ORIGINAL RESEARCH
published: 07 September 2020
doi: 10.3389/fncom.2020.00078

Learning Long Temporal Sequences in Spiking Networks by Multiplexing Neural Oscillations

Philippe Vincent-Lamarre, Matias Calderini and Jean-Philippe Thivierge*

School of Psychology and Center for Neural Dynamics, University of Ottawa, Ottawa, ON, Canada

Many cognitive and behavioral tasks—such as interval timing, spatial navigation, motor control, and speech—require the execution of precisely-timed sequences of neural activation that cannot be fully explained by a succession of external stimuli. We show how repeatable and reliable patterns of spatiotemporal activity can be generated in chaotic and noisy spiking recurrent neural networks. We propose a general solution for networks to autonomously produce rich patterns of activity by providing a multi-periodic oscillatory signal as input. We show that the model accurately learns a variety of tasks, including speech generation, motor control, and spatial navigation. Further, the model performs temporal rescaling of natural spoken words and exhibits sequential neural activity commonly found in experimental data involving temporal processing. In the context of spatial navigation, the model learns and replays compressed sequences of place cells and captures features of neural activity such as the emergence of ripples and theta phase precession. Together, our findings suggest that combining oscillatory neuronal inputs with different frequencies provides a key mechanism to generate precisely timed sequences of activity in recurrent circuits of the brain.

Keywords: neural oscillations, spiking neural networks, recurrent neural networks, temporal processing, balanced networks

OPEN ACCESS

Edited by:
Pulin Gong,
The University of Sydney, Australia

Reviewed by:
Louis Kang,
University of California, Berkeley,
United States
Upinder S. Bhalla,
National Centre for Biological
Sciences, India
Dilawar Singh,
National Centre for Biological
Sciences, Bangalore, India in
collaboration with reviewer UB
Bhanu Priya Somasekharan,
National Centre for Biological
Sciences, Bangalore, India in
collaboration with reviewer UB

****Correspondence:***
Philippe Vincent-Lamarre
pvinc058@uottawa.ca

Received: *13 February 2020*
Accepted: *24 July 2020*
Published: *07 September 2020*

Citation:
Vincent-Lamarre P, Calderini M and Thivierge J-P (2020) Learning Long Temporal Sequences in Spiking Networks by Multiplexing Neural Oscillations.
Front. Comput. Neurosci. 14:78.
doi: 10.3389/fncom.2020.00078

1. INTRODUCTION

Virtually every aspect of sensory, cognitive, and motor processing in biological organisms involves operations unfolding in time (Buonomano and Maass, 2009). In the brain, neuronal circuits must represent time on a variety of scales, from milliseconds to minutes and longer circadian rhythms (Buhusi and Meck, 2005). Despite increasingly sophisticated models of brain activity, time representation remains a challenging problem in computational modeling (Grondin, 2010; Paton and Buonomano, 2018).

Recurrent neural networks offer a promising avenue to detect and produce precisely timed sequences of activity (Abbott et al., 2016). However, it is challenging to train these networks due to their complexity (Pascanu et al., 2013), particularly when operating in a chaotic regime associated with biological neural networks (van Vreeswijk and Sompolinsky, 1996; Abarbanel et al., 2008).

One avenue to address this issue has been to use reservoir computing (RC) models (Jaeger, 2002; Maass et al., 2002). Under this framework, a recurrent network (the reservoir) projects onto a read-out layer whose synaptic weights are adjusted to produce a desired response. However, while RC can capture some behavioral and cognitive processes (Sussillo and Abbott, 2009; Laje and Buonomano, 2013; Nicola and Clopath, 2017), it often relies on biologically implausible mechanisms, like strong feedback form the readout to the reservoir or implausible learning rules required to control the reservoir's dynamics. Further, current RC implementations offer little insight to understand how the brain generates activity that does not follow a strict rhythmic pattern (Buonomano and Maass, 2009; Abbott et al., 2016). That is because RC models are either restricted to learning periodic functions, or require an aperiodic input to generate an aperiodic output, thus leaving the neural origins of aperiodic activity unresolved (Abbott et al., 2016). A solution to this problem is to train the recurrent connections of the reservoir to stabilize innate patterns of activity (Laje and Buonomano, 2013), but this approach is more computationally expensive and is sensitive to structural perturbations (Vincent-Lamarre et al., 2016).

To address these limitations, we propose a biologically plausible spiking recurrent neural network (SRNN) model that receives multiple independent sources of neural oscillations as input. The architecture we propose is similar to previous RC implementations (Nicola and Clopath, 2017), but we use a balanced SRNN following Dale's law, which are typically used to model the activity of cortical networks (van Vreeswijk and Sompolinsky, 1998; Brunel, 2000). The combination of oscillators with different periods creates a multi-periodic code that serves as a time-varying input that can largely exceed the period of any of its individual components. We show that this input can be generated endogenously by distinct sub-networks, alleviating the need to train recurrent connections of the SRNN to generate long segments of aperiodic activity. Thus, multiplexing a set of oscillators into a SRNN provides an efficient and neurophysiologically grounded means of controlling a recurrent circuit (Vincent-Lamarre et al., 2016). Analogous mechanisms have been hypothesized in other contexts including grid cell representations (Fiete et al., 2008) and interval timing (Miall, 1989; Matell and Meck, 2004).

This paper is structured as follows. First, we describe a simplified scenario where a SRNN that receives a collection of input oscillations learns to reproduce an arbitrary time-evolving signal. Second, we extend the model to show how oscillations can be generated intrinsically by oscillatory networks that can be either embedded or external to the main SRNN. Third, we show that a network can learn several tasks in parallel by "tagging" each task to a particular phase configuration of the oscillatory inputs. Fourth, we show that the activity of the SRNN captures temporal rescaling and selectivity, two features of neural activity reported during behavioral tasks. Fifth, we train the model to reproduce natural speech at different speeds when cued by input oscillations. Finally, we employ a variant of the model to capture hippocampal activity during spatial navigation. Together, results highlight a novel role for neural oscillations in regulating temporal processing within recurrent networks of the brain.

2. METHODS

2.1. Integrate-and-Fire Networks

2.1.1. Driven Networks

Our network consists of leaky integrate-and-fire neurons, where $N_{rnn} = 1,000$ for the SRNN projecting to the read-out units and $N_{osc} = 500$ for each oscillatory network, by default. Eighty percent of these neurons are selected to be excitatory while the remaining 20% are inhibitory. The membrane potential of all neurons is given by

$$C\frac{dV}{dt} = \frac{1}{R}(E_L - V) + g_{ex}(E_{ex} - V) + g_{in}(E_{in} - V) + I_{tonic} + I_{ext} \quad (1)$$

where C and R are the membrane capacitance and resistance, E_L is the leak reversal potential, g_{ex} and g_{in} are the time-dependent excitatory and inhibitory conductances, E_{ex} and E_{in} are the excitatory and inhibitory reversal potentials, I_{tonic} is a constant current applied to all neurons and I_{ext} is a time-varying input described below. Parameters were sampled from Gaussian distributions as described in **Table 1**. The excitatory and inhibitory conductances, g_{ex} and g_{in}, respectively, obey the following equations:

$$\tau_{ex_i}\frac{dg_{ex_i}}{dt} = -g_{ex_i} + \sum_{j=1}^{N_{rnn}} W_{i,j} G_{ex_i} \delta(t - t^{(j)} - T^{(j)}_{delay}) \quad (2)$$

$$\tau_{in_i}\frac{dg_{in_i}}{dt} = -g_{in_i} + \sum_{j=1}^{N_{rnn}} W_{i,j} G_{in_i} \delta(t - t^{(j)} - T^{(j)}_{delay}) \quad (3)$$

where τ_{ex} and τ_{in} are the time constants of the excitatory and inhibitory conductances, and G_{in} and G_{ex} are the change in conductance from incoming spikes to excitatory and inhibitory synapses. V_θ is the spiking threshold, $t^{(j)}$ denotes the time since the last spike of the pre-synaptic neuron j, after which V is set to V_{reset} for a duration equal to τ_{ref}. T_{delay} is the propagation delay of the action potential. The SRNN connectivity is defined as a $N_{rnn} \times N_{rnn}$ sparse and static connectivity matrix W with a density p_{rnn} (probability of having a non-zero pairwise connection). The non-zero connections are drawn from a half-normal distribution $f(0, \sigma_{rnn})$, where $\sigma_{rnn} = \frac{\gamma_{rnn}}{\sqrt{N_{rnn} \times p_{rnn}}}$. All ODEs are solved using a forward Euler method with time-step $\Delta t = 0.05$ ms. This results in an asynchronous regular network (**Figure S1**).

Each SRNN neuron is connected to each of the N_{inp} input units with probability p_{inp}. The external inputs (I_{ext}) follow:

$$I_{ext}(t) = M(\frac{A}{2}(\sin(2\pi \mathbf{f} t + \phi) + 1)) \quad (4)$$

where $I_{ext} \in \mathbb{R}^{N_{rnn}}$and $f, \phi \in \mathbb{R}^{N_{inp}}$. The frequency of the sine waves and their initial phase are represented by f (hz) and ϕ are drawn from the uniform distributions $U(-\pi, \pi)$ and $U(f_{min}, f_{max})$, respectively (fixed for each realization of a task). The sine wave is then transformed by adding 1 and dividing by 2 to limit its range to [0,A]. The full input-to-SRNN connectivity matrix M is a $N_{rnn} \times N_{inp}$ sparse and static matrix, with a density of p_{inp}. The non-zero connections of M are drawn from a normal distribution $\mathcal{N}(0, 1)$ and A is the amplitude of the input (30 pA by default).

All of the SRNN's excitatory neurons project to the readout units. Their spiking activity r is filtered by a double exponential:

$$\tau_d \frac{dr_j}{dt} = -r_j + h_j \quad (5)$$

TABLE 1 | SRNN parameters.

	Main SRNN	Input (oscillatory networks)
N	1,000	500
R	100 $M\Omega$	100 $M\Omega$
C	200 pF	200 pF
E_L	$\mu = -60$ mV, $\sigma = 1.2$ mV	$\mu = -60$ mV, $\sigma = 0.6$ mV
V_θ	$\mu = -50$ mV, $\sigma = 0.5$ mV	$\mu = -50$ mV, $\sigma = 0.5$ mV
V_{reset}	$\mu = -60$ mV, $\sigma = 1.2$ mV	$\mu = -60$ mV, $\sigma = 0.6$ mV
I_{tonic}	90 pA	90 pA
T_{delay}	$\mu = 1$ ms, $\sigma = 0.02$ ms	$\mu = 1$ ms, $\sigma = 0.01$ ms
τ_{ref}	$\mu = 2$ ms, $\sigma = 0.04$ ms	$\mu = 2$ ms, $\sigma = 0.02$ ms
G_{ex}	$\mu = 20$ pS, $\sigma = 0.4$ pS	$\mu = 30$ pS, $\sigma = 0.03$ pS
G_{in}	$\mu = 160$ pS, $\sigma = 3.2$ pS	$\mu = 140$ pS, $\sigma = 1.4$ pS
τ_{ex}	$\mu = 20$ ms, $\sigma = 0.4$ ms	$\mu = 20$ ms, $\sigma = 0.2$ ms
τ_{in}	$\mu = 20$ ms, $\sigma = 0.4$ ms	$\mu = 80$ ms, $\sigma = 0.8$ ms
E_{ex}	0 mV	0 mV
E_{in}	−80 mV	−80 mV
p_{rnn}	0.1	1
p_{inp}	0.3	1
γ_{rnn}	1	1

$$\tau_r \frac{dh_j}{dt} = -h_j + \frac{1}{\tau_d} \sum_{j=1}^{N} \delta(t - t^{(j)}) \tag{6}$$

where $\tau_r = 6$ ms is the synaptic rise time and $\tau_d = 60$ ms is the synaptic decay time.

2.2. Output and Target Functions

The signal derived from Equations (5) and (6) is sent to the N_{out} linear output units resulting in the final output $\hat{y} = W_{out}^T(t)r(t)$. W_{out} is initialized as a $N_{rnn}^{(ex)} \times N_{out}$ null matrix that is modified according to the learning rule described below (see training procedure). N_{out} is the number of readout units and $N_{rnn}^{(ex)}$ is the number of excitatory neurons in the SRNN.

Unless otherwise stated, the target functions $y \in \mathbb{R}^{N_{out}}$ employed to train the model were generated from white noise with a normal distribution $N(0, 30)$, then low-pass filtered with a cut-off at 6 Hz. To assess network performance, we computed the Pearson correlation between the output of the network ($\hat{y}$) and the target function (y).

2.3. Learning Algorithm for the Readout Unit

We used the recursive least square algorithm (Haykin, 2002) to train the readout units to produce the target functions. The W_{out} weight matrix was updated based on the following equations:

$$W_{out}(t) = W_{out}(t - \Delta t) - P(t)r(t)e(t)^T, \tag{7}$$

$$e(t) = \hat{y} - y(t). \tag{8}$$

Where the error $e(t)$ was determined by the difference between the values of the readout units obtained with the multiplication of the reservoir's activity with the weights W_{out}, and the target functions' values y at time t. Each weight update was separated by a time interval Δt of 2.5 ms for all simulations. P is a running estimate of the inverse of the correlation matrix of the network rates r (see Equation 5), modified according to Equation (9) and initialized with Equation (10).

$$P(t) = P(t - \Delta t) - \frac{P(t - \Delta t)r(t)r^T(t)P(t - \Delta t)}{1 + r^T(t)P(t - \Delta t)r(t)}, \tag{9}$$

$$P(0) = \frac{I}{\alpha}. \tag{10}$$

where I is the identity matrix and α is a learning rate constant.

2.4. Oscillatory Networks

2.4.1. External Drive

Each oscillatory network obey the same equations as the SRNNs. However, the I_{ext} term in Equation (1) is replaced by a summation of the following step functions:

$$I^{(ex)} = \begin{cases} 0 & \text{if } t_{stim} > t, \ or \ t > t_{end}^{ex} \\ AW_{in} & otherwise \end{cases} \tag{11}$$

$$I^{(in)} = \begin{cases} 0 & \text{if } t_{stim} > t, \ or \ t > t_{end}^{in} \\ AW_{in} & otherwise \end{cases} \tag{12}$$

where $t_{stim} = 500$ ms, t_{end}^{ex} denotes the end of the excitatory pulse and t_{end}^{in} (different across oscillatory networks) is the end of the inhibitory pulse, where $t_{end}^{ex} > t_{end}^{in}$ and $W_{in} \in \mathbb{R}^{N_{osc}}$ representing the connections from the tonic inputs to the oscillatory networks, where $A = 20$ pA.

The oscillatory networks each project to the SRNN with a $N_{rnn} \times (N_{osc} \times N_{inp})$ sparse and static connectivity matrix M, where each oscillatory network projects to the SRNN with a density of $p_{inp} = 0.5$. The non-zero connections are drawn from a normal distribution $\mathcal{N}(0, \sigma_{inp})$, where:

$$\sigma_{inp} = \frac{\gamma_{inp}}{\sqrt{p_{inp} N_{rnn} N_{osc}}} \tag{13}$$

and $\gamma_{inp} = 10$ (a.u.) by default.

Where stated in section 3, the SRNN projects back to the oscillatory networks with a $(N_{osc} \times N_{inp}) \times N_{rnn}$ sparse and static connectivity matrix M', where each SRNN unit projects to the oscillatory networks with a density of $p_{fb} = 0.5$ and $\gamma_{fb} = 0.5$. The non-zero connections are drawn from a normal distribution $\mathcal{N}(0, \sigma_{fb})$, where:

$$\sigma_{fb} = \frac{\gamma_{fb}}{\sqrt{p_{fb} N_{rnn} N_{osc}}}. \tag{14}$$

2.4.2. Selection of Stable Networks

As we were interested in regimes where the networks would produce reliable and repeatable oscillations to be used as an input to our model, we considered networks with an inter-trial correlation coefficient (10 trials) of their mean firing-rate greater than 0.95 as stable. A wide range of parameter combinations lead to reliable oscillations, but different random initializations of networks with the same parameters can lead to drastically different behavior, both in activity type (ansynchronous and synchronous) and inter-trial reliability (**Figure S4**).

2.5. Jitter Accumulation in Input Phase

While a perfect sinusoidal input such as the one in Equation (21) allows for well-controlled simulations, it is unrealistic from a biological standpoint. To address this issue, we added jitter to the input phase of each input unit. This was achieved by converting the static input phase injected into unit k ϕ_k to a random walk $\phi_k(t)$. First, we discretize time into non-overlapping bins of length Δt, such that $t_n = \Delta t * n$. From there, we iteratively define $\phi(t)$ (index k is dropped to alleviate the notation) as:

$$\phi_{t+\Delta t} = \phi_t + \varepsilon_t \Longleftrightarrow \phi_{n+1} = \phi_n + \varepsilon_{n+1} \tag{15}$$

with

$$\phi(0) = \phi_0 \tag{16}$$

$$\varepsilon \sim \mathcal{N}(0, \sigma_\phi^2 \Delta t) \tag{17}$$

from initial value ϕ_0, sampled from a uniform distribution as specified previously. More intuitively, $\phi(t)$ can be constructed as:

$$\begin{aligned} \phi(0) &= \phi_0 \\ \phi_1 &= \phi_0 + \varepsilon_1 \\ \phi_2 &= \phi_1 + \varepsilon_2 = \phi_0 + \varepsilon_1 \\ &\dots \\ \phi_N &= \phi_0 + \sum_{n=1}^{N} \varepsilon_n \end{aligned} \tag{18}$$

On average, the resulting deviation from the deterministic signal, i.e., $E\left[\phi_N - \phi_0\right]$, is null. On the other hand, one can calculate its variance:

$$\begin{aligned} var\left(\phi_N - \phi_0\right) &= var\left(\phi_0 + \sum_{n=1}^{N} \varepsilon_n - \phi_0\right) \\ &= var\left(\sum_{n=1}^{N} \varepsilon_n\right) \end{aligned}$$

Since all ε_n are i.i.d:

$$\begin{aligned} \sum_{n=1}^{N} var\left(\varepsilon_n\right) &= Nvar\left(\varepsilon_n\right) \\ &= N\sigma_\phi^2 \Delta t \\ \Leftrightarrow var\left(\phi_N - \phi_0\right) &= t\sigma_\phi^2 \end{aligned} \tag{19}$$

For ease of comparison, we can express the equivalent standard deviation in degrees (see **Figure S8**):

$$\sigma_{deg} = \sqrt{t\sigma_\phi^2} \frac{180}{\pi} \tag{20}$$

2.6. Model for Place Cells Sequence Formation

2.6.1. Network Architecture and Parameters

We employed a balanced recurrent network similar to the ones used for all other simulations, with a few key differences. The input consisted of $N_{inp} = 20$ oscillators with periods ranging from 7.5 to 8.5 Hz that densely projected to the SRNN ($p = 1$) and follow:

$$I_{ext}(t) = M(\sin(2\pi ft + \phi) + 1) \tag{21}$$

C was set to 100 pF for all neurons and γ_{rnn} was set to 0.5. We removed the readout unit and connections, and we selected 10 random excitatory cells (N_{place}) as place cells. Those cells had parameters identical to the other SRNN excitatory units, except:

1. We set the resting potential of those cells to the mean of E_L, to avoid higher values that could lead to high spontaneous activity (that in turn can lead to spurious learning).
2. A 600 ms sine wave at 10 Hz with an amplitude of 60 pA was injected in each of the place cells at a given time representing the animal going through its place field.
3. The connections between the input oscillators and the place cells were modified following Equation (22).

We modeled the environmental input (10 Hz depolarization of CA1 place cells) based on a representation of the animal's location that was fully dependent on time. In order to explain phase precession, our model relied on an environmental input of a slightly higher frequency than the background theta oscillation, as suggested in Lengyel et al. (2003).

The learning rule seeks to optimize the connections between the oscillating inputs and the place cells in order to make them fire whenever the right phase configuration is reached (Miall, 1989; Matell and Meck, 2004).

We used a band-pass filter between 4 and 12 Hz to isolate the theta rhythm in the SRNN. We then used a Hilbert transform to obtain the instantaneous phase of the resulting signal.

2.6.2. Learning Algorithm

We developed a correlation-based learning rule (Kempter et al., 1999) inspired by the results obtained by (Bittner et al., 2017):

$$M_{i,k}(t + \Delta t) = M_{i,k} + \alpha I_{ext_k}(t) \tag{22}$$

where $i \in \{1, \dots, N_{place}\}$, $k \in \{1, \dots, N_{inp}\}$ and $\alpha = 0.25$. With this rule, the weight update is only applied when a burst occurs in the place cells. A burst is defined as any spike triplets that occur within 50 ms. In experiments, these post-synaptic bursts were associated with Ca^{2+} plateaus in place cells (Bittner et al., 2017) that lead to a large potentiation of synaptic strength with as few as five pairings. The connections of M were initialized from

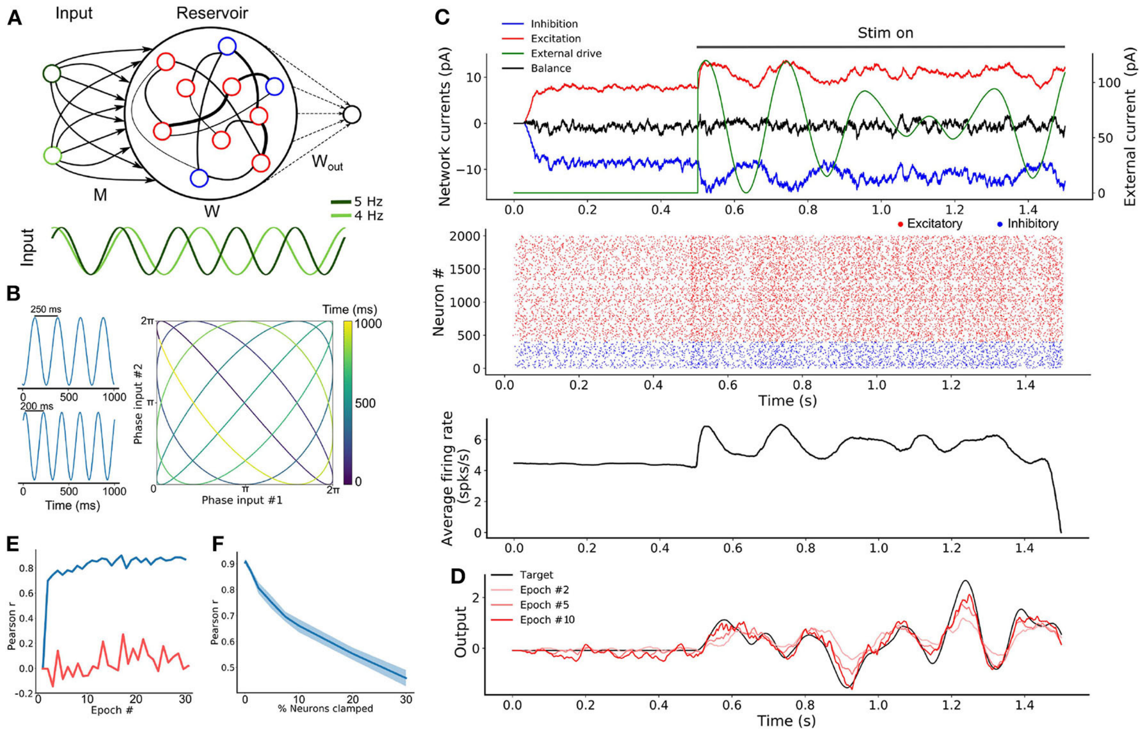

FIGURE 1 | Oscillation driven SRNN to learn complex temporal tasks. **(A)** Schematic of the model's architecture. This implementation has two input units that inject sine-waves in a subset of the SRNN's neurons. *M* denotes the connections from the input units to the SRNN neurons. *W* denotes the recurrent connectivity matrix of the SRNN. W_{out} denotes the trainable connections from the excitatory SRNN neurons to the readout unit. **(B)** Example showing how two sine waves of different frequencies can be combined to generate a two-dimensional function with a period longer than either sine wave. **(C)** Sample average current per neuron, raster and average instantaneous firing rate over one trial, respectively. The external drive is only delivered to a subset of SRNN units ($p_{inp} = 0.3$) for each input unit. **(D)** Output of the readout unit during testing trials (learning rule turned off) interleaved during training. The nearly constant output without the input is due to random activity in the SRNN **(E)** Pearson correlation between the output of the network and the target function as a function of the number of training epochs when driven with (blue) and without (red) multi-periodic input. **(F)** Pearson correlation between the output and the target function as neurons of the SRNN are clamped.

a half-normal distribution $f(0, \sigma_{inp})$, where $\sigma_{inp} = 0.1$ and the signal amplitude A was set to 1. M was bound between 0 and $5\sigma_{inp}$ during training.

2.7. Audio Processing for Speech Learning

We used the numpy/python audio tools from Kastner (2019), adapted by Sainburg (2018), to process the audio WAVE file. We used the built-in functions to convert the audio file to a mel-scaled spectrogram and to invert it back to a waveform.

3. RESULTS

3.1. A Cortical Network Driven by Oscillations

We began with a basic implementation of our model where artificial oscillations served as input to a SRNN (**Figure 1A**)—in a later section, we will describe a more realistic version where recurrent networks generate these oscillations intrinsically. In this simplified model, two input nodes, but potentially more (**Figure S2**), generate sinusoidal functions of different frequencies. These input nodes project onto a SRNN that is a conductance-based leaky integrate-and-fire (LIF) model (Destexhe, 1997) with balanced excitation and inhibition (van Vreeswijk and Sompolinsky, 1996). Every cell in the network is either strictly excitatory or inhibitory, thus respecting Dale's principle.

The combination of N_{inp} input oscillators will generate a sequence of unique N_{inp}-dimensional vectors where the sequence lasts as long as the least common multiple of the inputs' individual periods (Vincent-Lamarre et al., 2016). For instance, two sine waves with periods of 200 and 250 ms would create a multi-periodic input with a period lasting 1,000 ms. This effect can be viewed as a two-dimensional state-space where each axis is an individual sine wave (**Figure 1B**). Using a phase reset of the oscillations on every trial can then evoke a repeatable pattern of activity in the downstream population of neurons. Thus, multiplexed oscillations provide the network with inputs whose timescale largely exceeds that of individual units.

When a SRNN ($N_{res} = 2{,}000$) was injected with oscillations, excitatory and inhibitory populations modulated their activity over time, while the average input currents to individual neurons remained balanced (**Figure 1C**, top panel). To illustrate the benefits of oscillatory inputs on a SRNN, we designed a simple task where a network was trained to reproduce a target function consisting of a time-varying signal generated from low-pass filtered noise (**Figure 1C**, bottom panel). Simulations were split into a training and a testing phase. During the training phase, the network was training when receiving a combination of two oscillatory inputs at 4 and 5 Hz. Synaptic weights from the SRNN to the read-out were adjusted using the recursive least-squares learning algorithm (Haykin, 2002) adapted to spiking units (Nicola and Clopath, 2017). During the testing phase, synaptic weights were frozen and the network's performance was assessed by computing the Pearson correlation between the target function and the network's output. This correlation increased to 0.9 within the first 10 training epochs and remained stable thereafter (**Figures 1D,E**). By comparison, the output of a similar network with no oscillatory inputs (i.e., there is no external input and the SRNN is in a spontaneous chaotic regime) remained uncorrelated to the target function. Thus, oscillatory inputs create rich and reliable dynamics in the SRNN that enabled the read-out to produce a target function that evolved over time in a precise manner. The network displays some tolerance to deviations of the input phase of the oscillations that accumulates over time and Gaussian noise (**Figures S1, S8**).

Next, we investigated the resilience of the network to structural perturbations where a number of individual neurons from the SRNN were "clamped" (i.e., held at resting potential) after training (Vincent-Lamarre et al., 2016). Removing neurons can alter the trajectory of the SRNN upon the presentation of the same stimulus, and it reduces the dimensionality of the readout unit. We trained a network for 10 epochs, then froze the weights and tested its performance on producing the target output. We then gradually clamped an increasing proportion of neurons from the SRNN. The network's performance decreased gradually as the percentage of clamped units increased (**Figure 1F**). Remarkably, the network produced an output that correlated strongly with the target function (correlation of 0.7) even when 10% of neurons were clamped. Further exploration of the model shows a wide range of parameters that yield high performances (**Figure S2**). Oscillatory inputs thus enabled SRNN to produce precise and repeatable patterns of activity under a wide range of modeling conditions. Next, we improved upon this simple model by developing a more biologically-inspired network that generated oscillations intrinsically.

3.2. Endogenously Generated Oscillatory Activity

While our results thus far have shown the benefits of input oscillations when training a SRNN model, we did not consider their neural origins. To address this issue, we developed a model that replaces this artificial input with activity generated by an "oscillator" spiking network acting as a central pattern generator (Marder and Calabrese, 1996).

To do so, we took advantage of computational results showing that sparsely connected networks can transition from

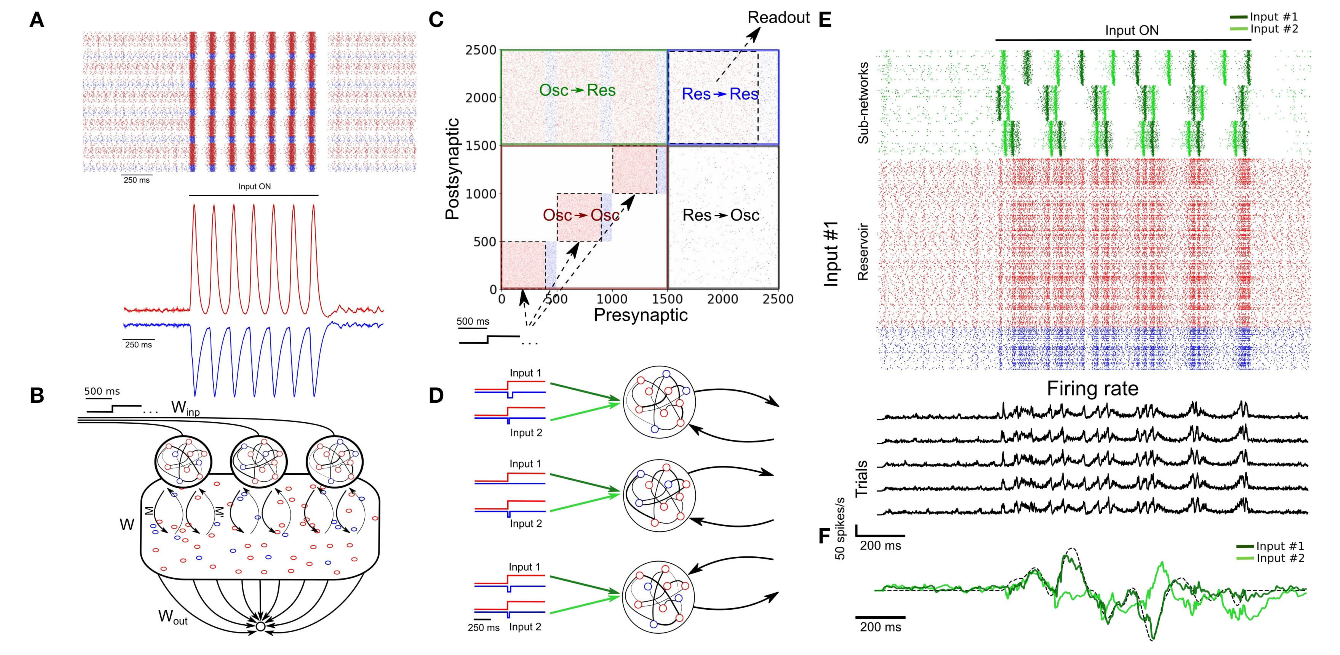

FIGURE 2 | SRNN driven with intrinsically generated oscillations. **(A)** Top: Sample rasters from one network on five different trials. Bottom: Average and standard error of the excitatory (red) and inhibitory (blue) conductance of a stable network on five different trials (see section 2). The network is asynchronous when the external drive is off, and becomes periodic when turned on. **(B)** Architecture of the augmented model. As in **Figure 1A**, *W* and W_{out} denote the recurrent and readout connections, respectively. *M* is the connection matrix from the input to the SRNN, except that the input units are now replaced by networks of neurons. M' denotes the feedback connections from the SRNN to the oscillatory network. W_{inp} denotes the connections providing the tonic depolarization to the oscillatory networks. **(C)** Connectivity matrix of the model. The external drive is provided solely to the oscillatory networks that project to the SRNN that in turn projects to the readout unit. **(D)** External inputs provided to the oscillatory networks with varied inhibitory transients associated with each excitatory input. **(E)** Top: Sample activity of the oscillatory networks (green) on two separate trials with different inputs, and the SRNN for one trial (input #1). Bottom: PSTH of the network's neurons activity on five different trials with input #1. **(F)** Post-training output of the network, where input #1 was paired with the target, but input #2 was not.

an asynchronous to a periodic synchronous regime in response to a step current (Brunel, 2000; Brunel and Hansel, 2006; Thivierge et al., 2014), thus capturing *in vivo* activity (Buzsáki and Draguhn, 2004) (see section 2; **Figure 2A**). The periodicity of the synchronous events could therefore potentially be used to biologically capture the effects of artificially generated sine waves.

In simulations, we found that this transition was robust to both synaptic noise and neuronal clamping (**Figure S3**). Further, the frequency of synchronized events could be modulated by adjusting the strength of the step current injected in the network, with stronger external inputs leading to a higher frequency of events (**Figure S3**). Thus, oscillator networks provide a natural neural substrate for input oscillations into a recurrent network.

From there, we formed a model where three oscillator networks fed their activity to a SRNN (**Figure 2B**). These oscillator networks had the same internal parameters except for the inhibitory decay time constants of their recurrent synapses (τ_{in} = 70, 100, and 130 ms for each network) thus yielding different oscillatory frequencies (**Figures S3, S4**). In order to transition from an asynchronous to a synchronous state, the excitatory neurons of the oscillator networks received a step current.

The full connectivity matrix of this large model is depicted in **Figure 2C**. As shown, the oscillator networks send sparse projections (with a probability of 0.3 between pairs of units) to the SRNN units. Only the excitatory neurons of the SRNN project to the readout units. There is weak feedback from the SRNN to the oscillators—in supplementary simulations, we found that strong feedback projections desynchronized the oscillator networks (**Figures S2, S5**). The feedback connections were included to simulate a case where the oscillators networks would be embedded in the SRNN (and therefore exhibit some degree of local connectivity), instead of sending efferent connections from an upstream brain region.

A sample of the full model's activity is shown in **Figure 2E**. Both the oscillator and the SRNN showed asynchronous activity until a step current was injected into the excitatory units of the oscillator networks. In response to this step current, both oscillator and SRNN transitioned to a synchronous regime. The

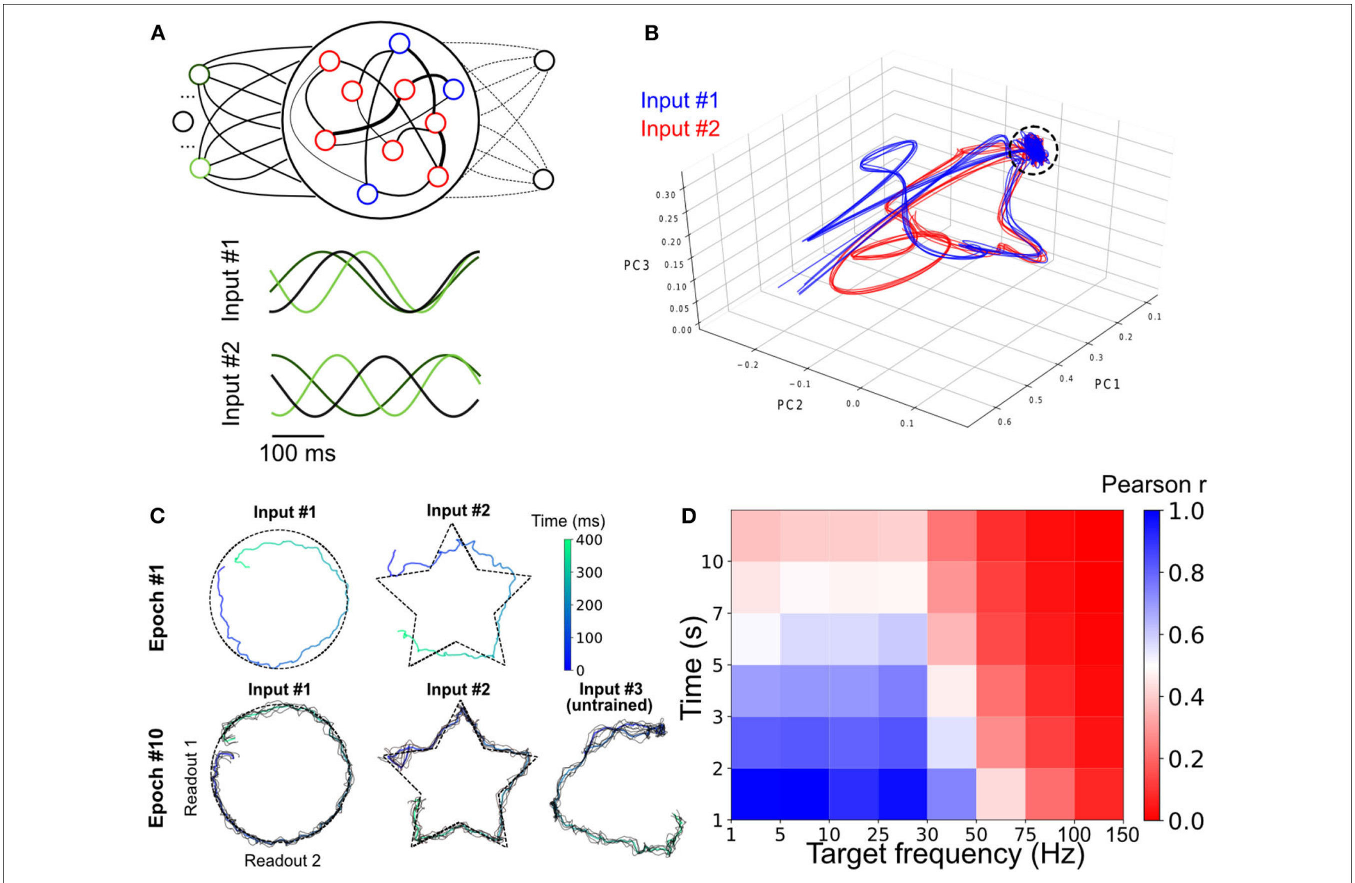

FIGURE 3 | Parallel training of multiple tasks. **(A)** Schematic of the model's architecture with an additional readout unit. An example of possible phase shifted inputs is shown at the bottom. **(B)** Trajectory of the SRNN on the first three components of a PCA. Without an external input, the network is spontaneously active and constrained in a subspace of the state-space (depicted by the circle). Upon injection of the input, the SRNN's activity is kicked into the trajectory related to the input. **(C)** Output of the two readout units after the first and tenth training epochs. The colored line of epoch #10 shows the average of 5 trials (black lines). **(D)** Heat map showing the performance of the model on tasks with different lengths and frequencies.

model reverted back to an asynchronous regime once the step current was turned off.

To illustrate the behavior of this model, we devised a "cued" task similar to the one described above, where the goal was to reproduce a random time-varying signal. When learning this signal, however, the oscillator networks received a cue ("Input 1") consisting of a combination of excitatory step current and transient inhibitory input (**Figure 2D**) that alters the relative phase of the input oscillators, but not their frequency. Following 20 epochs of training, we switched to a testing phase and showed that the model closely matched the target signal (**Figure 2F**). Crucially, this behavior of the model was specific to the cue provided during training: when a different, novel cue (shaped by inhibitory transients) was presented to the network ("Input 2"), a different output was produced (**Figure 2F**).

In sum, the model was able to learn a complex time-varying signal by harnessing internally-generated oscillations that controlled the ongoing activity of a SRNN. In the following section, we aimed to further explore the computational capacity of the model by training a SRNN on multiple tasks in parallel.

3.3. An Artificial Network That Learns to Multitask

To explore the ability of the model to learn two tasks concurrently, we reverted to our initial model with artificial oscillations, allowing for a more principled control of the input injected to the SRNN. The oscillatory input consisted of three sine waves of different frequencies (**Figure 3A**) (more inputs units lead to better performances, **Figure S2**). We trained this model on two different motor control tasks that required the network to combine the output of two readout units in order to draw either a circle or a star in two dimensions. Here, each of the outputs corresponded to x- and y-coordinates, respectively. The phase of the oscillations ("Input 1" vs. "Input 2") were individually paired with only one of the two tasks in alternation (**Figure 3A**).

We employed a principal component analysis (PCA) to visualize the activity of the SRNN before and during training (**Figure 3B**). Before injecting the oscillatory inputs, the network generated spontaneous activity that occupied a limited portion of the state space (**Figure 3B**, circle). During training, the oscillatory inputs were turned on, resulting in different trajectories depending on the relative phase of the oscillations. The network thus displayed a distinct pattern of activity for each of the two tasks.

Viewed in two dimensions, the outputs of the network rapidly converged to a circle and a star that corresponded to each of the two target shapes when given each respective input separately (**Figure 3C**). These shapes were specific to the particular phase of the oscillatory input—in a condition where we presented a randomly-chosen phase configuration to the network, the output did not match either of the trained patterns (**Figure 3C**).

Finally, we tested the ability of the model to learn a number of target signals varying in duration and frequency. We generated a number of target functions consisting of filtered noise (as described previously), and varied their duration as well as the cut-off frequencies of band-pass filtering. The network performed optimally for tasks with relatively low frequency (<30 Hz) and shorter duration (<5 s), and had a decent performance for even longer targets (e.g., correlation of $r = 0.5$ for a time of 7 s and a frequency below 30 Hz; **Figure 3D**).

In sum, the model was able to learn multiple tasks in parallel based on the phase configuration of the oscillatory inputs to the SRNN units. The range of target signals that could be learned was dependent upon their duration and frequency. The next section will investigate another aptitude of the network, where a target signal can be rescaled in time without further training.

3.4. Temporal Rescaling of Neuronal Activity

A key aspect of many behavioral tasks based on temporal sequences is that once learned they can be performed faster or slower without additional training. For example, when a new word is learned, it can be spoken faster or slower without having to learn the different speeds separately.

We propose a straightforward mechanism to rescale a learned temporal sequence in the model. Because the activity of the model strongly depends upon the structure of its oscillatory inputs, we conjectured that the model may generate a slower or faster output by multiplying the period of the oscillatory inputs by a common factor. Biologically, such a factor might arise from afferent neural structures that modulate oscillatory activity (Brunel, 2000). Due to the highly non-linear properties of the network, it is not trivial that rescaling the inputs would expand or compress its activity in a way that preserves key features of the output (Goudar and Buonomano, 2018).

To test the above mechanism, we trained a SRNN receiving sine wave inputs to produce a temporal sequence of low-pass filtered random activity. After the pair of input-target was trained for 10 epochs, we tested the network by injecting it with sine waves that were either compressed or expanded by a fixed factor relative to the original inputs (**Figure 4A**). To evaluate the network's ability to faithfully replay the learned sequence, we computed the Pearson correlation between the output of the network (**Figure 4B**) and a compressed or expanded version of the target signal.

Performance degraded gradually with inputs that were expanded or contracted in time relative to the target signal (i.e., as the rescaling factor moved further away from 1) (**Figure 4C**). Further, performance degraded more slowly beyond a rescaling factor of 1.5, particularly when input noise was absent, suggesting some capacity of the network to expand the target signal in time (**Figure S7**). This result offered a qualitative match to experimental findings (Hardy et al., 2018) and the performance of the network was tolerant to small phase deviations resulting from the addition of random jitter in the phase of all the input oscillations (**Figure S8**).

In sum, rescaling the speed of the input oscillations by a common factor lead to a corresponding rescaling of the learned task, with compressed neural activity resulting in more error than expanded activity. The next section examines some

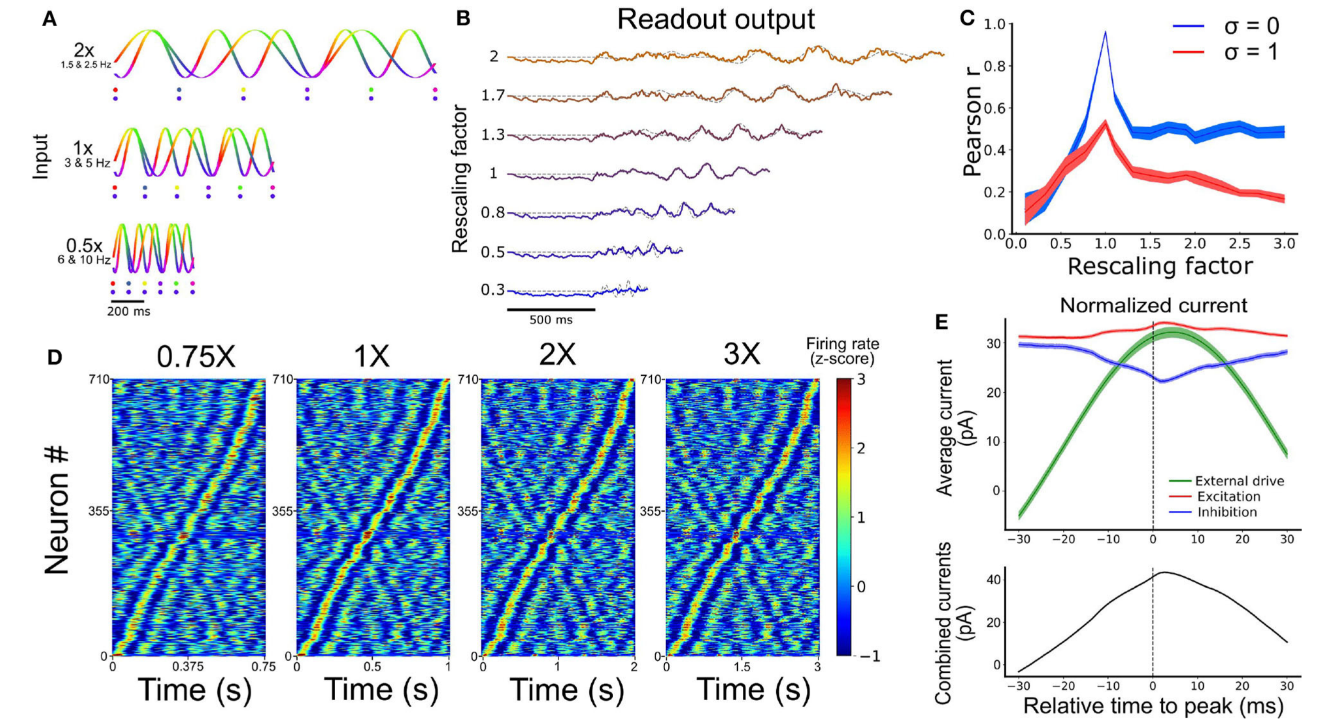

FIGURE 4 | Temporal rescaling of the network's activity. **(A)** Activity of two input units with different rescaling factors. Each phase is represented in a different color. The two dots shown for the different velocities are spaced at a constant proportion of the whole duration, and show that the phase alignment of the two oscillators is preserved across different rescaling factors. **(B)** Output of the network for different rescaling factors. The target is represented by the gray dashed line. **(C)** Pearson correlation between the output and the target function for different rescaling factors and different input noise variance (σ). **(D)** Spiking activity of the RNN at 0.75X, 1X, 2X, and 3X sorted based on the peak of the activity at the original scaling (1X). **(E)** Average excitatory (red), inhibitory (blue), and external input (green) received by each cell aligned on their peak firing rate (at t = 0 ms). The combined inputs (black, external + excitation - inhibition) is represented at the bottom.

of the underlying features of activity in a SRNN driven by oscillatory inputs.

3.5. Temporal Selectivity of Artificial Neurons

A hallmark of temporal processing in brain circuits is that some subpopulations of neurons increase their firing rate at specific times during the execution of a timed task (Harvey et al., 2012; Mello et al., 2015; Bakhurin et al., 2017). To see whether this feature was present in the model, we injected similar oscillatory inputs as above (10 input units between 5 and 10 Hz for 1 s) for 30 trials. To match experimental analyses, we convoluted the firing rate of each neuron from the SRNN with a Gaussian kernel (s.d. = 20 ms), averaged their activity over all trials, and converted the resulting values to a z-score. To facilitate visualization, we then sorted these z-scores by the timing of their peak activity. We retained only the neurons that were active during the simulations (71%). Results showed a clear temporal selectivity whereby individual neurons increased their firing rate at a preferred time relative to the onset of each trial (**Figure 4D**).

These "selectivity peaks" in neural activity were maintained in the same order when we expanded or contracted the input oscillations by a fixed factor (**Figure 4A**), and sorted neurons based on the original input oscillations (**Figure 4D**), thus capturing recent experimental results (Mello et al., 2015). To shed light on the ability of simulated neurons to exhibit temporal selectivity, we examined the timing of excitatory and inhibitory currents averaged across neurons of the SRNN. We then aligned these currents to the timing of selectivity peaks and found elevated activity around the time of trial-averaged peaks (**Figure 4E**). Therefore, both the input E/I currents and the external inputs drive the activity of the neurons near their peak response, showing that both intrinsic and external sources drives the temporal selectivity of individual neurons.

In sum, neurons from the SRNN show sequential patterns of activity by leveraging a combination of external drive and recurrent connections within the network. Next, we examined the ability of the model to learn a naturalistic task of speech production.

3.6. Learning Natural Speech, Fast, and Slow

In a series of simulations, we turned to a biologically and behaviorally relevant task of natural speech learning. This task is of particular relevance to temporal sequence learning given the precise yet flexible nature of speech production: once we learn to pronounce a word, it is straightforward to alter the speed at which this word is spoken without the need for further training. We thus

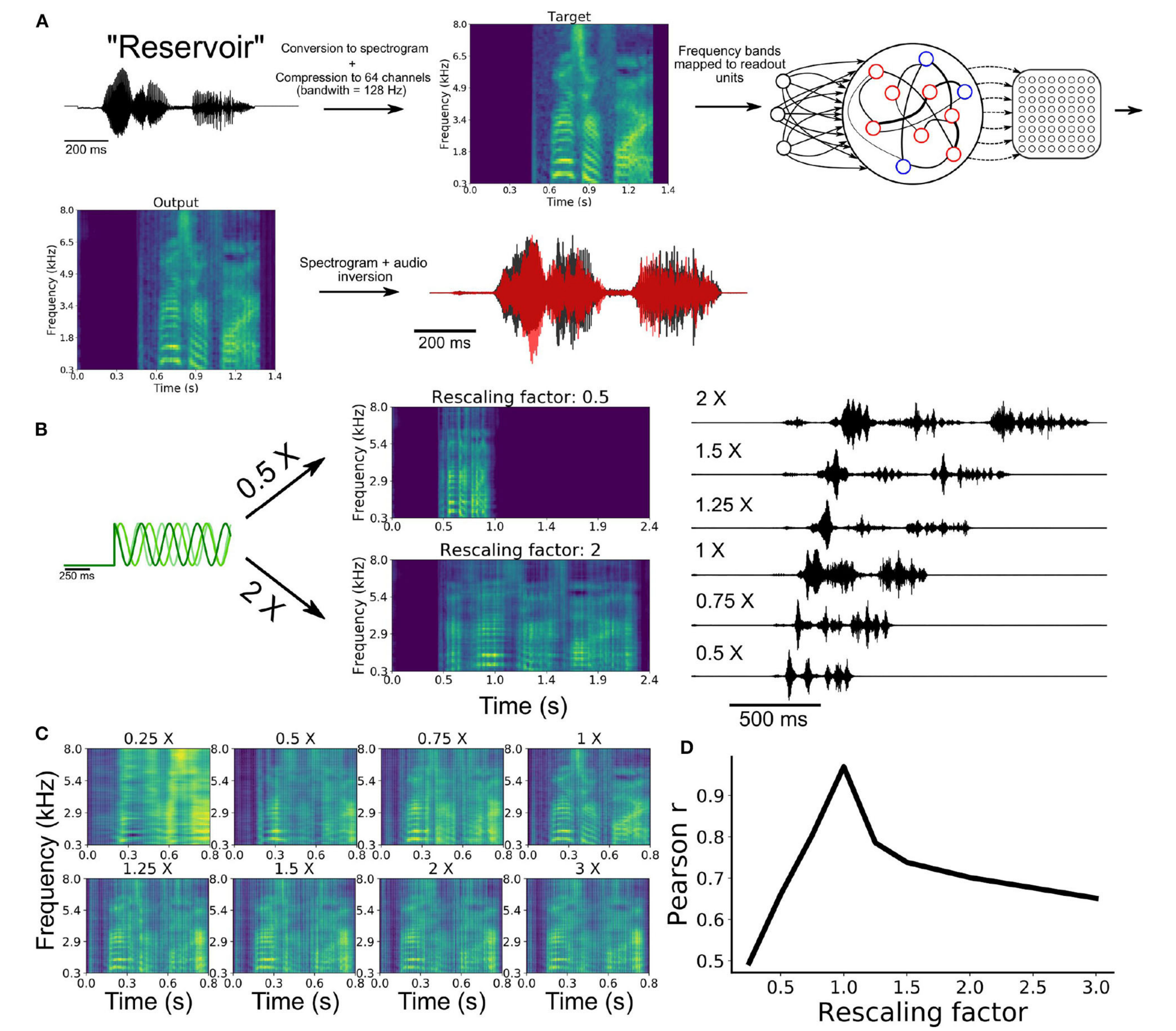

FIGURE 5 | Speech learning and production with temporal rescaling. **(A)** Workflow of the transformation and learning of the target audio sequence. **(B)** The input frequencies are either sped-up or slowed-down in order to induce temporal rescaling of the speed of execution of the task. **(C)** The rescaled output spectrograms are scaled back to the original speed in order to compare them to the target spectrogram. **(D)** Average correlation between the output and target of all channels of the spectrograms for the different rescaling factors.

designed a task where an artificial neural network must learn to utter spoken words in the English language and pronounce them slower or faster given the appropriate input, without retraining.

To train a network on this task, we began by extracting the waveform from an audio recording of the word "reservoir" and converting this waveform to a spectrogram (**Figure 5A**). We then employed a compression algorithm to bin the full range of frequencies into 64 channels spanning a range from 300 Hz to 8 kHz (see section 2). Each of these channels were mapped onto an individual readout unit of the model. Synaptic weights of the SRNN to the readout were trained to reproduce the amplitude of the 64 channels over time. The output spectrogram obtained from the readout units was converted to an audio waveform and compared to the target waveform (see **Movie S1**).

Following training, the network was able to produce a waveform that closely matched the target word (**Figure 5A**). To examine the ability of the network to utter the same word faster or slower, we employed the rescaling approach described earlier, where we multiplied the input oscillations by a constant factor (**Figure 4A**).

Our model was able to produce both faster and slower speech than what it had learned (**Figure 5B**). Scaling the outputs back to the original speed showed that the features of the spectrogram were well replicated (**Figure 5C**). The correlation between the rescaled outputs and target signal decreased as a function of the rescaling factor (**Figure 5D**), in a manner similar to the above results on synthetic signals (**Figure 4C**).

Results thus suggest that multiplexing oscillatory inputs enabled a SRNN to acquire and rescale temporal sequences obtained from natural speech. In the final section below, we employed our model to capture hippocampal activity during a well-studied task of spatial navigation.

3.7. Temporal Sequence Learning During Spatial Navigation

Thus far, we have modeled tasks where units downstream of the SRNN are learning continuous signals in time. In this section, we turned to a task of spatial navigation that required the model to learn a discrete sequence of neural activity.

A wealth of experiments shows that subpopulations of neurons become selectively active for specific task-related time intervals. A prime example is seen in hippocampal theta sequences (Foster and Wilson, 2007) that are observed during

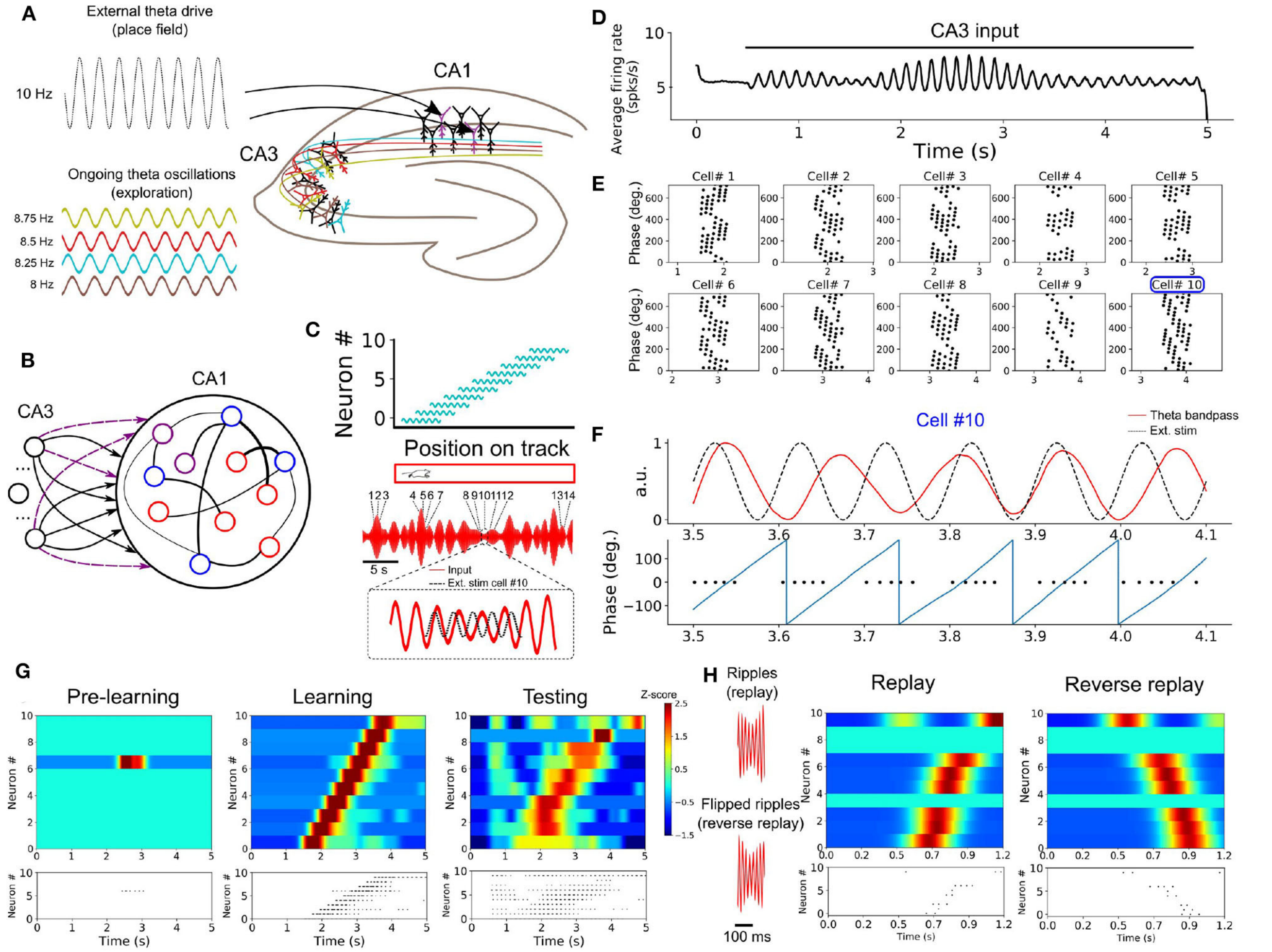

FIGURE 6 | Formation of place cell sequences and replay. **(A)** Subpopulations of CA3 cells oscillating in the theta range and projecting to CA1. CA1 place cells are driven by a slightly faster oscillating input upon entering their place field. **(B)** RC implementation of the phenomenological model. The input layer is composed of oscillatory units (CA3) and CA1 is modeled by a SRNN where 10 excitatory units (purple) are randomly selected as place cells. The connections from the input to the place cells are subject to training. **(C)** Top: Each of the 10 SRNN place cells were driven by a depolarizing oscillating input in a temporal sequence analogous to a mouse moving along a linear track. Bottom: Resulting theta frequency of the combined oscillations (red). Each number shows a place cell activated at a given time along the ongoing theta input. The multi-periodic input from CA3 guarantees that each place cell is activated with a unique combination of the input, following the sequence in which the cell is active. **(D)** Multiplexing CA3 inputs generates a visible theta oscillation in the CA1 SRNN. **(E)** Spike times of the ten place cells in relation to the phase of the population theta activity. Each dot represents a spike at a certain time/position. Each cell shows a shift toward earlier theta phase as the animal moves along its place fields. **(F)** Activity of cell #10 upon entering its place field. Top: theta band-passed activity of the SRNN (red) and place field related input (dashed black). Bottom: phase of the theta oscillation (gray) and spike times. **(G)** Top shows a heatmap of the place cell activity and bottom shows the spike raster during each phase of training. Before training: All place cells are silent. During training: place cells are depolarized upon entering their place field. After training: a similar sequence is evoked without the external stimulation used during training. **(H)** Rescaling the input (factor of X0.15) leads to a high-frequency input reminiscent of ripples. A compressed version of the sequence learned in **(G)** is evoked, and a reversed sequence is evoked when a reverse "ripple" is injected in the SRNN.

spatial navigation in rodents, where individual place cells (O'Keefe, 1976) increase their firing rate at a given location in space (place fields). During spatial navigation, the hippocampus shows oscillatory activity in the theta range (4–12 Hz), likely originating from both the medial septum (Dragoi et al., 1999) and within hippocampus (Traub et al., 1989; Goutagny et al., 2009).

In a series of simulations, we examined how neurons in a SRNN may benefit from theta oscillations to bind and replay such discrete place cells sequences. We designed a SRNN (associated with area CA1, Foster and Wilson, 2007) that received inputs from multiple input oscillators (CA3, Montgomery et al., 2009) (**Figures 6A,B**). We randomly selected 10 excitatory units within the SRNN and labeled them as "place cells." To simulate the response of place cells to an environmental input indicating the spatial location of the animal (Frank et al., 2000), we depolarized these cells by an oscillating input at 10 Hz for 600 ms with a specific onset that differed across neurons in order to capture their respective place fields (assuming a fixed spatiotemporal relation of 100 ms = 5 cm on a linear track). In this way, the sequential activation of place cells from the SRNN mimicked the response of CA1 neurons to an animal walking along a linear track (**Figure 6C**). Although we modeled the task as a sequence of multiple place cells, it could also be performed with individual place cells.

To capture the effect of theta oscillations on CA1 activity, all neurons from the SRNN were driven by a combination of multiple oscillating inputs where the frequency of each input was drawn from a uniform distribution in the range of 7–9 Hz. Connections from the input units and the place cells within the SRNN were modified by a synaptic plasticity rule (see section 2).

As expected from the input oscillators, mean population activity of the SRNN exhibited prominent theta activity (**Figure 6D**). To assess the baseline performance of the model, we ran an initial simulation with oscillatory inputs but no synaptic plasticity or place fields (i.e., no environmental inputs to the place cells). All place cells of the model remained silent (**Figure 6G**). Next, we ran a training phase simulating a single lap of the virtual track lasting 5 s, where place cells received oscillatory inputs (CA3) as well as a depolarizing oscillation (10 Hz) whenever the cell entered its place field. During this lap, individual place cells entered their respective field only once. We assessed the performance of the model during a testing phase where both synaptic plasticity and depolarizing oscillations were turned off. During the testing phase, place cells yielded a clear sequence of activation that matched the firing pattern generated during training (**Figure 6G**). Thus, place cell activity was linked to the phase of the oscillatory inputs after a single lap of exploration.

Going further, we explored two key aspects of place cell activity in the SRNN that are reported in hippocampus, namely phase precession and rapid replay. During phase precession, the phase of firing of place cells exhibits a lag that increases with every consecutive cycle of the theta oscillation (O'Keefe and Recce, 1993; Foster and Wilson, 2007). We examined this effect in the model by extracting the instantaneous phase of firing relative to the global firing rate filtered between 4 and 12 Hz. The activity of individual place cells from the SRNN relative to theta activity exhibited an increasing phase lag characteristic of phase precession (**Figures 6E,F**).

A second feature of hippocampal activity is the rapid replay of place cells during rest and sleep in a sequence that mirrors their order of activation during navigation (Lee and Wilson, 2002). This replay can arise in either a forward or reverse order from the original sequence of activation (Foster and Wilson, 2006). We compressed (factor of 0.15) the CA3 theta oscillations injected in the SRNN during training, resulting in rapid (50–55 Hz) bursts of activity (**Figure 6H**). These fast oscillations mimicked the sharp-wave ripples that accompany hippocampal replay (Lee and Wilson, 2002). In response to these ripples, place cells of the SRNN exhibited a pattern of response that conserved the order of activation observed during training (**Figure 6H**). Further, a reverse replay was obtained by inverting the ripples (that is, reversing the order of the compressed sequence) presented to the SRNN (**Figure 6H**).

In sum, oscillatory inputs allowed individual neurons of the model to respond selectively to external inputs in a way that captured the sequential activation and replay of hippocampal place cells during a task of spatial navigation.

4. DISCUSSION

4.1. Summary of Results

Taken together, our results suggest that large recurrent networks can benefit from autonomously generated oscillatory inputs in order to learn a wide variety of artificial and naturalistic signals, and exhibit features of neural activity that closely resemble neurophysiological experiments.

One series of simulations trained the model to replicate simple shapes in 2D coordinates. Based upon the structure of its oscillatory inputs, the model flexibly switched between two shapes, thus showing a simple yet clear example of multitasking with a recurrent network.

When we modulated the period of input oscillations delivered to neurons of the SRNN, the model was able to produce an output that was faithful to the target signal, but sped up or slowed down by a constant factor (Mello et al., 2015; Hardy et al., 2018). Oscillations served to train a recurrent network that reproduced natural speech and generated both slower and faster utterances of natural words with no additional training. Using further refinements of the model, we employed this principle of oscillation-driven network to capture the fast replay of place cells during a task of spatial navigation.

Below we discuss the biological implications of our model as well as its applications and limitations.

4.2. Biological Relevance and Predictions of the Model

Despite some fundamental limitations common to most computational models of brain activity, our approach was designed with several key features of living neuronal networks, including spiking neurons, Dale's principle, balanced excitation/inhibition, a heterogeneity of neuronal and synaptic parameters, propagation delays, and conductance-based synapses (van Vreeswijk and Sompolinsky, 1996; Sussillo and

Abbott, 2009; Ingrosso and Abbott, 2019). We used a learning algorithm that isn't biologically plausible to train the readout unit (recursive least-square). However, given that the SRNN's dynamics is independent of the readout's output, any other learning algorithm would be compatible with our model. We used a non-biological training algorithm (RLS) in this work, because we are interested in the trained representation of the network and not the learning itself.

Further, and most central to this work, our model included neural oscillations along a range of frequencies that closely matched those reported in electrophysiological studies (Buzsáki and Draguhn, 2004; Yuste et al., 2005). However, the optimal range of frequencies depends on the electrophysiological and synaptic time constants of the network (Nicola and Clopath, 2017). Although there is an abundance of potential roles for neural oscillations in neuronal processing, much of their function remain unknown (Wang, 2010). Here, we proposed that multiple heterogeneous oscillations may be combined to generate an input whose duration greatly exceeds the time-course of any individual oscillation. In turn, this multiplexed input allows a large recurrent network operating in the chaotic regime to generate repeatable and stable patterns of activity that can be read out by downstream units.

It is well-established that central pattern generators in lower brain regions such as the brain stem and the spinal cord are heavily involved in the generation of rhythmic movements that match the period of simpler motor actions (e.g., walking or swimming; Marder and Bucher, 2001). From an evolutionary perspective, it is compelling that higher brain centers would recycle the same mechanisms (Yuste et al., 2005) to generate more complex and non-repetitive actions (Rokni and Sompolinsky, 2011; Churchland et al., 2012). In this vein, Churchland et al. (2012) showed that both periodic and quasi-periodic activity underlie a non-periodic motor task of reaching. Our model provides a framework to explain how such activity can be exploited by living neuronal networks to produce rich dynamics whose goal is to execute autonomous aperiodic tasks.

While our model shows that oscillatory networks can generate input oscillations that control the activity of a SRNN (**Figure 2**), the biological identity of these oscillatory networks is largely circuit-dependent and may originate from either intrinsic or extrinsic sources. In the hippocampus, computational (Traub et al., 1989) and experimental (Goutagny et al., 2009) findings suggest an intrinsic source to theta oscillations. Specifically, studies raise the possible role of CA3 in forming a multi-periodic drive consisting of several interdependent theta generators that activate during spatial navigation (Montgomery et al., 2009). However the exact source of this periodic drive is still debated, where the medial septum and the entorhinal cortex could play a significant role in the theta activity recorded in the hippocampus (Buzsáki, 2002). Additionally, place cells are also found in other subfields of the hippocampus such as CA3 (Lee et al., 2004). Therefore, alternative architectures involving those hippocampal structures could be used to implement our oscillation based model of spatial navigation.

Similarly, the neural origin of the tonic inputs controlling the activity of the oscillatory networks is not explicitly accounted for in our simulations. However, it is well-established that populations of neurons can exhibit bistable activity with UP-states lasting for several seconds (Wang, 2016) that could provide the necessary input to drive the transition from asynchronous to synchronous activity in oscillatory networks.

Going beyond an *in silico* replication of neurophysiological findings, our model makes two empirically testable predictions. If one was to experimentally isolate the activity of the input oscillators, one could show that: (i) a key neural signature of a recurrent circuit driven by multi-periodic oscillations is the presence of inter-trial correlations between the phase of these oscillations; and (ii) the period of the input oscillators should appear faster or slower to match the rescaling factor of the network. This correspondence between the input oscillations and temporal rescaling is a generic mechanism behind the model's ability to perform a wide variety of tasks, from spatial navigation to speech production.

4.3. Related Models

Our approach was inspired by predecessors in computational neuroscience. Multiplexing multiple oscillations as a way to generate long sequences of non-repeating inputs was first introduced by Miall (1989), with a model of interval timing relying on the coincident activation of multiple oscillators of different frequencies. This idea served as a basis for the striatal beat frequency model (Matell and Meck, 2004), where multiple cortical regions are hypothesized to project to the striatum which acts as a coincidence detector that encodes timing intervals. A similar mechanism was also suggested for the representation of space by grid cells in rodents (Fiete et al., 2008). Grid cells have periodic activation curves spanning different spatial periods (Moser et al., 2008), and their activation may generate a combinatorial code employed by downstream regions to precisely encode the location of the animal in space (Fiete et al., 2008).

Other studies have suggested that phase precession during spatial navigation could originate from a dual oscillator process (O'Keefe and Recce, 1993; Burgess et al., 2007). Along this line, a recent model of the hippocampus uses the interference between two oscillators to model the neural dynamics related to spatial navigation (Nicola and Clopath, 2019). Although this model shares similarities with ours, a fundamental difference is that our model uses the phase of combined oscillators to create a unique input at every time-step of a task, whereas their model relies on the beat of the combined frequencies. Additionally, in our model, increasing the frequency of input oscillations by a common factor leads to compressed sequences of activity. By comparison, in the model of Nicola and Clopath (2019), sequences are compressed by removing an extrinsic input oscillator. More experimental data will be needed to support either model.

4.4. Limitations and Future Directions

In our model, periodic activity was readily observable in the SRNN dynamics due to the input drive (e.g., **Figure 1C** or **Figure 6D**). However, the architecture of our model represents a simplification of biological networks where several intermediate stages of information processing occur between sensory input and behavioral output. Oscillatory activity resulting from a multi-periodic drive might occur in one, but not necessarily all stages of processing. Further work could examine this issue by stacking SRNN connected in a feed-forward manner; such a hierarchical organization may have important computational benefits (Gallicchio et al., 2017).

In the spatial navigation task, we ensured that the location of the animal was perfectly correlated with the time spent in the place field of each cell. This is, of course, an idealized scenario that does not account for free exploration and variable speed of navigation along a track. These factors would decorrelate the spatial location of the animal and the time elapsed in the place field. Hence, further work would benefit from a more ecologically-relevant version of the navigation task. This new version of the task might aim to capture how the time spent in a given place field impacts the link between the activity of place cells and theta oscillations (Schmidt et al., 2009).

Finally, our task of speech production was restricted to learning a spectrogram of the target signal. This simplified task did not account for the neural control of articulatory speech kinetics, likely involving the ventral sensory-motor cortex (Conant et al., 2018; Anumanchipalli et al., 2019).

4.4.1. Applications

Our modeling framework is poised to address a broad spectrum of applications in machine learning of natural and artificial signals. With recent advances in reservoir computing (Salehinejad et al., 2017) and its physical implementations (Tanaka et al., 2018), our approach offers an alternative to using external arbitrary time-varying signals to control the dynamics of a recurrent network. Our model may also be extended to neuromorphic hardware, where it may benefit chaotic networks employed in robotic motor control (Folgheraiter et al., 2019). Finally, our model is, to our knowledge, the first to produce temporal rescaling of natural speech, with implications extending to conversational agents, brain-computer interfaces, and speech synthesis.

Overall, our model offers a compelling theory for the role of neural oscillations in temporal processing. Support from additional experimental evidence could impact our understanding of how brain circuits generate long sequences of activity that shape both cognitive processing and behavior.

DATA AVAILABILITY STATEMENT

The custom python scripts replicating the main results of this article can be found at https://github.com/lamvin/Oscillation_multiplexing.

AUTHOR CONTRIBUTIONS

PV-L conceived the model and ran the simulations. PV-L and MC performed the data analysis. PV-L and J-PT designed the experiments. PV-L, MC, and J-PT wrote the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by a doctoral scholarship from the Natural Sciences and Engineering Research Council (NSERC) to PV-L, the program of scholarships of the 2nd cycle from the Fonds de recherche du Québec—Nature et technologies (FRQNT) to MC, and a Discovery grant to J-PT from NSERC (Grant No. 210977).

ACKNOWLEDGMENTS

This manuscript has been released as a pre-print at bioRxiv (Vincent-Lamarre et al., 2020).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fncom.2020.00078/full#supplementary-material

REFERENCES

Abarbanel, H. D., Creveling, D. R., and Jeanne, J. M. (2008). Estimation of parameters in nonlinear systems using balanced synchronization. *Phys. Rev. E* 77:016208. doi: 10.1103/PhysRevE.77.016208

Abbott, L. F., DePasquale, B., and Memmesheimer, R.-M. (2016). Building functional networks of spiking model neurons. *Nat. Neurosci*. 19:350. doi: 10.1038/nn.4241

Anumanchipalli, G. K., Chartier, J., and Chang, E. F. (2019). Speech synthesis from neural decoding of spoken sentences. *Nature* 568:493. doi: 10.1038/s41586-019-1119-1

Bakhurin, K. I., Goudar, V., Shobe, J. L., Claar, L. D., Buonomano, D. V., and Masmanidis, S. C. (2017). Differential encoding of time by prefrontal and striatal network dynamics. *J. Neurosci*. 37:854. doi: 10.1523/JNEUROSCI.1789-16.2016

Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S., and Magee, J. C. (2017). Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science* 357, 1033–1036. doi: 10.1126/science.aan3846

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J. Comput. Neurosci*. 8, 183–208. doi: 10.1023/A:1008925309027

Brunel, N., and Hansel, D. (2006). How noise affects the synchronization properties of recurrent networks of inhibitory neurons. *Neural Comput*. 18, 1066–1110. doi: 10.1162/neco.2006.18.5.1066

Buhusi, C. V., and Meck, W. H. (2005). What makes us tick? Functional and neural mechanisms of interval timing. *Nat. Rev. Neurosci*. 6, 755–765. doi: 10.1038/nrn1764

Buonomano, D. V., and Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nat. Rev. Neurosci*. 10, 113–125. doi: 10.1038/nrn2558

Burgess, N., Barry, C., and O'Keefe, J. (2007). An oscillatory interference model of grid cell firing. *Hippocampus* 17, 801–812. doi: 10.1002/hipo.20327

Buzsáki, G. (2002). Theta oscillations in the hippocampus. *Neuron* 33, 325–340. doi: 10.1016/S0896-6273(02)00586-X

Buzsáki, G., and Draguhn, A. (2004). Neuronal oscillations in cortical networks. *Science* 304, 1926–1929. doi: 10.1126/science.1099745

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., et al. (2012). Neural population dynamics during reaching. *Nature* 487, 51–56. doi: 10.1038/nature11129

Conant, D. F., Bouchard, K. E., Leonard, M. K., and Chang, E. F. (2018). Human sensorimotor cortex control of directly measured vocal tract movements during vowel production. *J. Neurosci*. 38, 2955–2966. doi: 10.1523/JNEUROSCI.2382-17.2018

Destexhe, A. (1997). Conductance-based integrate-and-fire models. *Neural Comput*. 9, 503–514. doi: 10.1162/neco.1997.9.3.503

Dragoi, G., Carpi, D., Recce, M., Csicsvari, J., and Buzsáki, G. (1999). Interactions between hippocampus and medial septum during sharp waves and theta oscillation in the behaving rat. *J. Neurosci*. 19, 6191–6199. doi: 10.1523/JNEUROSCI.19-14-06191.1999

Fiete, I. R., Burak, Y., and Brookings, T. (2008). What grid cells convey about rat location. *J. Neurosci*. 28, 6858–6871. doi: 10.1523/JNEUROSCI.5684-07.2008

Folgheraiter, M., Keldibek, A., Aubakir, B., Gini, G., Franchi, A. M., and Bana, M. (2019). A neuromorphic control architecture for a biped robot. *Robot. Auton. Syst*. 120:103244. doi: 10.1016/j.robot.2019.07.014

Foster, D. J., and Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature* 440:680. doi: 10.1038/nature04587

Foster, D. J. and Wilson, M. A. (2007). Hippocampal theta sequences. *Hippocampus* 17, 1093–1099. doi: 10.1002/hipo.20345

Frank, L. M., Brown, E. N., and Wilson, M. (2000). Trajectory encoding in the hippocampus and entorhinal cortex. *Neuron* 27, 169–178. doi: 10.1016/S0896-6273(00)00018-0

Gallicchio, C., Micheli, A., and Pedrelli, L. (2017). Deep reservoir computing: a critical experimental analysis. *Neurocomputing* 268, 87–99. doi: 10.1016/j.neucom.2016.12.089

Goudar, V., and Buonomano, D. V. (2018). Encoding sensory and motor patterns as time-invariant trajectories in recurrent neural networks. *eLife* 7:e31134. doi: 10.7554/eLife.31134

Goutagny, R., Jackson, J., and Williams, S. (2009). Self-generated theta oscillations in the hippocampus. *Nat. Neurosci*. 12, 1491–1493. doi: 10.1038/nn.2440

Grondin, S. (2010). Timing and time perception: a review of recent behavioral and neuroscience findings and theoretical directions. *Attent. Percept. Psychophys*. 72, 561–582. doi: 10.3758/APP.72.3.561

Hardy, N. F., Goudar, V., Romero-Sosa, J. L., and Buonomano, D. V. (2018). A model of temporal scaling correctly predicts that motor timing improves with speed. *Nat. Commun*. 9:4732. doi: 10.1038/s41467-018-07161-6

Harvey, C. D., Coen, P., and Tank, D. W. (2012). Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature* 484, 62–68. doi: 10.1038/nature10918

Haykin, S. (2002). *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice Hall.

Ingrosso, A. and Abbott, L. F. (2019). Training dynamically balanced excitatory-inhibitory networks. *PLoS ONE* 14:e0220547. doi: 10.1371/journal.pone.0220547

Jaeger, H. (2002). "Adaptive nonlinear system identification with echo state networks," in *NIPS'02: Proceedings of the 15th International Conference on Neural Information Processing Systems* (Vancouver, BC), 593–600.

Kastner, K. (2019). *Audio Tools for Numpy/Python*. Available online at: https://gist.github.com/kastnerkyle/179d6e9a88202ab0a2fe

Kempter, R., Gerstner, W., and van Hemmen, J. L. (1999). Hebbian learning and spiking neurons. *Phys. Rev. E* 59, 4498–4514. doi: 10.1103/PhysRevE.59.4498

Laje, R., and Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci*. 16, 925–933. doi: 10.1038/nn.3405

Lee, A. K., and Wilson, M. A. (2002). Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron* 36, 1183–1194. doi: 10.1016/S0896-6273(02)01096-6

Lee, I., Yoganarasimha, D., Rao, G., and Knierim, J. J. (2004). Comparison of population coherence of place cells in hippocampal subfields CA1 and CA3. *Nature* 430, 456–459. doi: 10.1038/nature02739

Lengyel, M., Szatmáry, Z., and Erdi, P. (2003). Dynamically detuned oscillations account for the coupled rate and temporal code of place cell firing. *Hippocampus* 13, 700–714. doi: 10.1002/hipo.10116

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput*. 14, 2531–2560. doi: 10.1162/089976602760407955

Marder, E. and Bucher, D. (2001). Central pattern generators and the control of rhythmic movements. *Curr. Biol*. 11, R986–R996. doi: 10.1016/S0960-9822(01)00581-4

Marder, E., and Calabrese, R. L. (1996). Principles of rhythmic motor pattern generation. *Physiol. Rev*. 76, 687–717. doi: 10.1152/physrev.1996.76.3.687

Matell, M. S., and Meck, W. H. (2004). Cortico-striatal circuits and interval timing: coincidence detection of oscillatory processes. *Cogn. Brain Res*. 21, 139–170. doi: 10.1016/j.cogbrainres.2004.06.012

Mello, G. B., Soares, S., and Paton, J. J. (2015). A scalable population code for time in the striatum. *Curr. Biol*. 25, 1113–1122. doi: 10.1016/j.cub.2015.02.036

Miall, C. (1989). The storage of time intervals using oscillating neurons. *Neural Comput*. 1, 359–371. doi: 10.1162/neco.1989.1.3.359

Montgomery, S. M., Betancur, M. I., and Buzsáki, G. (2009). Behavior-dependent coordination of multiple theta dipoles in the hippocampus. *J. Neurosci*. 29, 1381–1394. doi: 10.1523/JNEUROSCI.4339-08.2009

Moser, E. I., Kropff, E., and Moser, M.-B. (2008). Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci*. 31, 69–89. doi: 10.1146/annurev.neuro.31.061307.090723

Nicola, W., and Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. *Nat. Commun*. 8:2208. doi: 10.1038/s41467-017-01827-3

Nicola, W., and Clopath, C. (2019). A diversity of interneurons and Hebbian plasticity facilitate rapid compressible learning in the hippocampus. *Nat. Neurosci*. 22:1168. doi: 10.1038/s41593-019-0415-2

O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Exp. Neurol*. 51, 78–109. doi: 10.1016/0014-4886(76)90055-8

O'Keefe, J., and Recce, M. L. (1993). Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus* 3, 317–330. doi: 10.1002/hipo.450030307

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 1310–1318. Available online at: https://dl.acm.org/doi/10.5555/3042817.3043083

Paton, J. J., and Buonomano, D. V. (2018). The neural basis of timing: distributed mechanisms for diverse functions. *Neuron* 98, 687–705. doi: 10.1016/j.neuron.2018.03.045

Rokni, U., and Sompolinsky, H. (2011). How the brain generates movement. *Neural Comput*. 24, 289–331. doi: 10.1162/NECO_a_00223

Sainburg, T. (2018). *Spectrograms, MFCCs, and Inversion in Python*. Available online at: https://timsainburg.com/python-mel-compression-inversion.html

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., and Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv [Preprint]*. arXiv:1801.01078.

Schmidt, R., Diba, K., Leibold, C., Schmitz, D., Buzsáki, G., and Kempter, R. (2009). Single-trial phase precession in the hippocampus. *J. Neurosci*. 29, 13232–13241. doi: 10.1523/JNEUROSCI.2270-09.2009

Sussillo, D., and Abbott, L. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron* 63, 544–557. doi: 10.1016/j.neuron.2009.07.018

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., et al. (2018). Recent advances in physical reservoir computing: a review. *arXiv preprint arXiv:1808.04962*. doi: 10.1016/j.neunet.2019.03.005

Thivierge, J.-P., Comas, R., and Longtin, A. (2014). Attractor dynamics in local neuronal networks. *Front. Neural Circ*. 8:22. doi: 10.3389/fncir.2014.00022

Traub, R. D., Miles, R., and Wong, R. K. S. (1989). Model of the origin of rhythmic population oscillations in the hippocampal slice. *Science* 243, 1319–1325. doi: 10.1126/science.2646715

van Vreeswijk, C., and Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* 274, 1724–1726. doi: 10.1126/science.274.5293.1724

van Vreeswijk, C., and Sompolinsky, H. (1998). Chaotic balanced state in a model of cortical circuits. *Neural Comput*. 10, 1321–1371. doi: 10.1162/089976698300017214

Vincent-Lamarre, P., Calderini, M., and Thivierge, J.-P. (2020). Learning long temporal sequences in spiking networks by multiplexing neural oscillations. *bioRxiv [Preprint]*. 766758. doi: 10.1101/766758

Vincent-Lamarre, P., Lajoie, G., and Thivierge, J.-P. (2016). Driving reservoir models with oscillations: a solution to the extreme structural sensitivity of chaotic networks. *J. Comput. Neurosci.* 41, 305–322. doi: 10.1007/s10827-016-0619-3

Wang, X.-J. (2010). Neurophysiological and computational principles of cortical rhythms in cognition. *Physiol. Rev.* 90, 1195–1268. doi: 10.1152/physrev.00035.2008

Wang, X.-J. (2016). Synaptic reverberation underlying mnemonic persistent activity. *Trends Neurosci.* 24, 455–463. doi: 10.1016/S0166-2236(00)01868-3

Yuste, R., MacLean, J. N., Smith, J., and Lansner, A. (2005). The cortex as a central pattern generator. *Nat. Rev. Neurosci.* 6, 477–483. doi: 10.1038/nrn1686

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.