Matching tutor to student: rules and mechanisms for efficient two-stage learning in neural circuits

Tiberiu Teşileanu[1,2], Bence Ölveczky[3], and Vijay Balasubramanian[1,2]

[1] *Initiative for the Theoretical Sciences, The Graduate Center, CUNY, New York, NY 10016*
[2] *David Rittenhouse Laboratories, University of Pennsylvania, Philadelphia, PA 19104*
[3] *Department of Organismic and Evolutionary Biology and Center for Brain Science, Harvard University, Cambridge, MA 02138*

August 26, 2016

Abstract

Existing models of birdsong learning assume that brain area LMAN introduces variability into song for trial-and-error learning. Recent data suggest that LMAN also encodes a corrective bias driving short-term improvements in song. These later consolidate in area RA, a motor cortex analogue downstream of LMAN. We develop a new model of such two-stage learning. Using a stochastic gradient descent approach, we derive how 'tutor' circuits should match plasticity mechanisms in 'student' circuits for efficient learning. We further describe a reinforcement learning framework with which the tutor can build its teaching signal. We show that mismatching the tutor signal and plasticity mechanism can impair or abolish learning. Applied to birdsong, our results predict the temporal structure of the corrective bias from LMAN given a plasticity rule in RA. Our framework can be applied predictively to other paired brain areas showing two-stage learning.

1 Introduction

Two-stage learning has been described in a variety of different contexts and neural circuits. During hippocampal memory consolidation, recent memories, that are dependent on the hippocampus, are transferred to the neocortex for long-term storage [12]. Similarly, the rat motor cortex provides essential input to sub-cortical circuits during skill learning, but then becomes dispensable for executing certain skills [21]. A paradigmatic example of two-stage learning occurs in songbirds learning their courtship songs [2, 36]. Zebra finches, commonly used in birdsong research, learn their song from their fathers as juveniles and keep the same song for life [19].

The birdsong circuit has been extensively studied; see Fig. 1A for an outline. Area HVC is a timebase circuit, with projection neurons that fire sparse spike bursts in precise synchrony with the song [14, 25, 31]. A population of neurons from HVC projects to the robust nucleus of the arcopallium (RA), a pre-motor area, which then projects to motor neurons controlling respiratory and syringeal muscles [33, 38, 24]. A second input to RA comes from the lateral magnocellular nucleus of the anterior nidopallium (LMAN). Unlike HVC and RA activity patterns, LMAN spiking is highly variable across different renditions of the song [28, 20]. LMAN is the output of the anterior forebrain pathway, a circuit involving the song-specialized basal ganglia [30].

Because of the variability in its activity patterns, it was thought that LMAN's role was simply to inject variability into the song. The resulting vocal experimentation would enable learning using reinforcement learning. For this reason, prior models tended to treat LMAN as a pure Poisson noise generator, and assume that a reward signal is received directly in RA [10]. More recent evidence, however, suggests that the reward signal reaches Area X, the song-specialized basal ganglia, rather than RA [23, 18]. Taken together with the fact that LMAN firing patterns are not uniformly random, but rather contain a corrective bias guiding plasticity in RA [2, 37], this suggests that we should rethink our models of song acquisition.

Here we build a general model of two-stage learning where one neural circuit "tutors" another. We develop a formalism for determining how the teaching signal should be adapted to a specific plasticity rule, to best instruct a student circuit to improve its performance at each learning step. We develop analytical results in a rate based model, and show through simulations that the general findings carry over to realistic spiking neurons. Applied to the vocal control circuit of songbirds, our model reproduces the observed changes in the spiking statistics of RA neurons as juvenile birds learn their song. Our framework also predicts how the LMAN signal should be adapted to properties of RA synapses. This prediction can be tested in future experiments.

Our approach separates the mechanistic question of *how* learning is implemented from what the resulting learning rules are. We nevertheless demonstrate that a simple reinforcement learning algorithm suffices to implement the learning rule we propose. Our framework makes general predictions for how instructive signals are matched to plasticity rules whenever information is transferred between different brain regions.

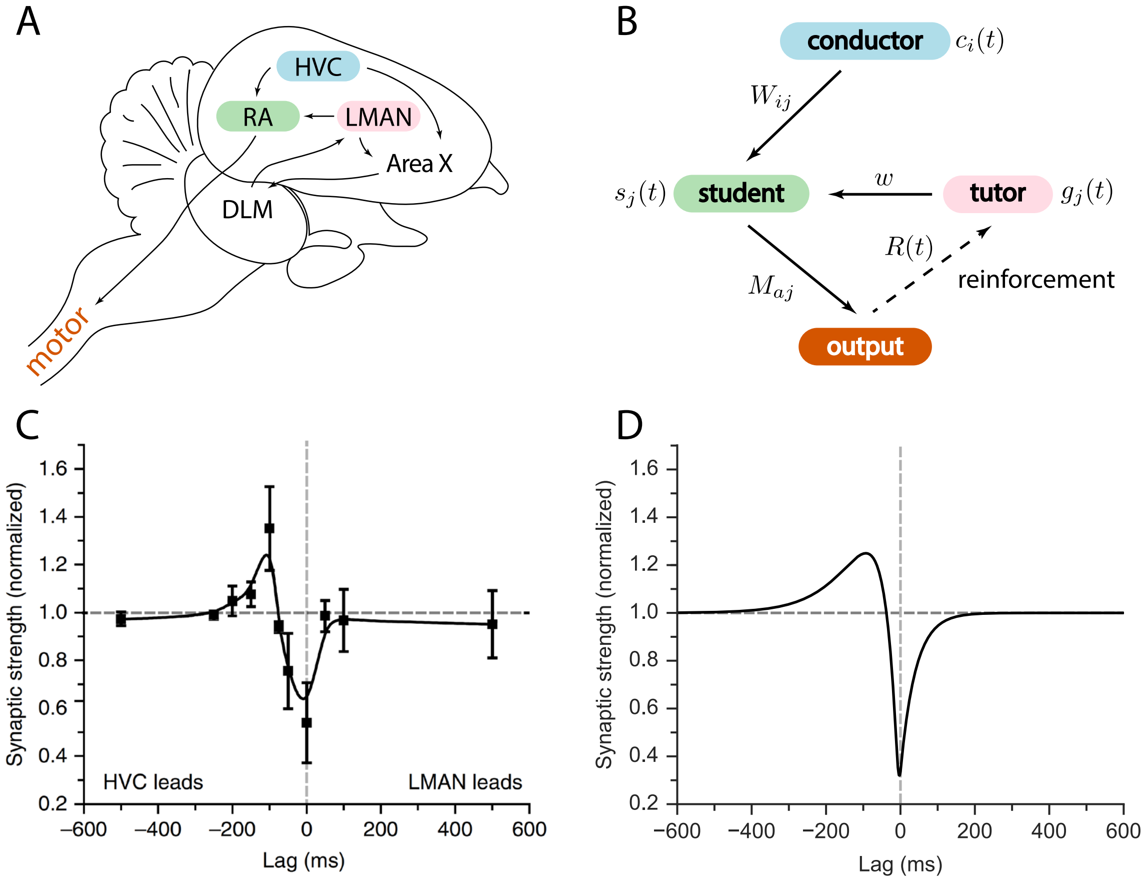

Figure 1: Relation between the song system in zebra finches and our model. **A.** Diagram of the major brain regions involved in birdsong. **B.** Conceptual model inspired by the birdsong system. The line from output to tutor is dashed because the reinforcement signal can reach the tutor either directly or, as in songbirds, indirectly. **C.** Plasticity rule measured in bird RA (measurement done in slice). When an HVC burst leads an LMAN burst by about 100 ms, the HVC–RA synapse is strengthened, while coincident firing leads to suppression. Figure adapted from [26]. **D.** Plasticity rule in our model that mimics the Mehaffey and Doupe [26] rule.

2 Results

2.1 Model

We considered a model for information transfer that is composed of three sub-circuits: a conductor, a student, and a tutor (see Fig. 1B). The conductor provides input to the student in the form of temporally precise patterns. The goal of learning is for the student to convert this input to a predefined output pattern. The tutor provides a signal that guides plasticity at the conductor–student synapses. For simplicity, we assumed that the conductor always presents the input patterns in the same order, and without repetitions. This allowed us to use the time t to label input patterns, making it easier to analyze the on-line learning rules that we studied. This model of learning is based on the logic implemented by the vocal circuits of the songbird (Fig. 1A). Relating this to the songbird, the conductor is HVC, the student is RA, and the tutor is LMAN. The song can be viewed as a mapping between clock-like HVC activity patterns and muscle-related RA outputs. The goal of learning is to find a mapping that reproduces the tutor song.

Birdsong provides interesting insights into the role of variability in tutor signals. If we focus solely on information transfer, the tutor output need not be variable; it can deterministically provide the best instructive signal to guide the student. This, however, would require the tutor to have a detailed model of the student. More realistically, the tutor might only have access to a scalar representation of how successful the student rendition of the desired output is, perhaps in the form of a reward signal. A tutor in this case has to solve the so-called 'credit assignment problem'—it needs to identify which student neurons are responsible for the reward. A standard way to achieve this is to inject variability in the student output and reinforcing the firing of neurons that precede reward (see for example [10] in the birdsong context). Thus, in our model, the tutor has a dual role of providing both an instructive signal and variability, as in birdsong.

We described the output of our model using a vector $y_a(t)$ where a indexes the various output channels (Fig. 2A). In the context of motor control a might index the muscle to be controlled, or, more abstractly, different features of the motor output, such as pitch and amplitude in the case of birdsong. The output $y_a(t)$ was a function of the activity of the student neurons $s_j(t)$. The student neurons were in turn driven by the activity of the conductor neurons $c_i(t)$. The student also received tutor signals to guide plasticity; in the songbird, the guiding signals for each RA neuron come from several LMAN neurons [4, 17, 13]. In our model, we summarized the net input from the tutor to the jth student neuron as a single function $g_j(t)$.

We started with a rate-based implementation of the model (Fig. 2A) that was analytically tractable but averaged over tutor variability. We further took the neurons to be in a linear operating regime (Fig. 2A) away from the threshold and saturation present in real neurons. We then relaxed these conditions and tested our results in spiking networks with initial parameters selected to imitate measured firing patterns in juvenile birds prior to song learning. The student circuit in both the rate-based and spiking models included a global inhibitory signal that helped to suppress excess activity driven by ongoing conductor and tutor input. Such recurrent inhibition is present in area RA of the bird [34]. In the spiking model we implemented the suppression as an activity-dependent inhibition, while for the analytic calculations we used a constant negative bias for the student neurons.

2.2 Learning in a rate-based model

Learning in our model was enabled by plasticity at the conductor–student synapses that was modulated by signals from tutor neurons (Fig. 2B). Many different forms of such hetero-synaptic plasticity have been observed. For example, in rate-based synaptic plasticity high tutor firing rates lead to synaptic potentiation and low tutor firing rates lead to depression [5, 6]. In timing-dependent rules, such as the one recently measured by Mehaffey and Doupe [26] in slices of zebra finch RA (see Fig. 1C and Fig. 1D) the relative arrival times of bursts of activity from different input pathways set the sign of synaptic change. To model learning that lies between these rate and timing-based extremes, we introduced a class of plasticity rules governed by two parameters α and β (see Methods and Fig. 2B). If we set α or β to zero in our rule, the sign of the synaptic change is determined solely by the firing rate of the tutor $g_j(t)$ as compared to a threshold, reproducing the rate rules observed in experiments. When $\alpha = \beta$, if the conductor leads the tutor, potentiation occurs, while coincident signals lead to depression (Fig. 2B), which mimics the empirical findings from [26]. For general α and β, the sign of plasticity is controlled by both the firing rate of the tutor relative to the baseline, and by the relative timing of tutor and conductor. It turns out that the overall scale of the parameters α and β can

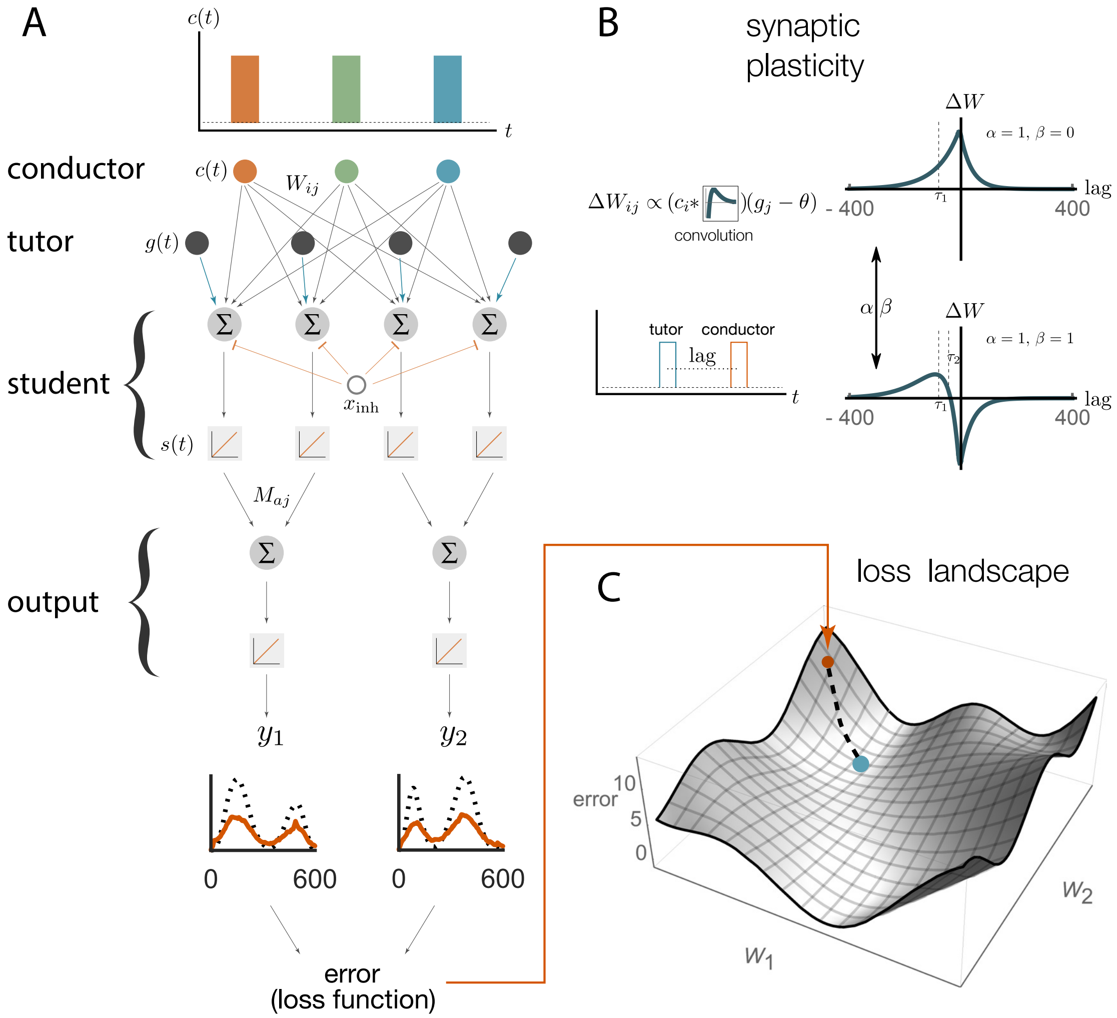

Figure 2: Schematic representation of our rate-based model. **A.** Conductor neurons fire precisely-timed bursts, similar to HVC neurons in songbirds. Conductor and tutor activities, $c(t)$ and $g(t)$, provide excitation to student neurons, which integrate these inputs and respond linearly, with activity $s(t)$. Student neurons also receive a constant inhibitory input, x_{inh}. The output neurons linearly combine the activities from groups of student neurons using weights M_{aj}. The linearity assumptions were made for mathematical convenience but are not essential for our qualitative results (see Supplementary Information). **B.** The conductor–student synaptic weights W_{ij} are updated based on a plasticity rule that depends on two parameters, α and β, and two timescales, τ_1 and τ_2 (see Methods). The tutor signal enters this rule as a deviation from a constant threshold θ. The figure shows how synaptic weights change (ΔW) for a student neuron that receives a tutor burst and a conductor burst separated by a short lag. Two different choices of plasticity parameters are illustrated. **C.** The amount of mismatch between the system's output and the target output is quantified using a loss (error) function. The figure sketches the loss landscape obtained by varying the synaptic weights W_{ij} and calculating the loss function in each case (only two of the weight axes are shown). The blue dot shows the lowest value of the loss function, corresponding to the best match between the motor output and the target, while the orange dot shows the starting point. The dashed line shows how learning would proceed in a gradient descent approach, where the weights change in the direction of steepest descent in the loss landscape.

be absorbed into the learning rate and so we set $\alpha - \beta = 1$ in all our simulations without loss of generality (see Methods).

We can ask how the conductor–student weights W_{ij} (Fig. 2A) should change in order to best improve the output $y_a(t)$. We first need a loss function L that quantifies the distance between the current output $y_a(t)$ and the target $\bar{y}_a(t)$ (Fig. 2C). We used a quadratic loss function, but other choices can also be incorporated into our framework (see Supplementary Information). Learning should change the synaptic weights so that the loss function is minimized, leading to a good rendition of the targeted output. This can be achieved by changing the synaptic weights in the direction of steepest descent of the loss function (Fig. 2C).

We used the synaptic plasticity rule from Fig. 2B to calculate the overall change of the weights, ΔW_{ij}, over the course of the motor program. This is a function of the time course of the tutor signal, $g_j(t)$. Not every choice for the tutor signal leads to motor output changes that best improve the match to the target. Imposing the condition that these changes follow the gradient descent procedure described above, we derived the tutor signal that was best matched to the student plasticity rule (detailed derivation in Methods). The result is that the best tutor for driving gradient descent learning must keep a memory of the motor error on a timescale τ_{tutor}, that is related to the synaptic plasticity parameters according to

$$\tau_{\text{tutor}} = \frac{\alpha\tau_1 - \beta\tau_2}{\alpha - \beta}\,. \tag{1}$$

This timescale is incorporated into the tutor firing rule as follows:

$$g_j(t) = \theta - \frac{\zeta}{\alpha - \beta}\frac{1}{\tau_{\text{tutor}}}\int_0^t \epsilon_j(t')e^{-(t-t')/\tau_{\text{tutor}}}\,dt'\,, \tag{2}$$

where the motor error at the student neuron j is defined as

$$\epsilon_j(t) = \sum_a M_{aj}\big(y_a(t) - \bar{y}_a(t)\big)\,. \tag{3}$$

Here M_{aj} are the weights describing the linear relationship between student activities and motor outputs (Fig. 2A) and ζ is a learning rate.

2.3 Matched *vs.* unmatched learning

Our rate-based model predicts that when the tutor is matched to the student plasticity rule as described above, learning will proceed efficiently. A mismatched tutor should slow or disrupt convergence to the desired output. To test this, we numerically simulated the birdsong circuit using the linear model from Fig. 2A with a motor output y_a filtered to more realistically reflect muscle response times (see Methods). We selected plasticity rules as described in Fig. 2B and picked a target output pattern to learn. In our simulations, the output typically involved two different channels, each with its own target, but for brevity, in figures we typically show the output from only one of these.

We tested tutors that were matched or mismatched to the plasticity rule to see how effectively they instructed the student. Fig. 3A shows convergence with a matched tutor when the sign of plasticity is determined by the tutor's firing rate. We see that the student output rapidly converged to the target. Fig. 3B shows convergence with a matched tutor when the sign of plasticity is largely determined by the relative timing of the tutor signal and the student output. We see again that the student converged steadily to the desired output, but at a somewhat slower rate than in Fig. 3A. To test the effects of mismatch between tutor and student, we used tutors with timescales that did not match eq. (1). All student plasticity rules had the same time constants τ_1 and τ_2, but different parameters α and β. Different tutors had different memory time scales τ_{tutor} (eq. (2)). Figs. 3C and 3D demonstrate that learning was more rapid for well-matched tutor-student pairs (the diagonal neighborhood). When the tutor timescale was shorter than the matched value in eq. (1) learning was often completely disrupted (many pairs below the diagonal in Figs. 3C and 3D). When the tutor timescale was longer than the matched value in eq. (1) learning was slowed down.

An interesting feature of the results from Figs. 3C, 3D is that the difference in performance between matched and mismatched pairs becomes less pronounced for timescales shorter than about 100 ms. This is due to the fact that the plasticity rule (Fig. 2B) implicitly smooths over timescales of the order of $\tau_{1,2}$,

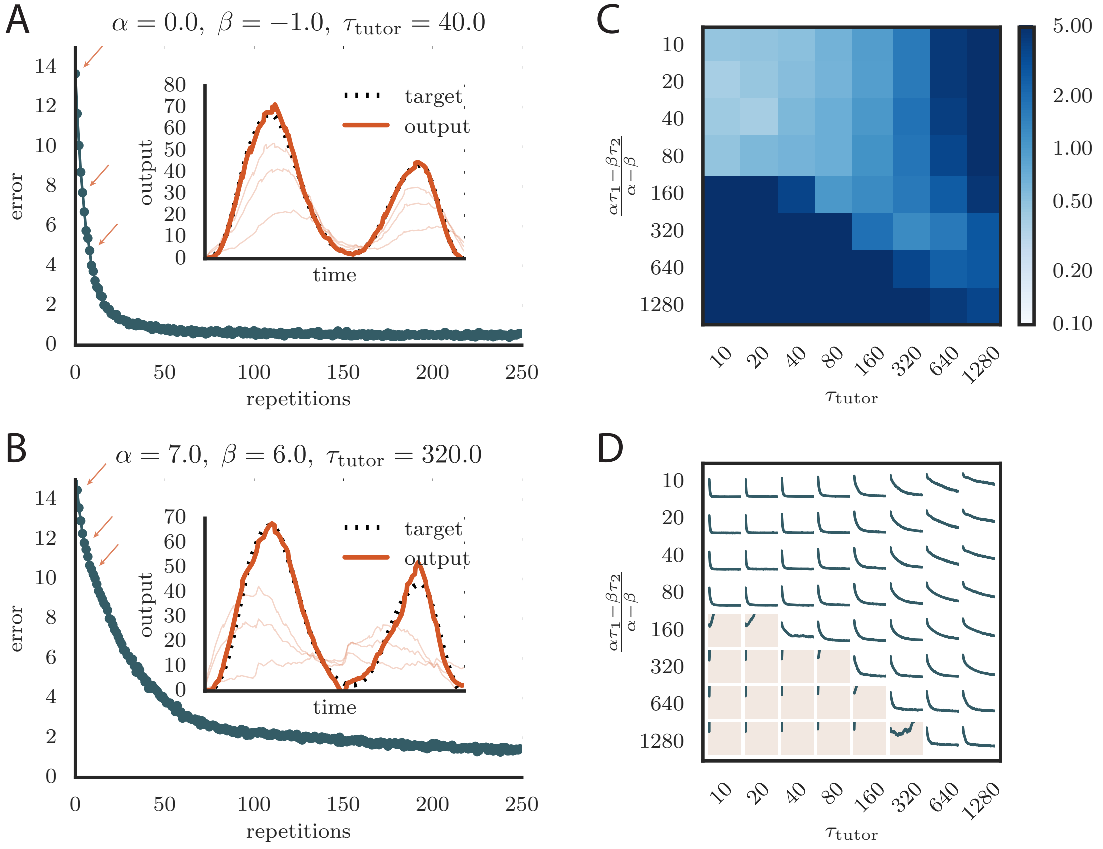

Figure 3: Learning with matched or mismatched tutors in rate-based simulations. **A.** Error trace showing how the average motor error evolved with the number of repetitions of the motor program for a rate-based plasticity rule paired with a matching tutor. **B.** The error trace and final motor output shown for a timing-based learning rule matched by a tutor with a long integration timescale. In both **A** and **B** the inset shows the final motor output for one of the two output channels (thick orange line) compared to the target output for that channel (dotted black line). The output on the first rendition and at two other stages of learning indicated by orange arrows on the error trace are also shown as thin orange lines. **C.** Effects of mismatch between student and tutor on reproduction accuracy. The heatmap shows the final reproduction error of the motor output after 250 learning cycles when a student with plasticity parameters α and β was paired with a tutor with memory timescale τ_{tutor}. The diagonal elements correspond to matched tutor and student, according to eq. (1). Here $\tau_1 = 80\,\text{ms}$ and $\tau_2 = 40\,\text{ms}$. **D.** Error evolution curves as a function of the mismatch between student and tutor. Each plot shows how the error in the motor program changed during 250 learning cycles for the same conditions as those shown in the heatmap. The region shaded in light pink shows simulations where the mismatch between student and tutor led to a deteriorating instead of improving performance during learning.

which in our simulations were equal to $\tau_1 = 80\,\text{ms}$, $\tau_2 = 40\,\text{ms}$. Thus, variations of the tutor signal on shorter timescales have little effect on learning. Using different values for the timescales $\tau_{1,2}$ in the plasticity rule can increase or decrease the range of parameters over which learning is robust against tutor–student mismatches (see Supplementary Information).

2.4 Robust learning with nonlinearities

In the model above, firing rates for the tutor were allowed to grow as large as necessary to implement the most efficient learning. However, the firing rates of realistic neurons typically saturate at some fixed bound. To test the effects of this nonlinearity in the tutor, we passed the ideal tutor activity (2) through a sigmoidal nonlinearity,

$$\tilde{g}_j(t) = \theta - \rho \tanh \frac{\zeta}{\alpha - \beta} \frac{1}{\tau_\text{tutor}} \int_0^t \epsilon_j(t') e^{-(t-t')/\tau_\text{tutor}}\, dt' \,. \tag{4}$$

where 2ρ is the range of firing rates. We typically chose $\theta = \rho = 80\,\text{Hz}$ to constrain the rates to the range 0–160 Hz [28, 13]. Learning slowed down with this change (Fig. 4A) as a result of the tutor firing rates saturating when the mismatch between the motor output and the target output was large. However, the accuracy of the final rendition was not affected by saturation in the tutor (Fig. 4A, inset). An interesting effect occurred when the firing rate constraint was imposed on a matched tutor with a long memory timescale. When this happened and the motor error was large, the tutor signal saturated and stopped growing in relation to the motor error before the end of the motor program. In the extreme case of very long integration timescales, learning became sequential: early features in the output were learned first, before later features were addressed, as in Fig. 4B. This is reminiscent of the learning rule described in [27].

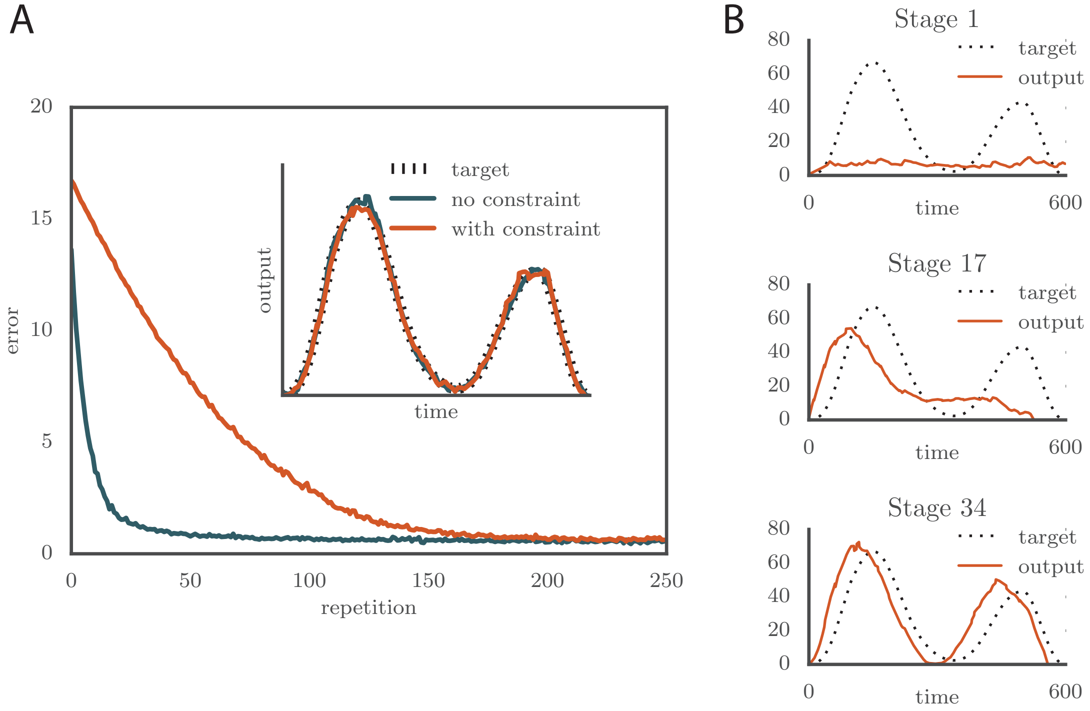

Figure 4: Effects of adding a constraint on the tutor firing rate to the simulations. **A.** Learning was slowed down by the firing rate constraint, but the accuracy of the final rendition stayed the same (inset, shown here for one of two simulated output channels). Here $\alpha = 0$, $\beta = -1$, and $\tau_\text{tutor} = 40\,\text{ms}$. **B.** Sequential learning occurred when the firing rate constraint was imposed on a matched tutor with a long memory scale. The plots show the evolution of the motor output for one of the two channels that were used in the simulation. Here $\alpha = 24$, $\beta = 23$, and $\tau_\text{tutor} = 1000\,\text{ms}$.

Nonlinearities can similarly affect the activities of student neurons. Our model can be readily extended to describe efficient learning even in this case. The key result is that for efficient learning to occur, the synaptic plasticity rule should depend not just on the tutor and conductor, but also on the activity of the postsynaptic student neurons (details in Supplementary Information). Such dependence on postsynaptic activity is commonly seen in experiments [5, 6].

The relation between student activations $s_j(t)$ and motor outputs $y_a(t)$ (Fig. 2A) is in general also nonlinear. Compared to the linear assumption that we used, the effect of a monotonic nonlinearity, $y_a = N_a(\sum_j M_{aj} s_j)$, with N_a an increasing function, is similar to modifying the loss function L, and does not significantly change our results (Supplementary Information). We also checked that imposing a rectification constraint that conductor–student weights W_{ij} must be positive does not modify our results either (Supplementary Information). This shows that our model continues to work with biologically realistic synapses that cannot change sign from excitatory to inhibitory during learning.

2.5 Spiking neurons and birdsong

To apply our model to vocal learning in birds, we extended our analysis to networks of spiking neurons. In the bird, juveniles produce a "babble" that converges through learning to an adult song strongly resembling the tutor song. This is reflected in the song-aligned spiking patterns in pre-motor area RA, which become more stereotyped and cluster in shorter, better-defined bursts as the bird matures (Fig. 5A). We tested whether our model could reproduce key statistics of spiking in RA over the course of song learning. In this context, our theory of efficient learning, derived in a rate-based scenario, predicts a specific relation between the teaching signal embedded in LMAN firing patterns, and the plasticity rule implemented in RA. We tested whether these predictions continued to hold in the spiking context.

Following the experiments of Hahnloser, Kozhevnikov, and Fee [14], we modeled each neuron in HVC (the conductor) as firing one short, precisely timed burst of 5-6 spikes at a single moment in the motor program. Thus the population of HVC neurons produced a precise timebase for the song. LMAN (tutor) neurons are known to have highly variable firing patterns that facilitate experimentation, but also contain a corrective bias [2]. Thus we modeled LMAN as producing inhomogeneous Poisson spike trains with a time-dependent firing rate given by eq. (4) in our model. Although biologically there are several LMAN neurons projecting to each RA neuron, we again simplified by "summing" the LMAN inputs into a single, effective tutor neuron, similarly to the approach in [10]. The LMAN-RA synapses were modeled in a current-based approach as a mixture of AMPA and NMDA receptors, following the songbird data [35, 13]. The initial weights for all synapses were tuned to produce RA firing patterns resembling juvenile birds [29], subject to constraints from direct measurements in slice recordings [13] (see Methods for details, and Fig. 5B for a comparison between neural recordings and spiking in our model). In contrast to the constant inhibitory bias that we used in our rate-based simulations, for the spiking simulations we chose an activity-dependent global inhibition for RA neurons. We also tested that a constant bias produced similar results (see Supplementary Information).

Synaptic strength updates followed the same two-timescale dynamics that was used in the rate-based models (Fig. 2B). The firing rates $c_i(t)$ and $g_j(t)$ that appear in the plasticity equation were calculated in the spiking model by filtering the spike trains from conductor and tutor neurons with exponential kernels. The synaptic weights were constrained to be non-negative. (See Methods for details.)

Once again, learning proceeded effectively when the tutor timescale and the student plasticity rule were matched (see Fig. 5C), with mismatches slowing down or abolishing learning, just as in our rate-based study (compare Fig. 5D with Fig. 3C). The rate of learning and the accuracy of the trained state were lower in the spiking model compared to the rate-based model. The lower accuracy arises because the tutor neurons fire stochastically, unlike the deterministic neurons used in the rate-based simulations.

Spiking in our model tends to be a little more regular than that in the recordings (compare Fig. 5A and Fig. 5B). This could be due to sources of noise that are present in the brain which we did not model. One detail that our model does not capture is the fact that many LMAN spikes occur in bursts, while in our simulation LMAN firing is Poisson. Bursts are more likely to produce spikes in downstream RA neurons particularly because of the NMDA dynamics, and thus a bursty LMAN will be more effective at injecting variability into RA [22]. Small inaccuracies in aligning the recorded spikes to the song are also likely to contribute apparent variability between renditions in the experiment. Indeed, some of the variability in Fig. 5A looks like it could be due to time warping and global time shifts that were not fully corrected.

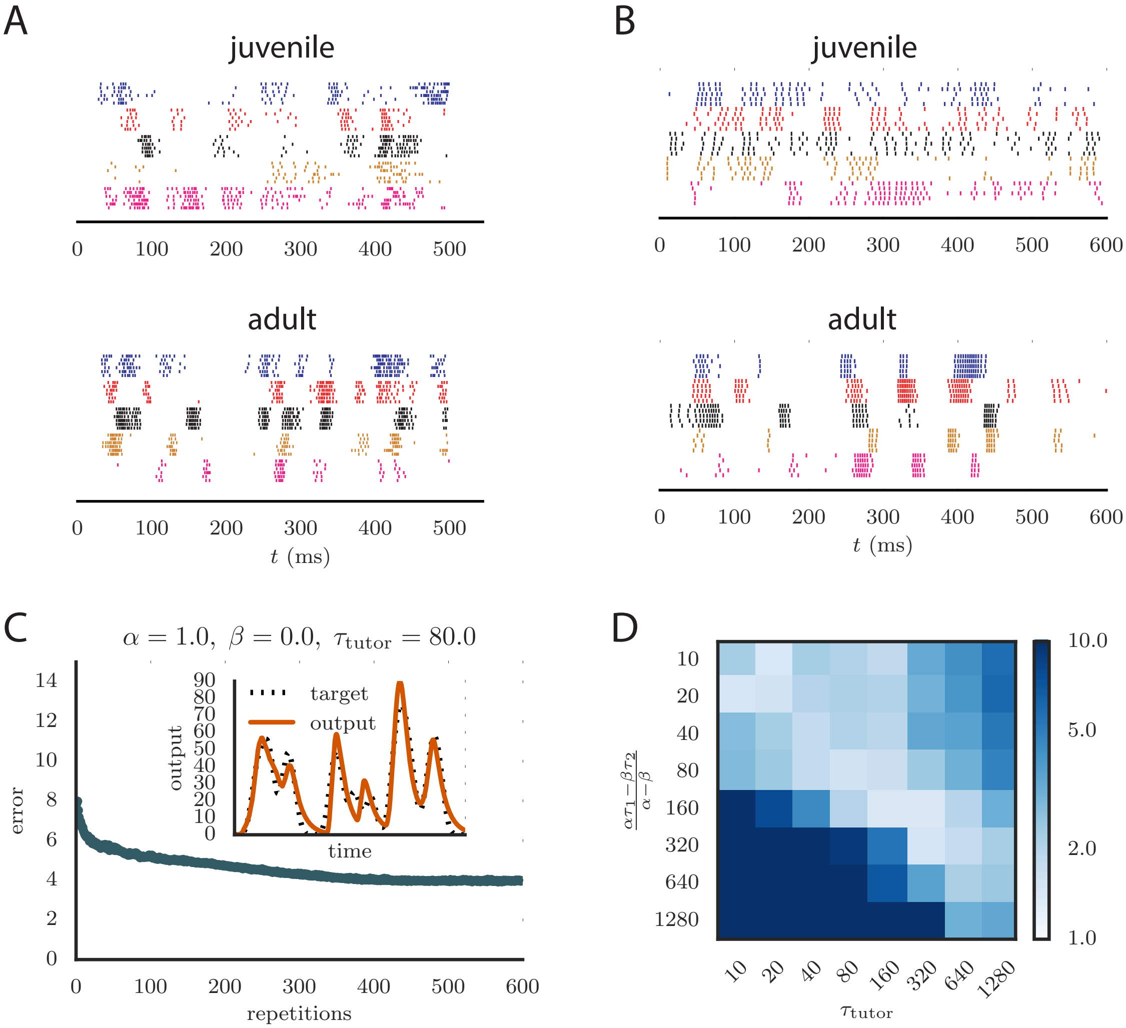

Figure 5: Results from simulations in spiking neural networks. **A.** Spike patterns recorded from zebra finch RA during song production, for a juvenile (top) and an adult (bottom). Each color corresponds to a single neuron, and the song-aligned spikes for six renditions of the song are shown. Adapted from [29]. **B.** Spike patterns from model student neurons in our simulations, for the untrained (top) and trained (bottom) models. The training used $\alpha = 1$, $\beta = 0$, and $\tau_{\text{tutor}} = 80\,\text{ms}$, and ran for 600 iterations of the song. Each model neuron corresponds to a different output channel of the simulation. In this case, the targets for each channel were chosen to roughly approximate the time course observed in the neural recordings. **C.** Progression of reproduction error in the spiking simulation as a function of the number of repetitions for the same conditions as in panel B. The inset shows the accuracy of reproduction in the trained model for one of the output channels. **D.** Effects of mismatch between student and tutor in the spiking model. The heatmap shows the final reproduction error of the motor output after 250 learning cycles when a student with plasticity parameters α, β was paired with a tutor with memory timescale τ_{tutor}. Here each simulation has two output channels, as in our rate-based simulations.

2.6 Robust learning with credit assignment errors

The calculation of the tutor output in our rule involved estimating the motor error ϵ_j from eq. (3). This required knowledge of the assignment between student activities and motor output, which in our model was represented by the matrix M_{aj} (Fig. 2A). In our simulations, we typically chose an assignment in which each student neuron contributed to a single output channel, mimicking the empirical findings for neurons in bird RA. Mathematically, this implies that each column of M_{aj} contained a single non-zero element. In Fig. 6A, we show what happened in the rate-based model when the tutor incorrectly assigned a certain fraction of the neurons to the wrong output. Specifically, we considered two output channels, y_1 and y_2, with half of the student neurons contributing only to y_1 and the other half contributing only to y_2. We then scrambled a fraction ρ of this assignment when calculating the motor error, so that the tutor effectively had an imperfect knowledge of the student–output relation. Fig. 6A shows that learning is robust to this kind of mis-assignment even for fairly large values of the error fraction ρ up to about 40%, but quickly deteriorates as this fraction approaches 50%.

Due to environmental factors that affect development of different individuals in different ways, it is unlikely that the student–output mapping can be innate. As such, the tutor circuit must learn the mapping. Indeed, it is known that LMAN in the bird receives an indirect evaluation signal *via* Area X, which might be used to effect this learning [2, 23, 18]. One way in which this can be achieved is through a reinforcement paradigm. We thus considered a learning rule where the tutor circuit receives a reward signal that enables it to infer the student–output mapping. In general the output of the tutor circuit should depend on an integral of the motor error, as in eq. (2), to best instruct the student. For simplicity, we start with the memory-less case, $\tau_{\text{tutor}} = 0$, in which only the instantaneous value of the motor error is reflected in the tutor signal; we then show how to generalize this for $\tau_{\text{tutor}} > 0$.

As before, we took the tutor neurons to fire Poisson spikes with time-dependent rates $f_j(t)$, which were initialized arbitrarily. Because of stochastic fluctuations, the actual tutor activity on any given trial, $g_j(t)$, differs somewhat from the average, $\bar{g}_j(t)$. Denoting the difference by $\xi_j(t) = g_j(t) - \bar{g}_j(t)$, the update rule for the tutor firing rates was given by

$$\Delta f_j(t) = \eta_{\text{tutor}}(R(t) - \bar{R})\xi_j(t)\,, \tag{5}$$

where η_{tutor} is a learning rate, $R(t)$ is the instantaneous reward signal, and $\bar{R}$ is its average over recent renditions of the motor program. The intuition behind this rule is that, whenever a fluctuation in the tutor activity leads to better-than-average reward ($R(t) > \bar{R}$), the tutor firing rate changes in the direction of the fluctuation for subsequent trials, "freezing in" the improvement. Conversely, the firing rate moves away from the directions in which fluctuations tend to reduce the reward.

To test our learning rule, we ran simulations using this reinforcement strategy and found that learning again converges to an accurate rendition of the target output (Fig. 6B, inset). The number of repetitions needed for training is greatly increased compared to the case in which the credit assignment is assumed known by the tutor circuit (compare Fig. 6B to Fig. 5C). This is because the tutor needs to use many training rounds for experimentation before it can guide the conductor–student plasticity. Because of the extra training time needed for the tutor to adapt its signal, the motor output in our reward-based simulations tends to initially overshoot the target (leading to the kink in the error at around 2000 repetitions in Fig. 6B). Interestingly, the subsequent reduction in output that leads to convergence of the motor program, combined with the positivity constraint on the synaptic strengths, leads to many conductor–student connections being pruned (Fig. 6D). This mirrors experiments on songbirds, where the number of connections between HVC and RA first increases with learning and then decreases [13].

The reinforcement rule described above responds only to instantaneous values of the reward signal and tutor firing rate fluctuations. In general, effective learning requires that the tutor keep a memory trace of its activity over a timescale $\tau_{\text{tutor}} > 0$, as in eq. (1). To achieve this in the reinforcement paradigm, we can use a simple generalization of eq. (5) where the update rule is filtered over the tutor memory timescale:

$$\Delta f_j(t) = \eta_{\text{tutor}} \frac{1}{\tau_{\text{tutor}}} \int^t dt'\, (R(t') - \bar{R})\xi_j(t') e^{-(t-t')/\tau_{\text{tutor}}}\,. \tag{6}$$

We tested that this rule leads to effective learning when paired with the corresponding student, *i.e.,* one for which eq. (1) is obeyed (Fig. 6C).

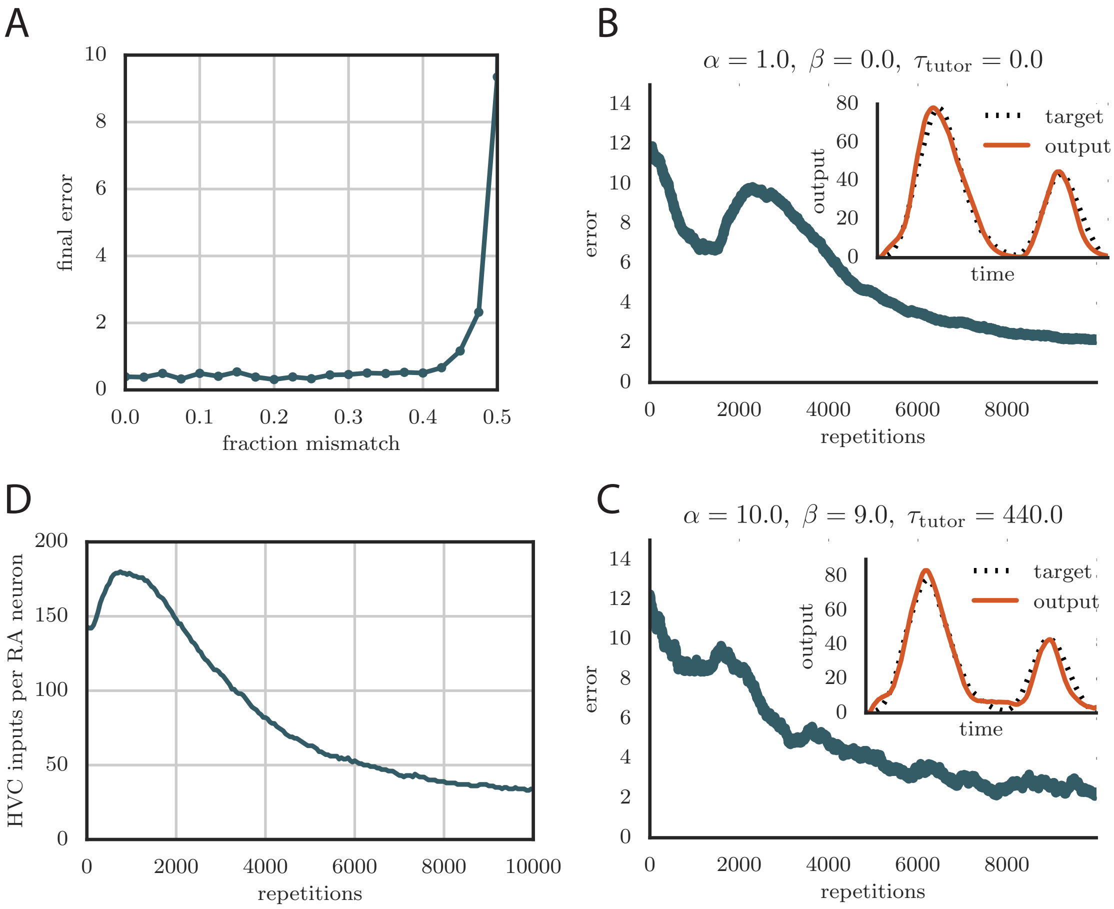

Figure 6: Credit assignment and reinforcement learning. **A.** Effects of credit mis-assignment on learning in a rate-based simulation. Here, the system learned output sequences for two independent channels. The student–output weights M_{aj} were chosen so that the tutor wrongly assigned a fraction of student neurons to an output channel different from the one it actually mapped to. The graph shows how the accuracy of the motor output after 250 learning steps depended on the fraction of mis-assigned credit. **B.** Learning curve and trained motor output (inset) for one of the channels showing two-stage reinforcement-based learning for the memory-less tutor ($\tau_{\text{tutor}} = 0$). The accuracy of the trained model is as good as in the case where the tutor was assumed to have a perfect model of the student–output relation. However, the speed of learning is reduced. **C.** Learning curve and trained motor output (inset) for one of the output channels showing two-stage reinforcement-based learning when the tutor circuit needs to integrate information about the motor error on a certain timescale. Again, learning was slow, but the accuracy of the trained state was unchanged. **D.** Evolution of the average number of HVC inputs per RA neuron with learning in a reinforcement example. Synapses were considered pruned if they admitted a current smaller than 1 nA after a pre-synaptic spike in our simulations.

The reinforcement rules proposed here are related to the learning rules from [11, 10] and [8]. However, those models focused on learning in a single pass, instead of the two-stage architecture that we studied. In particular, in [10], area LMAN was assumed to generate pure Poisson noise and reinforcement learning took place at the HVC–RA synapses. In our model, which is in better agreement with recent evidence regarding the roles of RA and LMAN in birdsong [2], reinforcement learning first takes place in the anterior forebrain pathway (AFP), for which LMAN is the output. A reward-independent heterosynaptic plasticity rule then solidifies the information in RA.

In our simulations, tutor neurons fire Poisson spikes with specific time-dependent rates which change during learning. The timecourse of the firing rates in each repetition then must be stored somewhere in the brain. In fact, in the songbird, there are indirect projections from HVC to LMAN, going through the basal ganglia (Area X) and the dorso-lateral division of the medial thalamus (DLM) in the anterior forebrain pathway (Fig. 1A) [30]. These synapses could store the required time-dependence of the tutor firing rates. In addition, the same synapses can provide the timebase input that would ensure synchrony between LMAN firing and RA output, as necessary for learning. Our reinforcement learning rule for the tutor area, eq. (5), can be viewed as an effective model for plasticity in the projections between HVC, Area X, DLM, and LMAN, as in [9]. In this picture, the indirect HVC–LMAN connections behave somewhat like the "hedonistic synapses" from [32], though we use a simpler synaptic model here. Implementing the integral from eq. (6) would require further recurrent circuitry in LMAN which is beyond the scope of this paper, but would be interesting to investigate in future work.

3 Discussion

We built a two-stage model of learning in which one area (the student) learns to perform a sequence of actions under guidance from a tutor area. This architecture is inspired by the song system of zebra finches, where area LMAN provides a corrective bias to the song that is then consolidated in the HVC–RA synapses. Using an approach rooted in the effective coding literature, we showed analytically that, in a simple model, the tutor output that is most likely to lead to effective learning by the student involves an integral over the recent magnitude of the motor error. We found that efficiency requires that the timescale for this integral should be related to the synaptic plasticity rule used by the student. Using simulations, we tested our findings in more general settings. In particular, we demonstrated that tutor-student matching is important for learning in a spiking-neuron model constructed to reproduce spiking patterns similar to those measured in zebra finches. Learning in this model changes the spiking statistics of student neurons in realistic ways, for example, by producing more bursty, stereotyped firing events as learning progresses. Finally, we showed how the tutor can build its error-correcting signal by means of reinforcement learning.

If the birdsong system has evolved to support efficient learning, our model predicts the temporal structure of firing patterns of RA-projecting LMAN neurons, given the plasticity rule implemented at the HVC–RA synapses. These predictions can be directly tested in recordings in behaving animals. Our model can be applied more generally to other systems in the brain exhibiting two-stage learning, such as motor learning in mammals. If the plasticity mechanisms in these systems are different from those in songbirds, our predictions for the structure of the guiding signal will vary correspondingly. This would allow a stringent test of our model of "efficient learning" in the brain.

Applied to birdsong, our model is best seen as a mechanism for learning song syllables. The ordering of syllables in song motifs seems to have a second level of control within HVC and perhaps beyond [3, 15]. Songs can also be distorted by warping their timebase through changes in HVC firing without alterations of the HVC–RA connectivity [1]. In view of these phenomena, it would be interesting to incorporate our model into a larger hierarchical framework in which the syllable sequence information is also learned. A model of transitions between syllables can be found in [7], where the authors use a "weight perturbation" optimization scheme in which each HVC–RA synaptic weight is perturbed individually. We did not follow this approach because there is no plausible mechanism for LMAN to provide separate guidance to each HVC–RA synapse; in particular, there are not enough LMAN neurons [10].

In this paper we assumed a two-stage architecture for learning, inspired by birdsong. An interesting question is whether and under what conditions such an architecture is more effective than a single-step model. Possibly, having two stages is better when a single tutor area is responsible for training several different dedicated controllers, as is likely the case in motor learning. It would then be beneficial to have an area that can learn arbitrary behaviors, perhaps at the cost of using more resources and having slower reaction times, along with the ability to transfer these behaviors into low-level circuitry that is only capable of producing stereotyped motor programs. The question then arises whether having more than two levels in this hierarchy could be useful, what the other levels might do, and whether such hierarchical learning systems are implemented in the brain.

Acknowledgments

We would like to thank Serena Bradde for fruitful discussions during the early stages of this work. We also thank Xuexin Wei and Christopher Glaze for useful discussions. We are grateful to Timothy Otchy for providing us with some of the data we used in this paper. During this work VB was supported by NSF grant PHY-1066293 at the Aspen Center for Physics and by NSF Physics of Living Systems grant PHY-1058202. TT was supported by the Swartz Foundation.

A Methods

A.1 Equations for rate-based model

The basic equations we used for describing our rate-based model (Fig. 2A) are the following:

$$\begin{aligned} y_a(t) &= \sum_j M_{aj} s_j(t)\,, \\ s_j(t) &= \sum_i W_{ij} c_i(t) + w g_j(t) - x_{\text{inh}}\,. \end{aligned} \tag{A.1}$$

In simulations, we further filtered the output using an exponential kernel,

$$\tilde{y}_a(t) = \sum_j M_{aj} \int_0^\infty s_j(t')\, e^{-(t-t')/\tau_{\text{out}}}\, dt'\,, \tag{A.2}$$

with a timescale τ_{out} that we typically set to 25 ms. The smoothing produces more realistic outputs by mimicking the relatively slow reaction time of real muscles, and stabilizes learning by filtering out high-frequency components of the motor output. The latter interfere with learning because of the delay between the effect of conductor activity on synaptic strengths *vs.* motor output. This delay is of the order $\tau_{1,2} - \tau_{\text{out}}$ (see the plasticity rule below).

The conductor activity in the rate-based model is modeled after songbird HVC [14]: each neuron fires a single burst during the motor program. Each burst corresponds to a sharp increase of the firing rate $c_i(t)$ from 0 to a constant value, and then a decrease 10 ms later. The activities of the different neurons are spread out to tile the whole duration of the output program. Other choices for the conductor activity also work, provided no patterns are repeated (see Supplementary Information).

A.2 Mathematical description of plasticity rule

In our model the rate of change of the synaptic weights obeys a rule that depends on a filtered version of the conductor signal (see Fig. 2B). This is expressed mathematically as

$$\frac{dW_{ij}}{dt} = \eta\, \tilde{c}_i(t)\, (g_j(t) - \theta)\,, \tag{A.3}$$

where η is a learning rate and $\tilde{c}_i = K * c_i$, with the star representing convolution and K being a filtering kernel. We considered a linear combination of two exponential kernels with timescales τ_1 and τ_2,

$$K(t) = \alpha K_1(t) - \beta K_2(t)\,, \tag{A.4}$$

with $K_i(t)$ given by

$$K_i(t) = \begin{cases} \tau_i^{-1} e^{-t/\tau_i} & \text{for } t \geq 0, \\ 0 & \text{else.} \end{cases} \tag{A.5}$$

Different choices for the kernels give similar results (see Supplementary Information). The overall scale of α and β can be absorbed into the learning rate η in eq. (A.3). In our simulations, we fix $\alpha - \beta = 1$ and keep the learning rate constant as we change the plasticity rule (see eq. (2)).

In the spiking simulations with and without reinforcement learning in the tutor circuit, the firing rates $c_i(t)$ and $g_j(t)$ were estimated by filtering spike trains with exponential kernels whose timescales were in the range 5 ms–40 ms. The reinforcement studies typically required longer timescales for stability, possibly because of delays between conductor activity and reward signals.

A.3 Derivation of the matching tutor signal

To find the tutor signal that provides the most effective teaching for the student, we first calculate how much synaptic weights change according to our plasticity rule, eq. (A.3). Then we require that this change matches the gradient descent direction. We have

$$\Delta W_{ij} = \int_0^T \frac{dW_{ij}}{dt}\, dt = \eta \int_0^T \tilde{c}_i(t)(g_j(t) - \theta)\, dt\,. \tag{A.6}$$

Because of the linearity assumptions in our model, it is sufficient to focus on a case in which each conductor neuron, i, fires a single short burst, at a time t_i. We write this as $c_i(t) = \delta(t - t_i)$, and so

$$\Delta W_{ij} = \int_0^T \frac{dW_{ij}}{dt}\, dt = \eta \int_0^T K(t - t_i)(g_j(t) - \theta)\, dt\,, \tag{A.7}$$

where we used the definition of $\tilde{c}_i(t)$. If the time constants τ_1, τ_2 are short compared to the timescale on which the tutor input $g_j(t)$ varies, only the values of $g_j(t)$ around time t_i will contribute to the integral. If we further assume that $T \gg t_i$, we can use a Taylor expansion of $\bar{g}_j \equiv g_j(t) - \theta$ around t_i to perform the calculation:

$$\begin{aligned}\Delta W_{ij} &\approx \eta \int_{t_i}^{\infty} K(t - t_i)\big(\bar{g}_j(t_i) + (t - t_i)\bar{g}_j'(t_i)\big)\, dt \\ &= \eta \bar{g}_j(t_i) \int_0^{\infty} K(t)\, dt + \eta \bar{g}_j'(t_i) \int_0^{\infty} tK(t)\, dt \\ &= \eta \bar{g}_j(t_i) \int_0^{\infty} \big(\alpha K_1(t) - \beta K_2(t)\big)\, dt + \eta \bar{g}_j'(t_i) \int_0^{\infty} t\, \big(\alpha K_1(t) - \beta K_2(t)\big)\, dt\,.\end{aligned} \tag{A.8}$$

Doing the integrals involving the exponential kernels K_1 and K_2, we get

$$\Delta W_{ij} = \eta\big[(\alpha - \beta)\bar{g}_j(t_i) + (\alpha\tau_1 - \beta\tau_2)\bar{g}_j'(t_i)\big]\,. \tag{A.9}$$

We would like this synaptic change to optimally reduce a measure of mismatch between the output and the desired target as measured by a loss function. A generic smooth loss function $L(y_a(t), \bar{y}_a(t))$ can be quadratically approximated when y_a is sufficiently close to the target $\bar{y}_a(t)$. With this in mind, we consider a quadratic loss

$$L = \frac{1}{2} \sum_a \int_0^T \big[y_a(t) - \bar{y}_a(t)\big]^2\, dt\,. \tag{A.10}$$

The loss function would decrease monotonically during learning if synaptic weights changed in proportion to the negative gradient of L:

$$\Delta W_{ij} = -\gamma \frac{\partial L}{\partial W_{ij}}\,, \tag{A.11}$$

where γ is a learning rate. This implies

$$\Delta W_{ij} = -\gamma \sum_a \int_0^T M_{aj}\big[y_a(t) - \bar{y}_a(t)\big]\, c_i(t)\,. \tag{A.12}$$

Using again $c_i(t) = \delta(t - t_i)$, we obtain

$$\Delta W_{ij} = -\gamma \epsilon_j(t_i)\,, \tag{A.13}$$

where we used the notation from eq. (3) for the motor error at student neuron j.

We now set (A.9) and (A.13) equal to each other. If the conductor fires densely in time, we need the equality to hold for all times, and we thus get a differential equation for the tutor signal $\bar{g}_j(t)$. This identifies the tutor signal that leads to gradient descent learning as a function of the motor error $\epsilon_j(t)$, eq. (2) (with the notation $\zeta = \gamma/\eta$).

A.4 Spiking simulations

We used spiking models that were based on leaky integrate-and-fire neurons with current-based dynamics for the synaptic inputs. The magnitude of synaptic potentials generated by the conductor–student synapses was independent of the membrane potential, approximating AMPA receptor dynamics, while the synaptic inputs from the tutor to the student were based on a mixture of AMPA and NMDA dynamics. Specifically, the equations describing the dynamics of the spiking model were:

$$
\begin{aligned}
\tau_m \frac{dV_j}{dt} &= (V_R - V_j) + R\left(I_j^{\text{AMPA}} + I_j^{\text{NMDA}}\right) - V_{\text{inh}}\,, \qquad \text{(except during refractory period)}\\
\frac{dI_j^{\text{AMPA}}}{dt} &= -\frac{I_j^{\text{AMPA}}}{\tau_{\text{AMPA}}} + \sum_i W_{ij} \sum_k \delta(t - t_k^{\text{conductor \#i}}) + rw \sum_k \delta(t - t_k^{\text{tutor}})\,,\\
\frac{dI_j^{\text{NMDA}}}{dt} &= -\frac{I_j^{\text{NMDA}}}{\tau_{\text{NMDA}}} + (1-r)wG(V_j) \sum_k \delta(t - t_k^{\text{tutor}})\,,\\
V_{\text{inh}} &= \frac{g_{\text{inh}}}{N_{\text{student}}} \sum_j S_j(t)\,,\\
\frac{dS_j}{dt} &= -\frac{S_j}{\tau_{\text{inh}}} + \sum_k \delta(t - t_k^{\text{student}})\,,\\
G(V) &= \left[1 + \frac{[\text{Mg}]}{3.57\,\text{mM}} \exp(-V/16.13\,\text{mV})\right]^{-1}\,.
\end{aligned}
\tag{A.14}
$$

Here V_j is the membrane potential of the j^{th} student neuron and V_R is the resting potential, as well as the potential to which the membrane was reset after a spike. Spikes were registered whenever the membrane potential went above a threshold V_{th}, after which a refractory period τ_{ref} ensued. Table A.1 gives the values of the parameters we used in the simulations. These values were chosen to match the firing statistics of neurons in bird RA, as described below.

Parameter	Symbol	Value	Parameter	Symbol	Value
No. of conductor neurons		300	No. of student neurons		80
Reset potential	V_R	$-72.3\,\text{mV}$	Input resistance	R	$353\,\text{M}\Omega$
Threshold potential	V_{th}	$-48.6\,\text{mV}$	Strength of inhibition	g_{inh}	$1.80\,\text{mV}$
Membrane time constant	τ_m	$24.5\,\text{ms}$	Fraction NMDA receptors	r	0.9
Refractory period	τ_{ref}	$1.1\,\text{ms}$	Strength of synapses from tutor	w	$100\,\text{nA}$
AMPA time constant	τ_{AMPA}	$6.3\,\text{ms}$	No. of conductor synapses per student neuron		148
NMDA time constant	τ_{NMDA}	$81.5\,\text{ms}$	Mean strength of synapses from conductor		$32.6\,\text{nA}$
Time constant for global inhibition	τ_{inh}	$20\,\text{ms}$	Standard deviation of conductor–student weights		$17.4\,\text{nA}$
Conductor firing rate during bursts		$632\,\text{Hz}$			

Table A.1: Values for parameters used in the spiking simulations.

The voltage dynamics for conductor and tutor neurons was not simulated explicitly. Instead, each conductor neuron was assumed to fire a burst at a fixed time during the simulation. The onset of each burst had additive timing jitter of $\pm 0.3\,\text{ms}$ and each spike in the burst had a jitter of $\pm 0.2\,\text{ms}$. This modeled the uncertainty in spike times that is observed in *in vivo* recordings in birdsong [14]. Tutor neurons fired Poisson spikes with a time-dependent firing rate that was set as described in the main text.

The initial connectivity between conductor and student neurons was chosen to be sparse (see Table A.1). The initial distribution of synaptic weights was log-normal, matching experimentally measured values for zebra finches [13]. Since these measurements are done in the slice, the absolute number of HVC synapses per RA neuron is likely to have been underestimated. The number of conductor–student synapses we start with in our simulations is thus chosen to be higher than the value reported in that paper (see Table A.1), and is allowed to change during learning. We checked that the learning paradigm described here is robust to substantial changes in these parameters, but we have chosen values that are faithful to birdsong experiments and which are thus able to imitate the RA spiking statistics during song.

The synapses projecting onto each student neuron from the tutor have a weight that is fixed during our simulations reflecting the finding in [13] that the average strength of LMAN–RA synapses for zebra finches does not change with age. There is some evidence that individual LMAN–RA synapses undergo plasticity concurrently with the HVC–RA synapses [26] but we did not seek to model this effect.

A.5 Matching spiking statistics with experimental data

We used an optimization technique to choose parameters to maximize the similarity between the statistics of spiking in our simulations and the firing statistics observed in neural recordings from the songbird. The comparison was based on several descriptive statistics: the average firing rate; the coefficient of variation and skewness of the distribution of inter-spike intervals; the frequency and average duration of bursts; and the firing rate during bursts. For calculating these statistics, bursts were defined to start if the firing rate went above 80 Hz and last until the rate decreased below 40 Hz.

To carry out such optimizations in the stochastic context of our simulations, we used an evolutionary algorithm—the covariance matrix adaptation evolution strategy (CMA-ES) [16]. The objective function was based on the relative error between the simulation statistics x_i^{sim} and the observed statistics x_i^{obs},

$$\text{error} = \left[\sum_i \left(\frac{x_i^{\text{sim}}}{x_i^{\text{obs}}} - 1 \right)^2 \right]^{1/2} . \tag{A.15}$$

Equal weight was placed on optimizing the firing statistics in the juvenile (based on a recording from a 43 dph bird) and optimizing firing in the adult (based on a recording from a 160 dph bird). In this optimization there was no learning between the juvenile and adult stages. We simply required that the number of HVC synapses per RA neuron, and the mean and standard deviation of the corresponding synaptic weights were in the ranges seen in the juvenile and adult by [13]. The optimization was carried out in `Python`, using code from `https://www.lri.fr/~hansen/cmaes_inmatlab.html`. The results fixed the parameter choices in Table A.1 which were then used to study our learning paradigm. While these choices are important for achieving firing statistics that are similar to those seen in recordings from the bird, our learning paradigm is robust to large variations in the parameters in Table A.1.

A.6 Software and data

We used custom-built `Python` code for simulations and data analysis. The software and data that we used can be accessed online at `https://github.com/ttesileanu/twostagelearning`.

B Supplementary Information

B.1 Effect of nonlinearities

We can generalize the model from eq. (A.1) by using a nonlinear transfer function from student activities to motor output, and a nonlinear activation function for student neurons:

$$\begin{aligned} y_a(t) &= N_a\Bigl(\sum_j M_{aj} s_j(t)\Bigr)\,, \\ s_j(t) &= F\Bigl(\sum_i W_{ij} c_i(t) + w g_j(t) - x_{\text{inh}}\Bigr)\,. \end{aligned} \tag{B.1}$$

Suppose further that we use a general loss function,

$$L = \int_0^T \mathcal{L}\bigl(\{y_a(t) - \bar{y}_a(t)\}\bigr)\, dt\,. \tag{B.2}$$

Carrying out the same argument as that from section A.3, the gradient descent condition, eq. (A.11), implies

$$\Delta W_{ij} = -\gamma \int_0^T \sum_a M_{aj} N_a' F' c_i(t)\, \frac{\partial \mathcal{L}}{\partial y_a}\bigg|_{y_a(t)-\bar{y}_a(t)}\,. \tag{B.3}$$

The departure from the quadratic loss function, $\mathcal{L} \neq \frac{1}{2}\sum_a (y_a(t) - \bar{y}_a(t))^2$, and the nonlinearities in the output, N_a, have the effect of redefining the motor error,

$$\epsilon_j(t) = \sum_a M_{aj} N_a' \,\frac{\partial \mathcal{L}}{\partial y_a}\bigg|_{y_a(t)-\bar{y}_a(t)}\,. \tag{B.4}$$

A proper loss function will be such that the derivatives $\partial\mathcal{L}/\partial y_a$ vanish when $y_a(t) = \bar{y}_a(t)$, and so the motor error ϵ_j as defined here is zero when the rendition is perfect, as expected. If we use a tutor that ignores the nonlinearities in a nonlinear system, *i.e.,* if we use eq. (3) instead of eq. (B.4) to calculate the tutor signal that is plugged into eq. (2), we still expect successful learning provided that $N_a' > 0$ and that $\mathcal{L}$ is itself an increasing function of $|y_a - \bar{y}_a|$ (see section B.2). This is because replacing eq. (B.4) with eq. (3) would affect the magnitude of the motor error without significantly changing its direction. In more complicated scenarios, if the transfer function to the output is not monotonic, there is the potential that using eq. (3) would push the system away from convergence instead of towards it. In such a case, an adaptive mechanism, such as the reinforcement rules from eqns. (5) or (6) can be used to adapt to the local values of the derivatives N_a' and $\partial\mathcal{L}/\partial y_a$.

Finally, the nonlinear activation function F introduces a dependence on the student output $s_j(t)$ in eq. (B.3), since F' is evaluated at $F^{-1}(s_j(t))$. To obtain a good match between the student and the tutor in this context, we can modify the student plasticity rule (A.3) by adding a dependence on the postsynaptic activity,

$$\frac{dW_{ij}}{dt} = \eta\, \tilde{c}_i(t)\, (g_j(t) - \theta)\, F'(F^{-1}(s_j(t)))\,. \tag{B.5}$$

In general, synaptic plasticity has been observed to indeed depend on postsynaptic activity [5, 6]. Our derivation suggests that the effectiveness of learning could be improved by tuning this dependence of synaptic change on postsynaptic activity to the activation function of postsynaptic neurons, according to eq. (B.5). It would be interesting to check whether such tuning occurs in real neurons.

B.2 Effect of different output functions

In the main text, we assumed a linear mapping between student activities and motor output. Moreover, we assumed a myotopic organization, in which each student neuron projected to a single muscle, leading to a student–output assignment matrix M_{aj} in which each column had a single non-zero entry. We also assumed that student neurons only contributed additively to the outputs, with no inhibitory activity. Here we show that our results hold for other choices of student–output mappings.

For example, assume a push-pull architecture, in which half of the student neurons controlling one output are excitatory and half are inhibitory. This can be used to decouple the overall firing rate in the student from the magnitude of the outputs. Learning works just as effectively as in the case of the purely additive student–output mapping when using matched tutors, Figs. B.1A and B.1B. The consequences of mismatching student and tutor circuits are also not significantly changed, Figs. B.1C and B.1D.

We can also consider nonlinear mappings between the student activity and the final output. If there is a monotonic output nonlinearity, as in eq. (B.1) with $N'_a > 0$, the tutor signal derived for the linear case, eq. (2), can still achieve convergence, though at a slower rate and with a somewhat lower accuracy (see Fig. B.1E for the case of a sigmoidal nonlinearity). For non-monotonic nonlinearities, the direction from which the optimum is approached can be crucial, as learning can get stuck in local minima of the loss function.[1] Studying this might provide an interesting avenue to test whether learning in songbirds is based on a gradient descent-type rule or on a more sophisticated optimization technique.

B.3 Different inhibition models

In the spiking model, we used an activity-dependent inhibitory signal that was proportional to the average student activity. Using a constant inhibition instead, $V_{\text{inh}} = \text{constant}$, does not significantly change the results: see Fig. B.1F for an example.

B.4 Effect of changing plasticity kernels

In the main text, we used exponential kernels with $\tau_1 = 80\,\text{ms}$ and $\tau_2 = 40\,\text{ms}$ for the smoothing of the conductor signal that enters the synaptic plasticity rule, eq. (A.3). We can generalize this in two ways: we can use different timescales τ_1, τ_2, or we can use a different functional form for the kernels. (Note that in the main text we showed the effects of varying the parameters α and β in the plasticity rule, while the timescales τ_1 and τ_2 were kept fixed.)

The values for the timescales $\tau_{1,2}$ were chosen to roughly match the shape of the plasticity curve measured in slices of zebra finch RA [26] (see Figs. 1C, 1D). The main predictions of our model, that learning is most effective when the tutor signal is matched to the student plasticity rule, and that large mismatches between tutor and student lead to impaired learning, hold well when the student timescales change: see Fig. B.2A for the case when $\tau_1 = 20\,\text{ms}$ and $\tau_2 = 10\,\text{ms}$. In the main text we saw that the negative effects of tutor–student mismatch diminish for timescales that are shorter than $\sim \tau_{1,2}$. In Fig. B.2A, the range of timescales where a precise matching is not essential becomes very small because the student timescales are short.

Another generalization of our plasticity rule can be obtained by changing the functional form of the kernels used to smooth the conductor input. As an example, suppose K_2 is kept exponential, while K_1 is replaced by

$$\bar{K}_1(t) = \begin{cases} \frac{1}{\bar{\tau}_1^2}\, te^{-t/\bar{\tau}_1} & \text{for } t \geq 0, \\ 0 & \text{else.} \end{cases} \tag{B.6}$$

An example of learning using an STDP rule based on kernels $\bar{K}_1$ and K_2 where $\bar{\tau}_1 = \tau_2$ is shown in Fig. B.2B. The matching tutor has the same form as before, eq. (2) with timescale τ_{tutor} given by eq. (1), but with $\tau_1 = 2\bar{\tau}_1 = 2\tau_2$. We can see that learning is as effective as in the case of purely exponential kernels.

B.5 More general conductor patterns

In the main text, we have focused on a conductor whose activity matches that observed in area HVC of songbirds [14]: each neuron fires a single burst during the motor program. Our model, however, is not restricted to this case. We generated alternative conductor patterns by using arbitrarily-placed bursts of activity, as in Fig. B.3A. The model converges to a good rendition of the target program, Fig. B.3B. Learning is harder in this case because many conductor neurons can be active at the same time, and the weight updates affect not only the output of the system at the current position in the motor program, but also at all the other positions where the conductor neurons fire. This is in contrast to the HVC-like conductor, where each neuron fires at a single point in the motor program, and thus the effect of weight updates is better localized. More

[1] We thank Josh Gold for this observation.

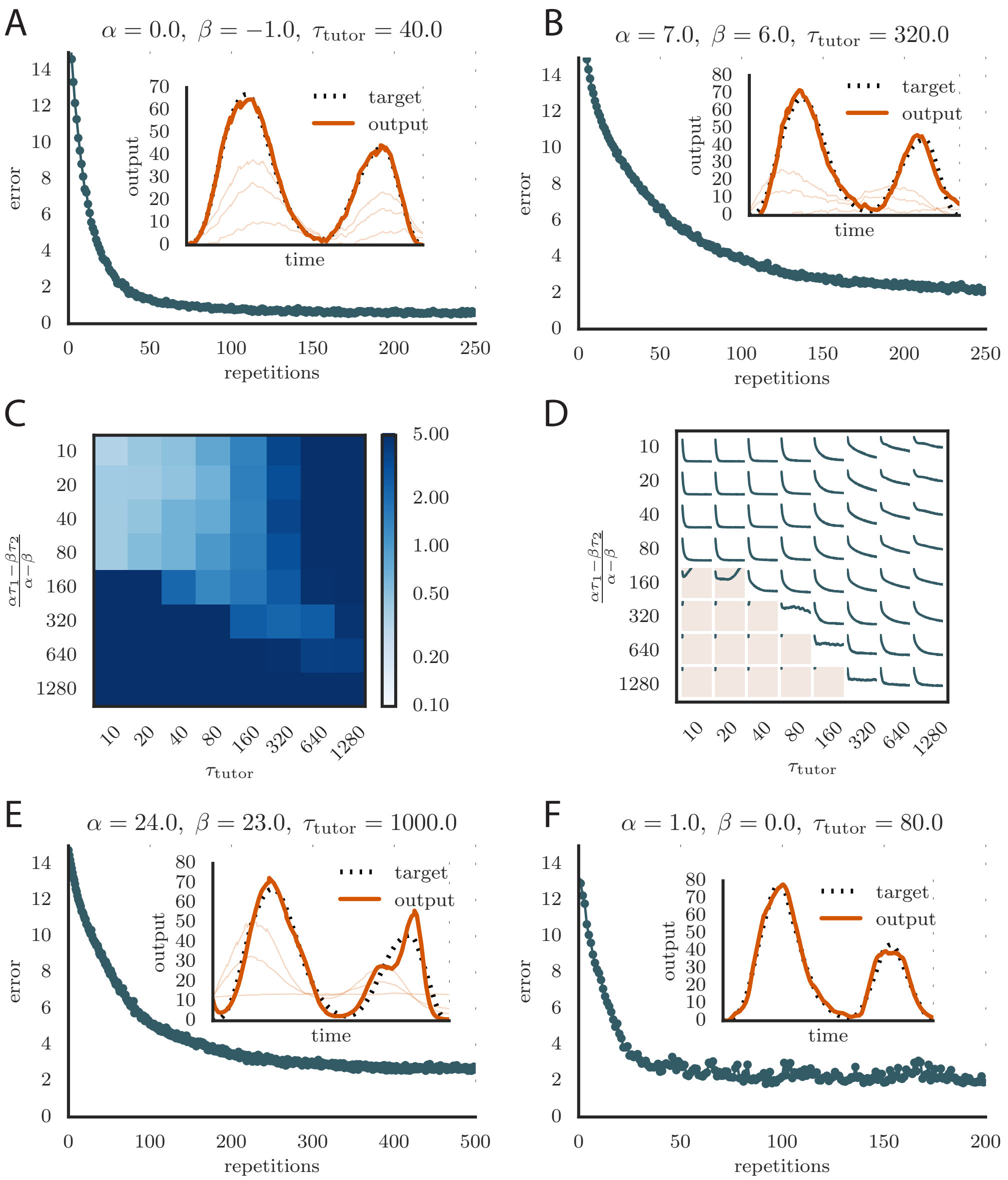

Figure B.1: Robustness of learning **A.** Error trace showing how average motor error evolves with repetitions of the motor program for rate-based plasticity paired with a matching tutor, when the student–output mapping has a push-pull architecture. The inset shows the final motor output (thick red line) compared to the target output (dotted black line). The output on the first rendition and at two other stages of learning are also shown. **B.** The error trace and final motor output shown for timing-based plasticity matched by a tutor with a long integration timescale. **C.** Effects of mismatch between student and tutor on reproduction accuracy when using a push-pull architecture for the student–output mapping. The heatmap shows the final reproduction error of the motor output after 250 learning cycles when a student with plasticity parameters α and β is paired with a tutor with memory timescale τ_{tutor}. Here $\tau_1 = 80\,\text{ms}$ and $\tau_2 = 40\,\text{ms}$. **D.** Error evolution curves as a function of the mismatch between student and tutor. Each plot shows how the error in the motor program changes during 250 learning cycles for the same conditions as those shown in the heatmap. The region shaded in light pink shows simulations where the mismatch between student and tutor leads to a deteriorating instead of improving performance during learning. **E.** Convergence in the rate-based model with a linear-nonlinear controller that uses a sigmoidal nonlinearity. **F.** Convergence in the spiking model when inhibition is constant instead of activity-dependent ($V_{\text{inh}} = \text{constant}$).

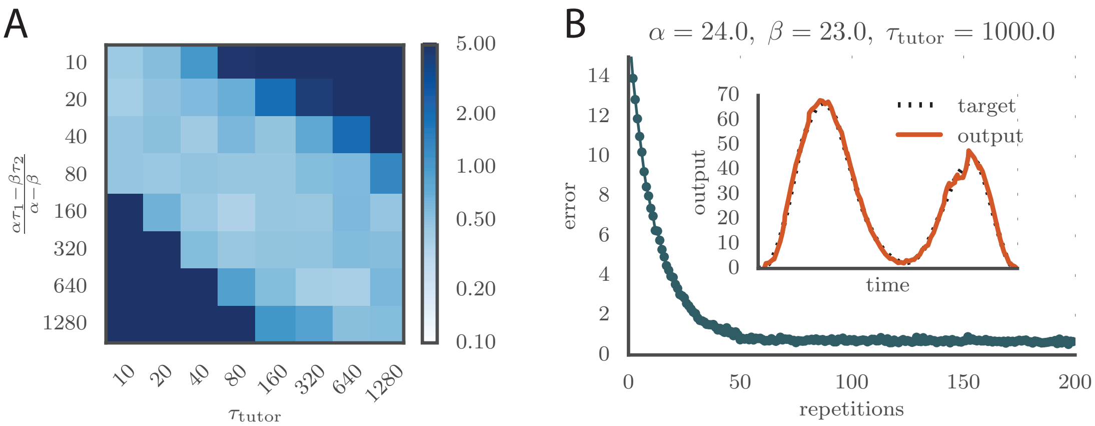

Figure B.2: Effect of changing conductor smoothing kernels in the plasticity rule. (**A**) Matrix showing learning accuracy when using different timescales for the student plasticity rule. Each entry in the heatmap shows the average rendition error after 250 learning steps when pairing a tutor with timescale τ_{tutor} with a non-matched student. Here the kernels are exponential, with timescales $\tau_1 = 20\,\text{ms}$, $\tau_2 = 10\,\text{ms}$. (**B**) Evolution of motor error with learning using kernels $\sim e^{-t/\tau}$ and $\sim te^{-t/\tau}$, instead of the two exponentials used in the main text. The tutor signal is as before, eq. (2). The inset shows the final output for the trained model, for one of the two output channels. Learning is as effective and fast as before.

generally, simulations show that the sparser the conductor firing, the faster the convergence (data not shown). The accuracy of the final rendition of the motor program (Fig. B.3B, inset) is also not as good as before.

B.6 Edge effects

In our derivation of the matching tutor rule, we assumed that the system has enough time to integrate all the synaptic weight changes from eq. (A.3). However, some of these changes occur tens or hundreds of milliseconds after the inputs that generated them, due to the timescales used in the plasticity kernel. Since our simulations are only run for a finite amount of time, there will in general be edge effects, where periods of the motor program towards the end of the simulations will have difficulty converging. To offset such numerical issues, we ran the simulations for a few hundred milliseconds longer than the duration of the motor program, and ignored the data from this extra period. Our simulations typically run for 600 ms, and the time reserved for relaxation after the end of the program was set to 400 ms. The long relaxation time was chosen to allow for cases where the tutor was chosen to have a very long memory timescale.

B.7 Parameter optimization for reproducing juvenile and adult spiking statistics

We set the parameters in our simulations to reproduce spiking statistics from recordings in zebra finch RA as closely as possible. Fig. B.4 shows how the distribution of summary statistics obtained from 50 runs of the simulation compares to the distributions calculated from recordings in birds at various developmental stages. Each plot shows a standard box and whisker plot superimposed over a kernel-density estimate of the distribution of a given summary statistic, either over simulation runs or over recordings from birds at various stages of song learning. We ran two sets of simulations, one for a bird with juvenile-like connectivity between HVC and RA, and one with adult-like connectivity (see Methods). In these simulations there was no learning to match the timecourse of songs—the goal was simply to identify parameters that lead to bridsong-like firing statistics.

The qualitative match between our simulations and recordings is good, but the simulations are less variable than the measurements. This may be due to sources of variability that we have ignored—for example, all our simulated neurons had exactly the same membrane time constants, refractory periods, and threshold

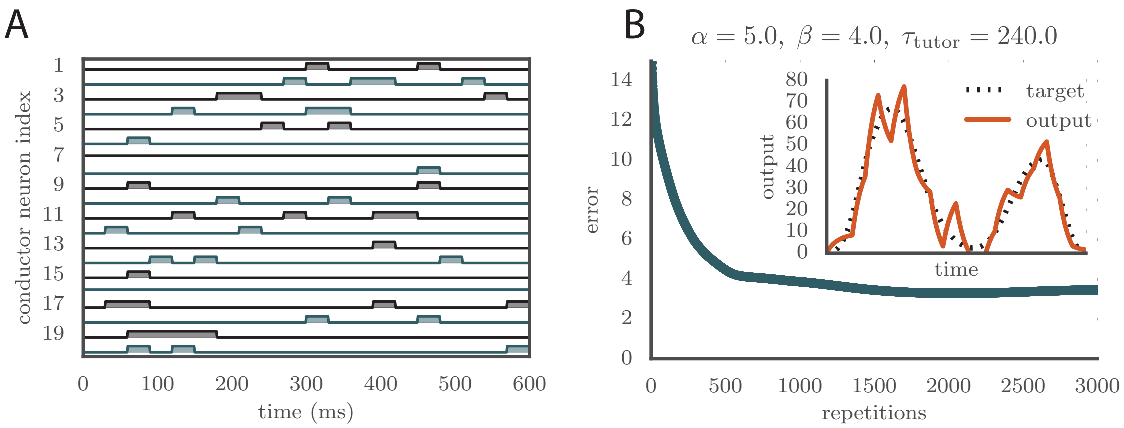

Figure B.3: Learning with arbitrary conductor activity. **A.** Typical activity of conductor neurons. 20 of the 100 neurons included in the simulation are shown. The activity pattern is chosen so that about 10% of the neurons are active at any given time. The pattern is chosen randomly but is fixed during learning. Each conductor burst lasts 30 ms. **B.** Convergence curve and final rendition of the motor program (in inset). Learning included two output channels but the final output is shown for only one of them.

potentials, which is not the case for real neurons. Another reason might be that in our simulations, all the runs were performed for the same network, while the measurements are from different cells in different birds.

References

[1] Farhan Ali et al. "The basal ganglia is necessary for learning spectral, but not temporal, features of birdsong". In: *Neuron* 80.2 (2013), pp. 494–506.

[2] Aaron S. Andalman and Michale S. Fee. "A basal ganglia-forebrain circuit in the songbird biases motor output to avoid vocal errors." In: *Proceedings of the National Academy of Sciences of the United States of America* 106.30 (2009), pp. 12518–23.

[3] Mark J. Basista et al. "Independent Premotor Encoding of the Sequence and Structure of Birdsong in Avian Cortex". In: *Journal of Neuroscience* 34.50 (2014), pp. 16821–16834.

[4] R. A. Canady et al. "Effect of testosterone on input received by an identified neuron type of the canary song system: a Golgi/electron microscopy/degeneration study". In: *The Journal of Neuroscience* 8.10 (1988), pp. 3770–3784.

[5] Marina Chistiakova and Maxim Volgushev. "Heterosynaptic plasticity in the neocortex". In: *Experimental Brain Research* 199.3-4 (2009), pp. 377–390.

[6] Marina Chistiakova et al. "Heterosynaptic Plasticity: Multiple Mechanisms and Multiple Roles". In: *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry* April (2014).

[7] Kenji Doya and Terrence J. Sejnowski. "A computational model of avian song learning". In: *The new cognitive neurosciences.* Ed. by Michael S. Gazzaniga. The MIT Press, 2000, pp. 469–482. arXiv: `arXiv:1011.1669v3`.

[8] Michael A. Farries and Adrienne L. Fairhall. "Reinforcement Learning With Modulated Spike Timing-Dependent Synaptic Plasticity". In: *Journal of neurophysiology* 98.6 (2007), pp. 3648–3665.

[9] Michale S. Fee and Jesse H. Goldberg. "A hypothesis for basal ganglia-dependent reinforcement learning in the songbird". In: *Neuroscience* 198 (2011), pp. 152–170.

[10] Ila R. Fiete, Michale S. Fee, and H. Sebastian Seung. "Model of birdsong learning based on gradient estimation by dynamic perturbation of neural conductances". In: *Journal of neurophysiology* 98.4 (2007), pp. 2038–57.

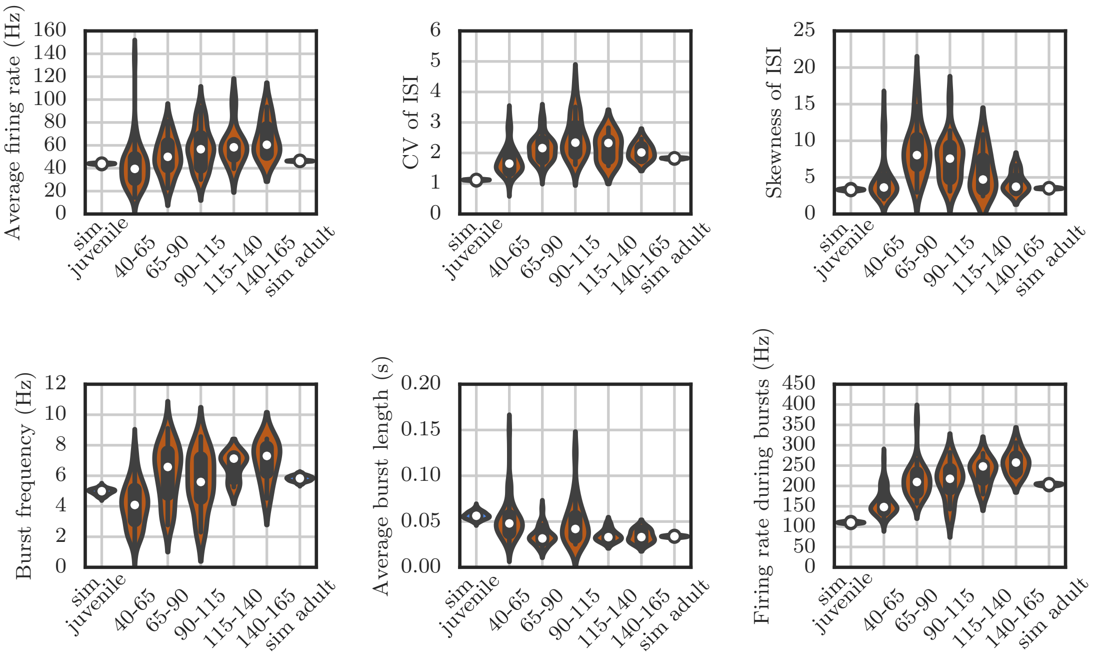

Figure B.4: Violin plots showing how the spiking statistics from our simulation compared to the statistics obtained from neural recordings. Each violin shows a kernel-density estimate of the distribution that a particular summary statistic had in either several runs of a simulation, or in several recordings from behaving birds. The circle and the box within each violin show the median and the interquartile range

[11] Ila R. Fiete and H. Sebastian Seung. “Gradient learning in spiking neural networks by dynamic perturbation of conductances”. In: *Physical Review Letters* 97.4 (2006), p. 048104. arXiv: `0601028v1 [arXiv:q-bio]`.

[12] Paul W. Frankland and Bruno Bontempi. “The organization of recent and remote memories”. In: *Nature reviews. Neuroscience* 6.2 (2005), pp. 119–30.

[13] Jonathan Garst-Orozco, Baktash Babadi, and Bence P. Ölveczky. “A neural circuit mechanism for regulating motor variability during skill learning”. In: *eLife* 3 (2014), e03697.

[14] Richard H. R. Hahnloser, Alexay A. Kozhevnikov, and Michale S. Fee. “An ultra-sparse code underlies the generation of neural sequences in a songbird”. In: *Nature* 419.1989 (2002), pp. 796–797.

[15] Kosuke Hamaguchi, Masashi Tanaka, and Richard Mooney. “A Distributed Recurrent Network Contributes to Temporally Precise Vocalizations”. In: *Neuron* 91 (2016), pp. 680–693.

[16] Nikolaus Hansen. “The CMA Evolution Strategy: A Comparing Review”. In: *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*. 2006, pp. 75–102.

[17] Kathrin Herrmann and Arthur P. Arnold. “The development of afferent projections to the robust archistriatal nucleus in male zebra finches: a quantitative electron microscopic study”. In: *The Journal of Neuroscience* 11.7 (1991), pp. 2063–2074.

[18] Lukas A. Hoffmann et al. “Dopaminergic Contributions to Vocal Learning”. In: *The Journal of Neuroscience* 36.7 (2016), pp. 2176–2189.

[19] K Immelmann. “Song development in the zebra finch and other estrildid finches”. In: *Bird vocalizations*. Ed. by R. A. Hinde. 1969, pp. 61–74.

[20] Mimi H. Kao, Brian D. Wright, and Allison J. Doupe. “Neurons in a Forebrain Nucleus Required for Vocal Plasticity Rapidly Switch between Precise Firing and Variable Bursting Depending on Social Context”. In: *The Journal of Neuroscience* 28.49 (2008), pp. 13232–13247.

[21] Risa Kawai et al. “Motor Cortex Is Required for Learning but Not for Executing a Motor Skill”. In: *Neuron* 86.3 (2015), pp. 800–812.

[22] Satoshi Kojima, Mimi H. Kao, and Allison J. Doupe. “Task-related ”cortical” bursting depends critically on basal ganglia input and is linked to vocal plasticity”. In: *PNAS* 110.12 (2013), pp. 4756–4761.

[23] L’ubica Kubikova and L’ubor Koštál. “Dopaminergic system in birdsong learning and maintenance”. In: *Journal of Chemical Neuroanatomy* 39 (2010), pp. 112–123.

[24] Anthony Leonardo and Michale S. Fee. “Ensemble Coding of Vocal Control in Birdsong”. In: *The Journal of Neuroscience* 25.3 (2005), pp. 652–661.

[25] Galen F. Lynch et al. “Rhythmic Continuous-Time Coding in the Songbird Analog of Vocal Motor Cortex”. In: *Neuron* 90.4 (2016), pp. 877–892.

[26] W. Hamish Mehaffey and Allison J. Doupe. “Naturalistic stimulation drives opposing heterosynaptic plasticity at two inputs to songbird cortex”. In: *Nature Neuroscience* 18.9 (2015), pp. 1272–1280.

[27] Raoul-Martin Memmesheimer et al. “Learning Precisely Timed Spikes”. In: *Neuron* 82 (2014), pp. 1–14.

[28] Bence P. Ölveczky, Aaron S. Andalman, and Michale S. Fee. “Vocal experimentation in the juvenile songbird requires a basal ganglia circuit.” In: *PLoS biology* 3.5 (2005), e153.

[29] Bence P. Ölveczky et al. “Changes in the neural control of a complex motor sequence during learning”. In: *Journal of Neurophysiology* 106.1 (2011), pp. 386–397.

[30] David J. Perkel. “Origin of the Anterior Forebrain Pathway”. In: *Annals of the New York Academy of Sciences* 1016 (2004), pp. 736–748.

[31] Michel A. Picardo et al. “Population-Level Representation of a Temporal Sequence Underlying Song Production in the Zebra Finch”. In: *Neuron* 90.4 (2016), pp. 866–876.

[32] H. Sebastian Seung. “Learning in Spiking Neural Networks by Reinforcement of Stochastic Synaptic Transmission”. In: 40 (2003), pp. 1063–1073.

[33] H. Blair Simpson and David S. Vicario. “Brain Pathways for Learned and Unlearned Vocalizations Differ in Zebra Finches”. In: *The Journal of Neuroscience* 10.5 (1990), pp. 1541–1556.

[34] John E. Spiro, Matthew B. Dalva, and Richard Mooney. “Long-Range Inhibition Within the Zebra Finch Song Nucleus RA Can Coordinate the Firing of Multiple Projection Neurons”. In: *Journal of neurophysiology* 81 (1999), pp. 3007–3020.

[35] Laura L. Stark and David J. Perkel. “Two-stage, input-specific synaptic maturation in a nucleus essential for vocal production in the zebra finch”. In: *The Journal of Neuroscience* 19.20 (1999), pp. 9107–9116.

[36] Robert S. Turner and Michel Desmurget. “Basal ganglia contributions to motor control: a vigorous tutor”. In: *Current Opinion in Neurobiology* 20.6 (2010), pp. 704–716.

[37] Timothy L. Warren et al. “Mechanisms and time course of vocal learning and consolidation in the adult songbird”. In: *Journal of Neurophysiology* 106 (2011), pp. 1806–1821.

[38] Albert C. Yu and Daniel Margoliash. “Temporal hierarchical control of singing in birds”. In: *Science* 273.5283 (1996), pp. 1871–1875.