RESEARCH ARTICLE

A neural network model for the orbitofrontal cortex and task space acquisition during reinforcement learning

Zhewei Zhang[1,2☯], Zhenbo Cheng[3☯], Zhongqiao Lin[1,2], Chechang Nie[1,2], Tianming Yang[1]*

1 Institute 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, **2** University of Chinese Academy of Sciences, Beijing, China, **3** Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China

☯ These authors contributed equally to this work.
* tyang@ion.ac.cn

OPEN ACCESS

Citation: Zhang Z, Cheng Z, Lin Z, Nie C, Yang T (2018) A neural network model for the orbitofrontal cortex and task space acquisition during reinforcement learning. PLoS Comput Biol 14(1): e1005925. https://doi.org/10.1371/journal.pcbi.1005925

Editor: Samuel J. Gershman, Harvard University, UNITED STATES

Received: July 27, 2017

Accepted: December 13, 2017

Published: January 4, 2018

Data Availability Statement: All relevant data are within the paper and its Supporting Information files. The source code for the model in the manuscript and the corresponding analyses has been uploaded and can be found at https://github.com/tyangLab/ReservoirNet_OFC_TaskState.

Funding: This work is supported by the Chinese Academy of Sciences (http://english.cas.cn/) Hundreds of Talents Program and Science and Technology Commission of Shanghai Municipality (http://www.stcsm.gov.cn/english/)

Abstract

Reinforcement learning has been widely used in explaining animal behavior. In reinforcement learning, the agent learns the value of the states in the task, collectively constituting the task state space, and uses the knowledge to choose actions and acquire desired outcomes. It has been proposed that the orbitofrontal cortex (OFC) encodes the task state space during reinforcement learning. However, it is not well understood how the OFC acquires and stores task state information. Here, we propose a neural network model based on reservoir computing. Reservoir networks exhibit heterogeneous and dynamic activity patterns that are suitable to encode task states. The information can be extracted by a linear readout trained with reinforcement learning. We demonstrate how the network acquires and stores task structures. The network exhibits reinforcement learning behavior and its aspects resemble experimental findings of the OFC. Our study provides a theoretical explanation of how the OFC may contribute to reinforcement learning and a new approach to understanding the neural mechanism underlying reinforcement learning.

Author summary

Many studies employ reinforcement learning models to explain human or animal behavior, in which it is assumed that the animals know the task structure. Yet in the real life, the task structure also has to be acquired through learning. The orbitofrontal cortex has been proposed to play important roles in representing task structure, yet it is poorly understood how it does it and why it can do it. Here, we use a neural network model based on reservoir computing to approach these questions. We show that it is critical for the network to receive reward information as part of its input. Just as the orbitofrontal cortex that receives converging sensory and reward inputs, the network is able to acquire task structure and support reinforcement learning by encoding a combination of sensory and reward events. The importance of reward inputs in the model explains the sophisticate

(15JC1400104) to TY, and by Public Projects of Zhejiang Province (http://www.zjzwfw.gov.cn/) (2016C31G2020069) and the 3rd Level in Zhejiang Province (http://www.zjzwfw.gov.cn/) "151 talents project" to ZC. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

representations of reward information in the orbitofrontal cortex and provides a theoretic account of the current experimental data.

Introduction

Even the simplest reinforcement learning (RL) algorithm captures the essence of operant conditioning in psychology and animal learning [1]. That is, actions that are rewarded tend to be repeated more frequently; actions that are punished are more likely to be avoided. RL requires one to understand the structures of the task and evaluate the value of the states in the task state space. Several studies have investigated the possible brain structures that may be involved in RL [2–6]. Notably, the orbitofrontal cortex (OFC) has been hypothesized to represent the task space and encode task states [7]. Several lesion studies showed that the animals with OFC lesions exhibited deficits acquiring task information for building a task structure [8–10]. Consistent with this idea, single unit recording experiments have revealed that the OFC neurons encode many aspects of task information, including reward value [11–15], probability [16], risk [17], information value [18], abstract rules [19], and strategies [20]. Yet, there is a lack of theoretical understanding how task structures may be encoded and represented by a neural network, and what sort of neuronal firing properties we expect to find in neurophysiological experiments. Furthermore, we do not know how to teach a task-agnostic neural network to acquire the structure of the task just based on reward feedbacks.

In the current study, we provide a solution based on the reservoir network [21–23]. Reservoir networks are recurrent networks with fixed connections. Within a reservoir network, neurons are randomly and sparsely connected. Importantly, the internal states of a reservoir exhibit rich temporal dynamics, which represents a nonlinear transformation of its input history and can be useful for encoding task state sequences. The information encoded by the network can be extracted with a linear output, which can be trained during learning. Reservoir networks have been shown to exhibit dynamics similar to that observed in the prefrontal cortex [24–26]. Furthermore, it has been shown that reservoir networks may be combined with reinforcement learning to learn action values [27].

One key feature of our reservoir-based network model that makes learning task structures possible is including reward itself as an input to the reservoir. Thereby, the network dynamics represents a combination of not only the sensory events, but also the reward outcome. Reinforcement learning helps to shape the output of the reservoir, essentially picking out the action that will lead to the event sequences with desired rewards.

We demonstrate with two commonly used learning paradigms how the network model works. Task event sequences, including reward events, are provided as inputs to the network. A simple yet biologically feasible reward-dependent Hebbian learning algorithm is used to adjust its output weights. We show that our network model can solve problems with different task structures and reproduce behavior experiments previously conducted in animals and humans. We further demonstrate the similarities between the reservoir network and the OFC. Manipulations to our network reproduce the behavior of animals with OFC lesions. Moreover, the reservoir neurons' response patterns resemble characteristics of the OFC neurons reported from previous electrophysiological experiments. Taken together, these results suggest a simple mechanism that naturally leads to the acquisition of task structure and supports RL. Finally, we propose some future experiments that may be used to test our model.

Results

We describe our results in three parts. We start with using our network to model a classical reversal learning task. We take advantage of the simplicity of the task to explain the principal ideas behind the network model and why we think the network resembles the OFC. Then we show such a network may be applied to a more complex scenario, both in the task structure and in the temporal dynamics, in which the OFC has been shown to play important roles. Finally, to further illustrate the similarities between our network model and the OFC, we demonstrate how the selectivity of the neurons in the network may resemble experimental findings in the OFC during value-based decision making.

Reversal learning

In a classical reversal learning task, the animals have to keep track of the reward contingency of two choice options that may be reversed during a test session [9, 28]. Normal animals were found to learn reversals faster and faster, which has been used as an indication of their ability of learning the structure of the task [7]. Such behavior was however found to be impaired in animals with OFC lesions and/or with lesions that contained fibers passing near the OFC [9, 29]. These animals were not able to learn reversals faster and faster when they were repeatedly tested. The learning impairments could be explained by a deficit in acquiring and representing the task structure [7].

Our neural network model consists of a state encoding layer (SEL), which is a reservoir network. It receives three inputs and generates two outputs (Fig 1A). The three inputs from the input layer (IL) to the SEL are the two choice options *A* and *B*, together with a reward input that indicates whether the choice yields a reward or not in the current trial. The outputs units in the decision-making output layer (DML) represent choice actions A and B for the next trial. The inputs are provided concurrently and the neural activity of the SEL at the end of the trial is used to determine the SEL's output (Fig 1B). The connections from the IL to the SEL and the connections within the SEL are fixed. Only the connection weights from the SEL to the DML are modified during the training with a reward dependent Hebbian rule, in which the weight changes are proportional to the reward prediction error and the pre- and post-synaptic neuronal activities.

The network is able to reproduce animals' behavior. The number of the error trials that takes for the network to achieve the performance threshold, which is set at 93% in the initial learning and at 80% in the subsequent reversals, decreases as the network goes through more and more reversals (Fig 1C). Interestingly, a learning deficit similar to that found in OFC-lesion animals is observed if we remove the reward input to the SEL (Fig 1C). As the OFC and its neighboring brain areas such as the ventromedial prefrontal cortex (vmPFC) are known to receive both the sensory inputs and reward inputs from sensory and reward circuitry in the brain [30–32], removing the reward input from our model mimics the situation where the brain has to learn without functional structures in or near the OFC.

Neurons in the SEL, as expected from a typical reservoir network, show highly heterogeneous response patterns. Some neurons are found to encode the stimulus identity, some neurons encode reward, and others show mixed tuning (Fig 2A). A principal component analysis (PCA) based on the population activity shows that the network can distinguish all four possible task states: choice *A* rewarded, choice *A* not rewarded, choice *B* rewarded, and choice *B* not rewarded (Fig 2B and S1 Fig). The first three principal components capture 92.0% variance of the population activity.

The ability to distinguish these states is essential for learning. To understand the task acquisition behavior exhibited by our model, we study how neurons with different selectivity

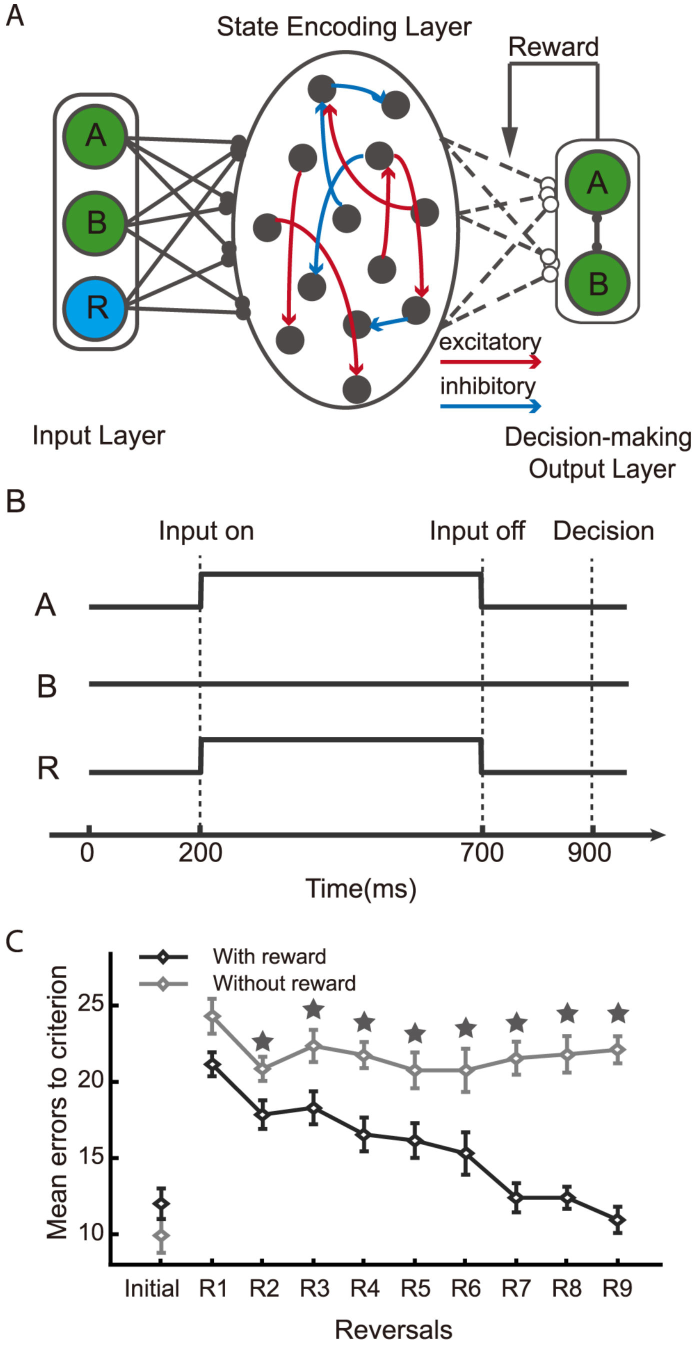

Fig 1. Reversal learning task. A. The schematic diagram of the model. The network is composed of three parts: input layer (IL), the state encoding layer (SEL) and the decision-making output layer (DML). **B.** The event sequence. The stimulus and reward inputs are given concurrently at 200 ms after the trial onset and last for 500 ms. After a 200 ms delay, the decision is computed with the neural activity at 900 ms after the trial onset. **C.** The number of the error trials made before the network achieves the performance threshold. The dark line indicates the performance of the network with the reward input; the light line indicates the performance of the network without the reward input as a model for animals of OFC lesions. Stars indicate significant difference (One-way ANOVA, $p<0.05$).

https://doi.org/10.1371/journal.pcbi.1005925.g001

contribute to the learning (Fig 2C and S2 Fig). We find that readout weights of the SEL neurons that are selective to the combination of stimulus and reward inputs (e.g. *AR* and *BR*) are mostly affected by the learning. The difference between the weights of their connections to the

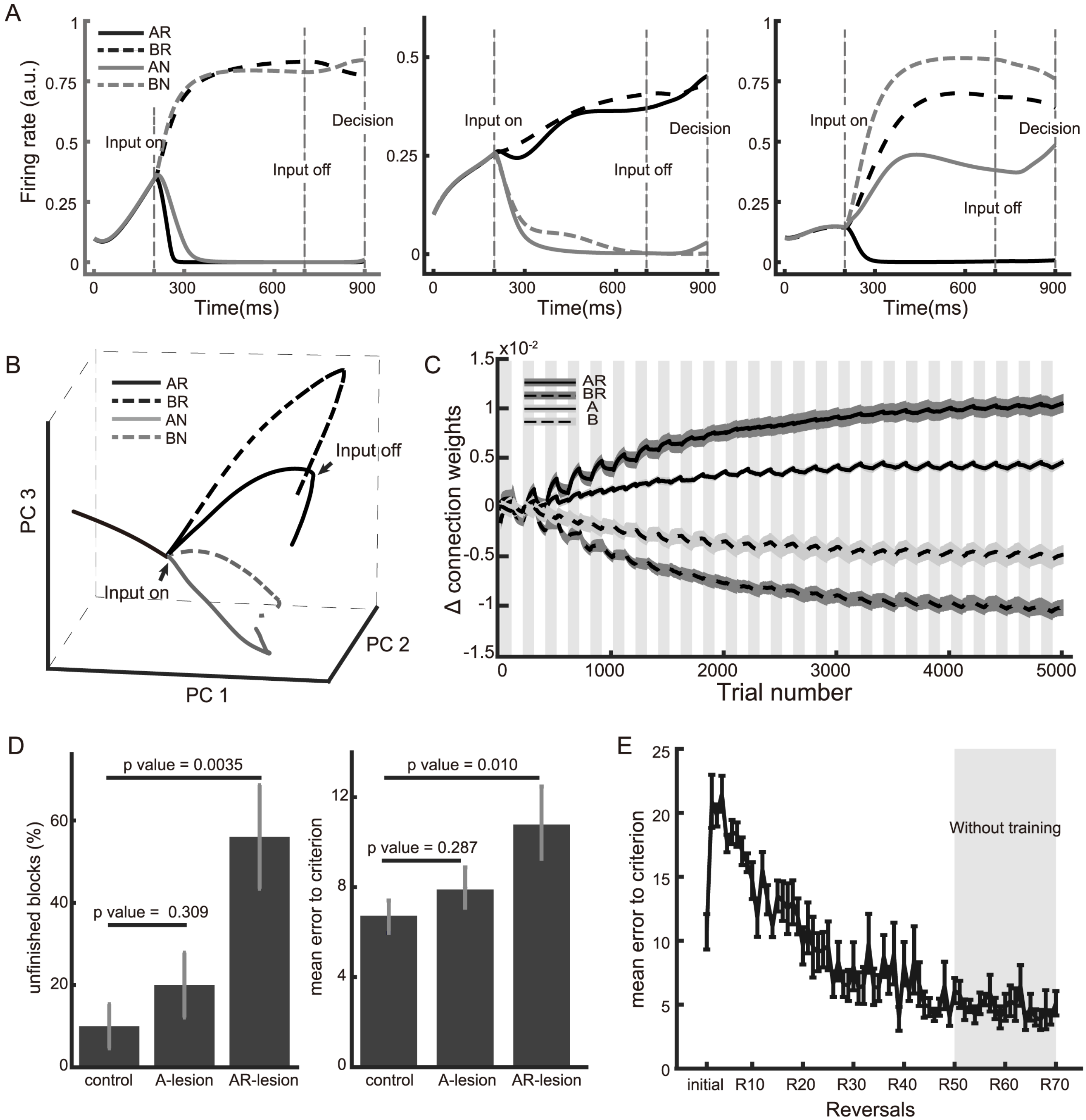

Fig 2. Network analyses for the reversal learning task. A. Selectivity of three example neurons in the reservoir network. Input units are set to 1 from 200ms to 700ms. Left panel: an example neuron that encodes choice options; middle panel: an example neuron that encodes reward outcomes; right panel: an example neuron with mixed selectivity. **B.** PCA on the network population activity. The network states are plotted in the space spanned by the first 3 PCA components. The activities in different conditions are differentiated after the cue onset. **C.** The difference between the SEL neurons' connection weights to DML unit *A* and DML unit *B*. The SEL neurons are grouped according to their selectivities. For example, *AR* represents the group of neurons that respond most strongly when the input units *A* and *R* are both activated. The gray and white area indicates the blocks in which the option *A* and the option *B* leads to the reward, respectively. **D.** Left. The proportion of the blocks in which the network does not reach the performance criterion within a block after we remove 50 neurons that are random chosen (control), A selective, or AR selective. Right. The number of errors that the network makes before reaching the criterion with the same 3 types of inactivation. Only the data from the A-rewarding blocks are analyzed.

The error bars are s.e.m. based on 10 simulation runs. A one-way ANOVA is used to determine the significance ($p<0.05$). **E.** The number of errors needed to reach the performance criterion is maintained after the training stops at the 50th reversal. The error bars are s.e.m. calculated based on 10 simulation runs.

https://doi.org/10.1371/journal.pcbi.1005925.g002

outputs *A* and *B* keeps evolving despite repeated reversals. In contrast, the weights of the output connections of pure stimulus-selective neurons (e.g. *A* and *B*) only wiggle around the baseline between reversals. Once the network is trained, the expected rewards from *AR*/*BN* and *BR*/*AN* inputs are exactly the opposite (S3 Fig).

The difference between these two groups of neurons explains why our network achieves flexible learning behavior only when the reward input is available. Let us first consider the *AR* neurons, which are selective for the situation when choice *A* leads to reward. In these *A*-rewarded blocks, the connections between the *AR* neurons and the DML neuron of choice *A* are strengthened. When the reward contingency is reversed and now choice *A* leads to no reward, the connections between the *AR* neurons and choice *A* are not affected very much. That is because the group of *AN* and then *BR* neurons instead of the *AR* neurons are activated in the blocks when choice *A* is not rewarded. As the result, the connections between the *AN* neurons and the DML neuron of choice *B* are strengthened and the connections between the *AN* neurons and the DML neuron of choice *A* are weakened. When the reward contingency is flipped again, the connections between the *AR* neurons and the DML neuron of choice *A* are strengthened further. This way, the learning is never erased by the reversals, and the network learns faster and faster. In comparison, let us now consider the *A* neurons, which encode only the sensory inputs and are activated whenever input *A* is present. In the *A*-rewarded blocks, the connections between the *A* neurons and the DML neuron of choice *A* are strengthened. In *B*-rewarded blocks, the connections between the *A* neurons and the DML neuron of choice *A* are however weakened when the network chooses *A* and gets no reward, and the learning in the previous block is reversed. Thus, the output connections of *A* neurons only fluctuate around the baseline with the reversals. They do not contribute much to the learning, and the overall behavior of the network is mostly driven by neurons that are activated by the combination of reward input and sensory inputs. Removing *R* deactivates these neurons and leads to the structure agnostic behavior.

The importance of the neurons that are selective for the combination of stimulus and reward inputs can be further illustrated by a simulated lesion experiment. After the network is well-trained, we stop the training and test the network's performance with a proportion of neurons randomly removed at the time of decision (Fig 2D). The neurons that are removed are either 50 randomly chosen neurons, 50 *A* neurons, or 50 *AR* neurons. This inactivation happens only at the time of decision making, therefore the state encoding in the reservoir is not affected. The inactivation of *AR* neurons produces the largest impairment in the network's performance. Compared to the network with random inactivation, the network with *AR*-specific inactivation cannot reach the criterion we set previously within a block in more than 50% of the blocks and makes significantly more errors to reach the criterion in the blocks that it does. Inactivation of A-selective neurons produces much smaller performance deficits.

It is important to note that although the reinforcement learning algorithm employs the same small learning rate for both the intact network and the "OFC-lesion" network, the former only requires a few number of trials to acquire a reversal in the later stage of training, indicating the reversal behavior may not have to be slow with a small learning rate. In fact, once the network is trained, learning is no longer necessary for the reversal behavior. The network takes very few trials to adapt to reversals without learning (Fig 2E). That is because the association between input *AR*/*BN* and decision *A* and the association between input *BR*/*AN* and decision *B* have been established in the network.

Two-stage Markov decision task

We further test our network with a two-stage decision making task. The task is similar to the Markov decision task used previously in several human fMRI studies and used to study the model-based reinforcement learning behavior in humans [6, 33–36]. In this task, the subjects have to choose between two options *A1* and *A2*. Their choices then lead to two intermediate outcomes *B1* and *B2* at different but fixed probabilities. The choice of *A1* more likely leads to *B1*, and the choice of *A2* is more likely followed by *B2*. Importantly, the final reward is contingent only on these intermediate outcomes, and the contingency is reversed across blocks (Fig 3A). Thus, the probability of getting a reward is higher for *B1* in one block and becomes lower in the next block. The probabilistic association between the initial choices and the intermediate outcomes never changes. The subjects are not informed of the structure of the task, and they have to figure out the best option by tracking not only the reward outcomes but also the intermediate outcomes.

We keep our network model mostly the same as in the previous task. Here, we have two additional input units that reflect the intermediate outcomes (Fig 3B). To demonstrate our network model's capability of encoding sequential events, the input units are activated sequentially in our simulations as they are in the real experiment (Fig 3C). We also add a non-reward input unit whose activity is set to 1 when a reward is not obtained at the end of a trial. The additional non-reward input facilitates learning but does not change the results qualitatively.

For a simple temporal difference learning strategy without using any knowledge of task structure, the probability of repeating the previous choice only depends on its reward outcome. The probability of repeating the previous choice is higher when a reward is obtained than when no reward is obtained. The intermediate outcome is ignored. However, this is no longer the case when the task structure is taken into account. For example, consider the situation when the subject initially chooses *A1*, the intermediate outcome happens to be *B2*, and a reward is obtained. If the subject understands *B2* is an unlikely outcome of choice *A1* (rare), but a likely outcome of choice *A2* (common), a reward obtained after the rare event *B2* should actually motivate the subject to switch from the previous choice *A1* and choose *A2* the next time. The subject should always choose the option that is more likely to lead to the intermediate outcome that is currently associated with the better reward.

To quantify the learning behavior, we first evaluate the impact of the previous trial's outcome on the current trial. We classify all trial outcomes into four categories: common-rewarded (*CR*), common-unrewarded (*CN*), rare-rewarded (*RR*) and rare-unrewarded (*RN*). Here, common and rare indicate whether the intermediate outcome is the more likely outcome of the chosen option or not. Glascher et al [6] showed that the model based learning led to a higher probability of repeating the previous choice in the *CR* and *RN* conditions. This is also what we observe in our network model's behavior (Fig 4A).

To illustrate how the network acquires the task structure, we define the task-structure index, which represents the tendency of employing task structure information (see the Method). The task-structure index grows larger as the training goes on (Fig 4B). It indicates that the network learns the structure of the task gradually and transits to a more efficient behavior from an initially task-agnostic behavior.

Similar to our findings in the first task, the network without the reward input in the SEL behaves in a task-agnostic manner. It does not show the transition that indicates the learning of the task structure (Fig 4B). We further quantify the contribution of task structure information to the network behavior using a model fitting procedure previously described by Glascher et al. [6], and the network without the reward input shows a significantly smaller weight for the usage of task structure, suggesting it is worse at picking up the task structure

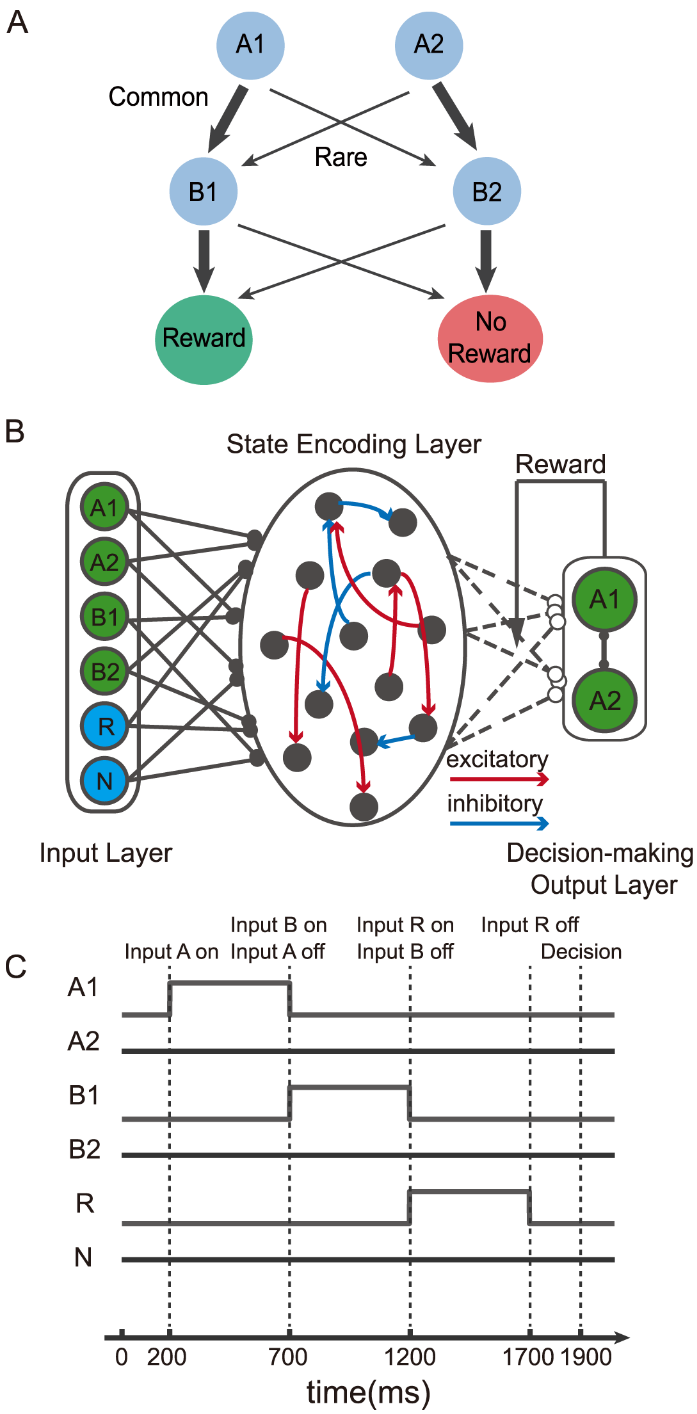

Fig 3. Two-stage Markov decision task. A. Task structure of the two-stage Markov decision task. Two options *A1* and *A2* are available, they lead to two intermediate outcomes *B1* and *B2* at different probabilities. The width of the arrows indicates the transition probability. Intermediate outcomes *B1* and *B2* lead to rewards at different probability, and the reward contingency of the intermediate outcomes is reversed between blocks. **B.** The schematic diagram of the model. It is similar to the model in Fig 1A. The only difference is that there are more input units. **C.** The event sequence. Units in the input layer are activated sequentially. In the example trial, option *A1* is chosen, *B1* is presented, and a reward is obtained.

https://doi.org/10.1371/journal.pcbi.1005925.g003

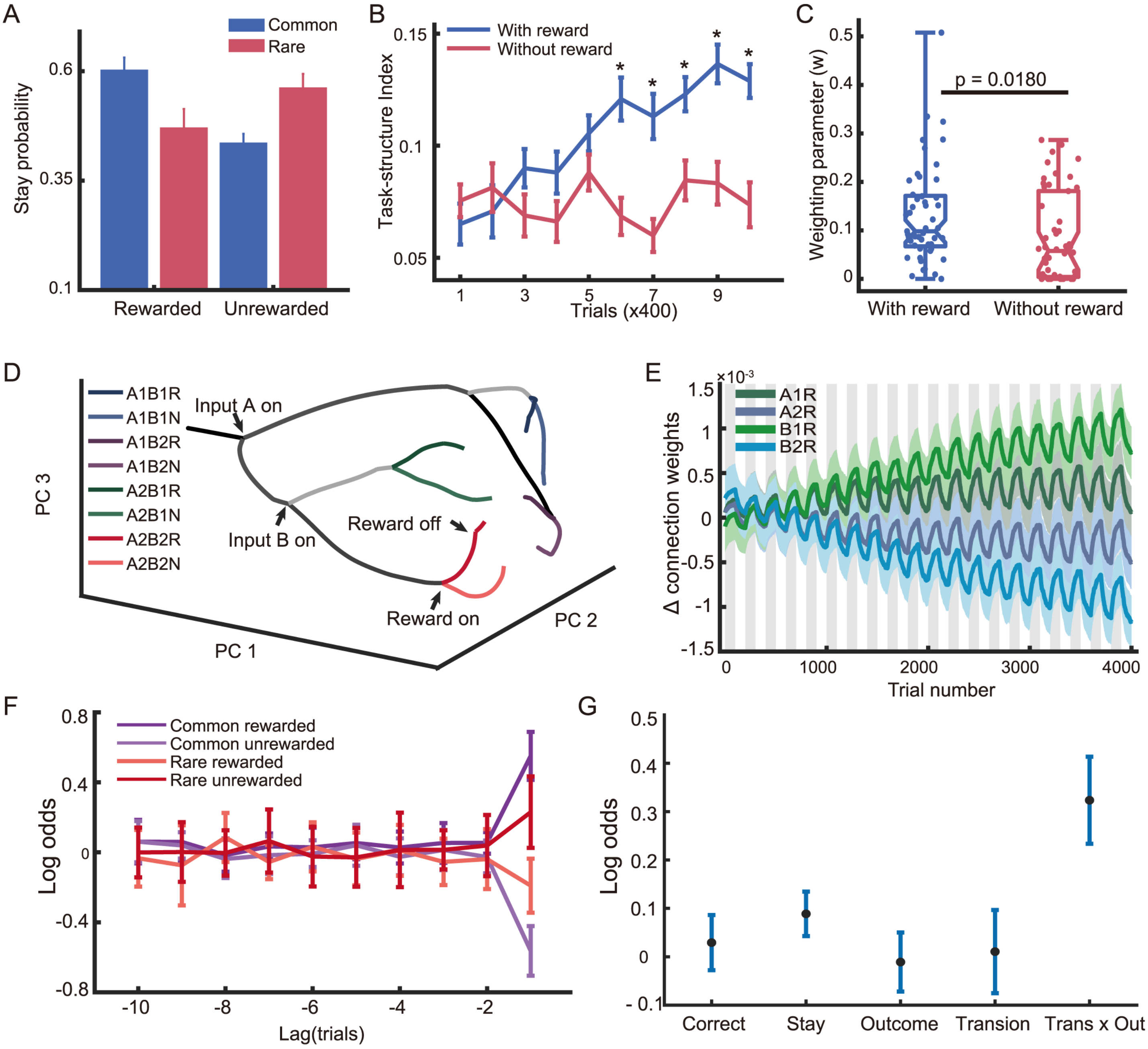

Fig 4. Network analyses for the Two-stage Markov decision task. A. Factorial analysis of choice behavior. The network is more likely to repeat the choice under the conditions common-rewarded (*CR*) and rare-unrewarded (*RN*) than under the conditions common-unrewarded (*CU*) and rare-rewarded (*RR*). **B.** The task structure index keeps growing in the intact network (blue line), but stays at a low level when the reward input is missing (red line). Stars indicate significant difference (One-way ANOVA, p<0.05). **C.** Fitting the behavioral performance with a mixture of task-agnostic and task-aware algorithms. The weight parameter *w* for learning with the knowledge of the task structure is significantly larger for the intact network (blue data points) than the network without the reward input (red data points). Each data point represents a simulation run. A one-way ANOVA is used to determine the significance (p<0.05). **D.** PCA on the network population activity. The network states are plotted in the space spanned by the first 3 PCA components. The network can distinguish all 8 different states. **E.** The weight differences between the connections between SEL neurons and the DML unit *A1* and DML unit *A2*. The gray and white areas indicate the blocks in which intermediate outcome *B1* is more likely to lead to a reward and the blocks in which *B2* is more likely to lead to a reward, respectively. **F.** Logistic regression shows that only the last trial's state affect the choice. The regression includes four different states (intermediate outcome x reward outcome) for each trial up to 10 trials before the current trials. Error bars show s.e.m. across simulation runs. **G.** Logistic regression reveals that only the combination of the intermediate states and the reward outcome in the last trial affects the decision. The factors being evaluated are: Correct—a tendency to choose the better choice in current block; Reward—a tendency to repeat the previous choice if it is rewarded; Stay—a tendency to repeat the previous choice; Transition—a tendency to repeat the same choice following common intermediate outcomes and switch the choice following rare intermediate outcomes; Trans x Out–a tendency to repeat the same choice if a common intermediate outcome is rewarded or a rare intermediate outcome unrewarded, and to switch the choice if a common intermediate outcome is unrewarded or a rare intermediate outcome rewarded.

https://doi.org/10.1371/journal.pcbi.1005925.g004

(Fig 4C and S4 Fig). When the network time constant is sufficiently long, the task-structure dependent behavior is not because the intermediate outcomes occur after the first stage outcomes so that the former having a stronger representation in the network at the time of decision (S5 Fig).

Again, a PCA on the SEL population activity shows that the SEL distinguishes different task states (Fig 4D). The first three principal components explain 83.97% variance of the population activity. Because the structure of the task in which the contingency between the first stage options and the intermediate outcomes is fixed, the network only needs to find out the current reward contingency of the intermediate outcomes. We found that the learning picks out the most relevant neurons that encode the contingency between the intermediate outcomes and the reward outcomes (*B1R*, *B2R*, etc.). Their connection weights to the DML neurons show better and better differentiation of the two choices throughout the training (Fig 4E). In contrast, the connection weights of the neurons that encode the association between the first stage options and the reward outcomes (*A1R*, *A2R*, etc.) are less differentiated.

These results suggest that the network acquires the task structure. It understands that the contingency between intermediate outcomes and reward outcomes is the key to the performance. Thus, its choice only depends on the interaction between the intermediate outcome and the reward outcome of the last trial, but not on the other factors (Fig 4F and 4G). The network behavior is similar to the *Reward-as-cue agent* described by Akam et al. [37].

Value representation by the OFC

Previous electrophysiology studies have shown that OFC neurons encode value during economic choices [11, 13]. In a series of studies carried out by Padoa-Schioppa and his colleagues, monkeys were required to make choices between two types of juice in different amounts. The monkeys' choices depended on both their juice preference and the reward magnitude. Recordings in the OFC revealed multiple classes of neurons encoding a variety of information, including the value of individual offers (offer value), the value of the chosen option (chosen value), and the identity of the chosen option (chosen identity) [38, 39].

Here we show that our network model may explain this apparent heterogeneous value encoding in the OFC. We model the two-alternative economic choice task by providing two inputs to the SEL, representing the reward magnitude of each option with range adaption (Fig 5A). The input dynamics are similar to that of the sensory neurons [40]. The network model reproduces the choice behavior of monkeys (Fig 5C)[11].

Then we study the selectivity of the SEL neurons. Just as in the previous experimental findings in the OFC, we find not only neurons that encode the value of each option (offer value neurons, middle panel in Fig 6A), but also neurons that encode the value of the chosen option (chosen value neurons, left panel in Fig 6A). Furthermore, a proportion of neurons show the selectivity for the choice as previously reported (chosen identity neurons, right panel in Fig 6A). We classify the neurons in the reservoir network into 10 categories as described in Padoa-Schioppa and Assad [11]. Interestingly, we are able to find neurons in the reservoir in 9 of the 10 previous described categories (Fig 6B and 6C). The only missing category (neurons encoding other/chosen value) was also rarely found in the experimental data. Although the proportions of neurons encoding each category are not an exact copy of the experimental data, but the similarity is apparent. This is surprising given that we do not tune the internal connections of the SEL to the task. The results are robust across different input connection gains, noise levels in the SEL, and dynamics of the input profiles (S6 Fig). The heterogeneity that is naturally expected from a reservoir network takes much more effort to explain with recurrent network models that have a well-defined structure [40, 41].

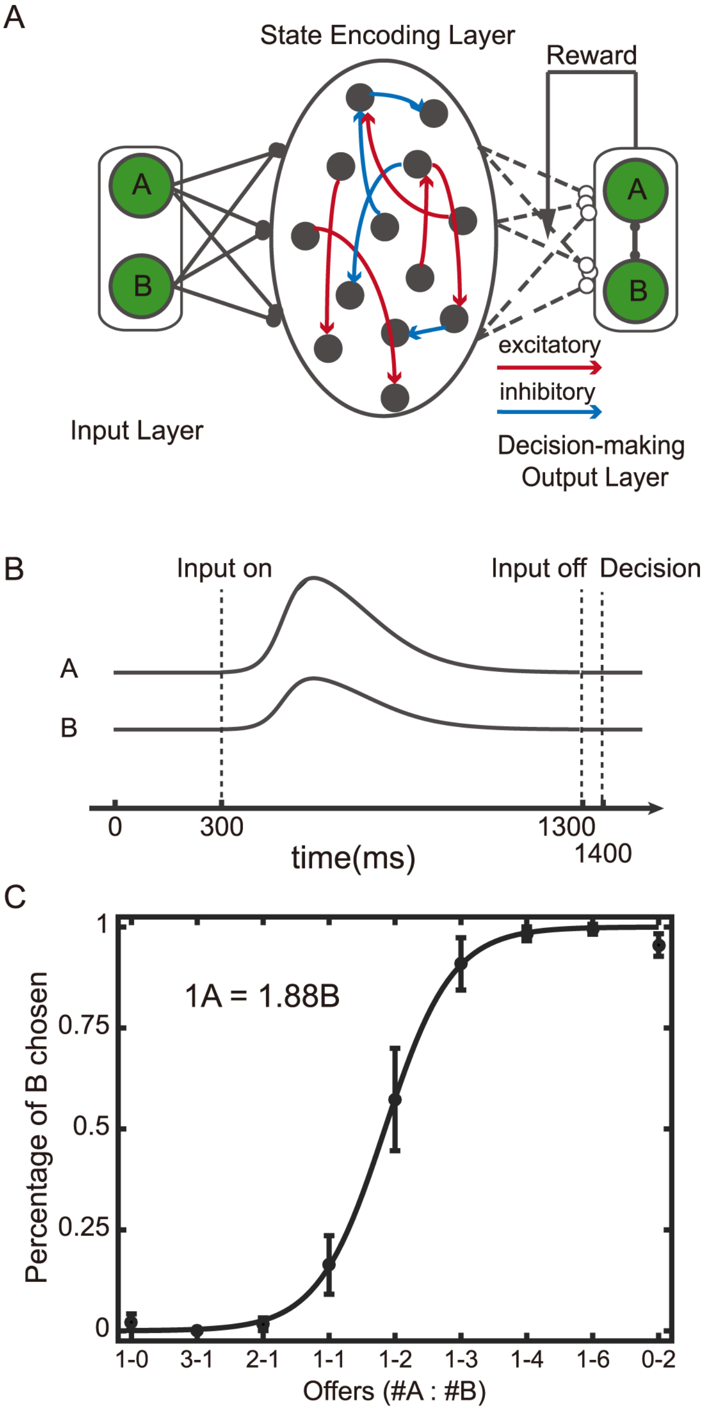

Fig 5. Value-based decision-making task. A. The schematic diagram of the model. **B.** The event sequence. The stimuli are presented between 300 ms and 1300 ms after the trial onset. The decision is computed with the neural activity at 1400 ms after the trial onset. The input neurons' activity profiles mimic those of real neurons (see Methods). **C.** Choice pattern. The relative value preference calculated based on the network behavior is indicated on the top left, and the actual relative value preference used in the simulation is $1A = 2B$.

https://doi.org/10.1371/journal.pcbi.1005925.g005

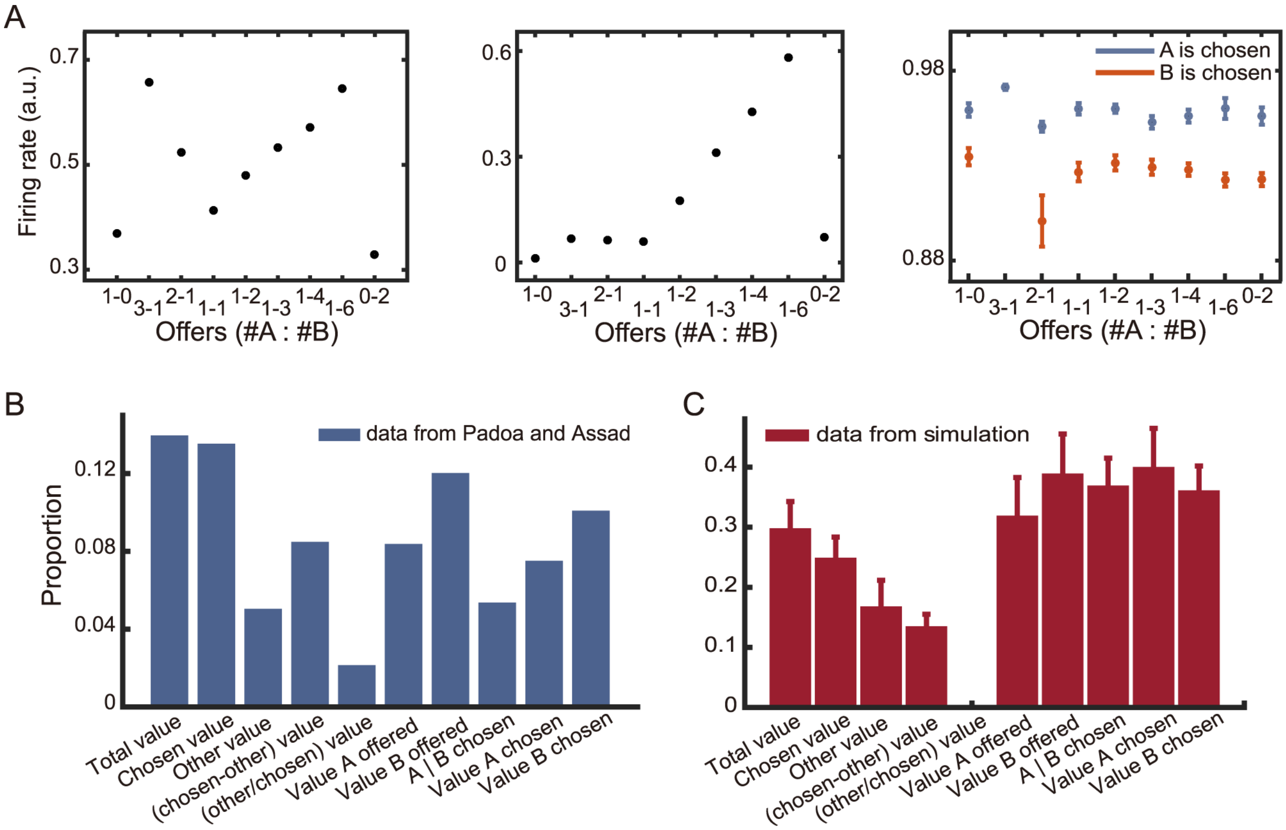

Fig 6. Value selectivity of the network neurons. A. Three example neurons in the SEL. Left panel: a neuron that encodes chosen value; middle panel: a neuron that encodes offer value; right panel: a neuron that encodes chosen juice. **B.** The proportions of the neurons with different selectivities from a previous experimental study [11]. **C.** The proportions of the neurons in the reservoir network with different selectivities.

https://doi.org/10.1371/journal.pcbi.1005925.g006

Discussion

So far, we have shown that a simple reservoir-based network model may acquire task structures. The more interesting question is that why the network is capable of doing so and how this network model may help us to understand the functions of the OFC.

Encoding of the task space

We place a reservoir network as the centerpiece of our model. Reservoir networks are large, distributed, nonlinear dynamical recurrent neural networks with fixed weights. Because of recurrent networks' complicated dynamics, they are especially useful in modeling temporal sequences including languages [42, 43]. Neurons in reservoir networks exhibit mixed selectivity that maps inputs into a high dimensional space. Such selectivity has been shown to be crucial in complex cognitive tasks, and experimental works have provided evidence that neurons in the prefrontal cortex exhibit mixed selectivity [44–46]. In our model, the reservoir network encodes the combinations of inputs that constitute the task state space. States are encoded by the activities of the reservoir neurons, and the learned action values are represented by the weights of the readout connections.

There are several reasons why we choose reservoir networks to construct our model. First reason is that we would like to pair our network model with reinforcement learning. Reservoir networks have fixed internal connections; the training occurs only at the readout. The number of parameters for training is thus much smaller, which could be important for efficient reinforcement learning. Generality is another benefit offered by reservoir networks. Because the internal connections are fixed, we may use the same network to solve a different problem by just training a different readout. The reservoir can serve as a general-purpose task state representation network layer. Lastly, our results as well as several other studies show that neurons in reservoir networks–even with untrained connections weights–show properties similar to that observed in the real brain [24, 25, 47], suggesting training within the network for specific tasks may not play a role as important as previously thought.

The fact that the internal connections are fixed in a reservoir network means that the selectivity of the reservoir neurons is also fixed. This may seem at odds with the experimental findings of many OFC neurons shifting their encodings rapidly during reversals [48]. However, these observations may be interpreted differently when we take into account rewards. The neurons that were found to have different responses during reversals might in fact encode a combination of sensory events and rewards. On the other hand, there is evidence that OFC neurons with inflexible encodings during reversals might be more important for flexible behavior [49].

The choice of a reservoir network as the center piece of task event encoding may appear questionable to some. We do not train the network to learn task event sequences. Instead, we use the dynamic patterns elicited by task event sequences as bases for learning. This approach has obvious weaknesses. One is that the chaotic nature of network dynamics limits how well the task states can be encoded in the network. We have illustrated the network works well for relatively simple tasks. However, when we consider tasks that have many stages or many events, the combination of possible states grows quickly and may exceed the capacity of the network. The fact that we do not train the internal network connections does not help in this regard. However, the purpose of our network model is not to solve very complicated tasks. Instead, we would like to argue this is a more biologically-realistic model than many other recurrent networks. First, it does not depend on supervised learning to learn task event sequences [47, 50]. Second, although the network performance may appear to be limited by task complexity, the real brain, however, also has limited capacity in learning multi-stage tasks [37]. Lastly, we show that a reservoir network can describe OFC neuronal responses during value-based decision making. Several other studies have also shown that reservoir networks may be a useful model of the prefrontal cortex [24, 25].

Reward input to the reservoir

One key observation is that reward events must also be provided as inputs to the reservoir layer for the network model to perform well. Including reward events allows the network to establish associations between sensory stimuli and rewards, thus facilitates task structure acquisition. Although reward modulates neural activities almost everywhere in the cortex, the OFC plays a central role in establishing the association between sensory stimuli and rewards [9, 48, 51, 52]. Anatomically, The OFC receives visual sensory inputs from inferior temporal and perirhinal cortex, as well as reward information from the brain areas in the reward circuitry, including the amygdala and ventral striatum, allowing it to have the information for establishing the association between visual information and reward [30–32]. Removing the reward input to the reservoir mimics the situation when animals cannot rely on such an association to learn tasks. In this case, the reservoir is still perfectly functional in terms of encoding task events other than rewards. This is similar to the situation when animals have to depend on their other memory structures in the brain–such as hippocampus or other medial temporal lobe structures–for learning. Consistent with this idea, it has been shown both the OFC and the ventral striatum are important for model-based RL [53]. The importance of the reward input to the reservoir explains the key role that the OFC plays in RL.

Several recent studies reported that selective lesions in the OFC did not reproduce the behavior deficits in reversal learning previously seen if the fibers passing through or near the OFC were spared [29]. Since these fibers probably carry the reward information from the midbrain areas, these results do not undermine the importance of reward inputs. Presumably, when the lesion is limited to the OFC, the projections that carrying the reward information are still available to or might even be redirected to other neighboring prefrontal structures, including ventromedial prefrontal cortex, which might take over the role of the OFC and contribute to the learning in animals with selective OFC lesions.

Model-based reinforcement learning

The acquisition of task structure is a prerequisite for model-based learning. Therefore, it is interesting to ask whether our network model is able to achieve model-based learning. The two-stage task that we model has been used in human literature to study model-based learning [5, 6, 33–36]. Our model, although exhibiting behavior similar to human subjects, can be categorized as the Reward-as-cue agent that was described and categorized as a form of model-free reinforcement learning agent by Akam et al. [37]. Yet, with reward incorporated as part of the task state space, goal-directed behavior can be achieved by searching in the state space for a task event sequence that ends with the desired goal and associating the sequence with appropriate actions. Thus, our network could in theory support model-based learning by providing the task structure to the downstream network layers.

Extending the network

The performance of our network depends on several factors. First, it is important that reservoir should be able to distinguish between different task states. The number of possible task states may be only 4 or 8 as in our examples, or may be impossibly large even if the number of inputs increases only modestly. The latter is due to the infamous combinatorial explosion problem. One may alleviate the problem by introducing learning in the reservoir to enhance the representation of relevant stimulus combinations and weed out irrelevant ones. A recent study showed that the selectivity pattern in the prefrontal neurons may be better explained by a random network with Hebbian learning [54]. Second, the dynamics of the reservoir should allow information to be maintained long enough in a decipherable form until the decision is made. The recent developed gated recurrent neural networks may provide a solution with units that may maintain information for long periods [55]. Third, the model exhibits substantial variability between runs, suggesting the initialization may impact its performance. Further investigation is needed to make the model more robust. Last, we show that a reinforcement learning algorithm is capable of solving the relatively simple tasks in this study. However, it has been shown that reinforcement learning is in general not very efficient for extracting information from reservoir networks. Especially, when the task demands the information to be held for an extended period, for example, across different trials, the current learning algorithm fails to extract such relevant information from the reservoir. A possible solution is to introduce additional layers to help with the readout [25].

Testable predictions

Our model makes several testable predictions. First, because of the reservoir structure, the inputs from the same source should be represented evenly in the network. For example, in a visual task, different visual stimuli should be represented at roughly the same strength in the

OFC, even if their task relevance may be drastically different. Second, we should be able to find neurons encoding all relevant task parameters in the network, even when a particular combination of task parameters is never experienced by the brain. Third, reducing the number of inputs may make the network to be more efficient in certain tasks. This may seem counterintuitive. But removing inputs reduces the number of states that the network has to encode, thus improves learning efficiency for tasks that do not require those additional states. For example, if we remove the reward input to the SEL, which is essential for learning tasks with volatile rewards, the network should however be more efficient at learning tasks in a more stable environment. Indeed, animals with OFC lesions were found to perform better than control animals when reward history was not important [56].

Summary

Our network does not intend to be a complete model of how the OFC works. Instead of creating a complete neural network solution of reinforcement learning or the OFC, which is improbable at the moment, we are aiming at the modest goal of providing a proof of concept that approaches the critical problem of how the brain acquires the task structure with a biologically realistic neural network model. By demonstrating the network's similarity to the experimental findings in the OFC, our study opens up new possibilities in future investigation.

Methods

Neural network model

The model is composed of three layers: an input layer (IL), a state encoding layer (SEL), and a decision-making output layer (DML) (Fig 1A).

The units in the input layer represent the identities of sensory stimuli and the obtained reward. The input neurons are sparsely connected to the SEL units. The connection weights $w_i^{(1)}$ are set to 0 at a probability of 1-p_{IR}. Nonzero weights are assigned independently from a Gaussian distribution with zero mean and a variance of ${g_{\mathrm{IR}}}^2$

In the SEL, there are $N = 500$ neurons. The neurons in the SEL are connected with a low probability $p = 0.1$ and the connections are randomly and independently set from a Gaussian distribution with zero mean and a variance of $g^2/(p^*N)$, where the gain g acts as the control parameter in the SEL. Connections in the SEL could be either positive or negative; a neuron may project both types of connections.

Each neuron in the SEL is described by an activation variable x_i for $i = 1, 2, \ldots, N$, which is initialized according to a normal distribution $N(0, {\sigma_{\mathrm{ini}}}^2)$ at the beginning of each trial. x_i is updated at each time step ($dt = 1\mathrm{ms}$) as follows:

$$\tau\frac{dx_i}{dt} = -x_i + g\sum\nolimits_{j=1}^{N} w_{ij}y_j + w_i^{(1)}I + \sigma_{noise}dW_i \qquad (1)$$

where τ represents the time constant, w_{ij} is the synaptic weight between neurons i and j, dW_i stands for the white noise, which is sampled from a uniform distribution [0, 1], and σ_{noise} is its variance. The firing rate y_i of neuron i is a function of the activation variable x_i relative to a minimal firing rate $y_{\min} = 0$ and the maximal rate $y_{\max} = 1$:

$$\mathrm{y} = \begin{cases} y_0 + y_0 tanh(x/y_0) & x \leq 0 \\ y_0 + (y_{max} - y_0) * tanh(x/(y_{max} - y_0)) & x > 0 \end{cases} \qquad (2)$$

Here $y_0 = 0.1$ is the baseline firing rate.

The SEL neurons project to the DML. The two competing neurons in the DML represent the two choices respectively. The total input of neuron k in the DML is

$$v_k = \sum\nolimits_i w_{ik}^{(2)} y_i \quad \text{for } k = 1, 2 \tag{3}$$

where $w_{ik}{}^{(2)}$ is the weight of the synapse between neuron i in the SEL circuit and neuron k in the DML. The synaptic weights between the SEL and DML are randomly initialized according to uniform distribution [0, 1], and normalized to keep the squared sum of synaptic weights projecting to the same DML unit equal to 1.

The synaptic weights between the SEL and DML are updated based on the choice and the reward outcome during the training phase. The decision is based on the activities of output neuron. The stochastic choice behavior of our model is described by a softmax function:

$$p_k = E[r_k] = \frac{e^{-\beta v_k}}{\sum_l e^{-\beta v_l}} \tag{4}$$

where $E[r_k]$ denotes the expected value of choice a_k, p_k represents the probability of choosing choice $a_{k,}$ and the other choice is chosen with probability 1- p_k. β adjusts the competition strength of two choices, and v_k is the summed input of the DML unit k. The firing rate of the unit k, z_k, is set to 1 if choice a_k is chosen, otherwise it is set to 0.

Reinforcement learning

At the end of each trial, the weights between the SEL and the DML neurons are updated based on the choice and the reward feedback.

The plastic weights in Eq (3) in trial n+1 are updated as follows:

$$w_{ik}^{(2)}(n+1) = w_{ik}^{(2)}(n) + \Delta w_{ik} \tag{5}$$

The update term Δw_{ik} depends on the reward prediction error and the responses of the neurons in the SEL circuit and DML:

$$\Delta w_{ik} = \eta(r - E[r])(y_i - y_{th})z_k \tag{6}$$

where η is the learning rate, and r is the reward. $E[r]$ denotes the expected value of the chosen option, which is equal to the probability of choosing choice a_k and calculated with Eq 4 [57, 58]. When the reward r is larger than $E[r]$, the connections between the SEL neurons whose firing rate is above the threshold y_{th} and the neurons in the DML would be strengthened, and the connections between the neurons whose firing rate is below y_{th} and the neurons in the DML would be weakened. After each update, the weights $w_{ik}^{(2)}(n)$ are normalized:

$$w_{ik}^{(2)}(n) = \frac{w_{ik}^{(2)}(n)}{\sqrt{\sum_{i=1}^{N}[w_{ik}^{(2)}(n)]^2}} \tag{7}$$

so that the vector length of $w_{ik}^{(2)}(n)$ remains constant. The normalization stops the weights from growing infinitely [59].

In the very first trial of a simulation run, the choice input is randomly selected, and the reward input is set according to the reward contingency in the block. The weights are not updated in the first trial. The choice output from the first trial and its associated reward outcome are then fed into the network as the 2nd trial's input, which are used to calculate the decision for the 2nd trial and update the weights as described above.

Behavior task

Reversal learning. The network has to choose between two options. One option leads to a reward, and the other does not. The stimulus-reward contingency is reversed every 100 trials. The criterion for learning is set to 28 correct trials in 30 successive trials for the initial learning and 24 correct trials in 30 successive trials for subsequent reversals.

The input layer units represent the identities of the two options and the reward. An option unit's response is set to 1 for if the corresponding option is chosen in the current trial, otherwise it is set to 0. The reward unit's response is set to 1 if the choice is rewarded in the current trial. The output of the network indicates its choice for the next trial. The input units representing choice options and reward are activated between 200 and 700ms after the trial onset. There is a delay period of 200 ms, at the end of which (900 ms after the trial onset) the neurons' activities at 900ms are used for decision making (Fig 1C).

The network parameters are set as follows. Time constant τ = 100ms, network gain g = 2, training threshold y_{th} = 0.2, temperature parameter β = 4, learning rate η = 0.001, noise gain σ_{noise} = 0.01, initial noise gain σ_{ini} = 0.01, input connection gain g_{IR} = 4, input connection probability p_{IR} = 0.2.

The selectivity of neurons in the SEL is determined at 900ms after the trial onset. A unit is defined as selective to a certain input or a combination of inputs if its responses are significantly higher under the condition when the input or all inputs of the combination are set to 1 than the other conditions (one-way ANOVA with multiple comparison and Bonferroni correction).

Two-stage Markov decision task. The network has to make a choice between options *A1* and *A2*. *A1* leads to intermediate outcome *B1* at the probability of 80%, and *B2* at the probability of 20%. Vice versa, option *A2* leads to *B2* at the probability of 80%, and *B1* at a lower probability of 20%. The contingency between options (*A1*, *A2*) and intermediate outcomes (*B1*, *B2*) is fixed. Initially, *B1* leads to a reward at the probability of 80% and *B2* leads to reward at the probability of 20%. The reward contingency is reversed every 50 trials.

The input layer contains 6 units, representing the identities of two first stage options *A1* and *A2*, two intermediate outcomes *B1* and *B2*, and the reward and non-reward conditions, respectively. The activity of option unit *A1* or *A2* is set to 1 when the respective option is chosen. The activity of intermediate outcome unit *B1* or *B2* is set to 1 when the respective intermediate outcome is presented. The reward unit's activity is set to 1 when a reward is obtained, and the non-reward unit's activity is set to 1 when no reward is obtained. The units are activated sequentially, reflecting the sequential nature of the task. The *A* units are activated between 200 and 700ms after the trial onset, the *B* units between 700 and 1200ms, and the reward units between 1200 and 1700ms. Decision is made based on the neurons' activity at 1900ms after the trial onset (Fig 3C).

The network parameters are set as follows. Time constant τ = 500ms, Network gain g = 2.25, training threshold y_{th} = 0.2, temperature parameter β = 2, learning rate η = 0.001, noise gain σ_{noise} = 0.01, initial noise gain σ_{ini} = 0.01, input connection gain g_{IR} = 2, input connection probability p_{IR} = 0.2.

The selectivity of neurons in the SEL is determined at the time point when the decision is made. There are 8 conditions in this task, namely *A1B1R*, *A1B1N*, *A2B1R*, *A2B1N*, *A1B2R*, *A1B2N*, *A2B2R*, and *A2B2N*. For example, *A1B1R* indicates the condition when *A1* is chosen, intermediate outcome *B1* is presented, and a reward is obtained. A neuron's preferred condition is the condition under which its activity is the largest and significantly higher than its activity under any other conditions (one-way ANOVA with multiple comparison and Bonferroni correction). Then the neurons are grouped into different categories based on their preferred conditions. The neurons in category *A1R* are the neurons whose preferred condition may be *A1B1R*, *A1B2R*, *A2B1N*, or *A2B2N*. All the preferred conditions of the neurons in category *A1R* provide evidence for associating *A1* with the reward. Similarly, the preferred conditions of the neurons in the category *B1N* are *A1B1N*, *A1B2R*, *A2B1N* and *A2B2R*. They provide evidence that *B1* is not associated with the reward.

In order to test how well the network uses the task structure information, we fit our data based on a simplified version of the model introduced by Daw et al. [5]. The model fits the behavioral results with a mixture of model-free (task-agnostic) and model-based (task-aware) learning algorithm. In our simplified task, the network makes only one choice in each trial. The network first undergoes 2,000 trials of training before the analysis.

For the model-free agent, the state values for B_1/B_2 and A_1/A_2 are updated as follows:

$$V_{MF}\ (s_{observed\_B}, t) = V_{MF}\ (s_{observed\_B}, t-1) + \alpha_1 * (r_t - V_{MF}\ (s_{observed\_B},\ t-1)) \tag{8}$$

$$V_{MF}\ (s_{chosen\_A}, t) = V_{MF}\ (s_{chosen\_A}, t-1) + \alpha_1 * \lambda * (V_{MF}(s_{observed\_B}, t) - V_{MF}\ (s_{observed\_B}, t-1)) \tag{9}$$

where V_{MF} ($s_{chosen\_A,\ t}$) and V_{MF} ($s_{observed\_B,\ t}$) are the value of states A_i and B_i that are observed in trial t, r_t represents the reward feedback in trial t, α_1 is the learning rate of the model-free learning algorithm, and the eligibility λ represents how large the proportion of credit from the reward can be given to states A_i and actions in our task paradigm. The state value for unobserved states is not changed.

For the model-based agent, the state values for B_1/B_2 and A_1/A_2 are updated as follows:

$$V_{MB}\ (s_{observed\_B}, t) = V_{MB}\ (s_{observed\_B}, t-1) + \alpha_2 * (r_t - V_{MB}\ (s_{observed\_B}, t-1)) \tag{10}$$

$$V_{MB}\ (s_{A1}, t) = P_{A1-B1} * V_{MB}\ (s_{B1}, t) + P_{A1-B2} * V_{MB}\ (s_{B2}, t) \tag{11}$$

$$V_{MB}\ (s_{A2}, t) = P_{A2-B1} * V_{MB}\ (s_{B1}, t) + P_{A2-B2} * V_{MB}\ (s_{B2}, t) \tag{12}$$

where α_2 is the learning rate of the model-based learning algorithm. $P_{Ai\text{-}Bi}$ indicates the probability of transition from state A_i to state B_i. The state value for unobserved states is not changed.

The net state value is defined as the weighted sum of the action values from the model-free agent and the model-based agent:

$$V_{net} = \mathrm{w} * V_{MB} + (1 - \mathrm{w}) * V_{MF} \tag{13}$$

where *w* is the weight. When *w* equals 1, the behavior uses full task information. When *w* equals 0, the behavior is completely task agnostic. The fitting is done by a maximum likelihood estimation procedure.

Finally, the probability of choosing option *i* is a softmax function of V_{net}.

$$p(a_t = A_i) = \frac{exp(\beta[V_{net}(s_{A_i,t})] + p * rep(A_i))}{\sum_{a'} exp(\beta[V_{net}(s_{a',t})] + p * rep(a'))} \tag{14}$$

where $rep(a)$ is set to 1 if action a is chosen in the previous trial. The inverse temperature parameter β is set to 2, which equals to the β term in e.q. (4) that generates the behavior. The parameter *p*, which captures the tendency for perseveration and switching, is set to 0. This is because we reset the network activity every trial. The conclusions hold when *p* is allowed to vary. Thus, there are only four free parameters, α_1, α_2, λ and *w*. Sessions with *w* deviating more than 3 standard deviations from the mean are excluded in Fig 4C for cosmetic reasons.

Including them increases the significance of weight difference between the two models without affecting the conclusion.

Inspired by the factorial analysis from Daw et al. [5], we define a task-structure (TS) index (Eq 15) to quantify how much task structure information is used in the network behavior. It is based on the tendency of repeating the choice in the last trial under different situations. The combination of the two reward outcomes and the two intermediate outcomes, common and rare, gives us four possible outcomes: common-rewarded (*CR*), common-unrewarded (*CN*), rare-rewarded (*RR*) and rare-unrewarded (*RN*). When the task structure is known, the agent is more likely to repeat the previous choice if the last trial is a *CR* or an *RN* trial. Higher TS index means that the behavioral pattern takes into account more task structure information.

$$\text{TS index} = \frac{p(stay|CR) + p(stay|RN) - p(stay|CN) - p(stay|RR)}{p(stay|CR) + p(stay|RN) + p(stay|CN) + p(stay|RR)} \tag{15}$$

The task state analysis is similar to what was used by Akam et al. [37]. Briefly, a logistic regression is used to estimate how states of past trials influence the current choice. The regression includes four different states (2 intermediate outcomes x 2 reward outcomes) for each trial up to 10 trials before the current trials.

Another logistic regression is used to estimate how several other potentially relevant factors affect choices. The factors considers include: Correct—a tendency to choose the better choice in current block; Reward—a tendency to repeat the previous choice if it is rewarded; Stay—a tendency to repeat the previous choice; Transition—a tendency to repeat the same choice following common intermediate outcomes and switch the choice following rare intermediate outcomes; Trans x Out–a tendency to repeat the same choice if a common intermediate outcome is rewarded or a rare intermediate outcome unrewarded, and to switch the choice if a common intermediate outcome is unrewarded or a rare intermediate outcome rewarded.

Value-based economic choice task. Unlike the two previous paradigms, both options in this paradigm lead to a reward. Two input units represent the rewards associated with the two options, respectively. The input strength is proportional to reward magnitude. In our simulations, the reward *A* is valued twice as much as reward *B* for the same reward magnitude. The relative value preference between the two options is not provided as an input to the network directly. It is only used for calculating the expected value. Thus, it does not affect the SEL. The value of the reward is defined as the product of the relative value and the reward magnitude. The reward options are presented between 300 and 1300 ms after the trial onset. After a 100 ms delay period, the network activity is used to calculate decisions (Fig 5B).

The activity of the input unit f(*t*) during the stimulus period (between 300ms and 1300ms after the trial onset), is described by the following equations [40].

$$\mathrm{g}(t) = 1/((1 + \exp(-(t - 475)/30)) * (1 + \exp((t - 700)/100))) \tag{16}$$

$$\mathrm{f}(t) = (\mathrm{mag}_{\mathrm{r}i} - \min(\mathrm{mag}_{\mathrm{r}i})) * \mathrm{g(t)}/(\max(\mathrm{mag}_{\mathrm{r}i}) - \min(\mathrm{mag}_{\mathrm{r}i})) * \max(\mathrm{g}(t)) \tag{17}$$

where *t* is the time in the unit of ms within a trial, $\mathrm{mag}_{\mathrm{r}i}$ is the magnitude of the reward type *i* in each trial, $\max(\mathrm{mag}_{\mathrm{r}i})$ is the maximal reward magnitude of reward type *i* within the block, and $\min(\mathrm{mag}_{\mathrm{r}i})$ represents the minimal reward magnitude of reward type *i*, which is always 0 in our simulations.

The network activity at 1400ms is used for decision making and network training. The expected value is the sum of the product of the probability of choosing the option and corresponding reward magnitude.

$$E(r) = p_1(\gamma * m_1) + p_2 m_2 \tag{18}$$

where p_i and m_i are the probability of choosing option i and its reward magnitude, and $\gamma = 2$ is the relative value preference between the two reward options. Only the data from the trials after 8000 trials training are included for the analyses. The network parameters are set as follows. Time constant $\tau = 100$ms, Network gain $g = 2.5$, training threshold $y_{th} = 0.2$, temperature parameter $\beta = 4$, learning rate $\eta = 0.005$, noise gain $\sigma_{\text{noise}} = 0.05$, initial noise gain $\sigma_{\text{ini}} = 0.2$, input connection gain $g_{\text{IR}} = 2$, input connection probability $p_{\text{IR}} = 0.2$.

As in Padoa-Schioppa and Assad [11], the following variables are defined for further analysis: total value (the sum of the value of two options), chosen value (the value of the chosen option), other value (the value of the unchosen option), value difference (chosen-other value), value ratio (other/chosen value), offer value (the value of the one option), chosen juice (the identity of the chosen option), value A chosen (the value of the option A when option A is chosen), and value B chosen (the value of the option B when option B is chosen).

We use an analysis similar to that in Padoa-Schioppa and Assad [11] to study the selectivity of SEL units during the post-offer period (0-500ms after the stimulus onset). Linear regressions are applied to each variable to fit the neural responses in this time window for each SEL unit separately.

$$y_i = a * var + b,$$

where *var* represents the variables previously mentioned. A variable is considered to explain the response of a neuron in the SEL if the slope of the fitting linear function, *a*, is significantly different from zero ($p<0.05$, one-way ANOVA with Bonferroni correction).

Supporting information

S1 Fig. PCA on the population activity of the sub-networks AR (grey traces) and BR (blue traces). The sub-networks AR and BR consist of neurons that are selective to AR and BR, respectively. The network states are plotted in the space spanned by the first 3 PCA components, which are from the same PC space as in Fig 2B and are calculated from all neurons in the network. Each trace represents a different stimulus condition.
(TIF)

S2 Fig. Connection weight evolution. A. The difference between the connection weights of the AN and BN neurons in the SEL layer to DML unit *A* and DML unit *B*. Positive values indicate an SEL neuron has a stronger connection to DML unit A than to DML unit B and supports choice A. The gray and white area indicates the blocks in which the option *A* and the option *B* leads to the reward, respectively. **B.** Top row: AN neurons' connections weight to DML unit A (left) and DML unit B (right). Bottom row: BN neurons' connections weight to DML unit A (left) and DML unit B (right).
(TIF)

S3 Fig. *E*[*r*] for choice A in AR and BN trials (top panel) and AN and BR trials (bottom row) in A blocks. Shade area indicates the standard s.e.m. *E*[*r*] at 900 ms (decision time) is used for updating the weights.
(TIF)

S4 Fig. Factorial analysis of choice behavior for the full network model (left panel) and the model without reward input (right panel). The full model exhibits stronger task-structure

effects.
(TIF)

S5 Fig. The model-based behavior in networks with different settings. A. tau = 100 ms, A1/A2 occurs first at 200ms after the trial onset, and B1/B2 occurs at 700ms after the trial onset. **B.** tau = 500 ms, B1/B2 occurs first at 200ms after the trial onset, and A1/A2 occurs at 700ms after the trial onset. **C.** tau = 100 ms, B1/B2 occurs first at 200ms after the trial onset, and A1/A2 occurs at 700ms after the trial onset. Left column: the model structure index. Right column: weights for the model-based behavior. All the significance is evaluated by one-way ANOVA. See Fig 4B and 4C for details.
(TIF)

S6 Fig. The proportions of the neurons in the reservoir network with different selectivities are stable across network models with different network parameters stable. A. input connection gain = 0.5. **B.** noise gain = 0.5. **C.** A step function is used to model the reward inputs. The step function's onset is 300 ms and its offset is 1300 ms after the trial onset. All results are based on 10 simulation runs.
(TIF)

Acknowledgments

We thank Yu Shan and Xiao-Jing Wang for discussions and comments during the study.

Author Contributions

Conceptualization: Tianming Yang.

Data curation: Tianming Yang.

Formal analysis: Zhewei Zhang, Zhenbo Cheng, Tianming Yang.

Funding acquisition: Tianming Yang.

Investigation: Zhewei Zhang, Zhenbo Cheng, Zhongqiao Lin, Chechang Nie, Tianming Yang.

Methodology: Tianming Yang.

Project administration: Tianming Yang.

Resources: Tianming Yang.

Software: Tianming Yang.

Supervision: Tianming Yang.

Validation: Tianming Yang.

Visualization: Zhewei Zhang, Tianming Yang.

Writing – original draft: Zhewei Zhang, Tianming Yang.

Writing – review & editing: Zhewei Zhang, Zhenbo Cheng, Zhongqiao Lin, Chechang Nie, Tianming Yang.

References

1. Rescorla RA, Wagner AR. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In: Black AH, Prokasy WF, editors. Classical conditioning II: Current research and theory. New York: Appleton-Century-Crofts; 1972. p. 64–99.

2. Haber SN, Kim KS, Mailly P, Calzavara R. Reward-related cortical inputs define a large striatal region in primates that interface with associative cortical connections, providing a substrate for incentive-based learning. J Neurosci. 2006; 26(32):8368–76. https://doi.org/10.1523/JNEUROSCI.0271-06.2006 PMID: 16899732.
3. Schultz W, Dayan P, Montague PR. A neural substrate of prediction and reward. Science. 1997; 275 (5306):1593–9. PMID: 9054347.
4. Kennerley SW, Behrens TE, Wallis JD. Double dissociation of value computations in orbitofrontal and anterior cingulate neurons. Nat Neurosci. 2011; 14(12):1581–9. https://doi.org/10.1038/nn.2961 PMID: 22037498; PubMed Central PMCID: PMCPMC3225689.
5. Daw ND, Gershman SJ, Seymour B, Dayan P, Dolan RJ. Model-Based Influences on Humans' Choices and Striatal Prediction Errors. Neuron. 2011; 69(6):1204–15. https://doi.org/10.1016/j.neuron.2011.02.027 WOS:000288886900015. PMID: 21435563
6. Glascher J, Daw N, Dayan P, O'Doherty JP. States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron. 2010; 66(4):585–95. https://doi.org/10.1016/j.neuron.2010.04.016 PMID: 20510862; PubMed Central PMCID: PMCPMC2895323.
7. Wilson RC, Takahashi YK, Schoenbaum G, Niv Y. Orbitofrontal cortex as a cognitive map of task space. Neuron. 2014; 81(2):267–79. https://doi.org/10.1016/j.neuron.2013.11.005 PMID: 24462094; PubMed Central PMCID: PMC4001869.
8. Hornak J, O'Doherty J, Bramham J, Rolls ET, Morris RG, Bullock PR, et al. Reward-related reversal learning after surgical excisions in orbito-frontal or dorsolateral prefrontal cortex in humans. J Cogn Neurosci. 2004; 16(3):463–78. https://doi.org/10.1162/089892904322926791 PMID: 15072681.
9. Izquierdo A, Suda RK, Murray EA. Bilateral orbital prefrontal cortex lesions in rhesus monkeys disrupt choices guided by both reward value and reward contingency. J Neurosci. 2004; 24(34):7540–8. https://doi.org/10.1523/JNEUROSCI.1921-04.2004 PMID: 15329401.
10. Takahashi YK, Roesch MR, Wilson RC, Toreson K, O'Donnell P, Niv Y, et al. Expectancy-related changes in firing of dopamine neurons depend on orbitofrontal cortex. Nat Neurosci. 2011; 14 (12):1590–7. https://doi.org/10.1038/nn.2957 PMID: 22037501; PubMed Central PMCID: PMCPMC3225718.
11. Padoa-Schioppa C, Assad JA. Neurons in the orbitofrontal cortex encode economic value. Nature. 2006; 441(7090):223–6. https://doi.org/10.1038/nature04676 PMID: 16633341; PubMed Central PMCID: PMC2630027.
12. Padoa-Schioppa C. Neurobiology of economic choice: a good-based model. Annu Rev Neurosci. 2011; 34:333–59. https://doi.org/10.1146/annurev-neuro-061010-113648 PMID: 21456961; PubMed Central PMCID: PMCPMC3273993.
13. Wallis JD, Miller EK. Neuronal activity in primate dorsolateral and orbital prefrontal cortex during performance of a reward preference task. The European journal of neuroscience. 2003; 18(7):2069–81. PMID: 14622240.
14. Jones JL, Esber GR, McDannald MA, Gruber AJ, Hernandez A, Mirenzi A, et al. Orbitofrontal cortex supports behavior and learning using inferred but not cached values. Science. 2012; 338(6109):953–6. https://doi.org/10.1126/science.1227489 PMID: 23162000; PubMed Central PMCID: PMC3592380.
15. Rudebeck PH, Mitz AR, Chacko RV, Murray EA. Effects of amygdala lesions on reward-value coding in orbital and medial prefrontal cortex. Neuron. 2013; 80(6):1519–31. https://doi.org/10.1016/j.neuron.2013.09.036 PMID: 24360550; PubMed Central PMCID: PMC3872005.
16. Kennerley SW, Wallis JD. Evaluating choices by single neurons in the frontal lobe: outcome value encoded across multiple decision variables. Eur J Neurosci. 2009; 29(10):2061–73. https://doi.org/10.1111/j.1460-9568.2009.06743.x PMID: 19453638; PubMed Central PMCID: PMC2715849.
17. O'Neill M, Schultz W. Economic risk coding by single neurons in the orbitofrontal cortex. J Physiol Paris. 2015; 109(1–3):70–7. https://doi.org/10.1016/j.jphysparis.2014.06.002 PMID: 24954027; PubMed Central PMCID: PMC4451954.
18. Blanchard TC, Hayden BY, Bromberg-Martin ES. Orbitofrontal cortex uses distinct codes for different choice attributes in decisions motivated by curiosity. Neuron. 2015; 85(3):602–14. https://doi.org/10.1016/j.neuron.2014.12.050 PMID: 25619657; PubMed Central PMCID: PMC4320007.
19. Wallis JD, Anderson KC, Miller EK. Single neurons in prefrontal cortex encode abstract rules. Nature. 2001; 411(6840):953–6. https://doi.org/10.1038/35082081 PMID: 11418860.
20. Tsujimoto S, Genovesio A, Wise SP. Comparison of strategy signals in the dorsolateral and orbital prefrontal cortex. J Neurosci. 2011; 31(12):4583–92. https://doi.org/10.1523/JNEUROSCI.5816-10.2011 PMID: 21430158; PubMed Central PMCID: PMC3082703.

21. Buonomano DV, Maass W. State-dependent computations: spatiotemporal processing in cortical networks. Nat Rev Neurosci. 2009; 10(2):113–25. https://doi.org/10.1038/nrn2558 PMID: 19145235.

22. Laje R, Buonomano DV. Robust timing and motor patterns by taming chaos in recurrent neural networks. Nat Neurosci. 2013; 16(7):925–33. https://doi.org/10.1038/nn.3405 PMID: 23708144; PubMed Central PMCID: PMCPMC3753043.

23. Maass W, Natschlager T, Markram H. Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput. 2002; 14(11):2531–60. https://doi.org/10.1162/089976602760407955 PMID: 12433288.

24. Barak O, Sussillo D, Romo R, Tsodyks M, Abbott LF. From fixed points to chaos: three models of delayed discrimination. Prog Neurobiol. 2013; 103:214–22. https://doi.org/10.1016/j.pneurobio.2013.02.002 PMID: 23438479; PubMed Central PMCID: PMC3622800.

25. 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. 2015; 114(6):3296–305. https://doi.org/10.1152/jn.00378.2015 PMID: 26445865.

26. Enel P, Procyk E, Quilodran R, Dominey PF. Reservoir Computing Properties of Neural Dynamics in Prefrontal Cortex. PLoS Comput Biol. 2016; 12(6):e1004967. https://doi.org/10.1371/journal.pcbi.1004967 PMID: 27286251; PubMed Central PMCID: PMCPMC4902312.

27. Szita I, Gyenes V, Lőrincz A, editors. Reinforcement Learning with Echo State Networks. Artificial Neural Networks–ICANN 2006; 2006 2006.

28. Jones B, Mishkin M. Limbic lesions and the problem of stimulus—reinforcement associations. Exp Neurol. 1972; 36(2):362–77. PMID: 4626489.

29. Rudebeck PH, Saunders RC, Prescott AT, Chau LS, Murray EA. Prefrontal mechanisms of behavioral flexibility, emotion regulation and value updating. Nat Neurosci. 2013; 16(8):1140–5. https://doi.org/10.1038/nn.3440 PMID: 23792944; PubMed Central PMCID: PMC3733248.

30. Carmichael ST, Price JL. Sensory and premotor connections of the orbital and medial prefrontal cortex of macaque monkeys. The Journal of comparative neurology. 1995; 363(4):642–64. https://doi.org/10.1002/cne.903630409 PMID: 8847422.

31. Carmichael ST, Price JL. Limbic connections of the orbital and medial prefrontal cortex in macaque monkeys. The Journal of comparative neurology. 1995; 363(4):615–41. https://doi.org/10.1002/cne.903630408 PMID: 8847421.

32. Eblen F, Graybiel AM. Highly restricted origin of prefrontal cortical inputs to striosomes in the macaque monkey. J Neurosci. 1995; 15(9):5999–6013. PMID: 7666184.

33. Wunderlich K, Dayan P, Dolan RJ. Mapping value based planning and extensively trained choice in the human brain. Nat Neurosci. 2012; 15(5):786–91. https://doi.org/10.1038/nn.3068 PMID: 22406551; PubMed Central PMCID: PMCPMC3378641.

34. Wunderlich K, Smittenaar P, Dolan RJ. Dopamine enhances model-based over model-free choice behavior. Neuron. 2012; 75(3):418–24. https://doi.org/10.1016/j.neuron.2012.03.042 PMID: 22884326; PubMed Central PMCID: PMCPMC3417237.

35. Smittenaar P, FitzGerald TH, Romei V, Wright ND, Dolan RJ. Disruption of dorsolateral prefrontal cortex decreases model-based in favor of model-free control in humans. Neuron. 2013; 80(4):914–9. https://doi.org/10.1016/j.neuron.2013.08.009 PMID: 24206669; PubMed Central PMCID: PMCPMC3893454.

36. Dezfouli A, Balleine BW. Actions, action sequences and habits: evidence that goal-directed and habitual action control are hierarchically organized. PLoS Comput Biol. 2013; 9(12):e1003364. https://doi.org/10.1371/journal.pcbi.1003364 PMID: 24339762; PubMed Central PMCID: PMCPMC3854489.

37. Akam T, Costa R, Dayan P. Simple Plans or Sophisticated Habits? State, Transition and Learning Interactions in the Two-Step Task. PLoS Comput Biol. 2015; 11(12):e1004648. https://doi.org/10.1371/journal.pcbi.1004648 PMID: 26657806; PubMed Central PMCID: PMCPMC4686094.

38. Padoa-Schioppa C. Neuronal origins of choice variability in economic decisions. Neuron. 2013; 80(5):1322–36. https://doi.org/10.1016/j.neuron.2013.09.013 PMID: 24314733; PubMed Central PMCID: PMCPMC3857585.

39. Cai X, Padoa-Schioppa C. Contributions of orbitofrontal and lateral prefrontal cortices to economic choice and the good-to-action transformation. Neuron. 2014; 81(5):1140–51. https://doi.org/10.1016/j.neuron.2014.01.008 PMID: 24529981; PubMed Central PMCID: PMCPMC3951647.

40. Rustichini A, Padoa-Schioppa C. A neuro-computational model of economic decisions. J Neurophysiol. 2015; 114(3):1382–98. https://doi.org/10.1152/jn.00184.2015 PMID: 26063776; PubMed Central PMCID: PMCPMC4556855.

41. Daie K, Goldman MS, Aksay ER. Spatial patterns of persistent neural activity vary with the behavioral context of short-term memory. Neuron. 2015; 85(4):847–60. https://doi.org/10.1016/j.neuron.2015.01.006 PMID: 25661184; PubMed Central PMCID: PMCPMC4336549.

42. Suykens JAK, Vandewalle J, Moor BLRd. Artificial neural networks for modelling and control of non-linear systems. Boston: Kluwer Academic Publishers; 1996. xii, 235 pages p.

43. Rodriguez P. Simple recurrent networks learn context-free and context-sensitive languages by counting. Neural Comput. 2001; 13(9):2093–118. https://doi.org/10.1162/089976601750399326 PMID: 11516359.

44. Barak O, Rigotti M, Fusi S. The sparseness of mixed selectivity neurons controls the generalization-discrimination trade-off. J Neurosci. 2013; 33(9):3844–56. https://doi.org/10.1523/JNEUROSCI.2753-12.2013 PMID: 23447596.

45. Rigotti M, Barak O, Warden MR, Wang XJ, Daw ND, Miller EK, et al. The importance of mixed selectivity in complex cognitive tasks. Nature. 2013; 497(7451):585–90. https://doi.org/10.1038/nature12160 PMID: 23685452; PubMed Central PMCID: PMC4412347.

46. Rigotti M, Ben Dayan Rubin D, Wang XJ, Fusi S. Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. Front Comput Neurosci. 2010; 4:24. https://doi.org/10.3389/fncom.2010.00024 PMID: 21048899; PubMed Central PMCID: PMCPMC2967380.

47. Sussillo D, Abbott LF. Generating coherent patterns of activity from chaotic neural networks. Neuron. 2009; 63(4):544–57. https://doi.org/10.1016/j.neuron.2009.07.018 PMID: 19709635

48. Rolls ET, Critchley HD, Mason R, Wakeman EA. Orbitofrontal cortex neurons: role in olfactory and visual association learning. J Neurophysiol. 1996; 75(5):1970–81. https://doi.org/10.1152/jn.1996.75.5.1970 PMID: 8734596.

49. Schoenbaum G, Saddoris MP, Stalnaker TA. Reconciling the roles of orbitofrontal cortex in reversal learning and the encoding of outcome expectancies. Ann N Y Acad Sci. 2007; 1121:320–35. https://doi.org/10.1196/annals.1401.001 PMID: 17698988; PubMed Central PMCID: PMCPMC2430624.

50. Song HF, Yang GR, Wang XJ. Reward-based training of recurrent neural networks for cognitive and value-based tasks. Elife. 2017; 6. https://doi.org/10.7554/eLife.21492 PMID: 28084991; PubMed Central PMCID: PMCPMC5293493.

51. Thorpe SJ, Rolls ET, Maddison S. The orbitofrontal cortex: neuronal activity in the behaving monkey. Exp Brain Res. 1983; 49(1):93–115. PMID: 6861938.

52. Walton ME, Behrens TE, Buckley MJ, Rudebeck PH, Rushworth MF. Separable learning systems in the macaque brain and the role of orbitofrontal cortex in contingent learning. Neuron. 2010; 65(6):927–39. https://doi.org/10.1016/j.neuron.2010.02.027 PMID: 20346766; PubMed Central PMCID: PMC3566584.

53. McDannald MA, Lucantonio F, Burke KA, Niv Y, Schoenbaum G. Ventral striatum and orbitofrontal cortex are both required for model-based, but not model-free, reinforcement learning. J Neurosci. 2011; 31 (7):2700–5. https://doi.org/10.1523/JNEUROSCI.5499-10.2011 PMID: 21325538; PubMed Central PMCID: PMCPMC3079289.

54. Lindsay GW, Rigotti M, Warden MR, Miller EK, Fusi S. Hebbian Learning in a Random Network Captures Selectivity Properties of Prefrontal Cortex. J Neurosci. 2017. https://doi.org/10.1523/JNEUROSCI.1222-17.2017 PMID: 28986463.

55. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. ArXiv e-prints [Internet]. 2014 December 1, 2014; 1412. Available from: http://adsabs.harvard.edu/abs/2014arXiv1412.3555C.

56. Riceberg JS, Shapiro ML. Reward stability determines the contribution of orbitofrontal cortex to adaptive behavior. J Neurosci. 2012; 32(46):16402–9. https://doi.org/10.1523/JNEUROSCI.0776-12.2012 PMID: 23152622; PubMed Central PMCID: PMCPMC3568518.

57. Law CT, Gold JI. Reinforcement learning can account for associative and perceptual learning on a visual-decision task. Nature neuroscience. 2009; 12(5):655–63. https://doi.org/10.1038/nn.2304 PMID: 19377473

58. Seung HS. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. Neuron. 2003; 40(6):1063–73. PMID: 14687542.

59. Royer S, Pare D. Conservation of total synaptic weight through balanced synaptic depression and potentiation. Nature. 2003; 422(6931):518–22. https://doi.org/10.1038/nature01530 PMID: 12673250