Efficient reinforcement learning of a reservoir network model of parametric working memory achieved with a cluster population winner-take-all readout mechanism

Zhenbo Cheng,1,2 Zhidong Deng,1 Xiaolin Hu,1 Bo Zhang,1 and Tianming Yang3
1State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, China; 2Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China; and 3Institute of Neuroscience, Key Laboratory of Primate Neurobiology, CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai, China

Submitted 17 April 2015; accepted in final form 7 October 2015

Cheng Z, Deng Z, Hu X, Zhang B, Yang T. Efficient reinforcement learning of a reservoir network model of parametric working memory achieved with a cluster population winner-take-all readout mechanism. J Neurophysiol 114: 3296–3305, 2015. First published October 7, 2015; doi:10.1152/jn.00378.2015.—The brain often has to make decisions based on information stored in working memory, but the neural circuitry underlying working memory is not fully understood. Many theoretical efforts have been focused on modeling the persistent delay period activity in the prefrontal areas that is believed to represent working memory. Recent experiments reveal that the delay period activity in the prefrontal cortex is neither static nor homogeneous as previously assumed. Models based on reservoir networks have been proposed to model such a dynamical activity pattern. The connections between neurons within a reservoir are random and do not require explicit tuning. Information storage does not depend on the stable states of the network. However, it is not clear how the encoded information can be retrieved for decision making with a biologically realistic algorithm. We therefore built a reservoir-based neural network model to model the neuronal responses of the prefrontal cortex in a somatosensory delayed discrimination task. We first illustrate that the neurons in the reservoir exhibit a heterogeneous and dynamical delay period activity observed in previous experiments. Then we show that a cluster population circuit decodes the information from the reservoir with a winner-take-all mechanism and contributes to the decision making. Finally, we show that the model achieves a good performance rapidly by shaping only the readout with reinforcement learning. Our model reproduces important features of previous behavior and neurophysiology data. We illustrate for the first time how task-specific information stored in a reservoir network can be retrieved with a biologically plausible reinforcement learning training scheme.

working memory; reservoir network; reinforcement learning

DECISION MAKING often requires one to compare sensory inputs against contents stored in memory. For example, when shopping at a store one often has to inspect the features of an item and make mental comparisons against the features of another item he just looked at. Such behavior requires the capability of storing sensory information in memory and retrieving it later for decision making. Recent advances in neuroscience show that the persistent activity of neurons in the prefrontal cortex (PFC) plays an important role in working memory (Baddeley 1992; Miller et al. 1996; Padmanabhan and Urban 2010). There have been many theoretical efforts to explain the neural mechanisms of working memory (Barak and Tsodyks 2014). Classic models assume that information in working memory is represented by the persistent activity of neurons in stable network states, which are called attractors. The attractor network models have received some experimental support (Deco et al. 2010; Machens et al. 2005; Miller et al. 2003; Verguts 2007; Wang 2008). However, further studies show that the persistent delay activity in the PFC is highly heterogeneous and dynamical, which is not reflected by this category of recurrent network models (Brody et al. 2003).

A different approach based on reservoir networks has recently started to gain attention in the theoretical field (Buonomano and Maass 2009; Laje and Buonomano 2013). The connection within a reservoir is fixed and task independent. The information is encoded by network dynamics. Task-relevant information retrieval depends on an appropriate readout mechanism. Reservoir networks can model the dynamical and heterogeneous neural activity that has been observed in experiment data (Barak et al. 2013). However, it has been a challenge to implement the training of a task-dependent readout mechanism with a biologically plausible learning scheme. Thus its application in modeling the neural circuits in the brain remains controversial.

In the present study, we show a solution of the readout and learning problem by modeling a somatosensory delayed discrimination task (Romo and Salinas 2003) with a multilayer network. The key components of our network are a reservoir network layer (RNL) and a cluster population layer (CPL) serving as its readout interface (Fig. 1A). First, we show that neurons in the RNL exhibit similar heterogeneity and dynamics as the PFC neuronal activity reported by Romo and colleagues (Brody et al. 2003; Hernandez et al. 2002; Jun et al. 2010). Second, we show that the CPL neurons achieve appropriate readouts via a winner-take-all mechanism within each cluster and the most sensitive neurons are selected for decision making by simple reinforcement learning. Finally, we demonstrate that the CPL greatly improves the efficiency of the reinforcement learning. Together, we manage to create a biologically plausible working memory model based on a reservoir network with a readout layer that can be efficiently trained by reinforcement learning.
MATERIALS AND METHODS

Behavior task. We use the same behavior task as described by Romo and colleagues (Hernandez et al. 2002; Romo and Salinas 2003). Two vibration stimuli are presented with a delay in between. Each stimulus lasts for 500 ms. The delay is set to 3,000 ms in the training sessions and randomly varied between 3,000 ms and 4,000 ms during the testing sessions. The first stimulus occurs at 500 ms after the trial starts. Each trial lasts for 5,000 ms. The stimulus frequencies are randomly chosen between 0.1 and 1.2 Hz. The task is to discriminate whether the frequency of the first or the second vibration is larger.

Neural network model. The model consists of four layers: an input layer (IL), an RNL, a CPL, and a decision making output layer (DML) (Fig. 1A).

The IL is just one neuron, whose firing rate \( f_1 \) is proportional to the vibrational frequencies of stimulus inputs, mimicking the responses of frequency-tuning cells found in S1 (Mountcastle et al. 1990; Salinas et al. 2000). For simplicity, its firing rate is set to the vibrational frequency during the presentation of stimulus; otherwise, it is set to 0.

The input neuron projects to the RNL. The synaptic weight \( w_{ij}^{(0)} \) of the synapse between the input neuron and neuron \( i \) in the RNL circuit is set to 0 with a probability of 0.5. Nonzero weights are assigned independently from a standard uniform distribution [0,1].

In the RNL, there are \( N = 1,000 \) neurons. Each neuron is described by an activation variable \( x_i \) for \( i = 1, 2, \ldots, N \), satisfying

\[
\frac{dx_i}{dt} = -x_i + g \sum_{j=1}^{N} w_{ij} y_j + w_i^{(1)} + \sigma_{\text{noise}} dW_i
\]

where \( \tau = 500 \) ms represents the time constant, \( w_{ij} \) is the synaptic weight between neurons \( i \) and \( j \), \( dW_i \) stands for the white noise, and \( \sigma_{\text{noise}} (= 0.001) \) is the noise variance. The firing rate \( y_i \) of neuron \( i \) is a function of the activation variable \( x_i \) relative to a fixed baseline firing rate \( y_0 \) (= 0.1) and the maximal rate \( y_{\text{max}} \) (= 1):

\[
y_i = \begin{cases} 
y_0 + y_0 \tanh(x/x_0), & x \leq 0 \\
y_0 + (y_{\text{max}} - y_0) \tanh(x/(y_{\text{max}} - y_0)), & x > 0
\end{cases}
\]

The firing rate thus varies between 0 and 1. Neurons in the RNL circuit are sparsely connected with probability \( p = 0.1 \). In other words, the weight \( w_{ij} \) of the synaptic connection between a presynaptic neuron \( j \) and a postsynaptic neuron \( i \) is set to and held at 0 at the probability of 0.9. Nonzero weights are chosen randomly and independently from a Gaussian distribution with zero mean and a variance of \( g^2/N \), where the gain \( g \) acts as the control parameter in the RNL circuit. In our simulations, we set \( g = 1.5 \).

The RNL neurons project to the CPL. The synaptic weight \( w_{ki}^{(2)} \) for the synapse between neuron \( i \) in the RNL and neuron \( k \) in the CPL is set to 0 with probability \( p_{\text{M}} = 0.9 \). Nonzero weights are set independently from a standard uniform distribution [0,1]. The firing rate of neuron \( k \) in the CPL is given as follows: \( u_k = \sum_{l=1}^{M} w_{kl}^{(2)} y_l \) for \( k = 1, 2, \ldots, M \). There are \( M = 50,000 \) neurons in the CPL, grouped into 500 clusters. Each cluster therefore has 100 neurons. Within each cluster, a winner-take-all mechanism selects a winning neuron as its output. The neuron with the maximal average response from the onset of f2 to the end of the trial is selected as the winner. Its firing rate is set to 1.0. The activity of all other neurons within the same cluster is set to zero at the end of a trial.

The CPL projects to the final layer, the DML, which has two competing neurons that correspond to the choices \( f_1 > f_2 \) and \( f_1 < f_2 \), respectively. The firing rate of neuron \( i \) is

\[
y_i = \sum_{l=1}^{2} W_{il}^{(3)} u_k
\]

where \( w_{li}^{(3)} \) represents the weight of the synapse between neuron \( k \) in the CPL circuit and neuron \( i \) in the decision making circuit. Synaptic weights are randomly initialized with uniform distribution [0,0.001] and then updated based on a reward signal during the training phase. We hold these synaptic weights fixed in the test phase for analyses.
The stochastic choice behavior of our model is described by a sigmoidal function:

\[ p_i = \frac{1}{1 + \exp(-\alpha \Delta r)} \]  

(4)

where \( p_i \) represents the probability for the choice \( f_1 > f_2 \), \( \Delta r = v_1 - v_2 \) is the difference between the firing rates of the two DML units, and \( \alpha \) is set to 20.

Reinforcement learning. At the end of trial \( n \), the plastic weights in Eq. 3 in trial \( n + 1 \) are updated as follows:

\[ w_{lk}^{(3)}(n + 1) = w_{lk}^{(3)}(n) + \Delta w_{lk} \]  

(5)

The update term \( \Delta w_{lk} \) depends on the reward prediction error and the responses of the neurons in the CPL circuit:

\[ \Delta w_{lk} = \eta(r - E[r])u_l v_i \]  

(6)

where \( \eta (= 0.001) \) is the learning rate and \( r \) is the reward signal for the current trial (1 if a reward is given and 0 otherwise). \( E[r] \) denotes the average of the previously received rewards, which represents the predicted probability of making a correct choice given \( f_1 \) and \( f_2 \) for that trial (Law and Gold 2009; Seung 2003). Based on the choice for that trial, \( E[r] \) is set to the probability \( p_i \) for the choice \( f_1 > f_2 \) and \( 1 - p_i \) for the choice \( f_1 < f_2 \). The difference between \( r \) and \( E[r] \) thus represents the reward prediction error. After each update, the weights \( w_{lk}^{(3)}(n) \) are normalized by

\[ w_{lk}^{(3)}(n) / \sqrt{\sum_{k=1}^{M} [w_{lk}^{(3)}(n)]^2} \]  

(7)

so that the vector length of \( w_{lk}^{(3)}(n) \) remains constant throughout the training phase. The normalization avoids the weights growing infinitely (Royer and Pare 2003).

Model without CPL. The model consists of only three layers: IL, RNL, and DML. The IL and RNL are the same as described above. However, the neurons in the DML in this model are directly connected with all neurons in the RNL. The firing rate of neuron \( l \) in the DML is

\[ v_l = \sum_{i} w_{li}^{(3)} y_i \text{ for } l = 1, 2, \text{ and } i = 1, 2, \ldots, N \]  

(8)

where \( w_{li}^{(3)} \) represents the weight of the synapse between neuron \( i \) in the RNL circuit and neuron \( l \) in the DML. Synaptic weights are randomly initialized with uniform distribution [0, 0.001] and then updated similarly as in the previous model according to Eqs. 5–7.

Data analysis. The results we present are based on 50 simulation runs unless otherwise noted. In each round, 200 training trials are run. Subsequently, the model is tested with 1,000 trials. All the synaptic weights in the model are reset for each round. The stimulus pairs used in the test phase might not all be presented during the training phase. Our analyses reported here are done with the test data sets, unless otherwise stated.

Chaos analysis. We determine whether the RNL is chaotic by calculating its maximal Lyapunov exponent (MLE) (Verstraeten et al. 2007). A positive MLE indicates that the network is chaotic. We numerically compute the MLE of the RNL as follows (Dechert et al. 1999): Let \( Y \) be an \( N \)-dimensional vector representing the firing rates of RNL neurons at a given time point. A copy \( Y' \) of \( Y \) representing the RNL’s output is generated. Every component of \( Y' \) is perturbed by a small positive number such that the Euclidean distance between \( Y \) and \( Y' \) equals \( d_0 = 10^{-8} \). At each time step, Eq. 1 is solved twice to obtain the new values for \( Y \) and \( Y' \). The distance \( d_t \) between the new values of \( Y \) and \( Y' \) is then calculated. The position of \( Y' \) is readjusted after each iteration to be in the same direction as that of the vector \( Y' - Y \), but with separation \( d_0 \) by adjusting each component \( Y'_{c_j} \) of \( Y' \) according to \( Y'_{c_j} = Y'_{c_j} + (d_0/d_j)(Y'_{c_j} - Y_{c_j}) \). The average of \( \log(d_t/d_j) \) in 4 s gives the MLE.

Simple linear regression model. Following the approach of Romo and Salinas (2003), we model the response of each neuron in the RNL during the \( f_1 \) and the delay periods as follows: firing rate = \( b_1 \times f_1 + b_2 \). A neuron is considered frequency tuned if \( b_1 \) is significantly different from zero more than two-thirds of the time points during the period being considered. The neuron is called early encoding if \( b_1 \) is significant over more than two-thirds of the time points during the first second but not significant during the last 2 s of the delay period. The neuron is called late encoding if \( b_1 \) is significant over more than two-thirds of the time points during the last second but not significant during the first 2 s of the delay period. A neuron is called persistent encoding if \( b_1 \) is significant over more than two-thirds of the time points during the entire delay period. Significance is determined at \( P < 0.01 \).

Full regression model and encoding types. We model the firing rates during the \( f_2 \) period as a linear function of the frequencies of both stimuli, \( f_1 \) and \( f_2 \): FR(\( t \)) = \( b_1(t)f_1 + b_2(t)f_2 + b_3(t) \), where \( t \) represents time and the coefficients \( b_1(t) \) and \( b_2(t) \) are calculated at each time point in a trial. To estimate the coefficients \( b_1 \) and \( b_2 \) through multivariate regression analysis, we consider the regression point, which is reduced to an ellipse that specifies the uncertainty as follows (Jun et al. 2010):

\[ \Delta \beta_{1}(t) S_{11} + \Delta \beta_{2}(t) S_{22} + 2 \Delta \beta_{1}(t) \Delta \beta_{2}(t) S_{12} \leq ps^2(t)F(p, v, 1 - \alpha) \]  

(9)

where \( \Delta \beta_{1} = \beta_{1} - \beta_{2} \) and

\[ S_{ij} = \sum_{t=1}^{T} [f_{1}(t) - \overline{f}_{1}]^{2} \]  

(10)

for \( j = 1, 2, i = 1, 2, \ldots, N_{T} \), and

\[ S_{12} = \sum_{t=1}^{T} (f_{1}(t) - \overline{f}_{1})(f_{2}(t) - \overline{f}_{2}) \]  

(11)

where \( N_{T} \) is the number of trials used in the regression, \( \overline{f}_{i} \) is the mean frequency of the stimulus ensemble over \( N_{T} \) trials, and \( b = [b_1, b_2] \) is the center of an ellipse in the coordinates described by \( \beta = [\beta_1, \beta_2] \). \( F(.) \) is the 1-\( \alpha \) percentile of the F-distribution with 3 parameters and the degrees of freedom \( v = N_{T} - 3 \). The variance is estimated from

\[ s^2(t) = [(Xb^2(t) - FR(t))^2]/(3N_{T}) \]  

(12)

where the regression model \( X \) is defined in the same way as Equation 4 in Jun et al. (2010).

The encoding type of a neuron is characterized by \( b_1 \) and \( b_2 \). A neuron is defined as early \((- f_1) \) encoding at a given time point if its response depends on \( f_1 \) and \( b_1 > 0 \) \((b_1 < 0)\) at that time point. Similarly, a neuron is early \( (f_2) \) encoding if its response depends on \( f_2 \) and \( b_2 > 0 \) \((b_2 < 0)\) at that time point. A neuron is \( f_1 - f_2 \) \((f_2 - f_1) \) encoding if \( b_1 \) and \( b_2 \) have opposite signs and \( b_1 > 0 \) \((b_2 > 0)\).

Winner-take-all competition. We study the importance of the winner-take-all competition in the CPL by varying the competition strength with a softmax function. In each cluster, the average firing rate \( \bar{u} \) of each neuron is computed from the onset \( f_2 \) to the end of a trial, and then normalized as follows:

\[ \bar{u} = \frac{1}{1 + e^{\beta(x - E[u])}} \]  

(13)

where the parameter \( \beta \) adjusts the competition strength and \( E(u) \) is the mean firing rate of all neurons within the cluster. As \( \beta \) increases, the normalization approximates the winner-take-all competition.

Tuning index. We quantify the discriminative sensitivity of each neuron in the CPL by its tuning index. If a neuron wins a group \( a \) times when \( f_1 \) is larger, and \( b \) times when \( f_2 \) is larger, its tuning index is defined as \( (a - b)/(a + b) \).

Neuronal correlation. To show how the CPL activity differentiates stimulus conditions, we calculate the correlation of the CPL neural
activity between the trials when f1 and f2 should lead to the same decision (both are either f1 < f2 or f1 > f2 trials) and between the trials when the stimuli should lead to different decisions (an f1 < f2 trial vs. an f1 > f2 trial). The calculation is based on 100 trials randomly picked from each run. The same trials are used in the calculation of both the same- and the different-condition correlation, but paired differently.

RESULTS

Model’s behavior performance. Our model reproduces the main characteristics of monkeys’ behavioral data in the delayed discrimination task (Hernandez et al. 2002). Performances in different conditions during the test phase are shown in Fig. 1, C–E. Similar to the monkeys’ performance, the model’s performance depends on the difference between the frequencies of the two stimuli, f1 and f2. It approaches 100% when the frequency difference is large and is at the chance level when the two frequencies are equal (Fig. 1, C and E). The performance does not depend on the absolute magnitude of the stimulus frequencies (Fig. 1D; F,0.100 = 0.97, P > 0.1). We reset all synaptic weights of the model in each simulation round, and the behavioral results are not significantly different (LSD t-test, P > 0.1). Thus the network performance is robust and does not depend on initial conditions.

To further test the model’s robustness, we vary two critical parameters of the RNL, namely the gain and the time constant, and calculate the model’s performance. We find that the gain of the RNL does not critically affect the model’s performance when it is between 0.5 and 3.5 (Fig. 2A). Sompolinsky et al. (1988) have shown that the RNL is in a chaotic regime in the absence of external input when the gain of the weight matrix is >1. Similarly, we calculate the maximal Lyapunov exponent for the RNL network and find that the RNL is chaotic when the gain is >3. However, the model achieves a reasonable performance even with a large gain. The model’s performance also does not depend critically on a specific setting of the time constant. It reaches >80% when the time constant is >100 ms (Fig. 2B).

Dynamical and heterogeneous representation of working memory in RNL. One of the key aspects of the delayed discrimination task is that it requires the maintenance of the first stimulus f1 in working memory during the delay period. The mnemonic representation of f1 is dynamical and heterogeneous during the delay period (Barak et al. 2010; Jun et al. 2010; Romo et al. 1999). Neurons in the RNL exhibit similar features.

The responses of many neurons in the RNL are monotonically modulated by the f1 frequency during both the stimulus and delay periods. Most of them fire persistently during the delay period. To quantify the tuning, we use a simple linear regression for each neuron (see MATERIALS AND METHODS for details). Among the total of 1,000 neurons, on average 769 ± 22 (SE) neurons encode f1 during the f1 stimulus period and 687 ± 19 encode f1 during the delay period. The numbers of positively and negatively tuned neurons do not differ significantly.

The neuronal activity during the delay period is not static. The activity patterns of six example neurons are shown in Fig. 3. Some neurons encode f1 early in the delay period, but the encoding disappears later during the delay period (early neurons; average number of neurons = 22; Fig. 3, A and B). Some neurons exhibit sustained frequency encoding during the entire delay period (persistent neurons; average number of neurons = 200; Fig. 3, C and D). Finally, a number of neurons develop frequency encoding relatively late during the last second (late neurons; average number of neurons = 39; Fig. 3, E and F).

We use the regression models to capture each neuron’s frequency tuning at each time point during a trial (see mate-
The number of type V and VI neurons should correlate with the model’s performance, as they encode the frequency difference between f1 and f2. Indeed, when we vary the gain in the RNL, we find that the number of type V and VI neurons varies similarly as the model performance varies (Fig. 4B). When the gain is set to 2, both the model’s performance and the number of type V and VI neurons reach a maximum.

In addition to the great heterogeneity described above, the RNL exhibits a dynamical encoding of stimulus frequencies. To compare the network dynamics with previous findings in the PFC neurons, we show how the population activity represents stimulus frequencies and how this representation evolves during a trial. We calculate an instantaneous population state as a vector of the firing rates of all neurons in the RNL circuit at a given time point (Barak et al. 2010). For each time point in the trial until the end of the delay period, we calculate the correlation between the population states at two frequencies. The correlation reflects how different stimuli drive the population response in the RNL to different states. The dynamics of the network mimics that of the PFC neurons. At the onset of f1, the correlations for all pairs of frequencies drop rapidly, diverging according to the degree of the frequency difference (Fig. 5A). The larger the difference is, the faster and to the lower value the correlations drop, indicating a higher level of response heterogeneity and a better differentiated encoding of the frequencies. The low correlation states last until the stimulus is switched off. Afterwards, the correlations again rise and plateau toward the end of the delay period. The different levels of correlations among different frequency pairs are maintained but degrade throughout the delay period.

### Table 1. Numbers of neurons of each type compared between model and previously reported data (Jun et al. 2010)

<table>
<thead>
<tr>
<th>Type</th>
<th>Model (n = 1,000)</th>
<th>Jun et al. (2010) (n = 912)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type I</td>
<td>68</td>
<td>1.8</td>
</tr>
<tr>
<td>Type II</td>
<td>5.5</td>
<td>4.6</td>
</tr>
<tr>
<td>Type III</td>
<td>12.5</td>
<td>5.2</td>
</tr>
<tr>
<td>Type IV</td>
<td>7</td>
<td>6.2</td>
</tr>
<tr>
<td>Type V</td>
<td>2.5</td>
<td>24.9</td>
</tr>
<tr>
<td>Type VI</td>
<td>1.5</td>
<td>28.6</td>
</tr>
</tbody>
</table>

Neurons of each type (%) compared between the model and the previously reported data [numbers of type I–IV neurons are read off from Fig. 3 in Jun et al. (2010); numbers of type V and VI neurons are quoted directly from Jun et al. (2010)].
To further compare our model against the dynamics of neuronal activity during the delay period described previously, we follow Barak and colleagues’ analysis and calculate two cross-correlations of the population states of the RNL neurons using two reference times (Barak et al. 2010). One is calculated against the population state at the onset of f1 (Fig. 5B, sensory), and the other is against the population state at the end of the delay period (Fig. 5B, memory). The former is a measure of how well the stimulus encoding is maintained, while the latter shows how the memory representation evolves. Consistent with the experimental observations (Barak et al. 2010), the stimulus representation decays during the delay period while the memory representation develops. Both representations coexist in the network, suggesting a mixture of states.

Decoding information from RNL by CPL. In this section, we show how the CPL decodes the information from the RNL for decision making.

We examine the activity of the CPL neurons after the f2 stimulus is presented (Fig. 6). We quantify the discriminative sensitivity of each neuron using a tuning index that indicates how often it wins a cluster when f1 is larger and when f2 is larger (see MATERIALS AND METHODS). The index values of some neurons in the CPL circuit are close to 0, indicating that their winning is independent of the frequency order between f1 and f2. These neurons do not provide useful readout for the task. Many neurons, however, win their competitions more consistently depending on which of the two frequencies is larger (Fig. 6A). These neurons are biggest contributors for the learning. The fraction of the clusters in which different neurons win the competition under f1 > f2 and f1 < f2 conditions increases as the difference between two frequencies grows larger (Fig. 6B).

The performance of the model depends on the parameters of the CPL. First of all, it grows with both the cluster size and the number of the clusters in the CPL (Fig. 7A). With the total number of neurons fixed, however, the number of the clusters turns out to be a larger factor in deciding the model’s performance. Moreover, the performance reaches >80% when the number of the clusters reaches >100, regardless of the cluster size. Second, the CPL depends critically on the winner-take-all mechanism. We vary the competition strength and find that the network performs poorly when the within-cluster competition is weak (Fig. 7B).

To further illustrate how the CPL helps decoding the information from the RNL, we calculate and compare the Pearson correlation coefficients between pairs of neurons for both the CPL and the RNL. While the RNL neurons show high correlations, the distribution of correlation coefficients of the CPL neurons shows a much wider range (Fig. 7C). We further calculate the correlation of the activities of neurons in the CPL between pairs of trials whose f1 and f2 should lead to the same decision (f1 < f2 or f1 > f2) and between pairs of trials whose stimuli should lead to different decisions (f1 < f2 vs. f1 > f2). The CPL neuronal responses show a significantly smaller correlation between the pairs of trials in which the stimuli should lead to different decision outcomes (Fig. 7D). These results indicate that the CPL circuit achieves substantial pattern decorrelation, which facilitates the learning that we describe below.

Reinforcement learning. In the final stage of our model, the CPL neurons project to the DML and form a decision. The weights of connections between the CPL and DML units are chosen randomly initially and updated during the training phase. The update depends on both the neuronal activity and the prediction error. During training, the connections between the DML and the neurons in the CPL with high tuning index values that are consistent with the correct choice are reinforced (Fig. 8A). Such changes are absent in the connections between the DML and the neurons in the CPL with opposite tuning indexes. Within the former group, the ones that have greater frequency sensitivity have stronger synaptic weights and contribute more to the decision after the training (Fig. 8, A and B). The synaptic weights increase rapidly within the first 50 trials and slow down to a steady rate as the training progresses (Fig. 8C). In summary, the training results in an increasingly selective readout of the most sensitive neurons in the CPL network.

The CPL is critical for the whole network to achieve adequate performance via reinforcement learning. To illustrate the importance of the CPL, we test the performance of a model that does not contain a CPL. The output DML neurons in this
model are directly connected with all neurons in the RNL. The synapse weights are initialized randomly and then adjusted by a reinforcement learning algorithm. For the first 100 training trials, the accuracy of the model without CPL remains at the chance level (Fig. 9). In contrast, the accuracy of the model with CPL reaches 80% rapidly with the same amount of training.

DISCUSSION

In this study, we first show that a reservoir network reproduces the response heterogeneity and temporal dynamics observed in the PFC during a delayed discrimination task (Hernandez et al. 2002; Jun et al. 2010; Romo et al. 1999). More importantly, we show that a simple cluster population circuit combined with a reinforcement learning algorithm decodes information from the reservoir and achieves good performance.

Reservoir networks have distinct features that are not found in attractor-based neural network models. A reservoir consists of a large number of neurons that are connected sparsely with random weights. In contrast to previously proposed attractor networks that model similar tasks (Deco et al. 2010; Machens et al. 2005; Miller and Wang 2006; Verguts 2007), the weights...
of synapses in our model do not need to be precisely tuned and the inputs do not depend on specific values (Deco et al. 2010) or gating mechanisms (Machens et al. 2005). The information is sparsely encoded by the reservoir network. The randomness creates a population of mixed selectivity, which has been suggested to be essential in task rule encoding (Rigotti et al. 2010). The encoding of information is not specific to the task. Thereby information stored in the reservoir can be used for different behavior tasks with a different readout scheme.

The dynamics of the RNL is the key to frequency difference tuning observed in some neurons (types V and VI in Fig. 4). These neurons switch their tuning direction for the f1 frequency during the delay period, suggesting that they receive recurrent inputs from neurons with opposite frequency tunings that arrive at different time points. This temporal difference also allows the representation of f1 and f2 to be combined into the frequency difference when the f2 stimulus is presented. Thus the dynamical nature of the RNL is important in encoding task-relevant information.

There are, however, important discrepancies between the RNL neurons and the experimental data previously reported. First of all, although we show that all types of neurons reported previously can be found in the RNL, their proportions differ significantly (Table 1). The type V and VI neurons only compose 4% in the total RNL population, whereas over half of the population were found to be of these two types in the experiment (Jun et al. 2010). This discrepancy cannot be eliminated within the range of parameters we have tested. It is not surprising, as the RNL is not tuned specifically for this task and does not benefit from the training. The neurons recorded in the previous study might reflect an overtrained PFC. Perceptual learning has been shown to have many different stages, which can be explained by plasticity at different brain areas (Shibata et al. 2014). The learning in our model may be considered as only the first stage of learning. The RNL in our model might correspond to a naive PFC. By introducing feedback connections from the CPL and the DML to the RNL, we may train the RNL and obtain results that match the experimental data more closely (Sussillo and Abbott 2009). It would also be interesting to introduce short-term plasticity into the RNL and see how it can affect the network’s performance. It has been suggested that short-term plasticity may play a role in working memory (Itskov et al. 2011; Mongillo et al. 2008). For simplicity, we omit plasticity in our model, but our preliminary results and a previous study (Barak et al. 2013) have suggested that adding plasticity can increase the stimulus encoding efficiency of the RNL. Second, the RNL is agnostic to the temporal structure of the task. For example, the temporal dynamics of type V and VI neurons do not completely mirror real neurons from the data. The sign flipping of some type V and VI neurons in the RNL occurs throughout the delay period, while real neurons only flip their tuning at the f1 offset and the f2 onset (Jun et al. 2010). Again, this discrepancy may be due to the lack of plasticity or feedback inputs in our reservoir model.

The heterogeneous and dynamical responses of the RNL create an obstacle for extracting relevant information from the network. Previous studies used supervised learning for training readout networks, which is not biologically feasible (Deng and Zhang 2007; Jaeger and Haas 2004; Joshi 2007; Sussillo and Abbott 2009). Biologically feasible algorithms such as reinforcement learning directly based on RNL are inefficient. In our network, reinforcement learning without CPL has to depend on type V and VI neurons, which encode the difference between f1 and f2 but are few in the reservoir. The inputs from these relevant neurons to the DML neurons are overwhelmed by other irrelevant neurons.

The winner-take-all cluster structure of CPL is critical in achieving good performance. Each individual neuron in CPL is essentially a randomly sampled readout of the reservoir. The winner-take-all mechanism within a cluster selects the neuron with the largest responses. The losing neurons are not relevant in the learning. As long as different neurons win when f1 > f2 and when f1 < f2, the cluster is able to contribute during the learning. The winner-take-all mechanism amplifies the response difference between two task conditions f1 > f2 and f1 < f2, and greatly enhances the learning efficiency.

The cluster structure of the CPL also helps to improve the temporal robustness of the network performance. We test the network performance with a variable delay period. As the reservoir is dynamic, the responses of its neurons may change dramatically over the time course of the task, especially when the gain is large. Any learning mechanism based on reservoir neurons’ responses directly tends to fail when a temporal variation is introduced into the task. In contrast, the winner-take-all mechanism used in the CPL depends more on neurons’ firing rate differences, which change relatively more slowly compared with instantaneous firing rates. Therefore the CPL becomes much more tolerant to the dynamics of the network and allows the model to achieve reasonable performance even when the gain is large and when temporal variations exist.

The CPL itself is independent from learning. Many CPL neurons do not contribute to this particular task and appear to be a waste. However, these apparently task-irrelevant neurons can be recruited by similar reinforcement learning rules for other tasks in which different stimulus parameters are relevant. Therefore, just as the RNL can serve as a task-independent generic model for working memory, the CPL can serve as a generic readout mechanism for the RNL. The combination of the two can be trained to perform different tasks with a task-specific reinforcement learning paradigm.

Several algorithms in the machine learning fields share features of the CPL. Conceptually, our model is analogous to boosting algorithms used in machine learning (Freund and Schapire 1999). Each cluster within the CPL serves as a weak classifier. The reinforcement learning adjusts the weights assigned to each cluster based on reward feedback and finds a set of weights that combines the outputs from all the clusters to form a much stronger classifier. The winner-take-all mecha-
nism in the CPL is also similar to the max-pooling algorithm used in the convolutional neural networks, which helps to improve the learning efficiency and the robustness of the network (LeCun et al. 2015).

The architecture of the CPL is similar to the columnar organization of neocortex that has been demonstrated in many brain areas and has been suggested to play important roles in learning (Johansson and Lansner 2007; Lucke and von der Malsburg 2004). One of the possible neural substrates of the CPL is the basal ganglia (BG). One can find similar structural features of the CPL in the BG. Within the BG, the striatum receives extensive projections from the PFC. Most striatum neurons send converging axons to the lateral pallidum as their first target (Levesque and Parent 2005). Compared with 31 million striatal spiny neurons, there are far fewer pallidal neurons. It was estimated that on average each pallidal neuron receives inputs from ~100 striatal neurons (Percheron et al. 1984), suggesting a converging information flow and a clustering organization within the striatum (Bolam et al. 2000). In addition, the involvement of the BG in reinforcement learning has been extensively studied. The primary mechanism of learning in the BG model is believed to depend on dopaminergic modulation of neurons that encodes reward prediction error (Cohen and Frank 2009; Joel et al. 2002; Schultz et al. 1997). A network similar to the CPL can also be found in the locust olfactory circuit, where the Kenyon cells in the mushroom body help to decode the information in a sparse manner (Perez-Onive et al. 2002; Rabinovich et al. 2000).

Our network model can be extended in several directions. First of all, we can create a spiking neural network model based on our present model. A spiking model provides further opportunities to study how reservoir networks can model the real brain. Second, our realization of the CPL is simplified. The winner-takes-all mechanism is not implemented with a full-scale firing rate model. However, there are plenty of existing implementations of winner-takes-all competition that are physiologically feasible (Gold and Shadlen 2001; Salinas and Abbott 1996; Wang 2002). Our CPL can be implemented with a network in which an inhibitory unit delivers a common inhibitory signal to all of the excitatory units within the cluster.

In summary, we demonstrate here a biologically feasible way to decode heterogeneous and dynamic information from a reservoir network. This opens up future possibilities of modeling work of various brain functions with reservoir networks. Further experimental work may shed light on whether the brain uses computational mechanisms similar to our model.

ACKNOWLEDGMENTS

We thank Bo Yang for his helpful discussions about echo state networks. We are grateful to Cheng Chen, Chengyu Li, Bailu Si, Xiao-Jing Wang, and Peter Dayan for their comments on the drafts.

GRANTS

This work is supported by Strategic Priority Research Program of the Chinese Academy of Science Grant XDB02050500, the CAS Hundreds of Talents Program, and the Shanghai Pujiang Program to T. Yang and by the National Science Foundation of China (NSFC) under Grants 91420106, 90820305, 60775040, and 61005085 and the Natural Science Foundation of Zhe Jiang (ZJNSF) under Grant Y2111013 to Z. Cheng. We thank the National Supercomputer Center in Tianjin for providing computing resources.

DISCLOSURES

No conflicts of interest, financial or otherwise, are declared by the author(s).

AUTHOR CONTRIBUTIONS

Author contributions: Z.C. conception and design of research; Z.C. performed experiments; Z.C., Z.D., X.H., and T.Y. analyzed data; Z.C. and T.Y. interpreted results of experiments; Z.C. and T.Y. prepared figures; Z.C. and T.Y. drafted manuscript; Z.C., Z.D., X.H., B.Z., and T.Y. edited and revised manuscript; Z.C., Z.D., X.H., B.Z., and T.Y. approved final version of manuscript.

REFERENCES


