A Minimum Perturbation Theory of Deep Perceptual Learning

Haozhe Shan[1, 2] and Haim Sompolinsky[1, 3, 4]

[1]Center for Brain Science, Harvard University, Cambridge, MA, United States
[2]Program in Neuroscience, Harvard Medical School, Boston, MA, United States
[3]Edmond and Lily Safra Center for Brain Sciences, Hebrew University of Jerusalem, Jerusalem, Israel
[4]Racah Institute of Physics, Hebrew University of Jerusalem, Jerusalem, Israel

October 5, 2021

Abstract

Perceptual learning (PL) involves long-lasting improvement in perceptual tasks following extensive training. Such improvement has been found to correlate with modifications in neuronal response properties in early as well as late sensory cortical areas. A major challenge is to dissect the causal relation between modification of the neural circuits and the behavioral changes. Previous theoretical and computational studies of PL have largely focused on single-layer model networks, and thus did not address salient characteristics of PL arising from the multiple-staged "deep" structure of the perceptual system. Here we develop a theory of PL in a deep neuronal network architecture, addressing the questions of how changes induced by PL are distributed across the multiple stages of cortex, and how do the respective changes determine the performance in fine discrimination tasks. We prove that in such tasks, modifications of synaptic weights of early sensory areas are both sufficient and necessary for PL. In addition, optimal synaptic weights in the deep network are not unique but span a large space of solutions. We postulate that, in the brain, plasticity throughout the deep network is distributed such that the resultant perturbation on prior circuit structures is minimized. In contrast to most previous models of PL, the minimum perturbation (MP) learning does not change the network readout weights. Our results provide mechanistic and normative explanations for several important physiological features of PL and reconcile apparently contradictory psychophysical findings.

1 Introduction

Perceptual learning (PL) refers to the improvement of performance in perceptual tasks after practice and is accompanied by long-lasting changes to response properties in sensory cortices[1, 2, 3, 4, 5, 6, 7, 8]. In part due to conflicting experimental observations, several important issues of neural mechanisms of PL remain outstanding after decades of research.

First, which cortical areas undergo modifications that drive PL? While behavioral specificity of PL[9, 10] points to an important role for plasticity in early sensory areas, single-unit response properties in visual areas V1 and V2 show only mild changes after PL[1, 2]. In addition, PL induces significant changes to single-unit properties in intermediate to late stages of visual processing, such as V4[3, 4, 8, 7], LIP[11], and IT[12, 13]. Taken together, these findings indicate that even for PL of low-level features, there can be broad and graded changes across the visual hierarchy. Importantly, the vast majority of PL experiments are observational, leaving open the question of whether these neural correlates *cause* PL.

Second, what are the functional consequences of observed changes? While analysis of changes in neuronal responses after PL indicates improved accuracy of the neural coding of the trained stimuli[6, 7, 8], this appears inconsistent with the behavioral finding that PL does not transfer to a different task even when using the

same stimuli[14, 15, 16]. There are also reports that PL is correlated with changes in decision-making areas (e.g., LIP) but not sensory areas (e.g., MT)[17, 11], suggesting that in these scenarios, PL primarily modifies the readout. The Reverse Hierarchy Theory[18] proposes that PL is initially driven by learning in later areas, which results in less specific learning; earlier areas are modified if the task is difficult, leading to more specific learning. Analysis of a reduced model of perceptual learning lent support for this theory[19] . However, recent experimental and computational studies questioned these predictions ([6, 7, 20]).

Perceptual learning has been the subject of several computational studies. Most of them, however, focused on either changing weights of the readout from a fixed sensory array [21, 22, 23] or changing only the input layer to a single cortical circuit [20], and as such do not address the neural correlates of PL in multiple cortical regions. A recent numerical work simulated learning in large, deep convolutional networks[24], the complexity of which hinders a systematic mechanistic analysis in a broad parameter range.

In the present work, we address neuronal mechanisms of PL from a computational perspective. We ask: what changes in the multi-stage neuronal systems are *necessary* for learning the task, and what are the network motifs characterizing the space of all possible solutions? In the face of a multiplicity of solutions, we study the role of plausible constraints on the plasticity in the circuit; such constraints on the learning process can either be imposed explicitly or implicitly through the choice of the synaptic learning dynamics. Finally, we analyze how changes to representations contribute to cross-stimuli specificity and transfer of PL.

More concretely, we studied PL in a simple deep neural network (DNN) model. DNNs imitate hierarchical structures of sensory areas by including multiple layers of feedforward synaptic weights stacked between an input layer and a linear readout[25, 26, 24]. We first developed an analytical theory of learning fine perceptual discrimination in the network model. The theory allows us to characterize the nature of representational changes needed for PL, as well as the space of all possible synaptic modifications that can give rise to such changes. We introduced a minimum-perturbation (MP) principle that targets the specific synaptic changes that solve the task while making the smallest overall changes to the pre-learning synaptic weights, and we then studied the resultant distribution of plasticity across the layers. Next, we simulated the dynamics of learning and found that the slow progression of PL naturally leads to modifications that agree well with those predicted by the MP constraint. Importantly, while modifying weights between the earliest areas is sufficient and necessary for PL, later representational weights are also modified to reduce the overall perturbation to weights. In contrast, the readout weights that connect the sensory representations to the decision-making stage are predicted to be the same as the ones that allow a baseline performance of the task for all stimuli range. The MP solution to PL improves performance by strengthening the signal in cortical representations across layers. Noise in the neuronal responses, caused by sensory noise, is not suppressed and may even increase, although only moderately relative to the signal amplification. Finally, we show that this solution alters cortical representations to give rise to rich cross-stimuli transfer patterns, which are readily testable. Analyzing the MP solution resolves several long-standing conflicts in experimental findings and delineates roles played by early and late areas in PL.

2 Results

We studied perceptual learning (PL) in a feedforward neural network with L layers of sensory neurons receiving input from N channels (Fig.1A). The feedforward weight matrix connecting presynaptic neurons in the $l-1$th layer to postsynaptic neurons at the lth layer is hereafter denoted as $\boldsymbol{W}^l$. A decision neuron produces an output using a linear readout integrating responses of the Lth layer neurons with synaptic weight vector $\boldsymbol{a}$. All other neurons have rectified linear activation functions mimicking rectification of synaptic potential by the neuron's firing threshold. Input channels represent a 1D angular stimulus (such as orientation or direction of motion) with bell-shaped tuning curves with width σ_s (Fig.1B) and independent and identically distributed additive Gaussian noise. We hereafter refer to input channels with large σ_s as having low input selectivity and vice versa (example 2D Gabor stimuli that correspond to different values of σ_s are given in Fig.M1). Importantly, our model neurons respond deterministically to a given activation of the input array. Thus, the noise in their responses is due to "input noise", i.e., the above mentioned noise in the input array. The effect of additional neuronal noise is discussed toward the end the results section.

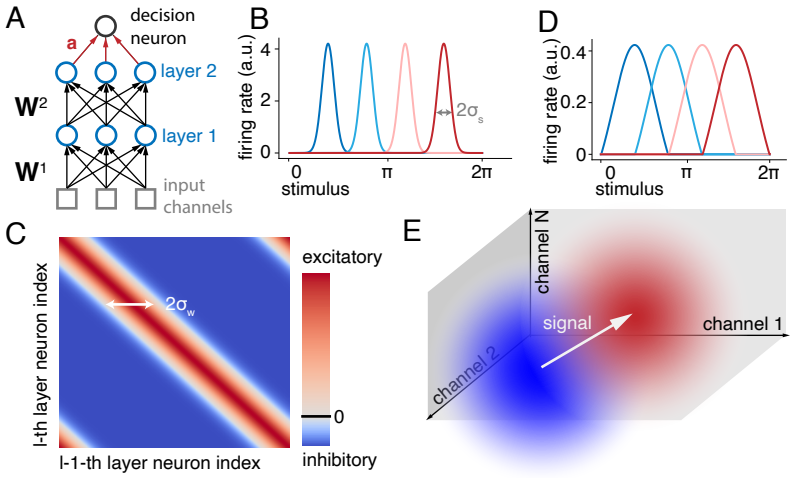

Figure 1: **Model of perceptual learning.**
(A) Architecture of a two-layer version of the model. The activation of input channels (gray squares) is passed through two layers of intermediate layers (blue circles) before getting read out by a linear readout ($\boldsymbol{a}$).
(B) Example tuning curves of input channels. Each curve represents a channel with a different preferred stimulus. The preferred stimuli of input channels uniformly tile $[0, 2\pi]$. Selectivity of them is controlled by σ_s (taken to be 0.2 in these examples). Larger σ_s indicates wider tuning curves and *less* selectivity.
(C) Example weight matrix, $\boldsymbol{W}^l$, before learning. Selectivity of weights is controlled by σ_w (taken to be 0.8 in this example). Larger σ_w indicates that each neuron in layer l receives input from more neurons in layer $l-1$ and has *less* selectivity. Weights connecting neurons with similar preferred stimuli tend to be excitatory and strong while those connecting neurons with dissimilar preferred stimuli tend to be weak and inhibitory.
(D) Example neuronal tuning curves before learning. Each curve is a different intermediate neuron. Due to structures in the pre-learning weights, these neurons have preferred stimuli that uniformly tile $[0, 2\pi]$, as well as bell-shaped tuning curves. ($\sigma_s = 0.2, \sigma_w = 0.8, L = 3$, and the last layer is analyzed)
(E) Schematics of the perceptual task in input space. Noisy representations of θ_+ (red) and θ_- (blue) are shown. The signal is in the direction of the difference between the means of these representations.

We assume pre-PL weight matrices to be circulant matrices with a Gaussian profile (Fig.1C), reflecting the spatial modulation of the synaptic input to a neuron. Specifically, every neuron receives strong excitation from neurons with similar preferred stimuli and weak inhibition from those with dissimilar preferred stimuli. The width of the synaptic spatial modulation is denoted σ_w (hereafter, networks with large σ_w are referred to as having low weight selectivity and vice versa). As a result, prior to PL, neurons have bell-shaped tuning curves (Fig.1D). Note that the circulant weight structure may not persist after PL in our model and weight selectivity only describes the pre-PL weights.

The perceptual task consists of fine discrimination between two similar stimuli, $\theta_\pm = \theta_{\rm tr} \pm \delta\theta$, where the center stimulus $\theta_{\rm tr}$ is called the trained stimulus. In each trial, a stimulus generates a noisy activation of the input array, denoted as an N-dimensional vector $\boldsymbol{x}^0 = \boldsymbol{f}^0(\theta_\pm) + \text{input noise}$, where $\boldsymbol{f}^0(\theta_\pm)$ is the noise-averaged activity for each of the two angles, respectively, and input noise is a vector of i.i.d. Gaussian distributed noise with variance σ^2. Viewed in input space, the task amounts to separating two high-dimensional spherical Gaussian clouds (Fig.1E), centered at $\boldsymbol{f}^0(\theta_\pm)$. The input signal direction, denoted by $\boldsymbol{s}$, is defined as a unit vector in the direction of $\boldsymbol{f}^0(\theta_+) - \boldsymbol{f}^0(\theta_-)$. In each trial, the decision neuron's activity r indicates whether the input comes from the θ_+ stimulus or from θ_- with $r > 0$ or $r < 0$, respectively. Stimuli are presented with equal probability; the optimal performance in the task is thus given by performing maximum likelihood discrimination (MLD,[21]). Importantly, in this task, MLD can be realized by directly summing the input layer activities with weight vector $\boldsymbol{s}$. However, in the cortical deep architecture, the decision neuron has direct access only to the top sensory layer.

The pre-PL value of the readout vector $\boldsymbol{a}$ is chosen to be the optimal linear readout for this task from the pre-PL top layer (Methods M9). This gives the network a well-above-chance but generally suboptimal performance (as shown below), imitating animals that understand the task but have not yet acquired the skills required for near optimal performance. Prior to learning, a similar readout applied for discriminating around other angles will generate the same (sub-optimal) performance, as the pre-learning sensory coding accuracy is the same for all angles.

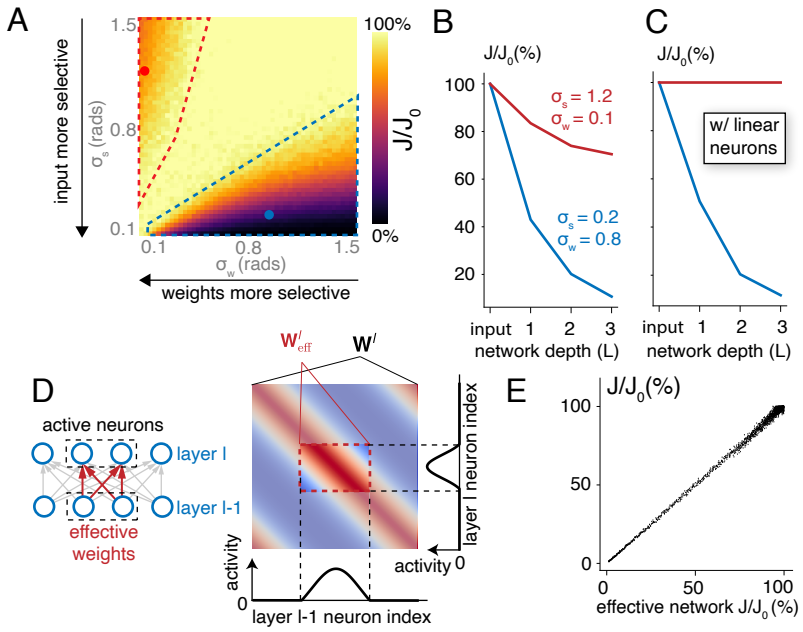

Figure 2: **Suboptimal neural representations before learning.**
(A) Linear Fisher information for the trained stimulus in layer 1 (J_1) divided by that in the input (J_0), for different input and weight selectivity. If $J/J_0 = 100\%$, no information is lost through the network and the pre-PL performance is optimal. The ratio is low for large σ_s, small σ_w (the "unselective-input-selective-weights" regime, red polygon) or small σ_s, large σ_w (the "selective-input-unselective-weights regime, blue polygon). Dots: example parameters used in B. $N = 1000$ in all panels.
(B) Linear Fisher information for the trained stimulus (J) in the last layer of networks of different depths, divided by that in the input layer (J_0). Each curve (red vs. blue) corresponds to a different set of σ_s, σ_w.
(C) Same as B, but assuming that all neurons are active; equivalently, the effective weight matrix $\boldsymbol{W}^l_{\text{eff}}$ is the entire weight matrix $\boldsymbol{W}^l$. Information loss in the unselective-input-selective-weights is prevented by this assumption, while the loss in the selective-input-unselective-weights is not significantly affected.
(D) Schematics showing the relationship between effective weights $\boldsymbol{W}^l_{\text{eff}}$ and all weights $\boldsymbol{W}^l$. On the left: black, dashed boxes mark neurons with above-zero average activation for the task ("active neurons"). Weights connecting active neurons from layer l to those in layer $l+1$ are the effective weights (red arrows). On the right: the red box shows the submatrix within $\boldsymbol{W}^l$ that composes the effective weights; black curves show the average response of neurons in layers l and $l-1$.
(E) Match between J_1/J_0 computed from nonlinear networks and that computed using linear effective networks. Every dot is a three-layer network with a different (σ_s, σ_w).

Suboptimal Neural Representations Before Learning

Since our pre-PL weights and neurons are tuned to the stimulus variable θ, is it possible that the network can perform the perceptual task optimally by an appropriate readout, without any modification of the representation? To address this question we chose Linear Fisher Information [27] as the metric of performance, as it determines the signal-to-noise ratio (SNR) of the best performance of a linear neuron reading out from the top layer (Lth). It is defined as

$$J_L = (d_\theta \boldsymbol{f}^L)^T (\boldsymbol{\Sigma}_L)^{-1} d_\theta \boldsymbol{f}^L, \tag{1}$$

where $d_\theta \boldsymbol{f}^L = (\boldsymbol{f}^L(\theta_+) - \boldsymbol{f}^L(\theta_-))/2\delta\theta$ and $\boldsymbol{f}^L(\theta_\pm)$ are the noise average response vectors of the top layer to the two stimuli. The matrix $\boldsymbol{\Sigma}_L$ is the noise covariance matrix in layer L (if this matrix is low-rank, then ${\boldsymbol{\Sigma}_L}^{-1}$ stands for the pseudoinverse[28]). For brevity, we will refer to J_L as the Fisher Information.

As mentioned above, a linear decoder reading from the input layer can have optimal performance, hence

an upper bound on the network Fisher information is given by the input SNR, which is the input Fisher information, $J_0 = \|\boldsymbol{f}^0(\theta_+) - \boldsymbol{f}^0(\theta_-)\|^2/\sigma^2$. Note that since the input noise statistics does not depend on θ, J_0 is the upper bound on the SNR of any decoder and is achieved by MLD for large input width[21]. In order for the network to have optimal performance, J_L must match J_0. While in our model the mapping from input to layer L is deterministic and does not inject additional noise, filtering the input through the pre-PL intermediate synaptic weights may lead to loss of J_L and degraded performance. Indeed, we found two distinct scenarios where such filtering significantly reduces J_L(Fig.2A for $L = 1$ and Fig.S1 for $L = 2, 3$): when input channels are unselective and pre-PL feedforward weights are selective (large σ_s, small σ_w, red polygon), and when input channels are selective but feedforward weights are unselective (small σ_s, large σ_w, blue polygon). Furthermore, under these conditions, the loss of information through the network filtering is bigger in deeper networks (Fig.2B). These findings counter the intuition that sharper tuning curves (coming from more selective input channels or feedforward weights) always produce better performance.

To explain these results, we identify two sources of information loss by the network. In the regime of selective-input-unselective-weights (small σ_s, large σ_w), information is lost because the rank of the network pre-PL weights $\boldsymbol{W}^l_{\text{pre}}$ is low, implying that they project to subsequent layers only part of the signal in the input. In this regime, information loss occurs regardless of the nonlinearity of representation neurons (Fig.2C). Importantly, the low-rankness arises from smoothness of the pre-PL weights and does not vary with width of the network. On the other hand, in the unselective-input-selective-weights (large σ_s, small σ_w) regime, the weight matrices project the full signal. However, due to firing rate rectification, a substantial fraction of the representation neurons are inactive for essentially all training stimuli. Thus, the weights connecting only active neurons are low-rank and are incapable of transmitting the full signal. Thus, in this regime, information loss is entirely due to the neuronal nonlinearity and disappears if we remove this nonlinearity (Fig.2C). Importantly, because in our paradigm $\theta_\pm$ stimuli generate highly overlapping input patterns and the input noise is substantially suppressed by the averaging performed by the weights, only neurons with preferred stimuli near the trained stimulus are activated by the set of stimuli involved in the task. Since the identity of active neurons is largely constant for most of the stimuli (full derivation in Methods M2), we can replace our nonlinear network with weights $\boldsymbol{W}^l$ by a linear network with effective weights $\boldsymbol{W}^l_{\text{eff}}$ (Fig.2D), similar to the approximation done in [29] for recurrent networks. Consistent with this observation, Fig.2E shows that the effective linear networks exhibit the same loss of information as their corresponding nonlinear networks. The theory also provides a unified perspective on why information is lost. In an effective linear network, the severity of information loss is related to the span of the product of all effective weight matrices (Sec.M3.1). Information loss is severe if the signal, $\boldsymbol{s}$, projects substantially outside of its span. Indeed, this is the case in both regimes of information loss(see Fig.S2). For simplicity, we will hereafter use $\boldsymbol{W}$ and $\boldsymbol{a}$ to denote the effective weights.

Weight Structures After Learning

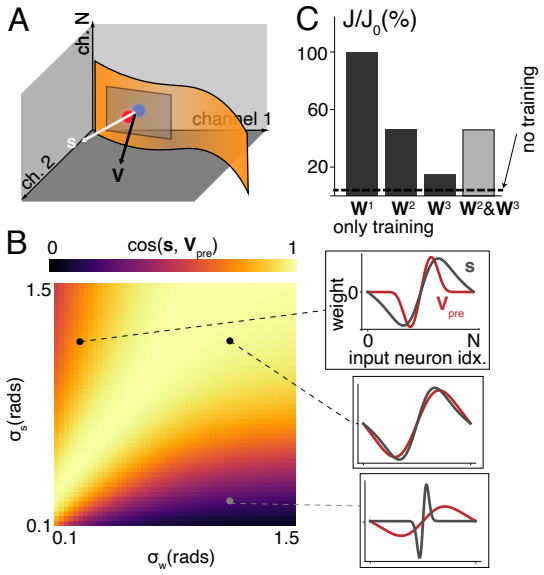

Figure 3: **Space of weights that give optimal performance.**
(A) Schematics showing the relationship between the network decision surface and the signal direction. Decision surface of the network (orange) in input space is globally nonlinear but locally linear (gray parallelogram). The local hyperplane is defined by the N-dimensional $\boldsymbol{V}$ (black arrow), which in turn depends on the effective weights. Discrimination is optimal if and only if $\boldsymbol{V}$ is parallel to the signal $\boldsymbol{s}$ (gray arrow), making the decision surface perpendicular to $\boldsymbol{s}$.
(B) Cosine angle between $\boldsymbol{s}$ and the pre-PL $\boldsymbol{V}$, $\boldsymbol{V}_{\text{pre}}$. Higher values indicate better alignment between these two vectors and a network performance that is closer to being optimal. Insets: some examples of $\boldsymbol{s}$ and $\boldsymbol{V}_{\text{pre}}$. All networks have and $L = 3, N = 1000$ (same for (C)).
(C) Best last-layer information (J_3) achievable if plasticity is restricted to some weight matrices in a three-layer network. Dashed line: performance if no weight matrix is modified. Modifying any weight matrix improves the performance, but only modifying $\boldsymbol{W}^1$ is sufficient and necessary for optimizing it. Note that these results are valid even if the all weights (not just the effective ones) in each matrix are trained. ($\sigma_s = 0.2, \sigma_w = 0.8$)

We first characterize the space of all possible solutions to the PL task, i.e., all possible weights that render optimal performance. This characterization is vastly simplified by our observation above, namely that we can replace our network by a linear network with effective weights. The network input-output relation is approximately linear and is given by $r(\theta) = \boldsymbol{V}^T \boldsymbol{x}^0(\theta)$ where $\boldsymbol{x}^0(\theta)$ is the single-trial noisy activity vector of input channels and the *input-output mapping* is given by (see Fig.3A)

$$\boldsymbol{V} = \boldsymbol{W}^{1^T} \boldsymbol{W}^{2^T} ... \boldsymbol{W}^{L^T} \boldsymbol{a}. \tag{2}$$

Conceptually, for the performance to be optimal, two conditions must be satisfied. First, the last-layer neural representation must contain the full Fisher information, as we have argued in the previous section. Second, the network readout must be accessing all the information. These two conditions can be combined into the requirement that $\boldsymbol{V} \propto \boldsymbol{s}$. This condition is not satisfied in pre-learning networks (even with optimized readouts), leading to suboptimal performance (Fig.3B).

To characterize solutions, we note that a *necessary and sufficient condition* is that the first layer effective weights are modified such that after learning,

$$\boldsymbol{W}^1 = \boldsymbol{u}\boldsymbol{s}^T + \boldsymbol{W}_\perp. \tag{3}$$

where $\boldsymbol{W}_\perp \boldsymbol{s} = 0$ (Methods M8). In order for the signal term to be read out by the network, the vector $\boldsymbol{u}$ must obey $\boldsymbol{u}^T \boldsymbol{W}^{2^T} ... \boldsymbol{W}^{L^T} \boldsymbol{a} \neq 0$. In addition, since the remainder weight matrix is perpendicular to the signal, it should not contribute to the network output. Hence, it must obey $\boldsymbol{W}_\perp^T \boldsymbol{W}^{2^T} ... \boldsymbol{W}^{L^T} \boldsymbol{a} = 0$. This result implies that, to obtain an optimal performance, higher layer weights including the readout weights can be essentially arbitrary and in particular can retain their pre-learning values, provided the first layer weights are appropriately modified. Conversely, restricting the plasticity to higher layer weights while freezing the first layer weights to their pre-PL values is insufficient for optimal performance, because the input to the plastic weights is already filtered suboptimally by the bottom frozen weights (Fig.3C).

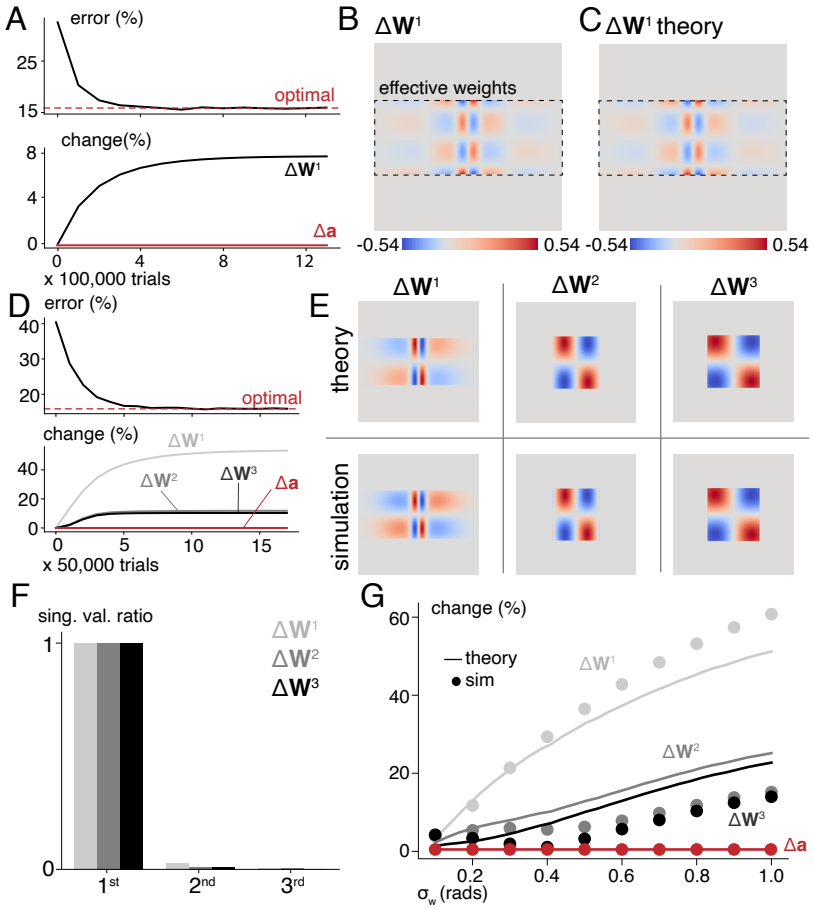

Figure 4: **Minimum-perturbation learning in networks**.
(A) Top: error over time converges to the optimum during the simulated PL of an $L = 1$ network. Bottom: Norms of changes to $\boldsymbol{W}^1$ and $\boldsymbol{a}$, divided by norms of $\boldsymbol{W}^1_{\text{pre}}$ and $\boldsymbol{a}_{\text{pre}}$. This figure describes learning in the selective-input-unselective-weights regime ($\sigma_s = 0.2$, $\sigma_w = 0.8$, learning rate $= 10^{-3}$, and $N = 1000$, unless otherwise noted). Results from the unselective-input-selective-weights regime are shown in Fig.S3.
(B) A visualization of changes to $\boldsymbol{W}^1$ after simulated learning. The x-axis is the index of input channels; the y-axis is the index of neurons in layer 1. Dashed box: effective weights.
(C) Same as (B), but for MP changes to $\boldsymbol{W}^1$.
(D) Same as (A), but for an $L = 3$ network.
(E) Structures of MP modifications (top row) and simulated changes (bottom row) in an $L = 3$ network. Colorbars are not shown. See (G) for the magnitude of changes. The axes are analogous to those in (B) and (C).
(F) Leading singular values (normalized by the top one) of simulated $\Delta \boldsymbol{W}^{1,2,3}$. That the first singular value dominates suggests that these matrices are approximately rank-1.
(G) Magnitude of simulation changes ("sim") and MP plasticity ("theory") for different values of σ_w.

Learning While Minimizing Network Perturbation

The analysis above shows that the fine discrimination task can be solved by networks with a large variety of weight patterns. Which one does the brain adopt after PL? To answer this question, we studied gradient descent supervised learning as the learning algorithm (Methods M1.4) because it has been shown to reproduce experimental findings in PL[25, 24, 30]. In this learning rule, at each plasticity step, the synaptic weights change in proportion to the negative gradient of the error signal with respect to a small change in the weights[31]. The actual magnitude of weight changes at each step is controlled by a learning rate constant. Here, we used small learning rates to imitate the slow progression of PL (Methods M11).

Shallow networks: First, we analyzed learning dynamics in a shallow ($L = 1$) network (Fig.4A) (for the selective-input-unselective-weights regime; similar results are shown in the unselective-input-selective-weights regime (Fig.S3)). As learning progresses, the synaptic weights converge to a solution, and the error rate converges to its minimal value (Fig.4A). Learning dynamics lead to three interesting features of weight modifications. First, PL-induced modifications are restricted to only the subset of weights connecting only neurons that were active initially (Fig.4B, dashed box). This is explained (Methods M7) by the fact that during gradient-based learning, the set of active neurons in every layer remains the same as that before learning. Consequently, the same subset of weights (the effective weights) contributes to the network output (and therefore the error) throughout learning, thus only those are modified. Second, although learning is enabled for both $\boldsymbol{W}^1$ and $\boldsymbol{a}$, only the former undergoes significant changes (Fig.4A, bottom). Finally, *changes* to weights appear to be rank-1.

To explain these salient features of the observed plasticity, we propose that they are the outcome of an implicit tendency of the slow gradient-based learning dynamics to minimize the overall changes in the weights relative to their pre-PL values. Since only the effective weights are modified, we focus our analysis on changes to the effective weights, denoted as $\Delta\boldsymbol{W}^1 \equiv \boldsymbol{W}^1_{\text{post}} - \boldsymbol{W}^1_{\text{pre}}$ ("post" denotes post-PL values). To test this hypothesis, we studied the solution to the PL task under the principle minimum-perturbation (MP) plasticity. According to this principle, out of all viable post-PL weights that solve the perceptual task, learning dynamics finds the one that minimizes network perturbation, as measured by $\sum_{l=1}^{L} \|\Delta\boldsymbol{W}^l\|^2 + \|\Delta\boldsymbol{a}\|^2$.

MP weight modifications are fully determined by pre-PL weights and task parameters (Methods M10). Analytical evaluation of the MP modifications for $L = 1$ exactly matched simulations, in both structure and magnitude (Fig.4C). Importantly, under MP modifications, the readout weights are essentially unchanged, so that $\boldsymbol{a} = \boldsymbol{a}_{\text{pre}}$. In addition, The change to $\boldsymbol{W}^1$ has a rank-1 structure, $\Delta\boldsymbol{W}^1 = \boldsymbol{a}(\boldsymbol{s}^T - \boldsymbol{a}^T\boldsymbol{W}^1_{\text{pre}})/\|\boldsymbol{a}\|^2$ so that $\boldsymbol{W}^1_{\text{post}}$ contains a rank-1 component $\boldsymbol{a}\boldsymbol{s}^T$ and a term that cancels out the projection of the pre-PL weight matrix on the readout $\boldsymbol{a}$, thus satisfying Eq.3. This structure is in contrast to a grandmother-cell strategy where a small number of individual neurons adopt optimal filters; instead, MP learning opts for a population-coding scheme where the linear filters of all active neurons are altered slightly.

MP learning in deep networks: Simulations of PL in deeper networks (Fig.4D) confirm that, similar to the case of single-layer networks, changes to synapses are concentrated in those between neurons that responded to the stimulus prior to learning (i.e., the effective weights, Fig.4E bottom row). Also, the readout weights remain essentially unchanged. We solved analytically the plasticity under MP constraint for deep networks(Methods M10). The theory predicts that changes in all effective weight matrices are confined to a rank-1 structure,

$$\Delta\boldsymbol{W}^l = \boldsymbol{V}_l\boldsymbol{U}_l^T \tag{4}$$

where $\boldsymbol{V}_l$ is the readout vector from the l-th layer and $\boldsymbol{U}_l$ is the effective signal vector propagating into this layer. They are given by

$$\boldsymbol{V}_l = {\boldsymbol{W}^{l+1}_{\text{post}}}^T \cdots {\boldsymbol{W}^{L}_{\text{post}}}^T \boldsymbol{a} \tag{5}$$

$$\boldsymbol{U}_l = \boldsymbol{W}^{l-1}_{\text{post}} \cdots \boldsymbol{W}^{1}_{\text{post}} \boldsymbol{\lambda}, \tag{6}$$

where the vector $\boldsymbol{\lambda}$ is determined by the requirement that after learning the full input-output mapping $\boldsymbol{V}$ (Eq.2) equals the signal $\boldsymbol{s}$. Note that the above equations need to be solved self-consistently as they involve the post-learning weight matrices.

Numerical solutions of the self consistent-equations of the MP theory for the case of L=3 (Fig.4E, top row) show that $\boldsymbol{W}^1$ undergoes the biggest change, followed by $\boldsymbol{W}^2$ and $\boldsymbol{W}^3$, with $\|\Delta \boldsymbol{W}^2\| \approx \|\Delta \boldsymbol{W}^3\|$. These results are supported by the gradient-based learning, indicating that also in deep networks, this learning dynamics expresses implicit biases qualitatively similar to MP learning (the same result for networks in the unselective-input-selective-weights regime is shown in Fig.S3).

We have shown that the largest changes occur at layer 1 weights but significant changes also occur at the higher layers. According to our analysis above (see Eq. 3), modifying $\boldsymbol{W}^1$ alone is sufficient for learning; so why do higher layers change? We hypothesized that higher layers undergo small perturbations so that the *total* network perturbation is minimized. To verify this interpretation, we computed MP plasticity in a three-layer network assuming that only $\boldsymbol{W}^1$ is modified and found that this consistently leads to more total perturbation than learning involving all matrices (Fig.5A).

While there is overall excellent agreement between gradient descent learning and MP theory predictions regarding the patterns of weight changes, there is a quantitative discrepancy about the magnitude of changes (Fig.4G), suggesting that in deep networks ($L > 1$) the gradient learning dynamics converges to a solution with a slightly larger overall change than the minimal value. Another important factor in the comparison between the two is the value of the learning rate. As shown in Fig. 5B, a substantial increase in the learning rate significantly increases the amount of induced synaptic changes, (Fig.5B) widening the discrepancy with the MP changes. This suggests that the slow progression of PL is essential for achieving MP plasticity.

We hypothesize that adding an explicit bias favoring small weight perturbation to the cost function of the learning dynamics will alleviate the quantitative discrepancy with the MP theory. Indeed, we repeated the learning simulations with an explicit penalty on the size of weight changes in the loss function. The strength of this penalty balances a trade-off between optimizing performance on the perceptual task and minimizing network perturbation. We adjusted the strength of this penalty to be the largest that allows convergence to optimal performance (Fig.5C). Simulations with this penalty term converged to MP plasticity (Fig.5D; changes to the readout are negligible for both simulation and theory).

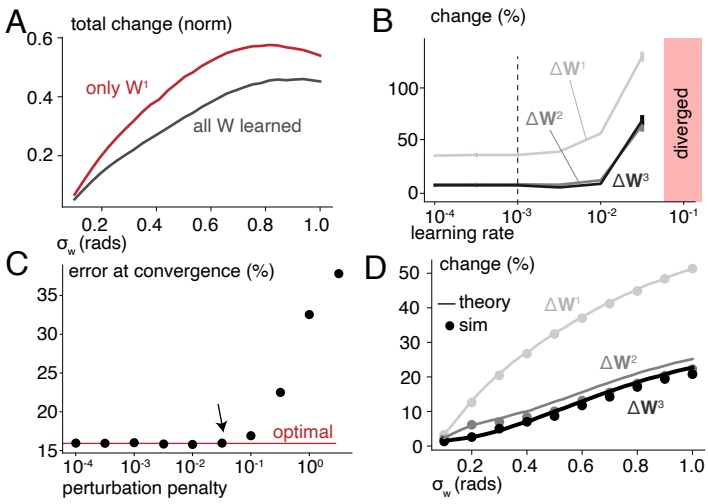

Figure 5: **Factors affecting network perturbation.**
(A) Restricting learning to $\boldsymbol{W}^1$ leads to more network-wide perturbation (measured by the sum of matrix norms of $\Delta\boldsymbol{W}^1, \Delta\boldsymbol{W}^2, \Delta\boldsymbol{W}^3$) than unrestricted learning. In either case, the readout $\boldsymbol{a}$ is also allowed to learn but does not change significantly following PL. In all panels , $\sigma_s = 0.2$, $\sigma_w = 0.8$, $L = 3$, Learning rate $= 10^{-3}$ and $N = 1000$, unless otherwise noted.
(B) Effects on perturbation when a higher learning rate is used. Average of 10 simulations. Error bars show standard error and are too small to be seen in most places. The red box indicates that when learning rates are greater than $\approx 10^{-1.5}$, gradient descent diverges. Dashed line: learning rate used by default.
(C) Effect of adding an additional perturbation penalty to the loss function on discrimination error after PL has converged. If the penalty is too strong (e.g. above 10^{-1} in this case), error at convergence is suboptimal because learning prioritizes reducing the magnitude of changes. Arrow: penalty strength used in (D), which is the maximal value without making error suboptimal.
(D) Match between the magnitude of MP changes to weights ("theory") and changes from simulations with a perturbation penalty ("sim").

Modifications of Neural Representations

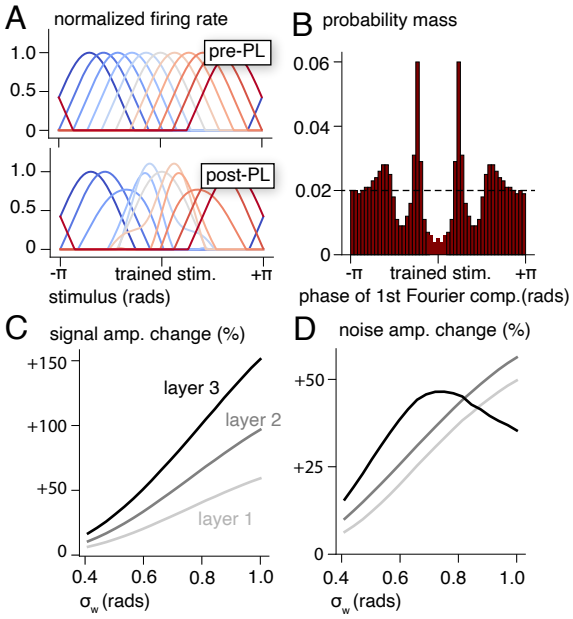

Figure 6: **MP Learning-induced changes to response properties.**
(A) Example pre-PL and post-PL tuning curves of neurons (selective-input-unselective-weights regime). Each color denotes a different neuron. All tuning curves are normalized by the max firing rate before learning. Before learning, tuning curves only differ by their preferred stimuli. In all panels, unless otherwise noted, $\sigma_s = 0.4, \sigma_w = 1.0, N = 1000, L = 3$, and the last layer is analyzed.
(B) Probability distribution of phase of the 1st Fourier component of tuning curves. Before PL, the distribution is uniform (dashed line). Location of the phase indicates preferred stimuli of the neuron. The depression near the trained stimulus indicates that fewer neurons prefer the trained stimulus following PL.
(C, D) PL-induced changes to signal(C) or noise(D) amplitude across layers for different weight selectivity. Changes are generally greater in higher layers and in networks with initial weights that are less selective (larger σ_w).

How does MP plasticity modify neural representations? Example pre-PL and post-PL tuning curves in the last layer of an $L = 3$ network are visualized in Fig.6A. After learning, tuning curve centers shift and their shapes are modified so that the population is sensitized towards changes in θ near the trained stimulus. As shown in Fig.6B, the new distribution of preferred angles exhibits a sharp peak at the side bands of the trained stimulus and is decreased on both sides, reflecting the concentration of tuning curve maximal slopes near the trained stimulus. Additional contribution to the enhanced selectivity comes from the tuning curve shape modification that leads to increased slopes of individual tuning curves near the trained stimulus.

How do observed changes to tuning curves contribute quantitatively to improvement of perceptual performance? We analyzed the two factors determining discrimination performance around the trained stimulus: signal and noise, defined such that $J_L = (\text{signal/noise})^2$. The signal amplitude at θ in layer l is defined as $\|d_\theta \boldsymbol{f}^l(\theta)\|$. Noise at θ in layer l is defined via $\text{noise}^{-2} = d_\theta \boldsymbol{f}^l(\theta)^T {\boldsymbol{\Sigma}^l}^{-1} d_\theta \boldsymbol{f}^l(\theta) / \|d_\theta \boldsymbol{f}^l(\theta)\|^2$. Note that this noise projects the inverse covariance of the neural fluctuations onto the signal direction, analogous to the information-limiting correlations[32, 33]. While early studies of PL highlighted the importance of increased tuning curve slopes (i.e., amplified signal), some recent work suggested that PL is achieved primarily by noise suppression [20]. Our model exhibits a pronounced amplification of signal (Fig.6C), with the effect being stronger in higher layers. Surprisingly, we found that PL also *amplifies* noise across all layers, although to a weaker extent than signal amplification (Fig.6D). Thus, MP learning improves perceptual performance by

strengthening the signal rather than weakening the noise.

Effects of Learning on Untrained Stimuli

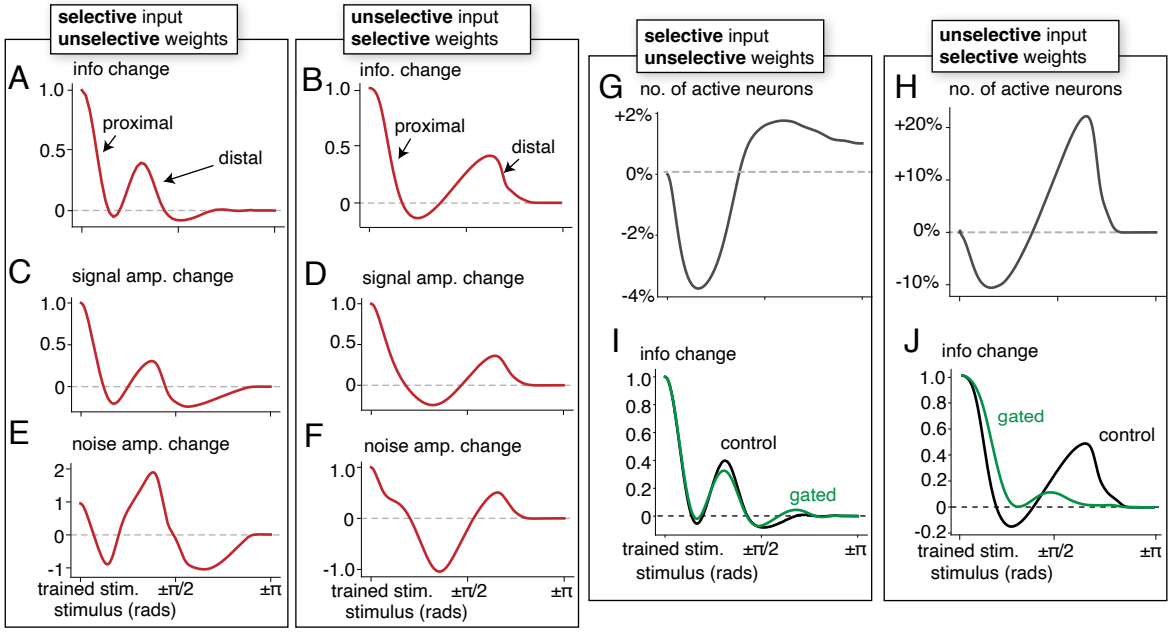

Figure 7: **Transfer of PL to Untrained Stimuli.**
(A, B) FI change in the last layer for different stimuli after PL, normalized by change for the trained stimulus. The change for the trained stimulus is 1 by definition. FI gain is prominent for stimuli close to the trained one ("proximal"), and those somewhat different from the trained one ("distal"). In all panels, $N = 1000, L = 3$, and the last layer is analyzed. For the selective-input-unselective-weights regime, $\sigma_s = 0.4, \sigma_w = 1.0$. For the unselective-input-selective-weights regime, $\sigma_s = 1.2, \sigma_w = 0.1$. The x-axis is shared among panels in each column.
(C, D) Change to signal amplitude for different stimuli, normalized by change for the trained stimulus.
(E, F) Change to noise amplitude for different stimuli, normalized by change for the trained stimulus.
(G, H) Changes to the number of active neurons for each stimulus (third layer, $L = 3$). There is no change for the trained stimulus. Before learning, the number of active neurons is the same for all θ.
(I, J) Effects of fixing active neurons ("gating") after PL on information change for different stimuli. The "control" curves (black) are the same as those shown in (A,B). For the "gated" curves (green), neurons in the post-PL network are gated such that for every stimulus, active neurons in each layer of the post-PL network are the same ones that have been active before PL.
All variables shown in this figure are symmetric around the trained stimulus.

PL-induced weight modification in our model leads to a nonuniform representation of angles post learning, raising the question of how the quality of coding of untrained stimuli is affected. To analyze the pattern of "transfer of learning" to untrained stimuli, we computed normalized information gain, defined as the information change at the last layer for each untrained stimulus divided by information change for the trained stimulus. Our analysis reveals a rich, non-monotonic pattern of transfer arising from MP plasticity. Consistent with experimental findings, PL transfers to stimuli similar to the trained stimulus across parameter regimes (see Fig.7A, B for $L = 3$; "proximal transfer"). Surprisingly, PL also transfers to distal stimuli, where the distance between trained and test stimuli is intermediate ("distal transfer"). Finally, as expected, representations for stimuli far away from the trained one are unaffected by learning.

How are such patterns of transfer connected to PL-induced changes to neuronal representations? We

first analyzed how the *signal* magnitude for untrained stimuli is affected by PL (Fig.7C, D). We found that stimuli that are sufficiently similar to the trained one also exhibit signal amplification, as expected from the smoothness of tuning curves. Interestingly, the signal for distal stimuli is also amplified (Figs.7C and D) . However, stronger signals may not necessarily lead to more information, if they are countered by a concurrent increase of noise. We found that noise *increases* for both proximal and distal stimuli (Fig.7E,F). Similar to the trained stimulus, information for these stimuli nevertheless increases because signal amplification is stronger than noise amplification.

How does PL amplify the signal? In general, signal amplification can be accomplished by two mechanisms: either by modifying the tuning curves of neurons which are active even before learning ("sharpening"), and/or activating neurons that were previously silent to recruit them to represent this stimulus ("recruiting"). For the trained stimulus, since no additional neurons are activated, signal amplification occurs strictly because of sharpening. However, following changes in connectivity by PL, the number of neurons (compared to the untrained network) responding to untrained stimuli may change, hence both mechanisms may contribute to transferred learning. To investigate this question, we first computed how the number of active neurons for each stimulus changes following PL (Fig.7G,H; third layer, $L = 3$). In the selective-input-unselective-weights regime (Fig.7G), the number of active neurons *decreases* for distal stimuli, suggesting that sharpening of population responses is responsible for transfer here. On the other hand, in the unselective-input-selective-weights scenario (Fig.7H), the number *increases* for distal stimuli, suggesting that recruiting could be driving signal amplification here. To test this hypothesis, we performed gating experiments where the identity of active neurons in the presence of distal stimulus is artificially kept the same as prior to learning in all layers. We found that in the selective-input-unselective-weights regime, this manipulation barely affected transfer (Fig.7I), confirming that sharpening drives transfer in this regime. This is consistent with our finding that gating does not significantly reduce signal amplitude in this regime (data not shown). On the other hand, gating significantly reduced distal transfer (Fig.7J) in the unselective-input-selective-weights regime, suggesting that recruiting is driving transfer here. Indeed, gating removed PL-induced effects on signal and noise for distal stimuli (data not shown).

Noise Correlations and Size Dependence

Our discussion of PL so far has not touched on how performance of the post-PL network depends on the numbers of input channels (N_{input}) and sensory representation neurons (N_{hidden}). For input channels, the signal amplitude increases linearly with the number of them. On the other hand, since noise is independent between individual input channels, noise amplitude (σ^2) does not depend on the number of them. As a result, J_0 increases linearly with the number of input channels as expected. Since J_L in a post-PL network matches J_0, J_L also increases linearly with the number of input channels(Fig.8A). The same holds for Fisher information before learning (Fig.S8).

We next consider the effects of changing the number of hidden neurons in the network. Our expression of the space of solutions (Eq.3) can be satisfied with only a few hidden neurons. Thus, learning can achieve optimal performance with small N_{hidden}. Viewed from a signal and noise perspective, both signal amplitude and noise increase linearly with the number of hidden neurons in trained networks, making the overall signal-to-noise ratio independent of N_{hidden}. The situation is different if intrinsic neuronal noise is considered. As a simple example, we consider the effects of injecting i.i.d. Gaussian noise ("output noise") to the activity of neurons in the top layer of the post-PL network (Fig.8B). In this case, noise in the last layer has two components: one is the input noise, filtered through upstream layers; the other is the output noise. We found that when N_{hidden} is small, information is primarily limited by the output noise component and increases linearly with N_{hidden}. When N_{hidden} exceeds approximately $N_{\text{input}}\sigma^2_{\text{output}}/\sigma^2$, the effect of output noise is drastically suppressed, and information saturates with large N_{hidden} to the value determined by the input noise (see detailed analysis in Sec.M3.2). This behavior of the input noise in our model is similar to the information-limiting correlations discussed by [32, 33] which limit information even at large N_{hidden}. Consideration of the more complex problem of having both input noise and intrinsic neuronal noise in all layers both during learning and afterwards is beyond the scope of this work.

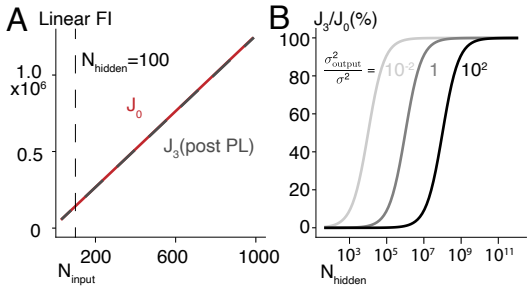

Figure 8: **Dependence of Information on Network Size.**
(A) Linear Fisher information in the input channels and the last layer (after PL) increases linearly with the number of input channels. After learning, the information in the last layer (gray dashed line) is always the same as that in the input layer (red line). In both panels, $\sigma_s = 0.2, \sigma_w = 0.8$, $L = 3$.
(B): Linear Fisher information in the last layer as a function of N_{hidden} in a post-PL network when i.i.d. output noise of different intensities is injected to the last layer. Each curve corresponds to a different magnitude of output noise. J_0 does not depend on N_{hidden}; thus these curves reflect dependency of J_3 on N_{hidden}. The contribution from this output noise to information degradation decreases with increasing N_{hidden}; at large N_{hidden}, the output noise contribution vanishes and the information approaches J_0. For this panel, $N_{\text{input}} = 1000$.

Discussion

We have presented the first theory of perceptual learning in a deep sensory network. We have shown that during a fine discrimination task, a deep sensory network is equivalent to a linear input-output mapping with readout vector $\boldsymbol{V}$ (Eq.2), determined by the subset of weights ("effective weights") connecting active neurons between layers. In the pre-learning network, optimal performance cannot be achieved despite an optimized readout weights $\boldsymbol{a}$, due to the low-rank nature of the effective synaptic matrices in upstream sensory layers. Thus, PL adjusts the effective weights such that $\boldsymbol{V}$ is aligned with the optimal input-output mapping for the task. While there is a large space of weight modification satisfying this objective, we propose that PL operates under the constraint that perturbation to prior synaptic weights be minimal ("MP learning"). We show that a gradient-based learning rule with small learning rates converges to a solution qualitatively similar to that predicted by MP learning. Furthermore, changes to neural representations and behavioral performance induced by MP learning are consistent with experimental observations and lead to new testable predictions.

Loci of Learning

An important counterintuitive prediction of the MP theory is that synaptic weights of the readout unit do not change during PL, a result replicated by a gradient-based learning rule with small learning rates. Crucial for this result is our assumption that readout weights are initialized as optimal, relative to the pre-learning sensory representations. This is contrary to most models of learning in neural networks, where task-specific readout synapses are assumed to be random before learning and undergo significant changes during learning. We argue that random initialization is not biologically plausible when considering naturalistic tasks which should yield above-chance performance prior to PL.

Since the readout does not change, the success of PL is entirely due to improvement of representations in upstream sensory layers. This requires increasing the SNR about the task-relevant stimuli in the sensory layers and ensuring that the existing readout can access all the information. This is similar to recent findings that attention improves performance by shaping cortical representations to fit a fixed readout[34]. Our result is consistent with [20] who found that PL is successful even when only the feedforward weights to a single sensory layer are modified.

Importantly, our results reconcile improved representations with psychophysical findings that were taken as evidence for stable representations during PL, namely observations that PL for one task did not transfer to another task using the same stimuli[14, 15, 16]. This result was interpreted as evidence that population codes for these stimuli did not improve[35]. However, our theory shows that the representation improvement during PL is itself task-specific. Therefore, PL will not transfer to an untrained discrimination task if it has a signal direction perpendicular to that of the trained task, even if the two tasks have highly overlapping mean stimuli. To illustrate this point, we considered a network trained on the θ discrimination task and tested its performance on a input-width discrimination task (telling apart stimuli with different σ_s) using the same stimuli and found no transfer (Fig.S7; details in Sec.M6). Thus, cross-task transfer may not occur despite extensive changes to cortical representations at every layer.

Another question of interest is which stage in the sensory system contributes the most to PL. We have shown analytically that with fine discrimination tasks, plasticity in the first sensory hidden layer is necessary for the success of PL. While this is difficult to test experimentally, it is in agreement with conclusions drawn from PL in Deep Convolutional Neural Networks(DCNNs) previously trained for an object recognition task [24]. Consistent with our results, their numerical simulations indicated that the weight matrix following the first intermediate layer undergoes the biggest modification and that preventing learning in this matrix significantly impacted performance after PL (in their model, the first intermediate layer contains the full information and is thus analogous to our input layer). However, the complexity of DCNNs obscures their interpretability and precludes elucidating the underlying mechanism. Our simplified model with stimuli and connectivity tuned only to a 1D angle allows the development of powerful analytical tools to study mechanisms underlying PL in deep neuronal sensory architectures. In particular, in contrast to DCNNs where higher layers are selective to higher-level complex features, in our model all stages are selective to the same angular feature. Thus, the fact that the orientation discrimination task depends on changes in lower layers rather than higher ones cannot be attributed to the difference in the nature of selectivity of the different stages (see [24, 36]). Rather, modifying early layers plays a critical role in PL due to the loss of information by propagation through the existing low-rank effective weights. Furthermore, even though in MP learning changes are distributed across all layers to minimize the overall perturbation, earlier matrices still consistently undergo bigger changes.

A confusing aspect of current discussions about the loci of PL is the confounding of changes to the synaptic connections (i.e., the magnitude of synaptic plasticity) and the changes in the neuronal response properties. Our theory shows that although changes to weight matrices are larger in lower layers, the increase in SNR and signal amplification are stronger in higher layers (Fig.S5) as they are the outcome of the accumulated changes in upstream layers. The Reverse Hierarchy Theory[18] claims that in a difficult PL task, higher-level layers that are closer to the readout change before lower-level ones. This is supported by a recent one-dimensional model of PL in deep linear networks[19]. Although we have not focused on the dynamics of PL, our numerical results do not substantiate these claims (see e.g., Fig.4). Note that in contrast to [19], the readout layer does not change during PL in our model. Both the MP theory and the gradient-based learning simulation show that the changes in the weights are confined largely to adding rank-1 terms to the existing weight matrices (see Fig.4). Indeed, recent work on supervised learning in deep networks highlighted the compressed dimensionality of gradient-based learning in deep networks[37] .

Nature of Changes to Representations

Our analysis shows that both signal and noise are increased by PL (Fig.6). Performance is improved because signal amplification is stronger. While we found changes to the structure of noise correlation(Fig.S9), such changes do not contribute causally to PL because the overall noise *increases*. The magnitude of predicted changes to tuning curves and SNR depends on input and weight selectivity (Fig.S5). In particular, changes are smaller when selectivity of input and weights is more similar, concomitantly with a smaller performance gap between the network before learning and the optimal level.

Given the large space of solutions, an important question is whether any of the predicted changes to tuning curves and noise are necessary for PL to be successful[20]. Our expression of the space of solutions allows us to compute the minimal amount of signal amplification needed for PL. We found that for deep networks,

signal amplification is indeed necessary for PL (Fig.S6A,B) and is of the same magnitude as the amplification caused by MP. Furthermore, MP changes amplify the signal to the minimal extent necessary for learning. This result suggests that in our model, changes to noise alone is not sufficient for PL; signal amplification, which arises from sharpened tuning curves, is required for PL. Indeed, we found that solving PL under a "soft" MP constraint, where post-PL weights are allowed to move away from the MP weights, leads to even greater signal amplification(Fig.S6C). This result is inconsistent with [20] who found that amplification is not necessary for PL. Their conclusion may be confined to the regime where performance is dominated by neural noise, not input noise as in ours. Additionally, their plasticity model differs from ours in that it assumes circularly invariant weights both before and after learning, which forces a global change of synaptic weights. In contrast, in our model, PL plasticity is localized to the neurons responding to the stimulus (if we require post-PL weights to be circularly invariant in our model, post-PL tuning curves have very unnatural multi-modal shapes. See Sec.M13). Finally, we note that our prediction of signal amplification stems from our observation of a fixed readout under MP learning. If the readout can be adapted in ways that violate the MP principle, signal amplification is not always necessary(Fig.S6D).

Minimum Perturbation and the Plasticity-Stability Dilemma

The plasticity-stability dilemma[38] refers to the brain's need for balancing between acquiring new skills (plasticity) and not altering existing circuits/representations in such a way that previously acquired skills are seriously affected (stability). It is particularly acute for fine discrimination PL tasks, since, as we have shown, they necessarily involve changes to early sensory areas, which need to maintain representations for a wide variety of untrained tasks[35]. We propose that during PL, this is achieved by choosing weight changes that minimize perturbation to existing weights. Indeed, MP learning induces significantly less degradation of discrimination performance of untrained stimuli than non-MP learning (e.g., if the learning rate is large) while reaching the same optimal performance for the trained task(Fig.S4).

For artificial neural networks, minimum perturbation schemes have been proposed to successfully prevent "catastrophic forgetting" of previously trained tasks when training a new task[39, 40]. These schemes suppress plasticity on synapses deemed important for previous learning episodes. Although they can be more effective than our MP model which weighs equally all synapses, they require memorizing not only the previous synaptic weights but also their contributions to previous tasks. Furthermore, this procedure may not be relevant for cases like ours where PL occurs on top of a natural baseline performance at all angles. Another advantage of our simple model is that gradient-based learning dynamics naturally finds solutions qualitatively similar to the MP solution when learning rate is slow. This may provide a normative explanation of why subjects typically require extensive training over a long time to reach asymptotic performance in PL experiments after understanding the task(e.g., in [2, 3]).

Experimental Evidence and Predictions

Our finding that PL is driven by improved sensory coding is consistent with the observed PL-induced changes to sensory representations in several electrophysiological experiments [41, 1, 2, 3, 4, 42, 5, 6, 43, 7, 8] and functional imaging studies[44, 45, 46, 47] across different model systems and tasks. Some of these studies reported representational changes that are closely related to behavioral improvements[6, 43, 7, 8], consistent with our predictions. Furthermore, we predict that for a fine discrimination task, processing between layers is approximately linear and thus all information present in an area is accessible to a linear decoder, as reported by [6]. However, our theory is inconsistent with studies that found little to no neural plasticity correlates of PL in sensory areas[48, 2, 11, 17]. Such inconsistency may arise from different task conditions and analysis methods. First, while MP synaptic changes are stronger in early layers (Fig.4G), changes to neuronal tuning properties are predicted to be more prominent in higher layers (Fig.6). This is consistent with experiments reporting bigger changes in monkey V4 than those in V1 and V2 following orientation discrimination PL[1, 2, 3, 4]. This may explain why some experiments failed to find significant changes in early sensory areas[48, 2]. Second, our theory predicts that PL-induced changes to tuning curves are localized near θ_{tr} and may cause some tuning curves to lose their pre-PL bell shapes. Thus, studies

(e.g.,[48, 2]) that excluded non-bell-shaped neurons from analysis or fitted bell-shaped functions to tuning curves may fail to detect these localized changes. Third, for motion direction discrimination in moving random dot patterns[11, 17], the relevant primary sensory layer may be MT rather than neurons in V1 with smaller receptive fields. Under this interpretation, changes in the readout layer from MT should be sufficient to yield an optimal performance.

The *types* of changes to sensory representations predicted by MP learning are largely consistent with experimental observations. At a single-cell level, MP learning causes localized sharpening of tuning curves and thus amplifies signal strength, as observed in experiments[43, 7, 3, 1, 4, 41, 42]. In addition, PL is predicted to decrease the number of neurons preferring $\theta_{\rm tr}$, as reported in [2, 3, 4]. On a population level, MP learning decreased mean noise correlation (Fig.S9), consistent with findings from simultaneous recordings of multiple units [5, 6, 7, 8]. We predict that mean firing rates of neurons responding to $\theta_{\rm tr}$ will be unaffected by learning, consistent with [5, 11, 46, 48, 42], while some others found increased[49, 50, 51, 45, 44, 3, 52, 53] or decreased activation[2, 54]. While these inconsistencies may arise from differences in tasks and setups, our theory indicates that increased/decreased activation does not play a causal role in PL[20, 6].

In terms of psychophysics, our theory predicts a rich pattern of cross-stimuli transfer. While PL transfers to stimuli highly similar to $\theta_{\rm tr}$ as expected, it causes performance for intermediate stimuli ("proximal", Fig.7) to drop below pre-PL levels. Indeed, some experiments report worse-than-baseline performance when subjects are tested on untrained stimuli following PL[55, 56, 57]. In addition, PL transfers to stimuli further away ("distal") from $\theta_{\rm tr}$. A more systematic examination how observed cross-stimuli transfer depends on stimulus similarity can further test our theory. In deriving our results, we have assumed a high-precision scenario where both signal and noise are small. A sufficiently large signal (i.e., low precision) and/or noise invalidate our assumption that a fixed subset of neurons are active throughout learning as well as during responses to the trained stimuli. PL under such conditions likely involves a broader subset of neurons and weights, potentially explaining why it leads to broader cross-stimuli transfer than high-precision PL does [58].

Many predictions from our theory depend quantitatively on the parameters σ_s, σ_w, which represent, respectively, input and weight selectivity. These parameters can be estimated indirectly by fitting experimentally observed amplitude and distribution of tuning changes (Fig.6) as well as transfer patterns (Fig.7) to our theory. The parameter σ_s can be varied by changing the width of the Gabor stimulus (Sec.M12). σ_w can be estimated from tuning properties of synaptic inputs to V1 neurons(e.g.,[59, 60, 61]). By fitting input tuning curves with different σ_w in our model to those measured from cat V1[59], we estimated σ_w in cat V1 to be in the range of of values shown here (roughly between between 0.7 and 1.3).

Finally, we mention several limitations of the present work. Our plasticity model does not include a mechanism of unsupervised learning, namely, plasticity triggered by the mere exposure to the stimulus, independent of task. Thus, task irrelevant phenomena observed in some PL studies[62, 63] are beyond the scope of the present work. Additionally, it would be interesting to add to our architecture recurrent connections within each layer, and to impose on our model units the constraints of being exclusively either excitatory or inhibitory. We also focus on how to counter input noise with learning and do not consider neuronal noise. Our scenario thus amounts to the high "external noise contrast" regime in threshold vs. contrast (TvC) analysis[64, 20], where input noise is the main factor limiting performance. Additional uncorrelated noise in the last layer is straightforward to analyze (see Fig.8B above). More general neural noise will be correlated by filtering with the feedforward and recurrent connections[29]. Studying MP learning in the presence of such noise is delegated to future study.

In conclusion, while the hierarchical nature of sensory systems in the brain can be beneficial for learning high-order categories such as objects, faces, and words, we show that a perceptual learning of fine discrimination tasks of low level features is challenged by the filtering of the signals through the multi-stage sensory systems. Our theory predicts the patterns of PL-induced changes in synaptic connections as well as the changes in neuronal responses throughout the deep sensory structure. Such changes overcome this filtering and give rise to optimal performance after training.

Acknowledgments

The authors would like to thank Andrew Saxe and Ravid Ziv for very helpful discussions. This research was partially supported by the Swartz Program in Theoretical Neuroscience at Harvard University, the Gatsby Charitable Foundation, the National Institute of Neurological Disorders and Stroke (Grant No.1U19NS104653), and the National Science Foundation (Grant No.1806818).

Methods

M1 Model and Task Setup

Our model of the sensory system consists of L layers of rectified linear neurons, connected with feedforward weights. Input to the system comes from N input channels (the 0th layer). Neurons do not have inherent noise; noise in their firing rates is entirely caused by noise in the input.

M1.1 Input Layer

The ith input channel has a preferred stimulus $\theta_i = \frac{i}{N} 2\pi$. Then the noise-averaged activities of input channels are generated with

$$f_i^0(\theta) = Z_s^{-1} \exp\left(\frac{\cos(\theta_i - \theta) - 1}{\sigma_s^2}\right), \tag{M.1}$$

where Z_s is chosen such that $\|\boldsymbol{f}^0(\theta)\| = \sqrt{N}$. The input vector fed to the network is $\boldsymbol{x}^0(\theta) = \boldsymbol{f}^0(\theta) + \boldsymbol{\epsilon}^0$, where $\boldsymbol{\epsilon}^0 \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \mathbb{I})$. Tuning and noise properties of input channels are not affected by learning.

The perceptual task consists of discrimination of two close-by stimuli, $\theta_\pm = \theta_{\text{tr}} \pm \delta\theta$. At every trial, the network is presented with either a sample of $\boldsymbol{x}^0(\theta_+)$ or $\boldsymbol{x}^0(\theta_-)$. Since the noise is Gaussian, the task can be performed optimally by a linear discriminator reading out directly from the input channels and using weights parallel to the signal,

$$\boldsymbol{s} = \left(\boldsymbol{f}^0(\theta_+) - \boldsymbol{f}^0(\theta_-)\right) / \|\boldsymbol{f}^0(\theta_+) - \boldsymbol{f}^0(\theta_-)\|. \tag{M.2}$$

To create an $O(1)$ signal-to-noise ratio in the input layer, we choose $\sigma^2 \sim O(1)$ and $\delta\theta \sim O(N^{-1/2})$. The exact values are given in Table M10.

To provide intuition for different values of σ_s, we present below some examples of 2D Gabor stimuli that are equivalent (see the procedure in Sec.M12) to different values of σ_s (Fig.M1A,B).

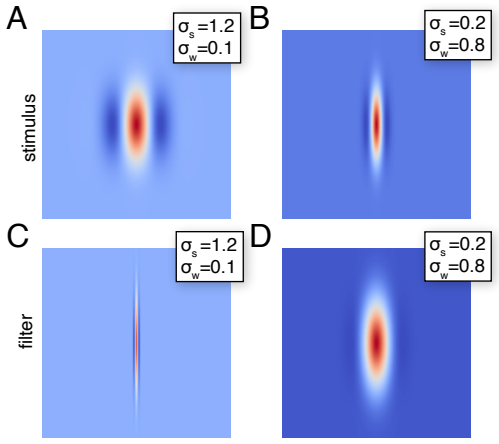

Figure M1: 2D equivalent Gabor stimuli and filters for different values of σ_s, σ_w. Both the stimuli and the filters can be rotated to correspond to different stimuli/preferred stimuli. (A,B) Example stimuli. Warmer color indicates higher intensity at that location. (C,D) Example filters. Warmer color indicates a larger filter weight for stimuli at that location.

M1.2 Model Architecture

Our model of the sensory system is a feedforward network with L hidden layers and a linear readout from the top layer. Let $\boldsymbol{x}^l(\theta)$ to denote the noisy population response vector of neurons in layer l, and $\boldsymbol{f}^l(\theta)$ its average over noise. $\{\boldsymbol{x}^l(\theta)\}$ are recursively given by

$$l = 1, 2, ..., L : \boldsymbol{x}^l(\theta) = \Phi(\boldsymbol{W}^l \boldsymbol{x}^{l-1}(\theta)),$$

where $\Phi(\cdot)$ is an element-wise activation function. The linear readout $\boldsymbol{a}$ produces a scalar network output from activity in the last layer

$$r(\theta) = \boldsymbol{a}^T \boldsymbol{x}^L(\theta).$$

Note that, by default we assume the number of input channels and the number of neurons in each layer of the network to be the same, denoted as N. In cases where we fix one and vary the other, we use N_{input} and N_{hidden} to refer to them, respectively.

M1.3 Pre-PL Weights

Pre-PL weights $\{\boldsymbol{W}^l\}_{l=1,2,...,L}$ are generated with

$$W^l_{ij,\text{pre}} = Z_w^{-1} \exp\left(\frac{\cos(\theta_i - \theta_j) - 1}{\sigma_w^2}\right) + b_w, \tag{M.3}$$

where Z_w is chosen such that each row of $\boldsymbol{W}^l_{\text{pre}}$ has norm $1/\sqrt{N}$ (i.e. each weight is $O(N^{-1})$; when $N_{\text{hidden}} \neq N_{\text{input}}$, N takes the value of the width of layer $l-1$). This normalization ensures that the noise-averaged input to any hidden neuron is $O(1)$. After this normalization, b_w is chosen such that each row sums to 0. This causes weights between neurons with very different preferred stimuli to be negative (i.e., inhibitory). 2D receptive field filters equivalent to different σ_w values are shown in Fig.M1C,D (see the procedure in Sec.M12).

The pre-PL readout $\boldsymbol{a}_{\text{pre}}$ is optimized for the discrimination task around θ_{tr} and pre-PL weights. As discussed above, the task can be performed optimally by a linear readout from the input with weights $\boldsymbol{s}$. We initialize the pre-PL readout such that it minimizes the loss function (Eq.M.4) prior to PL (see the expression in Sec.M9).

M1.4 Model of Learning

We model perceptual learning as the optimization of $\boldsymbol{W}^1, \boldsymbol{W}^2, ..., \boldsymbol{W}^L, \boldsymbol{a}$ over the loss function

$$E = \langle (r(\theta) - \boldsymbol{s}^T \boldsymbol{x}^0(\theta))^2 \rangle_{\theta = \theta_{\text{tr}} \pm \delta\theta, \boldsymbol{\epsilon}^0} \tag{M.4}$$

with gradient descent dynamics. Assuming PL to be slow, we model learning as "batch gradient descent" where each update to the weights is computed using the average loss function (i.e., E is averaged over stimuli and noise). The updates to the weights are given by

$$\Theta = \boldsymbol{W}^1, \boldsymbol{W}^2, ..., \boldsymbol{W}^L, \boldsymbol{a} : \frac{d\Theta}{dt} = -\eta \frac{dE}{d\Theta}. \tag{M.5}$$

The default value of η used in simulations is given in Table M10.

M2 Approximating Nonlinear Networks with Effective Linear Networks

In this section, we describe our approach to approximating nonlinear feedforward networks with effective linear networks during PL. We show that, for the setup described in Sec.M1,

- When the network is wide (large $N_{\text{input}}, N_{\text{hidden}}$), the linear Fisher information with respect to the stimulus θ in any layer of the nonlinear network can be approximately computed from an equivalent deep linear network.
- When the nonlinearity is the rectified linear function, every weight matrix in the equivalent deep linear network (the "effective weight matrix") is a submatrix of its counterpart in the nonlinear network. Note that the effective weights are different for discrimination around different stimuli.
- When the network is wide, the identity of the submatrix is fixed during gradient descent learning, as described in Sec.M1.4. That is to say, the effective weight matrix is always the same "part" of the full weight matrix in the nonlinear network throughout learning.

M2.1 Equivalent Linear Networks

When the network is wide, signal-induced and noise-induced fluctuation in the input to any neuron in the network is small (both scale as $N^{-1/2}$). We can thus expand activities of neurons around their average inputs (using $\odot$ to denote the element-wise product)

$$\boldsymbol{x}^l(\theta_\pm) = \Phi\left(\boldsymbol{W}^l\boldsymbol{x}^{l-1}(\theta_\pm)\right) \approx \Phi\left(\boldsymbol{W}^l\boldsymbol{f}^{l-1}(\theta_{\text{tr}})\right) + \Phi'\left(\boldsymbol{W}^l\boldsymbol{f}^{l-1}(\theta_{\text{tr}})\right) \odot \left[\pm\delta\theta\boldsymbol{W}^l d_\theta\boldsymbol{f}^{l-1} + \boldsymbol{W}^l\boldsymbol{\epsilon}^{l-1}\right], \quad \text{(M.6)}$$

where $d_\theta\boldsymbol{f}^l = \frac{\left[\boldsymbol{f}^l(\theta_+) - \boldsymbol{f}^l(\theta_-)\right]}{2\delta\theta}$ and $\boldsymbol{\epsilon}^l = \frac{1}{2}\left[\boldsymbol{x}^l(\theta_+) + \boldsymbol{x}^l(\theta_-) - \boldsymbol{f}^l(\theta_+) - \boldsymbol{f}^l(\theta_-)\right]$ are the signal- and noise-induced fluctuation in layer l, respectively. In the case of rectified linear units (ReLUs), every element of Φ' is 1 if its argument is positive and zero otherwise. For the ith neuron in layer l, we call it **active** if $[\boldsymbol{W}^l\boldsymbol{f}^{l-1}]_i > 0$ and **inactive** otherwise. In the limit of large N, inactive neurons have zero activity during typical single trials in the task. Furthermore, activities of active neurons are *linear* functions of activities of neurons in the previous layer. These linear functions are defined by corresponding effective weight matrices. Every effective weight matrix, $\boldsymbol{W}^l_{\text{eff}}$, is a submatrix of $\boldsymbol{W}^l$, given by

$$l \geq 2 : \left\{\left(W^l_{\text{eff}}\right)_{ij}\right\} = \left\{W^l_{ij}\middle| i,j : [\boldsymbol{W}^l\boldsymbol{f}^{l-1}(\theta_{\text{tr}})]_i > 0, [\boldsymbol{W}^{l-1}\boldsymbol{f}^{l-2}(\theta_{\text{tr}})]_j > 0\right\} \quad \text{(M.7)}$$

$$\left\{\left(W^1_{\text{eff}}\right)_{ij}\right\} = \left\{W^1_{ij}\middle| i : [\boldsymbol{W}^1\boldsymbol{f}^0(\theta_{\text{tr}})]_i > 0, ; j = 1, 2, ..., N\right\}. \quad \text{(M.8)}$$

The dependency of these equations on θ_{tr} highlights the fact that for the same nonlinear network, its effective linear network is different for different stimuli. We also note that effective weights for nonlinear networks at the limit of small fluctuations in inputs to individual neurons have been previously introduced in [29].

M2.2 The Identity of Effective Weights is Fixed During Learning

In the previous subsection, we showed that the identity of effective weights depends on element-wise signs of the vector $\boldsymbol{W}^l\boldsymbol{f}^{l-1}(\theta_{\text{tr}})$. During learning, $\boldsymbol{W}^l$ as well as the upstream weights may be altered, causing elements of this vector to change sign. If this happens, the identity of effective weights would shift during learning. However, we found that under the learning dynamics described in Sec.M1.4, $\boldsymbol{W}^l\boldsymbol{f}^{l-1}(\theta_{\text{tr}})$ is approximately stationary over the course of learning (assuming the network is wide; the derivations are given in Sec.M7). Thus, throughout PL, the identity of active/inactive neurons in any layer is constant. Consequently, the identity of effective weights is also fixed. To numerically verify the validity of these results, we computed how much $\boldsymbol{f}^l(\theta_{\text{tr}})$ changes over the course of PL in three-layer networks of different widths(Fig.M2). Indeed, changes are negligible for wide networks.

Importantly, MP plasticity(see Sec. M4) is derived under the assumption that MP learning does not alter the identity of effective weights. It can be verified that MP plasticity does not alter $\boldsymbol{W}^l\boldsymbol{f}^{l-1}(\theta_{\text{tr}})$ and is thus self-consistent with the assumption.

For brevity, we hereafter use $\boldsymbol{W}^1, ..., \boldsymbol{W}^L, \boldsymbol{a}$ to refer to $\boldsymbol{W}^1_{\text{eff}}, ..., \boldsymbol{W}^L_{\text{eff}}, \boldsymbol{a}_{\text{eff}}$, respectively.

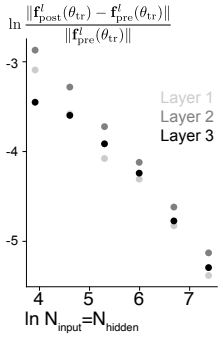

Figure M2: PL-induced changes to the noise-averaged response to the trained stimulus ($\boldsymbol{f}^l(\theta_{\rm tr})$) as a function of network width. Width of the network is controlled by adjusting the number of input channels ($N_{\rm input}$) and the number of intermediate neurons ($N_{\rm hidden}$) simultaneously. Parameters: $\sigma_s = 0.2, \sigma_w = 0.8, L = 3$, learning rate $= 10^{-3}$.

M3 Linear Fisher Information Analysis

M3.1 Loss of Fisher Information in the Effective Linear Network

Fig.2A (L=1) and Fig.S1 (for higher L) show the loss of Fisher Information in the deep network before learning. To explain this phenomenon, we first define the $N_{\rm hidden} \times N_{\rm input}$ matrix

$$\boldsymbol{P}^l \equiv \boldsymbol{W}^l \boldsymbol{W}^{l-1} ... \boldsymbol{W}^1. \tag{M.9}$$

In the effective linear network, we can write the signal and noise covariance in each layer in terms of $\boldsymbol{P}^l$ as,

$$\boldsymbol{\Sigma}_l = \sigma^2 \boldsymbol{P}^l \boldsymbol{P}^{l^T} \tag{M.10}$$

$$\boldsymbol{d}_\theta \boldsymbol{f}^l = \boldsymbol{P}^l d_\theta \boldsymbol{f}^0. \tag{M.11}$$

Assuming $\boldsymbol{P}^l$ to be of rank K (see Fig.S2A,B,E,F), we write it in terms of its truncated singular value decomposition, $\boldsymbol{P}^l = \boldsymbol{A}^l \boldsymbol{\Lambda}^l \boldsymbol{B}^l$, where $\boldsymbol{A}^l \in \mathbb{R}^{N_{\rm hidden} \times K}$ has orthonormal columns, $\boldsymbol{B}^l \in \mathbb{R}^{K \times N_{\rm input}}$ has orthonormal rows, and $\Lambda^l \in \mathbb{R}^{K \times K}$ is a diagonal matrix with the nonzero eigenvalues. Then,

$$\frac{J_l}{J_0} = \|\boldsymbol{B}^l \boldsymbol{s}\|^2. \tag{M.12}$$

$\boldsymbol{B}^l \boldsymbol{s}$ is $\boldsymbol{s}$ projected onto K orthonormal vectors. Before learning $\boldsymbol{s}$ is not fully embedded in the subspace spanned by these vectors (Fig.S2C,E), hence $J_l/J_0 < 1$. After learning, the rank of the post-PL B_l does not drastically change. However, its orthogonal vectors has been rotated so that the signal is fully spanned by them, recovering the full information, J_0 (Fig.S2D, H).

When the structure of $\boldsymbol{W}^l$ is smooth, both K and the orientation of the orthonormal vectors are not dependent on $N_{\rm input}, N_{\rm hidden}$. Thus, J_l/J_0, does not depend on network width. Since J_0 is linear in $N_{\rm input}$, J_l is also linear in $N_{\rm input}$, which we show for the pre-PL network in Fig.S8 and for the post-PL network in Fig.8A.

M3.2 Information Scaling in the Presence of Output Noise

In this subsection, we compute the linear Fisher information in the top layer L in the scenario where i.i.d. Gaussian noise with variance $\sigma^2_{\rm output}$ is added to the activity of layer L neurons. In particular, we assume

that the network has completed PL and thus performs optimally, before the output noise is injected. It is given by

$$J_L = d_\theta \boldsymbol{f}^L(\theta)^T (\sigma^2 \boldsymbol{P}^L {\boldsymbol{P}^L}^T + \sigma^2_{\text{output}} \mathbb{I})^{-1} d_\theta \boldsymbol{f}^L(\theta). \tag{M.13}$$

We assume both N_{input} and N_{hidden} to be large enough such that the approximations introduced in Sec.M2 hold. Writing $\boldsymbol{P}^L$ in terms of its SVD (see the previous subsection) and $d_\theta \boldsymbol{f}^L(\theta) = \boldsymbol{P}^L d_\theta \boldsymbol{f}^0(\theta)$,

$$J_L = \sum_{i=1}^{K} \frac{{\Lambda^L_{ii}}^2 [\boldsymbol{B}^L d_\theta \boldsymbol{f}^0(\theta)]_i^2}{\sigma^2_{\text{output}} + \sigma^2 {\Lambda^L_{ii}}^2}. \tag{M.14}$$

In this expression, $\sum_i^k {\Lambda^L_{ii}}^2 = \|\boldsymbol{P}^L\|^2 \sim O(N_{\text{hidden}}/N_{\text{input}})$. Since K does not depend on N_{input} nor N_{hidden}, ${\Lambda^L_{ii}}^2 \sim O(N_{\text{hidden}}/N_{\text{input}})$. On the other hand, $\sum_i^K [\boldsymbol{B}^L d_\theta \boldsymbol{f}^0(\theta)]_i^2 = \|\boldsymbol{B}^L \boldsymbol{s}\|^2 \|d_\theta \boldsymbol{f}^0(\theta)\|^2$. $\|\boldsymbol{B}^L \boldsymbol{s}\|^2$ does not scale with network width, as we have argued in the previous subsection; $\|d_\theta \boldsymbol{f}^0(\theta)\|^2 \sim O(N_{\text{input}})$. It follows that $[\boldsymbol{B}^L d_\theta \boldsymbol{f}^0(\theta)]_i^2 \sim O(N_{\text{input}})$.

The scaling relations explain the curves in Fig.8B. When N_{hidden} is small, the denominator is dominated by σ^2_{output}. When N_{hidden} is increased beyond approximately $N_{\text{input}}\sigma^2_{\text{output}}/\sigma^2$, the denominator is dominated by the input noise term. Thus, information saturates with growing N_{hidden} and approaches information without output noise (J_0).

M4 Minimum Perturbation (MP) Learning

MP changes to weights for PL are defined as those with minimal L2 norms that would still minimize the loss function (Eq.M.4). They are the solution to the constrained optimization problem

$$\min_{\Delta \boldsymbol{W}^1, ..., \Delta \boldsymbol{W}^L, \Delta \boldsymbol{a}} \|\Delta \boldsymbol{W}^1\|^2 + ... + \|\Delta \boldsymbol{W}^L\|^2 + \|\Delta \boldsymbol{a}\|^2$$

subject to

$$E(\boldsymbol{W}^1_{\text{pre}} + \Delta \boldsymbol{W}^1, ..., \boldsymbol{W}^L_{\text{pre}} + \Delta \boldsymbol{W}^L, \boldsymbol{a}_{\text{pre}} + \Delta \boldsymbol{a}) = 0. \tag{M.15}$$

For networks of any depth, the solutions have the general structure

$$\Delta \boldsymbol{W}^l = ({\boldsymbol{W}^{l+1}_{\text{post}}}^T \cdots {\boldsymbol{W}^L_{\text{post}}}^T \boldsymbol{a}_{\text{post}})(\boldsymbol{W}^{l-1}_{\text{post}} \cdots \boldsymbol{W}^1_{\text{post}} \boldsymbol{\lambda})^T. \tag{M.16}$$

where $\lambda \in \mathbb{R}^{N_{\text{input}}}$ is a Lagrange multiplier vector that enforces the constraint of zero error. As shown here, changes to every weight matrix are confined to a rank-1 matrix. The left vector is a linear readout from the lth layer *after PL*; the right vector is the effective signal, $\boldsymbol{\lambda}$, propagated through the *post-PL* upstream layers.

The full solution needs to be obtained by solving a system of self-consistent equations. We present these equations and further details for the cases of $L = 1, 2, 3$ in Sec.M10.

M5 Is signal amplification required for PL?

M5.1 Minimal Signal Amplitude in the Last Layer Required for PL

Both MP learning and gradient-descent learning lead to negligible changes to the readout vector $\boldsymbol{a}$. Assuming the readout to be fixed, we ask whether it is possible to complete PL without increasing the signal amplitude in last-layer representations by computing the minimal signal amplitude in the last layer required for PL.

For PL, the necessary and sufficient condition in Eq.3 translates to analogous conditions on the post PL value of the product matrix $\boldsymbol{P}^L$, i.e., $\boldsymbol{P}^L = \boldsymbol{u}\boldsymbol{s}^T + \boldsymbol{P}_\perp$, $\boldsymbol{P}_\perp \boldsymbol{s} = \boldsymbol{0}$, $\boldsymbol{P}_\perp^T \boldsymbol{a} = \boldsymbol{0}$. In addition, in order to minimize the loss function(Eq.M.4), $\boldsymbol{u}^T \boldsymbol{a} = 1$. The squared signal amplitude in the last layer is given by

$$\|d_\theta \boldsymbol{f}^L\big|_{\theta_{\text{tr}}}\|^2 = \|\boldsymbol{P}^L d_\theta \boldsymbol{f}^0\big|_{\theta_{\text{tr}}}\|^2 = \|\boldsymbol{u}\|^2 \|d_\theta \boldsymbol{f}^0\big|_{\theta_{\text{tr}}}\|^2 + \|\boldsymbol{P}_\perp d_\theta \boldsymbol{f}^0\|^2. \tag{M.17}$$

This is minimized (under the constraint that $\boldsymbol{u}^T\boldsymbol{a} = 1$) by $\boldsymbol{u} = \|\boldsymbol{a}\|^{-2}\boldsymbol{a}$ and $\boldsymbol{P}_\perp = \boldsymbol{0}$. Thus, the minimal post-PL signal amplitude is

$$\min\|d_\theta \boldsymbol{f}^L\big|_{\theta_{\rm tr}}\| = \|\boldsymbol{a}\|^{-1}\|d_\theta \boldsymbol{f}^0\big|_{\theta_{\rm tr}}\|. \tag{M.18}$$

As shown in Fig.S6A,B, the minimal post-PL signal amplitude is larger than the pre-PL one across parameters. Thus, signal amplification is indeed necessary for PL, assuming a fixed $\boldsymbol{a}$. We also note that the signal amplitude after MP learning is close to the minimal level.

M5.2 PL with An Amplified Readout Vector

A reasonable extension of our model is to consider the case where the direction of the readout vector stays the same but not amplitude is increased, namely ($\boldsymbol{a}_{\rm post} = (1+c)\boldsymbol{a}_{\rm pre}$), b We computed MP changes to weight matrices for several values of c (for networks with $L = 2$) and find that if c is sufficiently large, MP learning may even lead to a *decreased* signal amplitude (Fig.S6C).

M5.3 Learning Under a Soft MP Constraint

Compared to a non-MP post-PL solution to PL, does MP learning lead to smaller signal amplitudes? We consider a "soft" MP constraint where changes to the weights fluctuate in the space of solutions around MP changes. Concretely, for networks with one-layer and a fixed $\boldsymbol{a}$, we sample solution $\boldsymbol{W}^1$ with

$$\boldsymbol{W}^1_{\text{soft MP}} = \boldsymbol{W}^1_{\rm pre} + \Delta\boldsymbol{W}^1_{\rm MP} + \left(\mathbb{I} - \frac{\boldsymbol{a}\boldsymbol{a}^T}{\|\boldsymbol{a}\|^2}\right)\boldsymbol{E}, \tag{M.19}$$

where $E_{ij} \sim \mathcal{N}(0, \sigma^2_{\text{soft MP}}/N)$ and $\Delta\boldsymbol{W}^1_{\rm MP} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|^2}\boldsymbol{s}^T - \frac{\boldsymbol{a}\boldsymbol{a}^T\boldsymbol{W}^1_{\rm pre}}{\|\boldsymbol{a}\|^2}$ is the MP changes to $\boldsymbol{W}^1$ (derived under Sec.M10).

Every sampled $\boldsymbol{W}^1$ solves PL. We computed the post-PL signal amplitude for various $\sigma^2_{\text{soft MP}}$. $\sigma^2_{\text{soft MP}} = 0$ is the MP changes (Fig.S6D). We found that MP learning does indeed lead to a smaller signal amplitude than the average "soft" MP learning.

M6 Cross-Task Transfer Test

To test whether perceptual learning on the θ discrimination task transfers to a different task, we devised a σ_s discrimination task. In this task, the network has to discriminate between two close-by values of σ_s, $\sigma_{s,\rm tr}\pm\delta\sigma_s$ with the same θ. We ensure that the two tasks have the same difficulty by choosing values of $\delta\sigma_s$ such that $\|\delta\theta d_\theta \boldsymbol{f}^0\| = \frac{1}{2}\|\boldsymbol{f}^0(\theta_{\rm tr}; \sigma_s + \delta\sigma_s) - \boldsymbol{f}^0(\theta_{\rm tr}, \sigma_s - \delta\sigma_s)\|$. For typical network parameters that we considered, the performance on σ_s discrimination is suboptimal prior to training, leaving room for learning. Finally, we assume that the network uses two separate linear readouts from the last layer, $\boldsymbol{a}$ for θ discrimination and $\boldsymbol{a'}$ for σ_s discrimination. Importantly, the averaged stimuli in the σ_s discrimination task and the θ discrimination task are both $\boldsymbol{f}^0(\theta_{\rm tr})$.

To test for the presence of cross-task transfer, we trained the network on θ discrimination with gradient descent. At various points during this training (which does not affect $\boldsymbol{a'}$), we paused and optimized $\boldsymbol{a'}$ for σ_s discrimination under current weights and computed its performance (Fig.S7).

Our results suggest that PL for θ discrimination does not transfer to σ_s discrimination. This does not suggest that cross-task transfer *cannot* occur but merely provides an example where it *does not* occur despite extensive representational changes.

Extended Methods

M7 Stationarity of Identity of Effective Weights

In this section, we show that the identity of effective weights, introduced in Sec.M2, is approximately stationary during gradient-based PL(Sec.M1.4). For brevity, denote

$$h_i^l(\theta_\pm) = [\boldsymbol{W}^l \boldsymbol{x}^{l-1}(\theta_\pm)]_i, \tag{M.20}$$

and write $\boldsymbol{f}^0 = \boldsymbol{f}^0$. With this, input to the network can be written as

$$\boldsymbol{x}^0(\theta_\pm) = \boldsymbol{f}^0 \pm \delta\theta d_\theta \boldsymbol{f}^0 + \boldsymbol{\epsilon}^0. \tag{M.21}$$

M7.1 Networks With One Hidden Layer

In this case, output of the network is given by

$$r_\pm \equiv \sum_i a_i \phi\left({\boldsymbol{W}_i^1}^T \boldsymbol{x}^0(\theta_\pm)\right), \tag{M.22}$$

where $\boldsymbol{W}_i^1$ is the ith row of $\boldsymbol{W}^1$. Rewrite Eq.M.4 as

$$E(\boldsymbol{W}^1, \boldsymbol{a}) = \frac{1}{2}\langle (r_+ - \hat{r_+})^2 + (r_- - \hat{r_-})^2 \rangle, \tag{M.23}$$

where $\hat{r}_\pm = \boldsymbol{s}^T(\boldsymbol{f}^0 \pm \delta\theta d_\theta \boldsymbol{f}^0 + \boldsymbol{\epsilon}^0)$ and $\langle\cdot\rangle$ denotes average over noise.

Gradient for $\boldsymbol{W}_i^1$ is given by

$$\left\langle \nabla_{\boldsymbol{W}_i^1} E \right\rangle = \frac{1}{2}\left\langle \nabla_{\boldsymbol{W}_i^1}[(r_+ - \hat{r_+})^2] + \nabla_{\boldsymbol{W}_i^1}[(r_- - \hat{r_-}])^2 \right\rangle. \tag{M.24}$$

For brevity, we only examine the first term (analysis of the second term is very similar).

$$\frac{1}{2}\left\langle \nabla_{\boldsymbol{W}_i^1} (r_+ - \hat{r_+})^2 \right\rangle = \left\langle (r_+ - \hat{r_+}) \left(\frac{dr_+}{dx_i^1(\theta_+)}\right) \phi'(h_i^1(\theta_+)) \boldsymbol{x}^0(\theta_+) \right\rangle \tag{M.25}$$

$$= a_i \left\langle (r_+ - \hat{r_+}) \phi'(h_i^1(\theta_+)) \boldsymbol{x}^0(\theta_+) \right\rangle \tag{M.26}$$

$$= a_i \left\langle (r_+ - \hat{r_+}) \phi'(h_i^1(\theta_+)) \right\rangle \boldsymbol{f}^0 + a_i \left\langle (r_+ - \hat{r_+}) \phi'(h_i^1(\theta_+)) \right\rangle \delta\theta d_\theta \boldsymbol{f}^0 + a_i \left\langle (r_+ - \hat{r_+}) \phi'(h_i^1(\theta_+)) \boldsymbol{\epsilon}^0 \right\rangle. \tag{M.27}$$

Applying Stein's lemma to the last term yields

$$\begin{aligned} \frac{1}{2}\langle \nabla_{\boldsymbol{W}_i^1} (r_+ - \hat{r_+})^2 \rangle =& a_i \left\langle (r_+ - \hat{r_+}) \phi'(h_i^1(\theta_+)) \right\rangle \boldsymbol{f}^0 + a_i \left\langle (r_+ - \hat{r_+}) \phi'(h_i^1(\theta_+)) \right\rangle \delta\theta d_\theta \boldsymbol{f}^0 \\ &+ a_i \sigma^2 \left\langle \phi'(h_i^1(\theta_+)) \nabla_{\boldsymbol{\epsilon}^0} (r_+ - \hat{r_+}) \right\rangle \\ &+ a_i \sigma^2 \left\langle (r_+ - \hat{r_+}) \phi''(h_i^1(\theta_+)) \right\rangle \boldsymbol{W}_i^1. \end{aligned} \tag{M.28}$$

Every $\langle\cdot\rangle$ contains the product of two random variables, which can be written as $\langle XY\rangle = \langle X\rangle\langle Y\rangle + \mathrm{cov}(X,Y)$. At large N, it can be verified that each is dominated by $\langle X\rangle\langle Y\rangle$. Eliminating covariance terms and assuming that $\langle r_+ - \hat{r_+}\rangle = -\langle r_- - \hat{r_-}\rangle$ throughout training, Eq.M.24 becomes

$$\begin{aligned} \langle \nabla_{\boldsymbol{W}_i^1} E \rangle \approx& a_i \langle r_+ - \hat{r_+} \rangle \left\langle \phi'(h_i^1(\theta_+)) - \phi'(h_i^1(\theta_-)) \right\rangle \boldsymbol{f}^0 + a_i \langle r_+ - \hat{r_+} \rangle \left\langle \phi'(h_i^1(\theta_+)) + \phi'(h_i^1(\theta_-)) \right\rangle \delta\theta d_\theta \boldsymbol{f}^0 \\ &+ a_i \sigma^2 \left\langle \phi'(h_i^1(\theta_+)) + \phi'(h_i^1(\theta_-)) \right\rangle (\boldsymbol{V} - \boldsymbol{s}) + a_i \sigma^2 \langle r_+ - \hat{r_+} \rangle \left\langle \phi''(h_i^1(\theta_+)) - \phi''(h_i^1(\theta_-)) \right\rangle \boldsymbol{W}_i^1, \end{aligned} \tag{M.29}$$

where $\boldsymbol{V}$ is defined in Eq.2. Under the linear approximations, we assume $\left\langle \phi'(h_i^l(\theta_+)) - \phi'(h_i^l(\theta_-)) \right\rangle = 0$. Thus,

$$\langle \nabla_{\boldsymbol{W}_i^1} E \rangle \approx 2a_i \langle r_+ - \hat{r_+} \rangle \langle \phi'(h_i^l(\theta_+)) \rangle \delta\theta d_\theta \boldsymbol{f}^0 + 2a_i \sigma^2 \langle \phi'(h_i^l(\theta_+)) \rangle (\boldsymbol{V} - \boldsymbol{s}) . \tag{M.30}$$

$\boldsymbol{V}$ is perpendicular to $\boldsymbol{f}^0$ because otherwise the network would be a biased discriminator. In addition, $d_\theta \boldsymbol{f}^0 \perp \boldsymbol{f}^0$. Thus, $\langle \nabla_{\boldsymbol{W}_i^1} E \rangle \perp \boldsymbol{f}^0$ and $\boldsymbol{W}^1 \boldsymbol{f}^0$ is stationary in time during learning.

M7.2 Generalization to Deeper Networks

To show that the averaged input is stationary in deeper networks, we first derive a general expression for the dynamics of $\boldsymbol{W}^l$ during training in a deep network. To facilitate discussion, define $\tilde{\boldsymbol{W}}_{\text{eff}}^l$ to be the padded effective matrix of dimension $N \times N$ defined as

$$\tilde{W}_{\text{eff}ij}^l = \begin{cases} W_{ij}^l & \text{if } W_{ij}^l \in \boldsymbol{W}_{\text{eff}}^l \\ 0 & \text{otherwise.} \end{cases} \tag{M.31}$$

Define $\tilde{\boldsymbol{P}}^l \equiv \tilde{\boldsymbol{W}}_{\text{eff}}^l \tilde{\boldsymbol{W}}_{\text{eff}}^{\tilde{l-1}} ... \tilde{\boldsymbol{W}}_{\text{eff}}^1$ and $\tilde{\boldsymbol{\Sigma}} \equiv \sigma^2 \tilde{\boldsymbol{P}}^l \tilde{\boldsymbol{P}}^{l^T}$. Dynamics for $\boldsymbol{W}^l$ can be derived from

$$\begin{aligned} \langle \nabla_{\boldsymbol{W}_i^l} (r_+ - \hat{r_+})^2 \rangle =& \left\langle \frac{dr_+}{dx_i^l(\theta_+)} (r_+ - \hat{r_+}) \phi'(h_i^l(\theta_+)) \right\rangle \boldsymbol{f}^{l-1} + \left\langle \frac{dr_+}{dx_i^l(\theta_+)} (r_+ - \hat{r_+}) \phi'(h_i^l(\theta_+)) \right\rangle \delta\theta d_\theta \boldsymbol{f}^{l-1} \\ &+ \left\langle \frac{dr_+}{dx_i^l(\theta_+)} (r_+ - \hat{r_+}) \phi'(h_i^l(\theta_+)) \boldsymbol{\epsilon}^{l-1} \right\rangle, \end{aligned} \tag{M.32}$$

where $\frac{dr_+}{dx_i^l(\theta_+)}$ is a random variable with $O(N^{-1/2})$ mean and fluctuation. Applying Stein's lemma to the last term and eliminating non-leading order terms to get (let $\tilde{a}_i^l = \left\langle \frac{dr_+}{dx_i^l(\theta_+)} \right\rangle$)

$$\begin{aligned} \langle \nabla_{\boldsymbol{W}_i^l} (r_+ - \hat{r_+})^2 \rangle \approx& \tilde{a}_i^l \langle r_+ - \hat{r_+} \rangle \langle \phi'(h_i^l(\theta_+)) \rangle \boldsymbol{f}^{l-1} \\ &+ \tilde{a}_i^l \langle r_+ - \hat{r_+} \rangle \langle \phi'(h_i^l(\theta_+)) \rangle \delta\theta d_\theta \boldsymbol{f}^{l-1} \\ &+ \tilde{a}_i^l \langle \phi'(h_i^l(\theta_+)) \rangle \left(\boldsymbol{\Sigma}^{\tilde{l-1}} \tilde{\boldsymbol{a}}^{l-1} - \sigma^2 \boldsymbol{P}^{\tilde{l-1}} \boldsymbol{s} \right) . \end{aligned} \tag{M.33}$$

Combine $\langle \nabla_{\boldsymbol{W}_i^l} (r_+ - \hat{r_+})^2 \rangle$ and $\langle \nabla_{\boldsymbol{W}_i^l} (r_- - \hat{r_-})^2 \rangle$ to get

$$\langle \nabla_{\boldsymbol{W}_i^l} E \rangle = 2\tilde{a}_i^l \langle \phi'(h_i^l(\theta_+)) \rangle \left[\langle r_+ - \hat{r_+} \rangle \delta\theta d_\theta \boldsymbol{f}^{l-1} + \tilde{\boldsymbol{\Sigma}_{l-1}} \tilde{\boldsymbol{a}}^{l-1} - \sigma^2 \boldsymbol{P}^{\tilde{l-1}} \boldsymbol{s} \right] . \tag{M.34}$$

Note that under mean-field approximations, $\tilde{\boldsymbol{a}}^{l-1} = (\tilde{\boldsymbol{W}}_{\text{eff}}^l)^T (\tilde{\boldsymbol{W}}_{\text{eff}}^{\tilde{l+1}})^T ... (\tilde{\boldsymbol{W}}_{\text{eff}}^L)^T \boldsymbol{a}$. We have

$$\begin{aligned} \langle \nabla_{\boldsymbol{W}_i^l} E \rangle &= 2\tilde{a}_i^l \langle \phi'(h_i^l(\theta_+)) \rangle \left[\langle r_+ - \hat{r_+} \rangle \delta\theta d_\theta \boldsymbol{f}^{l-1} + \sigma^2 \tilde{\boldsymbol{P}}^{l-1} \boldsymbol{V} - \sigma^2 \tilde{\boldsymbol{P}}^{l-1} \boldsymbol{s} \right] \\ &= 2\tilde{a}_i^l \langle \phi'(h_i^l(\theta_+)) \rangle \tilde{\boldsymbol{P}}^{l-1} \left[\langle r_+ - \hat{r_+} \rangle \delta\theta d_\theta \boldsymbol{f}^0 + \sigma^2 \boldsymbol{V} - \sigma^2 \boldsymbol{s} \right] . \end{aligned} \tag{M.35}$$

We proceed to show that every component of $\langle \nabla_{\boldsymbol{W}_i^l} E \rangle$ is perpendicular to $\boldsymbol{f}^{l-1}$.

First, define a notion of parity for vectors. For an N-dimensional vector $\boldsymbol{v}$, it is odd if for all j, $v_{N/2-j} = -v_{N/2+j}$; we call it even if $v_{N/2-j} = v_{N/2+j}$. Without loss of generality, we consider the scenario where the input neuron preferring the trained stimulus has index $N/2$. It is easy to see that $d_\theta \boldsymbol{f}^0$ is an odd vector while $\boldsymbol{f}^0$ is an even vector. Furthermore, any odd vector is perpendicular to any even vector.

We make the ansatz that throughout training, $\boldsymbol{V}$ is an odd vector. If $\boldsymbol{V}$ was not odd (that is, it is even or the sum of even and odd vectors), its even component would be perpendicular to the signal $d_\theta \boldsymbol{f}^0$. This component would therefore be suboptimal because it would contribute to noise without contributing to signal. Unlike other sources of suboptimality discussed in the main text, this component can easy be

removed by making $\boldsymbol{a}$ an odd vector. Since we optimize $\boldsymbol{a}$ before learning, this component does not exist at the beginning of learning.

Since pre-PL weight matrices are circulant, it is easy to verify that they have the following property: if $\boldsymbol{v}$ is an odd/even vector, then $\boldsymbol{W}^l\boldsymbol{v}$ and $(\boldsymbol{W}^l)^T\boldsymbol{v}$ are also odd/even. Further, it can be verified that pre-PL effective weight matrices have the same property. We say that these matrices *preserve vector parity*. Product of matrices preserving vector parity preserves parity itself.

We now show that throughout learning, weight matrices and effective matrices preserve vector parity. Assume that at time t, weight matrices still preserve vector parity. Note that the gradient is a rank-1 matrix. The right vector of the gradient is an odd vector, since $d_\theta \boldsymbol{f}^0, \boldsymbol{V}, \boldsymbol{s}$ are all odd vectors, and the product matrix preserves parity. The left vector, which can be written as the element-wise product between an odd vector $\tilde{\boldsymbol{a}^l}$ and an even vector $\left\langle \phi'(\boldsymbol{h}^l(\theta_+))\right\rangle$, is an odd vector. Therefore, for a small τ, this matrix at time $t+\tau$ can be written as

$$\boldsymbol{W}^l_{t+\tau} = \boldsymbol{W}^l_t + \tau \boldsymbol{o}_1 \boldsymbol{o}_2^T, \tag{M.36}$$

where $\boldsymbol{o}_{1,2}$ are odd vectors. This new matrix will again preserve vector parity since (letting $\boldsymbol{o}$ denote any odd vector)

$$\boldsymbol{W}^l_{t+\tau}\boldsymbol{o} = \boldsymbol{W}^l_t\boldsymbol{o} + \tau \boldsymbol{o}_1 \boldsymbol{o}_2^T\boldsymbol{o}, \tag{M.37}$$

is a sum of two odd vectors and therefore still an odd vector and (letting $\boldsymbol{e}$ denote any even vector)

$$\boldsymbol{W}^l_{t+\tau}\boldsymbol{e} = \boldsymbol{W}^l_t\boldsymbol{e} + \tau \boldsymbol{o}_1 \boldsymbol{o}_2^T\boldsymbol{e} = \boldsymbol{W}^l_t\boldsymbol{e}, \tag{M.38}$$

which is even. Since weight matrices preserve parity at initialization, we can show by induction that they do so throughout learning.

We now return to Eq.M.35. Since weight matrices at all times preserve vector parity, the mean response in layer $l-1$, $\boldsymbol{f}^{l-1} = \tilde{\boldsymbol{W}^{l-1}_{\text{eff}}}\tilde{\boldsymbol{W}^{l-2}_{\text{eff}}}...\tilde{\boldsymbol{W}^{1}_{\text{eff}}}\boldsymbol{f}^0$ is always even.

In addition, $\langle \nabla_{\boldsymbol{W}^l_i} E\rangle$ is always odd because $\tilde{\boldsymbol{P}}^{l-1}$ always preserve parity. Therefore, $\langle \nabla_{\boldsymbol{W}^l_i} E\rangle$ is always perpendicular to $\boldsymbol{f}^{l-1}(\theta_{\text{tr}})$.

For brevity, hereafter we use $\boldsymbol{W}^1, ..., \boldsymbol{W}^L, \boldsymbol{a}$ to refer to $\boldsymbol{W}^1_{\text{eff,post}}, ..., \boldsymbol{W}^L_{\text{eff,post}}, \boldsymbol{a}_{\text{eff,post}}$, respectively; we use $\boldsymbol{W}^1_0, ..., \boldsymbol{W}^L_0, \boldsymbol{a}_0$ to refer to $\boldsymbol{W}^1_{\text{eff,pre}}, ..., \boldsymbol{W}^L_{\text{eff,pre}}, \boldsymbol{a}_{\text{eff,pre}}$, respectively.

M8 The Space of Solutions

In this section, we show that all $\boldsymbol{W}^1$ that solves the perceptual task satisfy the condition in Eq.3 and vice versa. Formally, let $\tilde{\boldsymbol{a}} = \boldsymbol{a}^T\boldsymbol{W}^L\boldsymbol{W}^{L-1}...\boldsymbol{W}^2$. We claim

$$\boldsymbol{W^1}^T\tilde{\boldsymbol{a}} = \boldsymbol{s} \iff \boldsymbol{W}^1 = \boldsymbol{u}\boldsymbol{s}^T + \boldsymbol{W}_\perp, \quad \boldsymbol{W}_\perp\boldsymbol{s} = 0, \quad \boldsymbol{W}_\perp^T\tilde{\boldsymbol{a}} = 0, \quad \boldsymbol{u}^T\tilde{\boldsymbol{a}} = 1. \tag{M.39}$$

Proof. Let $\boldsymbol{W}^1 = \sum_i \lambda_i \boldsymbol{\alpha}_i\boldsymbol{\beta}_i^T$ be its singular value decomposition. Then let $\hat{\boldsymbol{\beta}}_i = \boldsymbol{\beta}_i - \gamma_i\boldsymbol{s}$ s.t. $\hat{\boldsymbol{\beta}}_i \perp \boldsymbol{s}$ and $\hat{\boldsymbol{\alpha}}_i = \boldsymbol{\alpha}_i - \epsilon_i\tilde{\boldsymbol{a}}$ s.t. $\hat{\boldsymbol{\alpha}}_i \perp \tilde{\boldsymbol{a}}$. We have

$$\boldsymbol{W}^1 = \sum_i \lambda_i\hat{\boldsymbol{\alpha}}_i\hat{\boldsymbol{\beta}}_i^T + \sum_i \lambda_i\gamma_i\hat{\boldsymbol{\alpha}}_i\boldsymbol{s}^T + \sum_i \lambda_i\gamma_i\epsilon_i\tilde{\boldsymbol{a}}\boldsymbol{s}^T + \sum_i \lambda_i\epsilon_i\tilde{\boldsymbol{a}}\hat{\boldsymbol{\beta}}_i^T. \tag{M.40}$$

In order to satisfy $\boldsymbol{W^1}^T\tilde{\boldsymbol{a}} = \boldsymbol{s}$, the last term must be zero. Combine the second and third terms,

$$\boldsymbol{W}^1 = \sum_i \lambda_i\hat{\boldsymbol{\alpha}}_i\hat{\boldsymbol{\beta}}_i^T + \sum_i \lambda_i\gamma_i\boldsymbol{\alpha}_i\boldsymbol{s}^T. \tag{M.41}$$

Let $\boldsymbol{u} \equiv \sum_i \lambda_i\gamma_i\boldsymbol{\alpha}_i$. To satisfy the constraint, it must have $\boldsymbol{u}^T\tilde{\boldsymbol{a}} = 1$. Also let $\boldsymbol{W}_\perp \equiv \sum_i \lambda_i\hat{\boldsymbol{\alpha}}_i\hat{\boldsymbol{\beta}}_i^T$. It can be verified that $\boldsymbol{W}_\perp\boldsymbol{s} = 0, \boldsymbol{W}_\perp\tilde{\boldsymbol{a}} = 0$.

M9 Closed-Form Initialization of Readout Weights

Under the linear approximations, the optimization problem in Eq.M.4 can be solved in closed-form. We can express the optimal $\boldsymbol{a}_0$ as

$$\boldsymbol{a}_0 = \langle \boldsymbol{P}^L \boldsymbol{x}^0(\theta)\boldsymbol{x}^0(\theta)^T \boldsymbol{P}^{L^T} \rangle^{\dagger}_{\theta=\theta_\pm,\boldsymbol{\epsilon}^0} \boldsymbol{P}^L \langle \boldsymbol{x}^0(\theta)\boldsymbol{x}^0(\theta)^T \rangle_{\theta=\theta_\pm,\boldsymbol{\epsilon}^0} \boldsymbol{s}, \tag{M.42}$$

where $\boldsymbol{P}^L$ is defined in Sec.M3.1. This can be written more explicitly in term of the truncated singular value decomposition of $\boldsymbol{P}^L = \boldsymbol{A}^L \boldsymbol{\Lambda}^L \boldsymbol{B}^L$ (see Sec.M3.1) as

$$\boldsymbol{a}_0 = \frac{1+\mathrm{SNR}}{1+\mathrm{SNR}/\|\boldsymbol{B}^L\boldsymbol{s}\|^2} \boldsymbol{A}^L {\boldsymbol{\Lambda}^L}^{-1} \boldsymbol{B}^L \boldsymbol{s}, \tag{M.43}$$

where $\mathrm{SNR} = (\delta\theta)^2 \|d_\theta \boldsymbol{f}^0\|^2/\sigma^2$ is the input signal-to-noise ratio, which is set to 1 in simulations (see Sec.M11).

M10 Derivation of Minimum Perturbation (MP) Modifications

MP modifications are solutions to the constrained optimization problem posed in Eq.M.15. $\Delta\boldsymbol{W}^l = \boldsymbol{W}^l - \boldsymbol{W}_0$ and similarly for $\Delta\boldsymbol{a}$. We hereby provide solutions for $L = 1, 2, 3$.

M10.1 L=1

Define the Lagrangian

$$\mathcal{L} = \|\Delta\boldsymbol{W}^1\|^2 + \|\Delta\boldsymbol{a}\|^2 - \boldsymbol{\lambda}^T \left[\left(\Delta\boldsymbol{W}^1 + \boldsymbol{W}_0^1\right)^T (\Delta\boldsymbol{a} + \boldsymbol{a}_0) - \boldsymbol{s} \right], \tag{M.44}$$

where $\boldsymbol{\lambda}$ is a vector of N Lagrange multipliers. Extremizing the Lagrangian w.r.t. $\Delta\boldsymbol{W}$ and $\Delta\boldsymbol{a}$ yields

$$\Delta\boldsymbol{W}^1 = (\Delta\boldsymbol{a} + \boldsymbol{a}_0)\,\boldsymbol{\lambda}^T \tag{M.45}$$

$$\Delta\boldsymbol{a} = \left(\Delta\boldsymbol{W}^1 + \boldsymbol{W}_0^1\right)^T \boldsymbol{\lambda}. \tag{M.46}$$

Solve for $\boldsymbol{\lambda}$ to get

$$\boldsymbol{\lambda} = \left[(1-\|\boldsymbol{\lambda}\|^2)^{-1} {\boldsymbol{W}_0^1}^T \boldsymbol{W}_0^1 + \|\Delta\boldsymbol{a} + \boldsymbol{a}_0\|^2 \mathbb{I} \right]^{-1} \left[\boldsymbol{s} - (1-\|\boldsymbol{\lambda}\|^2)^{-1} {\boldsymbol{W}_0^1}^T \boldsymbol{a}_0 \right], \tag{M.47}$$

where $\mathbb{I}$ is the N-dimensional identity matrix. Defining scalar order parameters

$$\alpha = (1-\|\boldsymbol{\lambda}\|^2), \quad \beta = \|\Delta\boldsymbol{a} + \boldsymbol{a}_0\|^2, \tag{M.48}$$

we have

$$\boldsymbol{\lambda} = \left[\alpha^{-1} {\boldsymbol{W}_0^1}^T \boldsymbol{W}_0^1 + \beta\mathbb{I} \right]^{-1} \left[\boldsymbol{s} - \alpha^{-1} {\boldsymbol{W}_0^1}^T \boldsymbol{a}_0 \right]. \tag{M.49}$$

This expression can be plugged back into definitions of α, β to obtain two self-consistent equations for α, β. Values of the order parameters can then be solved numerically, yielding α^*, β^*. Plugging these back into expressions for $\Delta\boldsymbol{a}, \Delta\boldsymbol{W}^1$ gives the solution.

Assuming fixed $\boldsymbol{a}$ If we assume a fixed $\boldsymbol{a}$, the MP $\Delta\boldsymbol{W}^1$ can be given in closed form as

$$\Delta\boldsymbol{W}^1 = \frac{\boldsymbol{a}_0}{\|\boldsymbol{a}_0\|^2}\boldsymbol{s}^T - \frac{\boldsymbol{a}_0\boldsymbol{a}_0^T\boldsymbol{W}_0^1}{\|\boldsymbol{a}_0\|^2}. \tag{M.50}$$

M10.2 L=2

Since for $L = 1, 2, 3$, simulations suggest that $\Delta \boldsymbol{a}$ is neglible, we assume it to be zero for calculating solutions for $L = 2, 3$. Define the Lagrangian as

$$\mathcal{L} = \|\Delta \boldsymbol{W}^1\|^2 + \|\Delta \boldsymbol{W}^2\|^2 - \boldsymbol{\lambda}^T \left[\left(\Delta \boldsymbol{W}^1 + \boldsymbol{W}_0^1\right)^T \left(\Delta \boldsymbol{W}^2 + \boldsymbol{W}_0^2\right)^T \boldsymbol{a}_0 - \boldsymbol{s} \right]. \tag{M.51}$$

Extremizing yields

$$\Delta \boldsymbol{W}^1 = (\Delta \boldsymbol{W}^2 + \boldsymbol{W}_0^2)^T \boldsymbol{a}_0 \boldsymbol{\lambda}^T \tag{M.52}$$

$$\Delta \boldsymbol{W}^2 = \boldsymbol{a}_0 \boldsymbol{\lambda}^T (\Delta \boldsymbol{W}^1 + \boldsymbol{W}_0^1)^T \tag{M.53}$$

Solving for $\Delta \boldsymbol{W}^1, \Delta \boldsymbol{W}^2$ yields

$$\Delta \boldsymbol{W}^1 = (1 - \|\boldsymbol{a}_0\|^2 \|\boldsymbol{\lambda}\|^2)^{-1} \left[{\boldsymbol{W}_0^2}^T \boldsymbol{a}_0 + \|\boldsymbol{a}_0\|^2 \boldsymbol{W}_0^1 \boldsymbol{\lambda} \right] \boldsymbol{\lambda}^T \tag{M.54}$$

$$\Delta \boldsymbol{W}^2 = (1 - \|\boldsymbol{a}_0\|^2 \|\boldsymbol{\lambda}\|^2)^{-1} \boldsymbol{a}_0 \left[\|\boldsymbol{\lambda}\|^2 {\boldsymbol{W}_0^2}^T \boldsymbol{a}_0 + \boldsymbol{W}_0^1 \boldsymbol{\lambda} \right]^T. \tag{M.55}$$

Define scalar order parameters

$$u_1 = \left\| \frac{1}{1 - \|\boldsymbol{a}_0\|^2 v_1} \left[{\boldsymbol{W}_0^2}^T \boldsymbol{a}_0 + \|\boldsymbol{a}_0\| \boldsymbol{W}_0^1 \boldsymbol{\lambda} \right] \right\|^2 \tag{M.56}$$

$$v_1 = \|\boldsymbol{\lambda}\|^2. \tag{M.57}$$

Plugging expressions for $\Delta \boldsymbol{W}^1, \Delta \boldsymbol{W}^2$ into the constraint equation and solve for $\boldsymbol{\lambda}$ to get

$$\boldsymbol{\lambda} = \left[\|\boldsymbol{a}_0\|^2 {\boldsymbol{W}_0^1}^T \boldsymbol{W}_0^1 + (1 - \|\boldsymbol{a}_0\|^2 v_1) u_1 \mathbb{I} \right]^{-1} \left[(1 - \|\boldsymbol{a}_0\|^2 v_1) \boldsymbol{s} - (\boldsymbol{W}_0^2 \boldsymbol{W}_0^1)^T \boldsymbol{a}_0 \right]. \tag{M.58}$$

Plugging this expression back into Eqs.M.57 yields two self-consistent equations of u_1, v_1. Other variables in these equations are all stationary in time. Therefore, one can numerically solve for u_1, v_1 to obtain expressions for $\Delta \boldsymbol{W}^1, \Delta \boldsymbol{W}^2$.

M10.3 L=3

Setting up the Lagrangian and extremizing the variables to get

$$\Delta \boldsymbol{W}^1 = (\Delta \boldsymbol{W}^2 + \boldsymbol{W}_0^2)^T (\Delta \boldsymbol{W}^3 + \boldsymbol{W}_0^3)^T \boldsymbol{a}_0 \boldsymbol{\lambda}^T \tag{M.59}$$

$$\Delta \boldsymbol{W}^2 = (\Delta \boldsymbol{W}^3 + \boldsymbol{W}_0^3)^T \boldsymbol{a}_0 \boldsymbol{\lambda}^T (\Delta \boldsymbol{W}^1 + \boldsymbol{W}_0^1)^T \tag{M.60}$$

$$\Delta \boldsymbol{W}^3 = \boldsymbol{a}_0 \boldsymbol{\lambda}^T (\Delta \boldsymbol{W}^1 + \boldsymbol{W}_0^1)^T (\Delta \boldsymbol{W}^2 + \boldsymbol{W}_0^2)^T \tag{M.61}$$

Solving these equations for $\Delta \boldsymbol{W}^l$ gives ($u_{1,2}, v_{1,2}$ being order parameters defined below)

$$\Delta \boldsymbol{W}^1 = (1 - v_1 u_2)^{-1} \left[{\boldsymbol{W}_0^2}^T \boldsymbol{U} + u_2 \boldsymbol{W}_0^1 \boldsymbol{\lambda} \right] \boldsymbol{\lambda}^T \tag{M.62}$$

$$\Delta \boldsymbol{W}^2 = (1 - v_1 u_2)^{-1} \boldsymbol{U} \left[v_1 \boldsymbol{U}^T \boldsymbol{W}_0^2 + \boldsymbol{\lambda}^T {\boldsymbol{W}_0^1}^T \right] \tag{M.63}$$

$$\Delta \boldsymbol{W}^3 = v_2 \boldsymbol{a}_0 \boldsymbol{U}^T + (1 - v_1 u_2)^{-1} \boldsymbol{a}_0 \left[v_1 \boldsymbol{U}^T \boldsymbol{W}_0^2 + \boldsymbol{\lambda}^T {\boldsymbol{W}_0^1}^T \right] {\boldsymbol{W}_0^2}^T, \tag{M.64}$$

where

$$\boldsymbol{U} = \boldsymbol{Q}^{-1} \left[{\boldsymbol{W}_0^3}^T \boldsymbol{a}_0 + \|\boldsymbol{a}_0\|^2 (1 - v_1 u_2)^{-1} \boldsymbol{W}_0^2 \boldsymbol{W}_0^1 \boldsymbol{\lambda} \right] \tag{M.65}$$

$$\boldsymbol{\lambda} = \left[u_1 \mathbb{I} + (1 - v_1 u_2)^{-1} u_2 {\boldsymbol{W}_0^1}^T \boldsymbol{W}_0^1 + \|\boldsymbol{a}_0\|^2 (1 - v_1 u_2)^{-2} (\boldsymbol{W}_0^2 \boldsymbol{W}_0^1)^T \boldsymbol{Q}^{-1} \boldsymbol{W}_0^2 \boldsymbol{W}_0^1 \right]^{-1} \left[\boldsymbol{s} - (1 - v_1 u_2)^{-1} (\boldsymbol{W}_0^2 \boldsymbol{W}_0^1)^T \boldsymbol{Q}^{-1} (\boldsymbol{W}_0^3)^T \boldsymbol{a}_0 \right], \tag{M.66}$$

and

$$\boldsymbol{Q} = (1 - \|\boldsymbol{a}_0\|^2 v_2)\mathbb{I} - \|\boldsymbol{a}_0\|^2 (1 - v_1 u_2)^{-1} v_1 \boldsymbol{W}_0^{2^T} \boldsymbol{W}_0^2. \tag{M.67}$$

We have defined four scalar order parameters to be solved numerically.

$$u_1 = \|(1 - v_1 u_2)^{-1} [\boldsymbol{W}_0^{2^T} \boldsymbol{U} + u_2 \boldsymbol{W}_0^1 \boldsymbol{\lambda}]\|^2 \tag{M.68}$$

$$u_2 = \|\boldsymbol{U}\|^2 \tag{M.69}$$

$$v_1 = \|\boldsymbol{\lambda}\|^2 \tag{M.70}$$

$$v_2 = \|(1 - v_1 u_2)^{-1} [\boldsymbol{W}_0^1 \boldsymbol{\lambda} + v_1 \boldsymbol{W}_0^{2^T} \boldsymbol{U}]\|^2. \tag{M.71}$$

M10.4 Numerical Solvers of Self-Consistent Equations

In each case discussed above, we seek to solve k nonlinear equations of k scalar variables numerically. There are many algorithms for this purpose. In general, convergence to the true solution is not guaranteed and depends on initial estimates. To obtain good initial estimates, we used a two-step procedure to solve the equations for each set of network parameters.

In the first step, we used an iterative algorithm defined in Algorithm1.

Algorithm 1: Algorithm for solving self-consistent equations.

```
Initialize estimates for order parameters, α^0, β^0 ;
i = 0 ;
Initialize update factor γ ;
Initialize convergence threshold ϵ ;
while (α^{i+1} − α^i)^2 > ϵ or (β^{i+1} − β^i)^2 > ϵ do
    α^{i+1}, β^{i+1} = γ ∗ pseudo-self-consistent equations(α^i, β^i) + (1 − γ) ∗ (α^i, β^i);
    i = i + 1 ;
    if i > MaxIteration then
        break
    end
end
```

To aid convergence, we replaced all matrix inversions in the equations with pseudo-inverses (specifically, we only keep the 4 leading singular values and inverse them). These equations are referred to as "pseudo-self-consistent equations". We first used trial-and-error to find good initial estimates for a specific set of network parameters (e.g., $\sigma_s = 0.1, \sigma_w = 1$) and ran the algorithm until convergence. We then considered another pair of parameters that are close to the previous pair (e.g., $\sigma_s = 0.125, \sigma_w = 1$), using *final estimates* for $\sigma_s = 0.1, \sigma_w = 1$ as initial estimates for the new pair. We repeated this procedure recursively to cover all network parameter regimes of interest. After this step, we obtain solutions to the pseudo-self-consistent equations.

For the second step, we used `scipy.optimize.fsolve`, which implements a quasi-Newton method. For each set of network parameters, we used solutions to the pseudo-self-consistent equations as initial estimates for solutions to the true self-consistent equations. Upon convergence, we obtain solutions to the true self-consistent equations.

M11 Simulation Details

Gradient descent with minibatches was used for all weights in the network ($\boldsymbol{W}^l$ and $\boldsymbol{a}$), implemented with `pytorch`. The algorithm is terminated after the error estimated with $50,000$ examples is less or equal to the optimal level for more than 5 times. Default hyperparameters used in simulations are tabulated in

Variable	Value	Comments
Number of neurons per layer (N)	1000	All layers have the same number.
Learning rate (η)	0.001	
Input noise variance (σ^2)	0.01	
$\delta\theta$ (in rads)	≈ 0.0009	Adjusted so that signal-to-noise ratio in input is 1.
Minibatch size	50	

Table M10: Default Simulation Hyperparameters.

Table M10. We chose relatively small $\delta\theta$ and σ^2 so that the large N effects are already apparent with $N = 1000$, conserving computational resources. Python codes are available at `https://github.com/hzshan/perceptual_learning`.

M12 2D Gabor Equivalents of Stimuli and Filters in Our Model

In our model, the input vector has a 1D structure with width σ_s and the filter used by each hidden neuron is also 1D with width σ_w. Here, we describe the procedure to find their 2D equivalents. We consider 2D stimuli and 2D filters as given in [20]. The stimulus (for orientation θ_{stim}) is given by

$$G_s(i,j,\theta_{\text{stim}}) \propto \exp\left(-\frac{C_\perp^2}{2\sigma_{\perp,s}^2} - \frac{C_\parallel^2}{2\sigma_{\parallel,s}^2}\right)\cos(2\pi K_s C_\perp) \tag{M.72}$$

$$C_\perp = i\cos\theta_{\text{stim}} + j\sin\theta_{\text{stim}} \quad C_\parallel = j\cos\theta_{\text{stim}} - i\sin\theta_{\text{stim}}, \tag{M.73}$$

where (i,j) gives the coordinates in 2D, K_s is the spatial frequency, $\sigma_{\perp,s}^2$ controls the width perpendicular to the orientation θ_{stim} and $\sigma_{\parallel,s}^2$ controls the width parallel to the orientation. Analogously, the filter (for preferred stimulus θ_{pref}) is given by

$$G_f(i,j,\theta_{\text{pref}}) \propto \exp\left(-\frac{C_\perp^2}{2\sigma_{\perp,f}^2} - \frac{C_\parallel^2}{2\sigma_{\parallel,f}^2}\right)\cos(2\pi K_x C_\perp) \tag{M.74}$$

$$C_\perp = i\cos\theta_{\text{pref}} + j\sin\theta_{\text{pref}} \quad C_\parallel = j\cos\theta_{\text{pref}} - i\sin\theta_{\text{pref}}. \tag{M.75}$$

For both the stimulus and the filter, we used $i,j \in [-0.5, 0.5]$. Note that for each stimulus and each filter, one can generate an input tuning curve by fixing θ_{pref} and rotating the stimulus.

We first found a 2D Gabor filter equivalent to a value of σ_w, we fixed a stimulus with $C_{\perp,s} \ll 1$, $K_s = 0.75$, and $C_{\parallel,s} = 0.4$ and filter parameters $\sigma_{\parallel,f} = 0.5$. We then adjusted $\sigma_{\perp,f}$ while fixing $K = 0.3/\sigma_{\perp,f}$ until the input tuning curve had the same width as each row of the weight matrix with σ_w.

To find the 2D Gabor stimulus equivalent to a value of σ_s, we used a filter created as described above (which depends on σ_w). We then adjusted $\sigma_{\perp,s}$ until the input tuning curve had the same width as the input tuning curve with 1D stimuli/weights corresponding to this pair of σ_s, σ_w. bake

M13 Circulant Solutions to Perceptual Learning

For networks with one layer and a fixed $\boldsymbol{a}$, we seek a circulant $\boldsymbol{W}_{\text{circ}}^1$ such that ${\boldsymbol{W}_{\text{circ}}^1}^T\boldsymbol{a} = \boldsymbol{s}$. Note that all circulant matrices can be diagonalized as $\boldsymbol{W}_{\text{circ}}^1 = \boldsymbol{F}\boldsymbol{\Lambda}_W\boldsymbol{F}^T$, where eigenvectors in $\boldsymbol{F}$ are the Fourier bases. We then use

$$(\Lambda_W)_{ii} = \frac{(\boldsymbol{F}^T\boldsymbol{s})_i}{(\boldsymbol{F}^T\boldsymbol{a})_i + \epsilon_{\text{circ}}}, \tag{M.76}$$

where the regularizer ϵ_{circ} is chosen to be as large as possible without significantly increasing $\|{\boldsymbol{W}^1_{\text{circ}}}^T \boldsymbol{a} - \boldsymbol{s}\|$. We found that such $\boldsymbol{W}^1_{\text{circ}}$, computed numerically, leads to multi-modal tuning curves, which are unrealistic results (data not shown).

Supplementary Figures

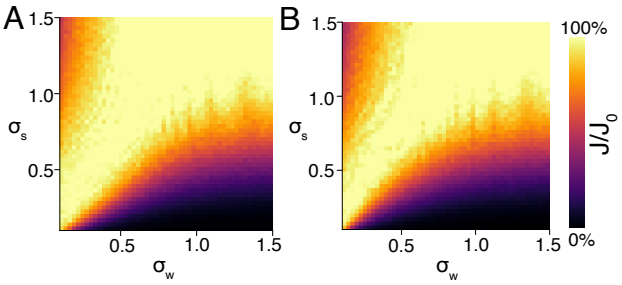

Figure S1: Information loss (J/J_0) in untrained $L = 2$ (A) and $L = 3$ (B) networks. The $L = 1$ case is shown in Fig.3A. The two panels share the color bar. In each case, there are two regimes corresponding to significant information loss. In all three cases, information loss is significant when the input is selective and the weights are unselective, or when the input is unselective and the weights are selective. Related to Fig.2.

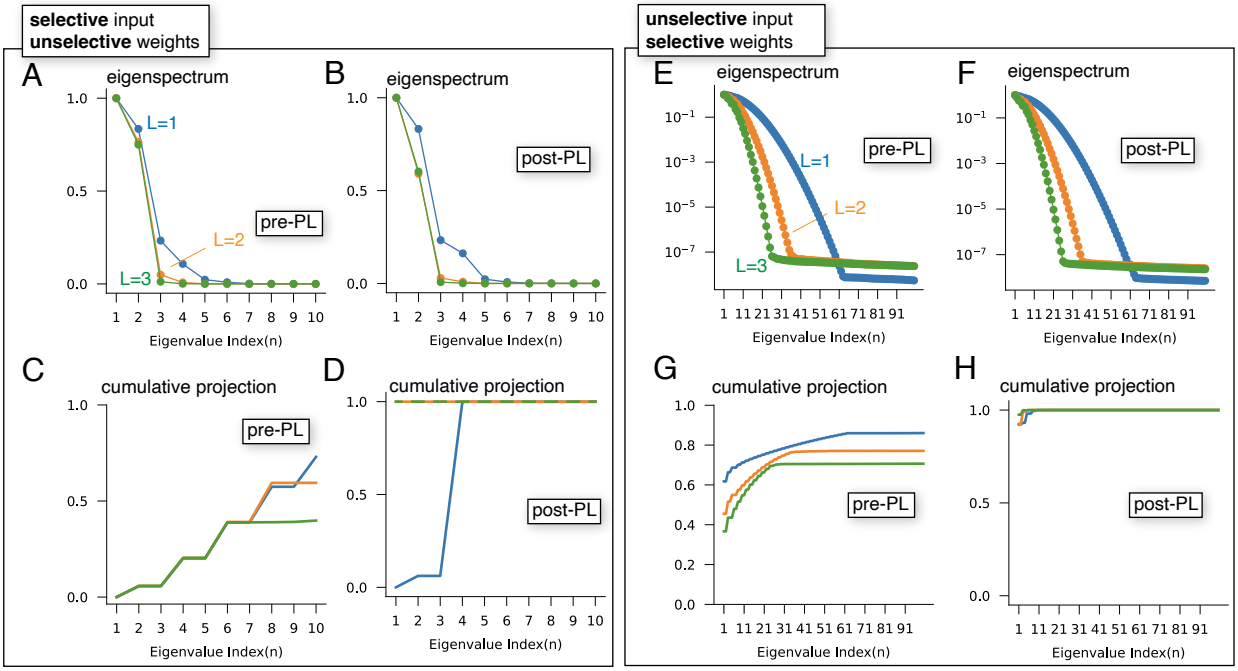

Figure S2: Low-rankness of weight matrices.
(A) The eigenspectrum of the product of effective weight matrices($\boldsymbol{P}^L$, see definition in Eq.M.9) before PL. Each curve corresponds to a network of a different depth. All networks are in the selective-input-unselective-weights regime ($\sigma_s = 0.2, \sigma_w = 0.8$). $\boldsymbol{P}^L$ is of lower rank for deeper networks.
(B) Same as (A), but for networks after PL. Rank of $\boldsymbol{P}^L$ for networks post-PL is approximately the same as that in pre-PL networks.
(C) Sum of squared projection of the signal vector ($\boldsymbol{s}$) onto the top n eigenvectors of $\boldsymbol{P}^L$.
(D) Same as (C), but for networks after PL.
(E)(F)(G)(H) Same as (A)(B)(C)(D), respectively, but for networks in the unselective-input-selective-weights regime ($\sigma_s = 1.2, \sigma_w = 0.1$). Related to Fig.2.

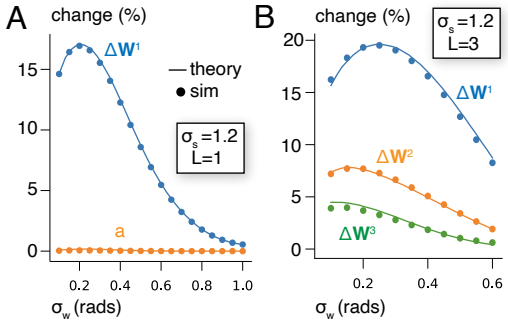

Figure S3: Comparing magnitude of MP learning changes ("theory") and changes induced by gradient descent ("sim") in the unselective-input-selective-weights regime. (A) L=1 network. (B) L=3 network. Learning rate is 10^{-3}. Related to Fig.4.

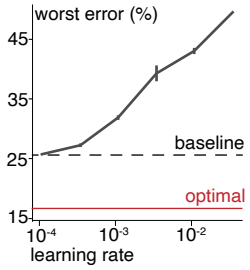

Figure S4: Worst error (over all test stimuli) after PL using different learning rates ($L = 1, \sigma_s = 0.2, \sigma_w = 0.8$). All learning rates tested here led to the same optimal performance on the trained task. Average of five runs (errorbars are standard errors). Baseline: pre-PL error. Optimal: MLD error. Related to Fig.4.

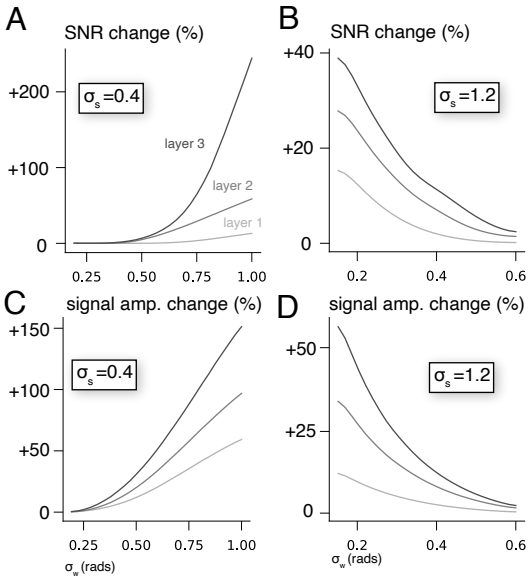

Figure S5: Magnitude of MP changes depends on selectivity parameters, σ_s, σ_w. Changes to the SNR and signal amplitude are bigger in higher layers. (A,B) Changes to the SNR in the last layer for different σ_w. (C, D) Changes to the signal amplitude in the last layer for different σ_w. $L = 3$ pre-PL networks in all panels. Related to Fig.5.

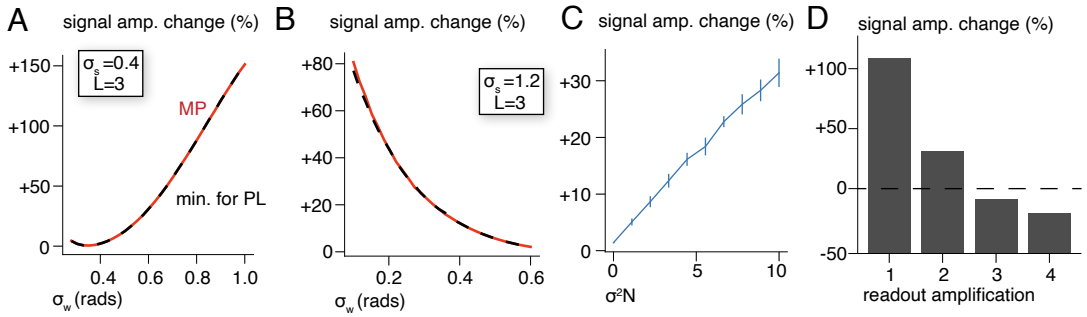

Figure S6: (A,B) comparison between the MP-induced signal amplification and the minimal signal amplification required for PL. (A) shows results in the selective-input-unselective-weights regime; (B) shows results in teh unselective-input-selective-weights regime. (C) Average signal amplification induced by "soft" MP learning, where weights are allowed to fluctuate around the MP weights. σ^2: magnitude of fluctuation. Errorbars show standard deviations over 10 independent samples. Results are taken from a 1-layer network with $\sigma_s = 0.4, \sigma_w = 1.0$. (D) Signal amplification induced by MP learning if we amplify the pre-PL readout weights. Results are taken from the last layer in a network with $\sigma_s = 0.4, \sigma_w = 1.0, L = 2$. Related to Fig.6.

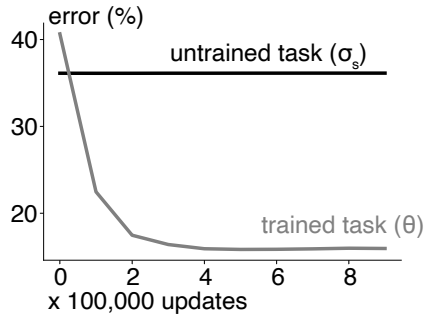

Figure S7: Cross-task transfer. Training a three-layer network on the θ discrimination task (gray) does not affect performance on a σ_s discrimination task (black). At every checkpoint, a separate task-specific $\boldsymbol{a}$ is used for the σ_s discrimination task. As a result, that the performance on the untrained task does not improve strictly reflects that the information content for σ_s discrimination does not increase. See Sec.S7 for details of the σ_s discrimination task. Related to Fig.7.

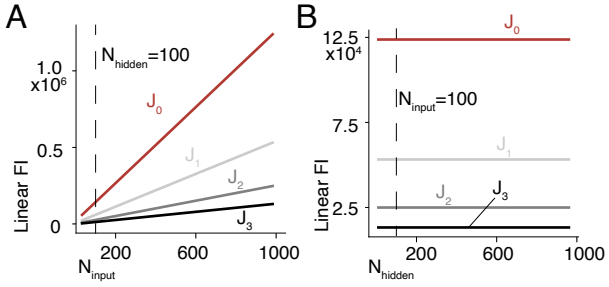

Figure S8: Scaling of Fisher information in the pre-PL network ($L = 3$, $\sigma_s = 0.2, \sigma_w = 0.8$). (A) Scaling of FI over different N_{input} and fixed N_{hidden}. (B) Scaling of FI over different N_{hidden} and fixed N_{input}. J_0:FI in the input; $J_{1,2,3}$: FI in layer 1,2,3. Related to Fig.8.

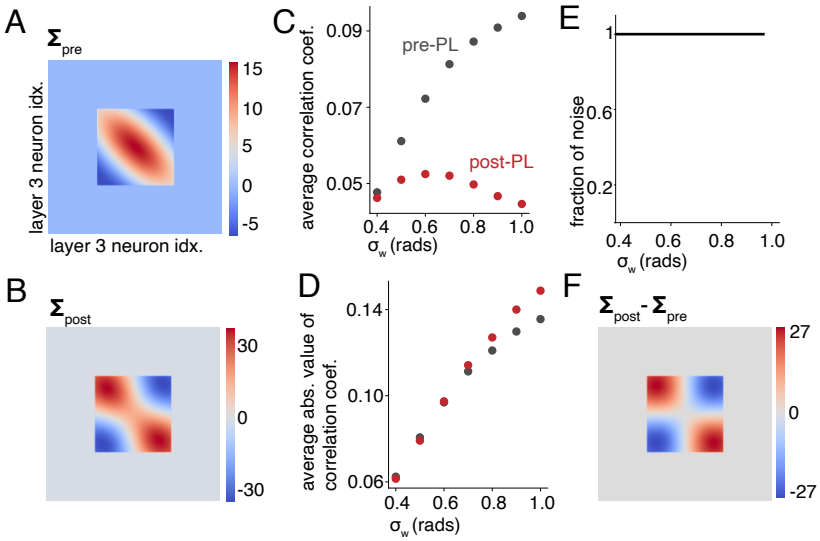

Figure S9: Structure of noise correlation in layer 3 of a $L = 3$ network when θ_{tr} is presented. (A) Normalized* pre-PL noise covariance matrix. (B) Post-PL covariance. (C) Averaged pair-wise Pearson's correlation coefficients before (red) and after PL (black), for different σ_w. (D) Same as C, averaged absolute values of coefficients are shown. (E) Fraction of PL-induced changes to covariance ($\mathbf{\Sigma}_{\text{post}} - \mathbf{\Sigma}_{\text{pre}}$) that project on the signal direction. (F) Visualization of PL-induced changes to covariance.
*Covariance matrices are multiplied by $N/\sigma^2\|\boldsymbol{f}^0\|^2\|\boldsymbol{f}^3\|^{-2}$, where $\boldsymbol{f}^0, \boldsymbol{f}^3$ are noise-averaged population response vectors in the input layer and layer 3, respectively. This scaling makes each element of the matrix $O(1)$ and adjusts for different activity levels in different layers. Related to Fig.8.

References

[1] Aniek Schoups, Rufin Vogels, Ning Qian, and Guy Orban. Practising orientation identification improves orientation coding in v1 neurons. *Nature*, 412(6846):549–553, 2001.

[2] Geoffrey M Ghose, Tianming Yang, and John HR Maunsell. Physiological correlates of perceptual learning in monkey v1 and v2. *Journal of neurophysiology*, 87(4):1867–1888, 2002.

[3] Tianming Yang and John HR Maunsell. The effect of perceptual learning on neuronal responses in monkey visual area v4. *Journal of Neuroscience*, 24(7):1617–1626, 2004.

[4] Steven Raiguel, Rufin Vogels, Santosh G Mysore, and Guy A Orban. Learning to see the difference specifically alters the most informative v4 neurons. *Journal of Neuroscience*, 26(24):6589–6602, 2006.

[5] Yong Gu, Sheng Liu, Christopher R Fetsch, Yun Yang, Sam Fok, Adhira Sunkara, Gregory C DeAngelis, and Dora E Angelaki. Perceptual learning reduces interneuronal correlations in macaque visual cortex. *Neuron*, 71(4):750–761, 2011.

[6] Yin Yan, Malte J Rasch, Minggui Chen, Xiaoping Xiang, Min Huang, Si Wu, and Wu Li. Perceptual training continuously refines neuronal population codes in primary visual cortex. *Nature neuroscience*, 17(10):1380–1387, 2014.

[7] Mehdi Sanayei, Xing Chen, Daniel Chicharro, Claudia Distler, Stefano Panzeri, and Alexander Thiele. Perceptual learning of fine contrast discrimination changes neuronal tuning and population coding in macaque v4. *Nature communications*, 9(1):1–15, 2018.

[8] Amy M Ni, Douglas A Ruff, Joshua J Alberts, Jen Symmonds, and Marlene R Cohen. Learning and attention reveal a general relationship between population activity and behavior. *Science*, 359(6374):463–465, 2018.

[9] Avi Karni and Dov Sagi. Where practice makes perfect in texture discrimination: evidence for primary visual cortex plasticity. *Proceedings of the National Academy of Sciences*, 88(11):4966–4970, 1991.

[10] Adriana Fiorentini and Nicoletta Berardi. Perceptual learning specific for orientation and spatial frequency. *Nature*, 287(5777):43–44, 1980.

[11] Chi-Tat Law and Joshua I Gold. Neural correlates of perceptual learning in a sensory-motor, but not a sensory, cortical area. *Nature neuroscience*, 11(4):505–513, 2008.

[12] Hamed Zivari Adab, Ivo D Popivanov, Wim Vanduffel, and Rufin Vogels. Perceptual learning of simple stimuli modifies stimulus representations in posterior inferior temporal cortex. *Journal of cognitive neuroscience*, 26(10):2187–2200, 2014.

[13] HP Op de Beeck, Johan Wagemans, and Rufin Vogels. Effects of perceptual learning in visual backward masking on the responses of macaque inferior temporal neurons. *Neuroscience*, 145(2):775–789, 2007.

[14] Merav Ahissar and Shaul Hochstein. Attentional control of early perceptual learning. *Proceedings of the National Academy of Sciences*, 90(12):5718–5722, 1993.

[15] Roy E Crist, Mitesh K Kapadia, Gerald Westheimer, and Charles D Gilbert. Perceptual learning of spatial localization: specificity for orientation, position, and context. *Journal of neurophysiology*, 78(6):2889–2894, 1997.

[16] Manfred Fahle. Specificity of learning curvature, orientation, and vernier discriminations. *Vision research*, 37(14):1885–1895, 1997.

[17] Syed A Chowdhury and Gregory C DeAngelis. Fine discrimination training alters the causal contribution of macaque area mt to depth perception. *Neuron*, 60(2):367–377, 2008.

[18] Merav Ahissar and Shaul Hochstein. The reverse hierarchy theory of visual perceptual learning. *Trends in cognitive sciences*, 8(10):457–464, 2004.

[19] Andrew Michael Saxe. *Deep linear neural networks: A theory of learning in the brain and mind*. Stanford University, 2015.

[20] Vikranth R Bejjanki, Jeffrey M Beck, Zhong-Lin Lu, and Alexandre Pouget. Perceptual learning as improved probabilistic inference in early sensory areas. *Nature neuroscience*, 14(5):642–648, 2011.

[21] H Sebastian Seung and Haim Sompolinsky. Simple models for reading neuronal population codes. *Proceedings of the national academy of sciences*, 90(22):10749–10753, 1993.

[22] Alexander A Petrov, Barbara Anne Dosher, and Zhong-Lin Lu. The dynamics of perceptual learning: an incremental reweighting model. *Psychological review*, 112(4):715, 2005.

[23] Barbara Anne Dosher, Pamela Jeter, Jiajuan Liu, and Zhong-Lin Lu. An integrated reweighting theory of perceptual learning. *Proceedings of the National Academy of Sciences*, 110(33):13678–13683, 2013.

[24] Li K Wenliang and Aaron R Seitz. Deep neural networks for modeling visual perceptual learning. *Journal of Neuroscience*, 38(27):6028–6044, 2018.

[25] Rachel Lee and Andrew Saxe. Modeling perceptual learning with deep networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36, 2014.

[26] Gad Cohen and Daphna Weinshall. Hidden layers in perceptual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4554–4562, 2017.

[27] Peggy Seriès, Peter E Latham, and Alexandre Pouget. Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nature neuroscience*, 7(10):1129–1135, 2004.

[28] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[29] Jeffrey Beck, Vikranth R Bejjanki, and Alexandre Pouget. Insights from a simple expression for linear fisher information in a recurrently connected population of spiking neurons. *Neural computation*, 23(6):1484–1502, 2011.

[30] Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, pages 1–13, 2020.

[31] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[32] Rubén Moreno-Bote, Jeffrey Beck, Ingmar Kanitscheider, Xaq Pitkow, Peter Latham, and Alexandre Pouget. Information-limiting correlations. *Nature neuroscience*, 17(10):1410–1417, 2014.

[33] Ingmar Kanitscheider, Ruben Coen-Cagli, Adam Kohn, and Alexandre Pouget. Measuring fisher information accurately in correlated neural populations. *PLoS Comput Biol*, 11(6):e1004218, 2015.

[34] Douglas A Ruff and Marlene R Cohen. Simultaneous multi-area recordings suggest that attention improves performance by reshaping stimulus representations. *Nature neuroscience*, pages 1–8, 2019.

[35] Barbara Dosher and Zhong-Lin Lu. Visual perceptual learning and models. *Annual Review of Vision Science*, 3:343–363, 2017.

[36] Takeo Watanabe and Yuka Sasaki. Perceptual learning: toward a comprehensive theory. *Annual review of psychology*, 66:197–221, 2015.

[37] Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

[38] Wickliffe C Abraham and Anthony Robins. Memory retention–the synaptic stability versus plasticity dilemma. *Trends in neurosciences*, 28(2):73–78, 2005.

[39] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[40] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 70:3987, 2017.

[41] Ehud Zohary, Simona Celebrini, Kenneth H Britten, and William T Newsome. Neuronal plasticity that underlies improvement in perceptual performance. *Science*, 263(5151):1289–1292, 1994.

[42] Hamed Zivari Adab and Rufin Vogels. Practicing coarse orientation discrimination improves orientation signals in macaque cortical area v4. *Current biology*, 21(19):1661–1666, 2011.

[43] Jasper Poort, Adil G Khan, Marius Pachitariu, Abdellatif Nemri, Ivana Orsolic, Julija Krupic, Marius Bauza, Maneesh Sahani, Georg B Keller, Thomas D Mrsic-Flogel, et al. Learning enhances sensory and multiple non-sensory representations in primary visual cortex. *Neuron*, 86(6):1478–1490, 2015.

[44] Lucia M Vaina, John W Belliveau, Eric B Des Roziers, and Thomas A Zeffiro. Neural systems underlying learning and representation of global motion. *Proceedings of the National Academy of Sciences*, 95(21):12657–12662, 1998.

[45] Mariano Sigman, Hong Pan, Yihong Yang, Emily Stern, David Silbersweig, and Charles D Gilbert. Top-down reorganization of activity in the visual pathway after learning a shape identification task. *Neuron*, 46(5):823–835, 2005.

[46] Janneke FM Jehee, Sam Ling, Jascha D Swisher, Ruben S van Bergen, and Frank Tong. Perceptual learning selectively refines orientation representations in early visual cortex. *Journal of Neuroscience*, 32(47):16747–16753, 2012.

[47] Yan Wang, Wei Wu, Xian Zhang, Xu Hu, Yue Li, Shihao Lou, Xiao Ma, Xu An, Hui Liu, Jing Peng, et al. A mouse model of visual perceptual learning reveals alterations in neuronal coding and dendritic spine density in the visual cortex. *Frontiers in behavioral neuroscience*, 10:42, 2016.

[48] Roy E Crist, Wu Li, and Charles D Gilbert. Learning to see: experience and attention in primary visual cortex. *Nature neuroscience*, 4(5):519–525, 2001.

[49] Gregg H Recanzone, Michael M Merzenich, William M Jenkins, Kamil A Grajski, and HUBERT R Dinse. Topographic reorganization of the hand representation in cortical area 3b owl monkeys trained in a frequency-discrimination task. *journal of Neurophysiology*, 67(5):1031–1056, 1992.

[50] Gregg H Recanzone, Christoph E Schreiner, and Michael M Merzenich. Plasticity in the frequency representation of primary auditory cortex following discrimination training in adult owl monkeys. *Journal of Neuroscience*, 13(1):87–103, 1993.

[51] Burkhard Pleger, Ann-Freya Foerster, Patrick Ragert, Hubert R Dinse, Peter Schwenkreis, Jean-Pierre Malin, Volkmar Nicolas, and Martin Tegenthoff. Functional imaging of perceptual learning in human primary and secondary somatosensory cortex. *Neuron*, 40(3):643–653, 2003.

[52] Sophie Schwartz, Pierre Maquet, and Chris Frith. Neural correlates of perceptual learning: a functional mri study of visual texture discrimination. *Proceedings of the National Academy of Sciences*, 99(26):17137–17142, 2002.

[53] Christopher S Furmanski, Denis Schluppeck, and Stephen A Engel. Learning strengthens the response of primary visual cortex to simple patterns. *Current Biology*, 14(7):573–578, 2004.

[54] Christine Schiltz, JM Bodart, S Dubois, S Dejardin, C Michel, A Roucoux, M Crommelinck, and GA Orban. Neuronal mechanisms of perceptual learning: changes in human brain activity with training in orientation discrimination. *Neuroimage*, 9(1):46–62, 1999.

[55] Aniek A Schoups, Rufin Vogels, and Guy A Orban. Human perceptual learning in identifying the oblique orientation: retinotopy, orientation specificity and monocularity. *The Journal of physiology*, 483(3):797–810, 1995.

[56] Manfred Fahle and Shimon Edelman. Long-term learning in vernier acuity: Effects of stimulus orientation, range and of feedback. *Vision research*, 33(3):397–412, 1993.

[57] Ling-Po Shiu and Harold Pashler. Improvement in line orientation discrimination is retinally local but dependent on cognitive set. *Perception & psychophysics*, 52(5):582–588, 1992.

[58] Pamela E Jeter, Barbara Anne Dosher, Alexander Petrov, and Zhong-Lin Lu. Task precision at transfer determines specificity of perceptual learning. *Journal of vision*, 9(3):1–1, 2009.

[59] Andrew YY Tan, Brandon D Brown, Benjamin Scholl, Deepankar Mohanty, and Nicholas J Priebe. Orientation selectivity of synaptic input to neurons in mouse and cat primary visual cortex. *Journal of Neuroscience*, 31(34):12339–12350, 2011.

[60] David McLaughlin, Robert Shapley, Michael Shelley, and Dingeman J Wielaard. A neuronal network model of macaque primary visual cortex (v1): Orientation selectivity and dynamics in the input layer 4cα. *Proceedings of the National Academy of Sciences*, 97(14):8087–8092, 2000.

[61] Jeffrey S Anderson, Matteo Carandini, and David Ferster. Orientation tuning of input conductance, excitation, and inhibition in cat primary visual cortex. *Journal of neurophysiology*, 84(2):909–926, 2000.

[62] Takeo Watanabe, Jose E Nanez, and Yuka Sasaki. Perceptual learning without perception. *Nature*, 413(6858):844–848, 2001.

[63] Aaron R Seitz and Takeo Watanabe. The phenomenon of task-irrelevant perceptual learning. *Vision research*, 49(21):2604–2610, 2009.

[64] Barbara Anne Dosher and Zhong-Lin Lu. Perceptual learning reflects external noise filtering and internal noise reduction through channel reweighting. *Proceedings of the National Academy of Sciences*, 95(23):13988–13993, 1998.