Reinforcement learning of 2-joint virtual arm reaching in a computer model of sensorimotor cortex

Samuel A. Neymotin[1, 2], **George L. Chadderdon**[2], **Cliff C. Kerr**[2, 3], **Joseph T. Francis**[2, 4 − 6], **William W. Lytton**[2, 4 − 8]

[1]Dept. Neurobiology, Yale University School of Medicine, New Haven, CT.

[2]Dept. Physiology & Pharmacology, SUNY Downstate, Brooklyn, NY.

[3]School of Physics, University of Sydney, Sydney, Australia.

[4]The Robert F. Furchgott Center for Neural and Behavioral Science, SUNY Downstate, Brooklyn, NY.

[5]Joint Program in Biomedical Engineering, NYU-Poly and SUNY Downstate, Brooklyn, NY.

[6]Program in Neural and Behavioral Science, SUNY Downstate, Brooklyn, NY.

[7]Dept. Neurology, SUNY Downstate, Brooklyn, NY.

[8]Dept. Neurology, Kings County Hospital, Brooklyn, NY.

Keywords: sensorimotor cortex, reinforcement learning, computer model.

Abstract

Neocortical mechanisms of learning sensorimotor control involve a complex series of interactions at multiple levels, from synaptic mechanisms to cellular dynamics to network connectomics. We developed a model of sensory and motor neocortex consisting of 704 spiking model-neurons. Sensory and motor populations included excitatory cells and two types of interneurons. Neurons were interconnected with AMPA/NMDA, and $GABA_A$ synapses. We trained our model using spike-timing-dependent reinforcement learning to control a 2-joint virtual arm to reach to a fixed target. For each of 125 trained networks, we used 200 training sessions, each involving 15 s reaches to the target from 16 starting positions. Learning altered network dynamics, with enhancements to neuronal synchrony and behaviorally-relevant information flow between neurons. After learning, networks demonstrated retention of behaviorally-relevant memories by utilizing proprioceptive information to perform reach-to-target from multiple starting positions. Networks dynamically controlled which joint rotations to utilize to reach a target, depending on current arm position. Learning-dependent network reorganization was evident in both sensory and motor populations – learned synaptic weights showed target-specific patterning optimized for particular reach movements. Our model embodies an integrative hypothesis of sensorimotor cortical learning which could be used to interpret future electrophysiological data recorded *in vivo* from sensorimotor learning experiments. We used our model to make the following predictions: learning en-

hances synchrony in neuronal populations and behaviorally-relevant information flow across neuronal populations; enhanced sensory processing aids task-relevant motor performance; the relative ease of a particular movement *in vivo* depends on the amount of sensory information required to complete the movement.

1 Introduction

Adaptive movements in response to stimuli sensed from the world are a vital biological function. Although arm reaching towards a target is a basic movement, the neocortical mechanisms allowing sensory information to be used in the generation of reaches are enormously complex and difficult to track (Shadmehr and Wise, 2005). Learning brings neuronal and physical dynamics together. In studies of birdsong, it has been demonstrated that reinforcement learning (RL) operates on random babbling (Sober and Brainard, 2009). In that setting, initially random movements initiated by motor neocortex may be rewarded or punished via an error signal affecting neuromodulatory control of plasticity via dopamine (Kubikova and Kostál, 2010). In primates, frontal cortex, including primary motor area M1, is innervated by dopaminergic projections from the ventral tegmental area (Luft and Schwarz, 2009; Molina-Luna et al., 2009; Hosp et al., 2011), and recent neurophysiological evidence points to reward modulation of M1 activity (Marsh et al., 2011). It has been suggested that similar babble/RL mechanisms may play a role in limb target-learning.

Many brain areas are involved in motor learning, likely including spinal cord, red nucleus, and thalamus, as well as the more well-characterized basal ganglia, cerebel-

lum, and neocortex (Sanes, 2003). In addition to individual brain areas, connections between areas are likely vital (Graybiel et al., 1994; Hikosaka et al., 2002). Learning in these different areas will likely play different roles for different types of tasks and at different times in development. Neonates can already perform directed reaching movements at birth, and learn to reach a target within 15 weeks, using proprioceptive and visual feedback (Berthier et al., 1999; von Hofsten, 1979). This process has been suggested to be primarily cortical (Berthier, 2011). Sensorimotor integration of reaching is learned through analysis of mismatches of perception and desired actions (Corbetta and Snapp-Childs, 2009). Similarly, adult learning of complex tasks, such as serving in tennis, utilizes the neocortical substrate at different stages of the learning process (Sanes, 2003).

Computational modeling of biologically-realistic neuronal networks can aid in validating theories of motor learning and predicting how it occurs *in vivo* (Houk and Wise, 1995). Recently, learning models of spiking neurons using a goal-driven or reinforcement learning signal have been developed (Farries and Fairhall, 2007; Florian, 2007; Izhikevich, 2007; Potjans et al., 2009; Seung, 2003), many using spike-timing-dependent plasticity (Roberts and Bell, 2002; Rowan and Neymotin, 2013; Song et al., 2000; Neymotin et al., 2011b). Here, we present a simplified sensorimotor cortex network with an input sensory area (S1) that processes inputs from muscles, and an output area, representing primary motor cortex (M1), that projects to muscles of a virtual arm.

The present paper extends our previous efforts to create a spiking neuronal model of cortical reinforcement learning of arm reaching (Chadderdon et al., 2012). In that paper, we demonstrated the feasibility of the dopamine system-inspired value-driven learning

algorithm used in this paper in allowing a swiveling forearm segment controller to learn a mapping from proprioceptive state to flexion and extension motor commands needed to direct the virtual hand to the target: a task requiring only one degree-of-freedom motion. Here, we extend the scope of the task to two degrees of freedom, permitting the hand to explore a more complete virtual 2D workspace. This is a more demanding and complex task because shoulder and elbow angle changes have the potential to interfere with each other in adjusting the hand-to-target error, and the proprioceptive-to-motor command mapping to be learned requires conjunction of the information of the two different joints. We also increased the number of synaptic connections which have active plasticity, which adds further challenges, as well as flexibility, for the learning method. In addition, we enhanced the robustness of the training and testing paradigm: we first trained the system, then turned off further learning, and only then quantified reach-to-target performance. Turning off further learning ensured that what was previously learned and the ongoing effects of the learning algorithm were isolated. Even with the added complexity of a 2D reaching task, and the new testing paradigm, the model was still able to learn the new reaching task. Analysis comparing naive and trained network dynamics showed a distinct increase of synchrony and task-relevant information flow, as measured by coefficient of variation and normalized transfer entropy, respectively. These results have predictive power and may allow for better future understanding of electrophysiological data recorded *in vivo* from sensorimotor learning experiments.

2 Methods

System overview

The entire closed-loop learning system architecture is shown in Fig. 1. A Brain and its interaction with an artificial Environment were modeled, with the Environment containing a Virtual Arm, which was a part of the simulated agent's body, and a Target object which the agent was supposed to reach for. The Virtual Arm possessed two segments (upper-arm and forearm) which could be swiveled through two joints (shoulder and elbow) so that the arm and hand were able to move in a planar space. Each of the arm joints possessed a pair of flexor and extensor muscles for increasing / decreasing the angles, respectively, and which output a "stretch receptor" signal to the degree that the muscle was contracted.

An Actor system consisting of proprioceptive sensory neurons (P), sensory cells (S), and motor cells (M) was used to control this system. The P cell receptive fields were tuned so that individual cells fired for a narrow range of particular "muscle stretches" for one of the four muscles. These P cells sent fixed random weights to the S cells, so that the S cells were capable of representing the conjunct of positions in both joints, though this feature was not optimally hard-wired for these cells. The S cells, then, sent plastic weights to the M cells, which possessed a separate population of cells for each of the four muscles capable of stimulating contraction to the degree the corresponding subpopulations were active. Plasticity was present within the S and M unit populations and between them in both directions. This Actor effectively performed a mapping between limb state, as measured by muscle stretch, and a set of commands for driving

each muscle. The extensor population activity was subtracted from the flexor activity for a particular joint (shoulder, elbow) to yield a joint angle rotation command for the virtual arm.

It should be noted that the Actor in this system was learning to make a "blind" reach for a single learned target (proprioceptive-to-motor-command mapping). The Critic component of the system, however, possessed a means of calculating the visual difference between the hand's location and the target (Error Evaluation) and determining from the last two viewed hand coordinates whether the hand was getting closer or farther away from the target. Based on which was the case, the Critic sent a global reward/punisher signal to the Actor. Plastic synapses kept eligibility traces which allow credit/blame assignment. Rewards caused a global increase in the tagged weights, and punishers caused a decrease, effectively implementing Thorndike's Law of Effect in the system (Thorndike, 1911), i.e., allowing rewarded behaviors to be "stamped in" and punished behaviors to be "stamped out."

As a result of this arrangement, even though the Actor did not possess vision, it was possible in theory for it to learn a mapping driving the hand towards a visual located target, provided that target was not moved after training. In an ideal learning case by this system, the limb configuration corresponding with the target's location would learn to not move in either direction, but an over-flexed arm would learn to extend and an over-extended arm would learn to flex, so that the Actor had learned an attractor for the remembered target stimulus. If the Critic was not turned off before testing, the system effectively possessed vision and could in theory learn to adjust its responses even if the target was moved, although this was not tested.

An important component to the system in the Actor was "babbling" noise that was injected into the M cells. An untrained system thus possessed some tendencies to move weakly in a random direction. The Critic, then, was able to allow operant conditioning to shape the motor commands in the context of limb state.

The model was implemented in the NEURON 7.2 (Carnevale and Hines, 2006) simulator for Linux and is available on ModelDB (Peterson et al., 1996) (https://senselab.med.yale.edu/modeldb). We collected performance for a number of targets, random network wirings, and sets of injected babbling noise, and ran both naive versions of the model for these and multi-epoch training sessions on a number of different starting positions of the arm. In addition to performance measures, we also compared the naive and trained models using measures of population firing synchrony and inter-population information flow.

The remaining Methods subsections elaborate on the details of the model's architecture. First, the environment (virtual arm and target) are explained, then the actor portion of the model, including the P cells, and the S and M cells (which constitute the primary spiking learning neuron portion of the model). Then the critic and reinforcement learning algorithm are explained in more detail, and finally the training and testing trial scheme we used and the measures we used for network population synchrony and information flow between network populations.

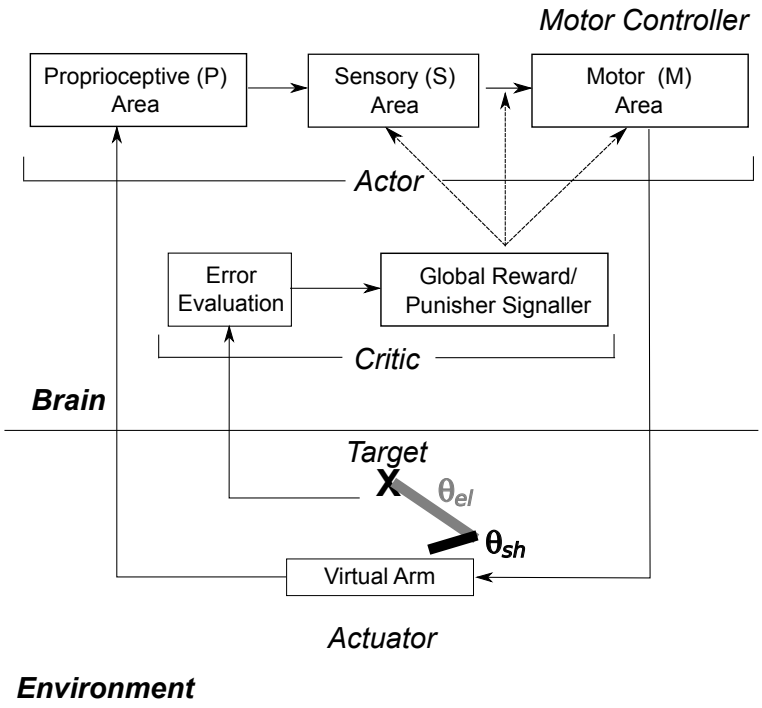

Figure 1: **Overview of model.** A virtual arm with joint angles θ_{sh} and θ_{el} (θ_{sh} : angle of upper arm with respect to x-axis; θ_{el} : angle of forearm with respect to upper arm) controlled by two pairs of flexor and extensor muscles, is trained to reach towards a target. A *proprioceptive (P)* sensory area translates muscle lengths into an arm configuration representation. Plasticity is present in excitatory to excitatory recurrent connections within the higher-order *sensory (S)* and the *motor (M)* areas, in feed-forward and feed-back excitatory to excitatory connections between the higher-order *sensory* and the *motor* areas, and in feed-forward connections from excitatory to inhibitory cells within each area. Motor units drive the muscles to change the joint angle. The *Actor* (above) is trained by the *Critic* which evaluates error and provides a global reward or punishment signal.

Environment: virtual arm and target

The virtual arm consisted of two segments representing the upper arm (length 1) and forearm (length 2). There were two joint angles for the two joints (shoulder: θ_{sh}; elbow: θ_{el}) that were allowed to vary from fully extended (θ_{sh}: $-45°$; θ_{el}: $0°$) to fully flexed (θ_{sh}: $135°$; θ_{el}: $135°$; large range of angles to more fully test learning). For each joint, an extensor and flexor muscle (lengths m_{ext} and m_{flex}) always reflected the current joint angle in the relationship as follows:

$$m_{ext} = \frac{(\theta - \theta_{min})}{\theta_{max} - \theta_{min}} \tag{1}$$

$$m_{flex} = 1 - m_{ext}. \tag{2}$$

Arm position updates were provided at 50 ms intervals, based on extensor and flexor EM (excitatory cells in the motor area) spike counts integrated from a 50 ms window that began 50 ms prior to update time (50 ms network-to-muscle propagation delay). The angle change $\Delta\theta = Spikes_{flexor} - Spikes_{extensor}$ for each joint was the difference between the corresponding EM spike counts from flexor and extensor populations during the prior interval, with each spike difference translating to a $1°$ rotation. For simplicity, the arm model did not contain physical attributes, such as mass and inertia. P drive activity updated after an additional 25 ms delay which represented peripheral and subcortical processing. Reinforcement occurred every 50 ms with calculation of hand-to-target error. The target remained stationary during the simulation.

Actor: unit types and interconnectivity

The actor consisted of the proprioceptive sensory (P), higher-order sensory (S), and motor (M) populations described above (Fig. 1). Details of the cell models are described below. Input to the S cells was provided by 192 P cells, representing muscle lengths in 4 groups (two flexor- and extensor-associated groups for each joint).

The rest of the network consisted of both S (sensory) and M (motor) cell populations. The S population included 192 excitatory cells (ES cells), 44 fast-spiking interneurons (IS), and 20 low-threshold spiking interneurons (ILS); similarly, the M network had 192 EM, 44 IM, and 20 ILM cells. The EM population was divided into four 48-cell subpopulations dedicated to extension and flexion about each joint, projecting to the extensor and flexor muscles. The number of excitatory and inhibitory cells within an area was selected to keep 75% $(192/256)$ of the neurons as excitatory, to approximate the ratios in neocortex.

Cells were connected probabilistically (fixed convergence; variable divergence) with connection densities and initial synaptic weights varying depending on pre- and post-synaptic cell types (Table 1). Connection densities were within the range determined experimentally, which are $\sim$1-100% depending on pre- and post-synaptic cell type (Thomson et al., 2002; Thomson and Bannister, 2003; Bannister, 2005). Initial synaptic weights were set to relatively low values, so as to resemble activity *in vivo*, which typically requires several pre-synaptic inputs to arrive within a short time-window in order to activate a postsynaptic neuron.

Table 1: **Connectivity parameters.**

Pre	$Post$	p	$Conv$	W	Pre	$Post$	p	$Conv$	W
P	ES	0.11250	22	15.0000	ES	ES	0.05625	11	* 1.3200
ES	IS	0.48375	93	* 1.9550	ES	ILS	0.57375	110	* 0.9775
ES	EM	0.09000	17	* 1.7600	IS	ES	0.49500	22	4.5000
IS	IS	0.69750	31	4.5000	IS	ILS	0.38250	17	4.5000
ILS	ES	0.39375	8	1.2450	ILS	IS	0.59625	12	2.2500
ILS	ILS	0.10125	2	4.5000	EM	ES	0.01913	4	* 0.4800
EM	EM	0.05625	11	* 1.1880	EM	IM	0.48375	93	* 1.9550
EM	ILM	0.57375	110	* 0.9775	IM	EM	0.49500	22	9.0000
IM	IM	0.69750	31	4.5000	IM	ILM	0.38250	17	4.5000
ILM	EM	0.39375	8	2.4900	ILM	IM	0.59625	12	2.2500
ILM	ILM	0.10125	2	4.5000					

Area (Pre: Presynaptic type; $Post$: Postsynaptic type) interconnection probabilities (p), convergence ($Conv$), and starting weights (W). * next to W represents plastic connection modified during learning. p is the probability of a connection being included among all possible connections between the 2 areas. $Conv$ is the number of inputs each cell of type $Post$ receives from type Pre. E cells used AMPA and NMDA synapses (NMDA, not displayed, had weights set at 10% of the colocalized AMPA synapse).

Actor: proprioceptive sensory (P) cell model

Proprioceptive sensory (P) cells (Fig. 1) were modeled using a standard, single compartment (diameter = 30 μm), parallel-conductance model with input current, to allow continuous mapping of muscle lengths to current injections provided to these cells. The rate of change of a P neuron's voltage (V) was represented as $-C_m \frac{dV}{dt} = g_{pas} * V + i_{drive}$, where C_m is the capacitive density (1 $\mu F/cm^2$), and i_{drive} was a current set according to muscle length. g_{pas} represents the leak conductance (0.001 nS), which was associated with a reversal potential of 0 mV. When a P neuron's voltage passed threshold, the neuron emitted a spike, and was set to a refractory state for 10 ms. Each P cell was tuned to produce bursting approaching 100 Hz (limited by refractory period) over a narrow range of adjacent, non-overlapping muscle lengths, by setting the P cell's i_{drive} variable to a heightened level. The i_{drive} variable of each P cell was updated when the arm moved (every 50 ms interval).

Actor: primary neuron model (sensory (S) and motor (M) cells)

Individual neurons in the higher-order sensory (S) and motor (M) areas were modeled as event-driven, rule-based dynamical units with many of the key features found in real neurons, including adaptation, bursting, depolarization blockade, and voltage-sensitive NMDA conductance (Lytton and Stewart, 2005, 2006; Lytton and Omurtag, 2007; Lytton et al., 2008a,b; Neymotin et al., 2011d; Kerr et al., 2012, 2013). Event-driven processing provides a faster alternative to network integration: a presynaptic spike is an event that arrives after a delay at postsynaptic cells; this arrival is then a subsequent

Table 2: **Neuron model parameters.**

Type	V_{RMP} (mV)	T_n (mV)	B_n (mV)	τ_A (ms)	R	τ_R (ms)	H (mV)	τ_H (ms)
E	-65	-40	-25	5	0.75	8.0	1.0	400
I	-63	-40	-10	2.5	0.25	1.5	0.5	50
IL	-65	-47	-10	2.5	0.25	1.5	0.5	50

Parameters (described below) of the neuron model for each major population type. These parameters are based on previously published models of neocortex, which were culled from the experimental and modeling literature (Kerr et al., 2012, 2013; Neymotin et al., 2011b,a,d).

event that triggers further processing in the postsynaptic cells. Cells were parameterized as excitatory (E), fast-spiking inhibitory (I), and low-threshold-spiking inhibitory (IL; Table 2).

Each neuron had a membrane voltage state variable (V_m) with a baseline value determined by a resting membrane potential parameter (V_{RMP}, set at -65 mV for pyramidal neurons and low-threshold-spiking interneurons, and at -63 mV for fast-spiking interneurons). This membrane voltage was updated by one of three events: synaptic input, threshold spike generation, and refractory period. These events are described briefly below; further detail can be found in the papers cited and code provided on ModelDB (Peterson et al., 1996) (https://senselab.med.yale.edu/modeldb).

Synaptic input

The response of the membrane voltage to synaptic input was modeled as an instantaneous rise and exponential decay: $V_n(t) = V_n(t_0) + w_s(1 - V_n(t_0)/E_i)e^{-\frac{t-t_0}{\tau_i}}$, where V_n is the membrane voltage of neuron n; t_0 is the synaptic event time (i.e., $t - t_0$ is the time since the event); w_s is the weight of synaptic connection s; E_i is the reversal potential of ion channel i, relative to resting membrane potential (where i = AMPA, NMDA, or $\mathrm{GABA_A}$; and $E_{\mathrm{AMPA}} = 65$ mV, $E_{\mathrm{NMDA}} = 90$ mV, and $E_{\mathrm{GABA_A}} = -15$ mV); and τ_i is the receptor time constant for ion channel i (where $\tau_{\mathrm{AMPA}} = 20$ ms; $\tau_{\mathrm{NMDA}} = 300$ ms; and $\tau_{\mathrm{GABA_A}} = 10$ or 20 ms for somatic and dendritic $\mathrm{GABA_A}$, respectively).

In addition to spikes generated by cells in the model, subthreshold Poisson-distributed spike inputs to each synapse of all units except the P and ES units were used to provide ongoing activity and babble (Table 3). These Poisson stimuli also represented inputs from other neurons not explicitly simulated. Since the neuron model is a point-neuron model, each synapse represents the locus of convergent inputs from multiple neurons.

Action potentials

A neuron fires an action potential at time t if $V_n(t) > T_n(t)$ and $V_n(t) < B_n$, where V_n, T_n, and B_n are the membrane voltage, threshold voltage (-40 mV for pyramidal neurons and fast-spiking interneurons, -47 mV for low-threshold-spiking interneurons), and blockade voltage (-10 mV for interneurons and -25 mV for pyramidal neurons), respectively, for neuron n. Action potentials arrive at target neurons at time $t_2 = t_1 + \tau_s$, where t_1 is the time the first neuron fired, and τ_s is the synaptic delay. τ_s values were

Table 3: **Noise parameters.**

$Cell$	$Synapse$	W	$Rate$
IS	GABA_A^{soma}	1.875	100
IS	AMPA^{dend}	4.125	200
IS	GABA_A^{dend}	1.875	100
ILS	GABA_A^{soma}	1.875	100
ILS	AMPA^{dend}	3.000	200
ILS	GABA_A^{dend}	1.875	100
EM	GABA_A^{soma}	1.875	100
EM	AMPA^{dend}	3.938	200
EM	GABA_A^{dend}	1.875	100
IM	GABA_A^{soma}	1.875	100
IM	AMPA^{dend}	4.125	200
IM	GABA_A^{dend}	1.875	100
ILM	GABA_A^{soma}	1.875	100
ILM	AMPA^{dend}	3.000	200
ILM	GABA_A^{dend}	1.875	100

Noise stimulation to synapses of the different cell types. Weight (W) values are afferent weights. Rate values are average stimulation frequencies in Hz (inputs are Poisson distributed). This stimulation represents afferent inputs from multiple presynaptic cells, which are not explicitly simulated.

selected from a uniform distribution ranging between 3 – 5 ms for dendritic AMPA, NMDA, and $GABA_A$ synapses, and were selected from a uniform distribution ranging between 1.8 – 2.2 ms for somatic $GABA_A$ synapses. Synaptic weights were fixed between a given set of populations except for those involved in learning (described in **System overview** above).

Refractory period

After firing, a neuron cannot fire during the absolute refractory period, τ_A (2.5 ms for interneurons and 5 ms for pyramidal neurons). Firing is reduced during the relative refractory period by two effects: first, an increase in threshold potential, $T_n(t) = \left(1 + Re^{-\frac{t-t_0}{\tau_R}}\right) T_n(t_0)$, where R is the fractional increase in threshold voltage due to the relative refractory period (0.25 for interneurons and 0.75 for pyramidal neurons) and τ_R is its time constant (1.5 ms for interneurons and 8 ms for pyramidal neurons); and second, by hyperpolarization, $V_n(t) = V_n(t_0) - He^{-\frac{t-t_0}{\tau_H}}$, where H is the amount of hyperpolarization (0.5 mV for interneurons and 1 mV for pyramidal neurons) and τ_H is its time constant (50 ms for interneurons and 400 ms for pyramidal neurons).

Critic: reinforcement learning algorithm

The RL algorithm implemented Thorndike's Law of Effect using global reward and punishment signals (Thorndike, 1911). The network is the *Actor*. The plastic AMPA weights in Table 1 were trained to implement the learned sensorimotor mappings. The *Critic*, a global reinforcement signal, was driven by the first derivative of error between position and target during 2 successive time points (reward for decrease; punishment for

increase), and therefore the reward/punishment signals were delivered at every movement generated by the network. As in (Izhikevich, 2007), we used a spike-timing-dependent rule to trigger eligibility traces to solve the credit assignment problem. The eligibility traces were binary, turning on for a synapse when a postsynaptic spike followed a presynaptic spike within a time window of 100 ms; eligibility ceased after 100 ms. When reward or punishment was delivered, eligibility-tagged synapses were potentiated (long-term potentiation LTP), or depressed (long-term depression LTD), correspondingly.

Synaptic weights $w(t)$ were updated (for LTP/reward and LTD/punishment) utilizing weight scale factors, w_s:

$$
\begin{aligned}
w(t) &= w_0 \cdot w_s(t) \\
w_s(t+1) &= w_s(t) + \Delta w_s \\
\Delta w_s &= \begin{cases} w_{inc} \cdot (1 - w_s(t)/w_s^{max}) & \text{for LTP reward} \\ -w_{inc} \cdot w_s(t)/w_s^{max} & \text{for LTD punisher} \end{cases}
\end{aligned}
$$

where w_s^{max} is maximum weight scale factor, w_0 is the initial synaptic weight, and w_{inc} is the weight scale increment. w_s was initialized to 1.0 for all synapses and varied between 0 and w_s^{max}. w_s^{max} was set to 6 and 2.5 times the synaptic weight of E $\rightarrow$ E and E $\rightarrow$ I baseline weights. w_{inc} was set to 25% of baseline synaptic weights.

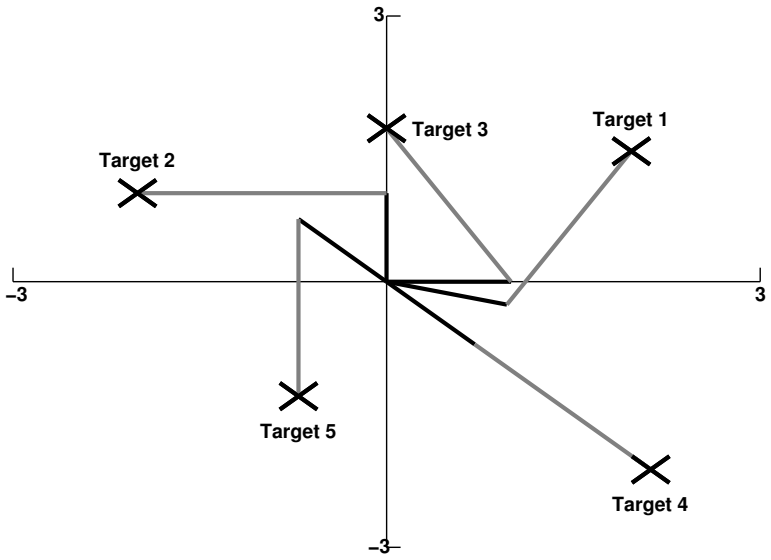

Figure 2: **Schematic of arm orientations at all five target locations.** Each "X" symbol represents a target. The arm is drawn in an orientation that maintains its endpoint on each of the five targets. Targets were chosen to allow thorough testing of reaching (both extrema and intermediate positions).

Training and testing paradigm

Networks were trained to reach the arm to a single target (targets shown in Fig. 2). Training a network to reach to a single target consisted of 200 training sessions. Each training session consisted of allowing the network to perform 15 s of reaching, once from each of 16 sequential starting positions. The 16 starting positions were arranged from minimum to maximum angles for the two joints. We configured starting angles in this way to teach the network to control movement of the arm to the target from the entirety of positions in the 2D plane. Targets were chosen to allow thorough testing of reaching (both extrema and intermediate positions).

After training, learning was turned off and each network's performance was assessed with the arm initialized from each of the 16 starting positions used for training. A reach was considered successful if the arm end-point was moved to a position where Cartesian error was ≤ 1. Overall learning performance for a target was calculated as the fraction of successful reach movements (Accuracy). A similar accuracy score was used for angular performance for each joint: when the angular error was ≤ 10 degrees, the reach for that given joint was a success.

Data analysis

Data obtained from 400 naive trials (5 random network wirings, 5 random input seeds, 16 starting positions) were compared with 2000 trained trials (5 random network wirings, 5 random input seeds, 5 targets, 16 starting positions).

Synchrony between cells within different populations was measured using a nor-

malized population coefficient of variation (CV_p (Tiesinga and Sejnowski, 2004)). CV_p makes use of the population's interspike interval, defined for the temporally ordered set of spikes generated by neurons in the population as: $\tau_v = t_{v+1} - t_v$, where t_i indicates the i^{th} spike time. CV_p is then defined as $\frac{\sqrt{<\tau_v^2> - <\tau_v>^2}}{<\tau_v>}$, where p stands for population and $<>$ denotes the average over all intervals. CV_p is normalized to be between 0–1 by subtracting 1 and dividing by $\sqrt{N}$. After the normalization, 0 indicates independent Poisson process synchrony and 1 indicates maximum synchrony. Values that dip below 0 are set to 0 to allow calculating means.

We used normalized transfer entropy (nTE) between multiunit activity vectors (MUAs) of different populations before and after training as a measure of information flow (Gourevitch and Eggermont, 2007; Neymotin et al., 2011a). MUA vectors were the time series formed by counting the number of spikes generated by a population in every 5 millisecond interval. nTE is a normalized version of transfer entropy, defined from probability distribution $X1$ to $X2$ as: $H\left(X2_{\text{future}}|X2_{\text{past}}\right) - H\left(X2_{\text{future}}|X2_{\text{past}}, X1_{\text{past}}\right)$. $X2_{\text{future}}$ and $X2_{\text{past}}$ represent the $X2$ probability distributions of future and past states, respectively, and H is the entropy of the given distribution. nTE from $X1$ to $X2$ is then defined as: $\frac{TE_{X1\rightarrow X2} - <TE_{X1_{\text{shuffled}}\rightarrow X2}>}{H(X2_{\text{future}}|X2_{\text{past}})}$. nTE removes bias from the estimate of transfer entropy by subtracting the average transfer entropy from $X1$ to $X2$ using a shuffled version of $X1$ denoted $< TE_{X1_{\text{shuffled}}\rightarrow X2} >$, over several shuffles. It then divides the estimate by the entropy of $H(X2_{\text{future}}|X2_{\text{past}})$, to get a value between 0 and 1. nTE will be 0 when $X1$ transfers no information to $X2$, and will be 1 when $X1$ transfers maximal information to $X2$. For calculating nTE, we shuffled each presynaptic MUA vector 30 times. For more information on calculating nTE, see Neymotin *et al.*, 2011

(Neymotin et al., 2011a).

For each network trained on a particular target, we calculated a per-joint bias-score, calculated as the difference between the sums of incoming excitatory weights to flexion and extension motor units. This bias measure was normalized to the range of -1 to 1, with -1 and 1 corresponding to extension and flexion biases, respectively.

3 Results

This study involved over 2000 15 s simulations of trained networks, using five different random wirings, five different input streams, five different targets, and sixteen initial arm positions, as well as 400,000 15 s simulations run during training (five random wirings, five input streams, five targets, sixteen initial arm positions, two hundred reaches from each position). The network learned to reach a two degree-of-freedom virtual arm from starting positions arrayed in a restricted subspace chosen around an oval (large set of θs with restricted r in polar coordinates). This choice provided curved solution trajectories, thereby avoiding the complex co-contractions of muscles associated with linear movements. Targets were set to test both extrema and intermediate positions. Simulations were run on Linux on a 2.27 GHz quad-core Intel XEON CPU. A 15-s simulation ran in 15 – 25 seconds, depending on the simulation type.

Learning alters network dynamics

Prior to training, firing rates of units in the motor (M) area (EM,ILM,IM) were low with sparse firing produced by the stochastic inputs into the motor area (Fig. 3A; Table 4).

This stochastic input was the source of motor babble, necessary to provide the variation that underlay motor learning. Before training, low variability of arm position kept proprioceptive sensory (P) cells at nearly constant low spiking rates (Fig. 3A). Due to strong fixed projections from P to higher-order sensory (S) populations, these low rates were able to maintain higher-order sensory cells at moderate levels of activity.

During training, plasticity was present at 3 sites: E→E recurrent connections in both S and M areas; bidirectional in E→E connections between S and M areas; local E→I connections within S and M areas. As in our prior simulations, E→I learning was provided in order to avoid the runaway gain sometimes seen with excitatory loop learning, even in the presence of LTD (Neymotin et al., 2011b). Excitatory weight gains between the different populations tended to increase 3-fold: ES→ES: 3.2×; ES→EM: 3.1×; EM→EM: 3.1×. However, synaptic weights did not saturate, remaining at intermediate values due to LTP and LTD co-occurring. By contrast, E→I projections increased only ∼30%. The result of the overall increased excitation was an increase in firing rates in most cell types (Table 4). However, ES rates were almost unchanged, although the inputs from P, which carried positional information, were now being used for control of reaching the arm to target (see below).

Synchrony between cells within the ES population was evident at baseline (vertical stripes in Fig. 3A), with normalized population coefficient of variation (CV_p (Tiesinga and Sejnowski, 2004)) showing non-zero synchrony, beyond independent Poisson process coincidence levels (0 on the CV_p scale of 0–1). Learning produced a significant increase in synchrony in several of the populations (Fig. 4A). Increase in synchrony was most evident in the M area, with significant increase in some S cell groups as well

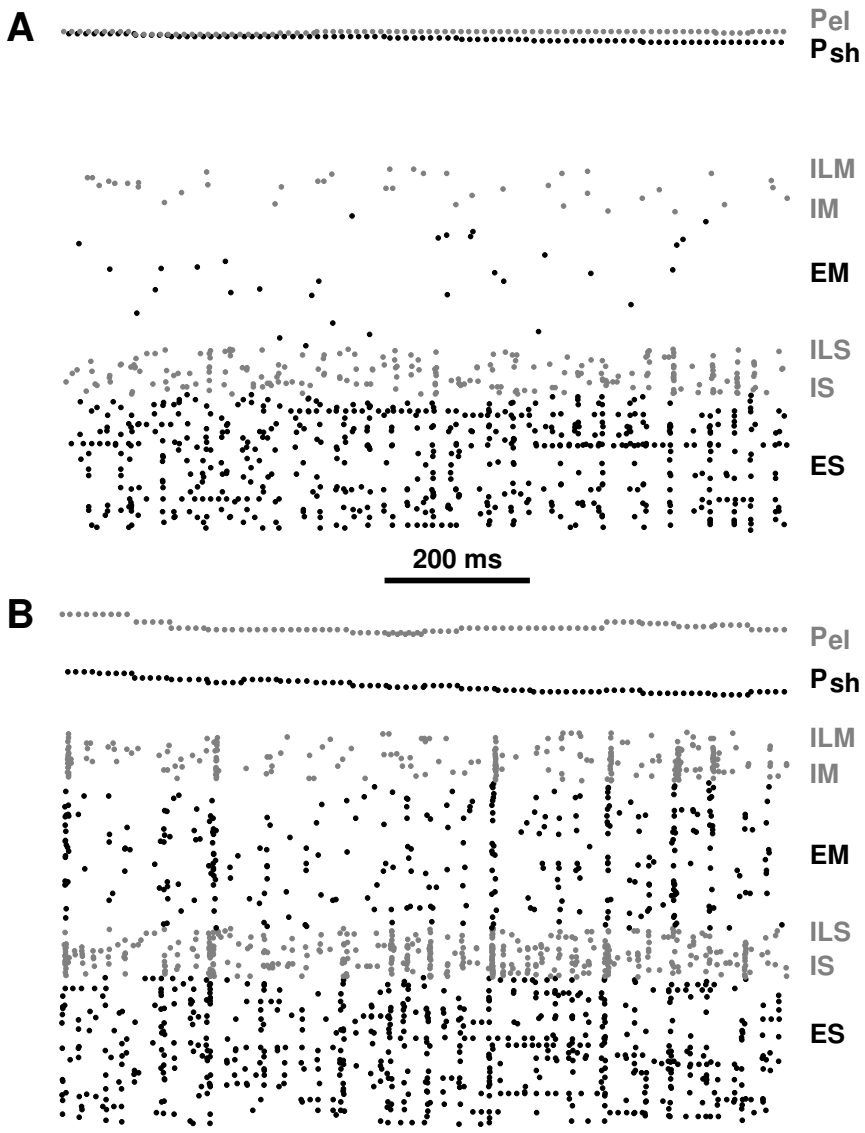

Figure 3: **Raster plot of network spiking. (A)** naive network **(B)** network after training. Gray (black) dots are spikes in inhibitory (excitatory) cells; ES,IS,ILS,EM,IM,ILM (E excitatory; I inhibitory fast-spiking; IL inhibitory low-threshold spiking interneurons; S higher-order sensory; M motor; P_{sh} proprioceptive sensory shoulder (gray); P_{el} proprioceptive sensory elbow (black)).

Table 4: **Firing rates.**

Condition	P	ES	IS	ILS	EM	IM	ILM
Naive	0.98	3.36	5.60	3.36	0.17	0.53	0.41
Trained	1.01	3.37	8.42	4.87	1.74	4.32	2.45

Average firing rates (Hz) for the different cell populations before (Naive) and after training.

(Fig. 3B; Fig. 4A). This demonstrated the development of temporal structure manifested as synchrony, a correlate of the dynamical structure required to perform the task.

Normalized transfer entropy (nTE) between multiunit activity vectors (MUAs) demonstrated increased information flow between populations (Fig. 4B; Fig. 5). Although P$\rightarrow$ES weights were fixed, there was a significant increase in nTE across these populations (0.0537 to 0.0625; SEMs $\leq 1\%$). This change demonstrated that network reorganization, due to changes in other projections onto the S area, allowed alteration of ES activity so as to better follow incoming proprioceptive information and improve performance. The large increase in nTE in the main feed-forward pathway from ES$\rightarrow$EM (0.0247 to 0.1752) reflected the presence of structure in proprioceptive information which provided the ES populations the ability to select particular EM units to activate for the signaled movement. Increase in local-connectivity nTE from E$\rightarrow$I within each region was consistent with tuning of network inhibition to suppress cells that would interfere with performance. This change suggested emergence of lateral inhibitory feedback influences (the equivalent of a geometrical inhibitory surround). The projection from EM$\rightarrow$P cells closed the loop. Increased nTE after learning (0.0008 to 0.0124)

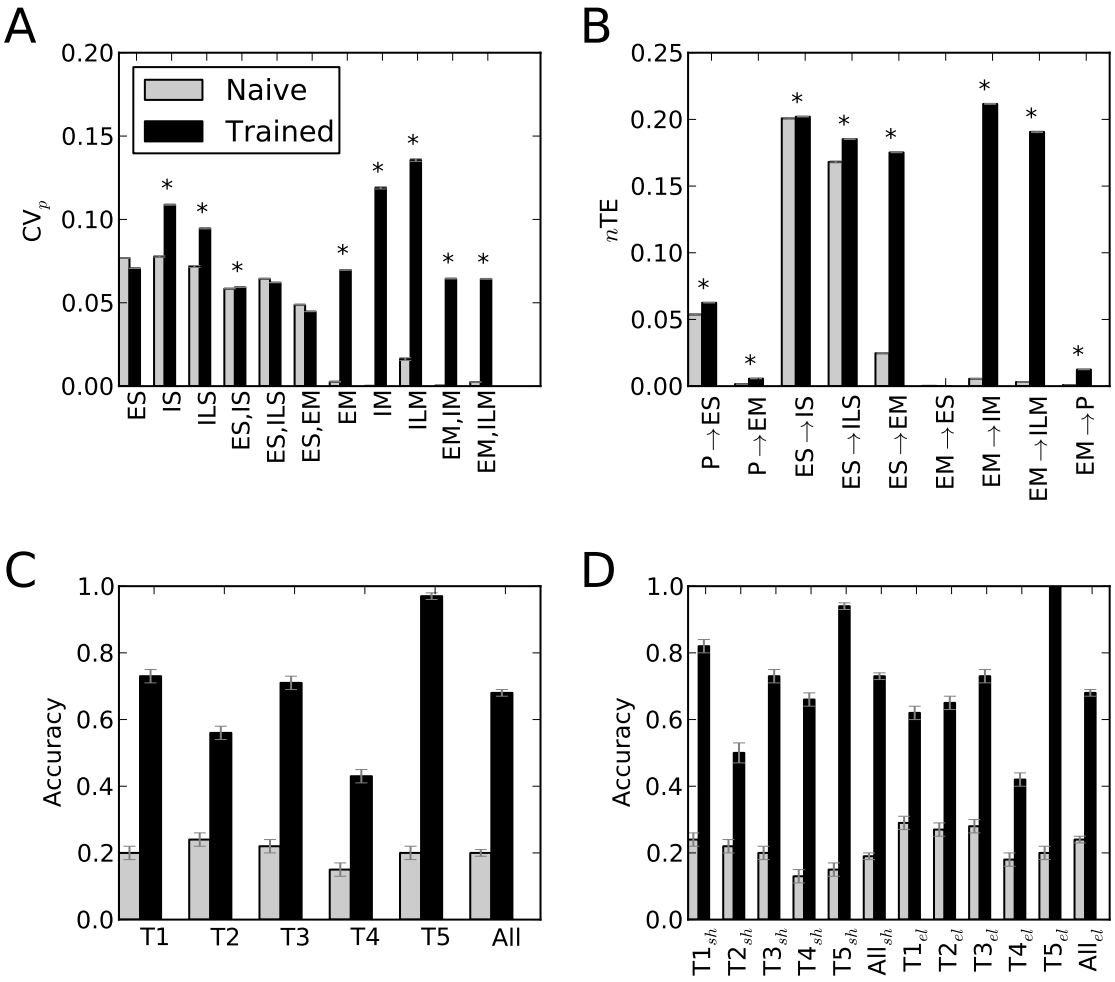

Figure 4: **Analysis across N=2400 simulations with different randomizations, targets, starting positions. (A)** Average population synchrony (CV_p). (SEMs not visible) Asterisks: significant increases; 2-sided t-test, $p < 0.01$. **(B)** Average nTE. (SEMs not visible). Asterisks, significant increases; 2-sided t-test, $p < 0.01$. **(C)** Successes (average hits $\pm$ SEM; all differences significant). **(D)** Angular hit scores (average $\pm$ SEM; all differences significant; *sh* shoulder; *el* elbow).

demonstrated that EM movement-related activity was then predictive of future proprioceptive states. This shift suggested how such a signal could also be utilized as efference copy.

Trained networks perform reaching

Individual networks were each trained to navigate toward a particular location (Fig. 6). Prior to training, the arm's end-point would move only slightly from its initial starting position (Fig. 6 gray traces). After training, with learning off, the network was able to move the arm from arbitrary starting positions to the trained target (Fig. 6, black traces). Generally, the network moved the arm successfully to its target in a near-optimal trajectory. The ongoing noise in the system tended to reduce the smoothness of the motion and often caused the arm to deviate slightly from the target, once reached. Arm movements were successfully made to targets from one extreme to the other (extreme extension to extreme flexion in Fig. 6A and the reverse in Fig. 6B). A single network learned a single target but could reach this target from any starting point at either side of the target. In Fig. 6C, the network moved the arm from maximum flexion towards the intermediately positioned target. The arm did not overshoot, demonstrating that the network was able to keep track of the endpoint position to determine which direction to move in. In Fig. 6D, the same network directed the arm towards the target from an initial position of maximum extension. Here, a slight overshoot was seen, but the arm immediately moved back towards the target afterwards.

Across targets, training significantly improved performance compared to the naive networks with substantial variability, ranging from 0.43 to 0.97 success for trained net-

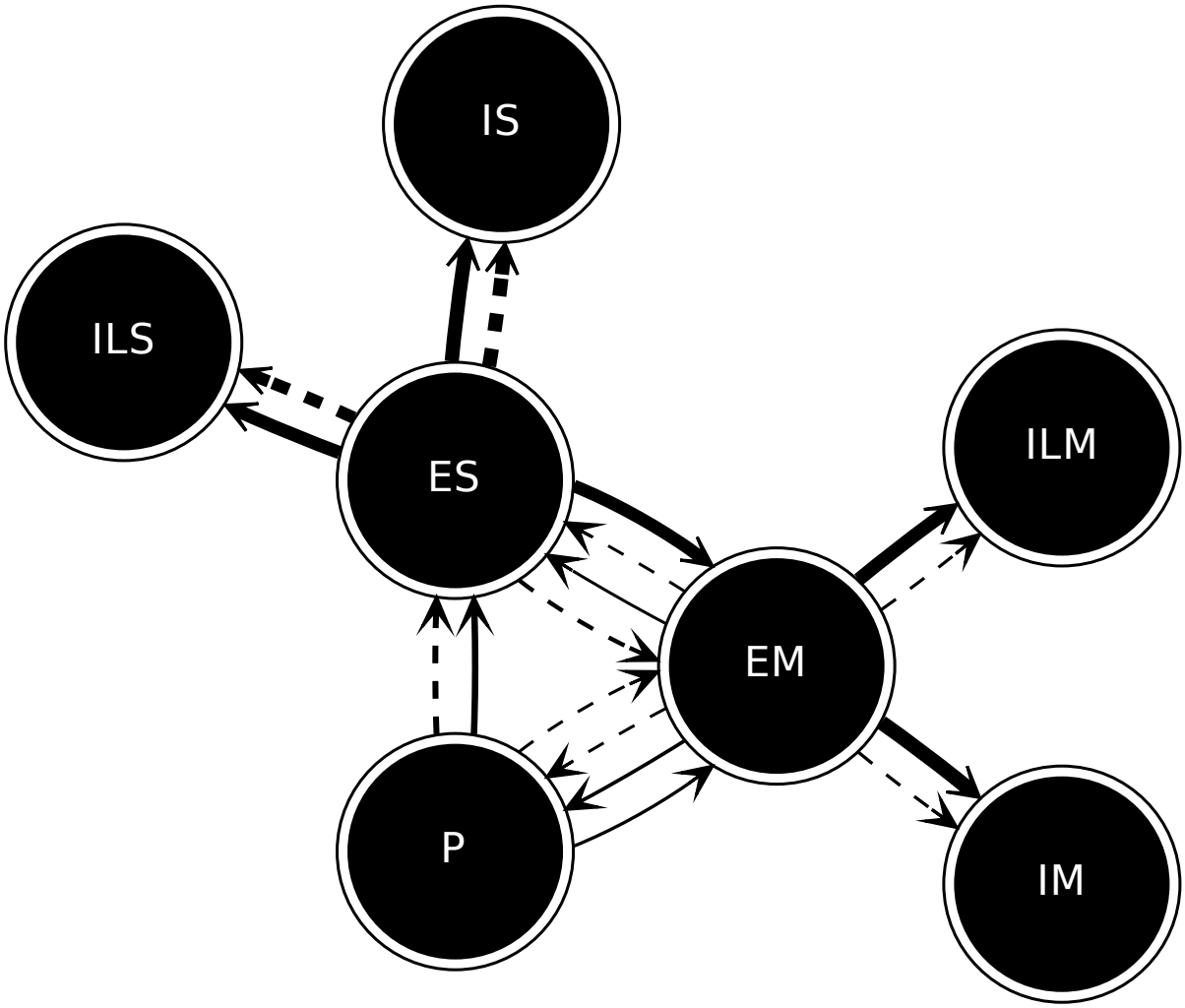

Figure 5: **Average nTE before (dotted lines) and after (solid lines) learning.** Each circle represents a population. Arrow represents direction of nTE. Thickness of lines indicates nTE magnitude (corresponding to bar magnitudes in Fig. 4B).

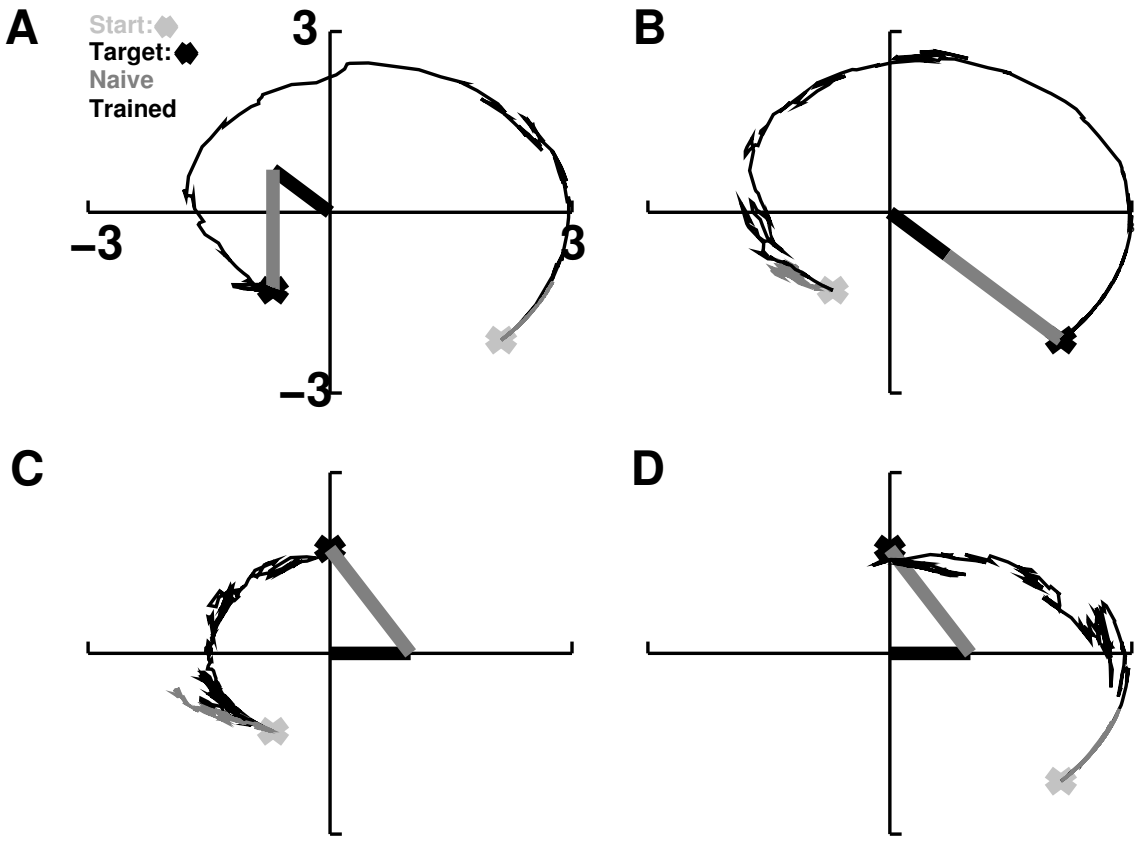

Figure 6: **Sample model performance on four desired arm trajectories.** Naive network trajectories in gray; trained in black. Arm is shown at target position (black: upper arm; gray: forearm). **(A)** maximal flexion at both joints (T5); **(B)** maximal extension (T4); **(C,D)** intermediate target (T3) approached from opposite directions.

works and 0.15 to 0.24 for naive networks (Fig. 4C). Success was calculated for naive and trained networks from identical initial conditions (starting position and random inputs), where for each of the N=2400 trials, a score of 1.0 indicated the hand had, during some time in the simulation, reached within a Cartesian distance of 1.0 from the target, and a score of 0.0 indicated this was not achieved. Networks were more readily trained for some targets, with maximum flexion being the easiest to reach and maximum extension being the most difficult. Training significantly improved performance for all targets ($p <$ 1e-9, two-tailed t-test; Fig. 4C).

Trained networks reduced error (approached the target) over time (Fig. 7). The panels here correspond to those in Fig. 6. In some cases, the initial movement of the trained network increased error; because hand location is constrained by rotation at the two joints provided, it must in some cases initially move away from the target in order to ultimately reach it. In these cases, movement begins to reduce error, after the arm passes through the vertical axis at about 5 s (*e.g.,* sharp drop of error in Fig. 7A). By contrast, when the target was centered, the error did not show this increase (Fig. 7C,D). In these cases, the arm oscillated more at the target, lacking the externally imposed constraint of the extremum as a counterbalance to attempted movement.

Once the arm reached the target, error remained relatively low, with small oscillations caused by the ongoing noise/babble. Overall, trained networks all showed substantially greater reduction in overall error as a function of time. Pearson correlation between error and time values were significant ($p < 0.05$) and negative for the trained networks (average -0.31 $\pm$ 0.01) and showed significant difference ($p < 0.05$, two-tailed t-test) from the naive networks. Performance for individual targets varied, but all

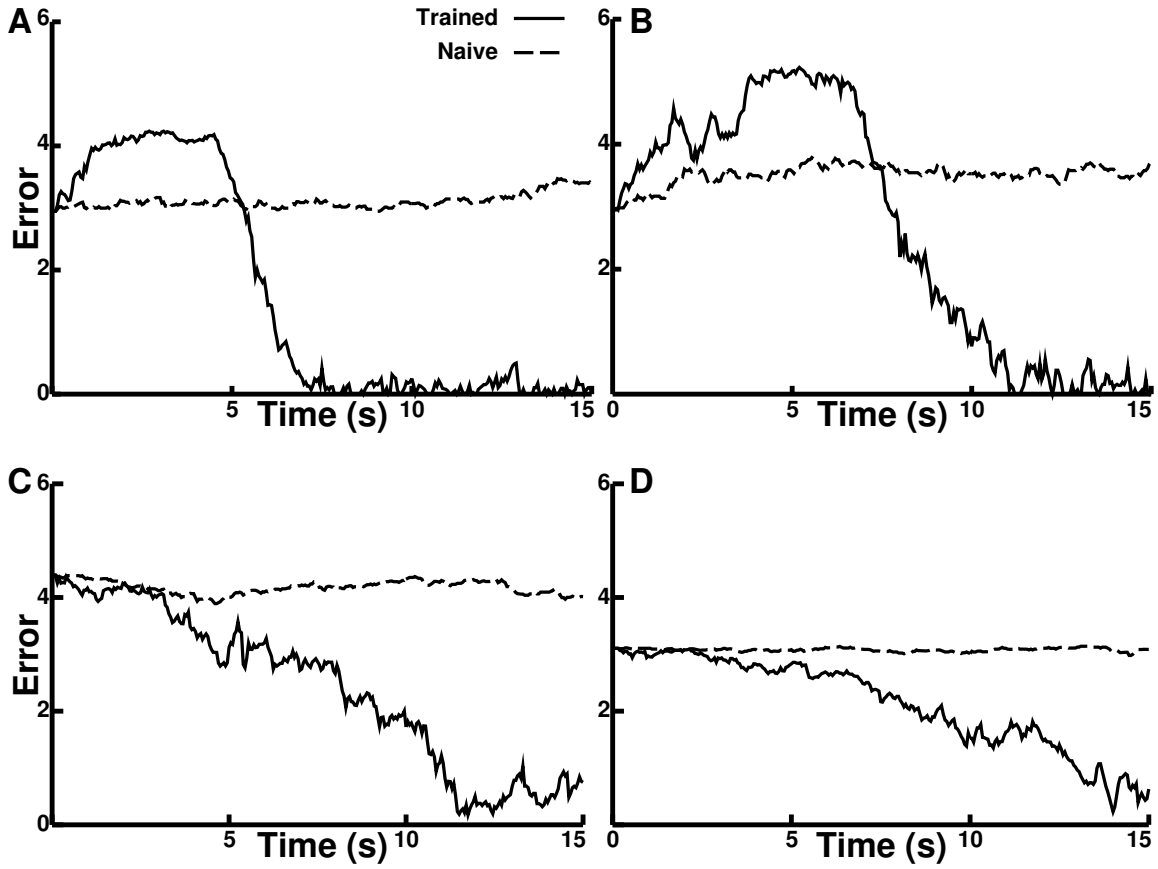

Figure 7: **Cartesian error vs. time performance on four desired trajectories.** The panels correspond to the trajectories over the 15 s simulations shown in the same Fig. 6 panels.

trained networks showed a trend towards decreasing error over time, as expected from motion of the arm from its starting position towards the target.

We examined trajectories of joint angles over time in individual reach trials (Fig. 8 and Fig. 4D). Trained networks were typically able to stabilize both joint positions within 10 degrees of target locations. After training and across targets, this occurred 73% and 68% of the time for shoulder and elbow angles, respectively, compared to only 19% and 24% for the naive networks (each of the N=2400 scores used for calcu-

lating means in Fig. 4D was set to 1.0 when, at some time during the simulation, the joint angle fell within 10 degrees of its target, and set to 0.0 when this did not occur). Depending on target, the accuracy of trained network performance ranged from 50% to 100% (Fig. 4D).

Fig. 8A,B correspond to the reaches depicted in Fig. 6C,D. These reaches were accomplished by a single trained network, with the arm beginning at opposite sides of the target. In Fig. 8A, the arm begins at maximum flexion. In this case, the majority of the reach is accomplished via rotation about the shoulder joint. Fig. 8B shows movement to the intermediate target from maximum extension. Here, the majority of the movement is accomplished via rotation about the elbow joint. The network only utilizes minimal shoulder movements to bring the arm close to the target. These examples demonstrate that a single trained network can dynamically reconfigure which joint to utilize for a reach, depending on currently available proprioceptive information.

We evaluated reach performance as a function of training epoch (Fig. 9) for the three targets shown in Fig. 6. Overall, training quickly reduced error below that of the naive networks. However, there was considerable variability in learning performance, depending on the target. The maximum flexion target showed fast learning, with error dropping close to zero after the first training epoch (Fig. 9A). This is consistent with best overall performance (Fig. 4C; T5). The error for the maximum extension target tended to oscillate with high deviations, also consistent with the lower performance of reach movements towards the maximally extended target (Fig. 4C; T4). The intermediate target showed intermediate performance, with lower amplitude oscillations in error. The oscillations in error were partially due to optimization of reach movements

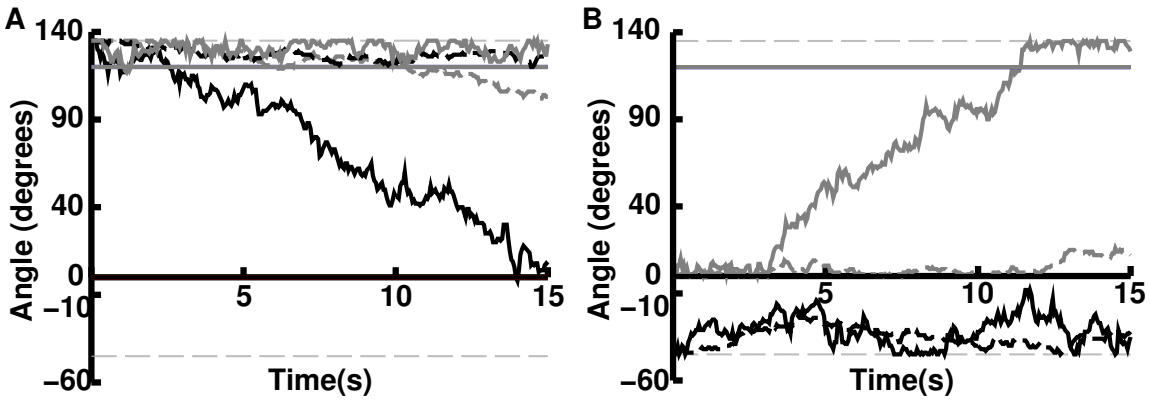

Figure 8: **Shoulder and elbow joint angle performance for two desired trajectories to same target. (A,B)** show the same trajectories as in Fig. 6C,D, both targeting T3. Shoulder (black) and elbow (gray) joint angles are shown over the course of the 15 second reach trial as controlled by a trained (solid lines) network and naive (dotted lines) network to a particular target. Horizontal solid lines indicate the target angles in degrees (shoulder target at zero degrees). Thin horizontal gray lines indicate minimum and maximum angles.

from specific starting positions, which could detrimentally impact reach performance from other starting positions. In addition, the babble noise was unmodulated as training progressed, which could lead to partial interference in learning.

By running a network for 15 s with both learning and muscle motions turned off and analyzing the direction that the EM units would have caused the arm to move in from a grid of 256 starting positions, we were able to extract *motor command maps* for four different cases of networks attempting to reach for a particular target (Fig. 10). Fig. 10A shows an untrained network trying to reach for the most-flexed target (T5); the motor commands at all of the starting positions are essentially insignificant, which is typical for all of the naive networks. After training, the vectors tended to point towards the target (Fig. 10B-D). Fig. 10B shows a trained network reaching for, again, T5. Most of the vectors are colored red, representing directional preferences that point towards the target. However, the gray vectors in the bottom right quadrant of Fig. 10B show movement preferences that actually increase error, yet are nonetheless required for arm movement, based on rotational constraints at the joints. During training, the network would be punished for following these trajectories from these points, yet the overall training permits these movement preferences to be learned. Because the target is at extreme flexion, the overall tendency of the learning is to reinforce flexion and suppress extension. This permits global learning that is contrary to local cues. Similarly, Fig. 10C shows a motor vector field pattern consistent with reinforced extension (T4).

With the target at an intermediate position (Fig. 10D; T3), the vectors are not as clearly oriented and are of reduced magnitude. The reduced magnitude promoted more conservative movements, advantageous because positioning the arm over an interme-

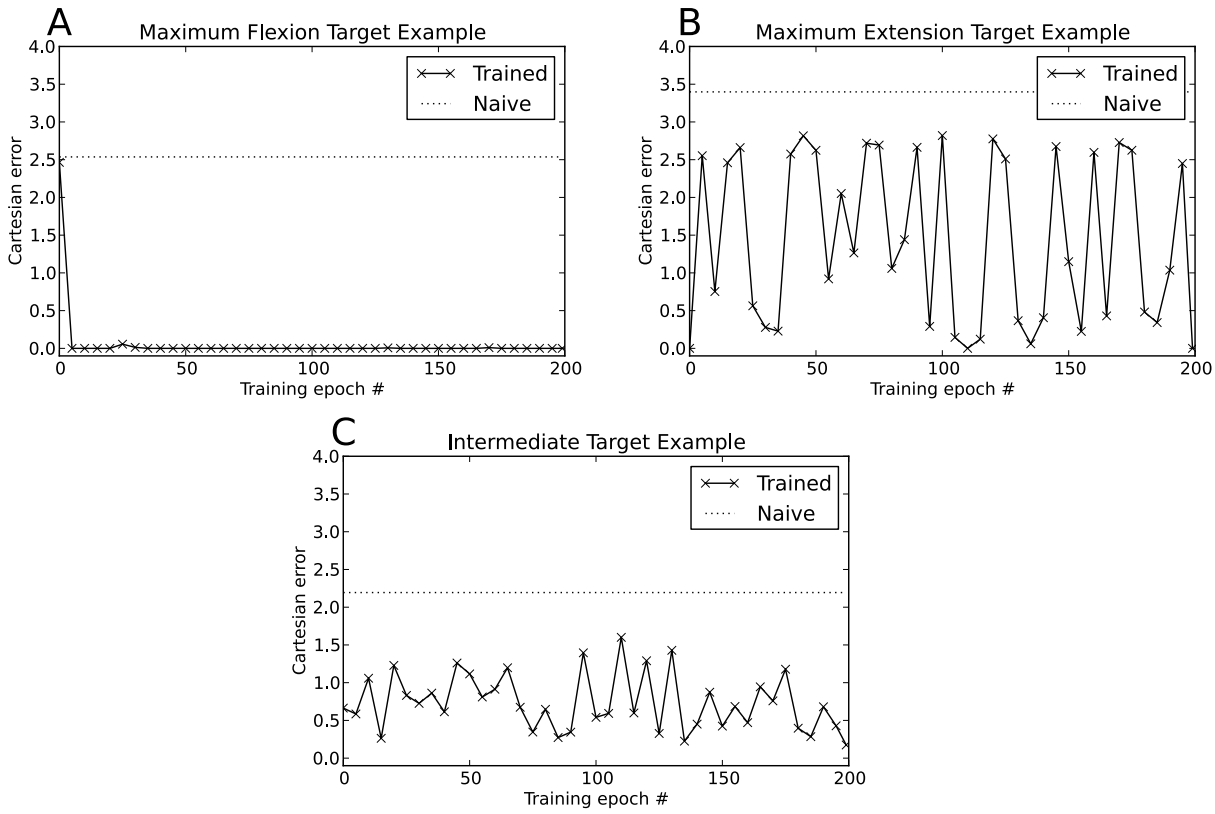

Figure 9: **Average minimum Cartesian error (across 16 starting positions) as a function of training epoch. (A), (B), (C)** correspond to the average (over all starting positions) performance on the targets used in Fig. 6A (T5), Fig. 6B (T4), and Fig. 6C,D (T3), respectively. Performance at epoch 0 is the performance after the first training session.

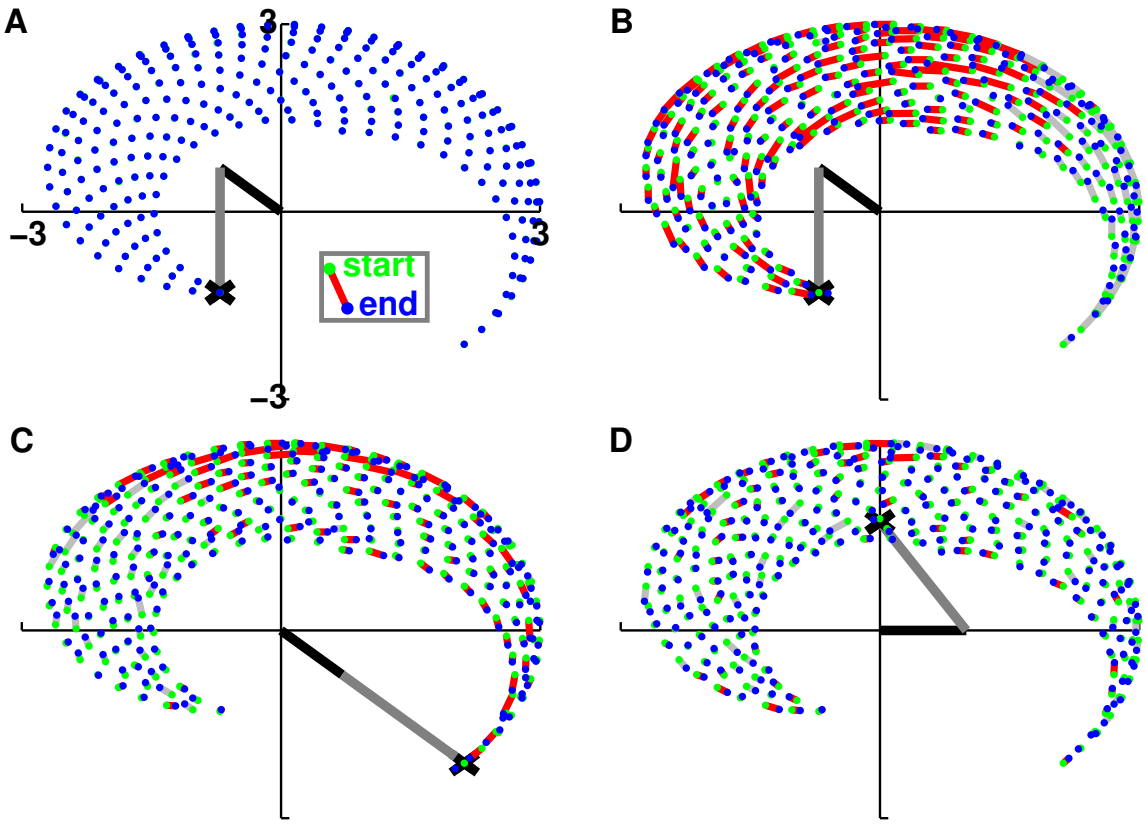

Figure 10: **Movement command vectors (15 s of simulation from grid of 256 starting positions).** Movement vectors are drawn from green to blue. Red vectors point toward target (decreasing Cartesian hand-to-target error); gray away (increasing error). Magnitude of each vector is scaled by 2X. **(A)** Movement commands generated by a naive network have no directional selectivity. **(B)** Maximum flexion and **(C)** maximum extension vectors tend to point towards the target. **(D)** Motor commands for intermediate target show directional selectivity towards the target from opposite directions, but have smaller magnitudes.

diate target required balance: too much extension or flexion results in over- or under-shooting the target. For the trained networks, average angular error reductions about each joint per movement command across targets and starting positions were -0.29^o per move for shoulder and -0.15^o for elbow (SEM $\leq$0.01^o), demonstrating that the network tended to generate movements that would reduce error. The larger reduction in shoulder error is due to its larger role in positioning the end-point of the arm, since the elbow position depends on the upper-arm position.

Flexion bias scores, representing difference between extension vs. flexion weights at a joint from -1 to 1, showed the expected flexion bias at both joints (average $\pm$ SEM: 0.22 $\pm$ 0.01 and 0.27 $\pm$ 0.01 for shoulder and elbow, respectively) in the case of the maximum flexion target. However, the maximum extension target produced networks with flexion bias at the elbow (0.03 $\pm$ 0.02), with only slight extension bias at the shoulder joint (-0.12 $\pm$ 0.03). This corresponded to the lower hit score for the maximum extension target compared to maximum flexion target. The intermediate targets had extension bias at the shoulder (T1: -0.14 $\pm$ 0.02; T2: -0.01 $\pm$ 0.01; T3: -0.12 $\pm$ 0.01), with primarily flexion bias at the elbow (T1: 0.03 $\pm$ 0.02; T2: -0.02 $\pm$ 0.01; T3: 0.05 $\pm$ 0.01). Balancing bias at the two joints appeared to be a strategy to allow movement to occur readily in opposing directions so as to allow for target acquisition from different initial points.

4 Discussion

The results in this paper demonstrate the flexibility of the network architecture and learning algorithm developed in (Chadderdon et al., 2012). Here we have extended the target tracking task to the more challenging problem of controlling two independent joints to perform the reach. ES cells contain random mappings from the proprioceptive P cells, which leads to individual ES cells forming conjunctive representations of configurations of both joints. The global reinforcement mechanism induces plasticity which shapes the EM cell response to the current limb configuration represented in ES. The target is effectively represented implicitly (see below) by the visual set point which the reinforcement algorithm uses to determine whether the network is rewarded or punished for the motor commands it issues in response to current limb configuration. Such a system effectively forms attractors for the target arm configuration by shaping the immediate response to particular points the arm is at in the trajectory. Fig. 10 graphically shows the type of motor command map that implements these attractors. These attractors may function either when learning is turned off (as is done during testing in this paper) or left on, though continued learning may add some interference to the learned attractor. As a consequence of the attractor structure, only one target may be learned by the system at a time.

Additionally, although we did not actively test it under this task, we have previously demonstrated that this model is capable of unlearning old attractors and relearning new ones based on a shift of the reinforcement schedule (Chadderdon et al., 2012): a feature that adds great adaptive flexibility to the simulated agent's reaction to its environment.

Punishment "stamps out" no-longer relevant attractors and babbling in conjunction with reward is able to "stamp in" newer, desired attractors. Although we turn off learning before we test the performance of the model in this paper (in order to control for the possibility of new learning affecting performance), there is biologically plausibility in always leaving the learning algorithm on at some level (Sober and Brainard, 2009). This is easily accomplished, and in the future, we intend to adapt the level of babbling motor noise according to the degree to which the agent is being rewarded (more reward, less injected noise). When this is done, learning should become even more efficacious and leaving learning on less detrimental than is presently evidenced in Fig. 9.

Learning produced alterations in network dynamics, including enhanced neuronal synchrony and enhanced information flow between neuronal populations. After learning, networks retained behaviorally-relevant memories and utilized proprioceptive information to perform reaches to targets from multiple starting positions. Trained networks were able to dynamically control which degree-of-freedom (elbow vs. shoulder) to utilize to reach a target, depending on current arm position. Learning-dependent dynamical reorganization was evident in sensory and motor populations, where synaptic weight patterning was produced through a balance of convergent excitatory weights onto motor populations projecting to extensors and flexors.

We make a number of specific, testable predictions from the model.

1. Balanced learning (changes in both E$\rightarrow$E and E$\rightarrow$I weights) is needed to produce selection of correct motor units while suppressing activation of incorrect motor units via selective inhibition. Testable using selective pharmacological blockade

or optogenetics.

2. Learning enhances synchrony in neuronal populations and enhances behaviorally-relevant information flow across neuronal populations. Testable with electrophysiological recording techniques (multiple areas and/or single units) and nTE. However, information flow (measured by nTE) can change across 2 populations due to dynamical factors in the absence of learning, or even direct synaptic connections, between these populations (*e.g.,* EM$\rightarrow$P in Fig. 4). Thus, although nTEcan sometimes provide evidence of learning (Lungarella and Sporns, 2006), it must be interpreted cautiously.

3. Enhanced sensory processing works in tandem with motor alterations to improve task-relevant motor performance. Testable *in vivo* by erasing memories from sensory areas (Pastalkova et al., 2006; Von Kraus et al., 2010). Additionally, motor cortex erasure could be used to demonstrate that re-learning is accelerated in the presence of the prior sensory learning. These predictions could also be tested further in our model.

4. Learning to a motion extremum is faster than learning to intermediate positions since motion limitations can be used, eliminating the need for learning balance across antagonist muscles (preliminary experiments confirm, P.Y. Chhatbar, personal communication). More generally, the relative ease of a particular movement *in vivo* depends on the amount of sensory information required to complete the movement. Testable by kinesiology.

Environmentally-constrained structure and function

Functional connectomics seeks to explain dynamics and neural function as emergent from detailed neuronal circuit connectivity (Sporns et al., 2005; Shepherd, 2004; Reid, 2012). Circuit changes have been correlated with brain diseases, such as epilepsy (Dyhrfjeld-Johnsen et al., 2007; Lytton, 2008) and autism (Qiu et al., 2011). Our past modeling work has confirmed the importance of microcircuit structure on neural function, demonstrating that alterations in connectivity change both dynamics and information transmission in neuronal networks (Neymotin et al., 2011a,d,c,b).

The embedding of brains, and by extension neuronal networks, in a physical (or simulated) world has been hypothesized to be an essential part of learning, as seen as the evolution of network dynamics (Almassy et al., 1998; Edelman, 2006; Krichmar and Edelman, 2005; Lungarella and Sporns, 2006; Webb, 2000). This theory maintains that the environment and brain influence each other as learning selects neuronal dynamics (selective hypothesis) (Edelman, 1987). In the present work, learning depended on the interaction with the rudimentary simulated environment: the virtual arm and target. This embodiment can now be used to make predictions for learning-related changes occurring during the perception-action-reward-cycle (Mahmoudi and Sanchez, 2011).

Embedding also provides a step towards using simulation to assess functional importance of various dynamical measures commonly utilized on *in vivo* electrophysiological data. Here, we found that synchrony and nTE were both enhanced after learning. These measures have been suggested as a means for brains to coordinate activity and process information (Engel et al., 1991; Lungarella and Sporns, 2006; Von der Mals-

burg and Schneider, 1986; Neymotin et al., 2011a; Uhlhaas and Singer, 2006). Our biomimetic brain model learned a function that can then be correlated with specific aspects of ensemble dynamics. The functional connectome can thus be dissected by looking at two steps: 1. the emergence of dynamics from connectivity (the dynamic connectome); 2. the relation of function to aspects of dynamics (the *functionome*: the set of functions a network can perform as constrained by its dynamics and dynamical embedding within the environment).

Target selection

Representation of both visual and somatosensory state information, including target information, is believed to be located in posterior parietal areas, and this information propagates to premotor and motor cortex (Shadmehr and Krakauer, 2008). These representations may be modulated via processes that select task-relevant information. Recent experiments have shown that premotor cortex activity is predictive of *changes of mind* that result in switching between targets mid-movement (Afshar et al., 2011).

Our present model selects the target implicitly via the Cartesian visual reference point that the reinforcement learning algorithm uses to determine whether the hand is moving closer (reward condition) or further away (punishment condition) from the desired location. A part of the brain upstream of the dopamine cells signaling error might perform the error calculation and cue the correct valence of reinforcement (internal reinforcement source), or the environment itself might provide actual rewards or punishers based on the the agent's choice (external source). In either case, reinforcement schedule can implicitly select the present target (Chadderdon et al., 2012). In future versions of

the model, however, we may utilize a premotor cortex representation in order to allow mappings to be learned which map the conjunction of cued target representation and limb state to directive motor commands. Such a representation would presumably be cued by dorsal visual stream information propagating through posterior parietal cortex when the agent views a target in a particular location in their visual field. This would also allow the model to move beyond the current limitation of being only able to retain a mapping to a single target at a given time.

Experiments have demonstrated that neuronal networks dynamically select between competing streams of information, depending on behavioral relevance (Kelemen and Fenton, 2010). This information selection is modulated via attention-like processes affecting neuronal dynamics and behavioral performance (Fenton et al., 2010). One dynamical mechanism implicated in attentional function is modulation of the level of oscillatory amplitude in the mu and alpha bands, elicited via top-down projections from higher- to lower-order brain areas (Mo et al., 2011; Jones et al., 2010). We previously developed models of neocortex showing altered dynamics with attentional modulation (Neymotin et al., 2011d). In these models, supragranular layers of neocortex received strengthened input, as a stand-in for higher-order brain area activation. This had the effect of increasing 8 – 12 Hz oscillation amplitude, while maintaining the peak oscillatory frequency location. We hypothesize that target information projecting from premotor- into supragranular layers of motor-cortex causes attentional modulation, allowing motor cortex to control movements to targets.

Learning molecules

A major challenge in neuroscience will be to bridge the gap in understanding how activity at disparate scales is linked (De Schutter, 2008; Lytton, 2008; Le Novère, 2007). A phenomenon such as learning has important dynamics at different scales of granularity ranging from molecular up to network and behavioral levels. In the present work, we utilized a phenomenological learning rule that had a spike-timing dependence. This rule operated at the synaptic scale and was further modulated by more global neuromodulatory-like reinforcement signals. These global reinforcement signals bridged the gap from synaptic and molecular signalling to the behavioral level and were effective in eliciting desired behavioral responses from the sensorimotor network via the synaptic learning process.

Dopamine, a key signaling molecule involved in modulating learning, bridges the gap between behavioral, cognitive, and molecular levels (Evans et al., 2012). There is evidence that increased (decreased) dopamine concentration leads to synaptic LTP (LTD) via action of D1-family receptors (Reynolds and Wickens, 2002; Shen et al., 2008). Our model provides a link between global reinforcement, mediated via dopamine signals, and sensorimotor learning. In future work, we will explore a more detailed model of the dopaminergic reward pathway, with potential implications for modeling disorders such as schizophrenia and Parkinson's disease (Frank et al., 2004; Cools, 2006; Frank and O'Reilly, 2006).

Acknowledgments

The authors would like to thank the reviewers for their helpful comments; Ashutosh Mohan for help with Fig. 5; Larry Eberle (SUNY Downstate) for Neurosim lab support; Michael Hines (Yale) and Ted Carnevale (Yale) for NEURON support; Tom Morse (Yale) for ModelDB support. Research supported by DARPA grant N66001-10-C-2008. The authors have no conflicts of interest to disclose.

References

Afshar, A., Santhanam, G., Yu, B., Ryu, S., Sahani, M., and Shenoy, K. (2011). Single-trial neural correlates of arm movement preparation. *Neuron*, 71(3):555–564.

Almassy, N., Edelman, G., and Sporns, O. (1998). Behavioral constraints in the development of neuronal properties: a cortical model embedded in a real-world device. *Cereb Cortex*, 8(4):346–361.

Bannister, A. (2005). Inter-and intra-laminar connections of pyramidal cells in the neocortex. *Neuroscience Research*, 53(2):95–103.

Berthier, N. (2011). The syntax of human infant reaching. In *8th International Conference on Complex Systems*, pages 1477–1487.

Berthier, N., Clifton, R., McCall, D., and Robin, D. (1999). Proximodistal structure of early reaching in human infants. *Exp Brain Res*, 127(3):259–269.

Carnevale, N. and Hines, M. (2006). *The NEURON Book*. Cambridge University Press, New York.

Chadderdon, G., Neymotin, S., Kerr, C., and Lytton, W. (2012). Reinforcement learning of targeted movement in a spiking neuronal model of motor cortex. *PLoS One*, 7(10):e47251.

Cools, R. (2006). Dopaminergic modulation of cognitive function-implications for l-dopa treatment in parkinson's disease. *Neurosci Biobehav Rev*, 30(1):1–23.

Corbetta, D. and Snapp-Childs, W. (2009). Seeing and touching: the role of sensory-motor experience on the development of infant reaching. *Infant Behav Dev*, 32(1):44–58.

De Schutter, E. (2008). Why are computational neuroscience and systems biology so separate? *PLoS Comput Biol*, 4(5):e1000078.

Dyhrfjeld-Johnsen, J., Santhakumar, V., Morgan, R., Huerta, R., Tsimring, L., and Soltesz, I. (2007). Topological determinants of epileptogenesis in large-scale structural and functional models of the dentate gyrus derived from experimental data. *J Neurophysiol*, 97(2):1566–1587.

Edelman, G. (1987). *Neural Darwinism: The theory of neuronal group selection*. Basic Books New York, New York.

Edelman, G. (2006). The embodiment of mind. *Daedalus*, 135(3):23–32.

Engel, A., Konig, P., Kreiter, A., Gray, C., and Singer, W. (1991). Temporal coding by coherent oscillations as a potential solution to the binding problem: physiological evidence. In Schuster, H., editor, *Nonlinear dynamics and neural networks*. VCH Verlagsgesellschaft, Weinheim.

Evans, R., Morera-Herreras, T., Cui, Y., Du, K., Sheehan, T., Kotaleski, J., Venance, L., and Blackwell, K. (2012). The effects of NMDA subunit composition on calcium influx and spike timing-dependent plasticity in striatal medium spiny neurons. *PLoS Comput Bio*, 8(4):e1002493.

Farries, M. and Fairhall, A. (2007). Reinforcement learning with modulated spike timing–dependent synaptic plasticity. *J Neurophysiol*, 98(6):3648–3665.

Fenton, A., Lytton, W., Barry, J., Lenck-Santini, P., Zinyuk, L., Kubík, Š., Bureš, J., Poucet, B., Muller, R., and Olypher, A. (2010). Attention-like modulation of hippocampus place cell discharge. *J Neurosci*, 30(13):4613–4625.

Florian, R. (2007). Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Comput*, 19(6):1468–1502.

Frank, M. and O'Reilly, R. (2006). A mechanistic account of striatal dopamine function in human cognition: psychopharmacological studies with cabergoline and haloperidol. *Behav Neurosci*, 120(3):497.

Frank, M., Seeberger, L., and O'Reilly, R. (2004). By carrot or by stick: cognitive reinforcement learning in parkinsonism. *Science*, 306(5703):1940–1943.

Gourevitch, B. and Eggermont, J. (2007). Evaluating information transfer between auditory cortical neurons. *J Neurophysiol*, 97(3):2533–2543.

Graybiel, A., Aosaki, T., Flaherty, A., and Kimura, M. (1994). The basal ganglia and adaptive motor control. *Science*, 265(5180):1826–1831.

Hikosaka, O., Nakamura, K., Sakai, K., and Nakahara, H. (2002). Central mechanisms of motor skill learning. *Current Opin Neurobiol*, 12(2):217–222.

Hosp, J., Pekanovic, A., Rioult-Pedotti, M., and Luft, A. (2011). Dopaminergic projections from midbrain to primary motor cortex mediate motor skill learning. *J Neurosci*, 31(7):2481–2487.

Houk, J. and Wise, S. (1995). Distributed modular architectures linking basal ganglia, cerebellum, and cerebral cortex: Their role in planning and controlling action. *Cereb Cortex*, 5(2):95–110.

Izhikevich, E. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb Cortex*, 17:2443–2452.

Jones, S., Kerr, C., Wan, Q., Pritchett, D., Hämäläinen, M., and Moore, C. (2010). Cued spatial attention drives functionally relevant modulation of the mu rhythm in primary somatosensory cortex. *J Neurosci*, 30(41):13760–13765.

Kelemen, E. and Fenton, A. (2010). Dynamic grouping of hippocampal neural activity during cognitive control of two spatial frames. *PLoS Biol*, 8(6):e1000403.

Kerr, C., Neymotin, S., Chadderdon, G., Fietkiewicz, C., Francis, J., and Lytton, W. (2012). Electrostimulation as a prosthesis for repair of information flow in a computer model of neocortex. *IEEE Trans Neural Syst Rehabil Eng*, 20:153–60.

Kerr, C., Van Albada, S., Neymotin, S., Chadderdon, G., Robinson, P., and Lytton, W. (2013). Cortical information flow in Parkinson's disease: a composite network/field model. *Front Comput Neurosci*, 7:39.

Krichmar, J. and Edelman, G. (2005). Brain-based devices for the study of nervous systems and the development of intelligent machines. *Artif Life*, 11(1-2):63–77.

Kubikova, L. and Kostál, L. (2010). Dopaminergic system in birdsong learning and maintenance. *J Chem Neuroanat*, 39(2):112–123.

Le Novère, N. (2007). The long journey to a systems biology of neuronal function. *BMC Syst Biol*, 1(1):28.

Luft, A. and Schwarz, S. (2009). Dopaminergic signals in primary motor cortex. *Int J Dev Neurosci*, 27(5):415–421.

Lungarella, M. and Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS Comput Biol*, 2(10):e144.

Lytton, W. (2008). Computer modelling of epilepsy. *Nature Rev Neurosci*, 9(8):626–637.

Lytton, W., Neymotin, S., and Hines, M. (2008a). The virtual slice setup. *J Neurosci Methods*, 171(2):309–315.

Lytton, W. and Omurtag, A. (2007). Tonic-clonic transitions in computer simulation. *J Clin Neurophys*, 24:175–181.

Lytton, W., Omurtag, A., Neymotin, S., and Hines, M. (2008b). Just-in-time connectivity for large spiking networks. *Neural Comput*, 20(11):2745–2756.

Lytton, W. and Stewart, M. (2005). A rule-based firing model for neural networks. *Int J Bioelectromagnetism*, 7:47–50.

Lytton, W. and Stewart, M. (2006). Rule-based firing for network simulations. *Neurocomputing*, 69(10-12):1160–1164.

Mahmoudi, B. and Sanchez, J. (2011). A symbiotic brain-machine interface through value-based decision making. *PloS One*, 6(3):e14760.

Marsh, B., Tarigoppula, A., and Francis, J. (2011). Correlates of reward expectation in the primary motor cortex: Developing an actor-critic model in macaques for a brain computer interface. *Society for Neuroscience Abstracts*, 41.

Mo, J., Schroeder, C., and Ding, M. (2011). Attentional modulation of alpha oscillations in macaque inferotemporal cortex. *J Neurosci*, 31(3):878–882.

Molina-Luna, K., Pekanovic, A., Röhrich, S., Hertler, B., Schubring-Giese, M., Rioult-Pedotti, M., and Luft, A. (2009). Dopamine in motor cortex is necessary for skill learning and synaptic plasticity. *PLoS One*, 4(9):e7082.

Neymotin, S., Jacobs, K., Fenton, A., and Lytton, W. (2011a). Synaptic information transfer in computer models of neocortical columns. *J Comput Neurosci*, 30(1):69–84.

Neymotin, S., Kerr, C., Francis, J., and Lytton, W. (2011b). Training oscillatory dynamics with spike-timing-dependent plasticity in a computer model of neocortex. In *2011 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6.

Neymotin, S., Lazarewicz, M., Sherif, M., Contreras, D., Finkel, L., and Lytton, W.

(2011c). Ketamine disrupts theta modulation of gamma in a computer model of hippocampus. *J Neurosci*, 31(32):11733–11743.

Neymotin, S., Lee, H., Park, E., Fenton, A., and Lytton, W. (2011d). Emergence of physiological oscillation frequencies in a computer model of neocortex. *Front Comput Neurosci*, 5:19.

Pastalkova, E., Serrano, P., Pinkhasova, D., Wallace, E., Fenton, A., and Sacktor, T. (2006). Storage of spatial information by the maintenance mechanism of LTP. *Science*, 313(5790):1141–1144.

Peterson, B., Healy, M., Nadkarni, P., Miller, P., and Shepherd, G. (1996). ModelDB: an environment for running and storing computational models and their results applied to neuroscience. *J Am Med Inform Assoc.*, 3(6):389–398.

Potjans, W., Morrison, A., and Diesmann, M. (2009). A spiking neural network model of an actor-critic learning agent. *Neural Comput*, 21(2):301–339.

Qiu, S., Anderson, C., Levitt, P., and Shepherd, G. (2011). Circuit-specific intracortical hyperconnectivity in mice with deletion of the autism-associated met receptor tyrosine kinase. *J Neurosci*, 31(15):5855–5864.

Reid, R. (2012). From functional architecture to functional connectomics. *Neuron*, 75(2):209–217.

Reynolds, J. and Wickens, J. (2002). Dopamine-dependent plasticity of corticostriatal synapses. *Neural Netw*, 15(4-6):507–521.

Roberts, P. and Bell, C. (2002). Spike timing dependent synaptic plasticity in biological systems. *Biol Cybern*, 87(5):392–403.

Rowan, M. and Neymotin, S. (2013). Synaptic scaling balances learning in a spiking model of neocortex. *Springer LNCS*, 7824:20–29.

Sanes, J. (2003). Neocortical mechanisms in motor learning. *Curr Opin Neurobiol*, 13(2):225–231.

Seung, H. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40(6):1063–1073.

Shadmehr, R. and Krakauer, J. W. (2008). A computational neuroanatomy for motor control. *Exp Brain Res*, 185(3):359–381.

Shadmehr, R. and Wise, S. (2005). *The computational neurobiology of reaching and pointing: a foundation for motor learning*. The MIT press. Cambridge, MA.

Shen, W., Flajolet, M., Greengard, P., and Surmeier, D. (2008). Dichotomous dopaminergic control of striatal synaptic plasticity. *Science*, 321(5890):848–851.

Shepherd, G. (2004). *The synaptic organization of the brain*. Oxford University Press, USA.

Sober, S. and Brainard, M. (2009). Adult birdsong is actively maintained by error correction. *Nat Neurosci*, 12(7):927–931.

Song, S., Miller, K., and Abbott, L. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nat Neurosci*, 3:919–926.

Sporns, O., Tononi, G., and Kotter, R. (2005). The human connectome: a structural description of the human brain. *PLoS Comput Biol*, 1(4):e42.

Thomson, A. and Bannister, A. (2003). Interlaminar connections in the neocortex. *Cereb Cortex*, 13(1):5–14.

Thomson, A., West, D., Wang, Y., and Bannister, A. (2002). Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2-5 of adult rat and cat neocortex: triple intracellular recordings and biocytin labelling in vitro. *Cereb Cortex*, 12:936–953.

Thorndike, E. (1911). *Animal intelligence*. New York: Macmillan.

Tiesinga, P. and Sejnowski, T. (2004). Rapid temporal modulation of synchrony by competition in cortical interneuron networks. *Neural Comput*, 16(2):251–275.

Uhlhaas, P. and Singer, W. (2006). Neural synchrony in brain disorders: relevance for cognitive dysfunctions and pathophysiology. *Neuron*, 52:155–168.

Von der Malsburg, C. and Schneider, W. (1986). A neural cocktail-party processor. *Biol Cybern*, 54:29–40.

von Hofsten, C. (1979). *Development of visually directed reaching: The approach phase*. Department of psychology, University of Uppsala [Psykologiska inst., Uppsala univ.].

Von Kraus, L., Sacktor, T., and Francis, J. (2010). Erasing sensorimotor memories via PKMζ inhibition. *PloS One*, 5(6):e11125.

Webb, B. (2000). What does robotics offer animal behaviour? *Animal Behav*, 60(5):545–558.