.

Zhang, X., Liu, S., & Chen, Z. (2021). *A geometric framework for understanding dynamic information integration in context-dependent computation.* bioRxiv preprint. https://biorxiv.org/cgi/content/short/2

Received December 16, 2020; accepted April 8, 2021.