

Fast and slow synaptic plasticity enables concurrent control and learning

Reviewed Preprint

v1 • March 24, 2025

Not revised

Brendan A Bicknell , Peter E Latham

Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom

 https://en.wikipedia.org/wiki/Open_access Copyright information

eLife Assessment

This paper provides an **important** proposal for why learning can be much faster and more accurate if synapses have a fast component that immediately corrects errors, as well as a slower component that corrects behavior averaged over a longer timescale. It is **convincingly** shown that integrating these two learning timescales improves performance compared to classical strategies, particularly in terms of robustness and generalization when learning new target signals. However, the biological plausibility and justification for the proposed rapid learning mechanism require further elaboration and supporting mechanistic examples.

<https://doi.org/10.7554/eLife.105043.1.sa2>

Abstract

During many tasks the brain receives real-time feedback about performance. What should it do with that information, at the synaptic level, so that tasks can be performed as well as possible? The conventional answer is that it should learn by incrementally adjusting synaptic strengths. We show, however, that learning on its own is severely suboptimal. To maximize performance, synaptic plasticity should also operate on a much faster timescale – essentially, the synaptic weights should act as a control signal. We propose a normative plasticity rule that embodies this principle. In this, fast synaptic weight changes greedily suppress downstream errors, while slow synaptic weight changes implement statistically optimal learning. This enables near-perfect task performance immediately, efficient task execution on longer timescales, and confers robustness to noise and other perturbations. Applied in a cerebellar microcircuit model, the theory explains longstanding experimental observations and makes novel testable predictions.

Introduction

The standard view of learning is that synaptic strengths are adjusted to minimize some loss – typically the performance on a task, or set of tasks (Richards and Kording 2023 [↗](#); Bredenberg and Savin 2023 [↗](#)). From this perspective, given the complexity of the nervous system and the range of tasks it must perform, conventional wisdom is that synaptic strengths change slowly. This makes sense for tasks with a discrete set of responses: Do I lick left or right in response to a stimulus? Is that grating oriented to the right or left of vertical? However, for naturalistic tasks with continuous output, such as reaching for an object, feedback about performance could be used to adjust weights on much faster timescales. In fact, for sufficiently fast feedback compared to the timescale over which the world changes, fast weight changes could, at least in principle, enable near-perfect performance.

Here we propose that synapses can maximize task performance by adjusting weights on two timescales. On fast timescales, synapses can use real-time feedback to suppress immediate errors. As long as the world changes relatively smoothly over time, if feedback at one moment indicates that neural output is too high or low, synaptic strengths can be transiently decreased or increased to compensate. On slow timescales, synapses can use the same feedback signal for statistically optimal learning.

We first illustrate the main concept with a toy model, demonstrating how two timescales of plasticity can work in concert. We then consider a more realistic model of a neuron, where synapses dynamically integrate input and feedback to extract useful learning signals from noisy observations. We derive synaptic update rules for this model by framing synaptic plasticity as an optimal control problem. This leads to substantial improvements over classical gradient-based learning. We then generalize the theory to incorporate small populations of neurons and delayed feedback transmitted via spikes. Applied to a model of temporal processing in the cerebellum, our theory provides normative explanations for common experimental observations, and makes novel testable predictions. Altogether, these results provide a principled account of how the brain can exploit multiple timescales of plasticity for efficient online adaptation and learning.

Results

We propose that neurons can optimize their performance by having two separate timescales of synaptic plasticity. To illustrate the general idea, consider a simple online linear regression problem. We model a single neuron that must learn to transform a set of time-varying input rates, $v_i(t)$, into a target output, $y^*(t)$. The output of the neuron is given by

$$y(t) = \sum_{i=1}^N w_i(t) v_i(t), \quad (1)$$

with parameters w_i denoting its tunable synaptic weights. The classical solution to this problem is to discretize time and iteratively modify the weights using the delta rule (Widrow and Hoff 1960 [↗](#)),

$$\Delta w_i = -\eta(y - y^*) v_i \quad (2)$$

where $\Delta w_i \equiv w_i(t + \Delta t) - w_i(t)$. This eventually minimizes the output error provided the learning rate, η , is small.

The delta rule is attractive because it is generally effective and it is biologically plausible – each synapse requires access only to its own local input rate, v_i , and a global error feedback signal, $y - y^*$. If, however, v_i and target output, y^* , change smoothly over time, temporal correlations can be exploited to dramatically improve performance. To exploit temporal correlations, we endow the synaptic weights with independent fast and slow components,

$$w_i = \bar{w}_i + \delta w_i. \quad (3)$$

The role of the fast weights, δw_i , is to transiently adjust neuronal output to suppress immediate errors – errors that can be predicted if y^* changes slowly. The role of the slow weights, \bar{w}_i , is to adapt, on a much slower timescale, to the true weights, thereby reducing the need for ongoing fast-weight corrections (**Fig. 1a** [↗](#)).

For the model described by **equation (1)** [↗](#), a simple implementation of this strategy is to observe the error at discrete times t , and then update all of the fast weights uniformly via

$$\Delta \delta w_i = -\frac{y - y^*}{N\nu}, \quad (4)$$

where $\Delta \delta w_i \equiv \delta w_i(t + \Delta t) - \delta w_i(t)$ and $\nu \equiv \frac{1}{N} \mathbb{E}(\sum_i v_i)$ is the expected input rate per synapse (computed via a time average, or assumed as a known parameter). Summed over synapses, the fast weights subtract the observed error from the output, and straightforward algebra gives us

$$y(t + \Delta t) \approx y^*(t + \Delta t), \quad (5)$$

where the approximation is good if there are a large number of synapses and Δt is small relative to how fast the target and input firing rates change (*Methods: Online linear regression*).

While suppressing error with rapidly fluctuating weights is effective, continually making these shortterm adjustments is likely to be energetically costly. This can be addressed by allowing the slow weights to learn. Intuitively, the closer the output is to the target, the less work will be needed to correct it. Seeking to make long-term adjustments to the slow weights, such that, ultimately, $\sum_i \bar{w}_i x_i = y^*$, leads to local updates,

$$\Delta \bar{w}_i = -\bar{\eta}(y - y^* - N\nu\delta w_i)v_i, \quad (6)$$

where $\bar{\eta}$ is a small learning rate (*Methods: Online linear regression*). This is similar to the usual delta rule, but with the fast weights subtracted off: it's not hard to see that $y - N\nu\delta w_i \approx \sum_i \bar{w}_i x_i$. Slow-weight learning is thus driven by a modified error signal that represents what would have been observed in the absence of all of the fast corrections, isolating the contribution of \bar{w}_i . This coupling between the two update rules makes them compatible: so long as the slow weights know what the fast weights are doing, greedily suppressing error does not come at the expense of learning – even though the error is a crucial teaching signal.

In simulations of this simple strategy, the weights that solve the regression problem can be learned just as efficiently as in the classical case, but with the striking difference that the output is pinned to the target from the outset (**Fig. 1b** [↗](#)). Thus, a neuron, or indeed an animal, could accurately execute a task even while it is still being learned. Below we develop a control theory framework that generalizes this to more realistic scenarios.

Synaptic plasticity as optimal control

We now apply this idea to a more realistic setting: a model neuron driven by multiple spike trains and subject to intrinsic noise. Similar to the regression problem above, the neuron must tune its synaptic weights to drive a downstream output, y , to match a time-varying target, y^* . The target

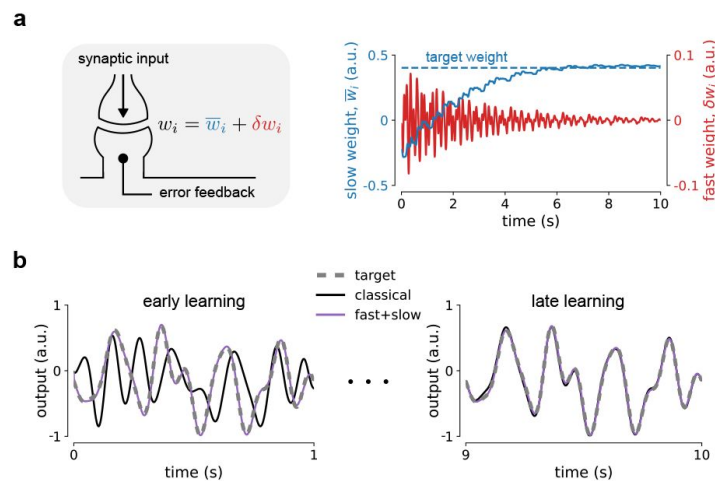


Figure 1

A theory of fast and slow synaptic plasticity.

a) Left: A synapse must make online adjustments to its strength by integrating local signals, such as its own input and error feedback. We propose that these signals can be optimally exploited through two timescales of plasticity. ‘Fast weights’, δw_i , fluctuate rapidly to suppress downstream error, whereas ‘slow weights’, \bar{w}_i , converge gradually to the values required of a given task. Right: In the toy-model simulation from panel (b), as the slow weights find the solution, fast-weight fluctuations are reduced. Shown are example weight trajectories from one randomly selected synapse out of 20. **b)** In an illustrative online regression task, a neuron must learn to match its output to a time-varying target (gray dashed line). With a classical delta rule (black line), weights adapt over time to eventually correct the output. With two timescales of plasticity (purple line), fast weights can pin the output to the target from the outset, while slow weights evolve in the background to learn the task.

could represent, for instance, the movement of a limb, but in this analysis we will take it to be an abstract, time-varying one-dimensional signal. We assume there are N synapses, and each one has access to two sources of information: the presynaptic spikes it receives, and an error signal.

The dynamics of the model neuron are described by equations for the total synaptic current, I , and firing rate, r ,

$$\tau_I \dot{I} = -I + \sum_i w_i x_i + \sigma_I \sqrt{2\tau_I} \xi_I \quad (7a)$$

$$\tau_r \dot{r} = -r + \alpha I + \sigma_r \sqrt{2\tau_r} \xi_r \quad (7b)$$

where a dot denotes a time derivative. In these equations, $\tau_I \sim 5$ ms and $\tau_r \sim 50$ ms denote the synaptic and rate-modulation time constants, w_i denotes the weight of synapse i , and each x_i is an independent Poisson spike train represented by a sum of delta functions,

$$x_i(t) = \sum_k \delta(t - t_{i,k}) \quad (8)$$

where $t_{i,k}$ is the time of the k^{th} spike arriving at synapse i . The parameter α sets the scale of firing rate response to synaptic current, and the final terms, ξ_I and ξ_r , denote zero-mean, unit-variance Gaussian white noise processes: $\langle \xi_i(t) \xi_j(t') \rangle = \delta(t - t')$, and similarly for ξ_r . Functionally, a barrage of input spikes is linearly filtered to drive the firing rate of the neuron.

The downstream output of the neuron, y , is driven by the firing rate of the neuron and white noise,

$$\tau_y \dot{y} = -y + r + \sigma_y \sqrt{2\tau_y} \xi_y. \quad (9)$$

The time constant, τ_y (taken to be ~ 100 ms), determines how quickly the output responds to changes in firing rate, and the final term is, as above, zero-mean, unit-variance Gaussian white noise.

We develop the theory using a teacher-student learning framework: the target, y^* , is generated by simulating [equations \(7\) and \(9\)](#), except with no noise and with a set of target weights, denoted w_i^* , in place of w_i . The target weights drift slowly around a mean value, μ_w , with time constant $\tau_w \sim 10^3$ s, modeling environmental variability or ongoing changes in the rest of the brain ([Aitchison et al. 2021](#)). We assume, for simplicity, that an instantaneous, continuous error signal is provided to all synapses, such as by diffuse transmission of a neuromodulator ([Magee and Grienberger 2020](#)); below we will relax this assumption, and incorporate delayed feedback communicated by spikes. The error signal is modeled as

$$f = y - y^* + \sigma_f \xi_f, \quad (10)$$

where ξ_f is zero-mean, unit-variance Gaussian white noise.

The full derivation of the plasticity rule is provided in *Methods: Derivation of the Bayesian plasticity rule*. Here we outline the general approach and main results. Similar to the analysis above, we decompose the synaptic weights into independent fast and slow components, $w_i = \bar{w}_i + \delta w_i$, as in [equation \(3\)](#). Framing synaptic plasticity as an optimal control problem, we find that by deriving fast-weight dynamics that minimize the error we simultaneously obtain a highly effective slow-weight learning rule.

To derive the fast-weight rule, we consider a typical control strategy: using past error signals (equation (10)), compute a control variable, $\hat{u}(t)$, such that applying it as an input to the system (in our case, at the level of synaptic currents) will minimize future output error. Analogous to the toy model above, rather than invoke an external control system for this purpose, we assume everything is implemented locally at individual synapses.

Suppose that each synapse can compute such an error-minimizing control, \hat{u} . Setting the fast weights uniformly as

$$\delta w_i = \frac{\hat{u}}{N\nu}, \quad (11)$$

with ν denoting the expected input rate per synapse, then leads to $\sum_i \delta w_i x_i \approx \hat{u}$.

We would like to choose \hat{u} to minimize the squared error, $(y - y^*)^2$. However, there is a cost to making the squared error very small: \hat{u} , and thus δw_i , will undergo large fluctuations. To navigate the tradeoff between minimizing the squared error while making sure the fast-weight fluctuations aren't so large, we find \hat{u} via

$$\hat{u} = \underset{u}{\operatorname{argmin}} \mathbb{E} \left[\frac{1}{T} \int_0^T (y - y^*)^2 + \lambda_u u^2 dt \right] \quad (12)$$

where T is the task duration and the expectation, $\mathbb{E} [\dots]$, is over the noise. The parameter λ_u is the 'cost of control'; making it larger reduces fluctuations in the fast weights but increases the squared error between y and y^* ; making it smaller has the opposite effect. The quadratic loss in equation (12), along with the Gaussian white noise in the dynamical equations, means that \hat{u} can be computed as a solution to the classic linear-quadratic-Gaussian (LQG) control problem. This approach results in a dynamical system that performs a Bayes-optimal estimate of unobserved states (such as the true underlying error, $y - y^*$), and uses that to construct an optimal linear controller (Crassidis and Junkins 2011).

The resulting equations predict how synapses should process input and feedback for optimal plasticity. For ease of intuition, we present the results here with some simplifying approximations. The results without approximations, given by equation (76), are used in simulations, although the approximate equations below provide comparable performance (Fig. S1).

To set the fast weights, u is computed by processing the error feedback signal, f (equation (10)), via

$$\tau_y \dot{\hat{\delta}} = -(1 + \alpha \mathcal{L}) \hat{\delta} + \mathcal{K}(f - \hat{\delta}) \quad (13a)$$

$$\hat{u} = -\mathcal{L} \hat{\delta} \quad (13b)$$

(Methods, equations (80a) and (92)). Equation (13a) filters the noisy feedback to compute an online estimate, $\hat{\delta}$, of the true error, $\delta = y - y^*$, using an optimal feedback gain, \mathcal{K} . The control variable is designed to negate the estimated error, with the magnitude of corrections determined by the optimal control gain, \mathcal{L} (equation (13b)). The gains \mathcal{K} and \mathcal{L} depend on the model parameters, reflecting both the integrative dynamics of the neuron and downstream output, and the statistics of the noise.

How do the fast weights evolve in this scenario? Taking a time derivative of equation (11), and substituting equations (13a) and (13b), leads to

$$\tau_y \dot{\delta w_i} = -(1 + \alpha \mathcal{L} + \mathcal{K}) \delta w_i - \frac{\mathcal{K} \mathcal{L}}{N\nu} f. \quad (14)$$

The fast weights are thus a low-pass filtered version of the negative of the error signal, which is the optimal strategy in the presence of noise. To see how this relates to the toy-model strategy above, we use the fact that when the control cost, λ_w , and feedback noise variance, σ_f^2 , are small, the optimal gains \mathcal{L} and \mathcal{H} are large (*Methods*, [equations \(87\)](#) and [\(91\)](#)). In this regime, the last term on the right-hand side of [equation \(14\)](#) dominates the dynamics; if we consider discrete updates with a time step Δt , [equation \(14\)](#) becomes

$$\Delta \delta w_i \approx -\frac{\Delta t \mathcal{K} \mathcal{L}}{\tau_y N \nu} (y - y^*) + \text{noise} \quad (15)$$

where we used [equation \(10\)](#) for f . Choosing the control cost to ensure that the product $\mathcal{H}\mathcal{L}$ obeys the relationship $\mathcal{K}\mathcal{L} = \frac{\tau_y}{\Delta t}$ and ignoring the noise, we recover the fast-weight updates for the toy model ([equation \(4\)](#)). Details aside, the key principle is the same: observe the current error, then greedily adjust weights to compensate.

What about the slow weights – should they even be updated? And if so, how? To answer the first question, we note that much of the error being controlled by the fast weights is due to mismatch between the slow weights, \bar{w}_i , and the target weights, w_i^* . Which makes sense: in the absence of noise, when $\bar{w}_i = w_i^*$ the weights do not need to be adjusted. That can be seen for the toy model in the right panel of [Fig. 1a](#); for the more realistic model it's shown explicitly in *Methods*, [equation \(48a\)](#). This tells us that the closer the slow weights are to their target values, the smaller the required control signal. Thus, given there is always a maximum control that can be applied, pushing the slow weights towards the target weights will improve performance.

A simple approach to updating the slow weights would be to use a modified delta rule, similar to [equation \(6\)](#). However, in this more realistic setting, the delta rule is suboptimal for two reasons. First, the error feedback signal, f , is noisy, which leads to noisy weight fluctuations, and even instability unless the learning rate is very small. Second, as pointed out by [Aitchison et al. \(2021\)](#), efficient weight updates should depend on uncertainty, with more uncertain weights updated more rapidly.

Qualitatively at least, the first problem can be fixed by filtering the feedback signal, f , and the second by scaling the learning rate by the uncertainty. Determining exactly how to design the filter and how much to scale, is, however, somewhat nontrivial. Fortunately, the analysis used to derive the update rule for the fast weights leads naturally to a rule for updating the slow weights that addresses these problems.

As shown in *Methods: Derivation of the Bayesian learning rule* (see in particular [equation \(80\)](#)), the update rule for the slow weights is

$$\dot{\bar{w}}_i = -\frac{z_i}{\sigma_f^2} (f - \hat{\delta}) - \frac{1}{\tau_w} (\bar{w}_i - \mu_w) \quad (16a)$$

$$\dot{\sigma_i^2} = -\frac{z_i^2}{\sigma_f^2} - \frac{2}{\tau_w} (\sigma_i^2 - \sigma_w^2) \quad (16b)$$

$$\tau_y \dot{z}_i = -(1 + \mathcal{K}) z_i + \alpha \sigma_i^2 x_i. \quad (16c)$$

The first term on the right-hand side of [equation \(16a\)](#) looks very similar to the delta rule. However, there are three notable differences. First, the term proportional to $(\bar{w}_i - \mu_w)$ leads to relaxation to the target-weight mean in the absence of feedback. This reflects the fact that, as mentioned above, the weights drift on a timescale of τ_w , so without feedback the mean, μ_w , becomes the optimal estimate of the weight. Second, the raw error feedback signal, f , is replaced by a prediction error, $(f - \hat{\delta})$. Because $\hat{\delta}$ is just a filtered version of f ([equation \(13a\)](#)), $(f - \hat{\delta})$ reflects deviations of the error signal about its time-averaged value. Third, and most important,

the effective learning rate is proportional to z_i , which plays the role of an eligibility trace, so is different for every synapse. Equation (16c) tells us that z_i increases with the variable σ_i^2 , which reflects weight uncertainty, so the higher the uncertainty the higher the learning rate – as expected. Determining the dependence of σ_i^2 on the firing rate is not completely straightforward, but we show in *Methods* (see in particular equation (99)) that it scales as $1/\sqrt{\nu_i}$ where, recall, ν_i is the rate of input to synapse i . This too is expected, at least qualitatively: the higher the firing rate, the more certain a synapse is about its value, and the lower the learning rate.

Mathematically, equations (16a) and (16b) represent the evolving mean and variance of a posterior distribution over target weights, similar to the approach introduced by Aitchison et al. (2021). This is the best estimate that can be formed by the synapse, given the observed data, and therefore the most efficient learning rule for \bar{w}_i . To highlight this qualitative difference from classical counterparts, and make connections to previous work, we will refer to our plasticity rule and its generalizations as ‘Bayesian plasticity’, in line with Aitchison et al. (2021).

Using the teacher-student task, we compared our plasticity rule to a classical gradient-based rule for a neuron with 1000 synapses. The results are shown in Fig. 2. The classical approach is a generalized delta rule based on the Real-Time Recurrent Learning algorithm (Williams and Zipser 1989; Pearlmutter 1995), which is the canonical approach for online learning in dynamic models. We included simulations of the full Bayesian rule, as well as versions using the slow and fast components alone. Learning with the Bayesian slow-weight rule alone is already superior to the classical rule, both in terms of total output error (Fig. 2a) and the speed of convergence of the weights towards their target values (Fig. 2b). Output error is reduced further still when the fast weights are included, due to suppression of noise and compensation for weight mismatch at early stages of learning. Moreover, weight convergence in this case is almost indistinguishable from when slow-weight learning is implemented in isolation. Although a similarly low level of error is obtained using the fast weights alone, this requires much larger control currents (Fig. 2c), and no memory of the task is actually retained in the weights. With some small modifications (*Methods: Feedback delays*), our rule also accounts for time lags in the feedback signal, representing communication delays in the brain. In this case, slow-weight learning is virtually unaffected and fast-weight control remains effective for delays on the order of the downstream time constant (Fig. 2d).

Control and learning with spiking feedback

So far we have considered a continuous feedback signal, representing diffuse transmission of a neuromodulator. But many important feedback pathways in the brain communicate via spikes; we now generalize the theory to incorporate these signals. Our modeling choices are specifically motivated by the sparse climbing-fiber inputs to cerebellar Purkinje cells, which provide instructive signals that drive plasticity at parallel-fiber synapses (Hull and Regehr 2022; Silva et al. 2024). However, the general formulation is applicable to other scenarios, such as dendritic spikes in cortical neurons that are triggered by feedback connections (Larkum 2013; Richards and Lillicrap 2019; Fişek et al. 2023).

We model the spiking feedback signal, f_S , as an inhomogeneous Poisson process with rate

$$\gamma(t) = \gamma_0 e^{\rho(y(t) - y^*(t))}. \quad (17)$$

Concretely, f_S comprises a series of delta functions centered at spike times that are generated with instantaneous rate $\gamma(t)$. The synapses ‘see’ the spikes, and can use those to update synaptic strength. The parameters γ_0 and ρ set the spontaneous rate of feedback and steepness of the nonlinear response to output error (representing, for instance, the transfer function of an error-coding neuron).

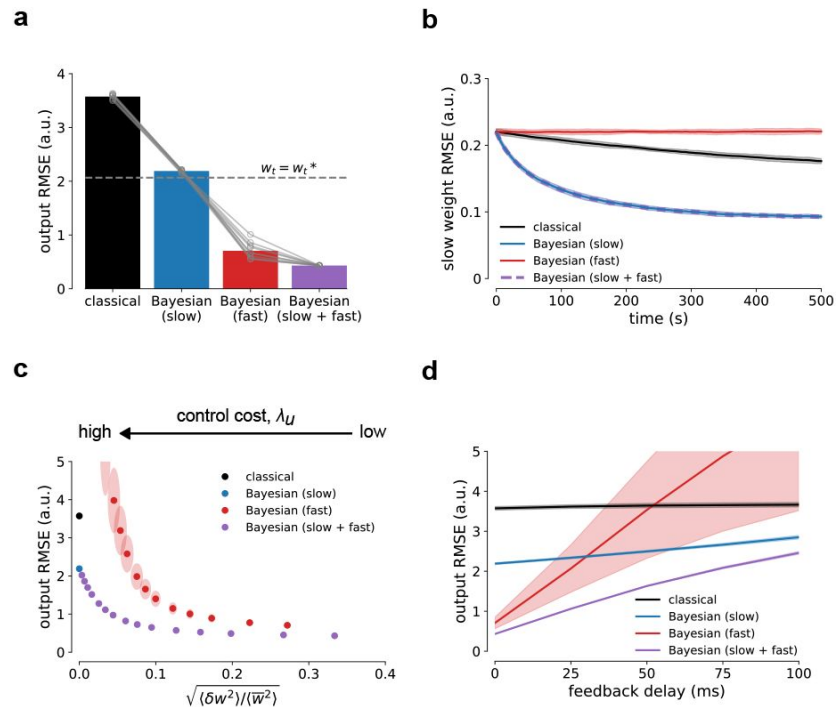


Figure 2

Synaptic plasticity as optimal control.

a) Performance comparison between the classical gradient-based rule (black) and Bayesian rule without fast weights (blue), with plastic fast weights and frozen slow weights (red), and with plastic slow and fast weights (purple). Bars denote root-mean-squared error (RMSE) between output, y , and target output, y^* , averaged over the entire task. Data points denote different random seeds. The dashed line gives the average error when slow weights are set to their target values at every point in time. The learning rate for the classical rule and control cost for the Bayesian rule were selected via grid search to minimize output error. **b)** The Bayesian rules yield faster convergence of weights to their target values compared to the classical rule, quantified as RMSE between weight vectors. Shaded areas are standard deviation from 10 random seeds. **c)** Root-mean-squared output error versus fast weight fluctuations, the latter computed as the root-mean-squared fast weight divided by root-mean-squared slow weight (averages taken over entire task). The size of the fast weight fluctuations were controlled by the cost parameter λ_U (see equation (12)). In the full plasticity rule (purple), even small fluctuations of $\sim 10\%$ lead to a large reduction in error. Without slow-weight learning (red), much more control is needed for a similar level of performance. Points denote the average over seeds for a fixed λ_U ; the shaded ellipses denote standard deviation across seeds. **d)** Control is effective for feedback delays shorter than the output time constant (here, $\tau_y = 100$ ms). Shaded areas are standard deviation from 10 random seeds.

With spiking feedback, the standard formalism cannot be used for estimating the error and target weights. Instead, similar to [Eden et al. \(2004\)](#) and [Pfister et al. \(2010\)](#), we derived a recursive Bayesian algorithm. The details are provided in *Methods: Spiking feedback*, with the full plasticity rule described by [equation \(117\)](#). This has a similar structure to the rule discussed above ([equations \(13\)](#) and [\(16\)](#)). Incoming feedback spikes are optimally filtered to form an estimate, $\hat{\delta}$, of the continuous-valued output error (**Fig. 3a**). This is used to set the fast weights via an optimal control gain as before. For the slow weights, $\hat{\delta}$ is transformed into an estimate of the instantaneous feedback rate, $\hat{\gamma}$, and combined with an eligibility trace, z_i , to drive learning,

$$\dot{\bar{w}}_i = -\rho z_i(f_S - \hat{\gamma}) - \frac{1}{\tau_w}(\bar{w}_i - \mu_w). \quad (18)$$

The eligibility trace, which evolves similarly to [equation \(16c\)](#), records the recent activity level of the synapse. Qualitatively, the slow weights of recently active synapses tend to decrease in steps when feedback spikes arrive and increase gradually in between. This process eventually reaches a steady state where the observed feedback rate is equal to the predicted rate, which happens when the target weights have been accurately estimated.

We compare the Bayesian rules to a spiking-feedback version of the classical rule used in the continuous model above (*Methods, equation (38)*). Notably, the classical spike-based rule conforms to the standard model of Purkinje cell plasticity ([Ito 2001](#); [Coesmans et al. 2004](#)): synaptic input alone leads to long-term potentiation (LTP), whereas the conjunction of synaptic input and feedback spikes are required for long-term depression (LTD).

Simulations with the spiking-feedback model were consistent with those of the continuous model, recapitulating all of the previously observed advantages of Bayesian plasticity (**Figs. 3b** and **S2**). Learning in this case is substantially harder, requiring an order of magnitude more time to reach equivalent levels of performance, primarily because spikes occur rarely. The spontaneous feedback rate, γ_0 , is therefore a key determinant of performance, with higher rates leading to faster learning and lower output error (**Fig. S2**).

Not only does a low feedback rate make learning hard, it also makes fast-weight updates less effective. This is not surprising, since online corrections can only be reliably made soon after receiving an error-encoding spike. However, we found performance can be boosted by distributing the task and feedback across multiple neurons. We explored this by simulating $M = 20$ neurons, each as described by [equation \(7\)](#). For ease of comparison with the single-neuron simulations, we placed 50 synapses on each of the M neurons to give the same total number of synapses. The firing rates in the population model, r_m , are summed to drive a common downstream output, modifying [equation \(9\)](#),

$$\tau_y \dot{y} = -y + \sum_{m=1}^M r_m + \sigma_y \sqrt{2\tau_y} \xi_y. \quad (19)$$

Each neuron receives a separate Poisson feedback signal, with spikes generated independently from the common error-dependent rate $\gamma(t)$. Because the collected feedback spikes now provide a denser error signal, effective control can be achieved at the population level even when γ_0 is small (**Fig. 3b**, dashed line).

Fast and slow synaptic plasticity in the cerebellum

Our theory was derived in a setting where the target weights are defined explicitly and the target output is guaranteed to be realizable by the model. To validate our results in a biological setting and, crucially, to make experimentally testable predictions, we apply it to a cerebellar learning problem.

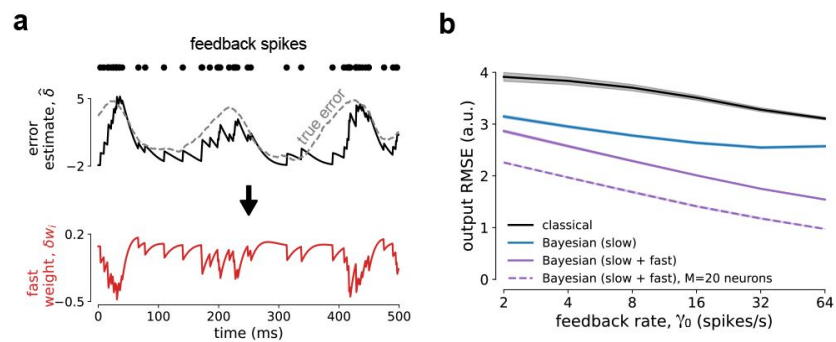


Figure 3

Control and learning with spiking feedback.

a) When feedback is communicated via noisy spiking activity, optimal plasticity requires the synapse to infer the true underlying continuous error signal. The estimated error is then used to drive both control and learning. Fast weights seek to improve performance by canceling the estimated error (lower panel), while slow weights seek to reduce the error permanently. **b)** Performance depends on the rate of feedback. High feedback spike rates increase the precision of error estimates, thereby enhancing learning and control. A population of neurons acting on multiple independent feedback signals (dashed purple trace) can compensate for very low individual feedback rates. Shaded areas are standard deviation from 10 random seeds.

Our population model can be mapped onto the structure of a cerebellar microzone – a group of Purkinje cells receiving correlated climbing-fiber input and whose outputs converge on common down-stream targets (**Fig. 4a**) (Apps et al. 2018). As an abstraction of microzone processing, we consider a population of 20 neurons that must learn to transform patterns of synaptic input into time-varying outputs (**Fig. 4b**). The inputs come from parallel fiber spikes, assumed to be driven by time-varying patterns of granule cell activity (Medina et al. 2000; Gilmer et al. 2023). The output represents a downstream motor signal, or a predicted motor output (Wolpert et al. 1998). Feedback is provided by sparse climbing-fiber spikes, with $\gamma_0 = 2$ spikes/s, leading to firing rates close to that observed *in vivo* (Armstrong and Rawson 1979).

We defined 10 pairs of input-output patterns $\{\mathbf{v}_p(t), y_p^*(t); p = 1, \dots, 10\}$, each lasting 1 s (**Fig. 4b**). The input patterns comprised time-varying vectors of synaptic input rates, \mathbf{v}_p , realized with Poisson spiking. We used 2000 synapses in total, distributed equally across the 20 Purkinje cells. The target output trajectories were defined as random wavelets with shapes that drift on a slow timescale of $\sim 10^4$ s (analogous to the target-weight drift in the teacher-student task). Input-output pairs were presented in a continuous stream in random order for 500 s. While all of the models can eventually solve this task, the Bayesian models learn much faster, and control via the fast weights can pin the output to the target from the outset (**Fig. 4c,d**).

So far we have focused on learning a particular input-output mapping. What happens when the desired input-output mapping changes? To address this, we tested the response of the trained models to changes in the target outputs. As above, we trained on 10 input-output mappings, but after training we changed 5 of the outputs (**Fig. 4e**). An efficient response requires synapses to adapt selectively to learn the new input-output pairs, without disrupting previously learned pairs. We find that Bayesian plasticity is much more robust than the classical approach, enabling rapid recovery over a few tens of pattern presentations. With fast-weight updates, the output error is almost unaffected by the perturbation. And in the background, the slow weights gradually converge to their new target values, reducing the magnitude of ongoing control currents and updating the stored memory of the task.

Signatures of fast and slow synaptic plasticity

We use the cerebellar learning task to make experimental predictions for the control and learning components of our theory. In the cerebellar model, the control signal depends strongly on climbing-fiber input, transiently adjusting activity whenever a feedback spike arrives. To measure the size of this effect in our model, we plotted the firing rates of neurons relative to the time of their associated climbing-fiber spikes. These simulations show a pronounced dip in firing rates after a climbing-fiber spike, but only when there are fast-weight updates; the dip disappears both for classical and Bayesian updates of slow weights only (**Fig. 5a**). Such patterns of activity – climbing-fiber-induced ‘spike pauses’ – have been widely observed *in vivo* (Bell and Grimm 1969; Bloedel and Roberts 1971; Sato et al. 1992; Barmack and Yakhnitsa 2003; Han et al. 2020). However, while widely observed, their computational significance has remained unclear; our model provides a novel, normative explanation that spike pauses are a strategy for suppressing downstream error. Moreover, it makes two experimentally testable predictions: the duration of the dip increases with the time constant of downstream dynamics, τ_y , and the magnitude of the dip decreases with feedback delays (**Fig. 5b**). Both predictions could be tested by manipulating output dynamics experimentally, or exploiting the fact that feedback delays vary across different regions of the cerebellum (Suvrathan et al. 2016; Jayabal et al. 2022).

Our model also makes predictions about the slow-weight dynamics. For that we used the trained models to simulate a common experimental plasticity protocol. In this protocol, synapses are repeatedly stimulated with a short burst of parallel fiber spikes, followed by one or more climbing-fiber spikes (here repeated 50 times at 2 repetitions/s) (**Fig. 5c**). In experiments, LTP is generally observed at zero or negative intervals between parallel and climbing-fiber stimulation, but

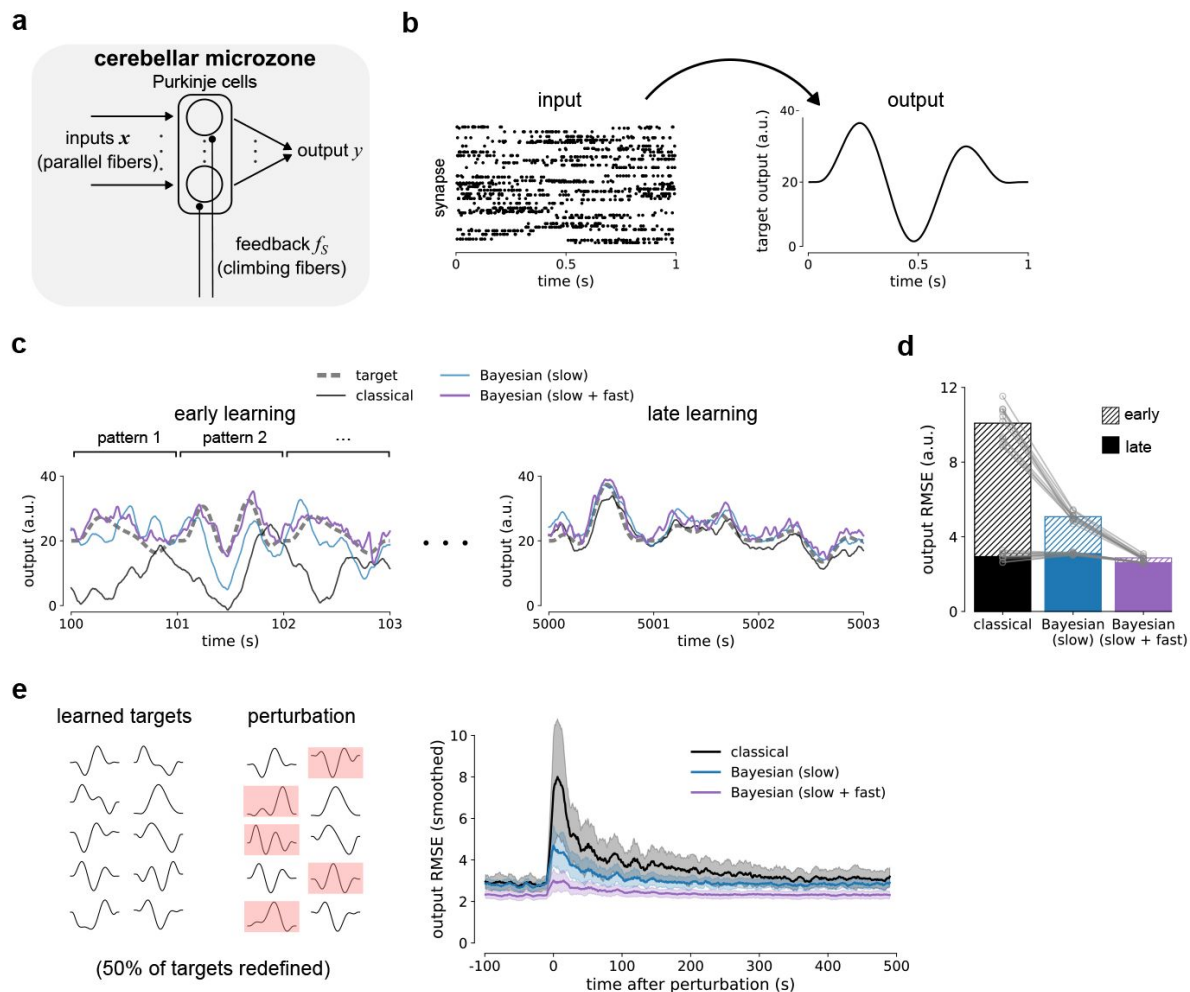


Figure 4

Fast and slow plasticity in the cerebellum.

a) Schematic of the cerebellar microzone model. A group of Purkinje cells receiving synaptic input from parallel fibers project to a common downstream output. Each Purkinje cell receives feedback spikes from a separate climbing fiber, signaling the error between actual and target outputs. **b)** Schematic of the learning task. Ten patterns of temporally organized parallel-fiber input must be mapped by a population of Purkinje cells to time-varying target outputs. **c)** Example outputs early and late in learning, replicating the results from the toy model in [Fig. 1b](#). **d)** Performance during early (100 – 200 s) and late (5000 – 5100 s) stages of learning. Bars denote RMSE between output and target output, averaged over 100 s. **e)** Left: After training on 10 input-output mappings, half of the target outputs were changed. Right: Bayesian plasticity confers faster recovery than the classical approach. With fast weights, the output is largely insensitive to the perturbation. Error curves have been smoothed with a moving average filter of width 10 s. Shaded areas are standard deviations from simulations pooled over 10 random training seeds \times 10 perturbations.

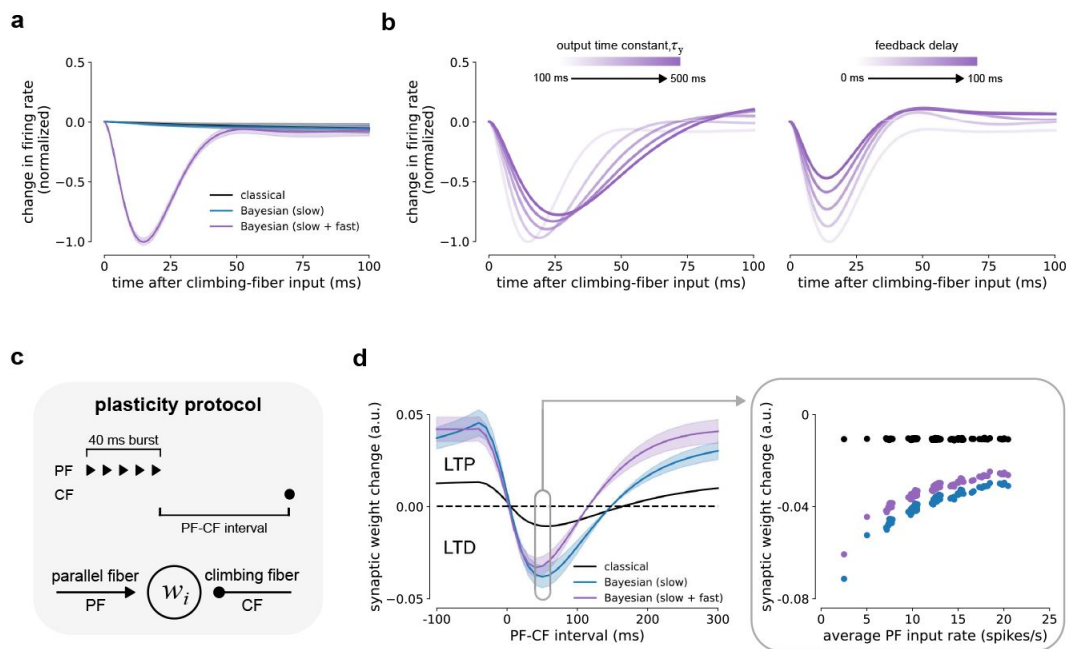


Figure 5

Signatures of fast and slow synaptic plasticity.

a) Suppression of Purkinje-cell firing rates after climbing-fiber input is a signature of fast-weight updates. The solid lines denote averages of firing rates aligned to climbing-fiber feedback spikes during the cerebellar learning task. Shaded areas are the standard deviation of the average firing rate change of 20 neurons \times 10 random seeds. **b)** The duration and magnitude of the predicted firing rate suppression vary systematically with the time constant of the downstream output and the feedback delay. **c)** A standard experimental protocol can discriminate between classical and Bayesian plasticity. After training on the cerebellar learning task, synapses were stimulated with conjunctions of parallel-fiber input bursts and climbing-fiber feedback spikes (50 repetitions at 2 reps/s). **d)** Left: The Bayesian rules produce LTD over a narrower range of PF-CF intervals than the classical rule, and also exhibit greater variability across synapses. Right: The variability is due to the adaptive learning rate: in the Bayesian rules, unlike the classical rule, the magnitude of weight changes depend on the average rate of input during the task. Solid lines denote synaptic weight changes measured at the end of the protocol, averaged over 100 tested synapses. Shaded areas are standard deviation.

switches to LTD for sufficiently positive intervals or sufficiently many climbing-fiber spikes (Wang et al. 2000 [↗](#); Safo and Regehr 2008 [↗](#); Mathy et al. 2009 [↗](#); Suvrathan et al. 2016 [↗](#); Bouvier et al. 2018). All of our models reproduced these observations, and have qualitatively similar behavior as the parameters of the plasticity protocol are varied (Fig. S3 [↗](#)). However, the Bayesian rules yielded sharper tuning to the interval, and, importantly, greater variability across synapses (Fig. 5d [↗](#)). The source of this variability is the adaptive learning rate of Bayesian plasticity: the magnitude of slow-weight changes during the protocol depends on the average rate of input received during the task. Our theory can therefore be discriminated from the standard model by combining well-established plasticity protocols with recording or manipulation of input rates. While this may be possible *in vivo* in the near future with optogenetics and voltage imaging (Fan et al. 2023), *in vitro*, we predict that preconditioning a synapse with a long train of parallel fiber spikes can block subsequent plasticity induction by driving down its intrinsic learning rate.

Discussion

We proposed that synaptic plasticity should operate at two timescales: a fast timescale to suppress immediate errors, and a slow timescale to learn. This multiscale plasticity rule significantly outperforms online gradient-based learning, confers robustness to noise and other perturbations, and means that neurons can accurately perform a task almost immediately – well before learning has stored a longer-term solution in the weights.

Fast weights for feedback control

Our theory frames synaptic plasticity as an optimal control problem. This is a natural framework for any system where real-time feedback is available to guide the dynamics. While there is a long tradition of using control theory to understand neural function (Todorov 2004 [↗](#); McNamee and Wolpert 2019 [↗](#)), our work is most closely related to recent studies that have also connected fast control mechanisms with slower processes of learning. In one approach, fast feedback was used to drive spiking network dynamics to a regime that enables local learning of complex tasks (Bourdoukan and Denève 2015 [↗](#); Denève et al. 2017 [↗](#); Alemi et al. 2018 [↗](#)). In another, the Deep Feedback Control theory uses feedback to drive network output to a target steady-state, and then exploits credit assignment signals implicit within the controller to gradually tune feedforward weights (Meulemans et al. 2021 [↗](#); Meulemans et al. 2022 [↗](#); Rossbroich and Zenke 2023 [↗](#)). In a recurrent network model of motor adaptation, Feulner et al. (2022) [↗](#) notably used the same error feedback signal for both control and learning, finding network dynamics resembling data from monkeys. Our theory differs markedly from all of these approaches, however, as the control signals in our models can be computed and implemented by individual synapses. Thus, in place of complex network-level control structures, we are positing a novel processing role for intracellular signaling at the synapse (Bhalla 2014 [↗](#); Benna and Fusi 2016 [↗](#); Zenke et al. 2017 [↗](#)). Exploiting this local layer of processing could allow the brain to operate much more efficiently: physically, it reduces the amount of wiring and neurons needed for high performance; algorithmically, neurons make dual use of the feedback signals they receive.

The fast mechanism we propose differs from typical forms of short-term plasticity (Zucker and Regehr 2002 [↗](#); Abbott and Regehr 2004 [↗](#)). Namely, in our model, fast weight changes depend strongly on error feedback, rather than presynaptic spike patterns alone. How could this be implemented biologically? If error signals are carried by a neuromodulator, as assumed in our continuous-feedback model, there are a range of candidate pathways. Neuromodulators including acetylcholine (McGehee et al. 1995 [↗](#); Gil et al. 1997 [↗](#)), serotonin (Feng et al. 2001 [↗](#)), dopamine (Higley and Sabatini 2010 [↗](#)), norepinephrine (Cheun and Yeh 1992 [↗](#)), among others (Marder 2012 [↗](#); Nadim and Bucher 2014 [↗](#)), are all capable of selectively regulating synaptic transmission and the magnitudes of postsynaptic currents. In common with our model, these effects are transient, can be bidirectional, and are distinct from influences on membrane excitability and

long-term plasticity. Alternatively, feedback signals might be processed by closely apposed glia. For instance, astrocytes would be well-placed to perform the computations we describe: they are known to integrate a variety of molecular signals and dynamically fine-tune both pre and postsynaptic properties (Perea et al. 2009 [↗](#); De Pittà et al. 2016 [↗](#); Papouin et al. 2017 [↗](#)).

In our cerebellar model, we assume that the fast fluctuations are driven by climbing fiber input. At a functional level, decades of past experiments are in striking agreement with our theory: when a climbing fiber spike is received by a Purkinje cell, there is a brief pause in output firing. Previous functional explanations for spike pauses include precise temporal or multiplexed coding (Steuber et al. 2007 [↗](#); De Schutter and Steuber 2009 [↗](#); Han et al. 2020 [↗](#)), encoding dendritic spike rates (Davie et al. 2008 [↗](#)), and transmitting teaching signals to downstream neurons (Mathews et al. 2012 [↗](#)). While not mutually exclusive, we offer the simple explanation that if climbing fibers are indeed signaling error, then pauses are an effective mechanism for making short-term corrections. Our modeling predicts this mechanism would be most apparent in experiments focused on regions and modalities with short feedback delays, such as those receiving signals from the spinal cord for postural maintenance, or proprioceptive feedback from the periphery. It will be interesting to investigate how feedback control of cerebellar output could also support the function of more complex, nested motor-control loops (Wolpert et al. 1998 [↗](#); Rotondo et al. 2023 [↗](#)), and cognitive computations more generally (Hull 2020 [↗](#); Kostadinov and Häusser 2022 [↗](#); Pemberton et al. 2023 [↗](#)).

Slow weights for learning

Optimizing weights for control on fast timescales also leads to highly efficient learning on slow timescales. The rule that arises naturally in our framework works efficiently by maintaining a Bayesian estimate of the target synaptic weight and its uncertainty – the best a synapse can do, given a sequence of noisy local observations. Updating weights slowly in line with this evolving estimate gradually reduces the magnitude of corrections that need to be applied via the fast weights.

Bayesian approaches to plasticity were explored in much earlier work, both in machine learning (Buntine and Weigend 1991; MacKay 1992 [↗](#)) and neuroscience (Dayan and Kakade 2000 [↗](#)), but have only recently begun to be broadly exploited (Blundell et al. 2015 [↗](#); Hernández-Lobato and Adams 2015 [↗](#); Kappel et al. 2015 [↗](#); Kirkpatrick et al. 2017 [↗](#); Drugowitsch et al. 2019 [↗](#); Hiratani and Latham 2020 [↗](#); Aitchison et al. 2021 [↗](#); Jegminat et al. 2022 [↗](#); Malkin et al. 2024 [↗](#)). Aitchison et al. (2021) [↗](#), in particular, were the first to develop a Bayesian theory of local synaptic plasticity. In common with our slow-weight rule, they found that tracking weight uncertainty leads to an adaptive learning rate, yielding superior performance to a simple online delta rule. As well as the novel connection to control, the slow-weight component of our theory generalizes their results in two important directions. First, Aitchison et al. (2021) [↗](#) assume that data seen by a synapse comprises independent samples, approximating away any history dependence. By contrast, our dynamic model and learning rule accounts for, and crucially exploits, the temporal correlations inherent in neural signals. Second, our theory incorporates sources of spiking feedback and communication delays, making Bayesian plasticity more widely applicable as a theory of learning in the brain. Simulating our more realistic model, we outlined how this theory could be tested (Fig. 5 [↗](#)): if synapses have evolved to use local data as efficiently as possible, then adding a pre-conditioning train of input to common *in vitro* plasticity protocols should unmask the characteristic adaptive learning rate.

Outlook

Our general framework can be used to make predictions about plasticity in other cells and circuits, including more complex scenarios than we have studied here. For instance, while we have focused on linear neural dynamics – a suitable approximation for cerebellar Purkinje cells (Llinás and Sugimori 1980 [↗](#); Brunel et al. 2004 [↗](#); Walter and Khodakhah 2006 [↗](#)) – the theory could be

adapted to account for nonlinear integration in cortical and hippocampal neurons (Poirazi et al. 2003 [↗](#); Payeur et al. 2021 [↗](#); Bicknell and Häusser 2021 [↗](#)). This could be achieved with a straightforward modification of the Kalman filtering approach that we used, for which there are already established theoretical tools for nonlinear filtering and control (Crassidis and Junkins 2011 [↗](#); Kutschireiter et al. 2020 [↗](#)). At the network level, combining fast and slow plasticity with recurrent connectivity would be a challenging, but likely very fruitful, direction. It has previously been shown, for instance, that the concerted action of multiple plasticity rules can enhance memory formation and stability (Zenke et al. 2015 [↗](#)).

Finally, to derive our fast and slow plasticity rules, the synapses had to know the underlying equations that transform synaptic drive to the output. How this is learned is an interesting, and important, problem. It is likely that some of the learning can be accomplished on much slower, even evolutionary, timescales (Friedrich et al. 2021 [↗](#)). On shorter timescales, Moore et al. (2024) [↗](#) proposed an elegant solution: learn the optimal controller from data without ever learning the underlying equations. Applying this formalism, they demonstrated that a wide range of neurophysiological data can be explained by modeling individual neurons as optimal feedback controllers. Incorporating similar ideas to study nonlinear computations in hierarchical and recurrent neural circuits is an important avenue for future work.

Methods

Online linear regression

Using the simple linear regression model described by equation (1) [↗](#), we illustrated how concurrent fast and slow updates to synaptic weights can be exploited to optimize task performance. Here we provide the derivation of the plasticity rules for that model (equations (4) [↗](#) and (6) [↗](#)).

We consider a discrete-time setting where the inputs, output and weights are updated at small intervals, Δt . Decomposing synaptic weights into fast and slow components, equation (1) [↗](#) becomes

$$y(t) = \sum_i (\bar{w}_i(t) + \delta w_i(t)) \nu_i(t). \quad (20)$$

After observing the local inputs, ν_i , and error, $y - y^*$, at time t , synapses update their fast and slow weight components in parallel,

$$\bar{w}_i(t + \Delta t) = \bar{w}_i(t) + \Delta \bar{w}_i(t) \quad (21a)$$

$$\delta w_i(t + \Delta t) = \delta w_i(t) + \Delta \delta w_i(t). \quad (21b)$$

The role of the fast weights, δw_i , is to greedily suppress the current error, so that y approximately matches y^* – independent of the setting of the slow weights. We achieve this by choosing the fast-weight updates to be proportional to the negative of the error,

$$\Delta \delta w_i(t) = -\frac{y(t) - y^*(t)}{N\nu}. \quad (22)$$

where N is the total number of synapses and $\nu \equiv \frac{1}{N} \mathbb{E}(\sum_i \nu_i)$ is the expected input rate. Using this expression, and inserting equation (21) into equation (20), at the next time step we have

$$\begin{aligned} y(t + \Delta t) &= \sum_i (\bar{w}_i(t) + \Delta \bar{w}_i(t) + \delta w_i(t) + \Delta \delta w_i(t)) \nu_i(t + \Delta t) \\ &\approx \sum_i (\bar{w}_i(t) + \delta w_i(t)) \nu_i(t) - (y(t) - y^*(t)) \frac{\sum_i \nu_i(t)}{N\nu}. \end{aligned} \quad (23)$$

To derive the approximate expression in the second line, we replaced $\nu_i(t + \Delta t)$ by $\nu_i(t)$, which is valid if Δt is small compared to the timescale over which the firing rates change, and dropped the term $\Delta \bar{w}_i$, which is valid if the slow weight updates are small. The first term is just $y(t)$ (see equation (20)); approximating $\sum_i \nu_i$ by its expectation then gives

$$y(t + \Delta t) \approx y^*(t + \Delta t). \quad (24)$$

Thus, with mild assumptions, the fast-weight updates given in equation (22) force the output to closely match the target at all times.

While the fast-weight updates ensure low output error, this comes at the cost of ongoing, and potentially large, fluctuations in weights. Those fluctuations can be seen in the right panel of **Fig. 1a**, especially at early times before the slow weights have been updated. We address this with a local rule for the slow weights that permanently pushes y closer to y^* , thereby reducing the need for ongoing corrections. To that end, at each time step, we adjust the slow weights with the aim of minimizing the loss

$$L = \frac{1}{2} \left(\sum_j \bar{w}_j(t) \nu_j(t) - y^*(t) \right)^2. \quad (25)$$

The loss is minimized by making updates in steps proportional to the negative of the gradient

$$\begin{aligned} \frac{\partial L}{\partial \bar{w}_i} &= \left(\sum_j \bar{w}_j(t) \nu_j(t) - y^*(t) \right) \nu_i(t) \\ &= \left(y(t) - y^*(t) - \sum_j \delta w_j(t) \nu_j(t) \right) \nu_i(t) \\ &\approx \left(y(t) - y^*(t) - N\nu \delta w_i(t) \right) \nu_i(t) \end{aligned} \quad (26)$$

For the last line we used the fact that the fast weights do not depend on the synapse index, so $\sum_j \delta w_j \nu_j = \delta w_i \sum_j \nu_j \approx N\nu \delta w_i$. This leads to a local slow-weight update rule

$$\Delta \bar{w}_i(t) = -\bar{\eta} \left(y(t) - y^*(t) - N\nu \delta w_i(t) \right) \nu_i(t), \quad (27)$$

where $\bar{\eta}$ is a small learning rate. **Equations (22)** and **(27)** for the fast and slow updates correspond to **equations (4)** and **(6)** in the main text.

Classical online learning

For the model described by **equations (7)** and **(9)**, with either continuous feedback (**equation (10)**) or spiking feedback (**equation (17)**), we compared our Bayesian plasticity rules to classical, gradient-based counterparts. The classical rules are based on the Real Time Recurrent Learning algorithm (RTRL; Williams and Zipser 1989; Pearlmutter 1995), which is the canonical online, gradient-based learning rule for dynamic models. While typically employed in recurrent neural networks, here we apply it to a single-neuron model, adapted slightly to enable fair comparisons with alternatives.

For continuous feedback, we aim to minimize the squared output error, $(y - y^*)^2$. However, unlike the simple model, here we need to account for history dependence – the output at any given time depends on inputs that were received in the past. To account for this, we start by considering a loss integrated over a time horizon, h , in which the weights are held fixed,

$$\mathcal{E} = \frac{1}{2} \int_{t_0}^{t_0+h} (y - y^*)^2 dt. \quad (28)$$

The derivative of the loss with respect to the weight of synapse i over that period is

$$\frac{\partial \mathcal{E}}{\partial w_i} = \int_{t_0}^{t_0+h} (y - y^*) \frac{\partial y}{\partial w_i} dt. \quad (29)$$

The second factor in the integral is a dynamic quantity, which can be computed by taking derivatives of equations (7) and (9) with respect to w_i . This gives the linear system

$$\tau_I \dot{s}_{I,i} = -s_{I,i} + x_i \quad (30a)$$

$$\tau_r \dot{s}_{r,i} = -s_{r,i} + \alpha s_{I,i} \quad (30b)$$

$$\tau_y \dot{s}_{y,i} = -s_{y,i} + s_{r,i} \quad (30c)$$

for the sensitivities $s_{I,i} \equiv \frac{\partial I}{\partial w_i}$, $s_{r,i} \equiv \frac{\partial r}{\partial w_i}$, and $s_{y,i} \equiv \frac{\partial y}{\partial w_i}$. Equation (30) is solved from the initial condition $s_{I,i}(t_0)$, $s_{r,i}(t_0)$, $s_{y,i}(t_0) = 0$. An exact gradient-descent weight update could then be applied at time $t_0 + h$ as

$$w_i(t_0 + h) = w_i(t_0) - \eta \int_{t_0}^{t_0+h} (y - y^*) s_{y,i} dt, \quad (31)$$

where η is the learning rate. Functionally, $s_{y,i}$ plays the role of an eligibility trace by recording the recent impact of the synapse on the output.

Formally, after the weights have been updated, the sensitivities should then be reset to the zero initial condition to track activity over the next time interval. However, assuming weight changes are slow compared to the model dynamics, the RTRL rule is commonly run as a fully online approximation by taking $h \rightarrow 0$, giving a differential equation for the weight dynamics, and neglecting the reset of the sensitivities (Pearlmutter 1995). We take this approach here and make two further modifications. First, we replace the term $(y - y^*)$ with its noisy observation, f . Second, we append an additional weight decay term to account for the slow drift of the target weights (see Aitchison et al. (2021) and the derivation of the Bayesian rule below for motivation). This leads to classical online updates for each synapse

$$\dot{w}_i^{\text{clas}} = -\eta f s_{y,i} - \frac{1}{\tau_w} (w_i^{\text{clas}} - \mu_w). \quad (32)$$

For spiking feedback, we use the same approach, but with a different loss function. In this case the output error is communicated via a Poisson process with rate $\gamma(t)$, as described by equation (17). To handle the random and discrete nature of spiking feedback, we work with spike probabilities.

Over a time horizon h , the feedback spike count, denoted n , will be distributed as

$$P(n|\gamma) = \left(\int_{t_0}^{t_0+h} \gamma dt \right)^n \frac{e^{-\int_{t_0}^{t_0+h} \gamma dt}}{n!}. \quad (33)$$

Updates to the weights should seek to match this distribution to the one that would arise if all of the weights were set correctly. We express this in terms of a ‘target rate’, γ^* , which defines a target distribution, $P(n|\gamma^*)$, via equation (33) [\[33\]](#).

We use the cross-entropy loss between target and actual distributions to derive the updates; this is given by

$$\mathcal{H} = - \sum_{n=0}^{\infty} P(n|\gamma^*) \ln P(n|\gamma). \quad (34)$$

The derivative of the loss with respect to weight w_i is, using $\frac{\partial \gamma}{\partial y} = \rho \gamma$,

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial w_i} &= - \sum_{n=0}^{\infty} \frac{P(n|\gamma^*)}{P(n|\gamma)} \frac{\partial P(n|\gamma)}{\partial w_i} \\ &= - \sum_{n=0}^{\infty} P(n|\gamma^*) \left[n \left(\int_{t_0}^{t_0+h} \gamma dt \right)^{-1} - 1 \right] \int_{t_0}^{t_0+h} \rho \gamma \frac{\partial y}{\partial w_i} dt \end{aligned} \quad (35)$$

As before, we use equation (30) [\[30\]](#) to compute the $\frac{\partial y}{\partial w_i}$ term, accounting for the history dependence.

Similar to the continuous feedback model, we define weight updates to be proportional to the negative of the loss gradient, and then take $h \rightarrow 0$ to yield a continuous-time expression. Expanding equation (35) [\[35\]](#) to first order in h , approximating the integrals as $\int_{t_0}^{t_0+h} \gamma dt \approx h \gamma(t_0)$, and exponentials as $e^{-h\gamma} \approx 1 - h\gamma$, only the $n = 0$ and $n = 1$ terms survive from the sum. Some straightforward algebra then leads to

$$\frac{\partial \mathcal{H}}{\partial w_i} \approx h(\gamma - \gamma^*) \rho s_{y,i}. \quad (36)$$

Thus, the spiking-feedback analogue of equation (31) [\[31\]](#) is

$$w_i(t_0 + h) = w_i(t_0) - \eta_S h (\gamma - \gamma^*) \rho s_{y,i}, \quad (37)$$

where η_S is the learning rate.

Taking $h \rightarrow 0$, and approximating the unobserved rate γ with feedback spikes, f_S , leads to a continuous-time learning rule

$$\dot{w}_i^{\text{spike}} = -\eta_S \rho (f_S - \gamma^*) s_{y,i} - \frac{1}{\tau_w} (w_i^{\text{spike}} - \mu_w) \quad (38)$$

where we have appended a weight decay term as before.

Finally, to set γ^* , we use the fact that when the weights are set correctly, the error comprises Gaussian fluctuations about zero due to noise. With the exponential nonlinearity in the feedback rate function (equation (17) [\[17\]](#)), the expected rate is the mean of a log-normal variable, giving

$$\gamma^* = \gamma_0 e^{\frac{1}{2} \rho^2 \sigma_{\text{noise},y}^2}, \quad (39)$$

where $\sigma_{\text{noise},y}^2$ is the stationary variance of the output noise. The stationary variance can be computed numerically in terms of the model parameters via the Lyapunov equation

$$\mathbf{A} \cdot \Sigma_{\text{noise}} + \Sigma_{\text{noise}} \cdot \mathbf{A}^T + \mathbf{\Lambda}_{\text{noise}} = 0. \quad (40)$$

In equation (40) [A](#) is a matrix encoding the model dynamics (see equation (70) [A](#), below), and Λ_{noise} is diagonal matrix comprising noise variances $(\frac{2\sigma_I^2}{\tau_I}, \frac{2\sigma_r^2}{\tau_r}, \frac{2\sigma_y^2}{\tau_y})$ on the diagonal. The relevant term $\sigma_{\text{noise},y}^2$ is found at the last row and last column of Σ_{noise} .

In all simulations, learning rates for the classical rules were selected via grid search to minimize output error.

Derivation of the Bayesian plasticity rule

Problem setting

Given the model dynamics described by equations (7) [A](#) and (9) [A](#), we consider a general task in which the goal is choose weights, w_i , to match the output, y , to a time-varying target, y^* . We assume y^* is generated via deterministic dynamics identical to y , using a set of target weights w_i^* . Following Aitchison et al. (2021) [A](#), to represent environmental variability or ongoing changes in the local circuit, we assume the mapping between input and target output drifts slowly over time, modeled as random drift of the target weights about a constant mean, μ_w ,

$$\tau_w \dot{w}_i^* = -(w_i^* - \mu_w) + \sigma_w \sqrt{2\tau_w} \xi_{w,i}. \quad (41)$$

The drift timescale, $\tau_w \sim 1000$ s, is orders of magnitude slower than the neuronal and output dynamics.

The above assumptions about the relationship of target weights to y^* , and the drift described by equation (41) [A](#), are exact in the teacher-student learning paradigm. In the cerebellar learning simulations, target weights are instead defined implicitly as weight values that would solve the task.

Error minimization from the perspective of a single synapse

As in the simple online linear regression model, we decompose the weights into independent slow and fast components, $w_i = \bar{w}_i + \delta w_i$. The fast weights, δw_i , will be responsible for suppressing error, while the slow weights, \bar{w}_i , gradually converge to the target weights, thereby reducing the magnitude of ongoing corrections. However, with the more realistic dynamics described by equations (7) [A](#) and (9) [A](#), writing down the plasticity rule is considerably more challenging. That's because the mapping from the weights, w_i , to the output, y , is more complicated: rather than a simple sum, as in equation (1) [A](#), there are several differential equations separating the weights from the output (equations (7) [A](#) and (9) [A](#)).

To derive learning rules in this more complicated setting, we start by introducing a set of variables equal to the difference between actual and target quantities,

$$\delta_I \equiv I - I^* \quad (42a)$$

$$\delta_r \equiv r - r^* \quad (42b)$$

$$\delta_y \equiv y - y^*. \quad (42c)$$

From [equations \(7\)](#) and [\(9\)](#), these evolve via

$$\tau_I \dot{\delta}_I = -\delta_I + \sum_j (w_j - w_j^*) x_j + \sigma_I \sqrt{2\tau_I} \xi_I \quad (43a)$$

$$\tau_r \dot{\delta}_r = -\delta_r + \alpha \delta_I + \sigma_r \sqrt{2\tau_r} \xi_r \quad (43b)$$

$$\tau_y \dot{\delta}_y = -\delta_y + \delta_r + \sigma_y \sqrt{2\tau_y} \xi_y. \quad (43c)$$

The target weights, w_j^* , are unknown, so on the surface these equations don't seem especially useful. However, the synapses have access to the feedback, f , which is a noisy version of δ_y (see [equation \(10\)](#)). Thus, they can use these equations to estimate w_j^* based on the mismatch between δ_y and f , and update the actual weights, w_j , with the aim of making δ_y as small as possible.

However, there's a problem: δ_y depends on the activity of all the synapses; information that any one synapse doesn't have. To remedy this, we'll work from the perspective of synapse i , and model δ_y in terms of local variables.

Decomposing synaptic weights into slow and fast components, the sum in [equation \(43a\)](#) can be expanded as

$$\sum_j (w_j - w_j^*) x_j = (\bar{w}_i - w_i^*) x_i + \sum_{j \neq i} (\bar{w}_j - w_j^*) x_j + \sum_j \delta w_j x_j. \quad (44)$$

The first term depends only on synapse i , so it's local. The second term represents a source of error due to the other $N - 1$ unobserved synapses. For Poisson input and large N , this can be approximated as Gaussian white noise ([Fourcaud and Brunel 2002](#)),

$$\sum_{j \neq i} (\bar{w}_j - w_j^*) x_j \approx \sigma_s \xi_s; \quad \sigma_s^2 = 2(N - 1) \sigma_w^2 \nu. \quad (45)$$

As above, ν denotes the expected input rate per synapse.

For the third term in [equation \(44\)](#), we view the sum over fast weights as approximating a single scalar control variable, denoted u , later to be chosen to minimize output error. Assuming that all of the synapses can compute u locally, by setting the fast weights uniformly as

$$\delta w_j = \frac{u}{N\nu} \quad (46)$$

the third term in [equation \(44\)](#) can be expressed as

$$\sum_j \delta w_j x_j \approx u \quad (47)$$

where the approximation is good in the large N limit. Inserting [equations \(45\)](#) and [\(47\)](#) into [equation \(44\)](#), and inserting that into [equation \(43\)](#), we arrive at a set of equations that uses only information local to synapse i ,

$$\tau_I \dot{\delta}_{I,i} = -\delta_{I,i} + (\bar{w}_i - w_i^*) x_i + u + \sigma_s \xi_s + \sigma_I \sqrt{2\tau_I} \xi_I \quad (48a)$$

$$\tau_r \dot{\delta}_{r,i} = -\delta_{r,i} + \alpha \delta_{I,i} + \sigma_r \sqrt{2\tau_r} \xi_r \quad (48b)$$

$$\tau_y \dot{\delta}_{y,i} = -\delta_{y,i} + \delta_{r,i} + \sigma_y \sqrt{2\tau_y} \xi_y \quad (48c)$$

Equation (48) is a model of the error that the synapse can use to guide plasticity. In our rule, the synapse learns by updating its slow weight, \bar{w}_i , to be equal to an evolving estimate of the target, w_i^* , while choosing u , and thereby the fast weight, δw_i , to cancel out the remaining error.

Optimal control solution

Because the noise in equation (48) is Gaussian and we want to minimize the squared error, $(y - y^*)^2$, the problem of finding the optimal u reduces to the well-known linear-quadratic-Gaussian (LQG) control problem (Crassidis and Junkins 2011). In this setting, the optimal control, denoted \hat{u} , satisfies

$$\hat{u} = \underset{u}{\operatorname{argmin}} \mathbb{E} \left[\frac{1}{T} \int_0^T (y - y^*)^2 + \lambda_u u^2 dt \right]. \quad (49)$$

Note that this includes a control cost, $\lambda_u u^2$, which limits the size of \hat{u} ; without that cost, \hat{u} would diverge. In general, LQG control employs a Bayesian filter to estimate unobserved model variables, and then uses those estimates to construct the controller.

To solve equation (49) using standard methods, we first need to express our model for the error (equation (48)) and feedback signal (equation (10)) in a canonical linear state space form. We start by combining the model variables into a single vector,

$$\Phi_i \equiv \begin{bmatrix} w_i^* - \mu_w \\ \delta_{I,i} \\ \delta_{r,i} \\ \delta_{y,i} \end{bmatrix}. \quad (50)$$

Then, we express the control variable as a vector, \mathbf{u}_i , and make a change of variables that will clean up an offset term,

$$\mathbf{v}_i \equiv \mathbf{u}_i + \begin{bmatrix} 0 \\ (\bar{w}_i - \mu_w)x_i \\ 0 \\ 0 \end{bmatrix}. \quad (51)$$

Introducing the vector

$$\mathbf{H} \equiv \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (52)$$

the dynamics of Φ_i and feedback, f , can be expressed together as

$$\dot{\Phi}_i = \mathbf{C}_i \cdot \Phi_i + \mathbf{B} \cdot \mathbf{v}_i + \xi^t \quad (53a)$$

$$f = \mathbf{H} \cdot \Phi_i + \sigma_f \xi_f \quad (53b)$$

where

$$\mathbf{C}_i \equiv \begin{bmatrix} -\frac{1}{\tau_w} & 0 & 0 & 0 \\ -\frac{x_i}{\tau_I} & -\frac{1}{\tau_I} & 0 & 0 \\ 0 & \frac{\alpha}{\tau_r} & -\frac{1}{\tau_r} & 0 \\ 0 & 0 & \frac{1}{\tau_y} & -\frac{1}{\tau_y} \end{bmatrix} \quad (54a)$$

$$\mathbf{B} \equiv \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\tau_I} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (54b)$$

and ξ is Gaussian white noise with covariance matrix

$$\Lambda' = \begin{bmatrix} \frac{2\sigma_w^2}{\tau_w} & 0 & 0 & 0 \\ 0 & \frac{\sigma_f^2}{\tau_f} + \frac{2\sigma_f^2}{\tau_I} & 0 & 0 \\ 0 & 0 & \frac{2\sigma_r^2}{\tau_r} & 0 \\ 0 & 0 & 0 & \frac{2\sigma_y^2}{\tau_y} \end{bmatrix}. \quad (55)$$

With this set up, the control problem in equation (49) can be written

$$\hat{\mathbf{v}}_i = \underset{\mathbf{v}_i}{\operatorname{argmin}} \mathbb{E} \left[\frac{1}{T} \int_0^T \Phi_i \cdot \mathbf{H} \mathbf{H} \cdot \Phi_i + \lambda_u \mathbf{v}_i \cdot \mathbf{v}_i dt \right] \quad (56)$$

where $\mathbf{H} \mathbf{H}$ is an outer product.

Although the control variable, \mathbf{v}_i , is now a vector instead of a scalar, u , the optimal solution will still only have one non-zero component (the second one). This is because multiplication by \mathbf{B} in equation (53a) means that only the second component of \mathbf{v}_i will influence the output, so the control cost term in equation (56) will force the other components to be zero. The scalar \hat{u}_i is recovered from $\hat{\mathbf{v}}_i$ by selecting the second component and then reversing the variable change in equation (51),

$$\hat{u}_i = \hat{v}_{i,2} - (\bar{w}_i - \mu_w)x_i \quad (57)$$

where the subscript 2 denotes the second component of the vector.

Nevertheless, equation (56) is still not identical to equation (49), because the variable change in equation (51) has shifted the quantity $(\bar{w}_i - \mu_w)x_i$ from the first term in the integral to the second term. This means λ_u parameterizes a slightly different control cost. We will ignore this detail, however, as it doesn't influence any of our modeling choices or the interpretation of results.

Note also that we have written the control variable with a subscript i , since at this stage the solution depends on local variables. We will later make approximations that remove this dependence, so that the control variables are the same for all synapses, as assumed in equation (46).

Ignoring biological constraints, LQG theory gives the solution to the problem defined by equations (50)–(56) as

$$\dot{\hat{\Phi}}_i = \mathbf{C}_i \cdot \hat{\Phi}_i + \mathbf{B} \cdot \hat{\mathbf{v}}_i + \mathbf{K}_i (f - \mathbf{H} \cdot \hat{\Phi}_i) \quad (58a)$$

$$\hat{\mathbf{v}}_i = -\mathbf{L}_i \cdot \hat{\Phi}_i \quad (58b)$$

The vector $\hat{\Phi}_i$ is a Bayesian (Kalman filter) estimate of the values of the unobserved variables in equation (50). \mathbf{K}_i and \mathbf{L}_i are optimal feedback and control gains, given by

$$\mathbf{K}_i = \frac{1}{\sigma_f^2} \mathbf{P}_i \cdot \mathbf{H} \quad (59a)$$

$$\mathbf{L}_i = \frac{1}{\lambda_u} \mathbf{B} \cdot \mathbf{S}_i \quad (59b)$$

where \mathbf{P}_i and \mathbf{S}_i are determined by a pair of matrix Riccati equations (Crassidis and Junkins 2011 [\[4\]](#)),

$$\dot{\mathbf{P}}_i = \mathbf{C}_i \cdot \mathbf{P}_i + \mathbf{P}_i \cdot \mathbf{C}_i^T - \frac{1}{\sigma_f^2} \mathbf{P}_i \cdot \mathbf{H} \mathbf{H}^T \cdot \mathbf{P}_i + \mathbf{A}', \quad \mathbf{P}_i(0) = \mathbf{P}_0 \quad (60a)$$

$$-\dot{\mathbf{S}}_i = \mathbf{C}_i^T \cdot \mathbf{S}_i + \mathbf{S}_i \cdot \mathbf{C}_i - \frac{1}{\lambda_u} \mathbf{S}_i \cdot \mathbf{B} \cdot \mathbf{B}^T \cdot \mathbf{S}_i + \mathbf{H} \mathbf{H}^T, \quad \mathbf{S}_i(T) = \mathbf{S}_T. \quad (60b)$$

The equation for \mathbf{P}_i , a covariance matrix representing uncertainty about the estimate $\hat{\boldsymbol{\Phi}}_i$, is solved forwards in time from an initial condition, whereas the equation for \mathbf{S}_i , used to compute the optimal control, is solved backwards in time from a terminal condition.

While these equations are exact, they are far from biologically plausible. In the next several sections we remedy that by making several approximations.

Control gain approximation

Solving equation (60b) [\[4\]](#) backwards in time poses a problem for online implementation, because the dynamics matrix, \mathbf{C}_i (equation (54a) [\[4\]](#)), depends on the synaptic input, x_i , which cannot be known in advance. We therefore simply drop the term x_i from the matrix \mathbf{C}_i . With this approximation and sufficiently large T , equation (60b) [\[4\]](#) relaxes to a steady state that is independent of the terminal condition and synapse index. The steady-state matrix \mathbf{S} is then defined implicitly as the positive-semidefinite solution to the algebraic Riccati equation

$$\mathbf{C}^T \cdot \mathbf{S} + \mathbf{S} \cdot \mathbf{C} - \frac{1}{\lambda_u} \mathbf{S} \cdot \mathbf{B} \cdot \mathbf{B}^T \cdot \mathbf{S} + \mathbf{H} \mathbf{H}^T = 0 \quad (61)$$

where \mathbf{C} is the same as \mathbf{C}_i but with x_i set to zero.

Writing equation (61) [\[4\]](#) in component form, and using the fact that \mathbf{S} is positive-semidefinite, straight-forward algebra shows that the first row and column of \mathbf{S} are all zero. Consequently, because of the structure of \mathbf{B} (equation (54b) [\[4\]](#)), the only nonzero elements of \mathbf{L}_i are the second through fourth elements of the second row. That in turn implies that only the second element of $\hat{\mathbf{v}}_i$ (equation (58b) [\[4\]](#)) is nonzero. That element is given by

$$\hat{v}_{i,2} = -\mathcal{L} \cdot \hat{\boldsymbol{\delta}}_i \quad (62)$$

where \mathcal{L} is proportional to the second through fourth elements of the second row of \mathbf{S} ,

$$\mathcal{L} \equiv \frac{1}{\lambda_u T} \mathbf{S}_{(2,2:4)}, \quad (63)$$

and $\hat{\boldsymbol{\delta}}_i$ consists of the second through fourth elements of $\hat{\boldsymbol{\Phi}}_i$,

$$\hat{\boldsymbol{\delta}}_i \equiv \hat{\boldsymbol{\Phi}}_{i(2:4)}. \quad (64)$$

We can now use equation (58b) [\[4\]](#) to derive an explicit expression for the control variable, \hat{u}_i ,

$$\begin{aligned} \hat{u}_i &= -\mathcal{L} \cdot \hat{\boldsymbol{\delta}}_i - (\bar{w}_i - \mu_w) x_i \\ &\approx -\mathcal{L} \cdot \hat{\boldsymbol{\delta}}_i. \end{aligned} \quad (65)$$

In the approximation in the last line, we ignored the second term, which will allow us to make the controller independent of i , and makes little difference to performance in practice.

We solve [equation \(61\)](#) numerically to obtain the control gain for simulations. An analytical approximation is provided in [equation \(91\)](#) below, for the simpler version of the plasticity rule presented in the main text; that approximation also works well in practice (**Fig. S1**).

Feedback gain approximation

Although we now have all the ingredients for a local online plasticity rule, we also approximate the feedback gain \mathbf{K}_i for the sake of interpretability.

We first unpack **equation (60a)** by partitioning the state-estimate covariance matrix into blocks,

$$\mathbf{P}_i = \begin{bmatrix} \sigma_i^2 & \mathbf{z}_i^\top \\ \mathbf{z}_i & \boldsymbol{\Sigma}_i \end{bmatrix} \quad (66)$$

where, to highlight the functional roles that each block will play in the plasticity rule, we introduced the variables

$$\sigma_i^2 \equiv P_{i,(1,1)} \quad (67a)$$

$$\mathbf{z}_i \equiv \mathbf{P}_{i,(2:4,1)} \quad (67b)$$

$$\boldsymbol{\Sigma}_i \equiv \mathbf{P}_{i,(2:4,2:4)} \quad (67c)$$

where the numeric subscripts on the right-hand side denote the range of row and column indices for each block. The variable σ_i^2 represents the uncertainty about the target weight, \mathbf{z}_i represents the covariance between the target weight and error vector (see **equation (64)**), which will act as an eligibility trace in the plasticity rule, and $\boldsymbol{\Sigma}_i$ represents the uncertainty about the estimate of the error vector.

Defining

$$\mathbf{h} \equiv \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (68)$$

and using the definitions in **equations (70)** and **(71)**, **equation (60a)** can be expanded to give the dynamics of each of the blocks as

$$\dot{\sigma}_i^2 = -\frac{1}{\sigma_f^2}(\mathbf{z}_i \cdot \mathbf{h})^2 - \frac{2}{\tau_w}(\sigma_i^2 - \sigma_w^2) \quad (69a)$$

$$\dot{\mathbf{z}}_i = \left(\mathbf{A} - \frac{1}{\tau_w} \mathbf{I} \right) \cdot \mathbf{z}_i - \frac{1}{\sigma_f^2}(\mathbf{z}_i \cdot \mathbf{h})\boldsymbol{\Sigma}_i \cdot \mathbf{h} - \begin{bmatrix} \frac{1}{\tau_f} \sigma_i^2 x_i \\ 0 \\ 0 \end{bmatrix} \quad (69b)$$

$$\dot{\boldsymbol{\Sigma}}_i = \mathbf{A} \cdot \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_i \cdot \mathbf{A}^\top - \frac{1}{\sigma_f^2} \boldsymbol{\Sigma}_i \cdot \mathbf{h} \mathbf{h} \cdot \boldsymbol{\Sigma}_i - \frac{1}{\tau_I} ([\mathbf{z}_i \mathbf{0}^{(3 \times 2)}] + [\mathbf{z}_i \mathbf{0}^{(3 \times 2)}]^\top) x_i + \boldsymbol{\Lambda} \quad (69c)$$

where $\mathbf{0}_{(3 \times 2)}$ denotes a 3×2 matrix of zeros, \mathbf{A} is the lower right $(2 : 4) \times (2 : 4)$ block of \mathbf{C}_i ,

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{\tau_I} & 0 & 0 \\ \frac{\alpha}{\tau_r} & -\frac{1}{\tau_r} & 0 \\ 0 & \frac{1}{\tau_y} & -\frac{1}{\tau_y} \end{bmatrix} \quad (70)$$

and Λ is the lower right $(2:4) \times (2:4)$ block of Λ' ,

$$\Lambda = \begin{bmatrix} \frac{\sigma_x^2}{\tau_f^2} + \frac{2\sigma_f^2}{\tau_f} & 0 & 0 \\ 0 & \frac{2\sigma_r^2}{\tau_r} & 0 \\ 0 & 0 & \frac{2\sigma_y^2}{\tau_y} \end{bmatrix}. \quad (71)$$

Because **equations (69a)** and **(69b)** track uncertainty related to the target weight, their temporal dynamics are strongly dependent on the local synaptic input, x_i – this is what that couples the weight to the observed feedback. By contrast, **equation (69c)** tracks the uncertainty of δ_i , which is dominated by uncertainty about the activity of unobserved synapses, so depends only weakly on x_i . Mathematically, this is reflected in the fourth term of **equation (69c)** being dominated by the order- N synaptic noise term in Λ (see **equations (45)** and **(71)**). We can, therefore, ignore the x_i -dependent term in **equation (69c)**. Then, Σ_i relaxes to a steady state that is independent of the synapse index. That steady state is the solution to the algebraic Riccati equation,

$$\mathbf{A} \cdot \Sigma + \Sigma \cdot \mathbf{A}^\top - \frac{1}{\sigma_f^2} \Sigma \cdot \mathbf{h} \mathbf{h}^\top \cdot \Sigma + \Lambda = 0. \quad (72)$$

We use **equation (72)** in place of (69c). But to capture the most important dependence on input, x_i , we solve **equations (69a)** and **(69b)** individually for each synapse.

Finally, we can express \mathbf{K}_i , **equation (59a)**, in terms of the quantities in **equation (69)**. Using the fact that \mathbf{K}_i is proportional to the fourth column of \mathbf{P}_i (see **equation (52)** for \mathbf{H}), the first component is given by

$$K_{i,1} = \frac{1}{\sigma_f^2} \mathbf{z}_i \cdot \mathbf{h}, \quad (73)$$

which is used to update the estimate of the target weight, \hat{w}_i^* . The remaining three components, used update the estimate of the error vector, $\hat{\delta}_i$, are given by the constant vector

$$\mathcal{K} \equiv \mathbf{K}_{i,(2:4)} = \frac{1}{\sigma_f^2} \Sigma \cdot \mathbf{h}. \quad (74)$$

Computations that should be performed by a synapse

Bringing everything together, we express the plasticity rule as a system of differential equations that optimally process local signals to enable concurrent control and learning.

Unpacking the error estimate from **equation (58a)**, expressing the feedback gain via **equations (72)** and **(74)**, and approximating $\mathbf{A} - \frac{1}{\tau_w} \mathbf{I} \approx \mathbf{A}$, which is valid because¹ is orders of magnitude smaller than other terms on the diagonal of \mathbf{A} , we get

$$\dot{\hat{\delta}}_i = \mathbf{A} \cdot \hat{\delta}_i + \begin{bmatrix} \frac{1}{\tau_f} (\bar{w}_i - \hat{w}_i^*) x_i + \frac{1}{\tau_f} \hat{u}_i \\ 0 \end{bmatrix} + \mathcal{K} (f - \mathbf{h} \cdot \hat{\delta}_i). \quad (75)$$

The only remaining issue is that the control variable, $\hat{u}_i = -\mathcal{L} \cdot \hat{\delta}_i$ (**equation (65)**), still depends on the synapse index via $\hat{\delta}_i$. As discussed when introducing the fast weights in **equation (46)**, we would like this to be identical for all synapses. To address this, we need only make the obvious choice of setting \bar{w}_i equal to its estimated target \hat{w}_i^* at all times. This makes the i -dependent term in the brackets in **equation (75)** vanish, and so too the i -dependence of $\hat{\delta}_i$ (assuming the system has been running long enough that initial conditions are forgotten).

Setting $\bar{w}_i = \hat{w}_i^*$, thereby dropping the synapse index on $\hat{\delta}_i$ and \hat{u}_i , and unpacking the rest of **equations (58a)** and **(69)** leads to

$$\dot{\hat{\delta}} = \mathbf{A} \cdot \hat{\delta} + \begin{bmatrix} \frac{1}{\tau_I} \hat{u} \\ 0 \\ 0 \end{bmatrix} + \mathcal{K}(f - \mathbf{h} \cdot \hat{\delta}) \quad (76a)$$

$$\dot{\hat{w}}_i^* = -\frac{1}{\tau_w}(\hat{w}_i^* - \mu_w) + \frac{\mathbf{z}_i \cdot \mathbf{h}}{\sigma_f^2}(f - \mathbf{h} \cdot \hat{\delta}) \quad (76b)$$

$$\dot{\sigma}_i^2 = -\frac{2}{\tau_w}(\sigma_i^2 - \sigma_w^2) - \frac{1}{\sigma_f^2}(\mathbf{z}_i \cdot \mathbf{h})^2 \quad (76c)$$

$$\dot{\mathbf{z}}_i = \mathbf{A} \cdot \mathbf{z}_i - \mathcal{K}(\mathbf{z}_i \cdot \mathbf{h}) - \begin{bmatrix} \frac{1}{\tau_I} \sigma_i^2 x_i \\ 0 \\ 0 \end{bmatrix}. \quad (76d)$$

Altogether, this gives synaptic weights

$$\begin{aligned} w_i &= \bar{w}_i + \delta w_i \\ &= \hat{w}_i^* - \frac{1}{N\nu} \mathcal{L} \cdot \hat{\delta}. \end{aligned} \quad (77)$$

Plasticity rule approximation for $\tau_I, \tau_r \ll \tau_y$

Control is most effective when the downstream time constant is large. This regime also permits a simple approximation of the plasticity rule that makes parameter dependencies explicit. We use these equations in the main text for ease of intuition (**equations (13a) – (16b)**). Here we provide the details of the approximation.

Starting from **equations (76a)** and **(76d)**, we express the components of the vectors $\hat{\delta}$ and \mathbf{z}_i using I, r and y subscripts. Assuming $\tau_I, \tau_r \ll \tau_y$, and using **equation (70)** for \mathbf{A} , we treat the I and r components as reacting instantaneously to their inputs,

$$\hat{\delta}_I \approx \hat{u} + \tau_I \mathcal{K}_I(f - \hat{\delta}_y) \quad (78a)$$

$$\hat{\delta}_r \approx \alpha \hat{\delta}_I + \tau_r \mathcal{K}_r(f - \hat{\delta}_y) \quad (78b)$$

$$z_{I,i} \approx -\tau_I \mathcal{K}_I z_{y,i} - \sigma_i^2 x_i \quad (78c)$$

$$z_{r,i} \approx \alpha z_{I,i} - \tau_r \mathcal{K}_r z_{y,i} \quad (78d)$$

where the subscripts on \mathcal{K} indicate their components,

$$\mathcal{K} \equiv \begin{bmatrix} \mathcal{K}_I \\ \mathcal{K}_r \\ \mathcal{K}_y \end{bmatrix}. \quad (79)$$

With these approximations, **equation (76)** becomes

$$\tau_y \dot{\hat{\delta}}_y \approx -\hat{\delta}_y + \alpha \hat{u} + \mathcal{K}(f - \hat{\delta}_y) \quad (80a)$$

$$\dot{\hat{w}}_i^* \approx -\frac{1}{\tau_w}(\hat{w}_i^* - \mu_w) + \frac{z_{y,i}}{\sigma_f^2}(f - \hat{\delta}_y) \quad (80b)$$

$$\dot{\sigma}_i^2 \approx -\frac{2}{\tau_w}(\sigma_i^2 - \sigma_w^2) - \frac{z_{y,i}^2}{\sigma_f^2} \quad (80c)$$

$$\tau_y \dot{z}_{y,i} \approx -(1 + \mathcal{K})z_{y,i} - \alpha \sigma_i^2 x_i \quad (80d)$$

where

$$\mathcal{K} \equiv \alpha \tau_I \mathcal{K}_I + \tau_r \mathcal{K}_r + \tau_y \mathcal{K}_y \quad (81)$$

is a weighted sum of the three components of the Kalman gain ([equation \(74\)](#)). [Equation \(80\)](#) has been copied to the main-text [equations \(13\)](#) and [\(16\)](#) with some minor cosmetic changes; to clean up the presentation, there we have dropped the y subscripts, explicitly denoted $\hat{w}_i^* = \bar{w}_i$, and made a sign change for the eligibility trace variable, $z_i \equiv -z_{y,i}$.

To compute \mathcal{K} , we note first of all that

$$\mathcal{K} = \tau \cdot \mathcal{K} = \frac{1}{\sigma_f^2} \tau \cdot \Sigma \cdot \mathbf{h} \quad (82)$$

where the second equality comes from [equation \(74\)](#) and we have defined

$$\tau \equiv \begin{bmatrix} \alpha \tau_I \\ \tau_r \\ \tau_y \end{bmatrix}. \quad (83)$$

To derive an explicit expression for $\tau \cdot \Sigma \cdot \mathbf{h}$, we operate on both sides of [equation \(72\)](#) with τ , giving us

$$\tau \cdot \mathbf{A} \cdot \Sigma \cdot \tau + \tau \cdot \Sigma \cdot \mathbf{A}^T \cdot \tau - \frac{1}{\sigma_f^2} \tau \cdot \Sigma \cdot \mathbf{h} \mathbf{h}^T \cdot \Sigma \cdot \tau + \tau \cdot \mathbf{A} \cdot \tau = 0. \quad (84)$$

Then, combining the fact that $\tau \cdot \mathbf{A} = -\mathbf{h}$ with [equation \(82\)](#), we arrive at

$$-2\sigma_f^2 \mathcal{K} - \sigma_f^2 \mathcal{K}^2 + \sigma_\delta^2 = 0, \quad (85)$$

Where

$$\sigma_\delta^2 = \alpha^2 (\sigma_s^2 + 2\tau_I \sigma_I^2) + 2\tau_r \sigma_r^2 + 2\tau_y \sigma_y^2. \quad (86)$$

Solving [equation \(85\)](#) gives the scalar feedback gain explicitly as

$$\mathcal{K} = \sqrt{1 + \frac{\sigma_\delta^2}{\sigma_f^2}} - 1. \quad (87)$$

For the control gain, we make use of the duality of state-estimation and control for the LQG problem ([Crassidis and Junkins 2011](#)), which means that knowing the parameters of the equation for $\hat{\delta}_y$ ([equation \(80a\)](#)) allows us to easily write down the Riccati equation needed to compute the corresponding controller. The analogue of [equation \(61\)](#), now for the scalar variable s , and with equivalent scalar parameters $c = -\frac{1}{\tau_y}$, $b = \frac{\alpha}{\tau_y}$ and $h = 1$, in place of \mathbf{C} , \mathbf{B} and \mathbf{H} , is

$$-\frac{2}{\tau_y} s - \frac{\alpha^2}{\lambda_{u'} \tau_y^2} s^2 + 1 = 0. \quad (88)$$

Defining

$$\mathcal{L} \equiv \frac{\alpha}{\lambda_{u'} \tau_y} s \quad (89)$$

as the analogue of [equation \(59b\)](#) then leads to

$$-\frac{2\lambda_{u'}}{\alpha} \mathcal{L} - \lambda_{u'} \mathcal{L}^2 + 1 = 0, \quad (90)$$

and thus a scalar control gain

$$\mathcal{L} = \sqrt{\frac{1}{\alpha^2} + \frac{1}{\lambda_{u'}}} - \frac{1}{\alpha}. \quad (91)$$

Here we have used the notation $\lambda_{u'}$ to signify that the control cost parameter is not identical to λ_u used above, and is fit separately in simulations of the approximate rule.

Inserting the optimal control

$$\hat{u} = -\mathcal{L}\hat{\delta}_y \quad (92)$$

into [equation \(80a\)](#) yields the plasticity rule corresponding to [equations \(13\)](#) and [\(16\)](#) in the main text.

Steady-state learning rate

Using the approximate plasticity rule given in [equation \(16\)](#), we can derive an expression for how the steady-state learning rate of a Bayesian synapse depends on key parameters, and especially on the firing rate.

The first observation is that because \mathcal{N} is typically large, after a spike z_i decays rapidly, and so whenever a new spike arrives it's effectively zero. Thus, assuming for the moment that σ_i is constant, [equation \(16c\)](#) tells us that when a spike occurs at time $t = 0$, $z_i(t)$ is given by (for times $t > 0$)

$$z_i(t) = \frac{\alpha\sigma_i^2}{\tau_y} e^{-(1+\mathcal{K})t/\tau_y}. \quad (93)$$

Inserting this into [equations \(16a\)](#) and [\(16b\)](#), and assuming for the moment that both are constant on a timescale of τ_y/\mathcal{N} , we see that when a spike occurs they change by

$$\Delta\bar{w}_{i,\text{spike}} = -\frac{\alpha\sigma_i^2}{(1+\mathcal{K})\sigma_f^2} (f - \hat{\delta}) + \text{noise} \quad (94a)$$

$$\Delta\sigma_{i,\text{spike}}^2 = -\frac{\alpha^2\sigma_i^4}{2\tau_y(1+\mathcal{K})\sigma_f^2} \quad (94b)$$

where the subscript ‘spike’ indicates that these are changes in response to a spike.

In the large \mathcal{N} limit, both $\Delta\bar{w}_{i,\text{spike}}$ and $\Delta\sigma_{i,\text{spike}}^2$ are small, justifying our assumption that w_i and σ_i^2 are approximately constant during a spike. The noise term in [equation \(94a\)](#) arises because the feedback error, f , has a white noise component (see [equation \(10\)](#)), so fluctuates around the relevant signal, $y - y^*$.

Because [equation \(94a\)](#) tells us the weight change per pre-synaptic spike, we can identify the term in front of $(f - \hat{\delta})$ as the effective learning rate,

$$\eta_i^{\text{eff}} = \frac{\alpha\sigma_i^2}{(1+\mathcal{K})\sigma_f^2}. \quad (95)$$

To derive an explicit expression for η_i^{eff} , we need the steady state value of σ_i . Using [equation \(16b\)](#), that's given by

$$\begin{aligned}\sigma_i^2 &= \sigma_w^2 - \frac{\tau_w}{2\sigma_f^2} \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T dt z_i^2(t). \\ &= \sigma_w^2 - \frac{\tau_w}{\tau_y} \frac{\nu_i \alpha^2 \sigma_i^4}{4(1 + \mathcal{K})\sigma_f^2}\end{aligned}\tag{96}$$

where we used [equation \(93\)](#) for the time course of $z_i(t)$ after a spike and, recall, ν_i is the input firing rate to synapse i . Note the slight abuse of notation: above, and in what follows in this section, σ_i^2 is the steady state variance; to reduce clutter we don't make this explicit. Solving [equation \(96\)](#) gives us

$$\sigma_i^2 = \frac{\tau_y}{\tau_w} \frac{2(1 + \mathcal{K})\sigma_f^2}{\nu_i \alpha^2} \left(\sqrt{1 + \frac{\tau_w}{\tau_y} \frac{\nu_i \alpha^2 \sigma_w^2}{(1 + \mathcal{K})\sigma_f^2}} - 1 \right).\tag{97}$$

For typical parameter values (see [Table 1](#)), and using the definitions in [equations \(86\)](#) and [\(87\)](#), we find that the second term inside the square root is large ($\sim 10^3 - 10^4$) as long as input rates, ν_i , are greater than ~ 0.1 spikes/s. [Equation \(97\)](#) thus simplifies to

$$\sigma_i^2 \approx 2 \sqrt{\frac{\tau_y}{\tau_w} \frac{(1 + \mathcal{K})\sigma_f^2 \sigma_w^2}{\nu_i \alpha^2}}.\tag{98}$$

Inserting this into [equation \(95\)](#), we arrive at

$$\eta_i^{\text{eff}} \approx 2 \sqrt{\frac{\tau_y}{\tau_w} \frac{\sigma_w^2}{(1 + \mathcal{K})\sigma_f^2 \nu_i}} \sim \frac{1}{\sqrt{\nu_i}}.\tag{99}$$

Notably, this scaling with input rate is consistent with the results of [Aitchison et al. \(2021\)](#), which they demonstrated was consistent with experiments.

Spiking feedback

We now consider the case where feedback is communicated via spikes. The spiking feedback signal, f_S , is modeled as an inhomogeneous Poisson processes with rate γ , as described by [equation \(17\)](#). The derivation is essentially the same as for the continuous feedback case above; the difference is that it's not a textbook problem, so we have to do it ourselves. The approach, though, is relatively standard: we compute the time evolution of the mean and covariance of Φ_i conditioned on feedback spikes. These correspond to $\hat{\Phi}_i$ and \mathbf{P}_i above, but their time evolution is, of course, different because of the spiking feedback.

Model discretization and notation

To model feedback spikes, we discretize time into bins of size Δt , and assume that there are either zero or one spike in each bin; this assumption is valid because we will eventually take the limit $\Delta t \rightarrow 0$. Our first step is to turn [equation \(53a\)](#) into discrete time updates rather than differential equations.

Parameter	Definition	Value
N	number of synapses	20 (Fig. 1), 1000 (Figs. 2, 3), 100 (Figs. 4, 5)
M	number of neurons	1 (Figs. 1-3, S1, S2, unless stated), 20 (Figs. 4, 5, S3)
T	task duration	10 s (Fig. 1), 500 s (Figs. 2, S1), 5000 s (Figs. 3, 4, S2)
Δt	simulation time step	1 ms (Fig. 1), 0.1 ms (Figs. 2-5, S1-S3)
τ_I	synaptic time constant	5 ms (Figs. 2-5, S2, S3), 1 ms (Fig. S1)
τ_r	firing rate time constant	50 ms (Figs. 2-5, S2, S3), 10 ms (Fig. S1)
τ_y	output process time constant	100 ms (unless stated)
τ_w	weight drift time constant	10^3 s (Figs. 2, S1), 10^4 s (Figs. 3-5, S2, S3)
α	current-rate gain	50
ν_{\max}	maximum input rate	50 spikes/s
μ_w	synaptic weight mean	0.01 (Figs. 2, 3, S1, S2), 0.016 (Figs. 4, 5, S3)
σ_w^2	synaptic weight variance	0.025 (Figs. 2-5, S1-S3)
σ_I^2	current noise variance	$0.05/M$ (Figs. 2-5, S1-S3)
σ_r^2	rate noise variance	$0.05/M$ (Figs. 2-5, S1-S3)
σ_y^2	output noise variance	0.05 (Figs. 2-5, S1-S3)
σ_f^2	continuous feedback noise variance	0.5 (Figs. 2, S1)
γ_0	spiking-feedback spontaneous rate	2 spikes/s (unless stated)
ρ	spiking-feedback response steepness	0.4

Table 1

Model and simulation parameters.

Discrete time variables are distinguished from their continuous versions with a t subscript, and several quantities are rescaled by the time step to reduce clutter. We define

$$\tilde{\mathbf{C}}_{i,t} \equiv \mathbf{I} + \Delta t \mathbf{C}_i \quad (100a)$$

$$\tilde{\mathbf{B}} \equiv \Delta t \mathbf{B} \quad (100b)$$

$$\tilde{\xi}'_t \equiv \sqrt{\Delta t} \xi' \quad (100c)$$

$$\tilde{\Lambda}' \equiv \Delta t \Lambda'. \quad (100d)$$

The synaptic inputs $x_{i,t}$ are implemented as discrete-time delta functions, taking the value $1/\Delta t$ when there is a spike and zero otherwise. The discrete-time version of [equation \(53a\)](#) is, then, given by

$$\Phi_{i,t+\Delta t} = \tilde{\mathbf{C}}_{i,t} \cdot \Phi_{i,t} + \tilde{\mathbf{B}} \cdot \mathbf{v}_{i,t} + \tilde{\xi}'_t. \quad (101)$$

For Poisson feedback with the rate determined by [equation \(17\)](#), the analogue of [equation \(53b\)](#) is an expression for the probability of observing a feedback spike given Φ_i . Assuming small Δt , and treating spikes as Bernoulli random variables $f_{S,t} \in \{0, 1\}$, this is given by

$$P(f_{S,t}|\Phi_{i,t}) = f_{S,t} \gamma_0 e^{\rho \Phi_{i,t} \cdot \mathbf{H}} \Delta t + (1 - f_{S,t})(1 - \gamma_0 e^{\rho \Phi_{i,t} \cdot \mathbf{H}} \Delta t), \quad (102)$$

with $\mathbf{H} = (0, 0, 0, 1)$, as above (see [equation \(52\)](#)).

Using [equations \(101\)](#) and [\(102\)](#) as the model for feedback observations, we aim to infer a distribution over the unobserved states, $\Phi_{i,t}$, conditioned on the history of local data seen by the synapse. The data is denoted

$$\mathcal{D}_{i,t} \equiv \{x_{i,0:t}, w_{i,0:t}, f_{S,0:t}\}. \quad (103)$$

The distribution is modeled as a Gaussian with conditional mean and covariance

$$\hat{\Phi}_{i,t} \equiv \mathbb{E}(\Phi_{i,t}|\mathcal{D}_{i,t-\Delta t}) \quad (104a)$$

$$\mathbf{P}_{i,t} \equiv \text{Cov}(\Phi_{i,t}|\mathcal{D}_{i,t-\Delta t}). \quad (104b)$$

Recursive Bayesian filter

We use standard Bayesian recursion to derive explicit expressions for the mean and covariance of Φ_i . Our approach is similar to that of [Eden et al. \(2004\)](#) and [Pfister et al. \(2010\)](#).

At each time step, we first update the distribution over $\Phi_{i,t}$ using observation $f_{S,t}$ via Bayes theorem, and then predict $\Phi_{i,t+\Delta t}$ from the model dynamics,

$$P(\Phi_{i,t}|\mathcal{D}_{i,t}) \propto P(f_{S,t}|\Phi_{i,t})P(\Phi_{i,t}|\mathcal{D}_{i,t-\Delta t}) \quad (105a)$$

$$P(\Phi_{i,t+\Delta t}|\mathcal{D}_{i,t}) = \int P(\Phi_{i,t+\Delta t}|\Phi_{i,t}, x_{i,t}, w_{i,t})P(\Phi_{i,t}|\mathcal{D}_{i,t}) d\Phi_{i,t}. \quad (105b)$$

The first equation holds for our model because $f_{S,t}$ only depends on $\Phi_{i,t}$, and $\Phi_{i,t}$ only depends on the input and weight from previous time steps. We use assumed density filtering for the update step, approximating the left-hand side of [equation \(105a\)](#) as Gaussian by matching the mean and covariance of the right-hand side.

When $f_{S,t} = 1$, the approximation is exact, giving a Gaussian

$$P(\Phi_{i,t}|\mathcal{D}_{i,t}, f_{S,t} = 1) \propto \Delta t \gamma_0 e^{\rho \Phi_{i,t} \cdot \mathbf{H}} P(\Phi_{i,t}|\mathcal{D}_{i,t-\Delta t}). \quad (106)$$

The normalizer is given by

$$\begin{aligned} q_{i,t} &= \Delta t \gamma_0 \int e^{\rho \Phi_{i,t} \cdot \mathbf{H}} P(\Phi_{i,t}|\mathcal{D}_{i,t-\Delta t}) d\Phi_{i,t} \\ &= \Delta t \gamma_0 e^{\rho \hat{\Phi}_{i,t} \cdot \mathbf{H} + \frac{1}{2} \rho^2 \mathbf{H} \cdot \mathbf{P}_{i,t} \cdot \mathbf{H}}. \end{aligned} \quad (107)$$

After some straightforward algebra, the updated mean and covariance are found to be

$$\mathbb{E}(\Phi_{i,t}|\mathcal{D}_{i,t}, f_{S,t} = 1) = \hat{\Phi}_{i,t} + \rho \mathbf{P}_{i,t} \cdot \mathbf{H} \quad (108a)$$

$$\text{Cov}(\Phi_{i,t}|\mathcal{D}_{i,t}, f_{S,t} = 1) = \mathbf{P}_{i,t}. \quad (108b)$$

When $f_{S,t} = 0$,

$$P(\Phi_{i,t}|\mathcal{D}_{i,t}, f_{S,t} = 0) = \frac{1}{1 - q_{i,t}} (1 - \Delta t \gamma_0 e^{\rho \Phi_{i,t} \cdot \mathbf{H}}) P(\Phi_{i,t}|\mathcal{D}_{i,t-\Delta t}), \quad (109)$$

which leads to, again after some straightforward algebra,

$$\mathbb{E}(\Phi_{i,t}|\mathcal{D}_{i,t}, f_{S,t} = 0) = \hat{\Phi}_{i,t} - \rho \frac{q_{i,t}}{1 - q_{i,t}} \mathbf{P}_{i,t} \cdot \mathbf{H} \quad (110a)$$

$$\text{Cov}(\Phi_{i,t}|\mathcal{D}_{i,t}, f_{S,t} = 0) = \mathbf{P}_{i,t} - \rho^2 \frac{q_{i,t}}{(1 - q_{i,t})^2} \mathbf{P}_{i,t} \cdot \mathbf{H} \mathbf{H} \cdot \mathbf{P}_{i,t}. \quad (110b)$$

Putting both cases together, we arrive at

$$\mathbb{E}(\Phi_{i,t}|\mathcal{D}_{i,t}) = \hat{\Phi}_{i,t} + \rho [f_{S,t} - (1 - f_{S,t}) \frac{q_{i,t}}{1 - q_{i,t}}] \mathbf{P}_{i,t} \cdot \mathbf{H} \quad (111a)$$

$$\text{Cov}(\Phi_{i,t}|\mathcal{D}_{i,t}) = \mathbf{P}_{i,t} - \rho^2 (1 - f_{S,t}) \frac{q_{i,t}}{(1 - q_{i,t})^2} \mathbf{P}_{i,t} \cdot \mathbf{H} \mathbf{H} \cdot \mathbf{P}_{i,t}, \quad (111b)$$

which characterizes the posterior distribution on the left-hand side of the update step in [equation \(105a\)](#) [↗](#).

Next, having completed the update step, we perform the prediction step in [equation \(105b\)](#) [↗](#) by evaluating the integral. The second factor in the integral is Gaussian by assumption, following from the previous update. The first factor, describing the state transition probability, is also Gaussian, following from the dynamics of [equation \(53a\)](#) [↗](#). This means the left-hand side of [equation \(105b\)](#) [↗](#) is Gaussian with

$$\mathbb{E}(\Phi_{i,t+\Delta t}|\mathcal{D}_{i,t}) = \tilde{\mathbf{C}}_{i,t} \cdot \mathbb{E}(\Phi_{i,t}|\mathcal{D}_{i,t}) + \tilde{\mathbf{B}} \cdot \mathbf{v}_{i,t} \quad (112a)$$

$$\text{Cov}(\Phi_{i,t+\Delta t}|\mathcal{D}_{i,t}) = \tilde{\mathbf{C}}_{i,t} \cdot \text{Cov}(\Phi_{i,t}|\mathcal{D}_{i,t}) \cdot \tilde{\mathbf{C}}_{i,t}^T + \tilde{\mathbf{A}}'. \quad (112b)$$

Substituting [equations \(111a\)](#) [↗](#) and [\(111b\)](#) [↗](#) into these expressions, and using the definitions in [equations \(104a\)](#) [↗](#) and [\(104b\)](#) [↗](#), gives a discrete-time estimator

$$\hat{\Phi}_{i,t+\Delta t} = \tilde{\mathbf{C}}_{i,t} \cdot \hat{\Phi}_{i,t} + \rho [f_{S,t} - (1 - f_{S,t}) \frac{q_{i,t}}{1 - q_{i,t}}] \tilde{\mathbf{C}}_{i,t} \cdot \mathbf{P}_{i,t} \cdot \mathbf{H} + \tilde{\mathbf{B}} \cdot \mathbf{v}_{i,t} \quad (113a)$$

$$\mathbf{P}_{i,t+\Delta t} = \tilde{\mathbf{C}}_{i,t} \cdot \mathbf{P}_{i,t} \cdot \tilde{\mathbf{C}}_{i,t}^T - \rho^2 (1 - f_{S,t}) \frac{q_{i,t}}{(1 - q_{i,t})^2} \tilde{\mathbf{C}}_{i,t} \cdot \mathbf{P}_{i,t} \cdot \mathbf{H} \mathbf{H} \cdot \mathbf{P}_{i,t} \cdot \tilde{\mathbf{C}}_{i,t}^T + \tilde{\mathbf{A}}'. \quad (113b)$$

Expanding to first order in Δt , using equations (100a)–(100d), and taking $\Delta t \rightarrow 0$ gives the continuous-time version:

$$\dot{\hat{\Phi}}_i = \mathbf{C}_i \cdot \hat{\Phi}_i + \mathbf{B} \cdot \mathbf{v}_i + \rho(f_S - \hat{\gamma}_i) \mathbf{P}_i \cdot \mathbf{H} \quad (114a)$$

$$\dot{\mathbf{P}}_i = \mathbf{C}_i \cdot \mathbf{P}_i + \mathbf{P}_i \cdot \mathbf{C}_i^T - \rho^2 \hat{\gamma}_i \mathbf{P}_i \cdot \mathbf{H} \mathbf{H} \cdot \mathbf{P}_i + \mathbf{\Lambda}^T. \quad (114b)$$

Here, the discrete-time feedback $f_{S,t}$ has converged to a sum of delta functions f_S , and

$$\hat{\gamma}_i = \gamma_0 e^{\rho \hat{\Phi}_i \cdot \mathbf{H} + \frac{1}{2} \rho^2 \mathbf{H} \cdot \mathbf{P}_i \cdot \mathbf{H}} \quad (115)$$

is the estimated feedback rate.

equations (114a) and (114b) correspond to equations (58a) and (60a) in the continuous feedback case, with $\rho \mathbf{P}_i \cdot \mathbf{H}$ playing the role of feedback gain \mathbf{K}_i , and $f_S - \hat{\gamma}_i$ playing the role of the prediction error. Whereas the optimal gain scales with the precision of feedback observations in the continuous case, here it scales with the steepness of the feedback rate function.

As above, the feedback gain can be approximated by partitioning the covariance matrix \mathbf{P}_i (see equation (66)), and ignoring the minor contribution of x_i to the lower-right block,

$$\dot{\sigma}_i^2 = -\frac{2}{\tau_w} (\sigma_i^2 - \sigma_w^2) - \rho^2 \hat{\gamma}_i (\mathbf{z}_i \cdot \mathbf{h})^2 \quad (116a)$$

$$\dot{\mathbf{z}}_i = \left(\mathbf{A} - \frac{1}{\tau_w} \mathbf{I} \right) \cdot \mathbf{z}_i - \rho^2 \hat{\gamma}_i (\mathbf{z}_i \cdot \mathbf{h}) \mathbf{\Sigma} \cdot \mathbf{h} - \begin{bmatrix} \frac{1}{\tau_I} \sigma_i^2 x_i \\ 0 \\ 0 \end{bmatrix} \quad (116b)$$

$$\dot{\mathbf{\Sigma}}_i = \mathbf{A} \cdot \mathbf{\Sigma}_i + \mathbf{\Sigma}_i \cdot \mathbf{A}^T - \rho^2 \hat{\gamma}_i \mathbf{\Sigma}_i \cdot \mathbf{h} \mathbf{h} \cdot \mathbf{\Sigma}_i + \mathbf{\Lambda}. \quad (116c)$$

A key difference between these equations and equation (69) is that in the latter $\mathbf{\Sigma}_i$ relaxes to a steady state, whereas here it depends dynamically on the estimate of the feedback rate $\hat{\gamma}_i$. Intuitively, the higher the feedback rate, the more uncertainty about the error is reduced.

The rest of the derivation proceeds identically to the continuous case. Unpacking equation (114a) into the original notation, and setting $\bar{w}_i = \hat{w}_i^*$, yields an error estimate, $\hat{\delta}$, covariance, $\mathbf{\Sigma}$, and feedback rate estimate, $\hat{\gamma}$, that are all independent of the synapse index. The control gain is computed identically to that above (equations (61) and (65)), yielding a control variable $\hat{u} = -\mathcal{L} \cdot \hat{\delta}$.

Putting everything together, the plasticity rule for spiking feedback is

$$\dot{\hat{\gamma}} = \gamma_0 e^{\rho \hat{\delta} \cdot \mathbf{h} + \frac{1}{2} \rho^2 \mathbf{h} \cdot \mathbf{\Sigma} \cdot \mathbf{h}} \quad (117a)$$

$$\dot{\hat{\delta}} = \mathbf{A} \cdot \hat{\delta} + \begin{bmatrix} \frac{1}{\tau_I} \hat{u} \\ 0 \\ 0 \end{bmatrix} + \rho \mathbf{\Sigma} \cdot \mathbf{h} (f_S - \hat{\gamma}) \quad (117b)$$

$$\dot{\hat{w}}_i^* = -\frac{1}{\tau_w} (\hat{w}_i^* - \mu_w) + \rho \mathbf{z}_i \cdot \mathbf{h} (f_S - \hat{\gamma}) \quad (117c)$$

$$\dot{\sigma}_i^2 = -\frac{2}{\tau_w} (\sigma_i^2 - \sigma_w^2) - \rho^2 \hat{\gamma} (\mathbf{z}_i \cdot \mathbf{h})^2 \quad (117d)$$

$$\dot{\mathbf{z}}_i = \mathbf{A} \cdot \mathbf{z}_i - \rho^2 \hat{\gamma} \mathbf{\Sigma} \cdot \mathbf{h} \mathbf{h} \cdot \mathbf{z}_i - \begin{bmatrix} \frac{1}{\tau_I} \sigma_i^2 x_i \\ 0 \\ 0 \end{bmatrix} \quad (117e)$$

$$\dot{\mathbf{\Sigma}} = \mathbf{A} \cdot \mathbf{\Sigma} + \mathbf{\Sigma} \cdot \mathbf{A}^T - \rho^2 \hat{\gamma} \mathbf{\Sigma} \cdot \mathbf{h} \mathbf{h} \cdot \mathbf{\Sigma} + \mathbf{\Lambda} \quad (117f)$$

with weights set via [equation \(77\)](#) as before.

Feedback delays

In the brain, error feedback cannot be provided instantaneously to a neuron due to communication and sensory processing delays, typically on the order of $\sim 10 - 100$ ms. We model this constraint by introducing a lag of time τ into the feedback signal, modifying [equation \(10\)](#) to become

$$f(t) = y(t - \tau) - y^*(t - \tau) + \sigma_f \xi_f. \quad (118)$$

Similarly, for the spiking feedback model, [equation \(17\)](#) becomes

$$\gamma(t) = \gamma_0 e^{\rho(y(t-\tau) - y^*(t-\tau))}. \quad (119)$$

For the slow weights, the learning rules can be adapted by simply feeding lagged copies of the input and weight parameters into the governing equations, such that the Bayesian filter estimate reflects the unobserved states as they were at time $t - \tau$. Working through the details, the only change required to the learning rules is to replace $x_i(t)$ with $x_i(t - \tau)$ in [equation \(76d\)](#), and similarly in the spiking feedback case. Such a lagged copy of the input could be maintained at the synapse via simple chemical signaling cascades, as shown by Jayabal et al. (2022). As long as the assumed target-weight drift is slow relative to the feedback delay, $\tau_w \gg \tau$, this simple modification of the learning rule maintains consistent performance in the presence of delays.

For the fast weights, we employ the Smith predictor technique to build a controller that accounts for the delay (Smith 1959). Conceptually, we need to compute a control signal at time t that will cancel the current error, but we only have an estimate from feedback observations of the error as it was at time $t - \tau$. We therefore use the known model dynamics and recent history of applied control signals to predict the current error by propagating the estimate δ forward in time. We use this to construct the controller

$$\hat{u} = -\mathcal{L} \cdot \hat{\delta}_\tau^{\text{pred}} \quad (120)$$

and thus the fast-weight rule

$$\delta w_i = -\frac{1}{N\nu} \mathcal{L} \cdot \hat{\delta}_\tau^{\text{pred}}, \quad (121)$$

where the control gain \mathcal{L} is the same as that above.

Writing $\hat{\mathbf{u}}(t) = (\hat{u}(t), 0, 0)$, the predicted error can be computed via

$$\begin{aligned} \hat{\delta}_\tau^{\text{pred}} &= e^{\mathbf{A}\tau} \cdot \hat{\delta} + \int_{t-\tau}^t e^{\mathbf{A}(t-s)} \cdot \frac{1}{\tau I} \hat{\mathbf{u}}(s) ds \\ &= e^{\mathbf{A}\tau} \cdot \hat{\delta} - \int_{t-\tau}^t e^{\mathbf{A}(t-s)} \cdot \begin{bmatrix} \frac{1}{\tau I} \\ 0 \\ 0 \end{bmatrix} \mathcal{L} \cdot \hat{\delta}_\tau^{\text{pred}}(s) ds \end{aligned} \quad (122)$$

The first term propagates $\hat{\delta}$ forward in time via a matrix exponential that encodes the deterministic dynamics of the error vector. The second term is a convolution that accounts for the influence of past controls – propagating the corrections applied by the controller at each time s (whose effects have not yet been observed) to the current time t .

The convolution can be simplified by considering an online implementation. Denoting the integral by $U(t)$, differentiation gives

$$\dot{U}(t) = \mathbf{A} \cdot \mathbf{U}(t) + \frac{1}{\tau_I} (\hat{\mathbf{u}}(t) - e^{\mathbf{A}\tau} \cdot \hat{\mathbf{u}}(t - \tau)), \quad (123)$$

which only requires knowing the most recently applied control and a single lagged variable from the past.

Multiple neurons

In **Fig. 3C** and **Fig. 4**, we simulated a small feedforward network of M neurons. In this model, each of the neurons has identical dynamics, as described by **equations (7a)** and **(7b)**, but receives independent synaptic input and noise. Noise variances were scaled by a factor $1/M$ relative to the single neuron simulations, for ease of comparison (see **Table 1** for parameters). A common downstream output is driven by the sum over firing rates, r_m , described by **equation (19)**. A spiking feedback signal, $f_{S,m}$, is generated independently for each neuron using the Poisson rate function in **equation (17)**.

In the plasticity rule, the current, rate and synaptic noise terms in **equation (71)** are scaled by M to reflect the contribution of all neurons to the error. The fast weight rule (**equation (46)**) also needs to be modified to account for the fact that the feedback signals, and therefore error estimates, differ across neurons. Ignoring locality constraints, the optimal strategy would be to combine the error estimates computed by each of the M neurons, $\hat{\delta}_m$, and their uncertainties, Σ_m , as a precision-weighted average,

$$\hat{\delta}_M = \left(\sum_{m'} \Sigma_{m'}^{-1} \right)^{-1} \cdot \sum_m \Sigma_m^{-1} \cdot \hat{\delta}_m, \quad (124)$$

and use that to construct a single global control variable, $\hat{u}_M = -\mathcal{L} \cdot \hat{\delta}_M$. We use a local approximation of this strategy, by replacing the precision weighting in **equation (124)** with a scaled identity matrix,

$$\begin{aligned} \hat{u}_M &= -\mathcal{L} \cdot \hat{\delta}_M \\ &\approx -\mathcal{L} \cdot \sum_m \frac{1}{1 + \beta(M-1)} \mathbf{I} \cdot \hat{\delta}_m \\ &= \frac{1}{1 + \beta(M-1)} \sum_m \hat{u}_m. \end{aligned} \quad (125)$$

In **equation (125)**, $\hat{u}_m = -\mathcal{L} \cdot \hat{\delta}_m$ is the control variable computed locally by neuron m , and $\beta \in [0, 1]$ is a free parameter (fitted in pilot simulations to minimize task output error). When feedback rates are very low, a small number of neurons would dominate the sum in **equation (124)** at any given time (error estimates are most precise immediately following a feedback spike, but relax to small, uncertain baseline values soon afterwards), so β should be small. In the limit of very high feedback rates, all terms would be weighted equally, so $\beta \approx 1$. Intuitively, β interpolates between sparse and dense feedback regimes. Using this approximation leads to scaled fast weights

$$\delta w_{m,i} = -\frac{1}{(1 + \beta(M-1))N\nu} \mathcal{L} \cdot \hat{\delta}_m. \quad (126)$$

Simulation details

Online linear regression

For the simulations in [Fig. 1](#), the model neuron was driven with sinusoidal inputs and required to learn weights that would produce a time-varying, periodic target output. We used $N = 20$ synapses and a period $T_{\text{period}} = 1$ s.

Input rates were defined as

$$\nu_i(t) = \begin{cases} 1 + \cos(\frac{2\pi i t}{T_{\text{period}}}), & i = 1, \dots, \frac{N}{2} \\ 1 + \sin(\frac{2\pi(i - \frac{N}{2})t}{T_{\text{period}}}), & i = \frac{N}{2} + 1, \dots, N \end{cases} \quad (127)$$

The target output was generated as

$$y^*(t) = \sum_i w_i^* \nu_i(t), \quad (128)$$

With $w_i^* \sim \mathcal{N}(0, \frac{1}{\sqrt{N}})$. Learning rules were as described by [equations \(2\)](#), [\(4\)](#) and [\(6\)](#), with learning rates $\eta, \bar{\eta} = 0.001$.

Teacher-student task

In [Figs. 2](#) and [3](#), simulations were initialized by independently drawing target weights $w_i^* \sim \mathcal{N}(\mu_w, \sigma_w^2)$, input rates $\nu_i \sim \text{uniform}(0, v_{\text{max}})$, slow weights $\bar{w}_i \sim \mathcal{N}(\mu_w, \sigma_w^2)$, and setting fast weights $\delta w_i = 0$. The weight uncertainties σ_i^2 were initialized with value σ_w^2 . Input spikes were generated at each time step by drawing binary variables $x_i \sim \text{Bernoulli}(\nu_i \Delta t)$. Feedback spikes were generated similarly in the spiking-feedback case, using the rate function in [equation \(17\)](#), and also enforcing a hard maximum of 200 spikes/s for stability. Target weights drifted slowly about the mean μ_w , as described by the Euler discretization of [equation \(41\)](#). Actual weights were updated according to the Euler discretization of [equation \(76\)](#) in the continuous case, [equation \(117\)](#) in the spiking-feedback case, and analogously for the classical rules. Performance was quantified by computing the root-mean-squared error between output and target output over all time points in the simulation, thus measuring both learning speed and steady-state error.

Cerebellar learning task

The cerebellar learning task in [Fig. 4](#) requires mapping time-varying patterns of synaptic input to associated target outputs. Unlike the teacher-student task, the target outputs are generated as random wavelets, rather than through explicit target weights.

Input patterns of duration $T_{\text{pattern}} = 1$ s were constructed in a similar manner to [Bicknell and Häusser \(2021\)](#). A synapse was selected to be active on a given pattern with probability $p_{\text{active}} = 0.5$. If active, its time-dependent presynaptic firing rate in that pattern was defined by drawing a peak time $t_{\text{peak},i} \sim \text{Uniform}(0, T_{\text{pattern}})$, maximum amplitude $\nu_{\text{max},i} \sim \text{Uniform}(90, 110)$ (spikes/s) and centering a Gaussian bump of activity within the pattern interval,

$$\nu_i(t) = \nu_{\text{max},i} e^{-\frac{(t - t_{\text{peak},i})^2}{2\sigma_v^2}}, \quad (129)$$

with width parameter $\sigma_v = 0.1$ s. Input spikes were generated stochastically each pattern presentation using this rate function.

For the target outputs, we generated random trajectories that drift slowly over time and could be stitched together in random order to yield a single continuous target (analogous to composing different sequences of motor actions). This was implemented using truncated Fourier series with

random coefficients that drift via Ornstein-Uhlenbeck processes. Different trajectories are forced to agree at the endpoints by using a common constant offset and multiplying the time-dependent parts by a bump function. Specifically, the drifting target output for pattern p was defined via

$$y_p^*(t) = y_0 + e^{-\frac{(2t-T_{\text{pattern}})^2}{T_{\text{pattern}}^2 - (2t-T_{\text{pattern}})^2}} \psi_p(t) \quad (130a)$$

$$\psi_p(t) = \sum_{n=1}^{N_{\text{modes}}} a_{p,n} \cos\left(\frac{2\pi nt}{T_{\text{pattern}}}\right) + b_{p,n} \sin\left(\frac{2\pi nt}{T_{\text{pattern}}}\right) \quad (130b)$$

$$\tau_w \dot{a}_{p,n} = -a_{p,n} + \sqrt{\tau_w / N_{\text{modes}}} \xi_{a_{p,n}} \quad (130c)$$

$$\tau_w \dot{b}_{p,n} = -b_{p,n} + \sqrt{\tau_w / N_{\text{modes}}} \xi_{b_{p,n}}, \quad (130d)$$

with initial coefficients $a_{p,n}, b_{p,n} \sim \mathcal{N}(0, \frac{1}{2N_{\text{modes}}})$. We used parameters $N_{\text{modes}} = 3$, which sets the maximum temporal frequency, and $\tau_w = 10^4$ s, equal to the weight drift parameter in the spiking-feedback teacher-student task.

Plasticity protocol

For the simulated plasticity experiments in [Fig. 5](#), we used models that had been trained on the cerebellar learning task. We applied the protocol to 100 synapses using the classical and Bayesian learning rules, initializing with the trained weights and uncertainties. The classical learning rate, η , and control cost, λ_w , were set at the values that minimized output error during the task. For most simulations, the protocol consisted of repeated pairings of parallel fiber input bursts (5 spikes at 10 ms intervals) and single climbing fiber spikes at a given delay. The numbers of spikes were varied in a subset of simulations ([Fig. S3c,d](#)). The pairing was repeated 50 times at 500 ms intervals. Plasticity effects were computed at the end of the protocol as the change in slow-weights (or classical weights) from their initial trained values.

Supplementary Figures

Figure S1

Approximation of the Bayesian learning rule.

Comparison of the full Bayesian rule, given by [equation \(76\)](#), and the approximation presented for ease of interpretation in [equations \(13\)](#) and [\(16\)](#), in terms of output error (**a**) and weight error (**b**). The approximation is valid whenever the output time constant dominates the model dynamics, $\tau_r, \tau_I \ll \tau_Y$. Here we use parameters $\tau_I = 1$ ms, $\tau_r = 10$ ms, and $\tau_Y = 100$ ms.

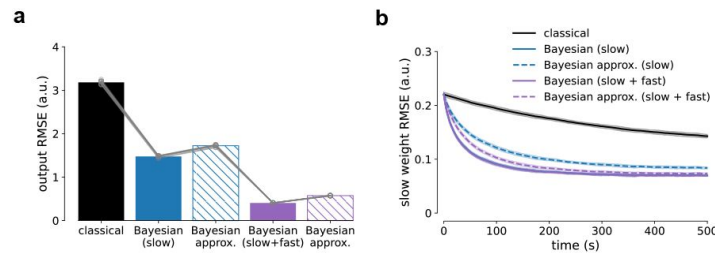
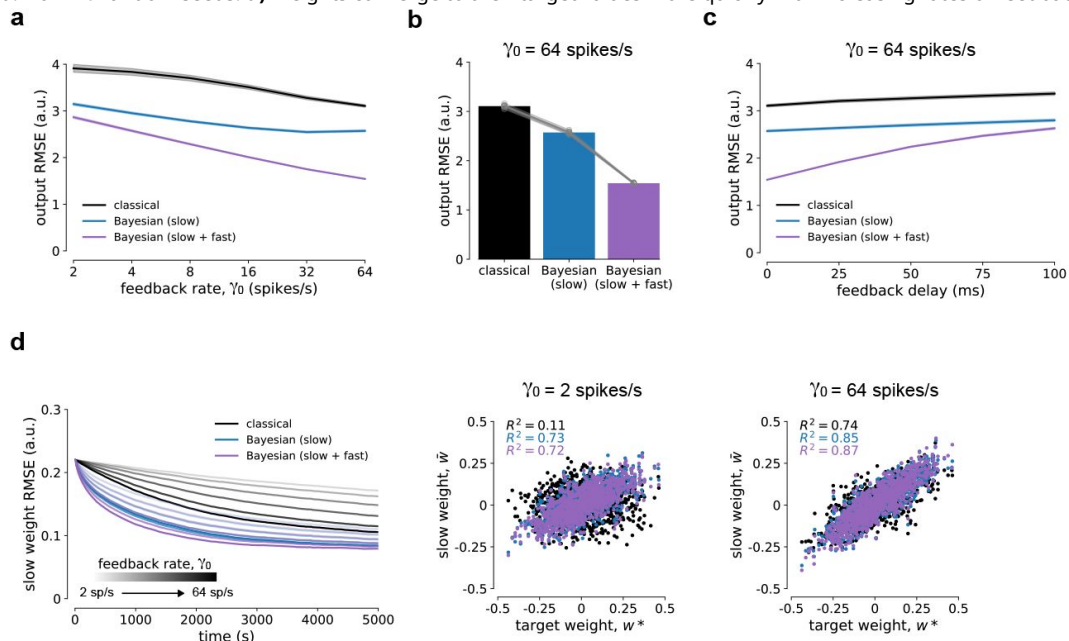


Figure S2

Simulations with spiking feedback.

Results from the spiking feedback model are consistent with the continuous feedback model ([Fig. 2](#)), although learning takes longer and performance depends strongly on the spontaneous feedback rate γ_0 . **a**) Performance as a function of feedback rate (replotted from [Fig. 3](#) for context). Shaded areas are s.d. from 10 random seeds. **b**) Detailed comparison between learning rules with $\gamma_0 = 64$ spikes/s. **c**) Control remains effective in the presence of feedback delays. Shaded areas are s.d. from 10 random seeds. **d**) Weights converge to their target values more quickly with increasing rates of feedback.



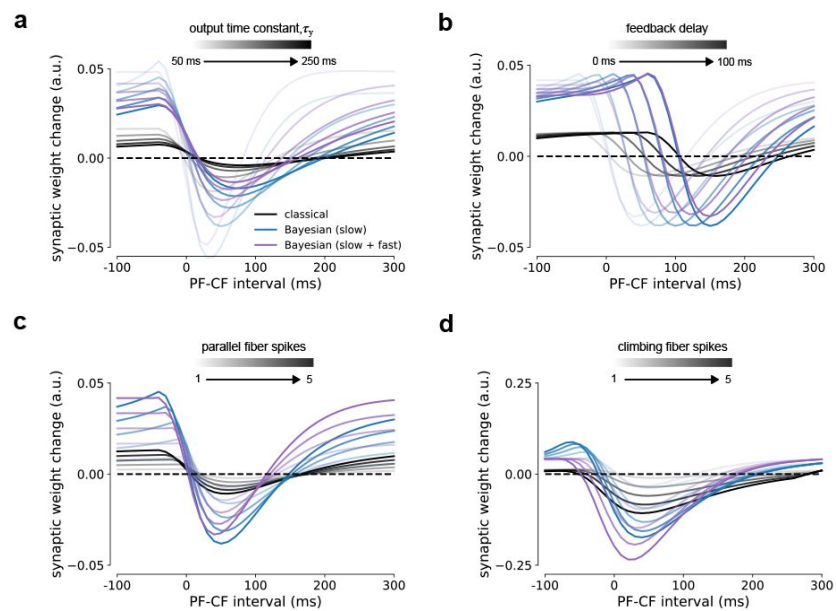


Figure S3

Parameter dependence of simulated plasticity experiments.


The LTP/LTD curve produced using the protocol in [Fig. 5c](#) depends systematically on the parameters of the model and protocol. **a)** Tuning to the PF-CF interval becomes broader as the output time constant τ_y increases. **b)** Peaks in the LTP and LTD lobes of the curve are shifted in proportion to the feedback delay. **c)** The magnitude of plasticity increases with the number of spikes in the parallel fiber burst. **d)** LTD is amplified and shifted to earlier PF-CF intervals with increasing numbers of climbing fiber spikes.

Acknowledgements

This work was supported by the Gatsby Charitable Foundation and the Wellcome Trust (110114/Z/15/Z). The authors acknowledge the use of the UCL Myriad High Performance Computing Facility (Myriad@UCL), and associated support services, in the completion of this work.

Additional information

Code availability

Simulation code is available at <https://github.com/babicknell/SynControl> 

References

- Abbott L., Regehr W.G. (2004) **Synaptic computation** *Nature* **431**:796–803
- Aitchison L., Jegminat J., Menendez J.A., Pfister J.-P., Pouget A., Latham P.E. (2021) **Synaptic plasticity as Bayesian inference** *Nat. Neurosci* **24**:565–571
- Alemi A., Machens C., Deneve S., Slotine J.-J. (2018) **Learning nonlinear dynamics in efficient, balanced spiking networks using local plasticity rules** *Proc. AAAI Conf. Artif. Intell* **32**
- Apps R., et al. (2018) **Cerebellar modules and their role as operational cerebellar processing units** *The Cerebellum* **17**:654–682
- Armstrong D., Rawson J. (1979) **Activity patterns of cerebellar cortical neurones and climbing fibre afferents in the awake cat** *J. Physiol* **289**:425–448
- Barmack N.H., Yakhnitsa V. (2003) **Cerebellar climbing fibers modulate simple spikes in Purkinje cells** *J. Neurosci* **23**:7904–7916
- Bell C.C., Grimm R. (1969) **Discharge properties of Purkinje cells recorded on single and double microelectrodes** *J. Neurophysiol* **32**:1044–1055
- Benna M.K., Fusi S. (2016) **Computational principles of synaptic memory consolidation** *Nat. Neurosci* **19**:1697–1706
- Bhalla U.S. (2014) **Molecular computation in neurons: a modeling perspective** *Curr. Opin. Neurobiol* **25**:31–37
- Bicknell B.A., Häusser M. (2021) **A synaptic learning rule for exploiting nonlinear dendritic computation** *Neuron* **109**:4001–4017
- Bloedel J.R., Roberts W.J. (1971) **Action of climbing fibers in cerebellar cortex of the cat** *J. Neurophysiol* **34**:17–31
- Blundell C., Cornebise J., Kavukcuoglu K., Wierstra D. (2015) **Weight uncertainty in neural network** In: *Proc. Int. Conf. Mach. Learn* pp. 1613–1622
- Bourdoukan R., Denève S. (2015) **Enforcing balance allows local supervised learning in spiking recurrent networks** *Adv. Neural Inf. Process. Syst* **28**
- Bouvier G., et al. (2018) **Cerebellar learning using perturbations** *eLife* **7**:e31599
- Bredenberg C., Savin C. (2023) **Desiderata for normative models of synaptic plasticity** *arXiv*
- Brunel N., Hakim V., Isope P., Nadal J.-P., Barbour B. (2004) **Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cell** *Neuron* **43**:745–757
- Buntine W.L., Weigend A.S. (1991) **Bayesian backpropagation** *Complex Syst* **5**:603–643

- Cheun J., Yeh H. (1992) **Modulation of GABAA receptor-activated current by norepinephrine in cerebellar Purkinje cells** *Neuroscience* **51**:951–960
- Coesmans M., Weber J.T., De Zeeuw C.I., Hansel C. (2004) **Bidirectional parallel fiber plasticity in the cerebellum under climbing fiber control** *Neuron* **44**:691–700
- Crassidis J., Junkins J. (2011) **Optimal estimation of dynamic systems** Chapman & Hall
- Davie J.T., Clark B.A., Häusser M. (2008) **The origin of the complex spike in cerebellar Purkinje cells** *J. Neurosci* **28**:7599–7609
- Dayan P., Kakade S. (2000) **Explaining away in weight space** *Adv. Neural Inf. Process. Syst* **13**
- De Pittà M., Brunel N., Volterra A. (2016) **Astrocytes: Orchestrating synaptic plasticity?** *Neuroscience* **323**
- De Schutter E., Steuber V. (2009) **Patterns and pauses in Purkinje cell simple spike trains: experiments, modeling and theory** *Neuroscience* **162**:816–826
- Denève S., Alemi A., Bourdoukan R. (2017) **The brain as an efficient and robust adaptive learner** *Neuron* **94**:969–977
- Drugowitsch J., Mendonça A.G., Mainen Z.F., Pouget A. (2019) **Learning optimal decisions with confidence** *Proc. Natl. Acad. Sci. U.S.A* **116**:24872–24880
- Eden U.T., Frank L.M., Barbieri R., Solo V., Brown E.N. (2004) **Dynamic analysis of neural encoding by point process adaptive filtering** *Neural Comput* **16**:971–998
- Fan L.Z., et al. (2023) **All-optical physiology resolves a synaptic basis for behavioral timescale plasticity** *Cell* **186**:543–559
- Feng J., Cai X., Zhao J., Yan Z. (2001) **Serotonin receptors modulate GABAA receptor channels through activation of anchored protein kinase C in prefrontal cortical neurons** *J. Neurosci* **21**:6502–6511
- Feulner B., Perich M.G., Miller L.E., Clopath C., Gallego J.A. (2022) **Feedback-based motor control can guide plasticity and drive rapid learning** *bioRxiv*
- Fişek M., Herrmann D., Egea-Weiss A., Cloves M., Bauer L., Lee T.-Y., Russell L.E., Häusser M. (2023) **Cortico-cortical feedback engages active dendrites in visual cortex** *Nature* **617**:769–776
- Fourcaud N., Brunel N. (2002) **Dynamics of the firing probability of noisy integrate-and-fire neurons** *Neural Comput* **14**:2057–2110
- Friedrich J., Golkar S., Farashahi S., Genkin A., Sengupta A., Chklovskii D. (2021) **Neural optimal feedback control with local learning rules** *Adv. Neural Inf. Process. Syst* **34**:16358–16370
- Gil Z., Connors B.W., Amitai Y. (1997) **Differential regulation of neocortical synapses by neuromodulators and activity** *Neuron* **19**:679–686
- Gilmer J.I., Farries M.A., Kilpatrick Z., Delis I., Cohen J.D., Person A.L. (2023) **An emergent temporal basis set robustly supports cerebellar time-series learning** *J. Neurophysiol* **129**:159–176

- Han K.-S., Chen C.H., Khan M.M., Guo C., Regehr W.G. (2020) **Climbing fiber synapses rapidly and transiently inhibit neighboring Purkinje cells via ephaptic coupling** *Nat. Neurosci* **23**:1399–1409
- Hernández-Lobato J.M., Adams R. (2015) **Probabilistic backpropagation for scalable learning of bayesian neural networks** In: *Proc. Intl. Conf. Mach. Learn* pp. 1861–1869
- Higley M.J., Sabatini B.L. (2010) **Competitive regulation of synaptic Ca²⁺ influx by D2 dopamine and A2A adenosine receptors** *Nat. Neurosci* **13**:958–966
- Hiratani N., Latham P.E. (2020) **Rapid Bayesian learning in the mammalian olfactory system** *Nat. Commun* **11**:3845
- Hull C. (2020) **Prediction signals in the cerebellum: beyond supervised motor learning** *eLife* **9**:e54073
- Hull C., Regehr W.G. (2022) **The cerebellar cortex** *Annu. Rev. Neurosci* **45**:151–175
- Ito M. (2001) **Cerebellar long-term depression: characterization, signal transduction, and functional roles** *Physiol. Rev* **81**:1143–1195
- Jayabal S., Bhasin B.J., Suvrathan A., DiSanto J., Goldman M.S., Raymond J.L. (2022) **Experience adaptively tunes the timing rules for associative plasticity** *bioRxiv*
- Jegminat J., Surace S.C., Pfister J.-P. (2022) **Learning as filtering: Implications for spike-based plasticity** *PLoS Comput. Biol* **18**:e1009721
- Kappel D., Habenschuss S., Legenstein R., Maass W. (2015) **Network plasticity as Bayesian inference** *PLoS Comput. Biol* **11**:e1004485
- Kirkpatrick J., et al. (2017) **Overcoming catastrophic forgetting in neural networks** *Proc. Natl. Acad. Sci* **114**:3521–3526
- Kostadinov D., Häusser M. (2022) **Reward signals in the cerebellum: origins, targets, and functional implications** *Neuron* **110**:1290–1303
- Kutschireiter A., Surace S.C., Pfister J.-P. (2020) **The Hitchhiker’s guide to nonlinear filtering** *J. Math. Psychol* **94**:102307
- Larkum M. (2013) **A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex** *Trends Neurosci* **3**:141–151
- Llinás R., Sugimori M. (1980) **Electrophysiological properties of in vitro Purkinje cell somata in mammalian cerebellar slices** *J. Physiol* **305**:171–195
- MacKay D.J. (1992) **A practical Bayesian framework for backpropagation networks** *Neural Comput* **4**:448–472
- Magee J.C., Grienberger C. (2020) **Synaptic plasticity forms and functions** *Annu. Rev. Neurosci* **43**:95–117
- Malkin J., O’Donnell C., Houghton C.J., Aitchison L. (2024) **Signatures of Bayesian inference emerge from energy-efficient synapses** *eLife* **12**, RP92595

- Marder E. (2012) **Neuromodulation of neuronal circuits: back to the future** *Neuron* **76**:1–11
- Mathews P.J., Lee K.H., Peng Z., Houser C.R., Otis T.S. (2012) **Effects of climbing fiber driven inhibition on Purkinje neuron spiking** *J. Neurosci* **32**:17988–17997
- Mathy A., Ho S.S., Davie J.T., Duguid I.C., Clark B.A., Häusser M. (2009) **Encoding of oscillations by axonal bursts in inferior olive neurons** *Neuron* **62**:388–399
- McGehee D.S., Heath M.J., Gelber S., Devay P., Role L.W. (1995) **Nicotine enhancement of fast excitatory synaptic transmission in CNS by presynaptic receptors** *Science* **269**:1692–1696
- McNamee D., Wolpert D.M. (2019) **Internal models in biological control** *Annu. Rev. Control Robot. Auton. Syst* **2**:339–364
- Medina J.F., Garcia K.S., Nores W.L., Taylor N.M., Mauk M.D. (2000) **Timing mechanisms in the cerebellum: testing predictions of a large-scale computer simulation** *J. Neurosci* **20**:5516–5525
- Meulemans A., Tristany Farinha M., García Ordóñez J., Vilimelis Aceituno P., Sacramento J., Grewe B.F. (2021) **Credit assignment in neural networks through deep feedback control** *Adv. Neural Inf. Process. Syst* **34**:4674–4687
- Meulemans A., Zucchet N., Kobayashi S., Von Oswald J., Sacramento J. (2022) **The least-control principle for local learning at equilibrium** *Adv. Neural Inf. Process. Syst* **35**:33603–33617
- Moore J., Genkin A., Tournoy M., Pughe-Sanford J., Steveninck R.R. van, Chklovskii D.B. (2024) **The neuron as a direct data-driven controller** *Proc. Natl. Acad. Sci. U.S.A* **121**:e2311893121
- Nadim F., Bucher D. (2014) **Neuromodulation of neurons and synapses** *Curr. Opin. Neurobiol* **29**:48–56
- Papouin T., Dunphy J., Tolman M., Foley J.C., Haydon P.G. (2017) **Astrocytic control of synaptic function** *Philos. Trans. R. Soc. B* **372**:20160154
- Payeur A., Guerguiev J., Zenke F., Richards B.A., Naud R. (2021) **Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits** *Nat. Neurosci* **24**:1010–1019
- Pearlmutter B.A. (1995) **Gradient calculations for dynamic recurrent neural networks: A survey** *IEEE Trans. Neural Netw* **6**:1212–1228
- Pemberton J., Chadderton P., Costa R.P. (2023) **Cerebellar-driven cortical dynamics enable task acquisition, switching and consolidation** *bioRxiv*
- Perea G., Navarrete M., Araque A. (2009) **Tripartite synapses: astrocytes process and control synaptic information** *Trends Neurosci* **32**:421–431
- Pfister J.P., Dayan P., Lengyel M. (2010) **Synapses with short-term plasticity are optimal estimators of presynaptic membrane potentials** *Nat. Neurosci* **13**:1271–1275
- Poirazi P., Brannon T., Mel B.W. (2003) **Pyramidal neuron as two-layer neural network** *Neuron* **37**:989–999
- Richards B.A., Lillicrap T.P. (2019) **Dendritic solutions to the credit assignment problem** *Curr. Opin. Neurobiol* **54**:28–36

- Richards B.A., Kording K.P. (2023) **The study of plasticity has always been about gradients** *J. Physiol* **601**:3141–3149
- Rossbroich J., Zenke F. (2023) **Dis-inhibitory neuronal circuits can control the sign of synaptic plasticity** *Adv. Neural Inf. Process. Syst* **37**
- Rotondo A.P., Raman D.V., O’Leary T. (2023) **How cerebellar architecture and dense activation patterns facilitate online learning in dynamic tasks** *bioRxiv*
- Safo P., Regehr W.G. (2008) **Timing dependence of the induction of cerebellar LTD** *Neuropharmacology* **54**:213–218
- Sato Y., Miura A., Fushiki H., Kawasaki T. (1992) **Short-term modulation of cerebellar Purkinje cell activity after spontaneous climbing fiber input** *J. Neurophysiol* **68**:2051–2062
- Silva N.T., Ramírez-Buriticá J., Pritchett D.L., Carey M.R. (2024) **Climbing fibers provide essential instructive signals for associative learning** *Nat. Neurosci* **27**:940–951
- Smith O.J. (1959) **A controller to overcome dead time** *ISA journal* **6**:28–33
- Steuber V., Mittmann W., Hoebeek F.E., Silver R.A., De Zeeuw C.I., Häusser M., De Schutter E. (2007) **Cerebellar LTD and pattern recognition by Purkinje cells** *Neuron* **54**:121–136
- Suvrathan A., Payne H.L., Raymond J.L. (2016) **Timing rules for synaptic plasticity matched to behavioral function** *Neuron* **92**:959–967
- Todorov E. (2004) **Optimality principles in sensorimotor control** *Nat. Neurosci* **7**:907–915
- Walter J.T., Khodakhah K. (2006) **The linear computational algorithm of cerebellar Purkinje cells** *J. Neurosci* **26**:12861–12872
- Wang S.S.-H., Denk W., Häusser M. (2000) **Coincidence detection in single dendritic spines mediated by calcium release** *Nat. Neurosci* **3**:1266–1273
- Widrow B., Hoff M.E. (1960) **Adaptive switching circuits** *IRE WESCON convention record* **4**:96–104
- Williams R.J., Zipser D. (1989) **A learning algorithm for continually running fully recurrent neural networks** *Neural Comput* **1**:270–280
- Wolpert D.M., Miall R.C., Kawato M. (1998) **Internal models in the cerebellum** *Trends Cog. Sci* **2**:338–347
- Zenke F., Agnes E.J., Gerstner W. (2015) **Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks** *Nat. Comms* **6**:6922
- Zenke F., Poole B., Ganguli S. (2017) **Continual learning through synaptic intelligence** In: *Int. Conf. Mach. Learn* pp. 3987–3995
- Zucker R.S., Regehr W.G. (2002) **Short-term synaptic plasticity** *Annu. Rev. Physiol* **64**:355–405

Author information

Brendan A Bicknell

Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom

ORCID iD: [0000-0002-2307-2109](https://orcid.org/0000-0002-2307-2109)

For correspondence: b.bicknell@ucl.ac.uk

Peter E Latham

Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom

ORCID iD: [0000-0001-8713-9328](https://orcid.org/0000-0001-8713-9328)

Editors

Reviewing Editor

Richard Naud

University of Ottawa, Ottawa, Canada

Senior Editor

Panayiota Poirazi

FORTH Institute of Molecular Biology and Biotechnology, Heraklion, Greece

Reviewer #1 (Public review):

Summary:

This paper proposes a new set of local synaptic plasticity rules that differs from classic rules in two regards: First, working under the assumption that signals coming into synapses change smoothly over time and thus have temporal correlations such that immediate activity is positively correlated with subsequent activity, it proposes both fast plasticity that immediately corrects errors as well as slower plasticity. Second, it derives these rules from optimal, Bayesian control theory principles that, even without the fast component of plasticity, are shown to provide more accurate performance than classic, non-Bayesian plasticity rules. As a proof of principle, it applies these to a simple cerebellar learning example that demonstrates how the proposed rules lead to learning performance that exceeds that achieved with classic cerebellar learning rules. The work also provides a potential normative explanation for post-climbing fiber spike pauses in Purkinje cell firing and proposes testable predictions for cerebellar experiments. Overall, I found the idea to be compelling and potentially broadly applicable across many systems. Further, I thought the work was a rare, very beautiful display of the application of optimal control theory to fundamental problems in neuroscience. My comments are all relatively minor and more expressions of interest than criticism.

Comments:

(1) The algorithm assumes, reasonably, that inputs are relatively smooth. However, I was wondering if this could make additional experimental predictions for the system being exceptionally noisy or otherwise behaving in signature ways if one were able to train a real biological network to match a rapidly changing or non-smooth function that does not align with the underlying assumptions of the model.

(2) The algorithm assumes that one can, to a good approximation, replace individual input rates by their across-synapse average. How sensitive is the learning to this assumption, as one might imagine scenarios where a neuron is sensitive to different inputs for different tasks or contexts so that a grand average might not be correct? Or, the functional number of inputs driving the output might be relatively low or otherwise highly fluctuating and less easily averaged over.

(3) On the cerebellar example, it is nice that the Bayesian example provides a narrower PF-CF interval for plasticity than the classical rules, but the window is not nearly as narrow as the Suvrathan et al. 2016 paper cited by the authors. Maybe this is something special about that system having well-defined, delayed feedback, but (optional) further comments or insights would be welcome if available.

(4) In the discussion, I appreciated the comparison with the Deneve work which has fast and slow feedback components. I was curious whether, although non-local, there were also conceptual similarities with FORCE learning in which there is also an immediate correction of activity through fast changing of synaptic weights, which then aids the slow long-term learning of synaptic weights.

<https://doi.org/10.7554/eLife.105043.1.sa1>

Reviewer #2 (Public review):

Summary:

Bricknell and Latham investigate the computational benefits of a dual-learning algorithm that combines a rapid, millisecond-scale weight adjustment mechanism with a conventional, slower gradient descent approach. A feedback error signal drives both mechanisms at the synaptic level.

Strengths:

Integrating these two learning timescales is intriguing and demonstrates improved performance compared to classical strategies, particularly in terms of robustness and generalization when learning new target signals.

Weaknesses:

The biological plausibility and justification for the proposed rapid learning mechanism require further elaboration and supporting mechanistic examples.

<https://doi.org/10.7554/eLife.105043.1.sa0>