

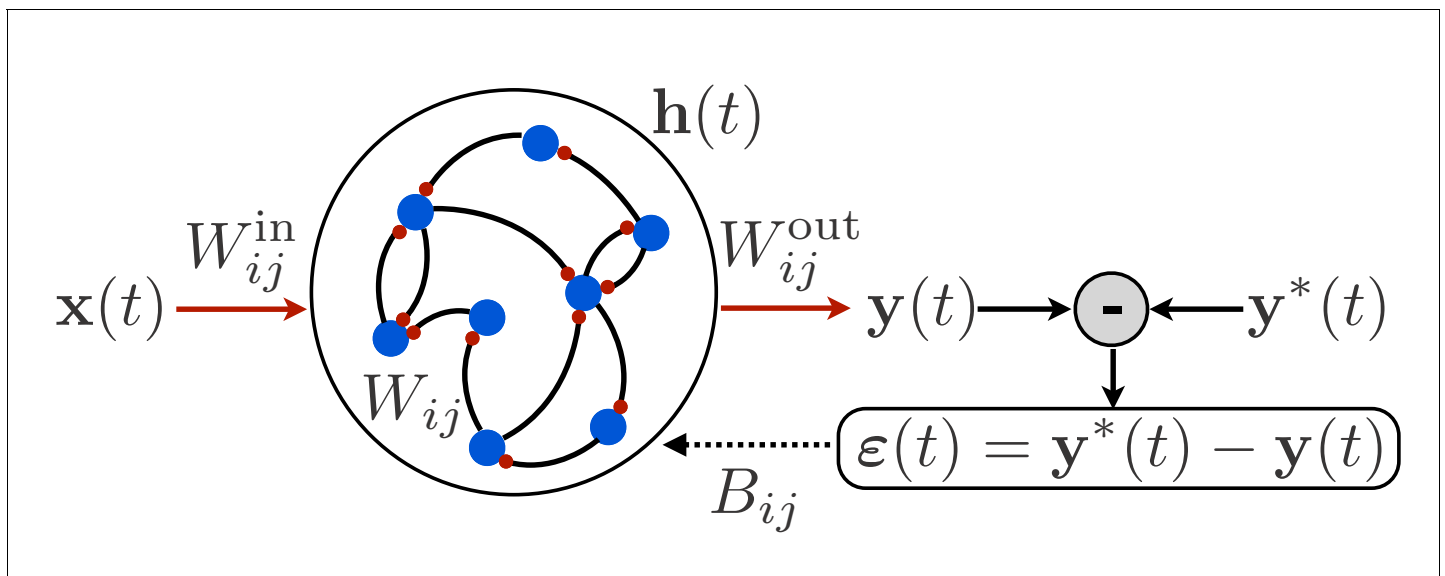


---

## Figures and figure supplements

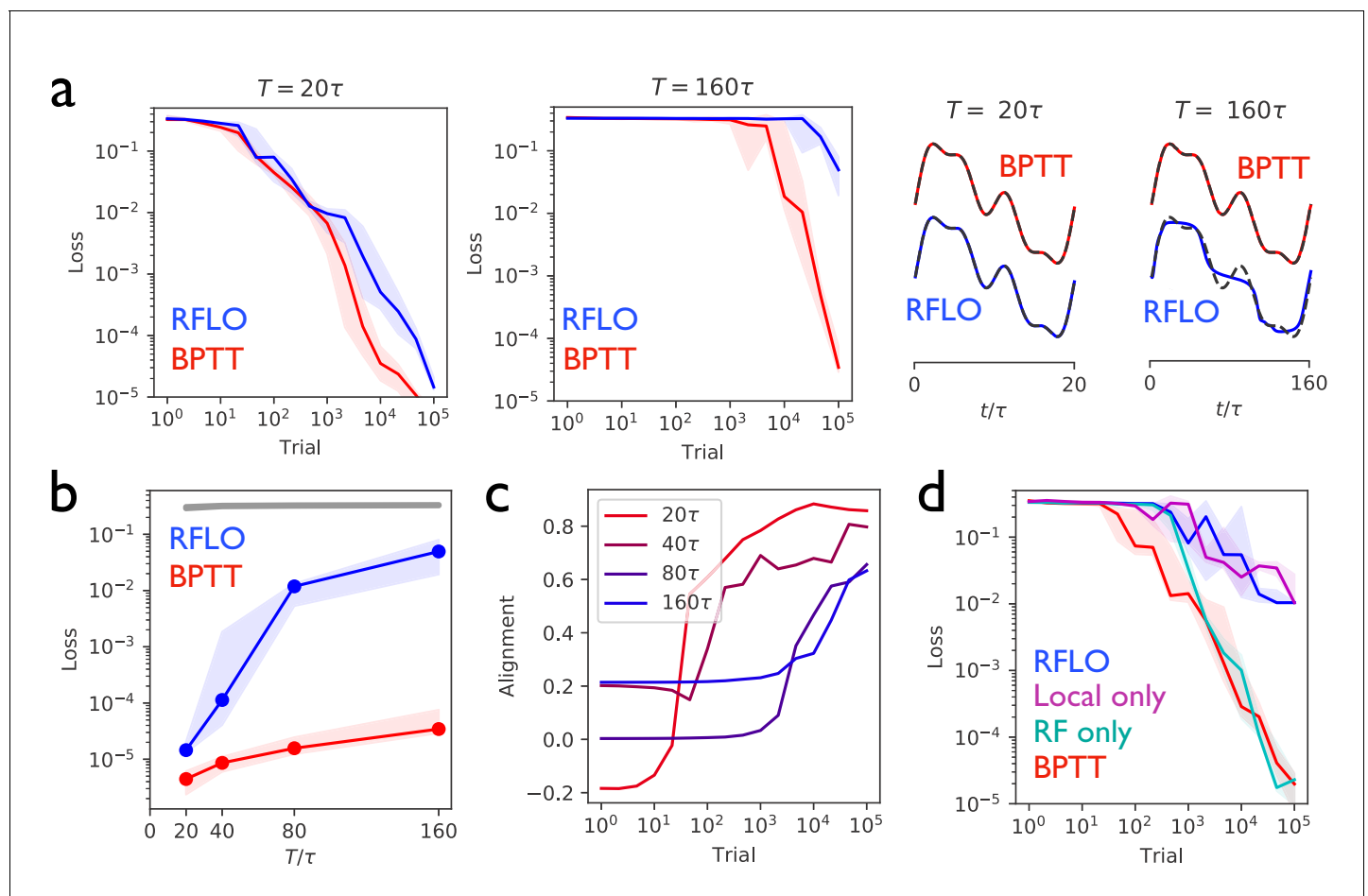
Local online learning in recurrent networks with random feedback

**James M Murray**



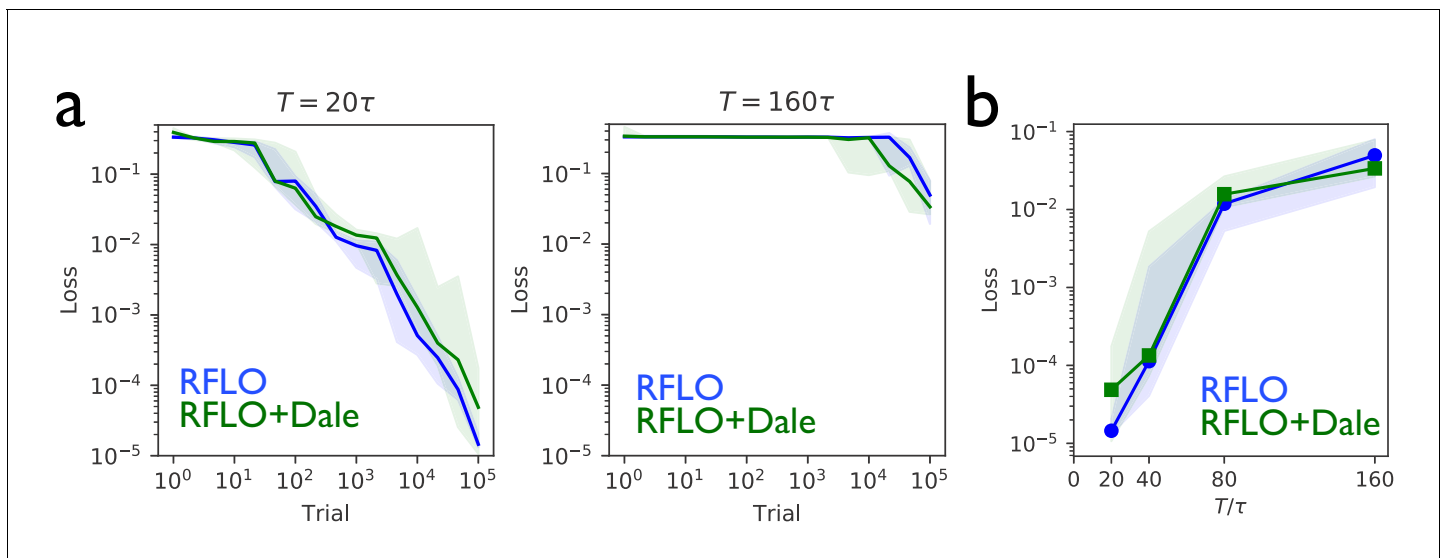
**Figure 1.** Schematic illustration of a recurrent neural network. The network receives time-dependent input  $\mathbf{x}(t)$ , and its synaptic weights are trained so that the output  $\mathbf{y}(t)$  matches a target function  $\mathbf{y}^*(t)$ . The projection of the error  $\epsilon(t)$  with feedback weights is used for learning the input weights and recurrent weights.

DOI: <https://doi.org/10.7554/eLife.43299.002>



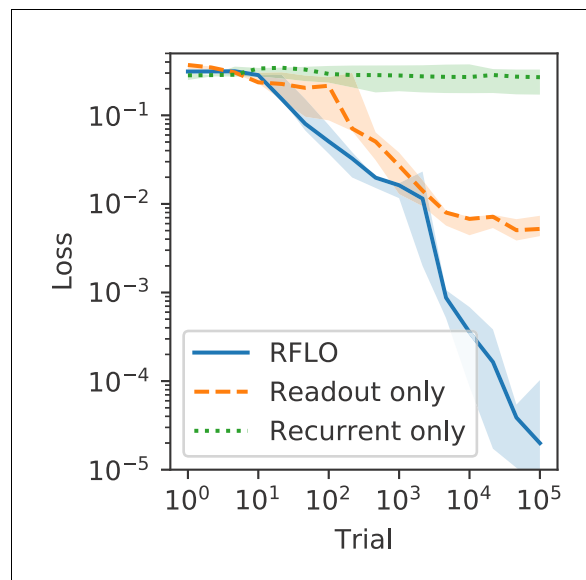
**Figure 2.** Periodic output task. (a) *Left panels:* The mean squared output error during training for an RNN with  $N = 30$  recurrent units and no external input, trained to produce a one-dimensional periodic output with period of duration  $T = 20\tau$  (left) or  $T = 160\tau$  (right), where  $\tau = 10$  is the RNN time constant. The learning rules used for training were backpropagation through time (BPTT) and random feedback local online (RFLO) learning. Solid line is median loss over nine realizations, and shaded regions show 25/75 percentiles. *Right panels:* The RNN output at the end of training for each type of learning (dashed lines are target outputs, offset for clarity). The colored lines correspond to the two learning rules from (a), while the gray line is the loss computed for an untrained RNN. (b) The loss function at the end of training for target outputs having different periods. The colored lines correspond to the two learning rules from (a), while the gray line is the loss computed for an untrained RNN. (c) The normalized alignment between the vector of readout weights  $\mathbf{W}^{\text{out}}$  and the vector of feedback weights  $\mathbf{B}$  during training with RFLO learning. (d) The loss function during training with  $T = 80\tau$  for BPTT and RFLO, as well as versions of RFLO in which locality is enforced without random feedback (magenta) or random feedback is used without enforcing locality (cyan).

DOI: <https://doi.org/10.7554/eLife.43299.003>



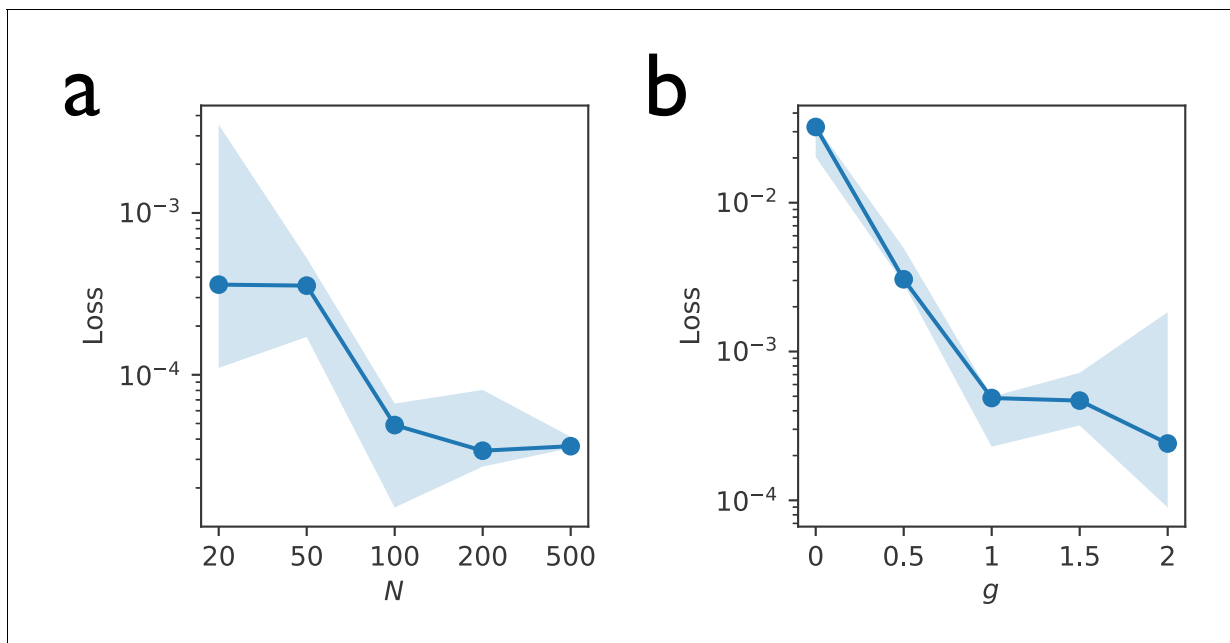
**Figure 2—figure supplement 1.** An RNN with sign-constrained synapses comporting with Dale's law attains performance similar to an unconstrained RNN. (a) In the periodic output task from **Figure 2**, the loss function during training with RFLO learning shows similar rate of decrease in an RNN with sign-constrained synapses (RFLO + Dale) compared with an RNN trained without sign constraint (RFLO), both for short-duration (left) and long-duration (right) outputs. (b) The final loss function value after training is similar for RNNs with and without sign-constrained synapses for outputs of various durations.

DOI: <https://doi.org/10.7554/eLife.43299.004>



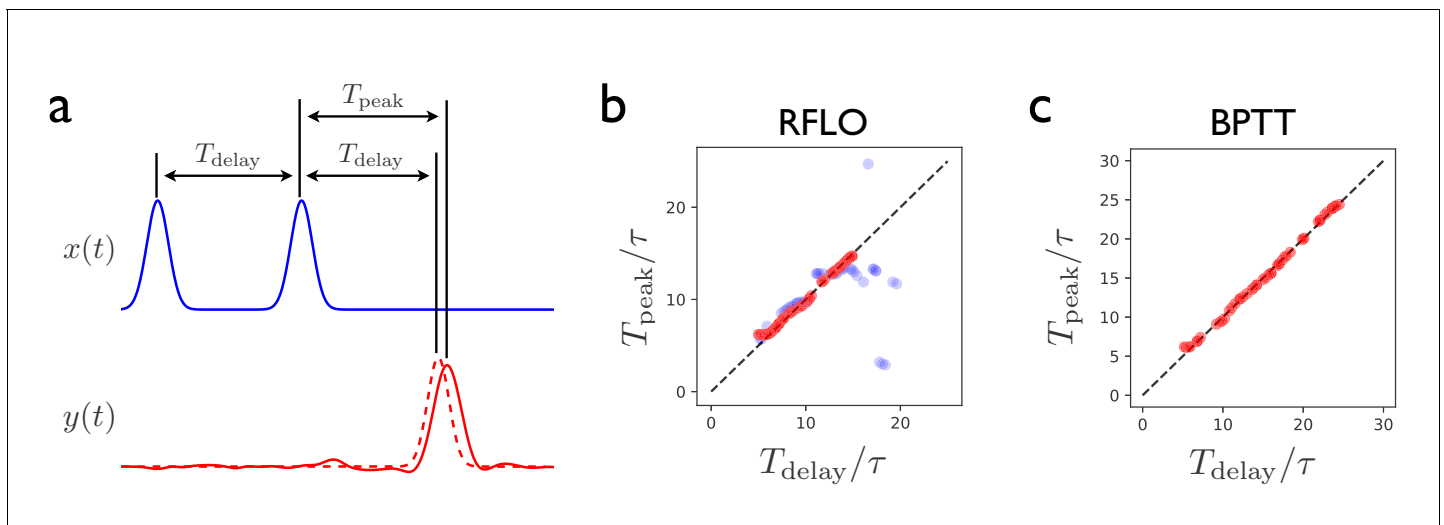
**Figure 2—figure supplement 2.** An RNN trained to perform the task from **Figure 2** with RFLO learning on recurrent and readout weights outperforms an RNN in which only readout weights or only recurrent weights are trained.

DOI: <https://doi.org/10.7554/eLife.43299.005>



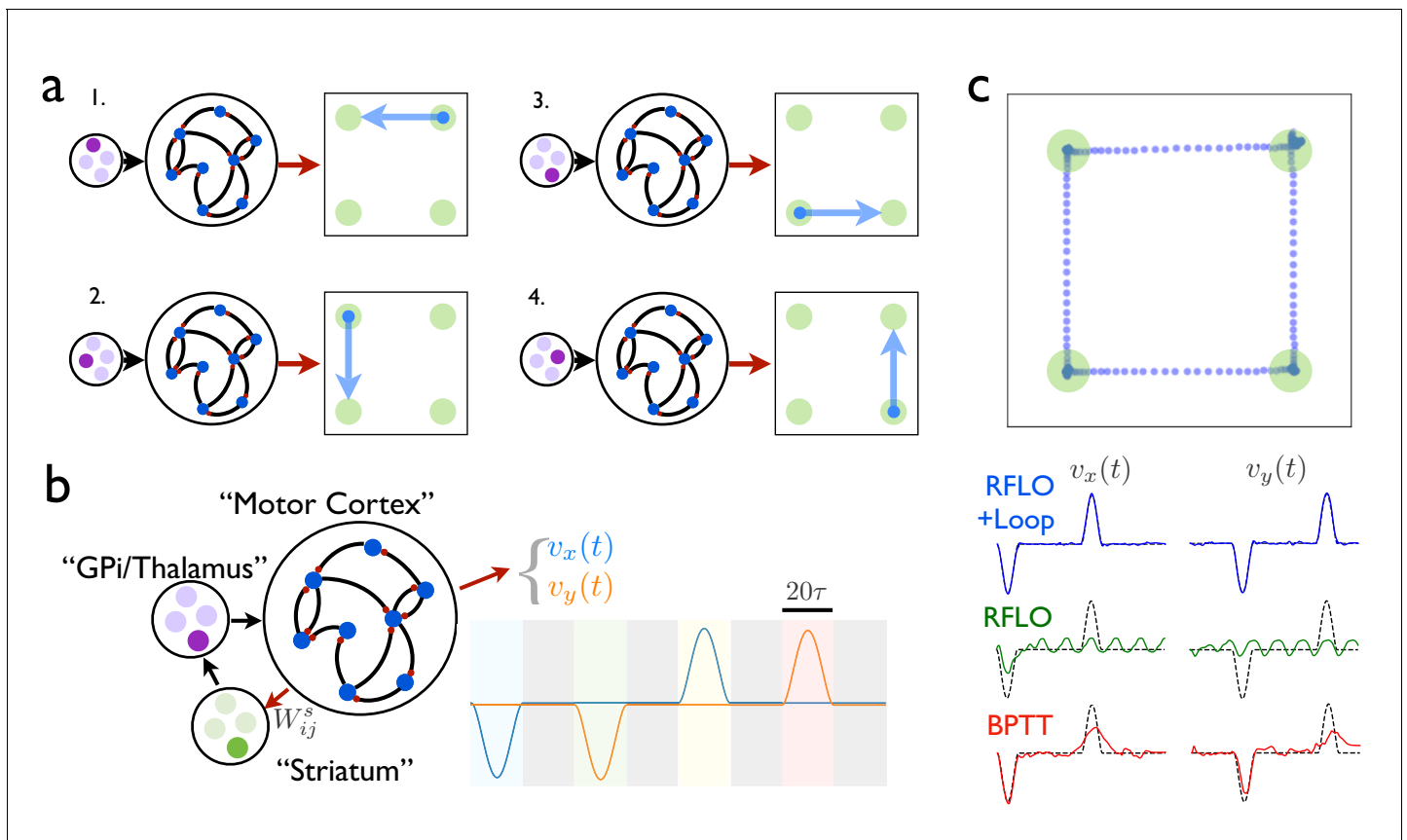
**Figure 2—figure supplement 3.** The performance of an RNN trained to perform the task from **Figure 2** with RFLO learning improves with larger network sizes and larger initial recurrent weights. (a) The loss after  $10^4$  trials in RNNs versus the number of recurrent units. (b) The loss after  $10^4$  trials in RNNs versus the standard deviation of the initial weights. In these RNNs, the recurrent weights were initialized as  $W_{ij} \sim \mathcal{N}(0, g^2/N)$ , and the readout weights were initialized as  $W_{ij}^{\text{out}} \sim \mathcal{U}(-1, 1)g/\sqrt{N}$ , where  $\mathcal{U}(-1, 1)$  is the uniform distribution over  $(-1, 1)$ .

DOI: <https://doi.org/10.7554/eLife.43299.006>



**Figure 3.** Interval-matching task. (a) In the task, the RNN input consists of two input pulses, with a random delay  $T_{\text{delay}}$  between pulses in each trial. The target output (dashed line) is a pulse trailing the second input pulse by  $T_{\text{delay}}$ . (b) The time of the peak in the RNN output is observed after training with RFLO learning and testing in trials with various interpulse delays in the input. Red (blue) shows the case in which the RNN is trained with interpulse delays satisfying  $T_{\text{delay}} \leq 15\tau$  ( $20\tau$ ). (c) Same as (b), but with the RNN trained using BPTT using interpulse delays  $T_{\text{delay}} \leq 25\tau$  for training and testing.

DOI: <https://doi.org/10.7554/eLife.43299.007>



**Figure 4.** An RNN with multiple inputs controlled by an auxiliary loop learns to produce sequences. (a) An RNN with a two-dimensional readout controlling the velocity of a cursor is trained to move the cursor in a different direction for each of the four possible inputs. (b) The RNN is augmented with a loop structure, which allows a readout from the RNN via learned weights  $\mathbf{W}^s$  to change the state of the input to the RNN, enabling the RNN state at the end of each cursor movement to trigger the beginning of the next movement. (c) The trajectory of a cursor performing four movements and four holds, where RFLO learning was used to train the individual movements as in (a), and learning of the weights  $\mathbf{W}^s$  was used to join these movements into a sequence, as illustrated in (b). Lower traces show comparison of this trajectory with those obtained by using either RFLO or BPTT to train an RNN to perform the entire sequence without the auxiliary loop.

DOI: <https://doi.org/10.7554/eLife.43299.008>