Spiking network model of the cerebellum as a reinforcement learning machine

Rin Kuriyama[1†], Hideyuki Yoshimura[2†], Tadashi Yamazaki[1*]

[1*]Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan.
[2]Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan.

*Corresponding author(s). E-mail(s): contact24@numericalbrain.org;
[†]These authors contributed equally to this work.

Abstract

Reinforcement learning (RL) is a machine learning algorithm that finds optimal solutions through exploration, making it applicable in scenarios where supervised learning cannot be utilized. The brain also uses RL as an adaptive system in a complex and changing world, and the basal ganglia are known to be involved. However, it remains unclear whether other brain regions also utilize RL. In this study, we focused on the cerebellum, which has recently been reconsidered as an RL machine rather than a supervised learning machine, and we implemented its spiking network model in an actor-critic manner. Our model successfully solved a simple RL task and a cerebellum-dependent motor learning task. Furthermore, the model reproduced results in a lesion study on the same motor learning task. These results provide a spike-based implementation of an RL algorithm and a fresh view on the learning principle of the cerebellum performing RL.

Keywords: Cerebellum, Spiking Neural Network, Reinforcement Learning, Simulation

Introduction

Animals, including humans, adjust behaviors time after time in response to their ever-changing environment. Such behaviors are thought to be acquired through reinforcement learning (RL), which is a machine learning algorithm that demonstrates

tremendous capability surpassing human performance [1, 2]. Thus, this powerful algorithm must be implemented somewhere in the brain, where neurons exchange short electrical pulses as bits called spikes for information processing.

The basal ganglia is a first candidate that seems to perform RL, where dopamine signals represent reward-related information called temporal difference (TD) errors [3, 4]. On the other hand, the cerebellum, which was traditionally considered an error-based supervised learning (SL) machine according to the Marr-Albus-Ito model [5–7], has recently been reconsidered and reformulated as an RL machine [8] after the discovery of TD-error-like activity of instruction signals to the cerebellum [9]. However, few attempts have been made to implement an RL algorithm in a spiking network model, which is a neural network in which neurons transmit spikes as biological neurons but not analog values as in conventional artificial neural networks. One exception is a study by Frémaux et al. [10], which developed a general spiking network model that can perform RL but is not a model of any specific regions in the brain.

In this study, we aimed to develop a spiking network model of the cerebellum that can perform RL, demonstrating a viable alternative to the 50-year-old view of the cerebellum as an SL machine [11]. Specifically, molecular layer interneurons (MLIs) and their associated synaptic plasticity, which had been missed in the Marr-Albus-Ito model, were taken into account explicitly. A biologically plausible learning rule for the synaptic plasticity was also formulated based on the known anatomy and physiology of the cerebellum.

Eventually, our spiking network model of the cerebellum was able to solve a simple RL task and also reproduce experimental results in cerebellar motor learning. To our knowledge, the model presented here is the first cerebellar spiking network model that can act as an RL machine.

1 Results

1.1 Spike-based implementation of cerebellum-style RL

In this research, we implemented a cerebellar spiking network model as an RL machine while referring to the cerebellar RL concept [8] and the spiking actor-critic framework [10]. A schematic of cell types and synaptic connections incorporated in the present model, along with a flow diagram of neural signaling, is shown in Figure 1. Parallel fibers (PFs) excite stellate cells (SCs), basket cells (BCs), and Purkinje cells (PCs). SCs and BCs inhibit PCs. BCs and PCs inhibit each other, and also both have a self-inhibitory connection. More specifically, BCs and PCs are divided into multiple groups, which correspond to action choices as described in Section 1.1.3. BCs inhibit PCs in the same group and BCs in other groups, whereas PCs inhibit BCs in the same group and PCs in other groups. PCs also inhibit neurons in the deep cerebellar nuclei (DCN), which are also divided into multiple groups. CFs provide excitatory inputs to SCs and PCs. We used CF signals to trigger synaptic plasticity. Therefore, we did not implement actual CF–SC and CF–PC connections.

In the cerebellar spiking network model, we implemented an RL algorithm known as the actor-critic model [12] based on the previous spiking actor-critic framework [10].In RL, an agent in state $s(t)$ at time t takes action $a(t)$ following a policy π. In

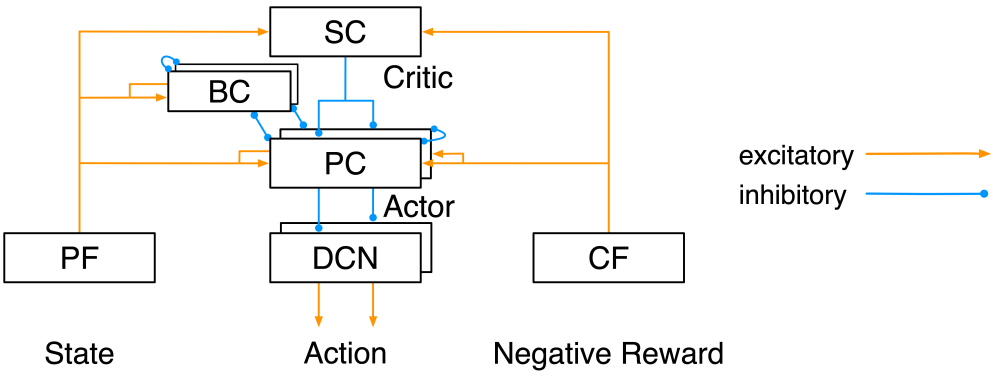

Fig. 1 Schematic of the present cerebellar network implemented as an actor-critic model. PCs serve as the actor, and SCs serve as the critic. Groups of BCs, PCs, and DCN cells are organized based on the number of actions, with each action defining a corresponding group. Orange triangle-headed arrows and blue circle-headed arrows represent excitatory synapses and inhibitory synapses, respectively. Bidirectional circle-headed arrows represent cross inhibitions.

response to action $a(t)$, the environment changes its state and returns a reward $r(t)$ to the agent. The agent learns to find the optimal policy that maximizes the expected future reward. Specifically, the agent of the actor-critic model consists of two modules: an actor and a critic. The actor calculates preferences of actions $h_\alpha(t)$ and decides on action $a(t)$, while the critic computes the state value $V(t)$ which refers to the expected cumulative future reward. The temporal difference error (TD error) is calculated from the returned reward and the state value for updating the inner parameters of both the critic and the actor.

In our implementation, $s(t)$ was transmitted through PFs to PCs, BCs, and SCs. The actor and the critic were PCs and SCs, respectively. Practically, the activity of PCs represented the avoidance of actions $-h_\alpha(t)$, while the activity of SCs represented the additive inverse of the state value $-V(t)$ rather than TD error which was represented by MLIs in a previous theoretical model [8]. BCs contributed to action selection by modulating the activity of PCs [13, 14]. Then, the most activated DCN group decided on an actual action $a(t)$. In response to the action, reward information was transmitted through CFs to PCs and SCs.

1.1.1 Representation of states and rewards

States were represented by PFs. For simplicity, we arranged the PFs on a two-dimensional grid plane composed of a number of small tiles. We named this plane the PF plane. Each tile had an individual receptive field that determined the PF's firing rate based on the observed state, as described in Section 3.1.

In the present study, rewards $r(t)$ were actually considered as punishments so that $r(t) \leq 0$. CFs delivered the information as excitatory signals by flipping the sign as follows:

$$\mathrm{CF}(t) = -r(t). \tag{1}$$

1.1.2 Critic

We designed the temporal activity of SCs to approximate the state value. Strictly speaking, the additive inverse of the state value was calculated by SCs as follows:

$$-V(t) = \nu \mathrm{SC}(t) - V_0, \tag{2}$$

where ν, $\mathrm{SC}(t)$, and V_0 represent a scaling factor, the temporal activity of SCs, and the baseline of the state value, respectively. The baseline V_0 was set at ν times the average value of $\mathrm{SC}(t)$, which was calculated during the latter half of an episode without learning. $\mathrm{SC}(t)$ was calculated as follows:

$$\mathrm{SC}(t) = \frac{1}{N_{\mathrm{SC}}} \sum_i \mathrm{SC}_i(t), \tag{3}$$

$$\mathrm{SC}_i(t) = \frac{\mathrm{SC}_{i,\mathrm{decay}}(t) - \mathrm{SC}_{i,\mathrm{rise}}(t)}{\tau_{\mathrm{SC,decay}} - \tau_{\mathrm{SC,rise}}}, \tag{4}$$

$$\frac{d\mathrm{SC}_{i,l}(t)}{dt} = -\frac{\mathrm{SC}_{i,l}(t)}{\tau_{\mathrm{SC},l}} + S_i(t), \tag{5}$$

where N_{SC} and $\mathrm{SC}_i(t)$ represent the number of SCs and the temporal activity of ith SC, respectively. l is a label in $\{\mathrm{rise}, \mathrm{decay}\}$. The temporal activity of ith SC rises with a time constant $\tau_{\mathrm{SC,rise}}$ and decays to 0 with a time constant $\tau_{\mathrm{SC,decay}}$. Spike train of the ith neuron is defined as $S_i(t) = \sum_f \delta_D(t - t_i^{(f)})$ with the Dirac delta function $\delta_D(t)$ and the fth spike of the ith neuron.

1.1.3 Actor and action selection

PCs performed the actor and determined the action taken [15]. We interpreted the temporal activity of PCs as expressing avoidance of certain actions. Specifically, the PCs in our model compute the avoidance of an action $h_{\bar{\alpha}}(t)$ instead of the preference for an action $h_\alpha(t)$. Therefore, the taken action $a(t)$ was determined by the least-avoided action in action space $\mathcal{A}$ at time t.

$$a(t) = \underset{\alpha \in \mathcal{A}}{\mathrm{argmin}}\ h_{\bar{\alpha}}(t). \tag{6}$$

The avoidance of action $\alpha \in \mathcal{A}$ was represented by the activity of PCs in $\bar{\alpha}$ group as follows :

$$h_{\bar{\alpha}}(t) = \frac{1}{N_{\mathbf{PC}_{\bar{\alpha}}}} \sum_{i \in \mathbf{PC}_{\bar{\alpha}}} \mathbf{PC}_i(t), \tag{7}$$

where $N_{\mathbf{PC}_{\bar{\alpha}}}$ is the number of neurons in the group $\mathrm{PC}_{\bar{\alpha}}$. $\mathrm{PC}_i(t)$ was the temporal activity of ith PC defined as follows:

$$\mathrm{PC}_i(t) = \frac{\mathrm{PC}_{i,\mathrm{decay}}(t) - \mathrm{PC}_{i,\mathrm{rise}}(t)}{\tau_{\mathrm{PC,decay}} - \tau_{\mathrm{PC,rise}}}, \tag{8}$$

$$\tau_l \frac{d\mathrm{PC}_{i,l}(t)}{dt} = -\mathrm{PC}_{i,l}(t) + S_i(t), \tag{9}$$

where l, τ_l, and $S_i(t)$ represent the label $l \in \{\mathrm{decay}, \mathrm{rise}\}$, the time constant for l, and the spike train of the ith PC, respectively. Practically, the actual action was generated at the downstream DCN (Section 3.1).

Those spike activities of PCs were used for updating synaptic weights to realize which action was selected. It was vital that one PC group pauses while the other PC groups activate. We referred to this pausing behavior as a "dent", while activation of neurons in a specific area was referred to as a "bump" by Frémaux et al. [10]. The inhibitory loop between BCs and PCs helped to pause the single group solely to make a dent. Although the original spiking actor-critic framework supports continuous action space, the present model only supports discrete action space with only two actions, because of the difficulty of making a single dent on PCs.

1.1.4 Weight-update rules

Our weight-update rule was based on TD long-term potentiation (LTP) which had been proposed in a previous spiking actor-critic model [10]. However, unlike the previous model, which calculated the time derivative of the state value directly to compute the TD error, our model did not calculate it, as there was no component in the molecular layer of our model that explicitly handled the TD error.

To address this problem, we conducted an eligibility trace and transformed the weight-update rule. As a result, our weight-update rule for the state value was as follows:

$$\frac{d\Delta w_{i,j}^n}{dt} = \alpha \left(R(t) z_{i,j}(t) + \frac{\tau_r - \tau_z}{\tau_r \tau_z} V(t) z_{i,j}(t) - V(t) f_{i,j}(t) \right), \tag{10}$$

$$\frac{dz_{i,j}(t)}{dt} = -\frac{z_{i,j}(t)}{\tau_z} + f_{i,j}(t), \tag{11}$$

$$f_{i,j}(t) = S_i(t) \kappa_{i,j}(t), \tag{12}$$

$$\frac{d\kappa_{i,j}(t)}{dt} = -\frac{\kappa_{i,j}(t)}{\tau_\kappa} + S_j(t), \tag{13}$$

where $\Delta w_{i,j}^n$, α, $R(t)$, $z_{i,j}(t)$, τ_r, τ_z, $f_{i,j}(t)$, $S_i(t)$, $S_j(t)$, $\kappa_{i,j}(t)$, and τ_κ are the amount of synaptic change in nth episode, the learning rate, the negative reward, the eligibility trace for $w_{i,j}$, the reward discount time constant, the decay time constant of eligibility trace, the forcing term of eligibility trace, the postsynaptic spike train, the presynaptic spike train, the window function for spike timing dependent plasticity, and the decay time constant of the window function. According to the previous model [10], the eligibility term memorizes the history of inputs before the time of the last postsynaptic neuron spike, $\kappa_{i,j}(t)$ is reset to 0 after the postsynaptic neuron spike. The relationship of our weight-update rule to the original weight-update rule [10, 16] is described in Section 1 of the Supplementary Information. The synaptic weight $w_{i,j}$ was updated at the end of every episode T_{end} by $w_{i,j}^{n+1} = w_{i,j}^n + \eta \Delta w_{i,j}^n(T_{\mathrm{end}})$ with learning rate η, and the synaptic weight was bounded between $[0, 1]$. Since SCs represented the additive

inverse state value and we denoted the negative reward $r(t) = -\mathrm{CF}(t)$, the equation for the PF–SC synapses was expressed as follows:

$$\frac{d\Delta w_{i,j}^{n}}{dt} = \mathrm{CF}(t) z_{i,j}(t) - \frac{\tau_r - \tau_z}{\tau_r \tau_z} V(t) z_{i,j}(t) + V(t) f_{i,j}(t). \tag{14}$$

On the other hand, the weight-update rule for PF–PC synapses was expressed as follows:

$$\frac{d\Delta w_{i,j}^{n}}{dt} = -\mathrm{CF}(t) z_{i,j}(t) + \frac{\tau_r - \tau_z}{\tau_r \tau_z} V(t) z_{i,j}(t) - V(t) f_{i,j}(t). \tag{15}$$

The specific learning rates for PF–SC synapses and PF–PC synapses are denoted as η_{SC} and η_{PC}, respectively.

Specifically, given that the conjunctive activation of PFs and CFs induces LTP at PF–SC synapses [17–20], our SCs learned the approximate additive inverse of the state-value. LTD at PF–SC synapses was due to PF firing, and was modulated by postsynaptic activation results in bidirectional synaptic plasticity [17–20]. At PF–PC synapses, the conjunctive activation of PFs and CFs induces LTD [21], while PF activation induces LTP. These synaptic plasticities are also modulated by inhibitions of SCs [22], and we assumed that PC activity contributes to eligibility trace and bidirectional synaptic plasticity [23, 24].

Additionally, we confirmed that our critic is capable of learning state values through a Linear Track Task simulation (Section 2 of Supplementary Information), which is a simplified version of the water-maze task used to isolate learning by the critic from that by the actor [10].

1.2 Simulation of Mountain Car Task

In order to evaluate the learning capability of our model as an RL machine, we applied our model to the mountain car task [25], a classic RL challenge, and evaluated its performance over 10 trials. In this task, an agent controls a car situated between two hills (Figure 2A). The goal is to have the car climb up the mountain to reach a flag positioned at the peak of the right hill. Due to the car's limited engine power and the hill's steepness, direct ascent is not possible. An agent observes a position $p_x(t)$ and a horizontal element of the car velocity $v_x(t)$, and decides the direction (either left or right) to push the car with a constant force $F(t) = \bar{F}a(t), \quad a(t) \in \{-1, 1\}$. In early episodes, the car was confined to moving at the bottom of the valley, and failed to reach the goal (Figure 2B). However, after hundreds of episodes, our model learned to utilize the slope for acceleration, and eventually reached the goal (Figure 2C). Specifically, we found that the trajectory involved the car ascending the right hill, then the left, and finally the right hill again. The additive inverse of the state value $-V(t)$, which was approximated by SC activity, was noisy but mostly stayed around 0 at every state before learning (Figure 2B). In a successful episode after learning (Figure 2C), the value was high at the start and decreased over time as the car approached the goal. The moving average of success rate over 10 trials progressively increased and stabilized at approximately 80% after 600 episodes (Figure 2D). These results indicate that our

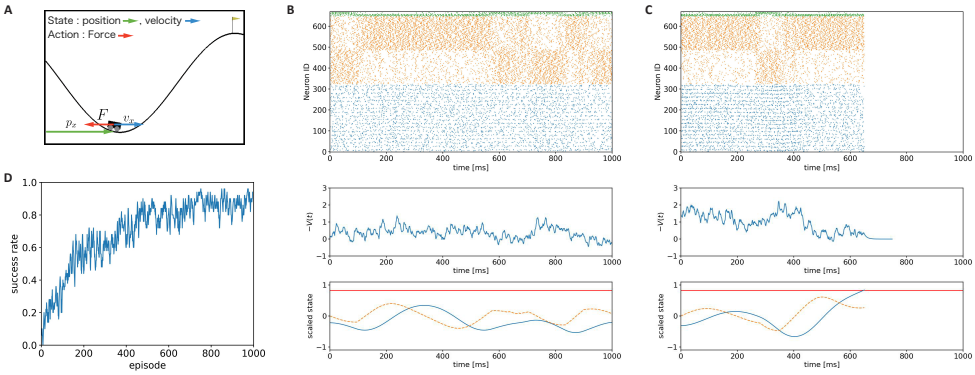

Fig. 2 Simulation of the mountain car task. **A**) Schematic of the mountain car task. p_x is the horizontal position of the cart, v_x is the horizontal velocity, and F is the external force applied to the cart by the agent. The flag marks the goal position. For clarity in the image, we adjusted the x-axis to start at 0 on the left edge. **B,C**) Comparison of behaviors in the mountain car task. **B**) Example behavior in early episodes. **C**) Example behavior in later episodes. Top panels show raster plots of SCs (blue), BCs (orange) and PCs (green). The number of these neurons are 324, 324, and 20, respectively. Middle panels show trajectory of $-V(t)$ computed from the activity of SCs. Bottom panels show trajectory of scaled states. Blue line and orange dashed line represent the normalized value of a position and that of a velocity. Red line represents the goal position. When the position reaches the goal, an episode ends while a simulation lasts 100 ms (Section 3.2.1). **D**) Success rate of mountain car task. The moving average of success rate across 10 trials with window size 5 episodes.

model solved the mountain car task, and suggest that our model successfully acted as an RL machine.

1.3 Simulation of Delay Eyeblink Conditioning Task

To examine whether our cerebellar model shows internal dynamics consistent with the biological cerebellum, we conducted simulation of a standard cerebellum-dependent motor learning task known as the delay eyeblink conditioning task [26]. In this task, an animal is presented with two types of stimuli: a conditioned stimulus (CS) and an unconditioned stimulus (US) (Figure 3A). The CS, a neutral stimulus such as a tone, initially does not trigger any response. On the other hand, the US, such as an air puff, naturally induces an unconditioned blink reflex in the animal. However, when the CS is repeatedly paired with the US, the animal begins to anticipate the air puff whenever the CS is presented. This anticipation leads the animal to blink in response to the CS alone, a learned behavior known as the conditioned response (CR). We modified the task to have two-dimensional state space: time t and eyelid position $p_e(t)$, along with two actions: opening and closing the eyelid (Figure 3B). The time axis ranges from 0 ms to 1,000 ms, corresponding to the duration of stimulus presentation, while the eyelid position is measured on a scale from fully closed (0) to fully open (1). The CS was presented from 0 ms to 1,000 ms as a time signal, and the US was introduced at 500 ms as a punishment if the eyelid was open, i.e., if $p_e(500)$ was greater than 0.1.

In early episodes, the agent preferred to open the eyelid (Figure 3C and 3B), so the agent received the US as a negative reward. As learning progressed, the anti-open

PC group became active, whereas the anti-close PC group paused as if PCs expected the US onset. Thus, the eyelid started to close before the US onset. For every 10 trials, the agent successfully acquired the CR to the CS: closing its eyelid to avoid a punishment before the arrival of the US (Figure 3E). Note that we defined a successful episode as a sufficiently closed eyelid to avoid the punishment at the US onset. When focusing on the activity changes of a single PC in the anti-close group (Figure 3F), the neuron acquired a behavior to pause before US onset. These behaviors were consistent with behavioral experiments of the delayed eyeblink conditioning task [26]. We also analyzed the activity of SCs during the task (Figure 3G). In early episodes, since the agent received the punishment as the US, the SC activity as the additive inverse of the state value $-V(t)$ increased exponentially toward the arrival of the US. These results are not inconsistent with biological experiments [27, 28], which show increases in MLIs' firing frequency from the time of CS onset to the time of US onset. In later episodes when the agent successfully closed the eyelid more than 80% of the time, the SC activity returned to zero. This returning behavior resulted from the disappearance of CF signals due to the successful avoidance of punishments.

We also conducted a simulation of the same task without PF–MLI synaptic plasticity. The activity of the anti-close PC group decreased as it approached the US onset (Figure 4A). However, the agent was not able to elicit a stable closing response in our settings. In a later episode, the time to close the eye was delayed (Figure 4B) compared to the that in the intact simulation (Figure 3D), so the agent could not close the eye sufficiently. These results are also consistent with experimental results where MLI–PC inhibitions were impaired [29]. Furthermore, these results suggested that our model successfully solved the eyeblink task with biologically plausible internal dynamics.

2 Discussion

In this study, we implemented the cerebellar spiking network model performing RL in an actor-critic manner. Our simulations confirmed the model's ability to solve a simple RL task and demonstrated a cerebellum-dependent motor leaning task with biologically plausible internal dynamics. In contrast, Hausknecht et al. [30] reported that their cerebellar model was not able to solve RL tasks, only SL tasks. This is natural, because their model was formulated as an SL machine while ignoring PF–MLI synaptic plasticity. Recently, Geminiani et al. [31] implemented a spiking network-based cerebellar model that considers synaptic plasticity at both PF–PC and PF–MLI synapses. They reported that considering multiple synaptic plasticity mechanisms enhance learning capability in SL. Thus, those computational studies considered the cerebellum as an SL machine. Spiking network models of the cerebellum as an SL machine have been used for realtime motor control and online learning of hardware robots [32, 33]. Our present cerebellar model will help those robotics studies by providing the capability of RL over SL.

In our cerebellar model, MLIs played essential roles as a critic. Recent studies have shown that MLIs actually contribute to learning in the cerebellum [27–29, 34] by their own learning capability through PF–MLI synaptic plasticity [17–19] and modulating

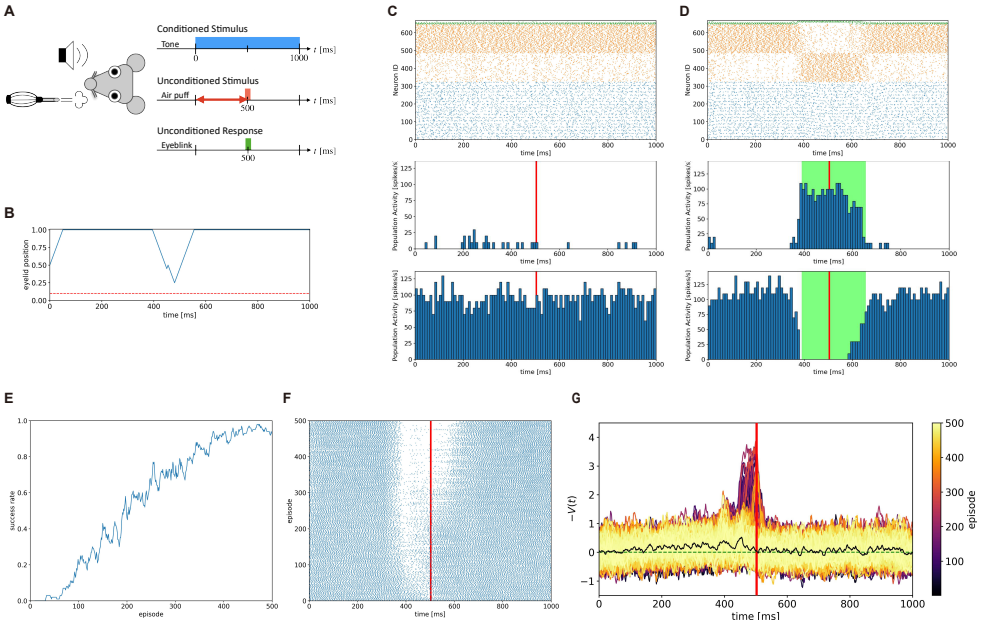

Fig. 3 Simulation of the delay eyeblink conditioning task. **A**) Schematic of eyeblink conditioning task. In this task, a neutral stimulus (CS) is presented to an agent, and during the presentation, an aversive stimulus (US) is also fed. Here, the time interval from the onset of the CS to the onset of the US is fixed. After a number of paired presentations, the agent closes their eyes (CR) in response to the CS without US. **B**) Example of a state trajectory on the state space of eyeblink conditioning task. The blue line shows an example of state trajectory. The horizontal axis represents the duration from CS onset, and the vertical axis represents the eyelid position. Dashed line represents a threshold. When the eyelid position is above the threshold at the onset of the US (t = 500 ms), the agent receives a punishment. **C,D**) Results of the delay eyeblink conditioning task in a failure episode (C) and a success episode (D). Top panels show raster plots of SCs (blue), BCs (orange) and PCs (green). The number of these neurons are 324, 324, and 20, respectively. Middle and bottom panels show peristimulus time histograms of anti-open PC population (middle) and those of anti-close PC population (bottom). Bin width is 10 ms. Red line represents the time of US onset, while green shaded intervals show when the agent close its eyelid around the US onset. **E**) The moving average of success rate over 10 trials. **F**) The activity changes of a single PC in the anti-close group. Dots represent spikes of the PC, and red line represents the time of US onset. **G**) Additive inverse of the state value during the eyeblink conditioning task. Each colored trace shows the additive inverse of the state value represented by the stellate cell neurons activity against time in 500 episodes (from black purple in the 1st trial to light yellow in the 500th trial, transitioning through shades of red, orange, and yellow). The black line shows the average of over last 10 trials. The green dashed line shows $-V(t) = 0$. The red thick line represents the time of US onset.

PF–PC synaptic plasticity [22], in addition to controlling PC activities through MLI–PC inhibition [13, 14, 20, 35–38]. In a behavioral experiment, genetically-modified mice lacking MLI–PC inhibition were able to acquire the CR, but the amplitude of eye closing was significantly lower than that of intact mice [29]. Our results of the lesion simulation in the delay eyeblink conditioning task are consistent with the experimental results. These results suggest that the cerebellum can solve the association task even

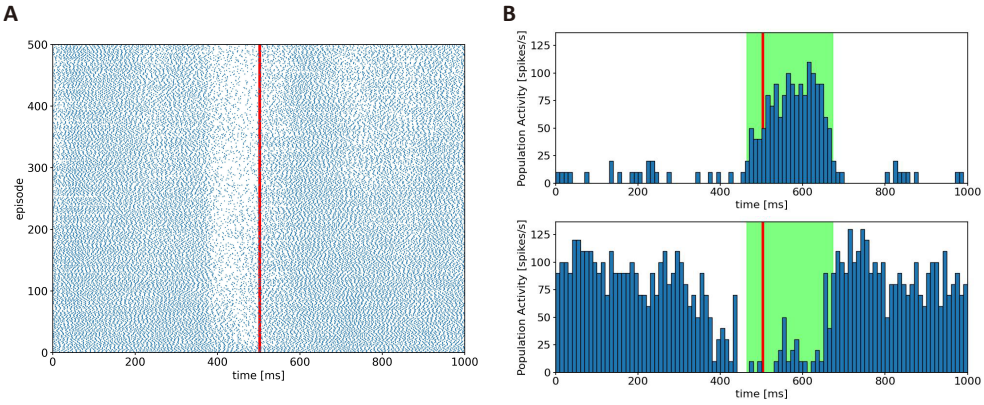

Fig. 4 Results of the delay eyeblink conditioning task without PF–SC synaptic plasticity. **A**) The activity changes of a single PC in the anti-close group. Dots represent spikes of the PC, and red line represents the time of US onset. **B**) Example peristimulus time histograms of anti-open PC population (top) and those of anti-close PC population (bottom) in the later episode. Bin width is 10 ms. Red lines represent the time of US onset, while the green shaded intervals show when the agent closes its eyelid around the US onset.

without normal SC activity, but the activity affects the performance of the acquired movement.

The present study assumed that CFs deliver negative reward information. In conventional cerebellar motor learning, CFs are considered to provide motor error signals, which are negative information. Furthermore, recent experimental results demonstrate that CFs carry not just motor-related information but also reward-related information such as reward delivery, prediction, and omission in mice [9, 39–41]. Thus, our assumption is consistent with those experimental findings. Hoang et al. [42] recapitulated the idea of negative reward information conveyed by CFs to Q-learning, which is a tabular-style reinforcement learning algorithm, and applied the Q-learning to a Go-No-go auditory discrimination task in mice. Furthermore, CF signals are provided in an all-or-none manner, and our model followed this property. However, several experiments have demonstrated that CF signals can carry graded information [43, 44], which might be useful to deliver reward information as analog values effectively.

In a standard RL algorithm called TD-learning, a reward-related signal tends to appear spontaneously at the onset of a signal that predicts the occurrence of a future reward after sufficient learning. The signal is called a reward-prediction signal or reward-prediction error. Ohmae and Medina [9] observed such reward-prediction signals conveyed by CFs at the CS onset after sufficient training of delay eyeblink conditioning in mice. Indeed, such prediction-error signals are prepared outside of the cerebellum. Therefore, in the present study, we did not consider such prediction error-type CF signals. If we incorporated them, those signals could have acted to control the balance between exploration and exploitation [12] while modulating the fluctuation of

the internal network state. Another potential role would be to represent the most distant state from the goal state for reward delivery. Nevertheless, further investigation will be necessary to study the functional role of the prediction error-type CF signals.

If both the cerebellum and the basal ganglia perform RL, how do these regions cooperate for learning? One possible interpretation is hierarchical RL (HRL) [12] as discussed in our previous research [45]. HRL considers two RL machines organized hierarchically. The higher RL machine breaks down a complex task into simpler, smaller subtasks. The lower RL machine tries to solve these subtasks one by one. Eventually, HRL involves creating a hierarchy of policies that makes the learning process more efficient and scalable [12, 46, 47]. If the basal ganglia and the cerebellum cooperate to perform HRL, the resulting network would demonstrate powerful learning capabilities.

To our knowledge, our model is the first to implement cerebellar RL with spiking neurons. Our research provides an implementation that supports cerebellar RL theory to challenge the traditional view of the cerebellum as primarily a SL machine. Furthermore, our interpretation of a potential cooperative role of the basal ganglia and the cerebellum as HRL sheds light on how the multiple brain regions act together synergistically as a whole.

3 Methods

3.1 Implementations

First, we discretized a state space and mapped it on a PF plane with grid size $N_x \times N_y$. A tile positioned at (i, j) on the PF plane had its own responsible state $\mathbf{o}_{i,j}$ as follows:

$$\mathbf{o}_{i,j} = [i\Delta s_x, j\Delta s_y]^T, \tag{16}$$

$$\Delta s_x = \frac{s_{x,\max} - s_{y,\min}}{N_x}, \quad \Delta s_y = \frac{s_{y,\max} - s_{y,\min}}{N_y} \tag{17}$$

where x and y are the axes of the PF plane, and $(s_{x,\min}, s_{x,\max})$ and $(s_{y,\min}, s_{y,\max})$ represent the ranges of state x and y, respectively. When the agent observed state $\mathbf{o}(t) = [x(t), y(t)]^T$, PFs on a tile at (i, j) fired as an inhomogeneous Poisson spike generator with a firing rate $\rho_{i,j}(\mathbf{o}(t))$ defined as follows :

$$\rho_{i,j}(\mathbf{o}(t)) = \frac{\rho_{\mathrm{PF}}}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{||\mathbf{o}(t) - \mathbf{o}_{i,j}||^2}{2\sigma^2}\right), \tag{18}$$

where $\rho_{\mathrm{PF}} = 50$ Hz is a scaling factor of firing rate, and $\sigma = 0.5$ is a standard deviation. To maintain the total activity of all PFs when the agent is at the edge of a state space, we added "margin" tiles as an outer frame. Although we modeled PFs as Poisson spike generators for simplicity, more realistic implementations are reported [48].

We utilized a Leaky Integrate-and-Fire model (LIF) to SCs, BCs, and PCs. A LIF model was implemented as follows.

$$C_m \frac{du_i}{dt} = -g_{\mathrm{L}}(u_i - E_{\mathrm{L}}) + I_{i,\mathrm{syn}}(t) + I_{\mathrm{ext}} + I_{i,\mathrm{noise}}(t), \tag{19}$$

$$t_i^{(f)} : u_i\left(t_i^{(f)}\right) = U_{\mathrm{th}}, \tag{20}$$

$$\lim_{t_i^{(f)} \to t,\ t > t_i^{(f)}} u_i(t) = U_{\mathrm{reset}}, \tag{21}$$

where C_m, u_i, t, g_{L}, E_{L}, $I_{i,\mathrm{syn}}$, I_{ext}, and $I_{i,\mathrm{noise}}$ represent the membrane capacitance, the membrane potential of the ith neuron, time, the leak conductance, the resting potential, the synaptic current to the ith neuron, the external input current, and the noise current to the ith neuron, respectively. When the membrane potential of the ith neuron reaches the threshold potential U_{th}, that time is denoted as $t_i^{(f)}$, and the neuron fires the fth spike and resets its membrane potential to the reset potential U_{reset}. A synaptic current was defined as follows:

$$I_{i,\mathrm{syn}}(t) = \sum_c s_c w_{i,j} I_{i,j}(t), \tag{22}$$

$$\frac{dI_{i,j}(t)}{dt} = -\frac{I_{i,j}(t)}{\tau_c} + S_j(t), \tag{23}$$

where c is a synapse label, s_c, $w_{i,j}$, τ_c, and $S_j(t)$ are the scaling factor, synaptic weight between the jth presynaptic neuron and the ith postsynaptic neuron, decay time constant, and spike train of the jth presynaptic neuron, respectively. Whether it is an inhibitory or excitatory current is determined by the sign of the scaling factor. Noise current $I_{i,\mathrm{noise}}(t)$ was modeled as follows:

$$\frac{dI_{i,\mathrm{noise}}}{dt} = -\frac{I_{i,\mathrm{noise}}}{\tau_{\mathrm{noise}}} + \zeta n(t), \tag{24}$$

where τ_{noise} is a time constant, and ζ is an amplitude factor. $n(t)$ is a uniform random variable in the range $[-1, 1]$. Parameters of all neurons were set as listed in Table 1, and parameters of all synapses were set as listed in Table 2.

Table 1 Neuron-specific parameters. These parameters were adjusted from those used in our previous model [49].

Type	SC	BC	PC
C_m [pF]	107	107	107
g_{L} [mS]	2.32	2.32	2.32
E_L [mV]	−68.0	−68.0	−65.0
U_{th} [mV]	−55.0	−55.0	−55.0
U_{r} [mV]	−70.0	−70.0	−66.0
I_{ext} [pA]	30.0	30.0	30.0
ζ [pA]	5.0	1.0	5.0

As mentioned above, PFs were distributed on the two-dimensional grid plane, and the grid size was determined by each task. Each tile contained 100 PFs. SCs were prepared 1 per tile. Each SC positioned at $(i\Delta s_x, j\Delta s_y)$ could receive excitatory inputs from PFs located in $s_{x,\min} \leq x \leq s_{x,\max}$ and $(j-1)\Delta s_y \leq y \leq (j+1)\Delta s_y$. The number of BCs was the same as that of SCs, and the number of PCs was 20. Both BCs and

Table 2 Synaptic parameters for each connection type. A hyphen in a weight cell means it will be modified because of synaptic plasticity.

Connection Type	scaling factor	weight	τ_c [ms]	probability
PF–SC	200.0	-	8.3	0.1
PF–BC	10.0	1.0	8.3	0.5
PF–PC	5.0	-	8.3	0.5
SC–PC	−0.05	0.03	10.0	0.5
BC–BC	−2.0	1.0	10.0	0.3
BC–PC	−1.0	1.0	10.0	0.3
PC–BC	−2.0	1.0	10.0	0.8
PC–PC	−19.8	1.0	5.0	1.0

PCs could receive excitatory PF inputs from all tiles. The BC–BC, BC–PC, PC–BC, and PC–PC connections were constructed based on the grouping manner mentioned in Section 1.1. The actual synapses were formed stochastically (Table 2). In this study, we divided BCs and PCs into two groups for all tasks. Additionally, for the sake of simplicity, we assumed that all CFs synchronized their activity so that our model had a single CF.

Synaptic plasticity was applied to PF–SC and PF–PC synapses. The details of the weight-update rules are described in Equations (14) and (15).

Finally, the outcome of actions were generated by DCN cells prepared in an equal number of PCs. Each DCN cell was inhibited by its counterpart PC. This configuration means that the activity of each DCN cell, denoted as DCN_i, was modulated by the corresponding PC activity $\mathrm{PC}_i(t)$ defined by Equation 8, as follows:

$$\tau_{\mathrm{DCN}} \frac{d\mathrm{DCN}_i(t)}{dt} = -\left(\mathrm{DCN}_i(t) - \mathrm{DCN}_0\right) - \nu \mathrm{PC}_i(t), \tag{25}$$

where τ_{DCN}, DCN_0, and ν represent the decay time constant, baseline activity of DCN, and a scaling factor, respectively.

Since PCs in the group $\mathrm{PC}_{\bar{\alpha}}$ represented the avoidance of action α and inhibited DCN cells, the activity of DCN group DCN_α represented the preference of actions $h_\alpha(t)$ as follows:

$$h_\alpha(t) = \frac{1}{N_{\mathrm{DCN}_\alpha}} \sum_{i \in \mathrm{DCN}_\alpha} \mathrm{DCN}_i(t), \tag{26}$$

where N_{DCN_α} is the number of neurons in the group DCN_α, and $\mathrm{DCN}_i(t)$ is the activity of ith DCN in the group DCN_α corresponding to action α. Therefore, the most activated DCN group represented the most preferred action so that the final outcome of actions was chosen as follows:

$$a(t) = \underset{\alpha \in \mathcal{A}}{\operatorname{argmax}}\ h_\alpha(t). \tag{27}$$

3.2 Task Definitions

3.2.1 Pre- and post-episode processes

As a pre-episode process, at the start of each episode, a free run for 200 ms was carried out to discard initial transient activity that could affect learning. Then, at the end of each episode, a simulation continued for 100 ms while stopping the firing of PFs to update synaptic weights. These pre- and post-episode processes were performed in all tasks.

3.2.2 Mountain Car Task

The dynamics of the car are described by the following differential equations :

$$\frac{dv_x(t)}{dt} = v_x(t) + \bar{F}a(t) - g\cos(3p_x(t)) \tag{28}$$

$$\frac{dp_x(t)}{dt} = v_x(t), \tag{29}$$

where $\bar{F} = 0.001$ is a force amplitude, $a(t) \in \{-1, 1\}$ represents an action as the force direction, $g = 0.0025$ represents the gravity. Observation spaces of the velocity and position are bounded by $[-1.2, 0.6]$ and $[-0.07, 0.07]$, respectively. When the car collides with the wall, its velocity is immediately reset to 0 without any negative reward associated with the collision. An agent receives constant negative reward at every step in the original settings. However, the maximum firing frequency of CF, which is approximately 10 Hz [50], was not enough to represent the constant negative reward. Therefore, in this study, we modeled a new negative reward function for CF. CF fired when the car was far from the goal and the speed was slow. Thus, the firing rate $\nu(t)$ was defined as follows:

$$\nu(t) = \bar{\nu}\left(1 - \frac{p_{\text{goal}} - p_x(t)}{p_{\text{goal}} - p_{\text{wall}}}\right)\left(1 - \frac{|v_x(t)|}{v_{\max}}\right), \tag{30}$$

where $\bar{\nu}$, p_{goal}, p_{wall}, and $v_{\max}$ are max firing rate of 5 Hz, goal position, left side wall position, and limit of speed respectively. Then a negative reward function is defined as follows:

$$R(t) = \begin{cases} -5 & \text{when CF fires at time } t, \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

The car starts from the bottom of the valley with a velocity of 0 in every episode. The action $a(t)$ represents the direction in which the agent pushes the car. An episode ends when the car reaches the goal or when the elapsed time exceeds 1,000 ms, with no special negative reward given.

The grid size of the PF plane was 16×16. The parameters of the state value function, as defined in Equation (2), were $\nu = 200$ and $V_0 = 2.48$. The reward discount time constant τ_r, the decay time constant of eligibility trace τ_z, and the decay time

constant of the window function τ_κ were set to 100 ms, 20 ms, and 20 ms, respectively. Initial synaptic weights of PF–SC and PF–PC were 0.05 and 0.8, respectively. Both learning rates for the critic and the agent were 0.4.

We conducted 10 trials of 1,000 episodes each. We calculated the changes in success rate for each trial using a moving average of results (success=1, failure=0) with a window size of 10 episodes, and then calculated the average across trials.

3.2.3 Delay Eyeblink Conditioning Task

The dynamics of the eyelid is described as follows:

$$\frac{dp_e(t)}{dt} = 0.01a(t), \tag{32}$$

where $a(t) \in \{-1, 1\}$ represents the action the agent takes at time t. Also, negative reward is defined as follows:

$$r(t) = \begin{cases} -5 & \text{when } t = 500 \text{ and } p_e(t) > 0.1, \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

The grid size of the PF plane was 16×16. The parameters of the state value function, which defined in Equation (2), were $\nu = 200$ and $V_0 = 2.48$. The reward discount time constant τ_r, the decay time constant of eligibility trace τ_z, and the decay time constant of the window function τ_κ were set to 100 ms, 20 ms, and 20 ms, respectively. To prioritize the 'open' action, we initially set synaptic weights for PF–$\mathrm{PC}_{\text{anti close}}$ to 0.8, while PF–$\mathrm{PC}_{\text{anti open}}$ was set to 0.7. We conducted the simulation for 500 episodes with learning rates $\eta_{\mathrm{SC}} = 0.05$ and $\eta_{\mathrm{PC}} = 0.3$.

The whole code of our model and all environments were written in C++. All ordinary differential equations were solved numerically with a forward Euler method with a temporal resolution of 1.0 ms.

Code Availability

During editorial process and peer-review process, the codes of this study are available from the corresponding author upon request. We anticipate that the final version will be ready and publicly available on GitHub upon acceptance.

Conflict of interest statement

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

Author contributions

TY conceived and designed the research. RK and HY contributed the formulation and the implementation of the algorithms. RK wrote the code, performed simulation, and

analyzed the data. RK and TY wrote the original draft. RK, HY, and TY discussed the draft and revised it. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by JSPS KAKENHI Grant numbers JP22KJ1372 and JP22H05161. Part of this study was supported by the MEXT Program for Promoting Researches on the Supercomputer Fugaku hp220162.

Acknowledgment

We would like to thank Profs. Kazuo Kitamura, Jun Igarashi, Shogo Ohmae, and Kenji Doya for their discussions and insights.

References

[1] Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).

[2] Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).

[3] Schultz, W., Dayan, P. & Montague, P. R. A Neural Substrate of Prediction and Reward. *Science* **275**, 1593–1599 (1997).

[4] Doya, K. Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology* **10**, 732–739 (2000).

[5] Marr, D. A theory of cerebellar cortex. *The Journal of Physiology* **202**, 437–470 (1969).

[6] Ito, M. Neurophysiological aspects of the cerebellar motor control system. *International journal of neurology* **7**, 162–76 (1970).

[7] Albus, J. S. A theory of cerebellar function. *Mathematical Biosciences* **10**, 25–61 (1971).

[8] Yamazaki, T. & Lennon, W. Revisiting a theory of cerebellar cortex. *Neuroscience Research* **148**, 1–8 (2019).

[9] Ohmae, S. & Medina, J. F. Climbing fibers encode a temporal-difference prediction error during cerebellar learning in mice. *Nature Neuroscience* **18**, 1798–1803 (2015).

[10] Frémaux, N., Sprekeler, H. & Gerstner, W. Reinforcement Learning Using a Continuous Time Actor-Critic Framework with Spiking Neurons. *PLoS Computational Biology* **9**, e1003024 (2013).

[11] Raymond, J. L. & Medina, J. F. Computational Principles of Supervised Learning in the Cerebellum. *Annual Review of Neuroscience* **41**, 233–253 (2018).

[12] Sutton, R. S. & Barto, A. G. *Reinforcement learning: an introduction* Second edition edn. Adaptive computation and machine learning series (The MIT Press, Cambridge, Massachusetts, 2018).

[13] Dizon, M. J. & Khodakhah, K. The Role of Interneurons in Shaping Purkinje Cell Responses in the Cerebellar Cortex. *The Journal of Neuroscience* **31**, 10463–10473 (2011).

[14] Brown, A. M. *et al.* Molecular layer interneurons shape the spike activity of cerebellar Purkinje cells. *Scientific Reports* **9**, 1742 (2019).

[15] Ito, M. *The Cerebellum and Neural Control* (Raven Press, New York, 1984).

[16] Doya, K. Reinforcement Learning in Continuous Time and Space. *Neural Computation* **12**, 219–245 (2000).

[17] Jörntell, H. & Ekerot, C.-F. Reciprocal Bidirectional Plasticity of Parallel Fiber Receptive Fields in Cerebellar Purkinje Cells and Their Afferent Interneurons. *Neuron* **34**, 797–806 (2002).

[18] Jörntell, H. & Ekerot, C.-F. Receptive Field Plasticity Profoundly Alters the Cutaneous Parallel Fiber Synaptic Input to Cerebellar Interneurons In Vivo. *The Journal of Neuroscience* **23**, 9620–9631 (2003).

[19] Jirenhed, D.-A., Bengtsson, F. & Jörntell, H. Parallel fiber and climbing fiber responses in rat cerebellar cortical neurons in vivo. *Frontiers in Systems Neuroscience* **7**, 16 (2013).

[20] Liu, S. J. *Stellate Cells: Synaptic Processing and Plasticity*, Vol. 2, 809–828 (Springer, 2013).

[21] D'Angelo, E. *The Organization of Plasticity in the Cerebellar Cortex: From Synapses to Control*, Vol. 210 of *Progress in Brain Research*, 31–58 (Elsevier, 2014).

[22] Rowan, M. J. *et al.* Graded Control of Climbing-Fiber-Mediated Plasticity and Learning by Inhibition in the Cerebellum. *Neuron* **99**, 999–1015.e6 (2018).

[23] Bonnan, A., Rowan, M. M. J., Baker, C. A., Bolton, M. M. & Christie, J. M. Autonomous Purkinje cell activation instructs bidirectional motor learning through evoked dendritic calcium signaling. *Nature Communications* **12**, 2153

(2021).

[24] Silva, N. T., Ramírez-Buriticá, J., Pritchett, D. L. & Carey, M. R. Climbing fibers provide essential instructive signals for associative learning. *Nature Neuroscience* 1–12 (2024).

[25] Towers, M. *et al.* Gymnasium (2023). URL https://zenodo.org/record/8127025.

[26] Jirenhed, D.-A., Bengtsson, F. & Hesslow, G. Acquisition, Extinction, and Reacquisition of a Cerebellar Cortical Memory Trace. *Journal of Neuroscience* **27**, 2493–2502 (2007).

[27] ten Brinke, M. *et al.* Evolving Models of Pavlovian Conditioning: Cerebellar Cortical Dynamics in Awake Behaving Mice. *Cell Reports* **13**, 1977–1988 (2015).

[28] Ma, M. *et al.* Molecular layer interneurons in the cerebellum encode for valence in associative learning. *Nature Communications* **11**, 4217 (2020).

[29] Boele, H.-J. *et al.* Impact of parallel fiber to Purkinje cell long-term depression is unmasked in absence of inhibitory input. *Science Advances* **4**, eaas9426 (2018).

[30] Hausknecht, M., Li, W.-K., Mauk, M. & Stone, P. Machine Learning Capabilities of a Simulated Cerebellum. *IEEE Transactions on Neural Networks and Learning Systems* **28**, 510–522 (2017).

[31] Geminiani, A. *et al.* Mesoscale simulations predict the role of synergistic cerebellar plasticity during classical eyeblink conditioning. *PLOS Computational Biology* **20**, e1011277 (2024).

[32] Abadía, I., Naveros, F., Ros, E., Carrillo, R. R. & Luque, N. R. A cerebellar-based solution to the nondeterministic time delay problem in robotic control. *Science Robotics* **6**, eabf2756 (2021).

[33] Antonietti, A. *et al.* Brain-inspired spiking neural network controller for a neurorobotic whisker system. *Frontiers in Neurorobotics* **16** (2022).

[34] Johansson, F., Jirenhed, D.-A., Rasmussen, A., Zucca, R. & Hesslow, G. Memory trace and timing mechanism localized to cerebellar Purkinje cells. *Proceedings of the National Academy of Sciences* **111**, 14930–14934 (2014).

[35] Wulff, P. *et al.* Synaptic inhibition of Purkinje cells mediates consolidation of vestibulo-cerebellar motor learning. *Nature Neuroscience* **12**, 1042–1049 (2009).

[36] Lee, K. *et al.* Circuit Mechanisms Underlying Motor Memory Formation in the Cerebellum. *Neuron* **86**, 529–540 (2015).

[37] Hirano, T., Yamazaki, Y. & Nakamura, Y. LTD, RP, and Motor Learning. *The Cerebellum* **15**, 51–53 (2016).

[38] Jelitai, M., Puggioni, P., Ishikawa, T., Rinaldi, A. & Duguid, I. Dendritic excitation–inhibition balance shapes cerebellar output during motor behaviour. *Nature Communications* **7**, 13722 (2016).

[39] Heffley, W. & Hull, C. Classical conditioning drives learned reward prediction signals in climbing fibers across the lateral cerebellum. *eLife* **8**, e46764 (2019).

[40] Kostadinov, D., Beau, M., Blanco-Pozo, M. & Häusser, M. Predictive and reactive reward signals conveyed by climbing fiber inputs to cerebellar Purkinje cells. *Nature Neuroscience* **22**, 950–962 (2019).

[41] Ikezoe, K. *et al.* Cerebellar climbing fibers multiplex movement and reward signals during a voluntary movement task in mice. *Communications Biology* **6**, 924 (2023).

[42] Hoang, H. *et al.* Negative reward-prediction errors of climbing fiber inputs for cerebellar reinforcement learning algorithm. *bioRxiv* (2023).

[43] Najafi, F. & Medina, J. F. Beyond "all-or-nothing" climbing fibers: graded representation of teaching signals in Purkinje cells. *Frontiers in Neural Circuits* **7**, 115 (2013).

[44] Zang, Y. & De Schutter, E. Climbing Fibers Provide Graded Error Signals in Cerebellar Learning. *Frontiers in Systems Neuroscience* **13**, 46 (2019).

[45] Yamazaki, T. *Evolution of the Marr-Albus-Ito model*, 239–255. Contemporary Clinical Neuroscience (Springer International Publishing, 2021).

[46] Morimoto, J. & Doya, K. Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems* **36**, 37–51 (2001).

[47] Kulkarni, T. D., Narasimhan, K. R., Saeedi, A. & Tenenbaum, J. B. *Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation*, 3682–3690. NIPS'16 (Curran Associates Inc., Red Hook, NY, USA, 2016).

[48] Rössert, C., Solinas, S., D'Angelo, E., Dean, P. & Porrill, J. Model cerebellar granule cells can faithfully transmit modulated firing rate signals. *Frontiers in Cellular Neuroscience* **8** (2014).

[49] Kuriyama, R., Casellato, C., D'Angelo, E. & Yamazaki, T. Real-Time Simulation of a Cerebellar Scaffold Model on Graphics Processing Units. *Frontiers in Cellular Neuroscience* **15**, 623552 (2021).

[50] Zeeuw, C. I. D. *et al.* Microcircuitry and function of the inferior olive. *Trends in Neurosciences* **21**, 391–400 (1998).