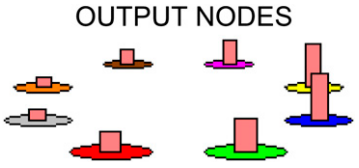
Supporting Information

Ajemian et al. 10.1073/pnas.1320116110

SI Methods

Neural Network. ***Input–output representation.*** Because the center–out reaching task is being simulated, the input to the network is one of eight target locations and the output is a corresponding movement command. For input representation, there are two input nodes, representing Cartesian coordinates of the target location normalized to between −1 and 1. For output representation, there are eight output nodes with a circular topography (the first node is a neighbor to the second and eighth nodes, etc.) (scheme below) normalized to between 0 and 1. The target values for the movement commands were set in a variety of different ways to ensure that the mapping as posed was arbitrarily nonlinear. In particular, some mappings were highly distributed, whereas others were discrete (full activation of a single output node only). The results were qualitatively the same in all cases. Note that, as mentioned in the text, no redundancy exists at the level of the output nodes. The actual motor output is compared against the desired motor output to determine whether there is an error. Therefore, redundancy at the level of spinal motor neurons and muscles/joints and beyond is not considered. The initial simulations treated the motor output as a dynamic command, that is, a control signal varying in time as the input to a skeletomuscular model of the arm. In this formulation, the feedback gain was adjusted so that the average feedback signal strength was ∼10% of the feed-forward command. However, given the tens of thousands of simulations to be run to test the effect of varying parameters (see below), the full-blown dynamical model became impractical to implement, and we simplified the problem to one of a static input–output mapping as described above. This simplification allowed us to straightforwardly vary the character of the system output from highly distributed (all output nodes active for each movement) to highly focal (only one output node activated for a single movement). Further, it allowed us to frame the learning problem in the most general terms possible—arbitrary function approximation—independent of the biomechanical properties of any specific end effector. To simulate feedback in the static case, learning was continued until the error decreased to less than 10% of the desired output, at which point feedback was considered to supplement the feed-forward command sufficiently to generate the correct output (other levels of error tolerance were also tried—see below). All of the simulation results run with the dynamic model in a hyperplastic and noisy mode agreed with the results from the static case.



Learning rule. Simple gradient descent is used for the learning algorithm with the standard update rule,

$$\Delta\vec{\mathbf{W}} = -\beta\nabla \mathrm{E}_f,$$

where the weight change is proportional to the gradient of the error function. Error is represented as the root mean squared error of the difference between the actual output vector and the target output vector. Additional simulations were run with a spontaneous decay term, $\Delta\vec{\mathbf{W}} = -\beta\nabla \mathrm{E}_f - \alpha\vec{\mathbf{W}}$, where α was generally set in the range of 0.000015 to keep the weights bounded from −1 to 1. The results were qualitatively the same in both cases.

Network architecture. The neural network used in the simulations shown is a standard multilayer perceptron with two hidden layers (for four total layers) and a sigmoidal transfer function. All nodes, except for the input nodes, assume positive values from 0 to 1. A variety of different transfer functions were implemented, as well as a variety of different network types, including composite multilayer perceptron/radial basis function networks. Results were qualitatively the same in all cases. In terms of network design, a large redundancy implies that the number of processing units in the hidden layers far exceeds the minimum number needed to achieve the desired mapping to the desired degree of accuracy. For the simulations shown, we used 12 hidden nodes (8 in the first layer and 4 in the second), which is at least four times as many nodes as are needed to implement the mappings we used. We did not increase the redundancy further because of the known problems that arise in multilayer perceptrons when the number of patterns becomes exceedingly small relative to network capacity. The redundancy was reduced to as few as 3 hidden nodes to observe the effects that lowered redundancy had on obtaining orthogonal solutions.

Learning rate and noise level. The learning rate is equal to the β term in the gradient descent equation. For networks with the architecture and input–output representations described above, the learning rate was varied between 0.005 and 0.5, a range of two orders of magnitude. Three types of network noise were considered: node perturbation, weight perturbation, and weight-change perturbation. Additive noise models were used with both constant and signal-dependent Gaussian noise. The reported simulations all used signal-dependent noise. For node perturbation and weight perturbation, signal-dependent noise levels from 5% to 30% were added to every node and weight (signal-dependent noise of **n**% is defined as adding to the signal a mean-zero Gaussian random process with a SD of **n**% of the signal strength). For weight-change perturbation, signal-dependent noise levels from 50% to 400% were added to every weight change (the weight-change process can tolerate higher levels of noise than the nodes or the weights because, for stochastic gradient descent, the gradient has only to point in the right direction on average). In general, the nonhyperplastic, noiseless regime was taken to have learning rates between 0.005 and 0.02, and the hyperplastic, noisy regime was taken to have learning rates between 0.1 and 0.3 and noise levels in the middle of the ranges supplied above. Of course, given the wide range of parameter variation, there were many combinations of noise levels and learning rates for which a given network did not meet the three required criteria (main text), but for all parameter combinations that resulted in a "high-noise" functioning network, the network behavior was qualitatively the same. Learning asymptote was taken to occur when >90% of the maximal possible reduction in error had occurred, leaving 10% of the mapping task to be accomplished through feedback mechanisms. Other levels of error at asymptote were implemented, all giving similar results.

Parameter variations. Each of the network features described above was parametrically varied, leading to a huge number of parameter combinations to test. Tens of thousands of simulations were run, covering as much of the parameter space as possible. This was done to establish that the results describe a qualitatively different type of network with a coherent set of characteristics, as opposed to an idiosyncratic network with a particular set of parameters. Indeed, although many of the

parameter choices did not result in functioning networks according to the three criteria delineated in the text, a class of networks with similar characteristics (i.e., the hyperplastic and noisy networks) was identifiable.

Geometric angle between intersecting solution manifolds. The angle of intersection between two skills was computed at a given time step as follows. An exact analytical expression was computed for the network gradient with respect to the current weight vector for each of the skills: $\nabla \mathbf{E}_f(\vec{\mathbf{W}})^{\mathcal{A}}$ and $\nabla \mathbf{E}_f(\vec{\mathbf{W}})^{\mathcal{B}}$. This was practical because we used fairly uncomplicated feed-forward architectures. The values for the network weights and nodal activations were then plugged in and the angle was computed as

$$\cos^{-1}\left(\nabla \mathbf{E}_f(\vec{\mathbf{W}})^{\mathcal{A}} \bullet \nabla \mathbf{E}_f(\vec{\mathbf{W}})^{\mathcal{B}}\right).$$

In this way, the magnitude and angle effects of the dynamic equilibrium condition were combined into a single measure for ease of analysis. The gradient norms could also be divided out to isolate each of those effects. Also note that this measure makes sense only when the error for both skills has asymptoted; otherwise, there is no intersection taking place, because an intersection means that the current network configuration is a (quasi-) solution for both skills. If the error for either skill is high, then the relative orientations of the two gradients cannot be interpreted and will randomly bounce around as is shown in the early trials of Fig. 2 *E* and *F*.

Simulation of behavioral results. To simulate the rotated mapping in the visuomotor perturbations, the target values were shifted topographically either "clockwise" or "counterclockwise" by the appropriate amount. The error—measured experimentally by angular deviation from a straight line path to the target at peak velocity—was determined in simulation from the vector error, using a normalized inner product. (Note that this angular error has nothing to do with angle between two solution manifolds.) The learning rate was set to 0.4 to match the behavioral finding that performance asymptoted after 18 cycles. At this learning rate, the noiseless simulations were ill-conditioned.

Simulation of dendritic spine turnover. These simulations were conducted using the same architectures/parameters as described above with two main differences. First, the connection strength between two network nodes was represented by a large number of potentially active binary-valued excitatory/inhibitory sites for dendritic spines, as opposed to a single continuous weight. We assumed 1×10^5 potential sites, so that a learning increase of, for example, 0.00083 resulted in the addition or activation of $0.00083 \times (1 \times 10^5)$ dendritic spines (and similarly for spine deactivation or removal). The exact number of potential sites does not matter if one is allowed to assume fractional spines, but we chose 10,000 simply because it has been estimated that neurons typically make roughly 10,000 synaptic contacts (1, 2). Second, we needed a rule for determining which dendritic spine sites, on a given trial, were activated or deactivated. We assumed that dendritic spines were added randomly and removed in the order in which they had been added. The purpose of removing spines in the order in which they were added is to comport with experimental findings that the longer a dendritic spine has been in existence, the more likely that spine is to persist in the face of new learning (3–5). Presumably, the longer-lasting spines have lasted as long as they have because they are a critical part of the basic synaptic scaffolding and are used for many tasks above and beyond the new one being learned. Spine growth, on the other hand, appears random and unpredictable. This assumption does not matter, however, because additional simulations were run in which spines were added in a fixed order, as opposed to randomly, and these trials gave qualitatively similar results to the ones reported (dendritic spine turnover was slightly reduced in comparison).

In a given simulation, the number of active dendritic spines was initially set randomly, and then the network learned how to make reaching movements in the absence of a perturbation (see above). Error decreased in the typical pseudoexponential form. Asymptote was defined in several possible ways, including the point after which subsequent error reduction was <10% (other definitions resulted in qualitatively similar results). When asymptote was reached, defined as $t = 0$, the total number of active dendritic spines was determined, as this group constitutes the pool of spines available for removal (or to which additions can be made) in the learning of a new task. The network was retrained to learn the visuomotor adaptation task. This condition represents the novel learning condition. The two-photon microscopy studies also indicate that there is a pool of spines that appears more fundamental to the overall network performance and hence is not available for learning and persists indefinitely (3–5). This pool constitutes 50–75% of the total number of spines. We conservatively chose the 75% value. Lower estimates would result in even higher percentages of synaptic turnover than are here reported. Thus, we have two groups of synaptic spines, one available for learning and one permanently in place. Next, learning continued until asymptote, and at each trial, the total number of spines added/removed was computed as a percentage of the total number of active dendritic spines at $t = 0$. The result is the black line in Fig. 5*B*. We also ran a control condition, the simple repetition condition, in which the visuomotor adaptation was learned to asymptote and then continued beyond asymptote. Counts of dendritic spine addition/removal in this group were made only after asymptote (the new $t = 0$). Simulation parameters (learning rate, network architecture, etc.) were varied as described previously with qualitatively similar results.

This model makes several simplifications. Whereas the studies in refs. 3–5 investigate spine growth and retraction as a function of time (during which learning trials are administered), we examine spine growth/retraction as a function of trial number. Further, no attempt is made to emulate the specific learning tasks in the two experiments (reaching for a food pellet in ref. 4 vs. remaining upright on a rotarod in ref. 5), alterations in the strengths of existing synapses are ignored, and no presynaptic component is included in the model. These caveats notwithstanding, the results are meaningful because the hyperplastic and noisy networks capture the basic phenomena under every choice of parameters, whereas the nonhyperplastic and noiseless networks fail to capture the same phenomena in an equally parameter-independent fashion. Thus, it can be concluded with some force that, at least under a connectionist framework, the observed dendritic spine dynamism is broadly consistent with one class of network but not with the other.

Sharpened postasymptotic tuning of model neurons. In a standard multilayer perceptron, some neurons will happen to be broadly tuned to movement direction. In each simulation run, we kept track of the neurons that were tuned at asymptote and assessed their tuning 20,000 trials later.

Motor Psychophysics. Seated subjects (men or women between the ages of 18 y and 30 y, all right-handed with normal vision and no history of neurological problems) grab a planar robotic manipulandum handle, which controls an onscreen computer cursor. The targets are arranged in a center–out fashion, with the peripheral targets 10 cm distant from the central target. One centimeter of handle movement corresponds to 1 cm of cursor movement. Full visual feedback of the cursor on the screen—although not of the hand on the manipulandum, which is blocked by an opaque board—is allowed. To ensure that movements are largely feed-forward in the presence of visual feedback, a speed

threshold is enforced so that subjects have to achieve a hand speed of greater than 55 cm/s to correctly perform the task. For the visuomotor perturbation, the direction of the cursor movement is rotated 60° from the direction of hand movement in the clockwise or the counterclockwise direction. The only other instructions given to the subject were to move to the target as fast as possible.

Twelve subjects were used with a 60° clockwise visuomotor perturbation for movements to two of the eight targets (Fig. 4*B*), $\mathbf{D_1}$ and $\mathbf{D_2}$. Each subject was presented with exactly the same sequence of targets outlined in Fig. 4*C*. Cycles consisted of eight targets that were either all $\mathbf{D_2}$ or pseudorandomly split between $\mathbf{D_1}$ and $\mathbf{D_2}$. The exceptions were cycles 34 and 98, each of which consisted of four targets, all $\mathbf{D_1}$. Error (angular deviation at peak velocity) was computed separately in each cycle for each target as the average angular error across the eight or four movements to that target. The exceptions again were cycles 34 and 98, where single movements were analyzed to discern the finer-grain time course of relearning. The experiment was briefly paused to explicitly inform subjects when the target was going to switch from $\mathbf{D_2}$ to $\mathbf{D_1}$ after the performance of 120 consecutive movements to $\mathbf{D_2}$. This transition occurred twice, once at the beginning of cycle 34 and once at the begiinning of cycle 98. The intervention was necessary to ensure that a jump in error did not result from a surprise in the target location, because subjects may well have habituated to the repeated presentation of a single target and thus prepared, in advance of the actual target presentation, a movement to the wrong target. To this end, there was a 5-s pause in advance of the target switch, during which time subjects were told the identity of the next target ($\mathbf{D_1}$). Furthermore, to make absolutely certain that no preparatory activity was leaking into the movement commands, subjects were also told not to resume movement to the next target until after they were verbally cued to do so, and this signal came 1 s after the presentation of the $\mathbf{D_1}$ target. This intervention was identically performed for each transition. Each of the three sessions (Fig. 4*C*) took ~15 min to complete, with a 1- to 2-min break in between. Three subjects were tested using a 60° counterclockwise rotation to confirm the similarity of the results. In each of the 15 subjects, there was a clearly visible jump in error that occurred at the beginning of cycle 34, the time of the first target switch from $\mathbf{D_2}$ to $\mathbf{D_1}$.

Several studies have already shown (e.g., refs. 6 and 7) that the learning of the visuomotor rotation does indeed transfer across neighboring targets, specifically from $\mathbf{D_2}$ to $\mathbf{D_1}$. However, two additional subjects were used to replicate this finding. The visuomotor rotation was learned until asymptote for $\mathbf{D_2}$ alone and then the transfer of this learning to $\mathbf{D_1}$ was assessed. The procedure was reversed on a different two subjects to test for the transfer of learning from $\mathbf{D_1}$ to $\mathbf{D_2}$. In both cases, roughly 75% of the learning transferred to the nearby target, meaning that the error was reduced by 75% compared with that of naive subjects who had no previous practice on the visuomotor rotation task. The fact that learning transfers at a high level to a neighboring target is consistent with the results from other studies.

SI Results

Geometry and Redundancy in High-Dimensional Weight Spaces. The theory of this paper relies critically on an assumption: So much redundancy is contained in the neural circuits for sensorimotor control that orthogonal solutions always exist, and these solutions are themselves high-dimensional submanifolds in weight space. The idea that redundancy facilitates nonoverlapping pattern storage in a neural network is not new, as it constitutes the essence of the Marr–Albus theory of pattern recoding in the cerebellum (8, 9) and Kanerva's theory of sparse distributed memory (10). According to Albus, the mapping from mossy fiber activation patterns to activity patterns across the granule cell layer embodies a highly redundant expansion recoding scheme. If the mossy fiber pattern capacity is approximated as 2^N and the granule cell layer pattern capacity as 100^N, Albus notes on p. 37,

> 2^N possible input patterns can be mapped very sparsely onto 100^N possible association [granule] cell patterns. If this is done randomly, the association cell patterns are likely to be highly dissimilar and thus easily recognizable. The ratio $100^N/2^N$ rapidly increases as N becomes large.

In this context, the term "highly dissimilar and thus easily recognizable" means, essentially, orthogonal. What is unique about our approach is that we extend this idea of redundancy-driven orthogonality in high-dimensional representational spaces to the dynamics of the learning process in even higher-dimensional weight spaces. To that end, we move beyond an examination of the metrical properties of input/output pairings in high-dimensional representational spaces to exploring the differential geometry of solution manifolds in even higher-dimensional configuration spaces. For this reason, we focus not on the size of the input/output pattern spaces, but on the dimensionality of the underlying configuration space (i.e., number of synapses).

An exact calculation of the degree of redundancy across any component of the human sensorimotor circuit is impossible. However, we can formulate the problem and make a rough calculation as follows. Suppose that there are **N** weights, **k** output nodes (interpreted as muscles, muscle fibers, or even α-motoneurons), and **P** input–output patterns. The function approximation constraint for a single pattern can be written as follows:

$$\begin{aligned} f\left(w_1, w_2, \ldots, w_N, \vec{\mathbf{X}}^p\right) &= \vec{\mathbf{T}}_1^p \\ f\left(w_1, w_2, \ldots, w_N, \vec{\mathbf{X}}^p\right) &= \vec{\mathbf{T}}_2^p \\ &\vdots \\ f\left(w_1, w_2, \ldots, w_N, \vec{\mathbf{X}}^p\right) &= \vec{\mathbf{T}}_{\mathbf{k}}^p. \end{aligned} \qquad \textbf{[S1]}$$

Given that there are **N** weights, the system has a redundancy of **N** – **k**, meaning that the solution manifold is an (**N** – **k**)-dimensional manifold embedded in an **N**-dimensional space. Consider the overlap of two distinct solution manifolds. As long as these manifolds have a dimension of greater than 1/2 of the dimension of the entire space—i.e., (**N** – **k**) > **1/2N**—then they have to overlap in at least (**N** – 2**k**) dimensions. Similarly, for all **P** patterns, the manifolds have to overlap in at least (**N** – **Pk**) dimensions. Is this a large number for biologically plausible values of **N**, **k**, and **P**? Is $\mathbf{N} \gg \mathbf{Pk}$?

The answer is yes. Suppose we consider **N** to be the number of synapses in the human cerebral cortex, **k** to be the total number of muscles in the human body, and **P** to be the total number of skills a human can learn. **N** is estimated at roughly 1×10^{14} synapses (1, 2). There are fewer than 1×10^3 muscles in the entire human body. Assume the number of skills that a person can perform at any one time is 1×10^5 (probably a high estimate). With these estimates, **N** is still greater than **Pk** by a factor of 1×10^6! Obviously, there are many ways to alter these estimates. Nonetheless, even these crude calculations strongly suggest that there is more than enough redundancy in neural circuits to support high-dimensional orthogonal solutions. In fact, our computation likely underestimates the amount of redundancy in several key respects. For example, given that there are more neurons in the cerebellum than in the rest of the brain (11) and that the cerebellum is a densely connected

sensorimotor-related structure, our estimate of **N** could be substantially increased.

Reducing Ill-conditioned Oscillations by Adding Noise. When the learning rate is set too high for a noiseless network, the configuration of the network can often bounce back and forth across opposing sides of an error valley, without converging smoothly to the locally optimal solution. This effect—that of being "trapped" in a type of oscillatory limit-cycle behavior within a fixed region of weight space such that sudden increases in error can occur at asymptote when switching patterns—is known as the cross-stitching effect (12). The existence of such error valleys constitutes a form of ill-conditioning and will arise naturally in neural networks for a variety of reasons (13). The addition of noise mitigates this problem by displacing the network in directions perpendicular to the error gradient, so that the current configuration can "escape" ill-conditioned regions of the workspace until arriving at better-conditioned regions. As described in *SI Methods*, this positive effect of noise was demonstrated for a variety of network architectures and parameter choices. The minimum performance threshold for ill-conditioning will tend to become more horizontal as the learning rate increases: At some point, the learning rate will be sufficiently large, relative to the typical contours of the error surface, that the network will be ill-conditioned regardless of the effects of noise.

Of course, there are higher-order (and nonbiological) numerical optimization methods of finding a solution (e.g., Newton's method, conjugate gradient method, etc.), and these methods do not require excessive iteration and so skirt the issue of a learning rate. But these approaches do not consider the orthogonality of solutions either. The bottom line is that high noise/high learning rate gradient descent is an effective means of exploring the workspace to find orthogonal solutions.

Orthogonal Solution Exhibits Lowest Present and Future Error. Consider two skills $\mathcal{A}$ and $\mathcal{B}$. If both skills have been learned, the weights must be configured, by definition, somewhere in the intersecting region of the two manifolds. The error will be minimal for the performance of either skill. But for a non-stationary network, there is the added problem of how the error for one skill will change, on average, as a function of practicing/performing the other skill. To address this issue, one must evaluate the weight change as a function of the current network configuration and the noise-induced error that will occur. The direction of weight change is determined by the gradient of the network output taken with respect to the network weight vector and evaluated at the input vector corresponding to the skill in question: $\nabla \mathbf{E}_f\left(\vec{\mathbf{W}}\right)^{\mathcal{A}}$ and $\nabla \mathbf{E}_f\left(\vec{\mathbf{W}}\right)^{\mathcal{B}}$. Suppose that these two vectors are orthogonal; i.e.,

$$\nabla \mathbf{E}_f\left(\vec{\mathbf{W}}\right)^{\mathcal{A}} \bullet \nabla \mathbf{E}_f\left(\vec{\mathbf{W}}\right)^{\mathcal{B}} = 0\,. \qquad \textbf{[S2]}$$

Performance of skill $\mathcal{A}$ will result in a deterministic component of weight change in the direction of the first gradient, either positive or negative depending on the sign of the error. In either case, due to the orthogonality condition, the net change in error of the second skill is zero. In the case of non-orthogonality, the second gradient will have a deterministic component that projects in the direction of the first gradient, so that when the weights change for skill $\mathcal{A}$, there will be a net increase in error for skill $\mathcal{B}$. (Note that due to noise in the weight change process, the weights will not change perfectly in the direction of the gradient, but only on average in the direction of the gradient.) If there are in actuality multiple skills, then it must be the case that every skill manifold is orthogonal to every other skill manifold, i.e., the orthogonality condition is satisfied on a pairwise basis for each pair of skills in a skill set. Otherwise, error will accumulate in the nonrehearsed skills. Therefore, in a distributed and non-stationary network asked to learn multiple skills with the goal of being able to perform those skills in any arbitrary order, the state of lowest error requires that the network assume a configuration where specific relationships are maintained across each skill pair (orthogonality), in addition to minimizing the error for each skill.

Derivation of Orthogonality Constraint for Feed-Forward Networks. Assume gradient descent learning ($\Delta\vec{\mathbf{W}} = -\beta\nabla\mathbf{E}_f$) and a quadratic error function for each skill, $\mathcal{A}_j$, so that $\mathbf{E}_f = \frac{1}{2}\sum_{\mathrm{k}}\left(\mathrm{T}_{\mathrm{k}}^{\mathcal{A}_j} - \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}\right)^2$, where k denotes output node. Solving for the gradient yields

$$-\nabla\mathbf{E}_f^{\mathcal{A}_j} = \sum_{\mathrm{k}}\epsilon_{\mathrm{k}}\frac{\partial \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}}{\partial\vec{\mathbf{W}}}\,, \qquad \textbf{[S3]}$$

where ϵ_{k} is the deterministic feed-forward error of the network (i.e., the error if there were no noise and no feedback) at the kth output node. As indicated previously, this error will be small, but not zero and not insignificant given the critical role of feedback in biological motor control. Further, both the sign and the absolute value of ϵ_{k} are fixed, yet indeterminate, as they depend upon the current network configuration (which obviously depends upon a host of unknown factors). Consider, therefore, ϵ_{k} to be a hidden variable. Inserting the expression above into the condition of dynamic equilibrium (Eq. **1**) yields

$$\left(\forall i, j : i \neq j\right)\left(\sum_{\mathrm{k}}\epsilon_{\mathrm{k}}\frac{\partial \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}}{\partial\vec{\mathbf{W}}}\right)\bullet\left(\beta\sum_{\mathrm{k}}\left(\mathrm{T}_{\mathrm{k}}^{\mathcal{A}_i} - \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_i}\right)\frac{\partial \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_i}}{\partial\vec{\mathbf{W}}}\right) \cong 0\,, \qquad \textbf{[S4]}$$

where the absolute error plus the feedback-compensated error at the kth node for pattern is embodied in the term $\left(\mathrm{T}_{\mathrm{k}}^{\mathcal{A}_j} - \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}\right)$. Both the sign and the value of this term are indeterminate because the network is suffused, during each trial, with independent zero-mean noise processes of significant magnitude. (Of course, the feedback is corrupted by noise as well but no attempt is made to model this element of the problem.) Let $\mu_{\mathrm{k}} = \left(\mathrm{T}_{\mathrm{k}}^{\mathcal{A}_j} - \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}\right)$. Because the value assumed by μ_{k} depends, in part, upon the noise profile, μ_{k} can be considered a random variable. (Because significant levels of noise are present throughout the network and because weight values are both positive and negative, it is reasonable to assume that μ_{k} is a zero-mean random variable.) Combining terms yields

$$\left(\forall i, j : i \neq j\right)\left(\beta\sum_{\mathrm{k}}\sum_{\mathrm{k}'}\epsilon_{\mathrm{k}}\mu_{\mathrm{k}}\frac{\partial \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}}{\partial\vec{\mathbf{W}}}\bullet\frac{\partial \mathrm{Z}_{\mathrm{k}'}^{\mathcal{A}_i}}{\partial\vec{\mathbf{W}}}\right) \cong 0\,, \qquad \textbf{[S5]}$$

Assuming that μ_{k} is a random variable, the equation above is guaranteed to be satisfied only when each of the individual terms inside the summand becomes small (same as Eq. **4**),

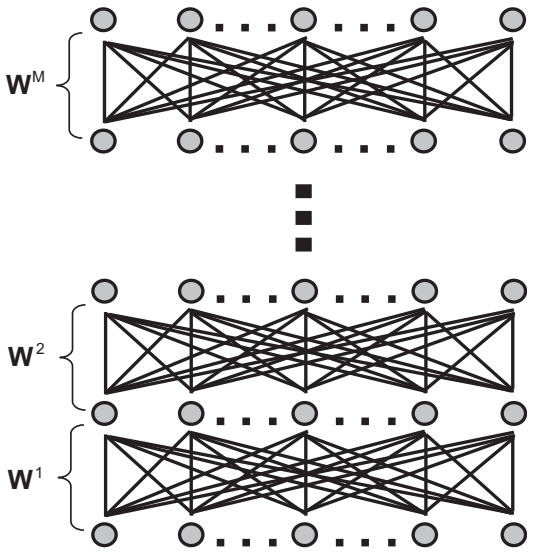
$$\left(\forall i, j : i \neq j\right)\ \left(\forall k, k'\right)\ \frac{\partial \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}}{\partial\vec{\mathbf{W}}}\bullet\frac{\partial \mathrm{Z}_{\mathrm{k}'}^{\mathcal{A}_i}}{\partial\vec{\mathbf{W}}} \cong 0\,, \qquad \textbf{[S6]}$$

where $\frac{\partial \mathrm{Z}_{\mathrm{k}}^{\mathcal{A}_j}}{\partial\vec{\mathbf{W}}}$ is the gradient of a network output node with respect to the network's entire weight vector evaluated for a specific skill. Intuitively what this means is that in a hyperplastic and noisy network, the products of gradients cannot vanish simply by making the errors small, as happens for a nonhyperplastic and noiseless network. Expanding the expression yields

$$(\forall i,j : i \neq j)\ (\forall k,k')\quad \begin{pmatrix} \frac{\partial Z_k}{\partial w_1} \\ \frac{\partial Z_k}{\partial w_2} \\ \vdots \\ \frac{\partial Z_k}{\partial w_N} \end{pmatrix}^{\mathcal{A}_j} \bullet \begin{pmatrix} \frac{\partial Z_{k'}}{\partial w_1} \\ \frac{\partial Z_{k'}}{\partial w_2} \\ \vdots \\ \frac{\partial Z_{k'}}{\partial w_N} \end{pmatrix}^{\mathcal{A}_i} \cong 0\,, \qquad \textbf{[S7]}$$

where N is the total number of weights contained in the entire network.

A feed-forward network can be expressed as an ordered series of network modules, with each module containing two network layers and the associated weight matrix, **W**, that links the two layers. The scheme below pictorially captures this description, assuming that there are $M + 1$ layers with M modules and M associated weight matrices.



The gradient above, expanded in terms of N individual weights, can be alternatively represented as a concatenation of M vectors of partial derivatives:

$$(\forall i,j : i \neq j)\ (\forall k,k')\quad \begin{pmatrix} \frac{\partial Z_k}{\partial \mathbf{w}^1} \\ \frac{\partial Z_k}{\partial \mathbf{w}^2} \\ \vdots \\ \frac{\partial Z_k}{\partial \mathbf{w}^M} \end{pmatrix}^{\mathcal{A}_j} \bullet \begin{pmatrix} \frac{\partial Z_{k'}}{\partial \mathbf{w}^1} \\ \frac{\partial Z_{k'}}{\partial \mathbf{w}^2} \\ \vdots \\ \frac{\partial Z_{k'}}{\partial \mathbf{w}^M} \end{pmatrix}^{\mathcal{A}_i} \cong 0\,. \qquad \textbf{[S8]}$$

A vector of these partial derivatives can be computed for the matrix of weights schematized in Fig. 6. There are $R \times S$ total weights in this matrix, and we know that for a given weight

$$\frac{\partial Z_k}{\partial w_{r\,s}} = \frac{\partial Z_k}{\partial u_s}\, g'(I_s)\nu_r. \qquad \textbf{[S9]}$$

Plugging in this expression and expanding yields

$$(\forall i,j : i \neq j)\ (\forall k,k')\quad \begin{pmatrix} \frac{\partial Z_k}{\partial u_1} g'(I_1)\nu_1 \\ \frac{\partial Z_k}{\partial u_2} g'(I_2)\nu_1 \\ \vdots \\ \frac{\partial Z_k}{\partial u_S} g'(I_S)\nu_1 \\ \frac{\partial Z_k}{\partial u_1} g'(I_1)\nu_2 \\ \vdots \\ \frac{\partial Z_k}{\partial u_1} g'(I_1)\nu_R \\ \vdots \\ \frac{\partial Z_k}{\partial u_S} g'(I_S)\nu_R \end{pmatrix}^{\mathcal{A}_j} \bullet \begin{pmatrix} \frac{\partial Z_{k'}}{\partial u_1} g'(I_1)\nu_1 \\ \frac{\partial Z_{k'}}{\partial u_2} g'(I_2)\nu_1 \\ \vdots \\ \frac{\partial Z_{k'}}{\partial u_S} g'(I_S)\nu_1 \\ \frac{\partial Z_{k'}}{\partial u_1} g'(I_1)\nu_2 \\ \vdots \\ \frac{\partial Z_{k'}}{\partial u_1} g'(I_1)\nu_R \\ \vdots \\ \frac{\partial Z_{k'}}{\partial u_S} g'(I_S)\nu_R \end{pmatrix}^{\mathcal{A}_i} \cong 0\,. \qquad \textbf{[S10]}$$

This inner product of two vectors with $R \times S$ terms can be more succinctly written as the product of a sum of R terms and a sum of S terms:

$$(\forall i,j : i \neq j)\ (\forall k,k')\ \left(\sum_{r=1}^{R} \nu_r^{\mathcal{A}_j} \nu_r^{\mathcal{A}_i}\right)\left(\sum_{r=1}^{S}\left(\frac{\partial Z_k}{\partial u_S} g'(I_S)^{\mathcal{A}_j}\right)\left(\frac{\partial Z_{k'}}{\partial u_S} g'(I_S)^{\mathcal{A}_i}\right)\right) \cong 0\,. \qquad \textbf{[S11]}$$

Each of these sums can itself be written as an inner product that, combined with the fact that $\frac{\partial Z_k}{\partial u_S} g'(I_S) = \frac{\partial Z_k}{\partial I_S}$, yields Eq. **5** in the main text.

Bicycle Riding and Orthogonality. It is important to realize that the orthogonality defined in this paper concerns mappings from inputs to outputs—that is, what is orthogonal are specific instances of a sensorimotor function (i.e., [a→b] is orthogonal to [c→d]), as opposed to specific motor outputs (b vs. d). One might be tempted to say that bicycle riding is not orthogonal to other skills that are routinely performed (such as walking up or down stairs), because many of the same muscles are recruited in similar sequences. This analysis would be incorrect, however, because it ignores the sensory components of bicycle riding, which include not only proprioceptive, tactile, and visual feedback, but also prominent vestibular feedback (a form of feedback not always engaged so critically). These modes of sensory feedback, together with task-based rules for corrective actions/feedback, collectively distinguish bicycle riding from other motorically similar tasks. If one grants that bicycle riding is orthogonal to other routinely performed tasks, then even though the network configuration is constantly moving in weight space, these movements are, on average, roughly parallel to the local tangent plane of the "bicycle-riding manifold" (Fig. 3*C*), leaving the input–output mapping for that skill, on average, unchanged. The skill will degrade over time, but at an extremely slow rate. A more quantitative investigation into the model's temporal dynamics of forgetting is required (e.g., how the speed of forgetting relates to the specific degree of similarity between the learned tasks and other routinely performed tasks).

Definitions of Hyperplasticity. Because inputs and outputs can be scaled, normalized, or arbitrarily transformed/processed in a variety of ways to solve the same learning problem and because networks can exhibit a multitude of architectural differences (including, for example, the number of layers), the numerical value of the learnng rate, β, cannot, by itself, be used to de-

termine whether a network is hyperplastic. For the fairly typical networks we used—multilayer perceptrons with normalized inputs and outputs—typical values of β for nonhyperplastic networks range between 0.005 and 0.02, whereas for a hyperplastic network, β ranges from 0.1 to 0.5. In other networks solving different problems, the correspondence between β and network behavior will also be different. Ultimately, hyperplasticity is determined by the properties of the network, and to that end we provide five operational definitions that link the learning characteristics of the network to intrinsic network dynamics:

i) Weight values never converge even when the behavioral error asymptotes.
ii) At behavioral asymptote, the variance of the weight change in the direction of the gradient is much larger than the mean weight change in that direction, indicating a transition from stage I learning to stage II learning.
iii) Ill-conditioning results if a system with noise is suddenly made noiseless.
iv) When a skill is first learned, a typical weight displacement in a typical region of weight space is not insignificant relative to the contours of the error surface, so that weight changes during early learning are substantial.
v) Practice order has a large effect on the shape of the learning curves.

Conventional neural networks do not exhibit these properties and, in fact, they have been explicitly designed and implemented to avoid exhibiting some of these properties, which are considered more "bugs" than "features" from the perspective of machine learning applications.

SI Discussion

Examples of External vs. Internal Redundancy. In the case of external system redundancy, different combinations of peripheral motor variables are used to obtain the same behavioral goal. For example, different trajectories in joint space bring the hand to the same target location, and different patterns of muscle activation lead to the same trajectory in joint space. At the even higher level of tool use, many articulations of tool kinematics can lead to the same behavioral goal (14). By internal system redundancy, it is meant that different combinations of internal system parameters that are not as easy to measure, such as synaptic connection strengths or patterns of neural activity, engender the same motor output. Examples include different patterns of cortical firing rates leading to an identical spinal input or different patterns of synaptic connectivity in the cortex leading to an identical cortical output.

Applicability of Conclusions to More Realistic Networks. In real neural networks with recurrent connectivity and top–down feedback, it would not be possible to cast Eq. **5** in such a simplified form, because the derivative of an output node's activity with respect to a given weight would depend upon a complex interdependency of all of the weights and activities throughout the network. Further, even beyond issues of connectivity, the connectionist neural networks that are used in this paper lack many of the properties of real neural systems. We use continuous-valued nodes, rather than actual spiking neurons with ion conductances, cable equations for signal transmission, etc. We do not include the temporal dynamics of synaptic plasticity, as embodied in the principle of spike-timing–dependent plasticity (SDTP). We use gradient-descent learning that contains a global error signal implausibly back-propagated throughout the network, as opposed to more realistic unsupervised or reinforcement-based learning principles based solely on local neuronal interactions. Our networks are feed-forward with neither recurrent connections within a layer nor top–down feedback across layers. As stated in the Introduction, these simplifications were implemented to make the analyses more tractable. However, the question remains as to how the results on skill learning and the neurophysiological consequences of orthogonality apply to these more realistic scenarios.

Our focus is on the ability of a system to adapt its own internal parameters for the purpose of accomplishing an arbitrary functional approximation task. From this perspective, the use of spiking neurons would make little sense because dozens of parameters would have to be added for each neuron, and most of these parameters would not impact the synaptic connectivity of the neuron to its neighbors. (A thorough exploration of parameter space, as was performed with our simplified continuous-valued nodes, would become practically impossible with spiking neurons.) SDTP does affect connectivity, as it describes how a synapse between two neurons is modified in a Hebb-like fashion, depending upon the delay between pre- and postsynaptic firing. Nonetheless, for the higher-order pattern-mapping problems that are the subject of this work, the goal is to simultaneously change all of the network synapses in a coordinated fashion—as opposed to modeling the details of individual synaptic change—to reduce the overall input–output error. Regarding the use of collective weight update rules other than gradient descent, every learning algorithm, whether supervised or unsupervised, must induce some component of weight change in the direction of the gradient (otherwise, it would not work). For this reason, the core results would still apply, although it is reasonable to assume that (*i*) network exploration would have to increase to support the same amount of learning (all other factors being equal) and (*ii*) the component of change not directed along the gradient could dilute the neurophysiological consequences derived from pure gradient descent learning.

Recurrent connectivity and top–down feedback provide a dramatically different architecture for information flow within a network and, thus, will significantly impact the neurophysiological consequences of learning. However, previous computational studies indicate that these two forms of network interaction will likely enhance the tendency for neural tuning to sharpen with increasing level of expertise, thereby accentuating the formation of specialized "task-specific" representations. In particular, a recurrent feedback architecture of lateral on-center excitation and off-surround inhibition acts as a sharpening mechanism used specifically in models of orientation tuning (15) and more generally as an all-purpose form of sensory processing (16). Pattern stabilization induced by the top–down intracortical feedback that is ubiquitous throughout neural circuits (17) will also enhance representational segregation. To these points, we note that neural sharpening was achieved in our simulations without such mechanisms, i.e., using only a standard multilayer perceptron. Their inclusion would invariably lead to significantly enhanced sharpening. From this perspective, recurrent connectivity and top–down feedback may, in addition their other functions, serve as specific circuit embodiments designed to facilitate the more general principle of hyperplasticity-induced network orthogonalization.

Tuning to Sensorimotor and Even Sensory Parameters. The idea of experience-driven shaping of cellular response is additionally supported by two studies demonstrating that highly abstract elements of motor task requirements, generally thought to be represented in upstream neural circuits, can be reflected in the activation of primary motor cortical neurons if enough training has occurred. Specifically, neurons in M1 have been shown to encode (*i*) serial order in a sequence-based movement task (18) and (*ii*) task mode in a mode-switching movement task (19). Critical to both experiments is the length of training: The animals were trained on a single task for at least 1.5 y and as many as 3 y. If we extend this reasoning even further, it may be possible for M1 neurons to demonstrate tuning to any type of sensorimotor parameter—even purely sensory cues like color (20)—if

that parameter is made critically salient to the trained movement task and if sufficient training takes place.

Declarative Memory Explanation. Even if one allows that motor memories are stored by hyperplastic and noisy networks forming orthogonal representations, it would seem implausible that the same approach could apply to declarative memory formation. After all, declarative memory embodies the retention of information in the form of specific facts or events that must, by definition, remain invariant across time. Thus, no flexibility or redundancy (at least external redundancy) exists in the concept of declarative memory, as the sole systems-wide constraint is to recall the original information accurately. A network with perpetually fluctuating synaptic connections would appear, on its face, incompatible with the requirement of veridical recall. However, we argue that hyperplastic and noisy networks may constitute a viable means for the formation/retention of declarative memories. There are two elements to this argument:

i) In extremely high-dimensional spaces, random vectors are almost always orthogonal. A declarative memory is often defined by a proscribed association of random elements.
ii) Even the so-called "one-shot" episodic memory is not really one shot.

Consider an episodic memory, such as the recollection of what a person did on their 21st birthday. The memory will consist of numerous features bound together, including the nature of the occasion along with multiple sensory and other cues about the location and the events that transpired. A critical point is that these cues do not reflect any overt causal coupling in the environment. It did not have to be the case that on one's 21st birthday, one attended a Red Sox game and sat in the bleachers with 10 friends while it was drizzling, and one was drinking beer legally in public for the first time. However, that is how it happened, and so these disparate representations are all bound together to form a unitary memory. Such a memory, by definition, constitutes an arbitrary association (arbitrary in the sense that it could have been different) between multiple neural representations. It is a well-established fact of high-dimensional geometry that random vectors in high-dimensional spaces are almost always orthogonal. Therefore, each memory is likely to be orthogonal to all other memories. Given this orthogonality, the memory will decay through pervasive synaptic turnover—particularly as the different constitutive representations are engaged through other experiences—but only at a rate so slow that effective mnemonic permanence is maintained. This argument for declarative memories being quasi-permanent is analogous to the argument of never forgetting how to ride a bicycle. Second, for many memories, even one-shot memories, there is the possibility that although the experience may only occur once, it is recalled multiple times. Each time it is recalled, "learning" may take place in that the association is reengaged. Both of these arguments are speculative, and further quantitative investigation into the model's temporal dynamics of forgetting is required to more rigorously support the viability of the proposed approach.

If random vectors in high-dimensional spaces are almost always orthogonal, then why are most motor memories not automatically orthogonal from their inceptions? Why is there any need for using practice to orthogonalize different skills? The answer is that motor memories, unlike declarative memories, do not consist of arbitrary associations between multiple representations. Rather, a motor memory for a given skill is constrained by the sensorimotor requirements of the task at both ends of the sensorimotor spectrum: An underspin backhand and a backhand volley share similar kinematic movment plans as well as similar muscle output patterns. By definition, then, these two sensorimotor associations are not randomly related to one another in the space of sensorimotor associations. Indeed, they are guaranteed to be highly similar and only by reorganizing the sensorimotor system's vast number of internal degrees of freedom can these two skills be segregated from one another for the purpose of distinct recall. From this perspective, the art of skill learning lies in the sensorimotor system's ability to make dissimilar what must be a priori similar as a result of having to funnel all motor behaviors through the constraint of a limited set of actuating elements, i.e., a fixed skeletomusculature.

1. Shepherd GO (1998) *The Synaptic Organization of the Brain* (Oxford Univ Press, New York).
2. Koch C (1999) *Biophysics of Computation. Information Processing in Single Neurons* (Oxford Univ Press, New York).
3. Holtmaat AJ, et al. (2005) Transient and persistent dendritic spines in the neocortex in vivo. *Neuron* 45(2):279–291.
4. Xu T, et al. (2009) Rapid formation and selective stabilization of synapses for enduring motor memories. *Nature* 462:915–919.
5. Yang G, Pan F, Gan WB (2009) Stably maintained dendritic spines are associated with lifelong memories. *Nature* 462:920–924.
6. Krakauer JW, Pine ZM, Ghilardi MF, Ghez C (2000) Learning of visuomotor transformations for vectorial planning of reaching trajectories. *J Neurosci* 20(23):8916–8924.
7. Krakauer JW, Ghez C, Ghilardi MF (2005) Adaptation to visuomotor transformations: Consolidation, interference, and forgetting. *J Neurosci* 25(2):473–478.
8. Marr D (1969) A theory of cerebellar cortex. *J Physiol* 202(2):437–470.
9. Albus JS (1971) A theory of cerebellar function. *Math Biosci* 10(2):25–6l.
10. Kanerva P (1988) *Sparse Distributed Memory* (MIT Press, Cambridge, MA).
11. Kandel ER, Schwartz JH, Jessel TM (1999) *Principles of Neural Science* (McGraw-Hill, New York).
12. Reed RD, Marks RJ (1999) *Neural Smithing* (MIT Press, Cambridge, MA).
13. Saarinen S, Bramley R, Cybenko G (1993) Ill-conditioning in neural network training problems. *SIAM J Sci Comput* 14:693–714.
14. Müller H, Sternad D (2004) Decomposition of variability in the execution of goal-oriented tasks: Three components of skill improvement. *J Exp Psychol Hum Percept Perform* 30(1):212–233.
15. Ferster D, Miller KD (2000) Neural mechanisms of orientation selectivity in the visual cortex. *Annu Rev Neurosci* 23:441–471.
16. Grossberg S (1980) How does a brain build a cognitive code? *Psychol Rev* 87(1):1–51.
17. Grossberg S (1982) *Studies of Mind and Brain* (Kluwer Academic, Amsterdam).
18. Carpenter AF, Georgopoulos AP, Pellizzer G (1999) Motor cortical encoding of serial order in a context-recall task. *Science* 283(5408):1752–1757.
19. Matsuzaka Y, Picard N, Strick PL (2007) Skill representation in the primary motor cortex after long-term practice. *J Neurophysiol* 97(2):1819–1832.
20. Zach N, Inbar D, Grinvald Y, Bergman H, Vaadia E (2008) Emergence of novel representations in primary motor cortex and premotor neurons during associative learning. *J Neurosci* 28(38):9545–9556.

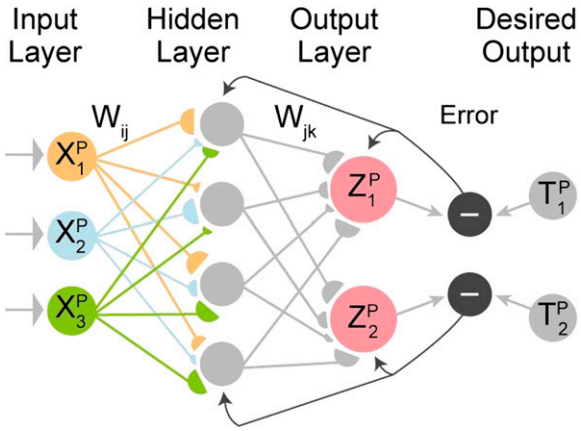


Fig. S1. Schematic feed-forward neural network with supervised learning. Inputs for pattern **P** (denoted as a superscript) are entered at the input layer, and their values are propagated forward through the network via weighted connections to downstream nodes. The activation values, **Z**, at the output layer are compared with target values, **T**, to generate an error signal that is then explicitly fed back through the system to adjust the weights. The network is repetitively fed the input–output associations it is required to learn until the weights reach a point in configuration space where the error is negligible for all of the patterns.

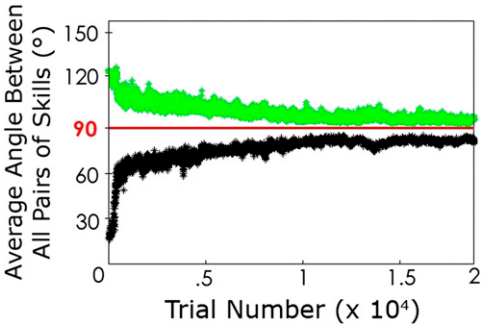


Fig. S2. Orthogonality in the simulations across all possible solution pairs. Fig. 2*F* shows the angle between two specific skills approaching 90° as training progresses. Because there are eight skills learned in the simulation, the number of skill pairs is 28 ($8 \times 7/2$), and the angle of intersection must approach 90° for all skill pairs to satisfy the orthogonality constraint. To generate a population measure, we computed, at each time step and across all skill pairs, the mean of all angles greater than 90° as well as the mean of all angles less than 90° (so that fluctuations above and below 90° do not cancel). The population-level analysis could also be performed by summing the absolute differences from 90° or by computing the variances about 90°. Shown are the results averaged over 1,000 simulations.

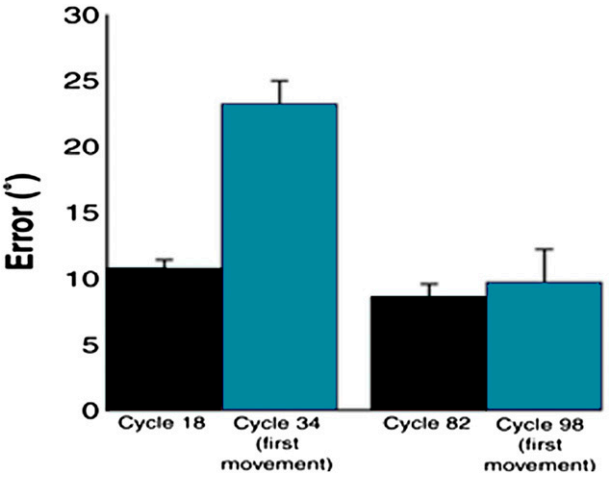


Fig. S3. Comparison of movement error in $\mathbf{D_1}$ before (black) and after (blue) extensive practice of $\mathbf{D_2}$ early in training and late in training. The bars denote SE.

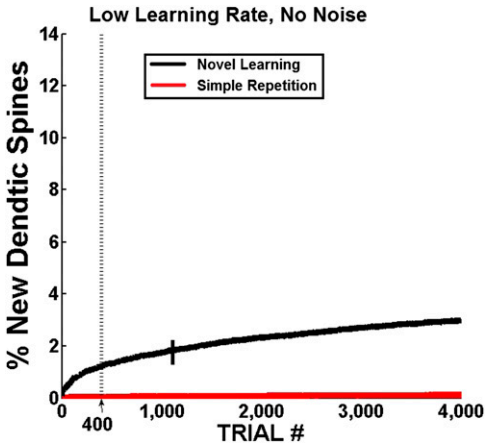


Fig. S4. Dendritic spine fluctuation in a network with a low learning rate and no noise. Fig. 5*B* shows that with a hyperplastic, noisy network, dendritic spine fluctuation is quite prominent. Even in the repetitive practice condition, the percentage of new spines formed after 100 trials rises above 2%. In contrast, the rates of spine fluctuation are severely reduced in the case of the nonhyperplastic, noiseless network, as illustrated. Note that thousands of trials have to be plotted, as opposed to hundreds of trials, to observe appreciable synaptic turnover. The vertical dotted line in notes, for comparison, 400 trials (the extent of the domain in Fig. 5*B*). For the novel learning condition, learning asymptotes after 1,000 trials (the "A" in the plot), at which time the percentage of new spines formed is roughly 2%. In the simple repetition condition, virtually no spines are newly created, as its plot overlaps with the *x* axis. These same basic results held for all parameter settings as long as the network had a low learning rate and no noise.