

# Evaluating the extent to which homeostatic plasticity learns to compute prediction errors in unstructured neuronal networks

Vicky Zhu and Robert Rosenbaum\*

February 21, 2022

## Abstract

The brain is believed to operate in part by making predictions about sensory stimuli and encoding deviations from these predictions in the activity of “prediction error neurons.” This principle defines the widely influential theory of predictive coding. The precise circuitry and plasticity mechanisms through which animals learn to compute and update their predictions are unknown. Homeostatic inhibitory synaptic plasticity is a promising mechanism for training neuronal networks to perform predictive coding. Homeostatic plasticity causes neurons to maintain a steady, baseline firing rate in response to inputs that closely match the inputs on which a network was trained, but firing rates can deviate away from this baseline in response to stimuli that are mismatched from training. We combine computer simulations and mathematical analysis systematically to test the extent to which randomly connected, unstructured networks compute prediction errors after training with homeostatic inhibitory synaptic plasticity. We find that homeostatic plasticity alone is sufficient for computing prediction errors for trivial time-constant stimuli, but not for more realistic time-varying stimuli. We use a mean-field theory of plastic networks to explain our findings and characterize the assumptions under which they apply.

## 1 Introduction

Cortical neuronal networks can make predictions about sensory stimuli and detect errors about these predictions. For example, in the visuomotor system, head movements produce predictable flows of an animal’s visual scene. Visual cortical circuits learn predictable associations between bottom-up input from the visual stream and top-down input from the motor system. Violations of the learned predictions, known as “mismatched stimuli” or “prediction errors”, produce distinct responses in visual cortical neurons, which can help the animal distinguish between self-driven and externally driven movements of its visual scene [1, 2, 3].

The idea that the brain uses predictions and prediction errors to encode and interpret sensory information dates back to 19th century work by Helmholtz [4, 5] and underlies more general theories of neural function such as predictive coding, predictive processing, active inference, and the free energy principle [6, 7, 8, 5]. The question of how neural circuits compute prediction errors and how they learn predictions through biologically plausible synaptic plasticity rules is not settled, but some theories have been put forward [9, 10, 11, 12, 13, 14, 15].

Cortical neurons are highly interconnected, even within a single cortical area and layer. This dense, recurrent, and intralaminar connectivity shapes the intrinsic dynamics and stimulus responses of local cortical circuits. The nonlinear firing rate dynamics that arise from this recurrent connectivity can interact with the slower dynamics of synaptic plasticity in complex ways. Homeostatic inhibitory synaptic plasticity is a widely observed and widely studied type of synaptic plasticity [16, 17, 18, 19, 20, 15, 21] in which the strength of inhibitory synapses are adjusted in an

---

\*This work was supported by the Air Force Office of Scientific Research (AFOSR) award number FA9550-21-1-0223 and NSF awards DMS-1517828 and DMS-1654268.

activity-dependent manner that tends to push the postsynaptic neurons' firing rates toward a homeostatic baseline targets. Simulations and theoretical analyses of mathematical models of homeostatic inhibitory plasticity show that, while firing rates are near their targets in response to stimuli on which the network has been trained, firing rates deviate from their targets in response to unfamiliar stimuli in these models [17, 22, 14, 23, 15, 24].

As in related computational work [14, 23, 15], we conjectured that homeostatic inhibitory plasticity could learn to perform some type of predictive coding. In particular, if the external input to a neural population were formed from bottom-up and top-down stimuli, then homeostatic plasticity in the network would naturally learn to produce baseline activity in response to "matched" top-down and bottom-up pairings (*i.e.*, pairings that are similar to those on which the network was trained). On the other hand, "mismatched" pairings (*i.e.*, pairings from outside the training distribution) would produce firing rate responses that are further from the homeostatic baseline. In this sense, the network should learn to encode prediction errors (*i.e.*, errors in the ability to predict top-down input from bottom-up input or vice versa) in the deviation of the firing rates from their baseline. Importantly, and in contrast to previous work [14, 23, 15], we conjectured that the network should not need to be imparted with any special structure or architecture to learn this computation since homeostatic plasticity should naturally achieve this result due to its tendency to produce baseline responses to stimuli on which the network was trained, but not in response to novel stimuli.

To test our conjecture, we used an unstructured, recurrent, spiking neuronal network model endowed with a homeostatic inhibitory plasticity rule receiving two sources of external input, modeling top-down and bottom-up stimuli. We trained the network with given patterns of top-down and bottom-up pairings, interpreted as "matched" stimuli, before presenting a "mismatched" stimulus that deviated from the pairings used during training. Numerical simulations showed that the network reliably produced baseline firing rates for a fixed pair of bottom-up and top-down inputs during training, and deviated from baseline in response to a mismatched stimulus. A mean-field, firing rate model and a mathematical analysis using a separation of timescales helped reveal the dynamics underlying these numerical simulations. Hence, homeostatic plasticity learned to compute prediction errors whenever top-down and bottom-up stimuli are fixed during training. However, useful predictive coding algorithms should learn to detect relationships between time-varying top-down and bottom-up inputs. We generalized our input model to vary the intensity of top-down and bottom-up inputs in unison. An effective learning algorithm should learn to detect a prediction error whenever the intensity changes out of unison. To our surprise, our spiking network with homeostatic synaptic plasticity was unable to learn to detect this type of prediction error, even in a relatively simple (time-varying) setting. Going back to our mean-field analysis helped to clarify how and why the model failed to perform predictive coding in this setting after succeeding in the simpler (time-constant) setting.

We conclude that homeostatic inhibitory synaptic plasticity alone is not sufficient to learn and perform non-trivial predictive coding in unstructured neuronal network models. Previous theoretical work shows that network models that carefully account for the connectivity structure of multiple inhibitory subtypes are able to learn prediction errors using homeostatic plasticity, even for inputs where top-down and bottom-up input co-vary in intensity [14, 23]. Hence, the failure of our model in this scenario implies that network structure is critical for successfully learning predictive coding tasks with homeostatic plasticity.

## 2 Results

### 2.1 Spiking network model description

We consider a computational model of a local cortical circuit composed of  $N = 5000$  randomly connected exponential integrate-and-fire (EIF) spiking neuron models ( $N_e = 4000$  of which are excitatory and  $N_i = 1000$  inhibitory) [25, 26]. The membrane potentials of neuron  $j$  in population  $a = e, i$  obeys

$$\tau_m \frac{dV_j^a}{dt} = -(V_j^a - E_L) + D_T e^{(V_j^a - V_T)/D_T} + I_j^a(t) \quad (1)$$

with the added condition that each time  $V_k(t)$  crosses a threshold at  $V_{th}$ , it is reset to  $V_{re}$  and a spike is recorded. The synaptic input to neuron  $j$  in population  $a$  is modeled by

$$I_j^a(t) = X_j^a(t) + \sum_{b=e,i} \sum_{k=1}^N J_{jk}^{ab} \alpha_b(t - t_{n,k}^b)$$

where  $X_j^a(t)$  models external synaptic input,  $J_{jk}^{ab}$  is a synaptic weight,  $t_{n,k}^b$  is the time of the  $n$ th spike of neuron  $k$  in population  $b$ , and  $\alpha_b(t) = (1/\tau_b)e^{-t/\tau_b}H(t)$  is a synaptic filter with  $H(t)$  the Heaviside step function.

Initial connectivity in the model is random (connection probability  $p = 0.1$ ) with initial weights,  $J_{jk}^{ab}$ , determined only by pre- and post-synaptic neuron type ( $J_{jk}^{ab} = j_{ab}$  for connected neurons). Excitatory connectivity,  $J_{jk}^{ae}$ , remained fixed, but inhibitory connectivity evolves according to a homeostatic, inhibitory spike-timing-dependent plasticity (iSTDP) rule [17, 19, 20, 24]. Specifically, each time that neuron  $j$  in population  $a = e, i$  spikes (which occurs at times  $t_{n,j}^a$ ), the inhibitory synaptic weights targeting that neuron are updated according to

$$J_{jk}^{ai} = J_{jk}^{ai} - \eta_a x_k^i(t_{j,n}^a)$$

where  $\eta_a$  is a learning rate and recall that  $t_{j,n}^a$  is the time of the  $n$ th spike of neuron  $j$  in population  $a = e, i$ . Additionally, each time inhibitory neuron  $k$  spikes, its outgoing synaptic weights are updated according to

$$J_{jk}^{ai} = J_{jk}^{ai} - \eta_a (x_j^a(t_{k,n}^i) - 2r_0^a)$$

where  $t_{k,n}^i$  is the time of the  $n$ th spike of inhibitory neuron  $k$ . The time series,  $x_j^a(t)$  are defined by the differential equation

$$\tau_{STDP} \frac{dx_j^a}{dt} = -x_j^a$$

in addition to the rule that  $x_j^a(t)$  is incremented each time that neuron  $j$  in population  $a = e, i$  spikes according to,

$$dx_j^a(t_{j,n}^a) \leftarrow dx_j^a(t_{j,n}^a) + \frac{1}{\tau_{STDP}}. \quad (2)$$

As a result,  $x_j^a(t)$  estimates the firing rate of neuron  $j$  in population  $a$  by performing an exponentially-weighted sliding average of the spike density. This plasticity rule tends to push excitatory and inhibitory firing rates toward their target rates,  $r_0^e$  and  $r_0^i$ , respectively (see [17, 19, 20, 22, 24] and the mean-field theory presented below).

We are interested in understanding the extent to which such networks can learn to perform predictive coding [6, 12, 5]. More specifically, we reasoned that neurons would spike close to their target

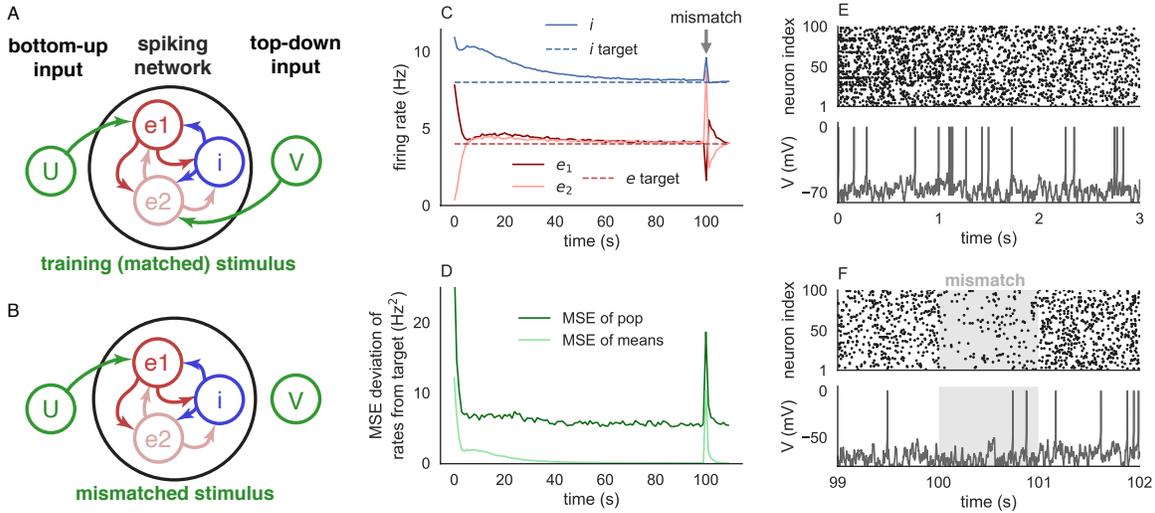


Figure 1: **Prediction errors after training on time-constant inputs to multiple sub-populations.** **A,B)** Network diagram with “training” and “mismatch” stimuli respectively. A randomly connected, recurrent spiking neural network of  $N = 5000$  neurons consisted of two excitatory sub-populations ( $e_1$  and  $e_2$ ) and one inhibitory ( $i$ ) population. During the first 100s of the simulation, the network received a “training” stimulus in which  $e_1$  and  $e_2$  received extra external input modeling bottom-up and top-down stimuli respectively (A). Then a “mismatch” stimulus was introduced for 1s by removing the top-down stimulus to population  $e_2$ . **C)** Homeostatic inhibitory synaptic plasticity caused population-averaged firing rates to converge to their targets during training, but they deviated from their targets in response to the mismatch stimulus. **D)** The deviation of the mean firing rates from their targets ( $MSE_{mean}$ ) and the mean deviation of individual neurons’ firing rates ( $MSE_{pop}$ ) quantify the deviation of firing rates from their targets. **E,F)** Raster plots (top) and membrane potential (bottom) of a random subset of neurons from population  $e_1$ .

rates in response to stimulus patterns similar to those on which they were trained, but deviate from the target rates in response to stimuli that deviate from the from the training stimuli. In other words, the deviation of firing rates from their targets should encode a “prediction error,” *i.e.*, a deviation of the inputs from the patterns that appeared during training.

## 2.2 Prediction errors after training on time-constant inputs to multiple sub-populations

For illustrative purposes, we first considered a simple input model for which the excitatory population was divided into two sub-populations,  $e_1$  and  $e_2$ , with  $N_{e_1} = N_{e_2} = 2000$  neurons in each sub-population (Figure 1A,B). Recurrent connectivity did not depend on sub-population membership, so the network was completely unstructured. During training, each neuron in populations  $e_1$  and  $e_2$  received external stimuli of the form (Figure 1A)

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + U \\ X_{e_2} &= X_e^0 + V \end{aligned} \right\} \text{matched} \quad (3)$$

where  $X_e^0$  is a baseline input that assures neurons spike at reasonable rates,  $U$  is a perturbation modeling bottom-up input, and  $V$  is a perturbation modeling top-down input. We used positive bottom-up

input and negative top-down input,

$$\begin{aligned} U &= X_e^0/5 \\ V &= -X_e^0/5, \end{aligned} \tag{4}$$

but our results are not sensitive to this specific choice of inputs. We refer to this as a “matched” stimulus because it defines the matching of bottom-up with top-down stimuli that the network is trained on. After training on matched stimuli, we modeled mismatched stimuli by the absence of top-down input (Figure 1B),

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + U \\ X_{e_2} &= X_e^0 \end{aligned} \right\} \text{mismatched.} \tag{5}$$

We refer to these stimuli as “mismatched” because the top-down and bottom-up inputs are mismatched when compared to the “matched” pairings used to train the network. Mismatched stimuli could also be modeled by an absence of bottom-up input, or any other deviation from the inputs used for training.

We hypothesized that, after training on matched stimuli, the network would produce firing rates close to the target rates in response to matched stimuli and produce firing rates further from the target rates in response to mismatched stimuli.

At the beginning of the simulation mean excitatory and inhibitory firing rates deviated from their targets, but inhibitory plasticity pushed them toward their targets over the course of tens of seconds (Figure 1C). After 100s of training on matched stimuli, we tested a mismatched stimulus for 1s. Consistent with our hypothesis, mean firing rates of each population were further from their targets in response to the mismatched stimulus (Figure 1C).

We quantified the distance of the firing rates from their targets from spiking network simulations using two methods. For the first method, we computed the MSE of the population-averaged firing rates (Figure 1D, light green),

$$MSE_{mean} = \sum_{a=e_1, e_2, i} q_a (r_a - r_a^0)^2$$

where  $r_a^0$  is the target rate and  $r_a$  the mean firing rate of each population averaged over neurons in that population and averaged over time windows of size  $T = 1$ s. The coefficients  $q_a = N_a/N$  represent the proportion of the network contained in each population ( $q_{e_1} = q_{e_2} = 0.4$  and  $q_i = 0.2$  for our network). Hence,  $MSE_{mean}$  weights the errors of larger sub-populations more heavily.

The  $MSE_{mean}$  measures how far the population-average rates differ from their target rates, but does not measure the deviation of individual neurons’ firing rates. Despite the fact that external input was constant across time and the simulations were deterministic (with the exception of “quenched” randomness from the random connectivity), neurons exhibited substantial variability in their spike timing and membrane potential dynamics (Figure 1E,F). These dynamics are characteristic of an asynchronous-irregular state [27, 28, 29, 30, 31, 32, 33].

To account for the deviation of individual neurons’ firing rates from spike-timing variability in spiking network simulations, we also computed the MSE across the entire network (Figure 1D, dark green),

$$MSE_{pop} = \frac{1}{N} \sum_{j=1}^N (r_j - r_j^0)^2$$

where  $r_j$  is the firing rate of neuron  $j = 1, \dots, N$  and  $r_j^0$  is its target rate. Both measures of MSE show a decrease during training and a sharp increase in response to the mismatched stimulus, but  $MSE_{pop}$  is larger overall due to the spike-timing variability of each neuron.

The results from the spiking network can be understood using a simpler dynamical mean-field model in which mean firing rates of each population are approximated by a system of differential equations,

$$\tau \odot \frac{d\mathbf{r}}{dt} = -\mathbf{r} + f(W\mathbf{r} + \mathbf{X}) \quad (6)$$

where  $\tau = [\tau_{e_1} \ \tau_{e_2} \ \tau_i]^T$  is a vector of time constants,  $\odot$  represents element-wise multiplication, and  $\mathbf{r} = [r_{e_1} \ r_{e_2} \ r_i]^T$  is a vector approximating the mean firing rates of the two excitatory sub-populations and the inhibitory population. Mean external input to each population is given by the vector

$$\mathbf{X} = \begin{bmatrix} X_{e_1} \\ X_{e_2} \\ X_i \end{bmatrix}$$

and the recurrent connectivity matrix is defined by

$$W = \begin{bmatrix} w_{e_1 e_1} & w_{e_1 e_2} & w_{e_1 i} \\ w_{e_2 e_1} & w_{e_2 e_2} & w_{e_2 i} \\ w_{i e_1} & w_{i e_2} & w_{i i} \end{bmatrix}$$

where [34, 35, 36, 37, 22, 24]

$$w_{ab} = N_b p_{ab} j_{ab}$$

Here,  $N_b$  is the number of neurons in population  $b = e_1, e_2, i$  (so  $N_{e_1} = N_{e_2} = N_e/2 = 2000$  and  $N_i = 1000$ ),  $p_{ab}$  is the connection probability from population  $b$  to population  $a$ , and  $j_{ab}$  is the mean non-zero synaptic weight (mean of  $J_{ab}^{jk}$  between connected neurons). The inhibitory entries,  $w_{ai}$  for  $a = e_1, e_2, i$ , are negative and evolve according to

$$\frac{dw_{ai}}{dt} = -\eta_a (r_a - r_a^0) r_i \quad (7)$$

where  $\eta_a$  sets the timescale of plasticity and  $r_a^0$  is the target rate of population  $a = e_1, e_2, i$ . For simplicity, we consider a rectified-linear f-I curve,

$$f(I) = \begin{cases} gI & I > 0 \\ 0 & I \leq 0 \end{cases}. \quad (8)$$

The gain,  $g$ , was fit to spiking network simulations (see Materials and Methods).

Simulating this model shows excellent agreement with the firing rates from the spiking network simulations (Figure 2) and the mean-field simulations are computationally more efficient than the spiking network simulations by a factor of 70 (6.0s for the mean-field simulation compared to 435.0s for the spiking network simulation). The deviation of the firing rates in the mean-field rate model from their targets can be quantified by

$$MSE_{mf} = \sum_{a=e_1, e_2, i} q_a (r_a - r_a^0)^2 \quad (9)$$

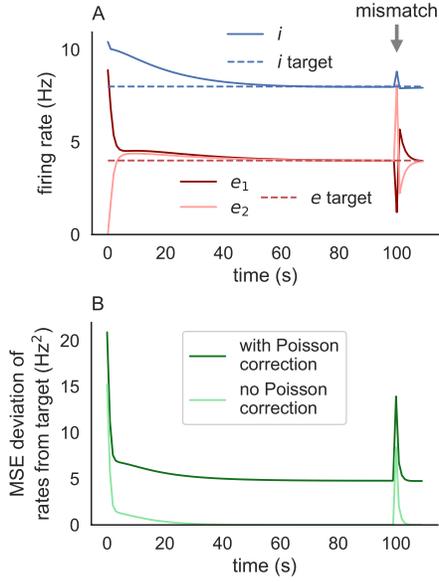


Figure 2: **A mean-field firing rate model captures the dynamics of the spiking network model.** **A)** Firing rates of the mean-field firing rate model defined by Eqs. (6) and (7). Compare to Figure 1C. **B)** MSE deviation of the firing rates from their targets ( $MSE_{mf}$ ; light green) and the MSE with a Poisson correction ( $MSE_{Poisson}$ ; dark green). Compare to Figure 1D.

which is identical to  $MSE_{mean}$  above except that  $r_a$  represents the rate from the mean-field simulations instead of the mean firing rates from the spiking net simulations. Indeed,  $MSE_{mf}$  closely matches  $MSE_{mean}$  from the spiking network simulations (Figure 2B, compare to Figure 1C), demonstrating that the two models have similar mean-field dynamics. The value of  $MSE_{pop}$  from the spiking network simulations does not have a direct analogue in the mean-field model, but under an assumption of Poisson-like spike-timing variability in the spiking network,  $MSE_{pop}$  can be approximated by (see Materials and Methods for derivation)

$$MSE_{Poisson} = MSE_{mf} + \frac{1}{T} \sum_a q_a r_a \quad (10)$$

where  $r_a$  is the firing rate of population  $a = e_1, e_2, i$  from the mean-field model and  $T$  is length of the time window over which firing rates are computed in the spiking network simulations. Specifically,  $MSE_{Poisson}$  represents the population-level MSE (*i.e.*,  $MSE_{pop}$ ) that would be produced by populations of Poisson spike trains with firing rates  $r_a$ . Indeed,  $MSE_{Poisson}$  shows close agreement with  $MSE_{pop}$  (Figure 2B, compare to Figure 1D), demonstrating that the deviation of  $MSE_{pop}$  away from the values of  $MSE_{mean}$  is consistent with Poisson-like spike-timing variability.

This example shows that homeostatic inhibitory synaptic plasticity can train a network to detect mismatched stimuli, which is a form of predictive coding. To better understand how and why the network is able to detect mismatched stimuli, we consider a fixed point analysis via a separation of timescales.

In the absence of plasticity ( $W$  fixed, *e.g.*,  $\eta_e = \eta_i = 0$ ), fixed point firing rates would satisfy  $\mathbf{r}_0 = f(W\mathbf{r}_0 + \mathbf{X})$ . Taking the rectified linear f-I curve from the dynamical mean-field model, if there were a fixed point with positive rates ( $r_a > 0$  for all  $a$ ) then it would be unique and given (as a

function of  $W$ ) by

$$\mathbf{r}(W) = [D - W]^{-1} \mathbf{X} = A\mathbf{X} \quad (11)$$

where  $D = (1/g)Id$  is a diagonal matrix,  $I$  is the identity matrix, and  $A = [D - W]^{-1}$ . With  $W$  fixed, the Jacobian matrix for the firing rate equation, Eq. (6), would be given by

$$\mathbf{J} = g \begin{bmatrix} (w_{e_1 e_1} - 1)/\tau_e & w_{e_1 e_2}/\tau_e & w_{e_1 i}/\tau_e \\ w_{e_1 e_2}/\tau_e & (w_{e_1 e_1} - 1)/\tau_e & w_{e_1 i}/\tau_e \\ w_{ie_1}\tau_i & w_{ie_2}\tau_i & (w_{ii} - 1)/\tau_i \end{bmatrix}$$

If the eigenvalues of this matrix have negative real part, then the fixed point given by Eq. (11) is stable and globally attracting.

Due to plasticity,  $W$  itself is time-dependent, so this fixed point analysis does not tell the full story. When plasticity is much slower than the firing rate dynamics ( $\eta$  sufficiently small and  $\tau$  sufficiently large, but  $\eta$  should not be compared directly to  $\tau$  because they have different dimensions), we can perform a separation of timescales under which  $\mathbf{r}$  relaxes to the quasi-steady-state value given by evaluating Eq. (11) at the current value of  $W$ , while  $W$  evolves more slowly according to Eq. (7). Putting this together, the separation of timescales approximation is defined by

$$\begin{aligned} \frac{dW}{dt} &= \begin{bmatrix} 0 & 0 & -\eta_e(r_{e_1} - r_0^e)r_i \\ 0 & 0 & -\eta_e(r_{e_2} - r_0^e)r_i \\ 0 & 0 & -\eta_i(r_i - r_0^i)r_i \end{bmatrix} \\ \mathbf{r} &= \begin{bmatrix} r_{e_1} \\ r_{e_2} \\ r_i \end{bmatrix} = [D - W]^{-1} \mathbf{X} = A\mathbf{X} \end{aligned} \quad (12)$$

Note that this is a 3-dimensional dynamical system because  $\mathbf{r}$  is defined by a functional relationship instead of a differential equations. Solving Eqs. (12) directly gives similar results to the full mean-field model and is 482 times more computationally efficient than the full mean-field simulations (Figure 3A,B;  $12.5 \times 10^{-3}$ s to simulate Eqs. (12) versus 6.0s for the full mean-field model) primarily because the slower dynamics allow for a larger time discretization (we used  $dt = 0.1$ ms for the full mean-field and  $dt = T = 1$ s to simulate Eqs. (12)). Simulating Eqs. (12) was 34751 times faster than the spiking network simulations. This speedup is not surprising given the lower dimension (2 versus 5000 dimensions) as well as the larger time discretization.

During training,  $\mathbf{X}$  is fixed to the ‘‘matched’’ value given by Eq. (25). During this phase, the slow-timescale system described by Eqs. (12) has a fixed point for which  $\mathbf{r} = \mathbf{r}^0$  where

$$\mathbf{r}^0 = \begin{bmatrix} r_e^0 \\ r_e^0 \\ r_i^0 \end{bmatrix}$$

is a vector of the target rates from the plasticity rule. However, this expression gives the fixed point in terms of  $\mathbf{r}$  whereas the dynamical system is described by the dynamics of the entries of  $W$ . If the network converges to the target rates during training, then the weight matrix,  $W$ , for the slow system converges to a value,  $W^0$  (or, equivalently,  $A$  converges to a value of  $A^0$ ) that satisfies

$$[D - W^0]^{-1} \mathbf{X}^m = A^0 \mathbf{X}^m = \mathbf{r}^0 \quad (13)$$

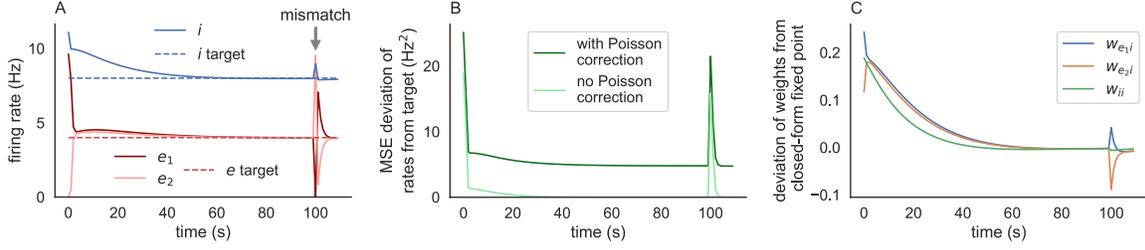


Figure 3: **Slow dynamics are captured by a separation-of-timescales approximation.** **A)** Firing rates of the model defined by Eqs. (12). Compare to Figures 1C and 2A. **B)** MSE deviation of the firing rates from their targets ( $MSE_{mf}$ ; light green) and the MSE with a Poisson correction ( $MSE_{Poisson}$ ; dark green) from the model defined by Eqs. (12). Compare to Figures 1D and 2B. **C)** Deviation of the inhibitory weights,  $w_{ai}$ , from the fixed point values given in Eqs. (14).

where

$$\mathbf{X}^m = \begin{bmatrix} X_e^0 + U \\ X_e^0 + V \\ X_i^0 \end{bmatrix}$$

is the value of  $\mathbf{X}$  for matched stimuli. Eq. (13) is a system of three equations for three unknowns ( $w_{e1i}$ ,  $w_{e2i}$ ,  $w_{ii}$ ) and its solution is given by

$$\begin{aligned} w_{e1i} &= \frac{r_e^0 - 2gr_e^0 w_{ee} - g(U + X_e^0)}{gr_i^0} \\ w_{e2i} &= \frac{r_e^0 - 2gr_e^0 w_{ee} - g(V + X_e^0)}{gr_i^0} \\ w_{ii} &= \frac{r_i^0 - 2r_e^0 w_{ie} + X_i^0}{gr_i^0} \end{aligned} \quad (14)$$

Indeed, the weights converged toward these fixed point values during the training period (before the mismatch stimulus; Figure 3C).

When the input is changed by a mismatched stimulus (so  $\mathbf{X}$  changes away from its value during training), firing rates deviate from their targets. Using the same quasi-steady state approximation, we can quantify the magnitude of this deviation as

$$\begin{aligned} d\mathbf{r} &:= \mathbf{r}^{mm} - \mathbf{r}^0 \\ &= A^0 \mathbf{X}^{mm} - \mathbf{r}^0 \\ &= A^0 (\mathbf{X}^{mm} - \mathbf{X}^m) \\ &= A^0 d\mathbf{X} \end{aligned} \quad (15)$$

where  $\mathbf{r}^{mm}$  is the vector of firing rates during a mismatched trial,  $\mathbf{r}^0 = [r_e^0 \ r_i^0]^T$  is the vector of target rates, and

$$d\mathbf{X} = \mathbf{X}^{mm} - \mathbf{X}^m = \begin{bmatrix} 0 \\ -V \\ 0 \end{bmatrix}$$

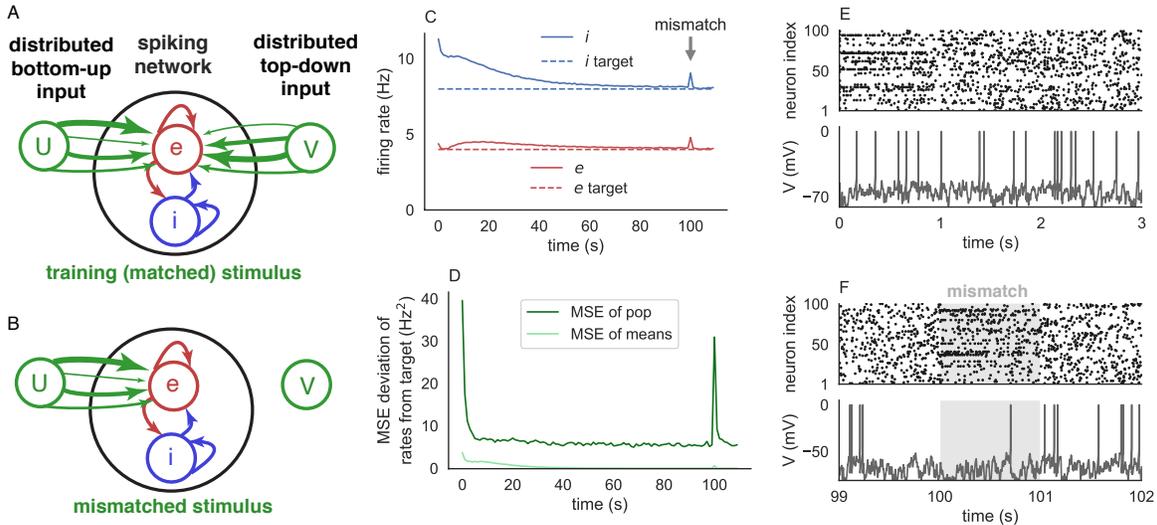


Figure 4: **Prediction errors after training on distributed time-constant inputs.** Same as Figure 1 except bottom-up and top-down inputs were modeled as distributed stimuli using multivariate Gaussian inputs vectors (Eq. (18)).

is the perturbation of the external stimulus away from its training value during the mismatched trial. This derivation makes it clear that larger perturbations of the stimulus (larger values of  $\|d\mathbf{X}\|$ ) generally lead to larger deviations of the firing rates from their targets (larger values of  $\|d\mathbf{r}\|$ ). Here and elsewhere,  $\|\cdot\|$  refers to the Euclidean norm.

Firing rate perturbations,  $\|d\mathbf{r}\|$ , are especially large if the input perturbations,  $d\mathbf{X}$ , point in a direction in which  $A^0 d\mathbf{X}$  is large. Such directions correspond to the directions indicated by the largest eigenvalue(s) of  $A^0$ . Since  $A^0 = [D - W^0]^{-1}$ , when  $W^0$  is much larger than  $D$  in magnitude, these directions correspond to directions indicated by the smallest eigenvalue(s) of  $W^0$ . This phenomenon is an instance of “imbalanced amplification” in which a perturbation that points toward the nullspace or “approximate nullspace” of the connectivity matrix,  $W^0$ , is amplified by the network, see [36] for more in-depth explanations.

Temporarily ignoring the direction of the perturbation, we can make the rough approximation that  $\|d\mathbf{r}\|$  is approximately proportional to  $\|d\mathbf{X}\|$ . This rough approximation provides the intuition for mismatched responses shown in the simulations above. Put simply, mismatched responses are caused by the deviation of a stimulus away from its “matched” training value and the magnitude of the mismatched response increases with the magnitude of the input perturbation. While this intuition may seem trivial for this example, its extensions will help explain some non-trivial, counterintuitive results below.

### 2.3 Prediction errors after training on distributed, time-constant inputs

The example above modeled a stimulus that was homogeneous across each neural population, *i.e.*, every neuron in population  $e_1$  received the same input and every neuron in population  $e_2$  received the same input. Stimulus representations in cortical circuits can be distributed in an inhomogeneous way across neural populations [38].

We next considered a spiking network model with distributed bottom-up and top-down inputs

(Figure 4A). As above, matched and mismatched stimuli were defined by the presence and absence of top-down input to population  $e_2$  (Eqs. (25) and (5)) to match the bottom-up input to population  $e_1$ , but these inputs are heterogeneous vectors ( $\vec{U}$  and  $\vec{V}$ ) instead of homogeneous scalars ( $U$  and  $V$ ). Specifically, matched and mismatched stimuli to excitatory neurons were defined by

$$X_e = X_e^0 + \vec{U} + \vec{V} \} \text{ matched} \quad (16)$$

and

$$X_e = X_e^0 + \vec{U} \} \text{ mismatched.} \quad (17)$$

where  $\vec{U}$  and  $\vec{V}$  are normally distributed  $N_e$ -dimensional vectors,

$$\begin{aligned} \vec{U} &\sim \sigma_s N(0, 1) \\ \vec{V} &\sim \sigma_s N(0, 1). \end{aligned} \quad (18)$$

Here,  $N(0, 1)$  is a standard multivariate normal distribution and  $\sigma_s = X_e^0/5$  controls the strength of the stimuli. Importantly, this means that each neuron receives a different value of top-down and bottom-up input, in contrast to the previous example (Eq. (4) and Figures 1–3) in which every neuron in the same excitatory sub-population received the same input.

Simulating this spiking network model shows that population-averaged firing rates converge to their targets during training on matched stimuli, as expected, but only deviate slightly from their targets in response to a mismatched stimulus (Figure 4C).

We suspected that the deviation of mean excitatory and inhibitory firing rates was small because some neurons increased their firing rates and some neurons decreased their firing rates in response to mismatched stimuli, so the increases and decreases cancelled at the level of population averages. Another way to see this is to note that the expected value of  $\vec{U}$  and  $\vec{V}$  is zero, so the absence of  $\vec{V}$  does not affect the population-averaged value of the inputs and (under a linear approximation) we should not expect a change in mean firing rates by removing  $\vec{V}$ . Under this reasoning, the firing rates of individual neurons would still change in response to a mismatched stimulus because individual elements of  $\vec{V}$  are non-zero. This line of reasoning implies that  $MSE_{mean}$  should not increase much for a mismatched stimulus, but  $MSE_{pop}$  should increase more for a mismatched stimulus. Indeed, this is exactly what we observed in simulations (Figure 4D).

In summary, our network model with iSTDP learned to adjust inhibitory weights in such a way to “match” or “cancel” top-down input with bottom-up input in the sense that the firing rates approach their target rates in response to matched stimuli after sufficient training. Moreover, the network responded to mismatched stimuli with deviations of the firing rates away from their target values. Note that the deviation of firing rates from their targets is not a consequence of the mismatch alone, but is due to the network being trained on matched stimuli. In this sense, the network is simply detecting deviations of its input patterns from the input patterns on which it was trained.

## 2.4 A lack of detectable prediction errors after training with time-varying stimuli

While instructive, the examples above were restricted to input patterns that were held fixed during training. In other words, the network only learned to associate *one* bottom-up input,  $U$ , with *one* top-down input,  $V$  (as schematized in Figure 3D). Since animals are exposed to multiple stimuli, a more realistic model would be trained on multiple pairings of top-down and bottom-up inputs. For

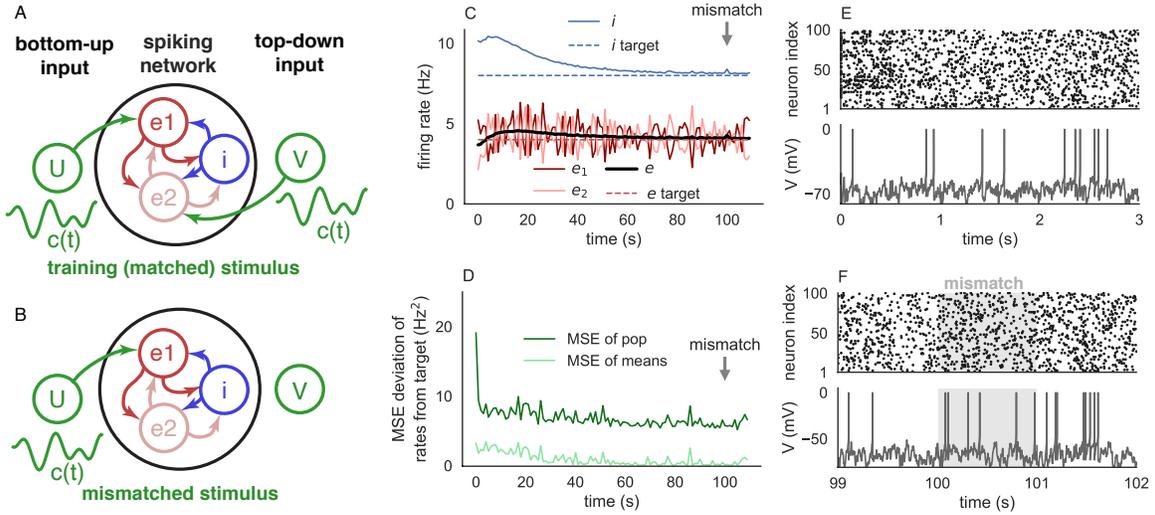


Figure 5: **A lack of detectable prediction errors in a model with time-varying stimuli.** **A,B)** Network schematic. Same as Figure 1A except the magnitude of the top-down and bottom-up stimuli were multiplied by the same time-varying signal,  $c(t)$ . **C-F)** Same as Figure 1C-F except we additionally plotted the mean excitatory firing rates (black curve in C).

example, in the visuomotor system, head motion (which we can interpret as top-down input,  $V$ ) is coupled with movement of an animal’s visual stimulus (which we can interpret as bottom-up input,  $U$ ). But head motion varies in direction and speed, and the movement of a visual scene covaries with it. Prediction errors arise whenever the learned covariation between head motion and visual stimulus is violated, *i.e.*, whenever there is a mismatch between top-down and bottom-up input [1, 3, 2, 39].

We next considered a simple extension of the first input model from Figures 1–3 to account for top-down and bottom-up inputs with time-varying intensity. Specifically, the excitatory neurons were again broken into two sub-populations,  $e_1$  and  $e_2$ . During training, each neuron in populations  $e_1$  and  $e_2$  received external stimuli of the form (Figure 1A)

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + c(t)U \\ X_{e_2} &= X_e^0 + c(t)V \end{aligned} \right\} \text{matched} \quad (19)$$

where  $c(t)$  is a scalar time-series that changes on each trial. Specifically,  $c(t)$  is drawn independently from a uniform distribution on  $[0, 2]$  at the start of each 1s trial. Hence, the expected value of  $c(t)$  is 1 and therefore, the expected values of  $X_{e_1}$  and  $X_{e_2}$  are the same as in the example from Figures 1–3, but they vary around this expectation across time. We used similar top-down and bottom-up, but needed to make the inputs weaker to avoid very large rate deviations,

$$\begin{aligned} U &= X_e^0/20 \\ V &= -X_e^0/20. \end{aligned} \quad (20)$$

Hence, bottom-up input,  $c(t)U$ , is matched by top-down input,  $c(t)V$ , during training. After training on matched stimuli, we again modeled mismatched stimuli by the absence of top-down input

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + U \\ X_{e_2} &= X_e^0. \end{aligned} \right\} \text{mismatched.} \quad (21)$$

The input to  $e_1$  is not out of the ordinary during a mismatched stimulus (it corresponds to the value when  $c(t) = 1$  is equal to its expectation) and the input to  $e_2$  is not out of the ordinary either (it corresponds to the value when  $c(t) = 0$ ), the joint value of the inputs to  $e_1$  and  $e_2$  together is out of the ordinary because the inputs are not matched (see Figure 5A for a schematic).

We reasoned that if our iSTDP rule could learn the relationship between top-down and bottom-up input during training, then it would detect the mismatch between them by evoking a larger deviation of firing rates from their targets. In other words, the network should detect the out-of-distribution input represented by a mismatch. However, our spiking network simulations contradicted this prediction. Firing rates deviated from the targets even during matched stimuli and the deviation in response to a mismatched stimulus was similar in magnitude (Figure 5B–F). Hence, the response to a mismatched stimulus was not detectable in the sense that it could not be distinguished from the response to matched stimuli.

## 2.5 A mean-field explanation for the absence of mismatch responses after training on time-varying inputs.

We now return to our mean-field theory to better understand why we do not see mismatch responses after training on time-varying inputs, but we do see them after training on time-constant inputs. We first simulated dynamical rate model from Eqs. (6)–(8) with the time-dependent stimuli defined by Eqs. (19)–(21). As above, the dynamical mean-field rate model captured the general trends from the spiking network simulations (compare Figure 6A,B to Figure 5C,D). Eq. (11) for the quasi steady-state firing rates generalizes to

$$\mathbf{r}(W) = [D - W]^{-1} \mathbf{X}(t) = A\mathbf{X}(t) \quad (22)$$

An assumption underlying Eq. (22) is that  $X(t)$  changes more slowly than the timescales ( $\tau_a$  for  $a = e, i$ ) at which firing rates evolve. This assumption is valid in our case because  $\mathbf{X}(t)$  switches every 1s while  $\tau_a \leq 6\text{ms}$ .

Now we can transition to the slower timescale dynamics of  $W$  by re-writing Eqs. (12) as

$$\begin{aligned} \frac{dW}{dt} &= \begin{bmatrix} 0 & 0 & -\eta_e(r_{e_1}(t) - r_0^e)r_i(t) \\ 0 & 0 & -\eta_e(r_{e_2}(t) - r_0^e)r_i(t) \\ 0 & 0 & -\eta_i(r_i(t) - r_0^i)r_i(t) \end{bmatrix} \\ \mathbf{r}(t) &= \begin{bmatrix} r_{e_1}(t) \\ r_{e_2}(t) \\ r_i(t) \end{bmatrix} = [D - W]^{-1} \mathbf{X}(t) = A\mathbf{X}(t) \end{aligned} \quad (23)$$

where we have only added the explicit time-dependence. Simulating this system shows general agreement with the trends from the spiking networks simulations and the dynamical mean-field model (Figure 7A,B, compare to Figure 5C,D and Figure 6A,B).

Due to the time-dependence of  $\mathbf{X}(t)$  in the current example, Eqs. (23) do not have a fixed point, so we cannot proceed directly with the fixed point analysis from above. To perform a fixed point analysis on  $W$ , we must assume that plasticity is slower than the stimulus, *i.e.*, that  $W(t)$  changes much more slowly than  $\mathbf{X}(t)$ . This assumption is valid for our simulations and even more so for biological neural circuits. Under this assumption, the slow timescale dynamics of  $W$  evolve based on

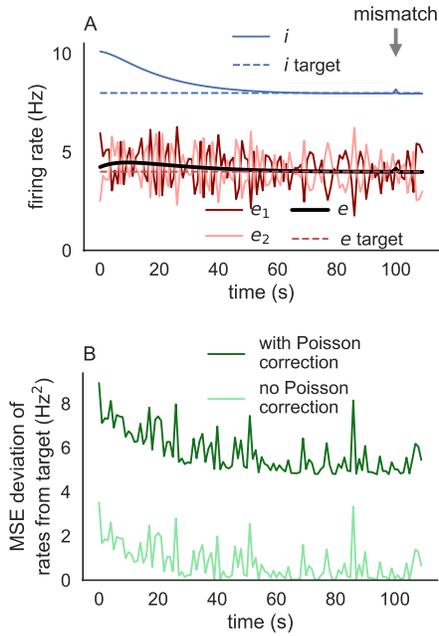


Figure 6: **Mean-field rate model with time-varying stimuli.** A,B) Same as Figure 2 except using the time-varying stimuli from Figure 5.

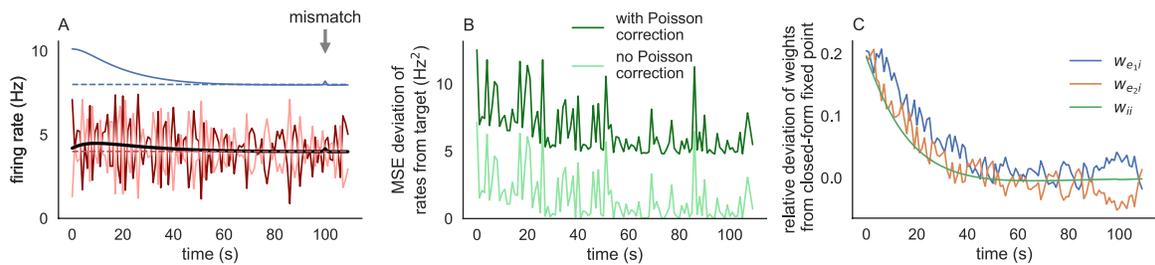


Figure 7: **Slow dynamics captured by a separation of timescales in a model with time-dependent stimuli.** B-C) Same as Figure 3 except using the time-varying stimuli from Figure 5B-C. D) Same as Figure 5D except time-dependent stimuli during training are represented by multiple dots (each one representing the inputs on one trial) and the mean is represented by a purple x.

the mean value of  $\mathbf{X}(t)$ . Specifically, we can use the approximation

$$\begin{aligned} \frac{dW}{dt} &= \begin{bmatrix} 0 & 0 & -\eta_e(\bar{r}_{e1} - r_0^e)\bar{r}_i \\ 0 & 0 & -\eta_e(\bar{r}_{e2} - r_0^e)\bar{r}_i \\ 0 & 0 & -\eta_i(\bar{r}_i - r_0^i)\bar{r}_i \end{bmatrix} \\ \bar{\mathbf{r}} &= \begin{bmatrix} \bar{r}_{e1} \\ \bar{r}_{e2} \\ \bar{r}_i \end{bmatrix} = [D - W]^{-1} \bar{\mathbf{X}} = A\bar{\mathbf{X}} \end{aligned} \quad (24)$$

where

$$\bar{\mathbf{X}} = E_t[\vec{X}(t)]$$

and  $E_t$  denotes the expectation over time during training, *i.e.*, during matched stimuli.

During training (for matched stimuli), we have from Eq. (19) that

$$\mathbf{X}^m(t) = \begin{bmatrix} X_e^0 + c(t)U \\ X_e^0 + c(t)V \\ X_i^0 \end{bmatrix} \quad (25)$$

Since  $E_t[c(t)] = 1$ , we have that

$$\bar{\mathbf{X}} = \begin{bmatrix} X_e^0 + U \\ X_e^0 + V \\ X_i^0 \end{bmatrix} \quad (26)$$

which is the same as the model from Figures 1–3. Hence, under this approximation,  $W$  should converge to the same fixed point in Eq. (14). Notably, this implies that the time-averaged rates should be equal to the target rates,  $\bar{\mathbf{r}} = \mathbf{r}^0$ . As predicted, simulations show that average firing rates are close to their targets (Figure 7A) and the weights do converge to the given fixed point with the addition of some noise (Figure 7C) coming from the noisy time-dependence of  $\mathbf{X}(t)$  and  $\mathbf{r}(t)$ .

Therefore, the state of the network (as represented by  $W$ ) after training is similar for the networks with time-constant and time-dependent stimuli. As a result, the deviation,  $d\mathbf{r}(t)$ , of the firing rates from their targets on any given trial takes the same form derived in Eq. (15),

$$\begin{aligned} d\mathbf{r}(t) &:= \mathbf{r}(t) - \mathbf{r}^0 \\ &= A^0 \mathbf{X}(t) - \mathbf{r}^0 \\ &= A^0 (\mathbf{X}(t) - \bar{\mathbf{X}}) \\ &= A^0 d\mathbf{X}(t) \end{aligned} \quad (27)$$

where  $d\mathbf{X}(t) = \mathbf{X}(t) - \bar{\mathbf{X}}$  is the deviation of the stimulus from the mean value it takes during training and  $A^0$  is the fixed point of  $A = [D - W]^{-1}$  after training (see Eq. (13) and surrounding discussion). This conclusion assumes that the mean-field approximation in Eq. (22) is approximately accurate or, more specifically, that the firing rate response to a perturbation is approximately a linear function of the input perturbation. This, in turn, requires that the input perturbation is not too strong.

As a heuristic, we can ignore the effect of  $A^0$  in Eq. (27) and make the approximation that  $d\mathbf{r}(t)$  is larger whenever  $d\mathbf{X}(t)$  is larger. In other words,

$$\begin{aligned} \|d\mathbf{r}(t)\| &= \|A^0 d\mathbf{X}(t)\| \\ &\approx \|A^0\| \|d\mathbf{X}(t)\| \\ &\propto \|d\mathbf{X}(t)\|. \end{aligned} \quad (28)$$

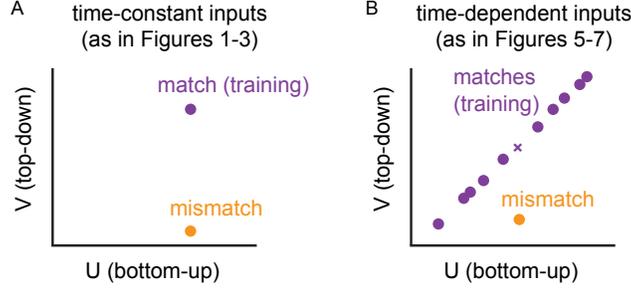


Figure 8: **Schematic illustrating why mismatch responses are detectable after training on time-constant, but not time-dependent stimuli.** **A)** Schematic representing inputs to the network in a model with time-constant stimuli. Training stimuli occupy a single point in  $(U, V)$  space (purple dot). The deviation of firing rates from their targets on any particular trial is approximately proportional to the distance of the input from its value during training (Eq. (15)). Since the mismatch stimulus (orange dot) is far from the matched, training stimulus, firing rates deviate from their target in response to the mismatched stimulus (as seen in Figures 1–3). **B)** Schematic representing inputs to the network in a model with time-varying stimuli. Training stimuli (purple dots) vary in  $(U, V)$  space along a predictable line. The mismatched stimulus lies far from this line. However, the deviation of firing rates from their targets on any particular trial is approximately proportional to the distance of the input from its *mean* value during training (Eq. (15)). Since the distance between the mismatch input (orange dot) and the mean training stimulus (purple x) is similar to the typical distance between the individual training stimuli (purple dots) and the mean training stimulus (purple x), the deviation of the firing rates from their targets is similar for matched and mismatched stimuli.

where  $\|A^0\|$  denotes the induced Euclidean norm on  $A^0$ . In other words, stimuli that are further from the mean training stimuli evoke larger firing rates. Note that we necessarily have  $\|A^0 d\mathbf{X}(t)\| \leq \|A^0\| \|d\mathbf{X}(t)\|$ , so this assumption is saying that  $\|A^0 d\mathbf{X}(t)\|$  is not much smaller than  $\|A^0\| \|d\mathbf{X}(t)\|$ . This approximation assumes that  $d\mathbf{X}(t)$  is not close to being orthogonal to the rows of  $A^0$ .

During matched stimuli, combining Eqs. (25) and (26) gives the perturbation for training stimuli

$$d\mathbf{X}^m(t) = \begin{bmatrix} (1 - c(t))U \\ (1 - c(t))V \\ 0 \end{bmatrix}.$$

Since  $|U| = |V|$ , we have

$$\begin{aligned} \|d\mathbf{X}^m(t)\|^2 &= 2(1 - c(t))^2|V| \\ &= 2u^2(t)|V| \end{aligned}$$

where  $u(t) = 1 - c(t)$  is uniformly distributed on  $[-1, 1]$ . Hence, the squared distance of  $\mathbf{X}(t)$  from its mean varies between 0 and  $2|V|$ . During the mismatched stimulus, we have from Eq. (21), that

$$\mathbf{X}^{mm} = \begin{bmatrix} X_e^0 + U \\ X_e^0 \\ X_i^0 \end{bmatrix}$$

Combining this with Eq. (26) shows that, during a mismatched stimulus, the input perturbation is

$$d\mathbf{X}^{mm} = \begin{bmatrix} 0 \\ -V \\ 0 \end{bmatrix}$$

and therefore

$$\|d\mathbf{X}^{mm}\|^2 = |V|.$$

Hence, the deviation of the external input,  $\mathbf{X}(t)$ , from its mean value during training is similar in magnitude during matched and mismatched stimuli. As a result, the deviation of the firing rates from their targets is also similar during matched and mismatched stimuli, so the mismatch is not detectable based on the deviation of firing rates from their targets alone.

This intuition, and how it differs from the time-constant model of Figures 1–3, is illustrated in Figure 8. For the model with time-constant inputs, there is only one stimulus during matched, training trials (Figure 8A, purple dot). Since the mismatch stimulus is far from this matched stimulus, the firing rate deviates from its target in response to the mismatched stimulus (as demonstrated in Figures 1–3). For the model with time-varying stimuli, there are multiple training stimuli that lie along a line (Figure 8B, purple dots). While the mismatch stimulus is clearly away from this line (Figure 8B, orange dot), the deviation of the firing rates from their targets is approximately proportional to how far an input is from the *mean* training stimulus (Figure 8B, purple x). Since this distance is similar for the mismatch stimulus and a typical training stimulus, the deviation of the firing rates from their targets is also similar during matched and mismatched stimuli (as demonstrated in Figures 5–7). While this intuition might seem obvious in hindsight, the complexity of dynamics in recurrent spiking neural network models can make this conclusion difficult to foresee without the benefit of the mean-field analysis provided here.

For the sake of completeness, we also considered a model with distributed, time-varying stimuli. Specifically, we combined the time-varying stimuli from the example in Figure 7 with the distributed stimuli from the example in Figure 4 to get inputs of the form (Figure 9A,B)

$$X_e = X_e^0 + c(t)\vec{U} + c(t)\vec{V} \} \text{ matched} \quad (29)$$

and

$$X_e = X_e^0 + \vec{U} \} \text{ mismatched.} \quad (30)$$

where  $c(t)$  is a scalar drawn from a uniform distribution on  $[0, 2]$  on each trial, and  $\vec{U}$  and  $\vec{V}$  are normally distributed  $N_e$ -dimensional vectors as in Eq. (18). Unsurprisingly, given the failure on the simpler example discussed above, the spiking network model did not produce an easily detectable response to mismatched stimuli (Figure 9C-F). Specifically, the deviation of the firing rates away from their targets was similar in matched and mismatched trials (Figure 9C,D).

## 2.6 How do our conclusions generalize to other network models?

The mean-field analysis above relied on several assumptions that were used to derive approximations. This raises the question of how general our conclusions are. Specifically, for which network models does the argument above imply an absence of noticeable mismatch responses? To answer this question, we can distill the argument above into three fundamental assumptions:

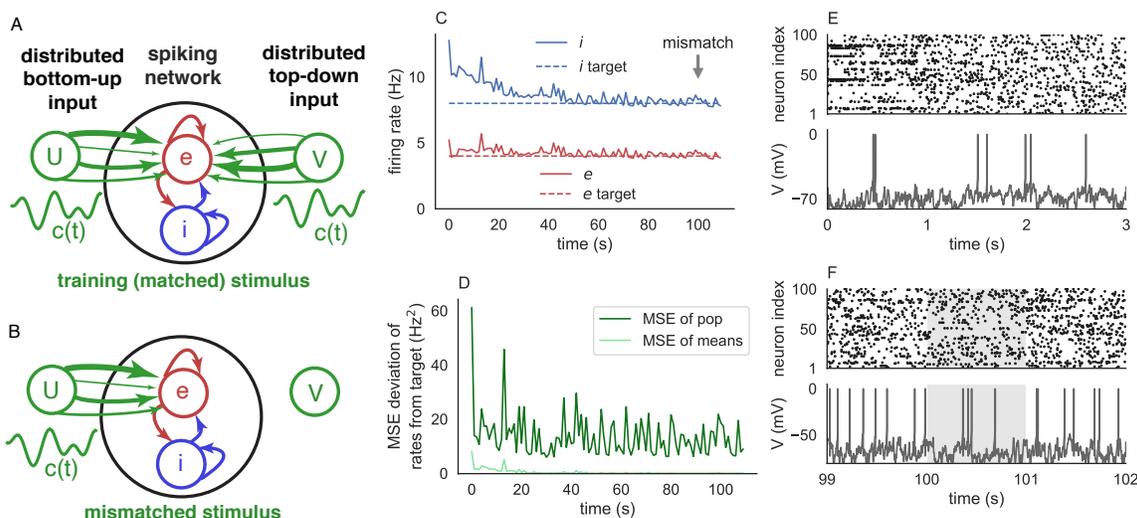


Figure 9: **Prediction errors after training on distributed time-dependent inputs.** Same as Figure 4 except bottom-up and top-down inputs were time-dependent, as described by Eqs. (29)–(30).

1. The linear approximation in Eq. (27) should be approximately accurate,

$$dr(t) \approx A^0 d\mathbf{X}(t)$$

While this assumption is strong, it should be satisfied when  $d\mathbf{X}(t)$  is sufficiently small. In addition, balanced excitation and inhibition linearize the firing rate responses of networks to external input [30, 40, 41, 42, 36, 43], so this assumption should hold in networks with balanced excitation and inhibition, which is encouraged by inhibitory synaptic plasticity [17, 20, 22, 24].

2. The approximation in Eq. (28) should be accurate, specifically

$$\|A^0 d\mathbf{X}(t)\| \approx \|A^0\| \|d\mathbf{X}(t)\|$$

which requires that  $d\mathbf{X}(t)$  not be close to orthogonal to the rows of  $A^0$ .

3. The magnitude of the input perturbations for a mismatched stimulus should be similar to a typical value during matched stimuli,

$$\|d\mathbf{X}^{mm}\| \approx \|d\mathbf{X}^m(t)\|$$

In general, if a model satisfies these three assumptions then  $dr(t)$  is similar in magnitude during matched and mismatched stimuli. Note that these assumptions are sufficient, but not necessary for a lack of mismatch responses. For example, if assumption 1 is violated because the rate perturbations are nonlinear, then the nonlinear model might still not compute mismatch responses.

Strictly speaking, assumption 2 is stronger than needed. Instead, we only need that the relationship between  $A^0$  and  $d\mathbf{X}$  is similar for matched and mismatched stimuli, *i.e.*, that

$$\frac{\|A^0 d\mathbf{X}^m(t)\|}{\|A^0\| \|d\mathbf{X}^m(t)\|} \approx \frac{\|A^0 d\mathbf{X}^{mm}\|}{\|A^0\| \|d\mathbf{X}^{mm}\|}$$

which is a weaker assumption because it allows for  $d\mathbf{X}$  to be aligned with the rows of  $A^0$  so long as the alignment is similar for matched and unmatched stimuli.

For our examples in which the network is trained on time-constant input (Figures 1–4), we have that  $\mathbf{X}^m(t) = \overline{\mathbf{X}}$ , so  $d\mathbf{X}^m(t) = 0$  whereas  $d\mathbf{X}^{mm} \neq 0$ , so assumption 3 above is not met. This explains why our examples trained on time-constant were able to produce robust mismatch responses.

In previous work [14], a network with homeostatic plasticity successfully computed prediction errors after training on time-varying stimuli. In that work, the weights of the connectivity matrix were carefully chosen so that  $A$  was singular and the directions of the input perturbation during matched stimuli (the “feedback” stimulus condition) was in the nullspace of  $A^0$ . See equation 28 in their appendix and note that  $A^0$  was called  $W$  in their analysis. As a result, the model studied there does not satisfy assumption 2 above. This explains how [14] were able to compute prediction errors with time-varying inputs.

In all of the examples we have considered so far, external input was provided to excitatory neurons only. However, our analysis implies that our overall results should still hold if input is provided to inhibitory neurons as well. Specifically, in Eqs. (27) and the surrounding equations and analysis, there is nothing preventing  $d\mathbf{X}(t)$  from having a non-zero component for the inhibitory population(s). To verify this prediction, we repeated all of the spiking network simulations (those in Figures 1, 4, 5, and 9) in models in which external input was also added to the inhibitory population. Our results show the same overall conclusions for all figures (see Supplementary Materials Section 1 and Supplementary Figures 1–2). Specifically, in all examples, a noticeable mismatch response was observed after training on time-constant inputs, but not after training on time-varying inputs.

Assumption 3 above implies that mismatch responses could be possible after training on time-varying stimuli if the mismatch stimulus is larger in magnitude than the matched stimuli used during training. While this is not necessarily a surprising finding (a larger stimulus should evoke a larger response), we decided to test it in a simulation. Specifically, we repeated the simulation from Figure 5, but we scaled the magnitude of the mismatched input by a factor of six. These simulations confirm that a mismatch response was produced in this case (Supplementary Figure 3).

In all of the examples above, we considered only a single inhibitory population and at most two excitatory populations. In reality, there are multiple inhibitory neuron subtypes in the cortex and previous work on mismatch responses with inhibitory plasticity accounts for this [14, 23]. Our analysis above implies that increasing the number of neuron populations alone should not affect our overall conclusions. To test our findings empirically on a model with several neural populations, we performed a simulation that was identical to the simulation in Figure 5 except we used three inhibitory and three excitatory populations. Consistent with our theoretical predictions, the results were qualitatively similar to those in Figure 5: After training on time-dependent stimuli, there was no noticeable deviation of firing rates in response to a mismatched stimulus (see Supplementary Figure 4).

### 3 Discussion

We combined numerical simulations of spiking networks and mean-field rate models with mathematical analysis to evaluate the extent to which homeostatic inhibitory synaptic plasticity can train an unstructured network to compute prediction errors. We found that the networks successfully learn to compute prediction errors when training stimuli are static. Specifically, if top-down and bottom-up inputs are fixed in time during training, then firing rates in the trained network will maintain a baseline

firing rates in response to stimuli that match the training stimuli, but firing rates will deviate from their baseline levels in response to mismatched stimuli. This result holds when stimuli are uniform (with each of a few sub-populations receiving homogeneous external input) or when stimuli are distributed (with each neuron receiving distinct, but time-constant levels of external input during training).

To our surprise, simulations showed that even under a simple model of time-varying stimuli, in which bottom-up and top-down inputs are modulated by the same time-varying factor, the same networks fail to produce reliable mismatch responses after training. Specifically, firing rates deviate from their baseline levels by a similar amount in response to stimuli that are matched (a shared modulation, as in training) or mismatched (one input is modulated differently than the other). We used a mean-field approximation to explain these empirical findings and elucidate a set of conditions under which robust mismatch responses do not occur. Our results therefore help to clarify the extent to which homeostatic inhibitory synaptic plasticity is sufficient to train a network to compute mismatch responses.

For networks trained on time-varying inputs, our results show a lack of mismatch responses in the sense that firing rates do not deviate from their baseline (when deviation is measured by mean-squared error) more during mismatched inputs than they do for matched stimuli. However, mismatch responses could potentially be detected by some linear projection of the firing rates and this linear projection could be fed as input to a readout neuron that would be able to detect mismatch responses. However, our main goal was to understand the situations under which a natural homeostatic plasticity rule would spontaneously produce elevated responses to mismatched stimuli. Training a separate linear projection is outside the scope of this goal.

Inhibitory homeostatic synaptic plasticity is only one of many homeostatic mechanisms in the brain [44]. While homeostatic plasticity is one candidate mechanism for predictive coding, other homeostatic mechanisms could play a role as well. Future work should consider the potential role of other homeostatic mechanisms in predictive coding and mismatch detection.

Previous work [14, 23] found that networks with homeostatic plasticity *can* learn to compute mismatch responses in models with time-varying stimuli that are similar to the time-varying stimuli that we used (in the cases where our networks failed). They used a more biologically detailed network model with multiple inhibitory subtypes and multi-compartment excitatory neurons. Importantly, connectivity in their model was constrained so that matched stimuli were in the nullspace of the effective connectivity matrix ( $A$  in our work,  $W$  in theirs). Our theoretical analysis agrees with their analysis showing that this assumption is necessary for their overall results. We additionally provided a set of conditions under which more general classes of models will not produce robust mismatch responses, which generalizes some of the theoretical results in [14] to more general classes of networks. The requirement that matched stimuli are in the nullspace of the effective connectivity matrix is a strong assumption because it implies that the connectivity matrices must be precisely tuned. Moreover, the dimension of the nullspace of the connectivity matrix must match the dimensionality of the training stimuli, which could make it difficult to train a network to maintain baseline firing rates on a higher dimensional space of training stimuli.

Our study and the previous work described above [14, 23] incorporates homeostatic synaptic plasticity, but does not account for any other of the wide variety of synaptic plasticity rules observed in neural recordings. Other work has shown that predictive coding can be learned in carefully constructed networks using learning rules that are not exclusively homeostatic [12]. Indeed, our approach of learning prediction errors in unstructured, randomly connected networks could potentially be made successful if the target rates,  $r_0^a$ , were effectively modulated by the top-down or bottom-

up input. Future work should consider the possibility of learning prediction errors in unstructured, random networks by combining these approaches.

## 4 Materials and Methods

All simulations were performed by numerically solving the corresponding differential equations using the forward Euler method in custom written Python code. Code to produce all figures can be found at <https://github.com/RobertRosenbaum/PCISP>.

For spiking network simulations (Eqs. (1)–(2); Figures 1, 4, and 5) and mean-field rate network simulations (Eqs. (6)–(7); Figures 2 and 6) we used a time step size of  $dt = 0.1\text{ms}$ . For the slow-timescale model (Eqs. (24); Figures 3 and 7) we used a time step size of  $dt = 1\text{s}$ .

For all spiking network simulations (Eqs. (1)–(2); Figures 1, 4, and 5), we used  $N_e = 4000$  and  $N_i = 1000$  excitatory and inhibitory neurons. All neurons were connected with probability  $p_{ee} = p_{ei} = p_{ie} = p_{ii} = 0.1$ . Connected neurons had initial synaptic weights  $j_{ee} = 7.07\text{mV/ms}$ ,  $j_{ei} = -49.5\text{mV/ms}$ ,  $j_{ie} = 31.8\text{mV/ms}$ , and  $j_{ii} = -70.7\text{mV/ms}$ . EIF neuron parameters were  $\tau_m = 15\text{ms}$ ,  $E_L = -72\text{mV}$ ,  $V_{re} = -73\text{mV}$ ,  $D_T = 2\text{mV}$ ,  $V_T = -55\text{mV}$ ,  $V_{th} = 0\text{mV}$ , and a reflecting lower boundary on the membrane potential was placed at  $V_{lb} = -80\text{mV}$  to approximate an inhibitory reversal potential. Synaptic timescales were  $\tau_e = 6\text{ms}$  and  $\tau_i = 4\text{ms}$ . Baseline external input to excitatory and inhibitory neurons was  $X_e^0 = 42.4\text{mV}$  and  $X_i^0 = 28.3\text{mV}$ . Parameters for the inhibitory plasticity rule were  $\eta_e = 56.6\text{mV}$ ,  $\eta_i = 28.3\text{mV}$ , and  $\tau_{STDP} = 200\text{ms}$  with target rates at  $r_0^e = 4\text{Hz}$  and  $r_0^i = 8\text{Hz}$ . For mean-field rate network simulations (Eqs. (6)–(7); Figures 2 and 6), we used a gain of  $g = 0.001\text{ms/mV}$ , which was derived by simulating the spiking network model without plasticity and then fitting the f-I curve  $r = f(I) = gIH(I)$  (where  $H$  is the Heaviside step function) to the time-averaged firing rates and input currents of all neurons in the simulation. Learning rates for rate network simulations were  $\eta_e = 8944\text{mV}$  and  $\eta_i = 4472\text{mV}$ . All other parameters were the same as those used in spiking network simulations or their derivations are given in Results. Python code to simulate the networks reproduce the figures can be found on the last author’s academic webpage.

### 4.1 Derivation of Eq. (10) for $MSE_{Poisson}$ .

Here, we derive Eq. (10) for  $MSE_{Poisson}$ . Consider a population of  $N$  neurons divided into  $M$  sub-populations where sub-population  $a$  contains  $N_a$  neurons for  $a = 1, \dots, M$  ( $M = 3$  and  $a = e_1, e_2, i$  for the models considered in this paper). Assume that each neuron in population  $a$  spikes like a Poisson process with a rate of  $r_a$ . Let  $n_j^a$  be the number of spikes emitted by neuron  $j = 1, \dots, N_a$  in population  $a = 1, \dots, M$  during a time interval of duration  $T$  and let

$$r_j^a = \frac{n_j^a}{T}$$

be the sample firing rate of neuron  $j$ . Then each  $n_j^a$  has expectation and variance

$$E[n_j^a] = \text{var}(n_j^a) = r_a T$$

so each sample rate has expectation

$$E[r_j^a] = E\left[\frac{n_j^a}{T}\right] = r_a$$

and variance

$$\text{var}(r_j^a) = \text{var}\left(\frac{n_j^a}{T}\right) = \frac{r_a}{T}.$$

Now suppose we have a target rates of  $r_a^0$  for each neuron in population  $a$  and we would like to compute the population-wide MSE deviation of the sample rates from their targets. This can be written as

$$\begin{aligned} MSE_{pop} &= \frac{1}{N} \sum_{j=1}^N (r_j - r_j^0)^2 \\ &= \frac{1}{N} \sum_{a=1}^M \sum_{j=1}^{N_a} (r_j^a - r_a^0)^2 \\ &= \sum_{a=1}^M q_a \frac{1}{N_a} \sum_{j=1}^{N_a} (r_j^a - r_a^0)^2 \end{aligned}$$

where  $q_a = N_a/N$  is the proportion of neurons in population  $a$ ,  $r_j$  is the sample rate, and  $r_j^0$  is the rate parameter for neuron  $j = 1, \dots, N$ . The inner sum can be written as

$$\begin{aligned} \frac{1}{N_a} \sum_{j=1}^{N_a} (r_j^a - r_a^0)^2 &= (r_a^0 - r_a)^2 \\ &+ \frac{1}{N_a} \sum_{j=1}^{N_a} (r_j^a - r_a)^2 - 2(r_a^0 - r_a)(r_j^a - r_a). \end{aligned}$$

The first term in the sum is the sample variance of  $r_j^a$ , so

$$\frac{1}{N_a} \sum_{j=1}^{N_a} (r_j^a - r_a)^2 \approx \text{var}(r_j^a) = \frac{r_a}{T}.$$

when  $N_a$  is large. The last term in the sum can be ignored when  $N_a$  is large because

$$\begin{aligned} \frac{1}{N_a} \sum_{j=1}^{N_a} (r_a^0 - r_a)(r_j^a - r_a) &= (r_a - r_a^0) \left( r_a - \frac{1}{N_a} \sum_{j=1}^{N_a} r_j^a \right) \\ &\approx 0 \end{aligned}$$

since  $r_a$  is the expected value of  $r_j^a$ . Putting this altogether gives

$$\begin{aligned} MSE_{pop} &\approx \sum_{a=1}^M q_a \left[ (r_a - r_a^0)^2 + \frac{r_a}{T} \right] \\ &= MSE_{mf} + \frac{1}{T} \sum_{a=1}^M q_a r_a \end{aligned}$$

where

$$MSE_{mf} = \sum_{a=1}^M q_a (r_a - r_a^0)^2$$

is the mean-field MSE defined in Eq. (9). This calculation motivates the definition of the Poisson-corrected MSE,

$$MSE_{Poisson} = MSE_{mf} + \sum_{a=1}^M \frac{q_a r_a}{T}$$

as defined in Eq. (10). Specifically, our calculations above show that  $MSE_{Poisson}$  approximates the population-level MSE (i.e.,  $MSE_{pop}$ ) that would be produced if all of the spike trains in each sub-populations were Poisson processes. The approximation becomes exact as  $N_a \rightarrow \infty$ .

## Declarations

### Funding and/or conflicts of interest.

This work was supported by US National Foundation of Science grants NSF-DMS-1654268 and NSF NeuroNex DBI-1707400, and the Air Force Office of Scientific Research (AFOSR) under award number FA9550-21-1-0223. The authors have no conflicts of interest to disclose.

## References

- [1] G. B. Keller, T. Bonhoeffer, and M. Hübener. Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. *Neuron*, 74(5):809–815, 2012.
- [2] M. Leinweber, D. R. Ward, J. M. Sobczak, A. Attinger, and G. B. Keller. A sensorimotor circuit in mouse cortex for visual flow predictions. *Neuron*, 95(6):1420–1432, 2017.
- [3] A. Attinger, B. Wang, and G. B. Keller. Visuomotor coupling shapes the functional development of mouse visual cortex. *Cell*, 169(7):1291–1302, 2017.
- [4] H. Von Helmholtz. *Handbuch der physiologischen Optik: mit 213 in den Text eingedruckten Holzschnitten und 11 Tafeln*, volume 9. Voss, 1867.
- [5] G. B. Keller and T. D. Mrsic-Flogel. Predictive processing: a canonical cortical computation. *Neuron*, 100(2):424–435, 2018.
- [6] R. P. Rao and D. H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.
- [7] K. Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010.
- [8] A. Clark. *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press, 2015.
- [9] C. Wacongne, J.-P. Changeux, and S. Dehaene. A neuronal model of predictive coding accounting for the mismatch negativity. *Journal of Neuroscience*, 32(11):3665–3678, 2012.
- [10] A. M. Bastos, W. M. Usrey, R. A. Adams, G. R. Mangun, P. Fries, and K. J. Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

- [11] R. P. Rao and T. J. Sejnowski. Predictive coding, cortical feedback, and spike-timing dependent plasticity. *Probabilistic models of the brain*, page 297, 2002.
- [12] R. Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017.
- [13] J. C. Whittington and R. Bogacz. Theories of error back-propagation in the brain. *Trends in Cognitive Sciences*, 23(3):235–250, 2019.
- [14] L. Hertäg and H. Sprekeler. Learning prediction error neurons in a canonical interneuron circuit. *Elife*, 9:e57541, 2020.
- [15] A. Schulz, C. Miehl, M. J. Berry II, and J. Gjorgjieva. The generation of cortical novelty responses through inhibitory plasticity. *Elife*, 10:e65309, 2021.
- [16] P. E. Castillo, C. Q. Chiu, and R. C. Carroll. Long-term plasticity at inhibitory synapses. *Current opinion in neurobiology*, 21(2):328–338, 2011.
- [17] T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–73, dec 2011.
- [18] Y. Luz and M. Shamir. Balancing feed-forward excitation and inhibition via hebbian inhibitory synaptic plasticity. *PLoS computational biology*, 8(1):e1002334, 2012.
- [19] T. P. Vogels, R. C. Froemke, N. Doyon, M. Gilson, J. S. Haas, R. Liu, A. Maffei, P. Miller, C. J. Wierenga, M. A. Woodin, F. Zenke, and H. Sprekeler. Inhibitory synaptic plasticity: spike timing-dependence and putative network function. *Frontiers in Neural Circuits*, 7(119), 2013.
- [20] G. Hennequin, E. J. Agnes, and T. P. Vogels. Inhibitory Plasticity: Balance, Control, and Codependence. *Annu. Rev. Neurosci.*, 40(1):557–579, 2017.
- [21] M. Capogna, P. E. Castillo, and A. Maffei. The ins and outs of inhibitory synaptic plasticity: Neuron types, molecular mechanisms and functional roles. *European Journal of Neuroscience*, 54(8):6882–6901, 2021.
- [22] C. Baker, V. Zhu, and R. Rosenbaum. Nonlinear stimulus representations in neural circuits with approximate excitatory-inhibitory balance. *PLoS computational biology*, 16(9):e1008192, 2020.
- [23] L. Hertäg and C. Clopath. Prediction-error neurons in circuits with multiple neuron types: Formation, refinement and functional implications. *bioRxiv*, 2021.
- [24] A. E. Akil, R. Rosenbaum, and K. Josić. Balanced networks under spike-time dependent plasticity. *PLoS Computational Biology*, 17(5):e1008958, 2021.
- [25] R. Brette and W. Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J Neurophysiol*, 94(5):3637–3642, 2005.
- [26] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

- [27] C. van Vreeswijk and H. Sompolinsky. Methods and models in neurophysics course 9: Irregular activity in large networks of neurons. *Les Houches*, 80:341–406, 2005.
- [28] C. van Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274(5293):1724–1726, 1996.
- [29] D. Amit and N. Brunel. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cereb Cortex*, 7(3):237–252, 1997.
- [30] C. van Vreeswijk and H. Sompolinsky. Chaotic balanced state in a model of cortical circuits. *Neural Comput*, 10(6):1321–1371, 1998.
- [31] N. Brunel and V. Hakim. Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Comput*, 11(7):1621–1671, 1999.
- [32] N. Brunel. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J Comput Neurosci*, 8(3):183–208, 2000.
- [33] A. Renart, J. de La Rocha, P. Bartho, L. Hollender, N. Parga, A. Reyes, and K. Harris. The Asynchronous State in Cortical Circuits. *Science*, 327(5965):587–590, 2010.
- [34] R. Pyle and R. Rosenbaum. Highly connected neurons spike less frequently in balanced networks. *Phys Rev E*, 93(4):040302(R), 2016.
- [35] R. Pyle and R. Rosenbaum. Spatiotemporal dynamics and reliable computations in recurrent spiking neural networks. *Physical Rev Lett*, 118(1):018103, 2017.
- [36] C. Ebsch and R. Rosenbaum. Imbalanced amplification: A mechanism of amplification and suppression from local imbalance of excitation and inhibition in cortical circuits. *PLoS Comp Bio*, 14(3):e1006048, 2018.
- [37] C. Baker, C. Ebsch, I. Lampl, and R. Rosenbaum. Correlated states in balanced neuronal networks. *Phys Rev E*, 99(5):052414, 2019.
- [38] S. Saxena and J. P. Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.
- [39] R. Jordan and G. B. Keller. Opposing influence of top-down and bottom-up input on excitatory layer 2/3 neurons in mouse primary visual cortex. *Neuron*, 108(6):1194–1206, 2020.
- [40] R. Rosenbaum and B. Doiron. Balanced networks of spiking neurons with spatially dependent recurrent connections. *Phys Rev X*, 4(2):021039, 2014.
- [41] S. Lim and M. S. Goldman. Balanced cortical microcircuitry for spatial working memory based on corrective feedback control. *J Neurosci.*, 34(20):6790–6806, 2014.
- [42] I. D. Landau, R. Egger, V. J. Dercksen, M. Oberlaender, and H. Sompolinsky. The impact of structural heterogeneity on excitation-inhibition balance in cortical networks. *Neuron*, 92(5):1106–1121, 2016.
- [43] Y. Ahmadian and K. D. Miller. What is the dynamical regime of cerebral cortex? *Neuron*, 109(21):3373–3391, 2021.

- [44] G. Turrigiano. Too many cooks? intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement. *Annual review of neuroscience*, 34:89–103, 2011.

# Supplementary Materials for Evaluating the extent to which homeostatic plasticity learns to compute prediction errors in unstructured neuronal networks

Vicky Zhu and Robert Rosenbaum  
University of Notre Dame  
Notre Dame, IN USA

## 1 Adding external input to inhibitory populations does not alter main conclusions.

In the main manuscript, we only considered examples where external input was provided exclusively to excitatory neurons. In this section, we empirically test whether our conclusions were sensitive to the assumption that only excitatory neurons received external input by repeating some simulations from the main text with external input provided to the inhibitory populations as well. Supplementary Figure 1 shows results from a simulation in which top down input was provided to the inhibitory population in addition to population  $e_2$  during training. For the mismatched stimulus, the top-down input was removed from the inhibitory population and from population  $e_2$ . Specifically, in Supplementary Figure 1C,D, we used

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + U \\ X_{e_2} &= X_e^0 + V \\ X_i &= X_i^0 + V \end{aligned} \right\} \text{matched}$$

and

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + U \\ X_{e_2} &= X_e^0 \\ X_i &= X_i^0 \end{aligned} \right\} \text{mismatched}$$

where

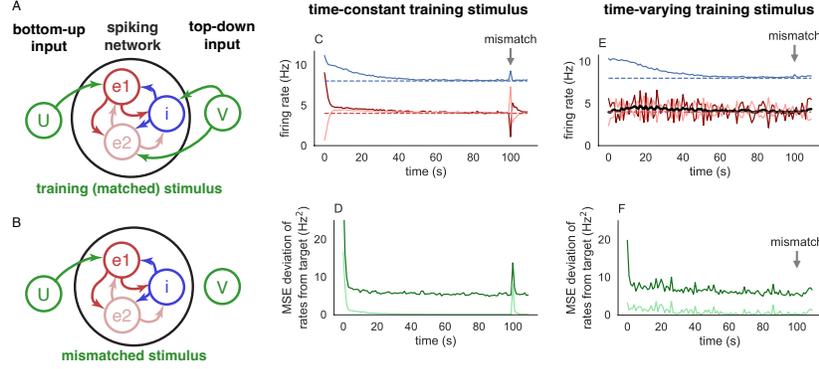
$$\begin{aligned} U &= X_e^0/5 \\ V &= -X_e^0/5. \end{aligned}$$

which is identical to Figure 1 from the main text, but with input  $V$  provided to  $i$  as well. In Supplementary Figure 1E,F, we used

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + c(t)U \\ X_{e_2} &= X_e^0 + c(t)V \\ X_i &= X_i^0 + c(t)V \end{aligned} \right\} \text{matched}$$

and

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + U \\ X_{e_2} &= X_e^0 \\ X_i &= X_i^0 \end{aligned} \right\} \text{mismatched.}$$



Supplementary Figure 1: **Similar results are obtained with input to inhibitory populations. A,B)** Network schematics. Same as Figure 1A,B except top-down external input was provided to the inhibitory population as well. **C,D)** Same as Figure 1C,D except top-down external input was provided to the inhibitory population as well. **E,F)** Same as Figure 5C,D except top-down external input was provided to the inhibitory population as well.

where

$$U = X_e^0/20$$

$$V = -X_e^0/20.$$

This is identical to Figure 5 from the main text, but with input  $V$  provided to  $i$  as well. Our results (Supplementary Figure 1) show a strong mismatch response after training on time-constant input, but not time-varying input.

We additionally tested whether similar results were obtained for distributed external input provided to the inhibitory and excitatory populations (Supplementary Figure 2). Specifically, in Supplementary Figure 2C,D, we used

$$\left. \begin{aligned} X_e &= X_e^0 + \vec{U}_e + \vec{V}_e \\ X_i &= X_i^0 + \vec{U}_i + \vec{V}_i \end{aligned} \right\} \text{matched}$$

and

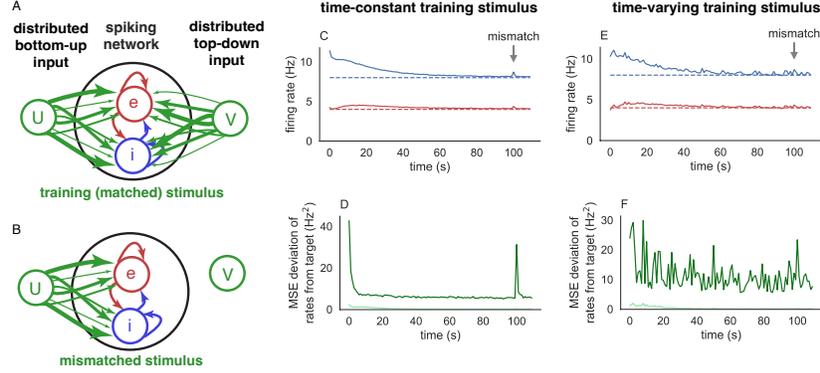
$$\left. \begin{aligned} X_e &= X_e^0 + \vec{U}_e \\ X_i &= X_i^0 + \vec{U}_i \end{aligned} \right\} \text{mismatched.}$$

where  $\vec{U}_a$  and  $\vec{V}_a$  are normally distributed  $N_a$ -dimensional vectors,

$$\vec{U}_a \sim \sigma_s N(0, 1)$$

$$\vec{V}_a \sim \sigma_s N(0, 1)$$

for  $a = e, i$ . This is identical to Figure 4 from the main text except distributed input was provided to



Supplementary Figure 2: **Similar results are obtained with distributed inputs to inhibitory populations.** **A,B)** Network schematics. Same as Figure 4A,B except distributed external input was provided to the inhibitory population as well. **C,D)** Same as Figure 4C,D except top-down external input was provided to the inhibitory population as well. **E,F)** Same as Figure 9C,D except top-down external input was provided to the inhibitory population as well.

the inhibitory population as well. In Supplementary Figure 2E,F, we used

$$\left. \begin{aligned} X_e &= X_e^0 + c(t)\vec{U}_e + c(t)\vec{V}_e \\ X_i &= X_i^0 + c(t)\vec{U}_i + c(t)\vec{V}_i \end{aligned} \right\} \text{matched}$$

and

$$\left. \begin{aligned} X_e &= X_e^0 + \vec{U}_e \\ X_i &= X_i^0 + \vec{U}_i \end{aligned} \right\} \text{mismatched.}$$

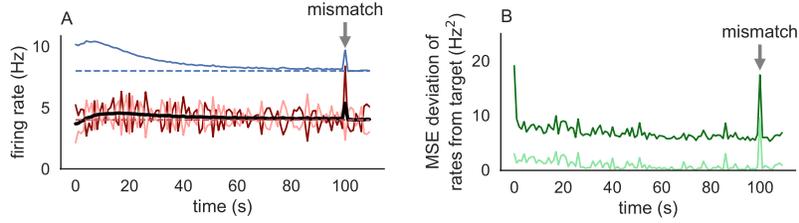
which is identical to Figure 9 from the main text except distributed input was provided to the inhibitory population as well. Our results (Supplementary Figure 2) show a strong mismatch response after training on time-constant distributed input, but not time-varying distributed input.

In conclusion, adding external input to the inhibitory population does not qualitatively affect our overall findings.

## 2 Increasing the strength of mismatched stimuli produces pronounced mismatch responses.

We next repeated the simulation from Figure 5 of the main manuscript, but increased the strength of the mismatched stimulus. In particular, we set

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + c(t)U \\ X_{e_2} &= X_e^0 + c(t)V \\ X_i &= X_i^0 \end{aligned} \right\} \text{matched}$$



Supplementary Figure 3: **Mismatch responses are observed with stronger mismatched stimuli.** **A,B)** Same as Figure 5 except the strength of the mismatched input was increased by six-fold.

and

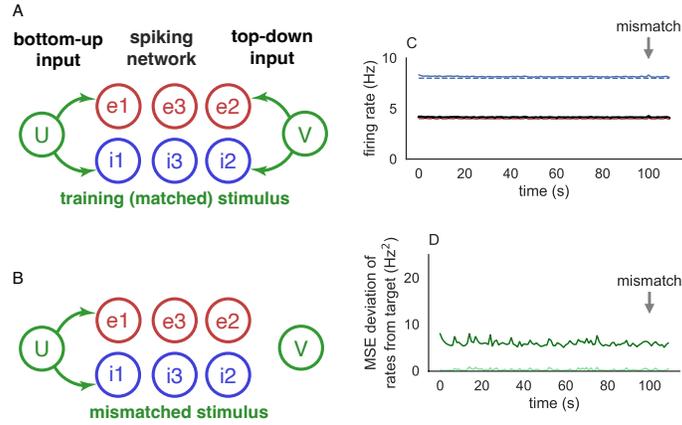
$$\left. \begin{aligned} X_{e_1} &= X_e^0 + 6U \\ X_{e_2} &= X_e^0 \\ X_i &= X_i^0 \end{aligned} \right\} \text{mismatched.}$$

In this case, the mismatched stimulus has a larger magnitude than any of the matched stimuli used for training (in addition to the mismatch that occurs). As predicted, we observed a pronounced mismatch response in this case (Supplementary Figure 3)

### 3 Including several excitatory and inhibitory populations does not change our conclusions.

In all examples considered so far, we considered a single inhibitory population and one or two excitatory populations. We next tested whether including more populations would affect our results. Specifically, we repeated the simulations from Figure 5, but we broke the excitatory and inhibitory populations each into three subpopulations (Supplementary Figure 4). During training (matched stimuli), populations  $e_1$  and  $i_1$  received bottom-up input from  $U$ , populations  $e_2$  and  $i_2$  received top-down input from  $V$ . And populations  $e_3$  and  $i_3$  received no external input. Specifically,

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + c(t)U \\ X_{e_2} &= X_e^0 + c(t)V \\ X_{e_3} &= X_e^0 \\ X_{i_1} &= X_i^0 + c(t)U \\ X_{i_2} &= X_i^0 + c(t)V \\ X_{i_3} &= X_i^0 \end{aligned} \right\} \text{matched}$$



Supplementary Figure. 4: **Mismatch responses are not observed when more populations are considered.** Same as Figure 5 except more populations were added. Connections between populations are not shown for simplicity of the diagram.

and

$$\left. \begin{aligned} X_{e_1} &= X_e^0 + U \\ X_{e_2} &= X_e^0 \\ X_{e_3} &= X_e^0 \\ X_{i_1} &= X_i^0 + U \\ X_{i_2} &= X_i^0 \\ X_{i_3} &= X_i^0 \end{aligned} \right\} \text{mismatched.}$$

Our results (Supplementary Figure 4C,D) shows no visible mismatch response, consistent with our original findings from Figure 5. Hence, simply adding more populations does not change our overall findings. This is consistent with the conclusions reached by our theoretical arguments in the main text.