Title:

Online reinforcement learning of state representation in recurrent network: the power of random feedback and biological constraints

Authors:

Takayuki Tsurumi[1,2], Ayaka Kato[2,3], Arvind Kumar[2,4], & Kenji Morita[1,2,5*]

Affiliations:

[1] Physical and Health Education, Graduate School of Education, The University of Tokyo, Tokyo, Japan

[2] Theoretical Sciences Visiting Program, Okinawa Institute of Science and Technology, Tancha, Okinawa, Japan

[3] Department of Psychiatry, Icahn School of Medicine at Mount Sinai, New York, NY, USA.

[4] Division of Computational Science and Technology, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden

[5] International Research Center for Neurointelligence (WPI-IRCN), The University of Tokyo, Tokyo, Japan

[*]*Corresponding author:*

Kenji Morita, Ph.D. (ORCID iD: orcid.org/0000-0003-2192-4248)

Physical and Health Education, Graduate School of Education, The University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

morita@p.u-tokyo.ac.jp

Author contributions

Conceptualization: KM; Formal analysis: KM, TT; Investigation: KM, TT, AyK; Writing – original draft: KM; Writing – review & editing: KM, TT, AyK, ArK

Acknowledgements

The authors thank Dr. Kenji Doya for valuable suggestions. KM was supported by Grants-in-Aid for Scientific Research 23H03295, 23K27985, and 25H02594 from Japan Society for the Promotion of Science (JSPS) and the Naito Foundation. AyK was supported by JSPS Overseas Research Fellowships. ArK was partially funded by Digital Futures (KTH) grant and StratNeuro SRA.

Abstract

Representation of external and internal states in the brain plays a critical role in enabling suitable behavior. Recent studies suggest that state representation and state value can be simultaneously learnt through Temporal-Difference-Reinforcement-Learning (TDRL) and Backpropagation-Through-Time (BPTT) in recurrent neural networks (RNNs) and their readout. However, neural implementation of such learning remains unclear as BPTT requires offline update using transported downstream weights, which is suggested to be biologically implausible. We demonstrate that simple online training of RNNs using TD reward prediction error and random feedback, without additional memory or eligibility trace, can still learn the structure of tasks with cue-reward delay and timing variability. This is because TD learning itself is a solution for temporal credit assignment, and feedback alignment, a mechanism originally proposed for supervised learning, enables gradient approximation without weight transport. Furthermore, we show that biologically constraining downstream weights and random feedback to be non-negative not only preserves learning but may even enhance it because the non-negative constraint ensures loose alignment - allowing the downstream and feedback weights to roughly align from the beginning. These results provide insights into the neural mechanisms underlying the learning of state representation and value, highlighting the potential of random feedback and biological constraints.

Introduction

Multiple lines of studies have suggested that Temporal-Difference-Reinforcement-Leaning (TDRL) is implemented in the cortico-basal ganglia-dopamine (DA) circuits where DA encodes TD reward-prediction-error (RPE) [1-6] and DA-dependent plasticity of cortico-striatal synapses corresponds to TD-RPE-dependent update of state/action values [7-9]. Traditionally, TDRL in the cortico-basal ganglia-DA circuits was considered to serve only for relatively simple behavior. However, subsequent studies suggested that more sophisticated, apparently goal-directed/model-based behavior can also be achieved by TDRL if states are appropriately represented [10-12] and that DA signals indeed reflect model-based predictions [13, 14]. Conversely, impairments in state representation may relate to behavioral or mental-health problems [15-19]. Early modeling studies treated state representations appropriate to the situation/task as given ('handcrafted' by the authors), but representation itself should be learnt in the brain [20-25]. Recently it was shown that appropriate state representation can be learnt through TDRL in a recurrent neural network (RNN) by minimizing squared TD value-error without explicit target [12, 26], while state value can be simultaneously learnt in connections downstream of the RNN.

However, whether such a learning -referred to as the value-RNN [26], can be implemented in the brain remains unclear. This is because the value-RNN [26] used the Backpropagation-Through-Time (BPTT) [27], which applies gradient-descent error-'backpropagation' (hereafter referred to as backprop) [28, 29] to a temporally unfolded RNNs. BPTT has been argued to be biologically implausible mainly due to problems with feedback and causality. Regarding feedback, updating upstream connections requires feedback whose weights are transported from downstream forward connections, but such weight transportation is difficult to implement biologically [30, 31]. In the case of the value-RNN, if the state-representing RNN and the value-encoding readout are implemented by the intra-cortical circuit and the striatal neurons, respectively, as generally assumed [1, 32, 33], this weight transportation means that the update (plasticity) rule for intra-cortical connections involves the downstream cortico-striatal synaptic strengths, which are not accessible from the cortex. Regarding the problem of causality in BPTT, the error needs to be incrementally accumulated in the temporally backward order, but such an acausal offline update is biologically implausible [34, 35].

Recently, a potential solution for the feedback problem of backprop has been proposed [36] (see also [37-44] for other approaches). Specifically, in supervised learning of feed-forward networks, it was shown that when the transported downstream weights used for updating upstream connections were replaced with fixed random strengths, comparable learning performance was still achieved [36]. This was suggested to be because the information of the random strengths transferred to the upstream connections and then to the downstream feed-forward connections so that these feed-forward connections became aligned to the random feedback and thereby the random feedback could work

similarly to the feedback with transported downstream weights in backprop. This mechanism was named the 'feedback alignment' (FA)[36], and was subsequently shown to work also in online supervised learning of RNN [34] and proposed to be ‘neurally’ implemented [45] (in a different way from the present study as we discuss in the Discussion).

The value-RNN [12, 26] differs from supervised learning considered in these previous feedback-alignment studies in two ways: i) it is TD learning, i.e., it approximates the true error by the TD-RPE because the true error, or true state value, is unknown, and ii) it uses a scalar error (TD-RPE) rather than a vector error. Scalar reward-based online learning of RNN with random feedback was actually shown to work in a different study [35] (their Supplementary Figure 5), but TD-RPE was not introduced in that setup, while the same study also examined another setup with TD-RPE but result with random feedback was not shown for this latter setup (their Figure 4 and 5). Therefore, it was nontrivial whether the value-RNN could be modified to incorporate online update using random feedback. Here we demonstrate that such a modified value-RNN could still work and provide a mechanistic insight into how it works.

Next we address other biological-plausibility issues with FA-based rule. Specifically, we imposed biological constraints that the downstream (cortico-striatal) weights and the fixed random feedback, as well as the activities of neurons in the RNN, were all non-negative. Moreover, we also remedied the non-monotonic dependence of the update of RNN connection-strength on post-synaptic neural activity. We found that the non-negative constraint appeared to aid, rather than degrade, the learning by ensuring that the downstream weights and the fixed random feedback were loosely aligned from the beginning. These results suggest how learning of state representation and value could be implemented via DA-dependent synaptic plasticity in cortical and striatal circuits, where DA encodes TD-RPE.

Results

1. Online value-RNN with fixed random feedback

We considered an online value-RNN in the cortico-basal ganglia circuits (Fig. 1). In this model a cortical region/population represents information of sensory observation ($\boldsymbol{o}$) and send it to another cortical region/population which estimates a state ($\boldsymbol{x}$) given the sensory inputs. We approximate this cortical population an RNN (number of RNN units was varied between 5 and 40). Neurons in the RNN learn to represent states by updating the strengths of recurrent connections **A** and feed-forward connections **B**. The activity of a population of striatal neurons that receive inputs from the RNN is supposed to learn to represent the state values (v), by learning the weights ($\boldsymbol{w}$) of cortico-striatal connections. DA neurons in the ventral tegmental area (VTA) receive information about the value and reward (r) from the striatum (both direct and indirect pathways) and other structures, and the activity of the DA neurons, as well as released DA, represents TD-RPE (δ). The TD-RPE-representing DA is released in the striatum and also in the cortical RNN through mesocorticolimbic projections, and used for modifying **A**, **B** and $\boldsymbol{w}$ (Fig. 1).

The original value-RNN [12, 26] adopted BPTT [27] as an update rule for the connections onto the RNN (**A** and **B**), which requires the (gradually changing) value weights ($\boldsymbol{w}$), but this is biologically implausible because the corticostriatal synaptic weights are not available in the cortex as mentioned above. Therefore, we examined a model (agent) in which these weights were replaced with fixed random strengths ($\boldsymbol{c}$), in comparison with a model that used these weights. Because the temporally acausal offline update used in BPTT is also biologically implausible, we used an online learning rule, which considers only the influence of the recurrent weights at the previous time step (see the Methods for details and equations). We refer to these models (agents) as the online value-RNN or oVRNN.

We assumed that a single RNN unit corresponds to a small population of neurons that intrinsically share inputs and outputs, for genetic or developmental reasons, and the activity of each unit represents the (relative) firing rate of the population. Cortical population activity is suggested to be sustained not only by fast synaptic transmission and spiking but also, even predominantly, by slower synaptic neurochemical dynamics [46] such as short-term facilitation, whose time constant can be around 500 milliseconds [47]. Therefore, we assumed that single time-step of our rate-based (rather than spike-based) model corresponds to 500 milliseconds.

In each simulation, the recurrent (**A**) and feed-forward connection (**B**) weights onto the RNN units were initialized to pseudo standard normal random numbers. As a negative control, we also conducted simulations in which these connections were not updated from initial values, referring to as the case with "untrained (fixed) RNN". Notably, the value weights $\boldsymbol{w}$ (i.e., connection weights

from the RNN to the striatal value unit) were still trained in the models with untrained RNN. The oVRNN models, and the model with untrained RNN, were continuously trained across trials in each task, because we considered that it was ecologically more plausible than episodic training of separate trials.

2.1 Simulation of a Pavlovian cue-reward association task with variable inter-trial intervals

First we took a small RNN with 7 units to represent state in the cortex, and simulated a Pavlovian cue-reward association task, in which a cue was followed by a reward three time-steps later, and inter-trial interval (i.e., reward to next cue) was randomly chosen from 4, 5, 6, or 7 time-steps (Fig. 2A). Given that single time-step corresponds to 500 milliseconds as mentioned above, three time-steps from cue to reward correspond to 1.5 sec, which matches the delay in the conditioning task used by Schultz et al. [2]. In this task, states after receiving of cue information can be defined by time-steps from the cue, and the state values of these states can be estimated by calculating the expected cumulative discounted future rewards [48] through simulations; we refer to them as "estimated true state values" (Fig. 2B, black line). Expected TD-RPE can be calculated from these estimated true values (Fig. 2B, red line).

First, for comparison, we examined traditional TD-RL agent with punctate state representation (without using the RNN), in which each state (time-step from a cue) was represented in a punctate manner, i.e., by a one-hot vector such as (1, 0, ..., 0), (0, 1, ..., 0), and so on. We examined two cases: one in which training was done in an episodic manner without continuation between trials (i.e., the value of the last state in a trial was not updated by TD-RPE upon entering the next trial) and the other in which training was done continuously across trials, as in the cases of agents using the RNN. The former agent developed positive values between cue and reward, and abrupt TD-RPE upon cue (Fig. 2C), whereas the latter agent developed positive values also for states in the inter-trial interval (Fig. 2D), looking similar to the true values (Fig. 2B).

We then examined our oVRNN agents, with backprop-type transported downstream weights (oVRNNbp: Fig. 2E) or with fixed random feedback (oVRNNrf: Fig. 2F), in comparison with the agent with untrained fixed RNN (Fig. 2G). As shown in the figures, oVRNNbp successfully learned the values of states between cue and reward, and oVRNNrf also learned these values, although to a somewhat smaller degree on average. On the other hand, the agent with untrained RNN developed the smallest state values on average among the three agents. This inferiority of untrained RNN may sound odd because there were only four states from cue to reward while random RNN with enough units is expected to be able to represent many different states (c.f., [49]) and the effectiveness of training of only the readout weights has been shown in reservoir computing studies [50-53]. However, there was a difficulty stemming from the continuous training across trials (rather than episodic training of separate trials): the activity of untrained RNN upon cue presentation generally differed

from trial to trial, and so it is non-trivial that cue presentation in different trials should be regarded as the same single state, even if it could eventually be dealt with at the readout level if the number of units increases.

The results above indicate that value-RNN could be trained online by fixed random feedback at least to a certain extent, although somewhat less effectively than by backprop-type feedback. Results of individual simulations shown in Fig. 2H,I indicate that state values developed in oVRNNrf were largely comparable to those developed in oVRNNbp once they were successfully learned but the success rate was smaller than oVRNNbp while still larger than the untrained RNN.

2.2 Systematic simulations and analyses

Next, we tested whether learning performance of oVRNNbp, oVRNNrf, and the agent with untrained fixed RNN depends on the number of RNN units (n). For a valid comparison with the previously shown cases with 7 RNN units, the learning rate for the value weights was normalized by dividing by $n/7$. Learning performance was measured by the mean of squares of differences between the state values developed by each of these three types of agents and the estimated true state values (Fig. 2B) between cue and reward at 1000-th trial. As shown in the left panel of Fig. 2J, on average across simulations, oVRNNbp and oVRNNrf exhibited largely comparable performance and always outperformed the untrained RNN ($p < 0.00022$ in Wilcoxon rank sum test for oVRNNbp or oVRNNrf vs untrained for each number of RNN units), although oVRNNbp somewhat outperformed or underperformed oVRNNrf when the number of RNN units was small (≤10 ($p < 0.049$)) or large (≥25 ($p < 0.045$)), respectively. As the number of RNN units increased from 5 to 15 or 20, all these three agents improved their performance. Additional increase of RNN units did not largely change the mean performance in oVRNNrf, while moderately decreased it in oVRNNbp and untrained RNN. The green line in Fig. 2J-right shows the performance of a special case where the random feedback in oVRNNrf was fixed to the direction of $(1, 1, ..., 1)^T$ (i.e., uniform feedback) with a random coefficient, which was largely comparable to, but somewhat worse than, that for the general oVRNNrf (blue line).

In order to examine the dimensionality of RNN dynamics, we conducted principal component analysis (PCA) of the time series (for 1000 trials) of RNN activities and calculated the contribution ratios of PCs in the cases of oVRNNbp, oVRNNrf, and untrained RNN with 20 RNN units. Figure 2K shows a log of contribution ratios of 20 PCs in each case. Compared with the case of untrained RNN, in oVRNNbp and oVRNNrf, initial component(s) had smaller contributions (PC1 (t-test p = 0.00018 in oVRNNbp; p = 0.0058 in oVRNNrf) and PC2 (p = 0.080 in oVRNNbp; p = 0.0026 in oVRNNrf)) while later components had larger contributions (PC3~10,15~20 $p < 0.041$ in oVRNNbp; PC5~20 $p < 0.0017$ in oVRNNrf) on average, and this is considered to underlie their superior learning performance. We noticed that late components had larger contributions in oVRNNrf than in

oVRNNbp, although these two models with 20 RNN units were comparable in terms of cue~reward state values (Fig. 2J-left).

We examined how learning proceeded across trials in the models with 20 RNN units. As shown in Fig. 2L, learning became largely converged by 1000-th trial, although slight improvement continued afterward. We further examined the cases with longer cue-reward delays. As shown in Fig. 2M, as the delay increased, the mean squared error of state values (at 3000-th trial) increased, but the relative superiority of oVRNNbp and oVRNNrf over the model with untrained RNN remained to hold, except for cases with small number of RNN units (5) and long delay (5 or 6) ($p < 0.0025$ in Wilcoxon rank sum test for oVRNNbp or oVRNNrf vs untrained for each number of RNN units for each delay).

2.3 Occurrence of feedback alignment and an intuitive understanding of its mechanism

Next, we questioned how feedback alignment contributes to the learnability of oVRNNrf. To address this question we used an RNN with 7 units and examined whether the value weight vector $\boldsymbol{w}$ became aligned to the random feedback vector $\boldsymbol{c}$ in oVRNNrf, by looking at the changes in the angle between these two vectors across trials. As shown in Fig. 3A, this angle, averaged across simulations, decreased over trials, indicating that the value weight $\boldsymbol{w}$ indeed tended to become aligned to the random feedback $\boldsymbol{c}$. We then examined whether better alignment of $\boldsymbol{w}$ to $\boldsymbol{c}$ related to better development of state value by looking at the relation between the angle between $\boldsymbol{w}$ and $\boldsymbol{c}$ and the value of the pre-reward state at 1000-th trial. As shown in Fig. 3B, there was a negative correlation such that the smaller the angle (i.e., more aligned), the larger the state value ($r = -0.288$, $p = 0.00362$), consistent with our expectation. These results indicate that the mechanism of feedback alignment, previously shown to work for supervised learning, also worked for TD learning of value weights and recurrent/feed-forward connections.

How did the feedback alignment mechanistically occur? We made an attempt to obtain an intuitive understanding. Assume that positive TD-RPE ($\delta(t) > 0$) is generated in a state, S (= $\boldsymbol{x}(t)$) in a trial. Because of the update rule for $\boldsymbol{w}$ ($\boldsymbol{w} \leftarrow \boldsymbol{w} + a\delta(t)\boldsymbol{x}(t)$) (Eq. 1.9 in the Methods), $\boldsymbol{w}$ is updated in the direction of $\boldsymbol{x}(t)$. Next, what is the effect of updates of recurrent/feed-forward connections ($\mathbf{A}$ and $\mathbf{B}$) on $\boldsymbol{x}$? For simplicity, here we consider the case where observation is null ($\boldsymbol{o} = \mathbf{0}$) and so $\boldsymbol{x}(t) = \boldsymbol{f}(\mathbf{A}\boldsymbol{x}(t-1))$ holds (but similar argument can be done in the case where observation is not null). If $\mathbf{A}$ is replaced with its updated one, it can be calculated that i-th element of $\mathbf{A}\boldsymbol{x}(t-1)$ will hypothetically change by $c_i \times$ (a positive value) (technical note: the value is $a\delta(t)\{\Sigma_j x_j(t-1)^2\}(0.5 + x_i(t))(0.5 - x_i(t))$ (c.f., Eq. 1.10) which is positive unless $x(t-1) = \mathbf{0}$), and therefore the vector $\mathbf{A}\boldsymbol{x}(t-1)$ as a whole will hypothetically change by a vector that is in a relatively close angle with $\boldsymbol{c}$, or more specifically, is in the same quadrant as (and thus within at maximum 90° from) $\boldsymbol{c}$ (for example, $[c_1\ c_2\ c_3]^{\mathrm{T}}$ and $[0.5c_1\ 1.2c_2\ 0.8c_3]^{\mathrm{T}}$). Then, because f is a monotonically increasing sigmoidal function, $\boldsymbol{x}(t) = \boldsymbol{f}(\mathbf{A}\boldsymbol{x}(t-1))$

will also hypothetically change by a vector that is in a relatively close angle with $\boldsymbol{c}$. This was indeed the case in our simulations as shown in Fig. 3C, which plotted the angle between the hypothetical change in $\boldsymbol{x}(t) = \boldsymbol{f}(\mathbf{A}\boldsymbol{x}(t-1),\mathbf{B}\boldsymbol{o}(t-1))$ in case $\mathbf{A}$ and $\mathbf{B}$ were replaced with their updated ones, multiplied with the sign of TD-RPE (sign($\delta(t)$)), and the fixed random feedback vector $\boldsymbol{c}$ across time-steps.

In this way, at state S where TD-RPE is positive, $\boldsymbol{w}$ is updated in the direction of $\boldsymbol{x}(t)$, and $\boldsymbol{x}(t)$ will hypothetically change by a vector that is in a relatively close angle with $\boldsymbol{c}$ if $\mathbf{A}$ is replaced with its updated one. Then, if the update of $\boldsymbol{w}$ and the hypothetical change in $\boldsymbol{x}(t)$ due to the update of $\mathbf{A}$ could be integrated, $\boldsymbol{w}$ would become aligned to $\boldsymbol{c}$ (if TD-RPE is instead negative, $\boldsymbol{w}$ is updated in the opposite direction of $\boldsymbol{x}(t)$, and $\boldsymbol{x}(t)$ will hypothetically change by a vector that is in a relatively close angle with $-\boldsymbol{c}$, and so the same story holds in the end).

There is, however, a caveat regarding how the update of $\boldsymbol{w}$ and the hypothetical change in $\boldsymbol{x}(t)$ can be integrated. Although technical, here we briefly describe the caveat and a possible solution for it. The updates of $\boldsymbol{w}$ and $\mathbf{A}$ use TD-RPE, which are calculated based on $v(t) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}(t)$ and $v(t+1) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}(t+1)$, and so $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t+1)$ should already be determined beforehand. Therefore, the hypothetical change in $\boldsymbol{x}(t)$ due to the update of $\mathbf{A}$, described in the above, does not actually occur (this was why we mentioned 'hypothetical') and thus cannot be integrated with the update of $\boldsymbol{w}$. Nevertheless, integration could still occur across successive trials, at least to a certain extent. Specifically, although TD-RPEs at S in successive trials would generally differ from each other, they would still tend to have the same sign, as was indeed the case in our simulations (Fig. 3D). Also, although the trajectories of RNN activity ($\boldsymbol{x}$) in successive trials would differ, we could expect a certain level of similarity because the RNN is entrained by observation-representing inputs, again as was indeed the case in our example simulation (Fig. 3E). Then, the hypothetical change in $\boldsymbol{x}(t)$ due to the update of $\mathbf{A}$, considered above, could become a reality in the next trial, to a certain extent, and could thus be integrated into the update of $\boldsymbol{w}$, explaining the occurrence of feedback alignment.

2.4 Simulation of tasks with probabilistic structures of reward timing/existence

Previous work [54] examined the response of DA neurons in cue-reward association tasks in which reward timing was probabilistically determined (early in some trials but late in other trials). There were two tasks, which were largely similar but there was a key difference that reward was given in all the trials in one task whereas reward was omitted in some randomly determined trials in another task. Starkweather et al. [54] found that the DA response to later reward was smaller than the response to earlier reward in the former task, presumably reflecting the animal's belief that delayed reward will surely come, but the opposite was the case in the latter task, presumably because the animal suspected that reward was omitted in that trial. Starkweather et al.[54] then showed that such response patterns could be explained if DA encoded TD-RPE under particular state representations

that incorporated the probabilistic structures of the task (called the 'belief state'). In that study, such state representations were 'handcrafted' by the authors, but the subsequent work [26] showed that the original value-RNN with backprop (BPTT) could develop similar representations and reproduce the experimentally observed DA patterns.

In order to examine if our online value-RNN with fixed random feedback could also explain those experimental results, we simulated two tasks (Fig. 4A) that were qualitatively similar to (though simpler than) the two tasks examined in the experiments [54]. In our task 1, a cue was always followed by a reward either two or four time-steps later with equal probabilities. Task 2 was the same as task 1 except that reward was omitted with 40% probability. In task 1, if reward was not given at the early timing (i.e., two-steps later than cue), agent could predict that reward would be given at the late timing (i.e., four-steps later than cue), and thus TD-RPE upon reward at the late timing is expected to be smaller than TD-RPE upon reward at the early timing (if agent perfectly learned the task structure, TD-RPE upon reward at the late timing should be 0). By contrast, in task 2, if reward was not given at the early timing, it might indicate that reward would be given at the late timing but might instead indicate that reward would be omitted in that trial, and thus TD-RPE upon reward at the late timing is expected to exist and can even be larger than TD-RPE upon reward at the early timing.

In these tasks, states can be defined in the following way. There were two types of trials, with early or late reward, in task 1, and additionally one more type of trial, without reward, in task 2 (Fig. 4Ba, top). For each timing after receival of cue information in each of these trial types, its value can be estimated through simulations (Fig. 4Ba, bottom). Agent could not know the current trial type until receiving reward at the early timing or the late timing or receiving no reward at both timings. Until these timings, agent could have probabilistic belief about the current trial type, e.g., 50% in the trial with early reward and 50% in the trial with late reward (in task 1) or 30% in the trial with early reward, 30% in the trial with late reward, and 40% in the trial without reward (in task 2) (Fig. 4Bb). States of the timings after receival of cue information can be defined by incorporating these probabilistic beliefs at each timing (Fig. 4Bc, top). Their true values can be calculated by taking (mathematical) expected value of the estimated values of each timing in each trial type (Fig. 4Bc, bottom). Expected TD-RPE calculated from the estimated true values (Fig. 4C) exhibited features that matched the conjecture mentioned above: in task 1, TD-RPE upon reception of late reward, which was actually 0, was smaller than TD-RPE upon reception of early reward, whereas in task 2, TD-RPE upon reception of late reward was larger than TD-RPE upon reception of early reward (as indicated by the inequality signs on Fig. 4C).

As mentioned above, the previous work [54] has shown that VTA DA neurons exhibited similar activity patterns to the abovementioned TD-RPE patterns, and the subsequent work [26] has shown that the original value-RNN with backprop (BPTT) could reproduce such TD-RPE patterns. We examined how our oVRNNbp and oVRNNrf, (with 12 RNN units) behaved in our simulated two

tasks. oVRNNbp developed the expected TD-RPE patterns, i.e., smaller TD-RPE upon late than early timing in task 1 but opposite pattern in task 2 (Fig. 4F), and oVRNNrf also developed such patterns although the effect size for task 1 was small (Fig. 4G). These results indicate that online value-RNN could learn the probabilistic structures of the tasks even with fixed random feedback. By contrast, agents with punctate state representation without or with continuous value update across trials (Fig. 4D,E), as well as agent with untrained fixed RNN (Fig. 4H), could not develop such patterns well.

3.1 Online value-RNN with further biological constraints

So far, the activities of neurons in the RNN ($\boldsymbol{x}$) were initialized to pseudo standard normal random numbers, and thereafter took numbers in the range between −0.5 and 0.5 that was the range of the sigmoidal input-output function. The value weights ($\boldsymbol{w}$) could also take both positive and negative values since no constraint was imposed. The fixed random feedback in oVRNNrf ($\boldsymbol{c}$) was generated by pseudo standard normal random numbers, and so could also be positive or negative. Negativity of the neurons' activities and the value weights could potentially be regarded as inhibitory or smaller-than-baseline quantities. However, because neuronal firing rate is non-negative and cortico-striatal projections are excitatory, it would be biologically more plausible to assume that the activities of neurons in the RNN and the value weights are non-negative. As for the fixed random feedback, if it is negative, the update rule becomes anti-Hebbian under positive TD-RPE, and so assuming non-negativity would be plausible since Hebbian property has been suggested for rapid plasticity of cortical synapses [55]. Regarding the connection weights in/onto the RNN, here we keep the original assumption that they could be positive or negative because it could be an approximate description of recurrent neuronal network with both recurrent excitation and inhibition. Later (Section 4.1) we will examine extended models that incorporate excitatory and inhibitory units and conform to the Dale's law.

Other than the sign of connection weights, there was another biological plausibility issue in the update rule for recurrent and feed-forward connections that were derived from the gradient descent. Specifically, the dependence on the post-synaptic activity was non-monotonic, maximized at the middle of the range of activity. It would be more biologically plausible to assume a monotonic increase (while an *opposite* shape of non-monotonicity, once decrease and thereafter increase, called the BCM (Bienenstock-Cooper-Munro) rule has actually been suggested [56-58]).

In order to address these issues, we considered revised models. We first considered a revised oVRNNbp (with backprop-type transported weights), referred to as oVRNNbp-rev, in which the RNN activities and the value weights were constrained to be non-negative, while the non-monotonic dependence of the update rule on the post-synaptic activity remained unchanged (Fig. 5A). We then considered a revised oVRNNrf, referred to as oVRNNrf-bio, in which the fixed random feedback, as

well as the RNN activities and the value weights, were constrained to be non-negative, and also the update rule was modified so that the dependence on the post-synaptic activity became monotonic (with saturation) (Fig. 5B).

We examined how these revised models, in comparison with agent with untrained RNN that also had non-negative constraint for $\boldsymbol{x}$ and $\boldsymbol{w}$, performed in the Pavlovian cue-reward association task examined above (the numbers of RNN-units and trials were set to 12 and 1500, respectively). oVRNNbp-rev well developed state values toward reward (Fig. 6A). oVRNNrf-bio also developed state values to a largely comparable extent (Fig. 6B). By contrast, agent with untrained RNN could not develop such a pattern of state values (Fig. 6C). This, however, could be because initially set recurrent/feed-forward connections were far from those learned in the online value-RNNs. Therefore, as a more strict control, we conducted simulations of agent with untrained RNN with non-negative $\boldsymbol{x}$ and $\boldsymbol{w}$, where in each simulation the recurrent/feed-forward connections were set to be those shuffled from the learnt connections in a simulation of oVRNNrf-bio (hereafter we refer to these two types of untrained RNN as "naive untrained RNN" and "shuffled untrained RNN"). The model with shuffled untrained RNN developed state values somewhat better than the naive untrained RNN case (Fig. 6D), but still worse than oVRNNbp-rev and oVRNNrf-bio.

3.2 Systematic simulations and analyses

We varied the number of RNN units (n), with the learning rate for value weights normalized by dividing by $n/12$, and compared the performance (mean of squared errors of state values between cue and reward at 1500-th trial) of oVRNNbp-rev and oVRNNrf-bio, in comparison with models with naive or shuffled untrained RNN. As shown in the left panel of Fig. 6E, oVRNNbp-rev and oVRNNrf-bio exhibited largely comparable performance and always outperformed the models with untrained RNN ($p < 2.5\times10^{-12}$ in Wilcoxon rank sum test for oVRNNbp-rev or oVRNNrf-bio vs naive or shuffled untrained for each number of RNN units), although oVRNNbp-rev somewhat outperformed or underperformed oVRNNrf-bio when the number of RNN units was small (≤ 10 ($p < 0.00029$)) or large (≥ 25 ($p < 3.7\times10^{-6}$)), respectively (Figure 6G shows the learned state values and TD-RPEs in oVRNNbp-rev and oVRNNrf-bio in the cases with 40 RNN units, compared to the estimated true values). Remarkably, oVRNNrf-bio generally achieved better performance than both oVRNNbp and oVRNNrf, which did not have the non-negative constraint (Wilcoxon rank sum test, vs oVRNNbp : $p < 7.8\times10^{-6}$ for 5 or ≥ 25 RNN units; vs oVRNNrf: $p < 0.021$ for ≤ 10 or ≥ 20 RNN units).

The left panel of Figure 6F shows the mean of the elements of the recurrent and feed-forward connections at 1500-th trial in the different models. As shown in this figure, these connections (initialized to pseudo standard normal random numbers) were learnt to become negative on average, in oVRNNbp-rev and oVRNNrf-bio. This learnt negative-dominance (inhibition-dominance) could

possibly be related, e.g., through prevention of excessive activity, to the good performance of oVRNNrf-bio and also the better performance of the shuffled untrained RNN than the naive untrained RNN. The green and light blue lines in the right panels of Figure 6E and Figure 6F show the results for special cases where the random feedback in oVRNNrf-bio was fixed to the direction of $(1, 1, ..., 1)^T$ (i.e., uniform feedback) with a random non-negative magnitude (green line) or a fixed magnitude of 0.5 (light blue line). The performance of these special cases, especially the former (with random magnitude) was somewhat worse than that of oVRNNrf-bio, but still better than that of the models with untrained RNN. The blue dotted lines in the right panels of Figure 6E and Figure 6F show the results where the modified update rule of oVRNNrf-bio was changed back to the original rule with non-monotonic dependence on the post-synaptic activity (Fig. 5A). The performance of this model was somewhat worse than oVRNNrf-bio, indicating that the biologically motivated modification of the update rule in fact improved the performance.

Figure 6H shows contribution ratios of PCs of the time series of RNN activities in each model with 20 RNN units. Compared with the cases with naive/shuffled untrained RNN, in oVRNNbp-rev and oVRNNrf-bio, later components had relatively high contributions (PC5~20 $p < 1.4\times10^{-6}$ (*t*-test vs naive) or < 0.014 (vs shuffled) in oVRNNbp-rev; PC6~20 $p < 2.0\times10^{-7}$ (vs naive) or PC7~20 $p < 5.9\times10^{-14}$ (vs shuffled) in oVRNNrf-bio), explaining their superior value-learning performance. Figure 6I shows how learning proceeded across trials in the models with 20 RNN units. While oVRNNbp-rev and oVRNNrf-bio eventually reached a comparable level of errors, oVRNNrf-bio outperformed oVRNNbp-rev in early trials (at 200, 300, 400, or 500 trials; $p < 0.049$ in Wilcoxon rank sum test for each). This is presumably because the value weights did not develop well in early trials and so the backprop-type feedback, which was the same as the value weights, did not work well, while the non-negative fixed random feedback worked finely from the beginning. Figure 6J shows the cases with longer cue-reward delays, with default or halved learning rates. As the delay increased, the mean squared error of state values (at 3000-th trial) increased, but the relative superiority of oVRNNbp-rev and oVRNNrf-bio over the models with untrained RNN remained to hold, except for a few cases with 5 RNN units (5 delay oVRNNrf-bio vs shuffled with default learning rate, 6 delay oVRNNrf-bio vs naive or shuffled with halved learning rate) ($p < 0.047$ in Wilcoxon rank sum test for oVRNNbp-rev or oVRNNrf-bio vs naive or shuffled untrained for each number of RNN units for each delay).

We further examined how the revised online value-RNN models performed in the two tasks with probabilistic structures examined above. The models with 12 RNN units appeared not able to produce the expected different patterns of TD-RPEs in the two tasks (TD-RPE at early reward > TD-RPE at late reward in task 1 and opposite pattern in task 2), and we increased the number of RNN units to 20. Then, both oVRNNbp-rev and oVRNNrf-bio produced such TD-RPE patterns (Fig. 7A,B) whereas the models with untrained RNN could not (Fig. 7C,D). This indicates that the online value-RNN with random feedback and further biological constraints could learn the differential

characteristics of the tasks.

3.3 Loose alignment and feedback alignment

Coming back to the original cue-reward association task, we examined how the angle between the value weights (***w***) and the random feedback (***c***) changed across trials in oVRNNrf-bio with 12 RNN units. As shown in Fig. 8A, the angle was on average smaller than 90°, which was the chance-level angle in the case without non-negative constraint, from the beginning, while there was no further alignment over trials. This could be understood as follows. Because both the value weights (***w***) and the random feedback (***c***) were now constrained to be non-negative, these two vectors were ensured to be in a relatively close angle (i.e., in the same quadrant) from the beginning. By virtue of this loose alignment, the random feedback could act similarly to backprop-type transported-weight feedback, even without further alignment.

We examined if the angle between the value weights (***w***) and the random feedback (***c***) at 1500-th trial was associated with the developed value of pre-reward state across simulations, but found no association (r = 0.0117, p = 0.908) (Fig. 8B). We then examined if the ***w***-***c*** angle at earlier trials (2nd - 500-th trials) was associated with the developed values at 500-th trial, with the number of simulations increased to 1000 so that small correlation could be detected. We found that the ***w***-***c*** angle at initial trials (2nd - around 10-th trials) was negatively correlated with the developed values of the reward state and preceding states at 500-th trial (Fig. 8C). As for the reward state, negative correlation at around 100-th - 300-th trial was also observed. These results suggest that better alignment of ***w*** and ***c*** at initial and early timings was associated with better development of state values, in line with the conjecture that loose alignment of ***w*** and ***c*** coming from the non-negative constraint supported learning. It should be noted, however, that there were cases where positive (although small) correlation was observed. Its exact reason is not sure, but it could be related to the fact that largeness of developed values or the speed of value development does not necessarily mean good learning.

As mentioned above, while the angle between ***w*** and ***c*** was on average smaller than 90° from the beginning, there was no further alignment over trials. This seemed mysterious because the mechanism for feedback alignment that we derived for the models without non-negative constraint was expected to work also for the models with non-negative constraint. As a possible reason for the non-occurrence of feedback alignment, we guessed that one or a few element(s) of ***w*** grew prominently during learning, and so ***w*** became close to an edge or boundary of the non-negative quadrant and thereby angle between ***w*** and other vector became generally large (as illustrated in Fig. 8D). Figure 8Ea shows the mean±SEM of the elements of ***w*** ordered from the largest to smallest ones after 1500 trials. As conjectured above, a few elements indeed grew prominently.

We considered that if a slight decay (forgetting) of value weights (c.f., [59-61]) was assumed, such

a prominent growth of a few elements of $\boldsymbol{w}$ may be mitigated and alignment of $\boldsymbol{w}$ to $\boldsymbol{c}$, beyond the initial loose alignment because of the non-negative constraint, may occur. These conjectures were indeed confirmed by simulations (Fig. 8Eb,c and Fig. 8F). The mean squared value error slightly increased when the value-weight-decay was assumed (Fig. 8G), however, presumably reflecting a decrease in developed values and a deterioration of learning because of the decay.

4.1 Models with excitatory and inhibitory units

As mentioned above, in oVRNNbp-rev and oVRNNrf-bio, the connection weights in/onto the RNN could be both positive and negative, against the Dale's law. Recent studies started to examine neural networks incorporating the Dale's law [62, 63] or other connectivity features [64]. So we examined extended models, named oVRNNbp-rev-ei and oVRNNrf-bio-ei, which incorporated excitatory E-units, modeling pyramidal cells, and inhibitory I-units, modeling fast-spiking (FS) cells (Fig. 9A). Cortical excitation can operate slowly due to slow synaptic dynamics [46, 47] (see the description about the time-step in the Methods for details). In contrast, inhibition from FS cells to pyramidal cells may operate more quickly, since it was shown [65] that observed phases of regular-spiking (RS, putatively pyramidal) cells' and FS cells' spikes [66] could be explained by fast FS→RS inhibition and temporally distributed recurrent excitation.

Given these, we assumed that excitation from E-units to E- and I-units took one time-step whereas I→E inhibition operated within a time-step, and also that each E-unit received inputs from all the E-units and a particular I-unit (although this assumption could be supported by the abovementioned suggestions, its validity remains largely open). Chemical and electrical connections between FS cells exist and are suggested to serve for synchronization or oscillation [67, 68], but we omitted I→I connections because our models did not describe fast spike dynamics. Since FS cells can fire at high frequencies, we assumed that the activation function for I-units was not saturating but linear. Lastly, we did not assume plasticity for connections from/to I-units. The connection weights onto the E- and I-units from the observation units and E-units were non-negatively initialized, specifically, initialized to pseudo normal random numbers with mean = 3 and standard deviation = 1 and rectified to 0 when becoming negative.

We examined how these extended models behaved in the Pavlovian task and the probabilistic tasks. As shown in Fig. 9B, oVRNNbp-rev-ei and oVRNNrf-bio-ei learned the state values in the Pavlovian task much more accurately than the models with naive untrained RNN or untrained RNN whose connections from the observation and E-units to E- and I-units were shuffled from oVRNNrf-bio-ei ($p < 5.2\times10^{-12}$ in Wilcoxon rank sum test for oVRNNbp-rev-ei or oVRNNrf-bio-ei vs naive or shuffled untrained for each number of RNN units). oVRNNbp-rev-ei somewhat outperformed or underperformed oVRNNrf-bio-ei when the number of RNN units was small (≤10 ($p < 0.0091$)) or relatively large (15~35 ($p < 0.027$)), respectively. Also, as shown in Fig. 9C, oVRNNbp-rev-ei and

oVRNNrf-bio-ei with 20 E-units and 20 I-units generated the expected different patterns of TD-RPEs in the two tasks (TD-RPE at early reward > TD-RPE at late reward in task 1 and opposite pattern in task 2) although the effect size for task 2 in oVRNNrf-bio-ei was small, while the models with untrained RNN did not.

As such, the extended models with E-units and I-units showed largely similar behaviors to those of the original oVRNNbp-rev and oVRNNrf-bio with mixed positive and negative RNN weights. This is actually reasonable, because combining the update equations for I-units and E-units in the extended models (top two equations in Fig. 9A) results in an equation largely similar to the update equation for RNN units in the original models (top equation in Fig. 1). In other words, the original models with mixed positive and negative RNN weights could be regarded as a simplified description of the models with E-units and I-units under the abovementioned assumptions. Therefore, for simplicity, we will return to mixed positive and negative RNN weights in the following.

4.2 Task with distractor cue

So far we have examined situations where there existed a reward and a cue associated with the reward. However, in real environments, it is likely that there exist both reward-associated and non-associated (distractor) cues and agent does not initially know which cue is associated with reward and which is not. Learning cue-reward association in such distractor-existing environments is generally not easy for biologically constrained models, and it has been addressed by only few previous work [69]. We examined whether our biologically constrained oVRNNrf-bio, as well as oVRNNbp-rev, could learn the cue-reward association under the presence of distractor cue. We considered a simple case where there existed a distractor cue, which was presented to the agent with a certain probability at every time-step, between cue-reward duration or reward-cue duration (i.e., inter-trial interval) or simultaneously with cue or reward. As for the agent's models, we assumed that the observation inputs had an additional element (dimension), which was set to 1 when the distractor was presented and 0 when not (Fig. 10A).

We examined how oVRNNbp-rev, oVRNNrf-bio, and the models with untrained RNN behaved in the modified Pavlovian task with a distractor cue, which was presented with probability 0, 0.1, 0.2, or 0.3 at every time-step (Fig. 10B-E, left panels). As a result, even when there was such a distractor cue, oVRNNbp-rev and oVRNNrf-bio could still learn the state values better than the models with naive or shuffled untrained RNN ($p < 1.7\times10^{-10}$ in Wilcoxon rank sum test for oVRNNbp-rev or VRNNrf-bio vs naive or shuffled untrained for each number of RNN units for each level of distractor probability) (Fig. 10C-E, middle panels), although the accuracy moderately decreased compared with the case without distractor cue (Fig. 10B, middle panel). These results suggest robustness of the learning ability of oVRNNrf-bio against distractor in realistic situations. We further examined how the models with E- and I-units behaved in the task with a distractor cue, and confirmed that even in

the presence of disctractor cue, oVRNNbp-rev-ei and oVRNNrf-bio-ei could learn the state values better than the models with naive or shuffled untrained RNN ($p < 3.2\times10^{-8}$ in Wilcoxon rank sum test for oVRNNbp-rev-ei or oVRNNrf-bio-ei vs naive or shuffled untrained for each number of RNN units for each level of distractor probability) (Fig. 10B-E, right panels).

4.3 Incorporation of action selection

Ultimate purpose of animals and RL agents is to optimize their policy, i.e., probability of action selection at each state to maximize rewards. Therefore, we examined if our models could be extended to incorporate action selection. In reference to the proposals that algorithm akin to the actor-critic method may be implemented in the brain [35, 70, 71], we considered extended models oVRNNbp-rev-ac and oVRNNrf-bio-ac, which incorporated an actor-critic architecture (Fig. 11A). Specifically, each RNN unit was assumed to connect to not only the state-value(v)-representing unit in the ventral striatum but also the action-value(q_k)-representing units in the dorsal striatum. Their non-negative weights (u_{kj}) represented (as a vector) action preferences, which slightly decayed with time so as not to unboundedly increase.

It has been suggested that action is selected through competition of neural populations in the striatum-thalamus-cortex(-striatum) circuit, which represent or receive action values, in the presence of noise [72-74]. However, because our models did not describe fast neural dynamics (see the description about the time-step in the Methods), we assumed a soft-max function for action selection, i.e., assumed that an action was selected according to a soft-max probability determined by the difference in the action values when there were two action candidates. We then assumed that there is a cortical region, which contains action-representing neural populations, implemented as 'action units' in the model (Fig. 11A). Action unit was assumed to become active (i.e., = 1) when the corresponding action was selected and inactive (i.e., = 0) otherwise, and these action units were assumed to send inputs to the RNN, similarly to the observation units informing the presence of cue and reward.

We examined how these models (oVRNNbp-rev-ac and oVRNNrf-bio-ac) and control models with naive or shuffled untrained RNN behaved in two-alternative choice tasks. In the first choice task (Fig. 11Ba), taking action 1 lead to a large (size 2) reward two time-steps later whereas taking action 2 lead to a small (size 1) reward two time-steps later. oVRNNbp-rev-ac and oVRNNrf-bio-ac, after training, successfully selected the better action, action 1, in most cases, whereas the models with untrained RNN did not develop strong tendency to select action 1 (Fig. 11Bb,c). Next, in the second choice task (Fig. 11Ca), taking action 1 lead to a large (size 2) reward two time-steps later whereas taking action 2 lead to a small (size 1) reward one time-step later. So this task imposed inter-temporal choice between delayed large reward and sooner small reward. oVRNNbp-rev-ac, after training, tended to select action 1, which was better than action 2 even when presumed temporal discounting

(0.8 per time-step) was taken into account (Fig. 11Cb,c). oVRNNrf-bio-ac also tended to select action 1 when the number of RNN units was not small. On the other hand, the models with untrained RNN tended to select sooner small reward. These results suggest that oVRNNrf-bio-ac, as well as oVRNNbp-rev-ac, could learn to select advantageous action to a certain extent, even in the case of inter-temporal choice.

Discussion

We have shown that state representation and value can be learned online in the RNN and its readout by using random feedback instead of biologically unavailable downstream weights. This was achieved through feedback alignment, and we have presented an intuitive understanding of its mechanism. We have further shown that the non-negative constraint realizes loose alignment of the forward weights and feedback from the beginning, which appeared to support learning.

Roles of DA

Midbrain DA neurons project to both striatum and cortex, including the prefrontal cortex [75] and the hippocampus [76]. As for striatal DA, DA-dependent cortico-striatal plasticity is considered to implement TD-RPE-based value update [7, 77]. By contrast, while cortical DA has been implicated in working memory [78-81], decision making [82], and aversive memory [83], its role as TD-RPE remains unclear, despite findings suggesting that cortical DA does encode (TD-)RPE [84-87] and modulate plasticity [88, 89]. Learnability of our biologically constrained online value-RNN suggests that TD-RPE-encoding cortical DA modulates plasticity of RNN so that appropriate state representation can be learnt.

Many studies reported heterogeneities of DA signals, which may come from encoding prediction errors other than RPE or feature-specific components of RPE [90]. Referring to a result [91] indicating DA's encoding of non-reward PEs and the fact that DA neurons receive inputs from the cerebellum [92, 93], which presumably implements supervised learning [94], a recent work [45] proposed that DA encodes vector-valued errors used for supervised learning of actions in continuous space. In contrast, we assumed DA's encoding of scalar TD-RPE, which can be consistent with the heterogeneity due to encoding of feature-specific RPE components [90]. The previous model [45] and our model can coexist, with different DA neuronal populations encoding different types of errors, or a single DA neuron switching its encoding depending on context/inputs.

VTA DA neurons project also to the basolateral amygdala (BLA) [95], and DA regulates plasticity also there [96]. Moreover, VTA→BLA DA entailed properties of TD-RPE, although increased also upon aversive event, and was not itself reinforcing but crucial for the formation of environmental model [97]. BLA has recurrent connections [98], projects to the striatum [99, 100], and engages in abstract context representation [101]. Thus, given that goal-directed-like behavior could be achieved through sophisticated state representation [10-12], it could potentially be learned by value-RNN-like mechanism in the BLA. Whether such sophisticated representation can be learned, however, remains open, and it might require multidimensional error [102] beyond TD-RPE.

There are many DA-related mechanisms that were not incorporated into our models, including

the distinctions of D1-direct and D2-indirect pathways [103-107] and cortical projections to them [108-111], as well as mechanisms underlying TD-RPE encoding (c.f., [106, 112]) or learning for it (c.f., [69]). Future studies are expected to incorporate these.

Predictions and implications of our models

oVRNNrf predicts that the feedback vector $\boldsymbol{c}$ and the value-weight vector $\boldsymbol{w}$ become gradually aligned, while oVRNNrf-bio predicts that $\boldsymbol{c}$ and $\boldsymbol{w}$ are loosely aligned from the beginning. Element of $\boldsymbol{c}$ could be measured as the magnitude of pyramidal cell's response to DA stimulation. Element of $\boldsymbol{w}$ corresponding to a given pyramidal cell could be measured, if striatal neuron that receives input from that pyramidal cell can be identified (although technically demanding), as the magnitude of response of the striatal neuron to activation of the pyramidal cell. Then, the abovementioned predictions could be tested by (i) identify cortical, striatal, and VTA regions that are connected, (ii) identify pairs of cortical pyramidal cells and striatal neurons that are connected, (iii) measure the responses of identified pyramidal cells to DA stimulation, as well as the responses of identified striatal neurons to activation of the connected pyramidal cells, and (iv) test whether DA→pyramidal responses and pyramidal→striatal responses are associated across pyramidal cells, and whether such associations develop through learning.

Testing this prediction, however, would be technically quite demanding, as mentioned above. An alternative way of testing our model is to manipulate the cortical DA feedback and see if it will cause (re-)alignment of value weights (i.e., cortical striatal strengths). Specifically, our model predicts that if DA projection to a particular cortical locus is silenced, effect of the activity of that locus on the value-encoding striatal activity will become diminished.

We have shown that oVRNNrf and oVRNNrf-bio could work even when the random feedback was uniform, i.e., fixed to the direction of $(1, 1, ..., 1)^T$, although the performance was somewhat worse. This is reasonable because uniform feedback can still encode scalar TD-RPE that drives our models, in contrast to a previous study [45], which considered DA's encoding of vector error and thus regarded uniform feedback as a negative control. If oVRNNrf/oVRNNrf-bio-like mechanism indeed operates in the brain and the feedback is near uniform, alignment of the value weights $\boldsymbol{w}$ to near (1, 1, ..., 1) is expected to occur. This means that states are (learned to be) represented in such a way that simple summation of cortical neuronal activity approximates value, thereby potentially explaining why value is often correlated with regional activation (fMRI BOLD signal) of cortical regions [113]. Notably, uniform feedback coupled with positive forward weights was shown to be effective also in supervised learning of one-dimensional output in feed-forward network [114], and we guess that loose alignment may underlie it.

On the RNN unit

In our oVRNNbp without non-negative constraint, as the number of RNN units increased, the squared error initially decreased but then increased (Fig. 2J) (while intriguingly it was not the case for the models with non-negative constraint (Fig. 6E)). In contrast, in the original value-RNN [12, 26], the ability to develop belief-state-like representation was reported to improve as the number of RNN units increased to 100 or 50. There are at least two possible reasons for this difference, other than the difference in the performance measures. First one is the difference in the update rules. As mentioned earlier, the original value-RNN used BPTT [27] whereas our oVRNNbp used an online learning rule, which only considered the influence of the recurrent weights at the previous time step.

Second one is a difference in the RNN unit. Specifically, the original value-RNN used the Gated Recurrent Unit (GRU) cell [115] whereas we used a simple sigmoidal function. RNN with simple nonlinear unit is known to have the vanishing gradient problem [116], which could be alleviated by using memorable/gated RNN such as the Long Short-Term Memory (LSTM) unit [117] or the GRU cell [115]. We used the simple sigmoidal unit because biological plausibility of the GRU cell appeared elusive. However, gated unit similar to the LSTM unit has actually been proposed to be implemented in cortical microcircuits [118], and incorporation of such biologically plausible gated unit into our online value-RNN would be a hopeful direction.

From a bottom-up viewpoint, our RNN unit did not incorporate spiking [41, 57, 119] nor nonlinear dendritic computations [44, 120, 121]. Recent studies suggest that dendritic mechanisms [37, 38], possibly in combination with burst-dependent plasticity [41, 42], can realize credit assignment without backprop in supervised and unsupervised learning [122, 123]. Also, recent model of hippocampus [25] has shown that a network of multi-compartment units could learn complex representations. Having dendritic mechanisms is different from just increasing the number of neural-network layers because of their own specific features/constraints, and it was argued [44] that adding such biological constraints enables learning in deep neural networks. Therefore, incorporation of biological details into RNN unit in our models would be hopeful also from the bottom-up viewpoint.

Comparison to other algorithms

As an alternative to backprop in hierarchical network, aside from feedback alignment [36], Associative Reward-Penalty ($A_{R\text{-}P}$) algorithm has been proposed [124-126]. In $A_{R\text{-}P}$, the hidden units behave stochastically, allowing the gradient to be estimated via stochastic sampling. Recent work [127] has proposed Phaseless Alignment Learning (PAL), in which high-frequency noise-induced learning of feedback projections proceeds simultaneously with learning of forward projections using the feedback in a lower frequency. Noise-induced learning of the weights on readout neurons from untrained RNN by reward-modulated Hebbian plasticity has also been demonstrated [128]. Such noise- or perturbation-based [40] mechanisms are biologically plausible because neurons and neural networks can exhibit noisy or chaotic behavior [129-131], and might improve the performance of value-RNN if implemented.

Regarding learning of RNN, "e-prop" [35] was proposed as a locally learnable online approximation of BPTT [27], which was used in the original value RNN [26]. In e-prop, neuron-specific learning signal is combined with weight-specific locally-updatable "eligibility trace". Reward-based e-prop was also shown to work [35], both in a setup not introducing TD-RPE with symmetric or random feedback (their Supplementary Figure 5) and in another setup introducing TD-RPE with symmetric feedback (their Figure 4 and 5). Compared to these, our models differ in multiple ways.

First, we have shown that alignment to random feedback occurs in the models driven by TD-RPE. Second, our models do not have "eligibility trace" (nor memorable/gated unit, different from the original value-RNN [26]), but could still solve temporal credit assignment to a certain extent because TD learning is by itself a solution for it (notably, recent work showed that combination of TD(0) and model-based RL well explained rat's choice and DA patterns [132]). However, as mentioned before, single time-step in our models was assumed to correspond to hundreds of milliseconds, incorporating slow synaptic dynamics, whereas e-prop is an algorithm for spiking neuron models with a much finer time scale. From this aspect, our models could be seen as a coarse-time-scale approximation of e-prop. On top of these, our results point to a potential computational benefit of biological non-negative constraint, which could effectively limit the parameter space and promote learning.

Methods

Online value-RNN with backprop (oVRNNbp)

We constructed an online value-RNN model based on the previous proposals [12, 26] but with several differences. We assumed that the activities of neurons in the RNN at time t+1 were determined by the activities of these neurons and neurons representing observation (cue, reward, or nothing) at time t:

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{o}(t)), \quad \text{(Eq. 1.1)}$$

where

$\boldsymbol{x} = (x_j)$: activity of j-th neuron in the RNN (j = 1, .., n)

$\boldsymbol{o} = (o_k)$: activity of k-th neuron in the observation layer (k = 1, 2)

if there was a cue at t, $\boldsymbol{o}(t) = (1\ 0)^{\mathrm{T}}$,

if there was a reward at t, $\boldsymbol{o}(t) = (0\ 1)^{\mathrm{T}}$,

and otherwise, $\boldsymbol{o}(t) = (0\ 0)^{\mathrm{T}}$

$\mathbf{A} = (A_{ij})$: recurrent connection strength from x_j to x_i

$\mathbf{B} = (B_{ik})$: feed-forward connection strength from o_k to x_i

$f(z) = 1/(1 + \exp(-z)) - 0.5$: (Eq. 1.2)

sigmoidal function representing neuronal input-output relation

The estimated value of the state at t was calculated as

$$v(t) = \boldsymbol{w}^{\mathrm{T}}\, \boldsymbol{x}(t) \quad \text{(Eq. 1.3)}$$

where

$\boldsymbol{w} = (w_j)$

were the value weights. The error between this estimated value and the true value, $v_{true}(t)$, was defined as:

$$\varepsilon(t) = v_{true}(t) - v(t) \quad \text{(Eq. 1.4)}$$

Parameters w_j, A_{ij}, and B_{ik} that minimize the squared error $\varepsilon(t)^2$ could be found by a gradient descent / error-backpropagation (backprop) method, i.e., by updating them in the directions of $-\partial(\varepsilon(t)^2)/\partial w_j$, $-\partial(\varepsilon(t)^2)/\partial A_{ij}$, and $-\partial(\varepsilon(t)^2)/\partial B_{ik}$. $-\partial(\varepsilon(t)^2)/\partial w_j$ was calculated as follows:

$$\begin{aligned}
&-\partial(\varepsilon(t)^2)/\partial w_j \\
&= -2\varepsilon(t)\partial\varepsilon(t)/\partial w_j \\
&= -2\varepsilon(t)\partial(v_{true}(t) - \boldsymbol{w}^{\mathrm{T}}\, \boldsymbol{x}(t))/\partial w_j \\
&= -2\varepsilon(t)(-\, x_j(t)) \\
&= 2\varepsilon(t)x_j(t) \\
&\approx 2\delta(t)x_j(t) \quad \text{(Eq. 1.5)}
\end{aligned}$$

In the last line, since $\varepsilon(t)$ was unavailable as $v_{true}(t)$ was unknown, it was approximated by the TD-RPE:

$$\delta(t) = r(t) + \gamma v(t+1) - v(t). \quad \text{(Eq. 1.6)}$$

$-\partial(\varepsilon(t)^2)/\partial A_{ij}$ was calculated as follows:

$$-\partial(\varepsilon(t)^2)/\partial A_{ij}$$
$$= -2\varepsilon(t)\partial(v_{true}(t) - \boldsymbol{w}^{\mathrm{T}}\,\boldsymbol{x}(t))/\partial A_{ij}$$
$$\approx 2\delta(t)\partial(\boldsymbol{w}^{\mathrm{T}}\,\boldsymbol{f}(\mathbf{A}\boldsymbol{x}(t-1) + \mathbf{B}\boldsymbol{o}(t-1)))/\partial A_{ij}$$
$$= 2\delta(t)x_j(t-1)(0.5 + x_i(t))(0.5 - x_i(t))w_i \quad \text{(Eq. 1.7)}$$

Similarly, $-\partial(\varepsilon(t)^2)/\partial B_{ik}$ was calculated as follows:

$$-\partial(\varepsilon(t)^2)/\partial B_{ik}$$
$$\approx 2\delta(t)o_k(t-1)(0.5 + x_i(t))(0.5 - x_i(t))w_i \quad \text{(Eq. 1.8)}$$

According to these, the online update rule for the value-RNN was determined as follows:

$$w_j \leftarrow w_j + a_{\mathrm{value}}\delta(t)x_j(t) \quad \text{(Eq. 1.9)}$$
$$A_{ij} \leftarrow A_{ij} + a_{\mathrm{RNN}}\delta(t)x_j(t-1)(0.5 + x_i(t))(0.5 - x_i(t))w_i \quad \text{(Eq. 1.10)}$$
$$B_{ik} \leftarrow B_{ik} + a_{\mathrm{RNN}}\delta(t)o_k(t-1)(0.5 + x_i(t))(0.5 - x_i(t))w_i, \quad \text{(Eq. 1.11)}$$

where a_{value} and a_{RNN} were the learning rates. In each simulation, the elements of **A** and **B** , as well as the elements of $\boldsymbol{x}$, were initialized to pseudo standard normal random numbers, and the elements of $\boldsymbol{w}$ were initialized to 0.

Online value-RNN with fixed random feedback (oVRNNrf)

We considered an implementation of the online value-RNN described above in the cortico-basal ganglia-DA system (Fig. 1):

$\boldsymbol{x}$: activities of neurons in a cortical region with rich recurrent connections
A : recurrent connection strengths among $\boldsymbol{x}$
$\boldsymbol{o}$: activities of neurons in a cortical region processing sensory inputs
B : feed-forward connection strengths from $\boldsymbol{o}$ to $\boldsymbol{x}$
f : sigmoidal relationship from the input to the output of the cortical neurons
$\boldsymbol{w}$: connection strengths from cortical neurons $\boldsymbol{x}$ to a group of striatal neurons
v : activity of the group of striatal neurons
δ : activity of a group of DA neurons / released DA

The update rule for $\boldsymbol{w}$

$$w_j \leftarrow w_j + a_{\mathrm{value}}\delta(t)x_j(t) \quad \text{(Eq. 1.9)}$$

could be naturally implemented as cortico-striatal synaptic plasticity, which depends on DA ($\delta(t)$) and pre-synaptic (cortical) neuronal activity ($x_j(t)$). However, an issue emerged in implementation of the update rules for **A** and **B**:

$$A_{ij} \leftarrow A_{ij} + a_{\mathrm{RNN}}\delta(t)x_j(t-1)(0.5 + x_i(t))(0.5 - x_i(t))w_i \quad \text{(Eq. 1.10)}$$
$$B_{ik} \leftarrow B_{ik} + a_{\mathrm{RNN}}\delta(t)o_k(t-1)(0.5 + x_i(t))(0.5 - x_i(t))w_i, \quad \text{(Eq. 1.11)}$$

Specifically, w_i included in the rightmost of these update rules (for the strengths of cortico-cortical synapses A_{ij} and B_{ik}) is the connection strength from cortical neuron x_i to striatal neurons, i.e., the strength of the cortico-striatal synapses (located within the striatum), which is considered to be

unavailable at the cortico-cortical synapses (located within the cortex).

As mentioned in the Introduction, this is an example of the long-standing difficulty in biological implementation of backprop, and recently a potential solution for this difficulty, i.e., replacement of the downstream connection strengths in the update rule for upstream connections with fixed random strengths, has been demonstrated in supervised learning of feed-forward and recurrent networks [34, 36, 45]. The online value-RNN, which we considered here, differed from supervised learning considered in these previous studies in two ways: i) it was TD learning, apparent in the approximation of the true error $\varepsilon(t)$ by the TD-RPE $\delta(t)$ in the derivation described above, and ii) it used a scalar error (TD-RPE) rather than a vector error. But we expected that the feedback alignment mechanism could still work at least to some extent, and explored it in this study. Specifically, we examined a modified online value-RNN with fixed random feedback (oVRNNrf), in which the update rules for **A** and **B** were modified as follows:

$$A_{ij} \leftarrow A_{ij} + a_{\mathrm{RNN}}\delta(t)x_j(t-1)(0.5 + x_i(t))(0.5 - x_i(t))c_i \quad \text{(Eq. 2.1)}$$

$$B_{ik} \leftarrow B_{ik} + a_{\mathrm{RNN}}\delta(t)o_k(t-1)(0.5 + x_i(t))(0.5 - x_i(t))c_i, \quad \text{(Eq. 2.2)}$$

where w_i in the update rules of the online value-RNN with backprop (oVRNNbp) was replaced with a fixed random parameter c_i. Notably, these modified update rules for the cortico-cortical connections **A** and **B** required only pre-synaptic activities ($x_j(t-1)$, $o_k(t-1)$), post-synaptic activities ($x_i(t)$), TD-RPE-representing DA ($\delta(t)$), and fixed random strengths (c_i), which would all be available at the cortico-cortical synapses given that VTA DA neurons project not only to the striatum but also to the cortex and random c_i could be provided by intrinsic heterogeneity. In each simulation, the elements of $\boldsymbol{c}$ were initialized to pseudo standard normal random numbers.

Revised online value-RNN models with further biological constraints

In the later part of this study, we examined revised online value-RNN models with further biological constraints. Specifically, we considered models, in which the value weights and the activities of neurons in the RNN were constrained to be non-negative. In order to do so, the update rule for $\boldsymbol{w}$ was modified to:

$$w_j \leftarrow \max(0, w_j + a_{\mathrm{value}}\delta(t)x_j(t)), \quad \text{(Eq. 3.1)}$$

where $\max(q_1, q_2)$ returned the maximum of q_1 and q_2. Also, the sigmoidal input-output function was replaced with

$$f(z) = 1/(1 + \exp(-z)), \quad \text{(Eq. 3.2)}$$

and the elements of $\boldsymbol{x}$ were initialized to pseudo uniform [0 1] random numbers. The backprop-based update rules for **A** and **B** in oVRNNbp were replaced with

$$A_{ij} \leftarrow A_{ij} + a_{\mathrm{RNN}}\delta(t)x_j(t-1)x_i(t)(1 - x_i(t))w_i \quad \text{(Eq. 3.3)}$$

$$B_{ik} \leftarrow B_{ik} + a_{\mathrm{RNN}}\delta(t)o_k(t-1)x_i(t)(1 - x_i(t))w_i. \quad \text{(Eq. 3.4)}$$

We referred to the model with these modifications to oVRNNbp as oVRNNbp-rev.

As a revised online value-RNN with fixed random feedback (oVRNNrf), in addition to the abovementioned modifications of the update of $\boldsymbol{w}$, the sigmoidal input-output function, and the initialization of $\boldsymbol{x}$, the fixed random feedback $\boldsymbol{c}$ was assumed to be non-negative. Specifically, the elements of $\boldsymbol{c}$ were set to pseudo uniform [0 1] random numbers. Moreover, the update rules for $\mathbf{A}$ and $\mathbf{B}$ were replaced with

(when $x_i(t) \leq 0.5$)

$$A_{ij} \leftarrow A_{ij} + a_{\mathrm{RNN}}\delta(t)x_j(t-1)x_i(t)(1 - x_i(t))c_i \quad \text{(Eq. 3.5)}$$

$$B_{ik} \leftarrow B_{ik} + a_{\mathrm{RNN}}\delta(t)o_k(t-1)x_i(t)(1 - x_i(t))c_i. \quad \text{(Eq. 3.6)}$$

(when $x_i(t) > 0.5$)

$$A_{ij} \leftarrow A_{ij} + 0.25a_{\mathrm{RNN}}\delta(t)x_j(t-1)c_i \quad \text{(Eq. 3.7)}$$

$$B_{ik} \leftarrow B_{ik} + 0.25a_{\mathrm{RNN}}\delta(t)o_k(t-1)c_i. \quad \text{(Eq. 3.8)}$$

so that the originally non-monotonic dependence on $x_i(t)$ (post-synaptic activity) became monotonic + saturation (Fig. 5B). These update rules with non-negative c_i could be said to be Hebbian with additional modulation by TD-RPE (Hebbian under positive TD-RPE). We referred to the model with these modifications to oVRNNrf as oVRNNrf-bio. In the right panels of Fig. 6E,F, we also examined the model where the modified update rules of oVRNNrf-bio were changed back to the original ones, referred to as oVRNNrf-rev. In some simulations in Fig. 8E-G, we examined a modified oVRNNrf-bio with a slight decay (forgetting) of value weights, in which each element of $\boldsymbol{w}$ decayed at every time-step:

$$w_i \leftarrow (1 - dr)w_i, \quad \text{(Eq. 3.9)}$$

where dr was the decay rate per time-step and was set to 0.001 or 0.002.

We further examined extensions of oVRNNbp-rev and oVRNNrf-bio, referred to as oVRNNbp-rev-ei and oVRNNrf-bio-ei, which incorporated excitatory E-units and inhibitory I-units (Fig. 9A). Based on biological suggestions (see the Results), we made the following assumptions. Each E-unit received inputs from the observation units $\boldsymbol{o}$ (connections: $\mathbf{B_E}$), all the E-units (connections: $\mathbf{A_E}$), and a particular I-unit (with a strength h), and projected to the striatal value unit (connections: $\boldsymbol{w}$). Each I-unit received inputs from the observation units $\boldsymbol{o}$ (connections: $\mathbf{B_I}$) and all the E-units (connections: $\mathbf{A_I}$). Excitation from the observation units and E-units to E- and I-units took one time-step whereas I→E inhibition operated within a time-step. The activation function for E-unit and plasticity rules for connections from/to E-units were the same as those for the RNN unit in the original models. I-unit had a linear activation function, and there was no plasticity for connections from/to I-units. Update rule for $\boldsymbol{w}$ was the same as the original one with the activity of the RNN units replaced with the activity of E-units. Equations for the activities of E-units and I-units, $\boldsymbol{x_E}$ and $\boldsymbol{x_I}$, are given as follows:

$$\boldsymbol{x_I}(t+1) = \mathbf{A_I}\boldsymbol{x_E}(t) + \mathbf{B_I}\boldsymbol{o}(t) \quad \text{(Eq. 3.10)}$$

$$\boldsymbol{x_E}(t+1) = \boldsymbol{f}(\mathbf{A_E}\boldsymbol{x}(t) + \mathbf{B_E}\boldsymbol{o}(t) - h\boldsymbol{x_I}(t+1)), \quad \text{(Eq. 3.11)}$$

where h was set to 1. The elements of $\mathbf{A_I}$, $\mathbf{B_I}$, $\mathbf{A_E}$, and $\mathbf{B_E}$ were initialized to be non-negative:

$$\max(0, 3 + z),$$

where z was a pseudo standard normal random number. The elements of $\boldsymbol{x_E}$ were initialized to pseudo

uniform [0 1] random numbers, and the initial values of $\boldsymbol{x}_{\mathbf{I}}$ were determined according to the abovementioned equation.

Incorporation of action selection

We considered extensions of oVRNNbp-rev and oVRNNrf-bio that incorporated an actor-critic architecture, referred to as oVRNNbp-rev-ac and oVRNNrf-bio-ac (Fig. 11A). Each RNN unit additionally connected to additional two units representing the action-values of action 1 and action 2 (q_1 and q_2):

$$\boldsymbol{q}(t) = \mathbf{U}\boldsymbol{x}(t), \quad \text{(Eq. 4.1)}$$

where $\mathbf{U} = (u_{kj})$ consisted of two row vectors that represented the preferences of the two actions. At the time-step next to cue-presentation, one action was selected in a soft-max manner based on the action values. Specifically, action k was selected with the probability of

$$\exp(\beta q_k)/(\exp(\beta q_1) + \exp(\beta q_2)), \quad \text{(Eq. 4.2)}$$

where β was the inverse temperature parameter, set to 1 or 2, representing the degree of exploitation over exploration. Selected action was then informed to the RNN units. Specifically, the observation layer had two additional elements ($\boldsymbol{o}'$ in Fig. 11A) corresponding to the two actions. These elements became 1 when the corresponding action was selected and 0 otherwise. The preference of selected action k was updated by using the TD-RPE $\delta(t)$ as follows:

$$u_{kj} \leftarrow u_{kj} + a_{\text{pref}}\delta(t)x_j(t), \quad \text{(Eq. 4.3)}$$

where a_{pref} was the learning rate. In order to prevent unbounded increase of action preference, we assumed a slight decay of all the action preferences at every time-step:

$$u_{kj} \leftarrow (1 - dr)u_{kj}, \quad \text{(Eq. 4.4)}$$

where dr was the decay rate per time-step and was set to 0.001. u_{kj} (i.e., the elements of $\mathbf{U}$) were initialized to 0. The connection weights from the action-observation units $\boldsymbol{o}'$, as well as from $\boldsymbol{o}$ and the RNN units, onto the RNN units were initialized to pseudo standard normal random numbers.

Simulation of the tasks

In the Pavlovian cue-reward association task, at time 1 of each trial, cue observation was received by the RNN, and at time 4, reward observation was received. Trial was pseudo-randomly ended at time 7, 8, 9, or 10, and the next trial started from the next time-step (i.e., inter-trial interval (ITI) was 4, 5, 6, or 7 time-steps with equal probabilities). Reward size was $r = 1$. We also conducted simulations with longer cue-reward delays, in which reward was given at time 5, 6, or 7, and the end of trial was shifted accordingly. The tasks with probabilistic structures (task 1 and task 2) were implemented in the same way except that reward timing was not time 4 but time 3 or 5 with equal probabilities, specifically, 50% and 50% in task 1 and 30% and 30% in task 2, and there was no reward in the remaining 40% of trials in task 2. The tasks with action selection were also

implemented in the same way except that size 2 reward was received, i.e., the reward term in the TD-RPE calculation as well as the reward-corresponding element of the observation inputs were set to 2, at time 4 when action 1 was selected whereas size 1 reward was received at time 4 (in the first choice task) or time 3 (in the second choice task) when action 2 was selected.

The cue or reward state/timing, mentioned in the text and marked in the figures, was defined to be the timing when the RNN received the cue or reward observation, respectively. Specifically, if $\boldsymbol{o}(t) = (1\ 0)^{\mathrm{T}}$ or $\boldsymbol{o}(t) = (0\ 1)^{\mathrm{T}}$ at time t, $t + 1$ was defined to be a cue or reward timing, respectively. For the agents with punctate state representation, which is also referred to as the complete serial compound (CSC) representation [1, 48, 133], each timing from a cue in the tasks was represented by a 10-dimensional one-hot vector, starting from $(1\ 0\ 0\ ...\ 0)^{\mathrm{T}}$ for the cue state, with the next state $(0\ 1\ 0\ ...\ 0)^{\mathrm{T}}$ and so on.

In the simulations of the cue-reward association task with distractor cue, the observation units $\boldsymbol{o} = (o_k)$ had an additional element o_3 (Fig. 10A), which was 1 at the time-steps where distractor cue was present and 0 otherwise. We examined four cases, in which the probability of the presence of distractor cue at every time-step throughout the task was 0, 0.1, 0.2, and 0.3 (Fig. 10B-E).

Learning rates were set as follows. For the models with punctate state representation, $a_{\mathrm{value}} = 0.1$. For oVRNNbp, oVRNNrf, and the model with untrained RNN compared with these two, $a_{\mathrm{RNN}} = 0.1$ and $a_{\mathrm{value}} = 0.1/(n/7)$. For oVRNNbp-rev, oVRNNrf-bio, and the models with untrained RNN compared with these two, $a_{\mathrm{RNN}} = 0.1$ and $a_{\mathrm{value}} = a_{\mathrm{pref}} = 0.1/(n/12)$ except for the right panels of Fig. 6J, and $a_{\mathrm{RNN}} = 0.05$ and $a_{\mathrm{value}} = 0.05/(n/12)$ for the right panels of Fig. 6J. Time discount factor (γ) was set to 0.8.

Estimation of true state/timing values

As for the Pavlovian cue-reward association task, we defined states after agent's receival of cue information by relative timings from the cue, and estimated their (true) values by simulations according to the definition of state value. We generated a sequence of cues and rewards corresponding to 1000 trials with the ITI after the first trial, ITI_1, fixed to one of the possible lengths (4, 5, 6, or 7 time-steps), and calculated cumulative discounted future rewards within the sequence:

$$\Sigma_{t_\mathrm{rew}}(r\gamma^{t_\mathrm{rew}}),$$

where t_rew denotes the time-step of each reward counted from the starting state, starting from +1, ..., and $+3 + \mathrm{ITI}_1$ time steps from a cue (the last one corresponded to the cue timing of the next trial). For each case where $\mathrm{ITI}_1 = 4$, 5, 6, or 7, we repeated this 1000 times, generating 1000 sequences (i.e., 1000 simulations of 1000 trials), with different sets of pseudo-random numbers, and calculated the average over these 1000 sequences (we refer to these as ITI_1-specific values). We estimated the value of each state of +1, ..., and +7 time steps from cue (i.e., −2, ..., +4 time-steps from reward) by taking the average of the ITI_1-specific values for four possible ITI_1.

We also estimated the true values of the cue timing and one and two timing(s) before it in the following way; these values could not be estimated in the abovementioned way because agent should not know the length of ITI (i.e., when ITI ends) until receiving cue information at the cue timing. In the case where ITI is in fact 4 time-steps, until receiving the next cue, agent should think that ITI can be 4, 5, 6, or 7 time-steps with equal probabilities (1/4 for each). Thus, the value of next cue timing and one and two timing(s) before it should be the average of the four ITI_1-specific values of +4, +3, and +2 time-steps from reward. Similarly, in the case where ITI is in fact 5 time-steps, until the previous time-step of the next cue, agent should think that ITI can be 4, 5, 6, or 7 time-steps with equal probabilities (1/4 for each). Thus, the value of one and two timing(s) before next cue should be the average of the four ITI_1-specific values of +4 and +3 time-steps from reward. On the other hand, at the timing of the next cue, agent should think that ITI can be 5, 6, or 7 (but not 4) time-steps with equal probabilities (1/3 for each). Thus, the value of next cue timing should be the average of the three ITI_1-specific values (for ITI_1 = 5, 6, or 7) of +5 time-steps from reward. Similar considerations can be made for the cases where ITI is in fact 6 or 7 time-steps. And then, the "true" value of (next) cue timing can be calculated as the average of the values of next cue timing in the cases where ITI is in fact 4, 5, 6, or 7 time-steps. Using these estimated true state values, we calculated TD-RPE at each state/timing (−2, −1, ..., and +5 time steps from cue). True state/timing values in the cases where the cue-reward delay was 4, 5, or 6 time-steps were estimated in the same way.

We also estimated true state/timing values for tasks 1 and 2 that had probabilistic structures. As for task 1, we first estimated the values of each timing in each of the trial types (Fig. 4Ba, left), in which reward was given at early (2 time-steps after cue) or late (4 time-steps after cue) timing, in the same manner (but using 10000 rather than 1000 simulations for each condition) as done for the cue-reward association task mentioned above (values of the cue timing and the one and two timing(s) before cue after each trial type were also estimated). Then, based on the agent's belief about trial types (Fig. 4Bb, left), we defined the following states: +1 and +2 time steps from cue (i.e., states visited (entered) before knowing whether reward was given at the early timing (= +2 time step from cue)), +3, 4, 5, and 6 time steps from cue after reception of reward at the early timing, and +3, 4, 5, and 6 time steps from cue after no reception of reward at the early timing (Fig. 4Bc, left-top). We calculated the true values of these states and also of the cue timing and one and two timing(s) before cue (Fig. 4Bc, left-bottom) by taking (mathematical) expected value of the abovementioned estimated value of each timing in each trial type. Using these true values, we calculated TD-RPE (Fig. 4C, left).

As for task 2, we first estimated the values of each timing in each of the trial types (Fig. 4Ba, right), in which reward was given at early (2 time-steps after cue) or late (4 time-steps after cue) timing or was not given. Then, based on the agent's belief about trial types (Fig. 4Bb, right), we defined the following states: +1 and +2 time steps from cue (i.e., states visited (entered) before knowing whether reward was given at the early timing), +3, 4, 5, and 6 time steps from cue after

reception of reward at the early timing, +3 and 4 time steps from cue after no reception of reward at the early timing (states visited (entered) before knowing whether reward was given at the late timing (= +4 time step from cue)), +5 and 6 time steps from cue after reception of reward at the late timing, and +5 and 6 time steps from cue after no reception of reward at both early and late timings (Fig. 4Bc, right-top). We estimated the true values of these states and also of the cue timing and one and two timing(s) before cue (Fig. 4Bc, right-bottom) in the same manner as for task 1, and using these true values, we calculated TD-RPE (Fig. 4C, right).

Analyses, software, and code availability

SEM (Standard error of the mean) was approximated by SD (standard deviation)/$\sqrt{N}$ (number of samples). Cohen's *d* using an average variance was calculated as (difference in the means) / (square root of the average of variances). Linear regression, principal component analysis (PCA), Wilcoxon rank sum test, and *t*-tests were conducted by using R (functions lm, prcomp, wilcox.exact (in package exactRankTests), and t.test). Difference in Wilcoxon rank sum test and *t*-tests was reported when $p < 0.05$. Simulations were conducted by using MATLAB, and pseudo-random numbers were implemented by using rand, randn, and randperm functions. The codes for simulations and analyses are available at GitHub (https://github.com/kenjimoritagithub/oVRNN1).

Figure legends

Figure 1

Implementation of the online value-RNN in the cortico-basal ganglia-DA circuits.

Figure 2

Simulation of a Pavlovian cue-reward association task. **(A)** Simulated task with variable inter-trial intervals. **(B)** Black line: Estimated true values of states/timings through simulations according to the definition of state value, i.e., expected cumulative discounted future rewards, taking into account the effect of probabilistic inter-trial interval (ITI). Red line: TD-RPEs calculated from the estimated true state/timing values. **(C-G)** State values (black lines) and TD-RPEs (red lines) at 1000-th trial, averaged across 100 simulations (error-bars indicating mean ± SEM across simulations; same applied to the followings unless otherwise mentioned), in different types of agent: (C) TD-RL agent having punctate state representation and state values without continuation between trials (i.e., the value of the last state in a trial was not updated by TD-RPE upon entering the next trial); (D) TD-RL agent having punctate state representation and continuously updated state values across trials; (E) Online value-RNN with backprop (oVRNNbp). The number of RNN units was 7 (same applied to (F,G); (F) Online value-RNN with fixed random feedback (oVRNNrf); (G) Agent with untrained RNN. **(H)** State values at 1000-th trial in individual simulations of oVRNNbp (top), oVRNNrf (middle), and untrained RNN (bottom). **(I)** Histograms of the value of the pre-reward state (i.e., the state one-time step before the reward state) at 1000-th trial in individual simulations of the three models. The vertical black dashed lines indicate the true value of the pre-reward state (estimated through simulations). **(J)** *Left*: Mean of the squares of differences between the state values developed by each agent and the estimated true state values between cue and reward (referred to as the mean squared value-error) at 1000-th trial in oVRNNbp (red line), oVRNNrf (blue line), and the model with untrained RNN (gray line) when the number of RNN units (n) was varied from 5 to 40. Learning rate for value weights was normalized by dividing by $n/7$ (same applied to the followings unless otherwise mentioned). *Right*: Mean squared value-error in oVRNNrf (blue line: same data as in the left panel) and oVRNN with uniform feedback (green line). **(K)** Log contribution ratios of the principal components of the time series (for 1000 trials) of RNN activities in each model with 20 RNN units. **(L)** Mean squared value-error in each model with 20 RNN units across trials. **(M)** Mean squared value-error in each model at 3000-th trial in the cases where the cue-reward delay was 3, 4, 5, or 6 time-steps (top to bottom panels).

Figure 3

Occurrence of feedback alignment and an intuitive understanding of its mechanism. **(A)** Over-trial changes in the angle between the value-weight vector $\boldsymbol{w}$ and the fixed random feedback vector $\boldsymbol{c}$ in

the simulations of oVRNNrf (7 RNN units). The solid line and the dashed lines indicate the mean and ± SD across 100 simulations, respectively. **(B)** Negative correlation (r = −0.288, p = 0.00362) between the angle between $\boldsymbol{w}$ and $\boldsymbol{c}$ (horizontal axis) and the value of the pre-reward state (vertical axis) at 1000-th trial. The dots indicate the results of individual simulations, and the line indicates the regression line. **(C)** Angle between the hypothetical change in $\boldsymbol{x}(t) = \boldsymbol{f}(\mathbf{A}\boldsymbol{x}(t-1),\mathbf{B}\boldsymbol{o}(t-1))$ in case **A** and **B** were replaced with their updated ones, multiplied with the sign of TD-RPE (sign($\delta(t)$)), and the fixed random feedback vector $\boldsymbol{c}$ across time-steps. The black thick line and the gray lines indicate the mean and ± SD across 100 simulations, respectively (same applied to (D)). **(D)** Multiplication of TD-RPEs in successive trials at individual states (top: cue, 4th from the top: reward). Positive or negative value indicates that TD-RPEs in successive trials have the same or different signs, respectively. **(E)** *Left*: RNN trajectories mapped onto the primary and secondary principal components (horizontal and vertical axes, respectively) in three successive trials (red, blue, and green lines (heavily overlapped)) at different phases in an example simulation (10th-12th, 300th-302th, 600th-602th, and 900th-902th trials from top to bottom). The crosses and circles indicate the cue and reward states, respectively. *Right*: State values (black lines) and TD-RPEs (red lines) at 11th, 301th, 601th, and 901th trial.

Figure 4

Simulation of two tasks having probabilistic structures, which were qualitatively similar to the two tasks examined in experiments [54] and modeled by the original value-RNN with BPTT [26]. **(A)** Simulated two tasks, in which reward was given at the early or the late timing with equal probabilities in all the trials (task 1) or 60% of trials (task 2). **(B) (a)** *Top*: Trial types. Two trial types (with early reward and with late reward) in task 1 and three trial types (with early reward, with late reward, and without reward) in task 2. *Bottom*: Value of each timing in each trial type estimated through simulations. **(b)** Agent's probabilistic belief about the current trial type, in the case where agent was in fact in the trial with early reward (top row), the trial with late reward (second row), or the trial without reward (third row in task 2). **(c)** *Top*: States defined by considering the probabilistic beliefs at each timing from cue. *Bottom*: True state/timing values calculated by taking (mathematical) expected value of the estimated value of each timing in each trial type. **(C)** Expected TD-RPE calculated from the estimated true values of the states/timings for task 1 (left) and task 2 (right). Red lines: case where reward was given at the early timing, blue lines: case where reward was given at the late timing. It is expected that TD-RPE at early reward is larger than TD-RPE at late reward in task 1 whereas the opposite is the case in task 2, as indicated by the inequality sings. **(D-H)** TD-RPEs at the latest trial within 1000 trials in which reward was given at the early timing (red lines) or the late timing (blue lines), averaged across 100 simulations (error-bars indicating ± SEM across simulations), in the different types of agent: (D,E) TD-RL agent having punctate state representation and state values without (D) or with (E) continuation between trials; (F) oVRNNbp. The number of RNN units was 12 (same applied to (G,H)); (G) oVRNNrf; (H) agent with untrained RNN. The p

values are for paired *t*-test between TD-RPE at early reward and TD-RPE at late reward (100 pairs, two-tailed), and the *d* values are Cohen's *d* using an average variance and their signs are with respect to the expected patterns shown in (C) (same applied to Fig. 8 and Fig. 9C).

Figure 5

Revised online value-RNN models with further biological constraints. **(A)** oVRNNbp-rev: oVRNNbp (online value-RNN with backprop) was modified so that the activities of neurons in the RNN (***x***) and the value weights (***w***) became non-negative. **(B)** oVRNNrf-bio: oVRNNrf (online value-RNN with fixed random feedback) was modified so that ***x*** and ***w***, as well as the fixed random feedback (***c***), became non-negative and also the dependence of the update rules for recurrent/feed-forward connections (**A** and **B**) on post-synaptic activity became monotonic + saturation.

Figure 6

Performances of the revised online value-RNN models in the cue-reward association task, in comparison with models with untrained RNN that also had the non-negative constraint. **(A-D)** State values (black lines) and TD-RPEs (red lines) at 1500-th trial in oVRNNbp-rev (A), oVRNNrf-bio (B), agent with naive untrained RNN (i.e., randomly initialized RNN) with ***x*** and ***w*** constrained to be non-negative (C), and agent with untrained RNN with connections shuffled from those learnt in oVRNNrf-bio and also with non-negative ***x*** and ***w*** (D). The number of RNN units was 12 in all the cases. Error-bars indicate mean ± SEM across 100 simulations; same applied to the followings unless otherwise mentioned. The right histograms show the across-simulation distribution of the value of the pre-reward state in each model. The vertical black dashed lines in the histograms indicate the true value of the pre-reward state (estimated through simulations). **(E)** *Left*: Mean squared value-error at 1500-th trial in oVRNNbp-rev (red line), oVRNNrf-bio (blue line), agent with naive untrained RNN (gray solid line: partly out of view), and agent with shuffled untrained RNN (gray dotted line) when the number of RNN units (*n*) was varied from 5 to 40. Learning rate for value weights was normalized by dividing by *n*/12 (same applied to the followings). *Right*: Mean squared value-error in oVRNNrf-bio (blue line: same data as in the left panel), oVRNN-bio with random-magnitude uniform feedback (green line), oVRNN-bio with fixed-magnitude (0.5) uniform feedback (light blue line), and oVRNNrf-rev where the update rule of oVRNNrf-bio was changed back to the original one (blue dotted line). **(F)** *Left*: Mean of the elements of the recurrent and feed-forward connections (at 1500-th trial) of oVRNNbp-rev (red line), oVRNNrf-bio (blue line), and naive untrained RNN (gray solid line). *Right*: Mean of the elements of the recurrent and feed-forward connections of oVRNNrf-bio (blue line: same data as in the left panel), oVRNN-bio with random-magnitude uniform feedback (green line), oVRNN-bio with fixed-magnitude (0.5) uniform feedback (light blue line), and oVRNNrf-rev (blue dotted line). **(G)** Learned state values (left panel) and TD-RPEs (right panel) in oVRNNbp-rev (red lines) and oVRNNrf-bio (blue lines) in the cases with 40 RNN units, compared to the estimated true values (black lines). **(H)** Log of contribution ratios of the principal components

of the time series (for 1500 trials) of RNN activities in each model with 20 RNN units. **(I)** Mean squared value-error in each model with 20 RNN units across trials. **(J)** Mean squared value-error at 3000-th trial in each model in the cases where the cue-reward delay was 3, 4, 5, or 6 time-steps (top to bottom panels). Left and right panels show the results with default learning rates and halved learning rates, respectively.

Figure 7

Performances of the revised online value-RNN models with further biological constraints in the two tasks having probabilistic structures, in comparison with models with untrained RNN. TD-RPEs at the latest trial within 2000 trials in which reward was given at the early timing (red lines) or the late timing (blue lines) in task 1 (left) and task 2 (right), averaged across 100 simulations (error-bars indicating ± SEM across simulations), are shown for the four types of agent: (A) oVRNNbp-rev; (B) oVRNNrf-bio; (C) agent with naive untrained RNN; (D) agent with untrained RNN with connections shuffled from those learnt in oVRNNrf-bio. The number of RNN units was 20 for all the cases.

Figure 8

Loose alignment of the value weights (***w***) and the random feedback (***c***) in oVRNNrf-bio (with 12 RNN units). **(A)** Over-trial changes in the angle between the value weights ***w*** and the fixed random feedback ***c***. The solid line and the dashed lines indicate the mean and ± SD across 100 simulations, respectively. **(B)** No correlation between the ***w***-***c*** angle (horizontal axis) and the value of the pre-reward state (vertical axis) at 1500-th trial (r = 0.0117, p = 0.908). The dots indicate the results of individual simulations. **(C)** Correlation between the ***w***-***c*** angle at k-th trial (horizontal axis) and the value of the cue, post-cue, pre-reward, or reward state (top-bottom panels) at 500-th trial across 1000 simulations. The solid lines indicate the correlation coefficient, and the short vertical bars at the top of each panel indicates the cases in which p-value was less than 0.05. **(D)** Distribution of the angle between two 12-dimensional vectors when the elements of both vectors were drawn from [0 1] uniform pseudo-random numbers (a) or when one of the vectors was replaced with [1 0 0 ... 0] (i.e., on the edge of the non-negative quadrant) (b) or [1 1 0 ... 0] (i.e., on the boundary of the non-negative quadrant) (c). **(E)** Across-simulations histograms of elements of ***w*** in oVRNNrf-bio with 12 RNN units ordered from the largest to smallest ones after 1500 trials when there was no value-weight-decay (a) or there was value-weight-decay with decay rate (per time-step) of 0.001 (b) or 0.002 (c). The error-bars indicate the mean ± SEM across 100 simulations. **(F)** Over-trial changes in the angle between the value weights ***w*** and the fixed random feedback ***c*** when there was value-weight-decay with decay rate (per time-step) of 0.001 (top panel) or 0.002 (bottom panel). Notations are the same those in (A). **(G)** Mean squared value-error at 1500-th trial in oVRNNrf-bio with 12 RNN units with the rate of value-weight-decay was varied (horizontal axis). The error-bars indicate the mean ± SEM across 100 simulations.

Figure 9

oVRNNbp-rev-ei and oVRNNrf-bio-ei models incorporating excitatory E-units and inhibitory I-units. **(A)** Schematic illustration of the models' architecture. For ease of viewing, only limited parts of units and connections are drawn. **(B)** Mean squared value-error at 1500-th trial in the cue-reward association task in oVRNNbp-rev-ei (red line), oVRNNrf-bio-ei (blue line), and E-/I-units-incorporated models with naive untrained RNN (i.e., randomly initialized RNN) (gray solid line) or untrained RNN with connections shuffled from those learnt in oVRNNrf-bio-ei (gray dotted line). **(C)** Patterns of TD-RPE in the tasks with probabilistic structures generated in the four models with E-/I-units. Simulation conditions and notations are the same those in Fig. 7.

Figure 10

Cue-reward association task with distractor cue. **(A)** Modification of oVRNNbp-rev and oVRNNrf-bio to incorporate possible existence of distractor cue. The observation units ***o*** had an additional element (leftmost circle labeled as 'Dist'), which was 1 at the time-steps where distractor cue was present and 0 otherwise. **(B-E)** Results of the cases where the probability of the presence of distractor cue at every time-step was 0 (B), 0.1 (C), 0.2 (D), and 0.3 (E). *Left panels*: Examples of the presence of distractor ('D'), reward-associated cue ('C'), and reward ('R') over 100 time-steps. *Middle panels*: Mean squared value-error at 1500-th trial in oVRNNbp-rev (red line), oVRNNrf-bio (blue line), and the models with naive or shuffled untrained RNN (gray solid or dotted line). *Right panels*: Results for the models with E- and I-units (oVRNNbp-rev-ei: red line, oVRNNrf-bio-ei: blue line, models with naive or shuffled untrained RNN: gray solid or dotted line), which were modified to incorporate possible existence of distractor cue in the same manner as in (A).

Figure 11

Incorporation of action selection. **(A)** Schematic illustration of the models incorporating an actor-critic architecture. **(B)** Two-alternative choice task. **(a)** Task diagram. **(b,c)** Proportion of Action 1 selection in 2901~3000-th trials in oVRNNbp-rev-as (red line), oVRNNrf-bio-as (blue line), and the models with naive untrained RNN (i.e., randomly initialized RNN) (gray solid line) or untrained RNN with connections shuffled from those learnt in VRNNrf-bio-as (gray dotted line). Error-bars indicate the mean ± SEM over 100 simulations. The inverse temperature was set to 1 (b) or 2 (c). **(C)** Inter-temporal choice task. The notations are the same as those in (B).

References

1. Montague, P.R., Dayan, P. & Sejnowski, T.J. A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *J Neurosci* **16**, 1936-1947 (1996).
2. Schultz, W., Dayan, P. & Montague, P.R. A neural substrate of prediction and reward. *Science* **275**, 1593-1599 (1997).
3. Niv, Y. & Schoenbaum, G. Dialogues on prediction errors. *Trends Cogn Sci* **12**, 265-272 (2008).
4. Cohen, J.Y., Haesler, S., Vong, L., Lowell, B.B. & Uchida, N. Neuron-type-specific signals for reward and punishment in the ventral tegmental area. *Nature* **482**, 85-88 (2012).
5. Steinberg, E.E., *et al.* A causal link between prediction errors, dopamine neurons and learning. *Nat Neurosci* **16**, 966-973 (2013).
6. Kim, H.R., *et al.* A Unified Framework for Dopamine Signals across Timescales. *Cell* **183**, 1600-1616.e1625 (2020).
7. Reynolds, J.N., Hyland, B.I. & Wickens, J.R. A cellular mechanism of reward-related learning. *Nature* **413**, 67-70 (2001).
8. Shen, W., Flajolet, M., Greengard, P. & Surmeier, D.J. Dichotomous dopaminergic control of striatal synaptic plasticity. *Science* **321**, 848-851 (2008).
9. Yagishita, S., *et al.* A critical time window for dopamine actions on the structural plasticity of dendritic spines. *Science* **345**, 1616-1620 (2014).
10. Russek, E.M., Momennejad, I., Botvinick, M.M., Gershman, S.J. & Daw, N.D. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS Comput Biol* **13**, e1005768 (2017).
11. Stachenfeld, K.L., Botvinick, M.M. & Gershman, S.J. The hippocampus as a predictive map. *Nat Neurosci* **20**, 1643-1653 (2017).
12. Qian, L., *et al.* Prospective contingency explains behavior and dopamine signals during associative learning. *Nat Neurosci* **10.1038/s41593-025-01915-4** (2025).
13. Langdon, A.J., Sharpe, M.J., Schoenbaum, G. & Niv, Y. Model-based predictions for dopamine. *Curr Opin Neurobiol* **49**, 1-7 (2018).
14. Keiflin, R., Pribut, H.J., Shah, N.B. & Janak, P.H. Ventral Tegmental Dopamine Neurons Participate in Reward Identity Predictions. *Curr Biol* **29**, 93-103.e103 (2019).
15. Redish, A.D., Jensen, S., Johnson, A. & Kurth-Nelson, Z. Reconciling reinforcement learning models with behavioral extinction and renewal: implications for addiction, relapse, and problem gambling. *Psychol Rev* **114**, 784-805 (2007).
16. Gershman, S.J., Jones, C.E., Norman, K.A., Monfils, M.H. & Niv, Y. Gradual extinction prevents the return of fear: implications for the discovery of state. *Front Behav Neurosci* **7**, 164 (2013).
17. Shimomura, K., Kato, A. & Morita, K. Rigid reduced successor representation as a potential

mechanism for addiction. *Eur J Neurosci* **53**, 3768-3790 (2021).
18. Feng, Z., Nagase, A.M. & Morita, K. A Reinforcement Learning Approach to Understanding Procrastination: Does Inaccurate Value Approximation Cause Irrational Postponing of a Task? *Front Neurosci* **15**, 660595 (2021).
19. Sato, R., Shimomura, K. & Morita, K. Opponent learning with different representations in the cortico-basal ganglia pathways can develop obsession-compulsion cycle. *PLoS Comput Biol* **19**, e1011206 (2023).
20. Gershman, S.J. & Niv, Y. Learning latent structure: carving nature at its joints. *Curr Opin Neurobiol* **20**, 251-256 (2010).
21. Niv, Y. Learning task-state representations. *Nat Neurosci* **22**, 1544-1553 (2019).
22. George, T.M., de Cothi, W., Stachenfeld, K.L. & Barry, C. Rapid learning of predictive maps with STDP and theta phase precession. *Elife* **12**, e80663 (2023).
23. Bono, J., Zannone, S., Pedrosa, V. & Clopath, C. Learning predictive cognitive maps with spiking neurons during behavior and replays. *Elife* **12**, e80671 (2023).
24. Fang, C., Aronov, D., Abbott, L.F. & Mackevicius, E.L. Neural learning rules for generating flexible predictions and computing the successor representation. *Elife* **12**, e80680 (2023).
25. Cone, I. & Clopath, C. Latent representations in hippocampal network model co-evolve with behavioral exploration of task structure. *Nat Commun* **15**, 687 (2024).
26. Hennig, J.A., *et al.* Emergence of belief-like representations through reinforcement learning. *PLoS Comput Biol* **19**, e1011067 (2023).
27. Rumelhart, D.E., Hinton, G.E. & Williams, R.J. Learning Internal Representations by Error Propagation. in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition Volume 1 Foundations* (ed. D.E. Rumelhart, McClelland, J.L., The PDP Group) 318-362 (MIT Press, Cambridge, 1985).
28. Amari, S. A Theory of Adaptive Pattern Classifiers. *IEEE Transactions on Electronic Computers* **EC-16**, 299-307 (1967).
29. Rumelhart, D.E., Hinton, G.E. & Williams, R.J. Learning representations by back-propagating errors. *Nature* **323**, 533-536 (1986).
30. Grossberg, S. Competitive learning: from interactive activation to adaptive resonance. *Cognitive Science* **11**, 23-63 (1987).
31. Crick, F. The recent excitement about neural networks. *Nature* **337**, 129-132 (1989).
32. Doya, K. Complementary roles of basal ganglia and cerebellum in learning and motor control. *Curr Opin Neurobiol* **10**, 732-739 (2000).
33. O'Doherty, J., *et al.* Dissociable roles of ventral and dorsal striatum in instrumental conditioning. *Science* **304**, 452-454 (2004).
34. Murray, J.M. Local online learning in recurrent networks with random feedback. *Elife* **8**, e43299 (2019).
35. Bellec, G., *et al.* A solution to the learning dilemma for recurrent networks of spiking neurons.

Nat Commun **11**, 3625 (2020).
36. Lillicrap, T.P., Cownden, D., Tweed, D.B. & Akerman, C.J. Random synaptic feedback weights support error backpropagation for deep learning. *Nat Commun* **7**, 13276 (2016).
37. Guerguiev, J., Lillicrap, T.P. & Richards, B.A. Towards deep learning with segregated dendrites. *Elife* **6**, e22901 (2017).
38. Sacramento, J., Costa, R.P., Bengio, Y. & Senn, W. Dendritic cortical microcircuits approximate the backpropagation algorithm. in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)* (2018).
39. Whittington, J.C.R. & Bogacz, R. Theories of Error Back-Propagation in the Brain. *Trends Cogn Sci* **23**, 235-250 (2019).
40. Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J. & Hinton, G. Backpropagation and the brain. *Nat Rev Neurosci* **21**, 335-346 (2020).
41. Payeur, A., Guerguiev, J., Zenke, F., Richards, B.A. & Naud, R. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nat Neurosci* **24**, 1010-1019 (2021).
42. Greedy, W., Zhu, H.W., Pemberton, J., Mellor, J. & Costa, R.P. Single-phase deep learning in cortico-cortical networks. in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (2022).
43. Song, Y., *et al.* Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nat Neurosci* **27**, 348-358 (2024).
44. Pagkalos, M., Makarov, R. & Poirazi, P. Leveraging dendritic properties to advance machine learning and neuro-inspired computing. *Curr Opin Neurobiol* **85**, 102853 (2024).
45. Wärnberg, E. & Kumar, A. Feasibility of dopamine as a vector-valued feedback signal in the basal ganglia. *Proc Natl Acad Sci U S A* **120**, e2221994120 (2023).
46. Mongillo, G., Barak, O. & Tsodyks, M. Synaptic theory of working memory. *Science* **319**, 1543-1546 (2008).
47. Morishima, M., Morita, K., Kubota, Y. & Kawaguchi, Y. Highly differentiated projection-specific cortical subnetworks. *J Neurosci* **31**, 10380-10391 (2011).
48. Sutton, R.S. & Barto, A.G. *Reinforcement Learning: An Introduction (Second Edition)* (MIT Press, Cambridge, MA, 2018).
49. Rajan, K. & Abbott, L.F. Eigenvalue spectra of random matrices for neural networks. *Phys Rev Lett* **97**, 188104 (2006).
50. Dominey, P.F. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biol Cybern* **73**, 265-274 (1995).
51. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput* **14**, 2531-2560 (2002).
52. Jaeger, H. Echo state network. *Scholarpedia* **2**, 2330 (2007).
53. Tanaka, G., *et al.* Recent advances in physical reservoir computing: A review. *Neural Netw* **115**, 100-123 (2019).

54. Starkweather, C.K., Babayan, B.M., Uchida, N. & Gershman, S.J. Dopamine reward prediction errors reflect hidden-state inference across time. *Nat Neurosci* **20**, 581-589 (2017).
55. Feldman, D.E. Synaptic mechanisms for plasticity in neocortex. *Annu Rev Neurosci* **32**, 33-55 (2009).
56. Bienenstock, E.L., Cooper, L.N. & Munro, P.W. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J Neurosci* **2**, 32-48 (1982).
57. Gjorgjieva, J., Clopath, C., Audet, J. & Pfister, J.P. A triplet spike-timing-dependent plasticity model generalizes the Bienenstock-Cooper-Munro rule to higher-order spatiotemporal correlations. *Proc Natl Acad Sci U S A* **108**, 19383-19388 (2011).
58. Shouval, H.Z. What is the appropriate description level for synaptic plasticity? *Proc Natl Acad Sci U S A* **108**, 19103-19104 (2011).
59. Morita, K. & Kato, A. Striatal dopamine ramping may indicate flexible reinforcement learning with forgetting in the cortico-basal ganglia circuits. *Front Neural Circuits* **8**, 36 (2014).
60. Kato, A. & Morita, K. Forgetting in Reinforcement Learning Links Sustained Dopamine Signals to Motivation. *PLoS Comput Biol* **12**, e1005145 (2016).
61. Kato, A. & Morita, K. Striatal Gradient in Value-Decay Explains Regional Differences in Dopamine Patterns and Reinforcement Learning Computations. *bioRxiv* **https://doi.org/10.1101/2025.01.24.631285** (2025).
62. Cornford, J., *et al.* Learning to live with Dale's principle: ANNs with separate excitatory and inhibitory units. *bioRxiv* **https://doi.org/10.1101/2020.11.02.364968** (2021).
63. Li, P., Cornford, J., Ghosh, A. & Richards, B. Learning better with Dale's Law: A Spectral Perspective. *bioRxiv* **https://doi.org/10.1101/2023.06.28.546924** (2023).
64. Mastrogiuseppe, F. & Ostojic, S. Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks. *Neuron* **99**, 609-623.e629 (2018).
65. Morita, K., Kalra, R., Aihara, K. & Robinson, H. Recurrent synaptic input and the timing of gamma-frequency-modulated firing of pyramidal cells during neocortical "UP" states. *J Neurosci* **28**, 1871-1881 (2008).
66. Hasenstaub, A., *et al.* Inhibitory postsynaptic potentials carry synchronized frequency information in active cortical networks. *Neuron* **47**, 423-435 (2005).
67. Wang, X.-J. Neurophysiological and computational principles of cortical rhythms in cognition. *Physiol Rev* **90**, 1195-1268 (2010).
68. Buzsáki, G. & Wang, X.J. Mechanisms of gamma oscillations. *Annu Rev Neurosci* **35**, 203-225 (2012).
69. Cone, I., Clopath, C. & Shouval, H.Z. Learning to express reward prediction error-like dopaminergic activity requires plastic representations of time. *Nat Commun* **15**, 5856 (2024).
70. Sutton, R. & Barto, A. *Reinforcement Learning* (MIT Press, Cambridge, MA, 1998).
71. Takahashi, Y., Schoenbaum, G. & Niv, Y. Silencing the critics: understanding the effects of

cocaine sensitization on dorsolateral and ventral striatum in the context of an actor/critic model. *Front Neurosci* **2**, 86-99 (2008).

72. Wang, X.-J. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron* **36**, 955-968 (2002).
73. Lo, C. & Wang, X. Cortico-basal ganglia circuit mechanism for a decision threshold in reaction time tasks. *Nat Neurosci* **9**, 956-963 (2006).
74. Hunt, L.T., *et al.* Mechanisms underlying cortical activity during value-guided choice. *Nat Neurosci* **15**, 470-476, S471-473 (2012).
75. Williams, S.M. & Goldman-Rakic, P.S. Widespread origin of the primate mesofrontal dopamine system. *Cereb Cortex* **8**, 321-345 (1998).
76. Broussard, J.I., *et al.* Dopamine Regulates Aversive Contextual Learning and Associated In Vivo Synaptic Plasticity in the Hippocampus. *Cell Rep* **14**, 1930-1939 (2016).
77. Samejima, K., Ueda, Y., Doya, K. & Kimura, M. Representation of action-specific reward values in the striatum. *Science* **310**, 1337-1340 (2005).
78. Brozoski, T.J., Brown, R.M., Rosvold, H.E. & Goldman, P.S. Cognitive deficit caused by regional depletion of dopamine in prefrontal cortex of rhesus monkey. *Science* **205**, 929-932 (1979).
79. Sawaguchi, T. & Goldman-Rakic, P.S. D1 dopamine receptors in prefrontal cortex: involvement in working memory. *Science* **251**, 947-950 (1991).
80. Durstewitz, D., Seamans, J.K. & Sejnowski, T.J. Dopamine-mediated stabilization of delay-period activity in a network model of prefrontal cortex. *J Neurophysiol* **83**, 1733-1750 (2000).
81. Brunel, N. & Wang, X. Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. *J Comput Neurosci* **11**, 63-85 (2001).
82. Floresco, S.B. & Magyar, O. Mesocortical dopamine modulation of executive functions: beyond working memory. *Psychopharmacology (Berl)* **188**, 567-585 (2006).
83. Tsetsenis, T., *et al.* Midbrain dopaminergic innervation of the hippocampus is sufficient to modulate formation of aversive memories. *Proc Natl Acad Sci U S A* **118**, e2111069118 (2021).
84. O'Doherty, J.P., Dayan, P., Friston, K., Critchley, H. & Dolan, R.J. Temporal difference models and reward-related learning in the human brain. *Neuron* **38**, 329-337 (2003).
85. Takahashi, Y.K., *et al.* Expectancy-related changes in firing of dopamine neurons depend on orbitofrontal cortex. *Nat Neurosci* **14**, 1590-1597 (2011).
86. Starkweather, C.K., Gershman, S.J. & Uchida, N. The Medial Prefrontal Cortex Shapes Dopamine Reward Prediction Errors under State Uncertainty. *Neuron* **98**, 616-629.e616 (2018).
87. Takahashi, Y.K., *et al.* Expectancy-related changes in firing of dopamine neurons depend on hippocampus. *bioRxiv* **https://doi.org/10.1101/2023.07.19.549728** (2023).
88. Otani, S., Daniel, H., Roisin, M.P. & Crepel, F. Dopaminergic modulation of long-term synaptic plasticity in rat prefrontal neurons. *Cereb Cortex* **13**, 1251-1256 (2003).
89. Sayegh, F.J.P., *et al.* Ventral tegmental area dopamine projections to the hippocampus trigger

long-term potentiation and contextual learning. *Nat Commun* **15**, 4100 (2024).
90. Lee, R.S., Sagiv, Y., Engelhard, B., Witten, I.B. & Daw, N.D. A feature-specific prediction error model explains dopaminergic heterogeneity. *Nat Neurosci* **27**, 1574-1586 (2024).
91. Avvisati, R., *et al.* Distributional coding of associative learning in discrete populations of midbrain dopamine neurons. *Cell Rep* **43**, 114080 (2024).
92. Watabe-Uchida, M., Zhu, L., Ogawa, S.K., Vamanrao, A. & Uchida, N. Whole-brain mapping of direct inputs to midbrain dopamine neurons. *Neuron* **74**, 858-873 (2012).
93. Carta, I., Chen, C.H., Schott, A.L., Dorizan, S. & Khodakhah, K. Cerebellar modulation of the reward circuitry and social behavior. *Science* **363** (2019).
94. Marr, D. A theory of cerebellar cortex. *J Physiol* **202**, 437-470 (1969).
95. Beier, K.T., *et al.* Circuit Architecture of VTA Dopamine Neurons Revealed by Systematic Input-Output Mapping. *Cell* **162**, 622-634 (2015).
96. Li, C. & Rainnie, D.G. Bidirectional regulation of synaptic plasticity in the basolateral amygdala induced by the D1-like family of dopamine receptors and group II metabotropic glutamate receptors. *J Physiol* **592**, 4329-4351 (2014).
97. Sias, A.C., *et al.* Dopamine projections to the basolateral amygdala drive the encoding of identity-specific reward memories. *Nat Neurosci* **27**, 728-736 (2024).
98. Headley, D.B., Kyriazi, P., Feng, F., Nair, S.S. & Pare, D. Gamma Oscillations in the Basolateral Amygdala: Localization, Microcircuitry, and Behavioral Correlates. *J Neurosci* **41**, 6087-6101 (2021).
99. Britt, J.P., *et al.* Synaptic and behavioral profile of multiple glutamatergic inputs to the nucleus accumbens. *Neuron* **76**, 790-803 (2012).
100. Lee, I.B., *et al.* Persistent enhancement of basolateral amygdala-dorsomedial striatum synapses causes compulsive-like behaviors in mice. *Nat Commun* **15**, 219 (2024).
101. Saez, A., Rigotti, M., Ostojic, S., Fusi, S. & Salzman, C.D. Abstract Context Representations in Primate Amygdala and Prefrontal Cortex. *Neuron* **87**, 869-881 (2015).
102. Stalnaker, T.A., *et al.* Dopamine neuron ensembles signal the content of sensory prediction errors. *Elife* **8**, e49315 (2019).
103. Gerfen, C.R. & Surmeier, D.J. Modulation of Striatal Projection Systems by Dopamine. *Annu Rev Neurosci* **34**, 441-466 (2011).
104. Collins, A.G. & Frank, M.J. Opponent actor learning (OpAL): modeling interactive effects of striatal dopamine on reinforcement learning and choice incentive. *Psychol Rev* **121**, 337-366 (2014).
105. Mikhael, J.G. & Bogacz, R. Learning Reward Uncertainty in the Basal Ganglia. *PLoS Comput Biol* **12**, e1005062 (2016).
106. Morita, K. & Kawaguchi, Y. A Dual Role Hypothesis of the Cortico-Basal-Ganglia Pathways: Opponency and Temporal Difference Through Dopamine and Adenosine. *Front Neural Circuits* **12**, 111 (2019).

107. Lowet, A.S., *et al.* An opponent striatal circuit for distributional reinforcement learning. *Nature* **639**, 717-726 (2025).
108. Wall, N.R., De La Parra, M., Callaway, E.M. & Kreitzer, A.C. Differential innervation of direct- and indirect-pathway striatal projection neurons. *Neuron* **79**, 347-360 (2013).
109. Morita, K. Differential cortical activation of the striatal direct and indirect pathway cells: reconciling the anatomical and optogenetic results by using a computational method. *J Neurophysiol* **112**, 120-146 (2014).
110. Hooks, B.M., *et al.* Topographic precision in sensory and motor corticostriatal projections varies across cell type and cortical area. *Nat Commun* **9**, 3549 (2018).
111. Morita, K., Im, S. & Kawaguchi, Y. Differential striatal axonal arborizations of the intratelencephalic and pyramidal-tract neurons: analysis of the data in the MouseLight database. *Front Neural Circuits* **13**, 71 (2019).
112. Tian, J., *et al.* Distributed and Mixed Information in Monosynaptic Inputs to Dopamine Neurons. *Neuron* **91**, 1374-1389 (2016).
113. Levy, D.J. & Glimcher, P.W. The root of all value: a neural common currency for choice. *Curr Opin Neurobiol* **22**, 1027-1038 (2012).
114. Konishi, M., Igarashi, K.M. & Miura, K. Biologically plausible local synaptic learning rules robustly implement deep supervised learning. *Front Neurosci* **17**, 1160899 (2023).
115. Cho, K., *et al.* Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *arXiv* **1406.1078** (2014).
116. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **6**, 107-115 (1998).
117. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput* **9**, 1735-1780 (1997).
118. Costa, R.P., Assael, Y.M., Shillingford, B., De Freitas, N. & Vogels, T. Cortical microcircuits as gated-recurrent neural networks. *Advances in Neural Information Processing Systems* (2017).
119. Shouval, H.Z., Wang, S.S. & Wittenberg, G.M. Spike timing dependent plasticity: a consequence of more fundamental learning rules. *Front Comput Neurosci* **4**, 19 (2010).
120. Poirazi, P., Brannon, T. & Mel, B. Pyramidal neuron as two-layer neural network. *Neuron* **37**, 989-999 (2003).
121. Morita, K. Possible role of dendritic compartmentalization in the spatial working memory circuit. *J Neurosci* **28**, 7699-7724 (2008).
122. Körding, K.P. & König, P. Supervised and unsupervised learning with two sites of synaptic integration. *J Comput Neurosci* **11**, 207-215 (2001).
123. Illing, B., Ventura, J., Bellec, G. & Gerstner, W. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. in *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)* (2021).
124. Barto, A.G. & Jordan, M.I. Gradient Following Without Back-Propagation in Layered Networks.

in *Proceedings of the First Annual International Conference on Neural Networks Vol. II* 629-636 (San Diego, CA., 1987).

125. Mazzoni, P., Andersen, R.A. & Jordan, M.I. A more biologically plausible learning rule for neural networks. *Proc Natl Acad Sci U S A* **88**, 4433-4437 (1991).

126. Mazzoni, P., Andersen, R.A. & Jordan, M.I. A more biologically plausible learning rule than backpropagation applied to a network model of cortical area 7a. *Cereb Cortex* **1**, 293-307 (1991).

127. Max, K., *et al.* Learning efficient backprojections across cortical hierarchies in real time. *Nature Machine Intelligence* **6**, 619–630 (2024).

128. Hoerzer, G.M., Legenstein, R. & Maass, W. Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning. *Cereb Cortex* **24**, 677-690 (2014).

129. Faisal, A.A., Selen, L.P. & Wolpert, D.M. Noise in the nervous system. *Nat Rev Neurosci* **9**, 292-303 (2008).

130. Aihara, K. & Matsumoto, G. Chaotic oscillations and bifurcations in squid giant axons. in *Chaos* (ed. A.V. Holden) (Princeton University Press, 1986).

131. van Vreeswijk, C. & Sompolinsky, H. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* **274**, 1724-1726 (1996).

132. Krausz, T.A., *et al*. Dual credit assignment processes underlie dopamine signals in a complex spatial environment. *Neuron* **111**, 3465-3478.e3467 (2023).

133. Ludvig, E.A., Sutton, R.S. & Kehoe, E.J. Evaluating the TD model of classical conditioning. *Learn Behav* **40**, 305-319 (2012).

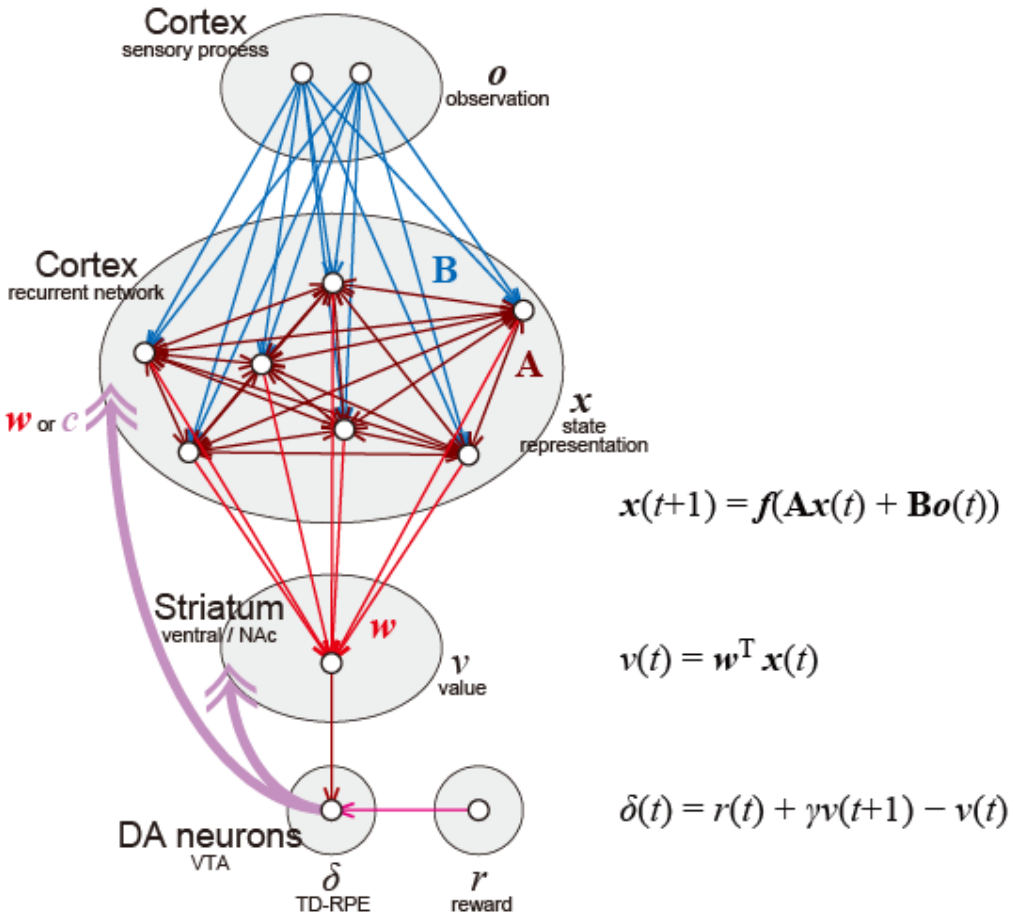
Cortex
sensory process
o
observation
Cortex
recurrent network
B
A
x
state
representation
w or c
Striatum
ventral / NAc
w
v
value
DA neurons
VTA
δ
TD-RPE
r
reward
$\boldsymbol{x}(t+1) = \boldsymbol{f}(\mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{o}(t))$
$v(t) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}(t)$
$\delta(t) = r(t) + \gamma v(t+1) - v(t)$

Figure 1

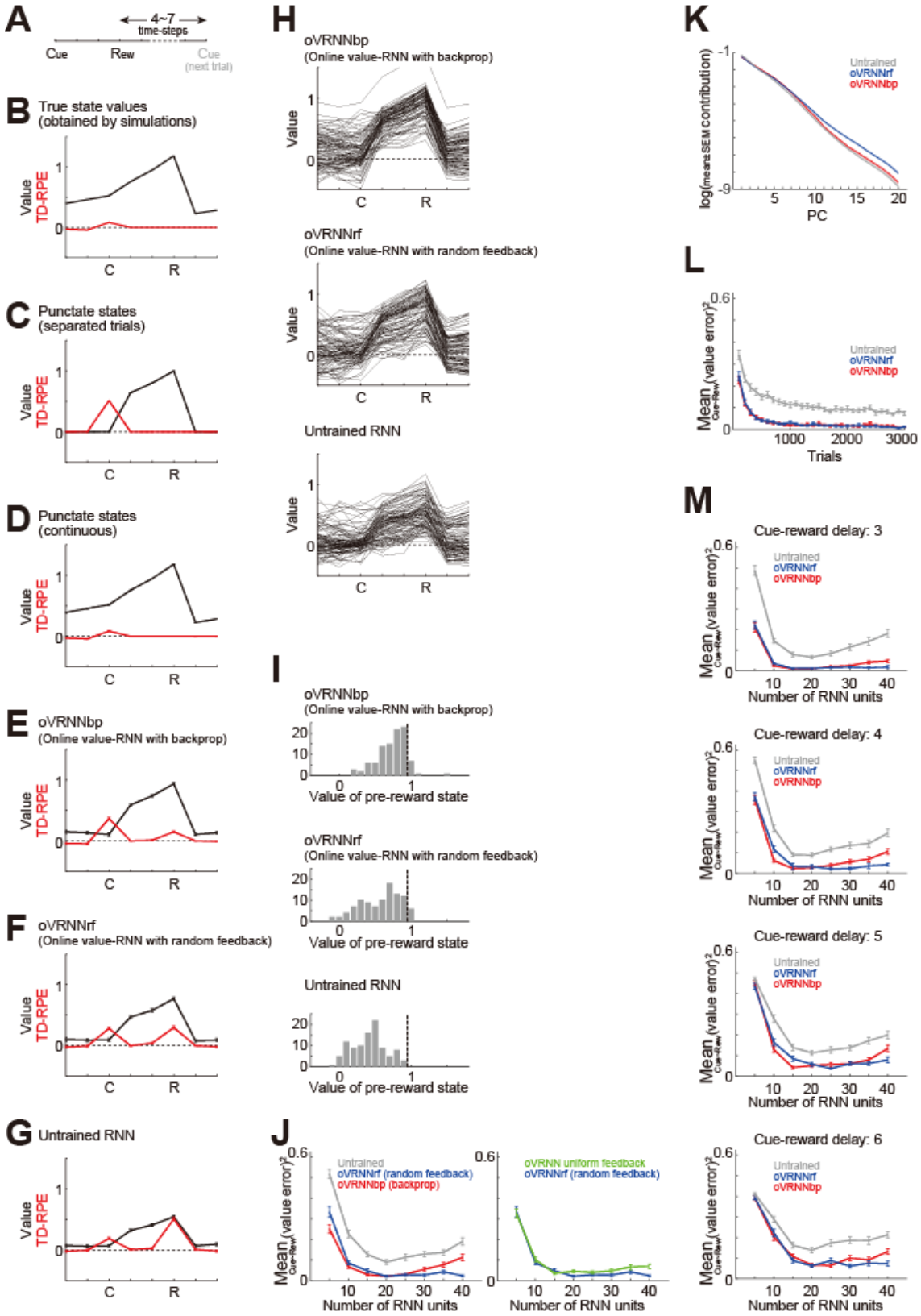
A
4~7
time-steps
Cue
Rew
Cue
(next trial)
B True state values
(obtained by simulations)
C Punctate states
(separated trials)
D Punctate states
(continuous)
E oVRNNbp
(Online value-RNN with backprop)
F oVRNNrf
(Online value-RNN with random feedback)
G Untrained RNN
Value
TD-RPE
C
R
H
oVRNNbp
(Online value-RNN with backprop)
oVRNNrf
(Online value-RNN with random feedback)
Untrained RNN
I
Value of pre-reward state
J
Untrained
oVRNNrf (random feedback)
oVRNNbp (backprop)
oVRNN uniform feedback
Number of RNN units
K
log(mean±SEM contribution)
PC
L
Trials
M
Cue-reward delay: 3
Cue-reward delay: 4
Cue-reward delay: 5
Cue-reward delay: 6

Figure 2

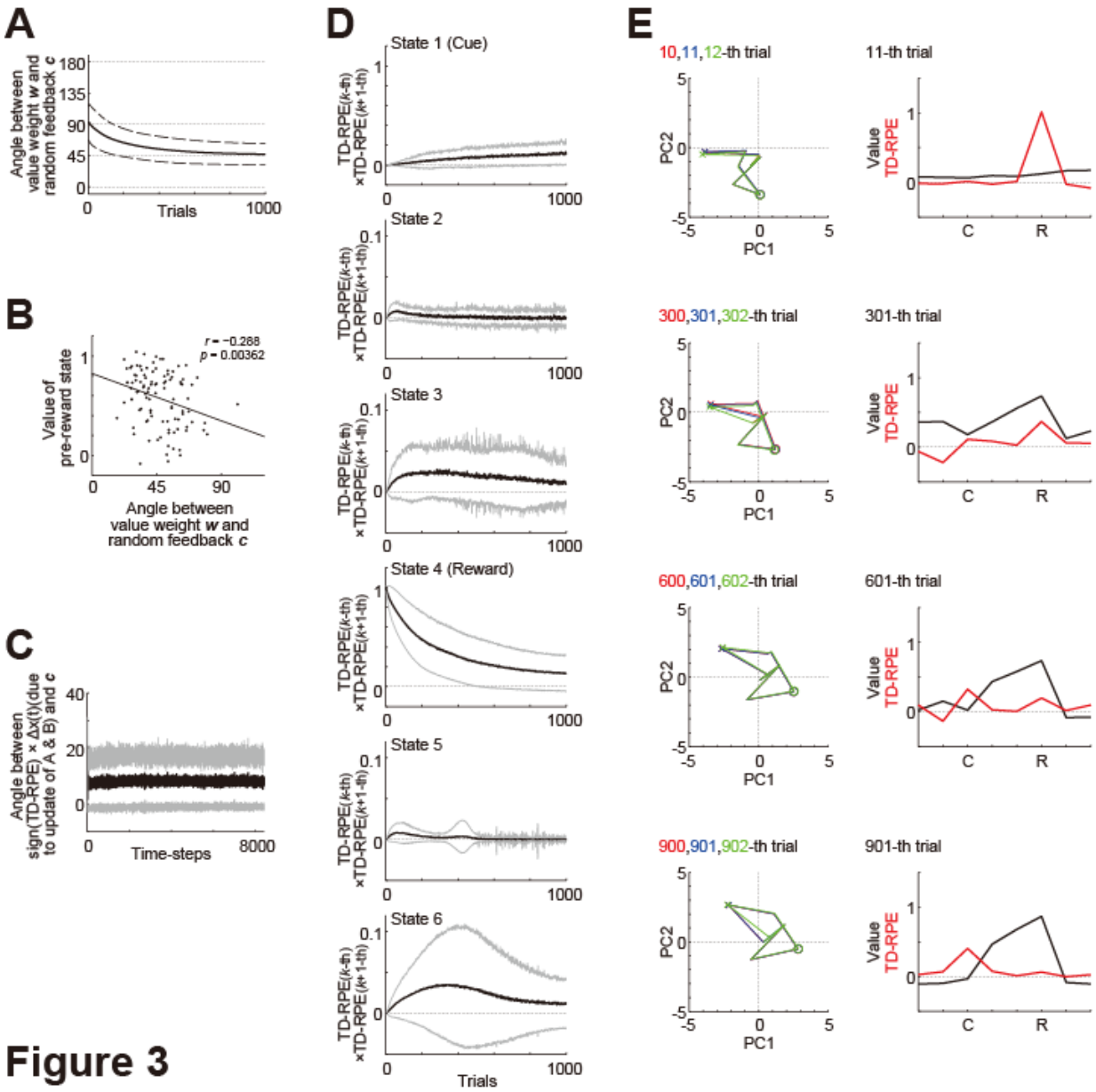
A
Angle between value weight w and random feedback c
180
135
90
45
0
0
Trials
1000
B
r = −0.288
p = 0.00362
Value of pre-reward state
1
0
0
45
90
Angle between value weight w and random feedback c
C
Angle between sign(TD-RPE) × Δx(t)(due to update of A & B) and c
40
20
0
0
Time-steps
8000
D
State 1 (Cue)
State 2
State 3
State 4 (Reward)
State 5
State 6
TD-RPE(k-th) ×TD-RPE(k+1-th)
0
1000
Trials
E
10,11,12-th trial
11-th trial
300,301,302-th trial
301-th trial
600,601,602-th trial
601-th trial
900,901,902-th trial
901-th trial
PC1
PC2
Value
TD-RPE
C
R

Figure 3

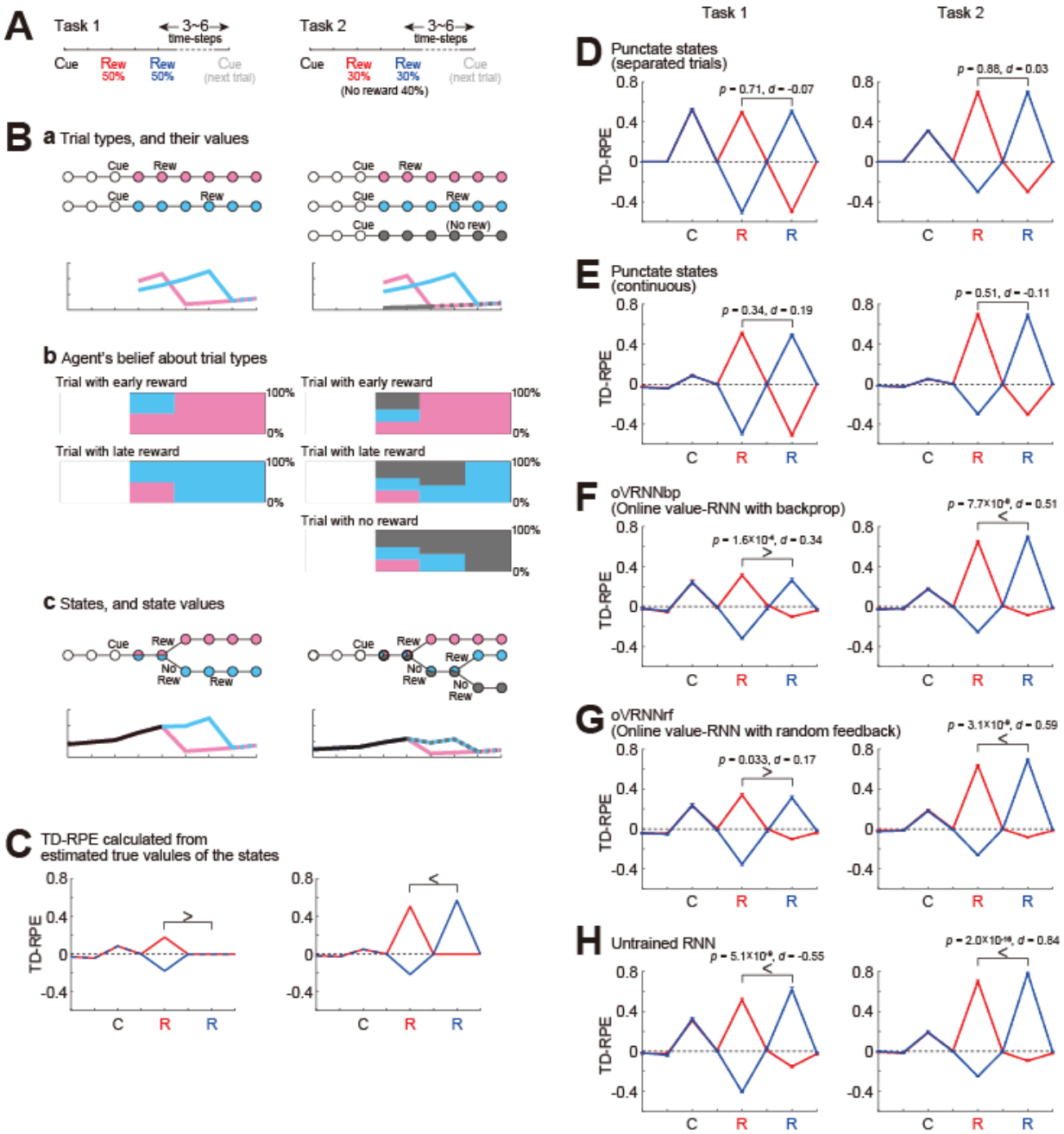
A
Task 1
3~6
time-steps
Cue
Rew 50%
Rew 50%
Cue (next trial)
Task 2
3~6
time-steps
Cue
Rew 30%
Rew 30%
(No reward 40%)
Cue (next trial)
B
a Trial types, and their values
Cue
Rew
(No rew)
b Agent's belief about trial types
Trial with early reward
Trial with late reward
Trial with no reward
100%
0%
c States, and state values
Cue
Rew
No Rew
C
TD-RPE calculated from estimated true valules of the states
TD-RPE
0.8
0.4
0
-0.4
C
R
R
Task 1
Task 2
D
Punctate states (separated trials)
p = 0.71, d = -0.07
p = 0.88, d = 0.03
E
Punctate states (continuous)
p = 0.34, d = 0.19
p = 0.51, d = -0.11
F
oVRNNbp
(Online value-RNN with backprop)
p = 1.6×10⁻⁴, d = 0.34
p = 7.7×10⁻⁶, d = 0.51
G
oVRNNrf
(Online value-RNN with random feedback)
p = 0.033, d = 0.17
p = 3.1×10⁻⁶, d = 0.59
H
Untrained RNN
p = 5.1×10⁻⁶, d = -0.55
p = 2.0×10⁻¹⁸, d = 0.84

Figure 4

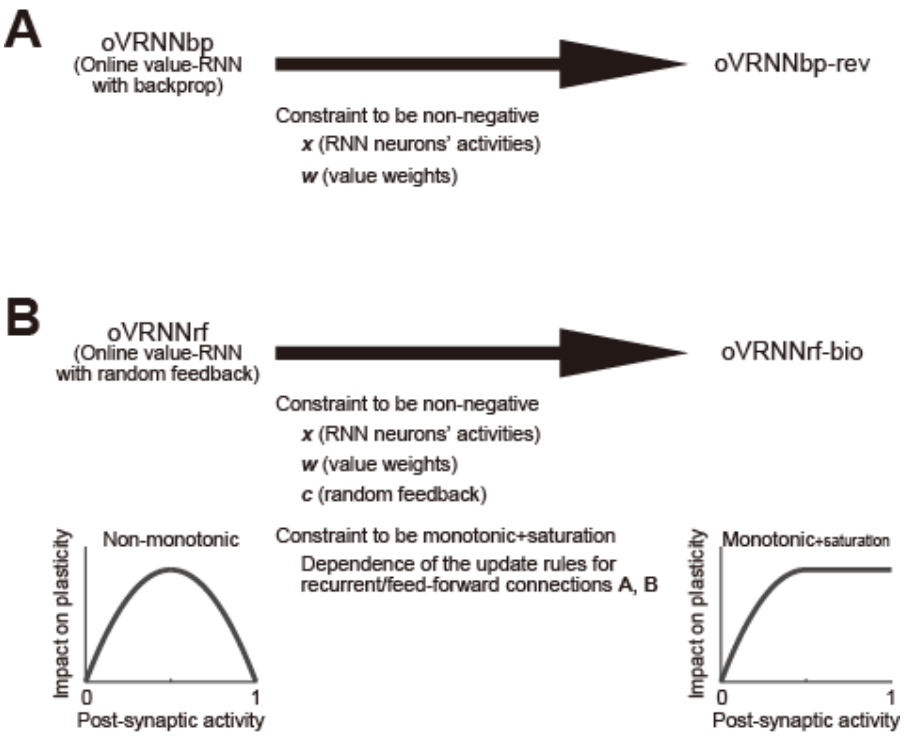
A
oVRNNbp
(Online value-RNN
with backprop)
oVRNNbp-rev
Constraint to be non-negative
x (RNN neurons' activities)
w (value weights)
B
oVRNNrf
(Online value-RNN
with random feedback)
oVRNNrf-bio
Constraint to be non-negative
x (RNN neurons' activities)
w (value weights)
c (random feedback)
Non-monotonic
Impact on plasticity
0
1
Post-synaptic activity
Constraint to be monotonic+saturation
Dependence of the update rules for
recurrent/feed-forward connections A, B
Monotonic+saturation
Impact on plasticity
0
1
Post-synaptic activity

Figure 5

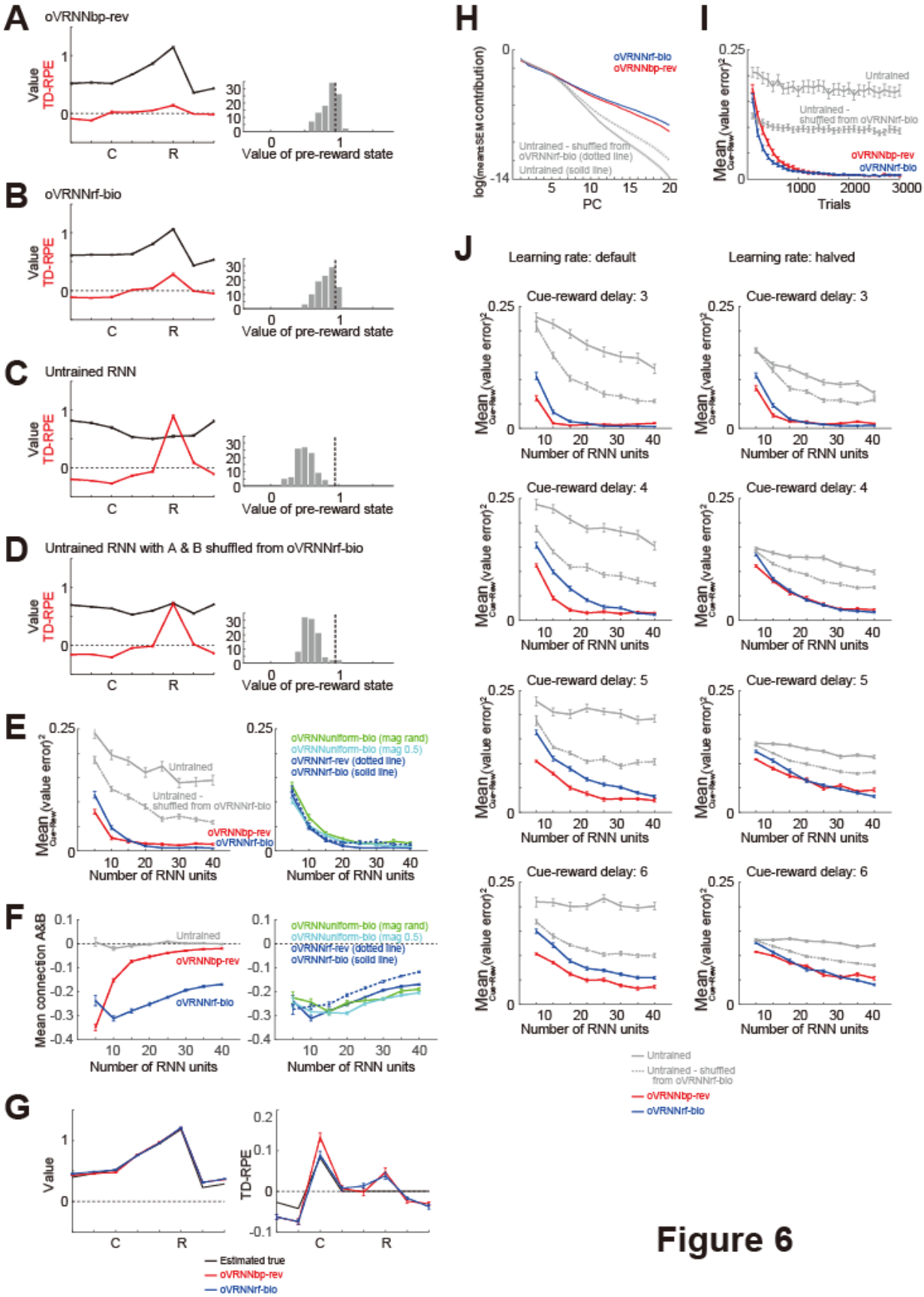
A
oVRNNbp-rev
Value
TD-RPE
C
R
Value of pre-reward state
B
oVRNNrf-bio
C
Untrained RNN
D
Untrained RNN with A & B shuffled from oVRNNrf-bio
E
Mean (value error)²
Untrained
Untrained - shuffled from oVRNNrf-bio
oVRNNbp-rev
oVRNNrf-bio
oVRNNuniform-bio (mag rand)
oVRNNuniform-bio (mag 0.5)
oVRNNrf-rev (dotted line)
oVRNNrf-bio (solid line)
Number of RNN units
F
Mean connection A&B
G
Estimated true
H
log(mean±SEM contribution)
Untrained - shuffled from oVRNNrf-bio (dotted line)
Untrained (solid line)
PC
I
Trials
J
Learning rate: default
Learning rate: halved
Cue-reward delay: 3
Cue-reward delay: 4
Cue-reward delay: 5
Cue-reward delay: 6

Figure 6

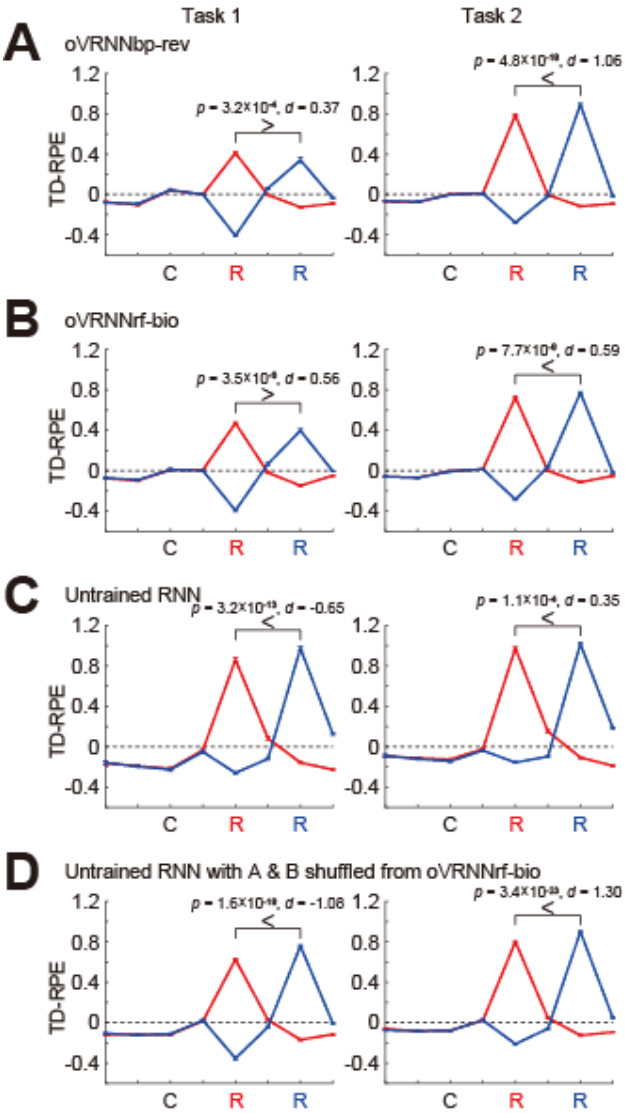
Task 1
Task 2
A
oVRNNbp-rev
p = 3.2×10⁻⁴, d = 0.37
p = 4.8×10⁻¹⁸, d = 1.06
B
oVRNNrf-bio
p = 3.5×10⁻⁴, d = 0.56
p = 7.7×10⁻⁶, d = 0.59
C
Untrained RNN
p = 3.2×10⁻¹³, d = -0.65
p = 1.1×10⁻⁴, d = 0.35
D
Untrained RNN with A & B shuffled from oVRNNrf-bio
p = 1.6×10⁻¹⁸, d = -1.08
p = 3.4×10⁻³³, d = 1.30
TD-RPE
1.2
0.8
0.4
0
-0.4
C
R
R

Figure 7

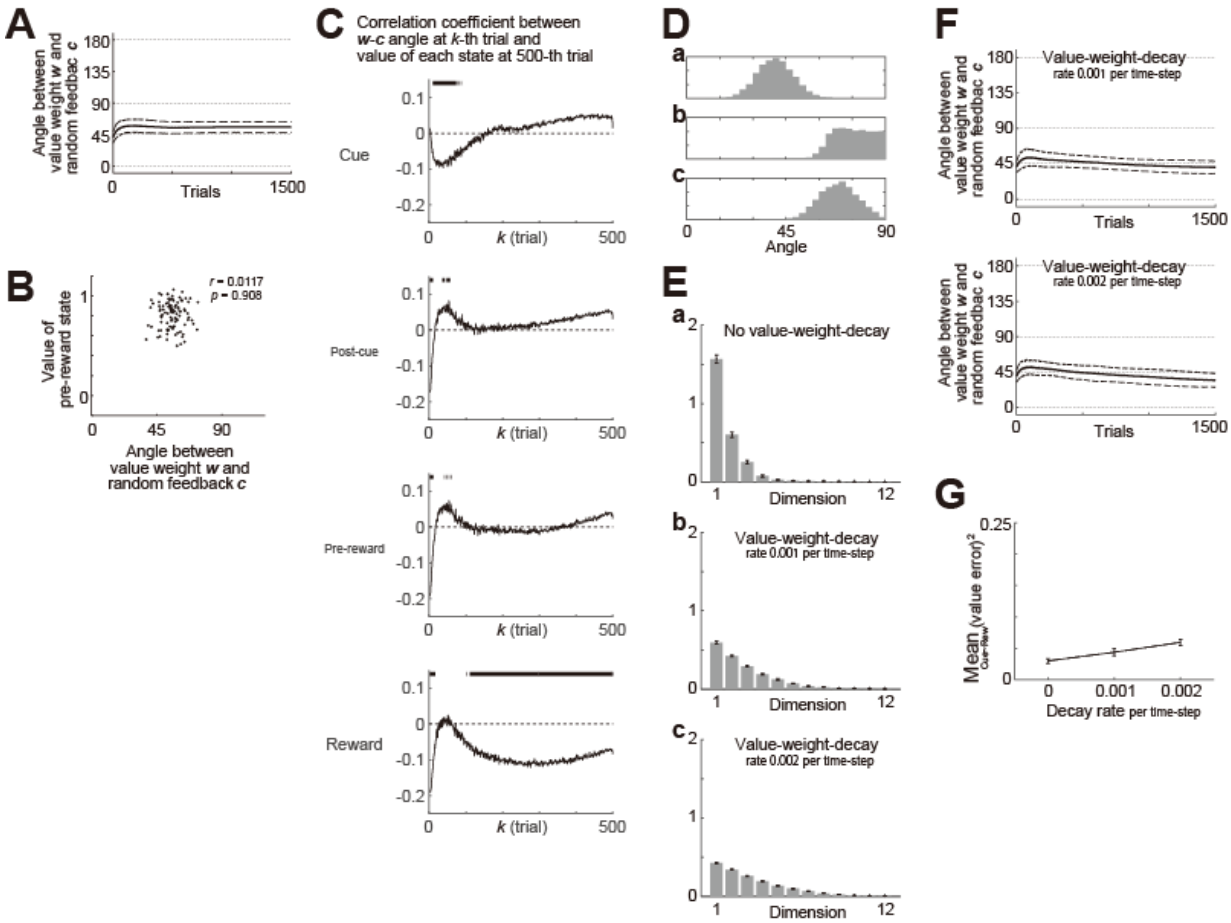
A
Angle between value weight w and random feedbac c
180
135
90
45
0
0
Trials
1500
B
r = 0.0117
p = 0.908
Value of pre-reward state
1
0
0
45
90
Angle between value weight w and random feedback c
C
Correlation coefficient between w-c angle at k-th trial and value of each state at 500-th trial
Cue
Post-cue
Pre-reward
Reward
0.1
0
-0.1
-0.2
0
k (trial)
500
D
a
b
c
0
45
90
Angle
E
a
No value-weight-decay
b
Value-weight-decay
rate 0.001 per time-step
c
Value-weight-decay
rate 0.002 per time-step
2
1
0
1
Dimension
12
F
Value-weight-decay
rate 0.001 per time-step
Value-weight-decay
rate 0.002 per time-step
Angle between value weight w and random feedbac c
180
135
90
45
0
0
Trials
1500
G
Mean (value error)²
0.25
0
0
0.001
0.002
Decay rate per time-step

Figure 8

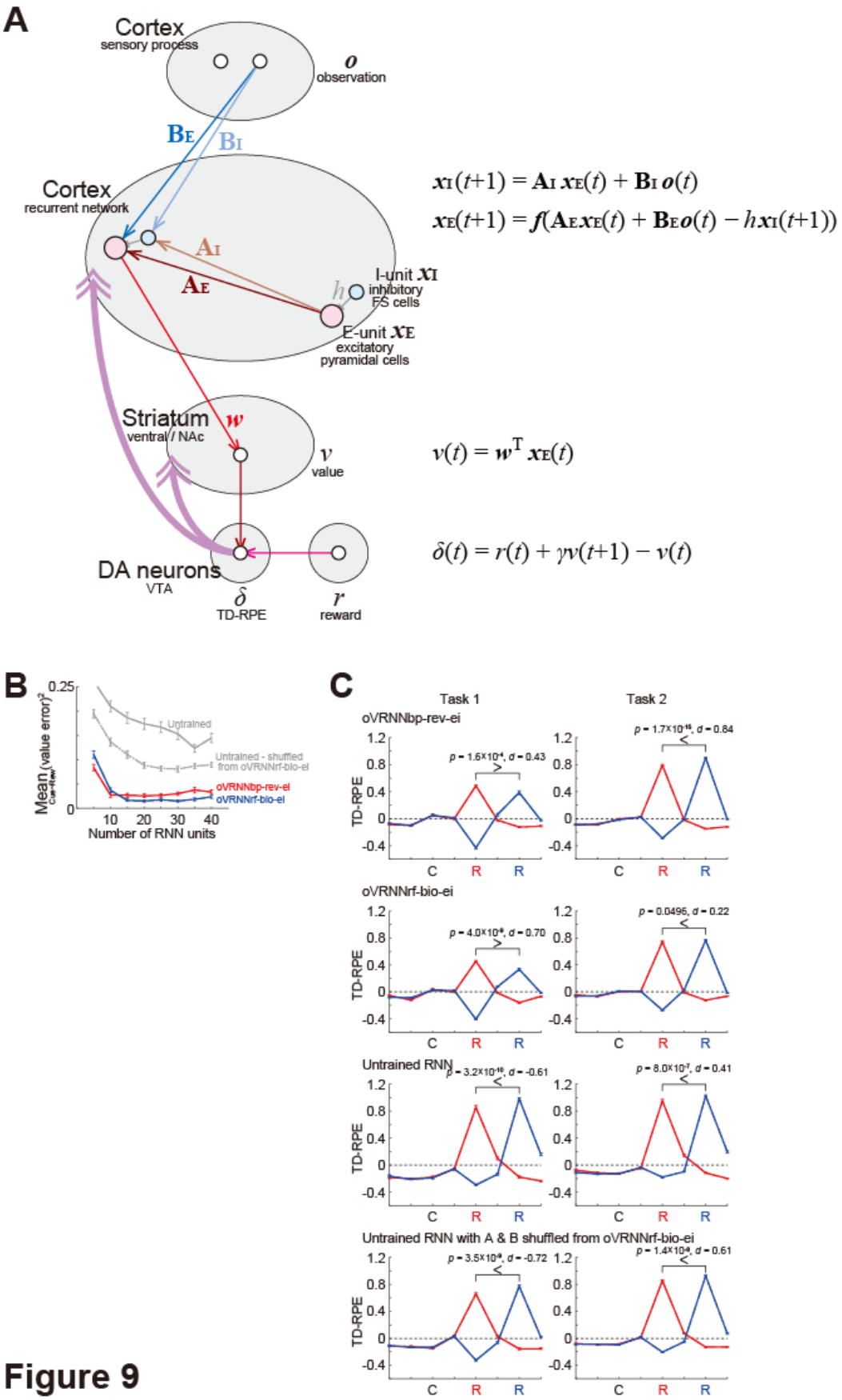
A
Cortex
sensory process
o
observation
BE
BI
Cortex
recurrent network
AI
AE
h
I-unit xI
inhibitory
FS cells
E-unit xE
excitatory
pyramidal cells
Striatum
ventral / NAc
w
v
value
DA neurons
VTA
δ
TD-RPE
r
reward
xI(t+1) = AI xE(t) + BI o(t)
xE(t+1) = f(AE xE(t) + BE o(t) − h xI(t+1))
v(t) = wT xE(t)
δ(t) = r(t) + γv(t+1) − v(t)
B
Mean (value error)2
Untrained
Untrained - shuffled from oVRNNrf-bio-ei
oVRNNbp-rev-ei
oVRNNrf-bio-ei
Number of RNN units
C
Task 1
Task 2
oVRNNbp-rev-ei
oVRNNrf-bio-ei
Untrained RNN
Untrained RNN with A & B shuffled from oVRNNrf-bio-ei
TD-RPE
C
R
R

Figure 9

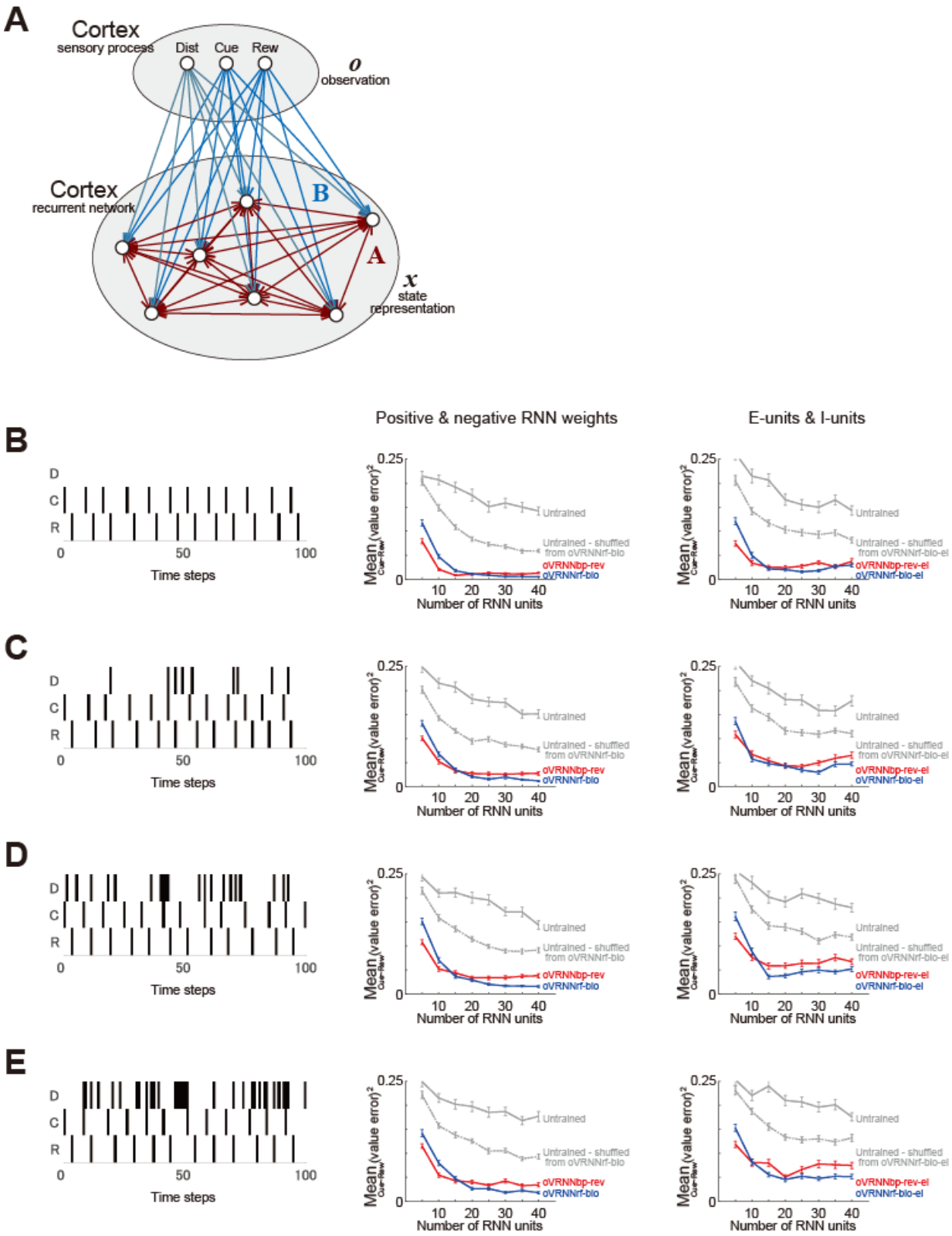
A
Cortex
sensory process
Dist
Cue
Rew
o
observation
Cortex
recurrent network
B
A
x
state
representation
Positive & negative RNN weights
E-units & I-units
B
C
D
E
D
C
R
0
50
100
Time steps
Untrained
Untrained - shuffled from oVRNNrf-bio
oVRNNbp-rev
oVRNNrf-bio
Untrained - shuffled from oVRNNrf-bio-el
oVRNNbp-rev-el
oVRNNrf-bio-el
0.25
0
10
20
30
40
Number of RNN units

Figure 10

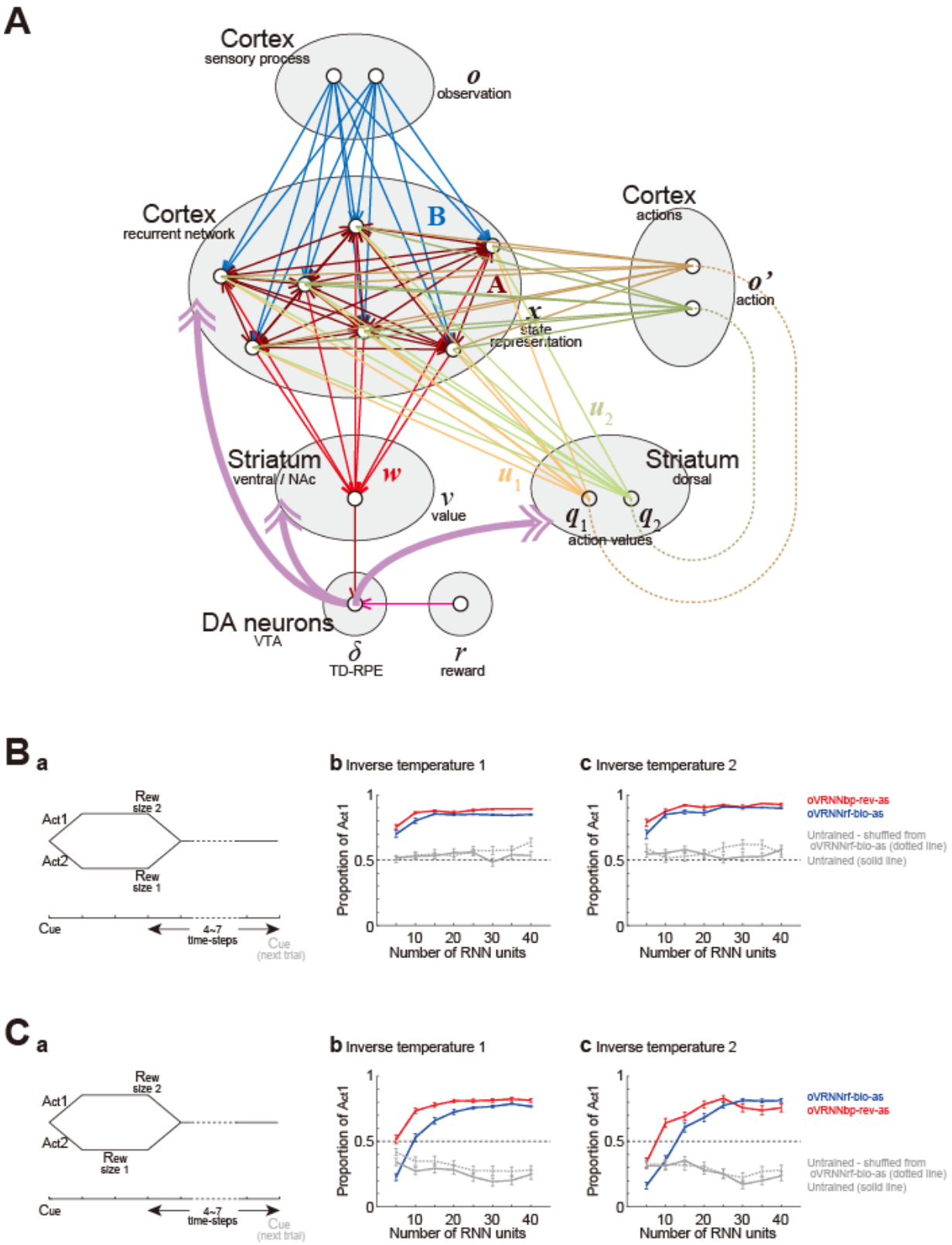
A
Cortex
sensory process
o
observation
Cortex
recurrent network
B
A
x
state
representation
Cortex
actions
o'
action
u2
u1
Striatum
ventral / NAc
w
v
value
Striatum
dorsal
q1
q2
action values
DA neurons
VTA
δ
TD-RPE
r
reward
B a
Rew
size 2
Act1
Act2
Rew
size 1
Cue
4~7
time-steps
Cue
(next trial)
b Inverse temperature 1
Proportion of Act1
Number of RNN units
c Inverse temperature 2
oVRNNbp-rev-as
oVRNNrf-bio-as
Untrained - shuffled from
oVRNNrf-bio-as (dotted line)
Untrained (solid line)
C a
Rew
size 2
Act1
Act2
Rew
size 1
Cue
4~7
time-steps
Cue
(next trial)
b Inverse temperature 1
Proportion of Act1
Number of RNN units
c Inverse temperature 2
oVRNNrf-bio-as
oVRNNbp-rev-as
Untrained - shuffled from
oVRNNrf-bio-as (dotted line)
Untrained (solid line)

Figure 11