

Originally published 10 November 2011; corrected 15 December 2011



www.sciencemag.org/cgi/content/full/science.1211095/DC1

Supporting Online Material for

Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks

T. P. Vogels,* H. Sprekeler,* F. Zenke, C. Clopath, W. Gerstner

*To whom correspondence should be addressed. E-mail: tim.vogels@epfl.ch

Published 10 November 2011 on *Science* Express
DOI: 10.1126/science.1211095

This PDF file includes:

- Materials and Methods
- SOM Text
- Figs. S1 to S11
- Tables S1 and S2
- References
- Appendix

Correction: The axes for Fig. S2, A and B, have been corrected.

Supporting Online Material
for

Inhibitory Plasticity Balances Excitation and Inhibition
in Sensory Pathways and Memory Networks

T.P. Vogels^{1*†}, H. Sprekeler^{1*}, F. Zenke¹, C. Clopath^{1,2}, and W. Gerstner¹

*These authors contributed equally to this work.

†To whom correspondence should be addressed; E-mail: tim.vogels@epfl.ch.

This PDF file includes

Materials and Methods
Mathematical Analysis
Additional Simulation Results
Figs. S1 to S11
References
Tables 1 and 2
Technical Appendix

Supporting Online Material

Contents

1 Simulation Methods	3
1.1 Neuron Model	3
1.2 Inhibitory Synaptic Plasticity	3
1.3 Single Cell Simulations	4
1.3.1 Input Signals & Synapse Tuning	4
1.3.2 Parameter Values	5
1.3.3 Correlation Measure	5
1.4 Recurrent Network Simulations	5
1.4.1 Memory Pattern	6
1.4.2 Inhibitory Synaptic Plasticity	6
1.4.3 Network Dynamics	6
1.4.4 Annotated Protocol for Figure 4 & S4	7
2 Mathematical Analysis of Inhibitory Synaptic Plasticity	8
2.1 Assumptions	8
2.1.1 Network Architecture	8
2.1.2 Neuron Model	9
2.1.3 STDP Model	10
2.2 From STDP to Rate-Based Learning	10
2.3 Energy Function	11
2.4 Rate Stabilization	11
2.5 Current Balance	12
2.6 Stimulus Co-Tuning	12
2.7 Effect of Delays	13
2.8 Discussion	15
2.9 Theory Appendix: Proof that the learning rule is a gradient descent.	16
3 Supplementary Simulation Results & Discussion	17
3.1 Single Cell Simulations	17
3.1.1 Uncorrelated Signals	17
3.1.2 Delayed Signals	18
3.1.3 Convergence Times	18
3.1.4 Sparse Responses	19
3.2 Network Simulations	21
3.2.1 Additional Network Statistics	21
3.2.2 Additional Recall Protocols	22
3.2.3 Stability and Parameter Robustness	22
4 References	26
5 Technical Appendix	28
5.1 Tabular Parameter Summaries for the Network Simulations	28
5.2 MATLAB [®] Code	30
5.2.1 Input Signals	30
5.2.2 Single Cell Simulations	35

1 Simulation Methods

1.1 Neuron Model

The model used in all our simulations is a leaky integrate-and-fire neuron, characterized by a time constant, $\tau = 20\text{ ms}$, and a resting membrane potential, $V_{\text{rest}} = -60\text{ mV}$. Whenever the membrane potential crosses a spiking threshold of $\Theta = -50\text{ mV}$, an action potential is generated and the membrane potential is reset to the resting potential, where it remains clamped for a $\tau_{\text{ref}} = 5\text{ ms}$ refractory period. To set the scale for currents and conductances in the model, we use a membrane resistance of $100\text{ M}\Omega$ ($g^{\text{leak}} = 10\text{ nS}$).

We model synapses onto each neuron as conductances, so the sub-threshold membrane voltage obeys

$$\tau \frac{dV_i}{dt} = (V^{\text{rest}} - V_i) + (g_i^E(V^E - V_i) + g_i^I(V^I - V_i) + I_b) \times \frac{1}{g^{\text{leak}}}. \quad (1)$$

Reversal potentials are $V^E = 0\text{ mV}$ and $V^I = -80\text{ mV}$. For the single cell simulations for Fig. 1, 2, & 3, I_b was set to 0 pA ; in the network simulations for Fig. 4 a constant current $I_b = 200\text{ pA}$ was used to maintain a minimum amount of activity.

When the neuron i receives a presynaptic action potential from neuron j , the appropriate postsynaptic conductance is increased, $g_i^E \rightarrow g_i^E + \Delta g_{ij}^E$ for an excitatory spike and $g_i^I \rightarrow g_i^I + \Delta g_{ij}^I$ for an inhibitory spike. Otherwise, these parameters obey the equations

$$\tau_E \frac{dg_i^E}{dt} = -g_i^E \quad \text{and} \quad \tau_I \frac{dg_i^I}{dt} = -g_i^I, \quad (2)$$

with synaptic time constants $\tau_E = 5\text{ ms}$ and $\tau_I = 10\text{ ms}$. The conductance of each synapse is constructed such that $\Delta g_{ij} = \bar{g}W_{ij}$ where \bar{g} is a constant. W_{ij} can be plastic or fixed, depending on the identity of the synapse (see below). The integration time step for our simulations was 0.1 ms . All simulations were programmed in C. For convenience we have attached a slightly modified, annotated matlab script that qualitatively reproduces the findings in Fig. 1.

1.2 Inhibitory Synaptic Plasticity

Recent experimental results (17–19, 31–33) show that inhibitory synapses can be modified by coincident pre- and postsynaptic activity within a coincidence time window τ_{STDP} . Additionally, sole presynaptic spikes lead to a reduction of synaptic efficacy. For the sake of simplicity and in accordance with theoretical results presented below, we model this behavior by a symmetric spike-timing dependent learning rule between a presynaptic neuron j and a postsynaptic neuron i . Potentiation occurs as a function of $\Delta t = |t_j^f - t_i^f|$ (in which t_j^f and t_i^f denote the time of a pre- and postsynaptic spike respectively). Depression occurs at the moment of each presynaptic spike by a fixed amount α . This spike-timing dependent plasticity (STDP) rule was implemented for inhibitory synapses projecting onto excitatory cells. In order to calculate the changes to each W_{ij} , a synaptic trace x_i is assigned to each neuron (or spike train, in case of the single cell simulations for Fig. 1, 2, & 3). x_i increases with each spike $x_i \rightarrow x_i + 1$ and decays otherwise, following

$$\tau_{\text{STDP}} \frac{dx_i}{dt} = -x_i, \quad (3)$$

with the time constant $\tau_{\text{STDP}} = 20 \text{ ms}$. The synaptic weight W_{ij} from neuron j to neuron i is updated for every pre- or postsynaptic event such that

$$W_{ij} \rightarrow W_{ij} + \eta(x_i - \alpha) \quad \text{for presynaptic spikes at time } t_j^f \quad (4)$$

$$\text{and } W_{ij} \rightarrow W_{ij} + \eta x_j \quad \text{for postsynaptic spikes at time } t_i^f \quad (5)$$

where η is the learning rate and $\alpha = 2 \times \rho_0 \times \tau_{\text{STDP}}$ is the depression factor, and ρ_0 is a constant parameter with units 1/time (see also theoretical analysis below).

1.3 Single Cell Simulations

1.3.1 Input Signals & Synapse Tuning

In the first part of the paper, we model the arrival of multiple signals at a single integrate-and-fire cell. To mimic sensory input during learning, we constructed 8 independent traces of low-pass filtered, half-wave rectified white noise signals (13, 16, 34). These input signals $s_k(t)$, where $k = \{1 \dots 8\}$ denotes signal identity, were designed as follows:

The raw signal \hat{s}_k was drawn from $\hat{s}_k(t + dt) = \xi - (\xi - \hat{s}_k(t)) \times e^{-\frac{dt}{\tau_s}}$, where ξ was a random number drawn from a uniform distribution $\xi \in [-0.5 \dots 0.5]$, $dt = 0.1 \text{ ms}$ is the simulation time step, and the filter time constant was set to $\tau_s = 50 \text{ ms}$. We then rectified \hat{s}_k (by setting all negative values to zero), and normalized it to a peak value of $500 \text{ Hz} \times dt (= 0.05)$.

By design, \hat{s}_k is a signal that is “ON” about 50 % of the time, leading to average pairwise co-activity with a second signal \hat{s}_l ,

$$\Omega_{kl} = \frac{\hat{s}_k \times \hat{s}_l}{\sqrt{\hat{s}_k^2 \times \hat{s}_l^2}} = 0.5. \quad (6)$$

In order to reduce the number of simultaneously active signals and thus accelerate the establishment of co-tuned inhibition without having to grossly increase the learning rate η , we sparsified each signal by deleting roughly every second activity bump, leading to co-activity values of $\Omega_{kl} = 0.25$ for the sparser signal s_k (for details see example code in the technical appendix).

Finally, s_k was added to a constant value of $5 \text{ Hz} \times dt (= 0.0005)$, to provide background activity of 5 Hz, and used as a spiking probability for Poisson processes to create 100 excitatory and 25 inhibitory spike trains for each input signal. This led to a total of 1000 independently created spike trains (800 excitatory and 200 inhibitory), divided into 8 groups that shared the same input signal s_k . The normalization of input signal traces as discussed above and the resulting maximum spike probability were set to achieve average presynaptic firing rates of 13 Hz with peak firing rates at $\sim 150 \text{ Hz}$ for each spike train. To avoid bursting in each spike train we implemented an absolute refractory period of 5 ms once a spike was initiated. Example code in MATLAB® (35) can be found in the technical appendix below.

The resulting input spike trains were delivered to the neuron through synapses with varying strengths $\Delta g_{ij} = \bar{g}W_{ij}$. Excitatory synapses were set to produce tuned input currents with tuning shapes as reported elsewhere (7). To this end we set $\bar{g}^E = 140 \text{ pS}$, and adjusted the weight factor W_{ij} to obey the (arbitrary) function $W_{ij} = 0.3 + \frac{1.1}{(1+(K(j)-P))^4} + \xi$, where $K \in \{1 \dots 8\}$

is the group index of the presynaptic synapse j , $\xi \in [0...0.1]$ is a noise term, and $P = 5$ is the position of the peak of the tuning curve, i.e the signal with the strongest synapses. This led to synaptic weights ranging from 42 pS to 210 pS, with an average weight of $\langle \Delta g_{ij}^E \rangle = 90$ pS, and resulted in PSP amplitudes on the order of ~ 0.1 mV ($[0.04\text{mV}...0.2\text{mV}], \emptyset 0.08\text{mV}$) at V^{rest} .

Inhibitory synaptic conductances were initially set to $\bar{g}^I = 350$ pS, so that with $W_{ij} = 1$, a single inhibitory event $\Delta g_{ij}^I = \bar{g}W_{ij}$ produced -0.18 mV IPSPs at V^{rest} . At the beginning of a simulation, we set $W_{ij}^I = 0.1$, resulting in dramatically underpowered inhibition.

Unless otherwise noted, the parameters above were left constant throughout all single cell simulations. In places, we used the same realization of the input signals s_k for multiple trials (for averaging, Fig. 2 A, B). We also used step-like stimuli with step sizes between 5 Hz and 50 Hz to map the receptive field properties (Fig. 2 C-H).

1.3.2 Parameter Values

The variables that determine the dynamics of inhibitory synaptic plasticity ($\eta = 10^{-4}$, the learning rate, and $\alpha = 0.2$ ($\rho_0 = 5$ Hz), the depression factor) were kept fixed unless otherwise noted. To reproduce the temporal dynamics of the experiment (7) discussed in Fig. 3, we set $\eta = 10^{-6}$, $\alpha = 0.35$ ($\rho_0 = 8.75$ Hz), and increased the average firing rate of the inhibitory spike trains 4 fold in comparison to their excitatory counterparts from 13 Hz to approximately 50 Hz, to adjust the temporal dynamics of depression and potentiation relative to each other. In those simulations, we waited until detailed balance was established by the plasticity mechanism and then switched P , the peak position of the excitatory tuning curve, from $P = 5$ to $P = 3$.

1.3.3 Correlation Measure

We used a correlation measure (11, 13, 34) to determine the impact of each input signal on the output. To do this, we calculated firing rate histograms $r(t)$ (PSTH bin size = 5 ms) of the output signals over 100 trials with identical input signals and determined its time-averaged firing rate \bar{r} . The correlation of the histogram with the input signal k is

$$C_k = \frac{\langle (s_k(t) - \bar{s}_k)(r(t) - \bar{r}) \rangle_t}{\sigma_{s_k} \sigma_r}, \quad (7)$$

where the brackets denote an average over time, $s_k(t)$ and \bar{s}_k are the firing rate and its average for a given input signal, and σ_r, σ_{s_k} are the standard deviations of the corresponding firing rates. We define the impact of each signal as C_k^2 , which can be interpreted as the total amount of variance of the output firing pattern that can be explained by each input signal.

1.4 Recurrent Network Simulations

In the last part of the paper we studied the effect of inhibitory synaptic plasticity in a large recurrent network. To this end, we simulated a network of 8,000 excitatory and 2,000 inhibitory leaky integrate-and-fire neurons, laid out on a 100×100 grid. All neurons have random con-

nectivity of 2% to the rest of the network. The connectivity remains fixed during a simulation, although the weights of the inhibitory synapses onto excitatory cells can change according to inhibitory plasticity (see below). We chose the specific network because of its publication history (13, 36) and because it is small enough to be studied within reasonable computation times but large enough to avoid boundary effects (37). Network parameters were chosen in keeping with both general properties of cortical circuits and previous work (13, 34, 37, 38) and can be found in tabular summaries following the standard form (39, 40) below.

1.4.1 Memory Pattern

In addition to the general architecture, we introduced specific patterns into the weight matrix by defining two groups of $28 \times 28 = 784$ excitatory neurons as Hebbian assemblies. We strengthened all existing connections between the neurons within each group by a factor of five. We allowed the patterns to overlap by selecting $8 \times 8 = 64$ neurons to be part of both groups. The synaptic weights between two neurons that both belonged to both patterns was increased by a factor of 5 only once. Additionally we defined a third, control assembly of neurons which did not take part in either pattern. The strength of intra-group synapses of this third pattern remained un-strengthened.

1.4.2 Inhibitory Synaptic Plasticity

We can distinguish between excitatory to excitatory, excitatory to inhibitory, inhibitory to inhibitory and inhibitory to excitatory connections in our network. Only the latter group is plastic. For simplicity, we assume that the structure of the connectivity matrix remains fixed after the network has been initialized. Particularly this means that we will restrict inhibitory plasticity operations to already existing connections. Note however that the weight of an existing connection W_{ij} can decay to zero. For the simulations shown here, we set $\eta = 10^{-4}$ and $\alpha = 0.12$ ($\rho_0 = 3$ Hz), unless stated otherwise.

1.4.3 Network Dynamics

To characterize the global state of the network (Fig. S2) we monitored individual spike trains, the population firing rate (the average of firing rates across the network), and the population rate's standard deviation σ_{Rate} , as well as average membrane potentials, and interspike intervals (ISIs). The irregular asynchronous network activity that is thought to mimic cortical dynamics has a roughly constant population firing rate with low spiking correlation values (4) and coefficients of variation of the interspike intervals (ISI CVs) near 1. The ISI CV for a neuron is the ratio of the standard deviation of the ISI distribution and its mean. ISI CV values close to zero indicate regular spiking patterns, values near 1 indicate irregular spiking, and values larger than 1 indicate, in our simulations, burstiness in the firing pattern (1, 41).

Additionally, we calculated distributions of the spiking correlations (4) and the ISI CVs of neurons in two groups: We collected data from 392 neurons within one memory pattern and an equal number of cells from the control group. ISI CV histograms were calculated as above.

Following Renart et al. (4) we computed the spiking correlation coefficient X_{ij} between spike trains $S_i(t)$ and $S_j(t)$. We first constructed filtered spike trains F_i defined as

$$F_i(t) = S_i(t) * K(t), \quad (8)$$

in which the spike train $S_i = \sum_f \delta(t - t_i^f)$ is convolved with a symmetric bi-exponential kernel $K(t)$ (with $\int_{-\infty}^{\infty} K(t)dt = 0$) defined as

$$K(t) = \frac{1}{\tau_1} \exp\left(-\frac{|t|}{\tau_1}\right) - \frac{1}{\tau_2} \exp\left(-\frac{|t|}{\tau_2}\right) \quad (9)$$

with $\tau_1 = 50$ ms and $\tau_2 = 4 \times \tau_1$. The unnormalized covariance $V_{ij} = \sum_t F_i(t)F_j(t)$ over all discrete times t then leads to correlation coefficients

$$X_{ij} = \frac{V_{ij}}{\sqrt{V_{ii}V_{jj}}}. \quad (10)$$

To gain insight into ensemble statistics we calculate $\sim 38,000$ random correlation coefficients between the filtered spiketrains of a given group and plotted them in histograms (Fig. 4). All values were computed in discrete time with a resolution of $dt = 1$ ms. The combination of ISI CV and spiking correlations provides a measure of the regularity of individual spike trains and the synchrony of the local network activity as we change the connectivity matrix of the network as described below.

To recall a stored pattern we externally stimulated a subset of neurons within the pattern for 1 second. Stimulated neurons were randomly connected (5% connectivity) to a group of 1000 independent Poisson processes with a mean firing rate of 100 Hz. For all recall experiments we only stimulated a set of neurons that was disjunct to the neurons used for computing the spike train statistics.

1.4.4 Annotated Protocol for Figure 4 & S4

The simulation protocol for Fig. 4 was structured as follows:

Start: $t = -1$ min: The AI network dynamics of the original network (13) without inhibitory plasticity. This phase serves as a reference and is not shown.

4, A: $t = 0$ min: Inhibitory to excitatory synapses are turned to 0 efficacy. The network is forced out of the AI regime and begins to fire at high rates. Simultaneously, inhibitory plasticity is turned on.

4, B: $t = 60$ min: Inhibitory plasticity has restored AI activity.

4, C: $t = 60$ min, 5 s: The excitatory non-zero weights of the 2 designated memory patterns are increased ad-hoc by a factor of 5. The neurons of the subset begin to exhibit elevated and more synchronized activity.

4, D: $t = 120$ min: Inhibitory plasticity has successfully suppressed any elevated activity from the pattern and restored the global background state.

4, E: $t = 120$ min, 5 s: By delivering an additional, 1 s long stimulus as described above to 25% of the cells within one memory pattern, the whole pattern is activated. Activity inside the pattern stays asynchronous and irregular, and the rest of the network, including the other pattern, remains nearly unaffected.

Fig. S4 continues the protocol:

S4, F: $t = 120$ min, 11 s: The other (blue) memory pattern is activated with a stimulus analogous to the one used in Fig. 4 E.

S4, G: $t = 120$ min, 17 s: Both memory patterns respond with elevated AI activity to a stimulus to 25% of the cells, including the cells shared between both patterns.

2 Mathematical Analysis of Inhibitory Synaptic Plasticity

The goal of this analysis is to provide a mathematical background for the findings presented in the main article. To this end, we study a network of neurons with plastic feedforward inhibition (Fig. S1). The analysis is done for linear Poisson neurons, which enables an analytic treatment of most phenomena observed in the simulations. The limitations of the approach are discussed.

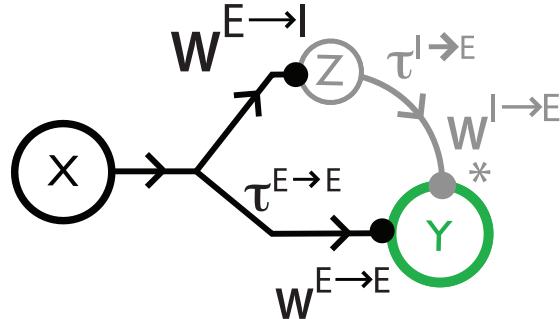


Fig. S1: Overview of the variables used in the theoretical analysis.

An excitatory cell population (in black) with firing rate x delivers postsynaptic currents to an inhibitory (grey) and an excitatory (green) population of neurons. The currents are proportional to the strength of the excitatory synapses $\mathbf{W}^{E \rightarrow I}$ and $\mathbf{W}^{E \rightarrow E}$, respectively. Currents to the excitatory (green) population are delivered with a delay $\tau^{E \rightarrow E}$. Additionally the excitatory target population receives inhibitory input currents with delay $\tau^{I \rightarrow E}$. These currents are the product of z , the firing rate of the inhibitory neuron population, and $\mathbf{W}^{I \rightarrow E}$ the synaptic strength of the inhibitory synapses, here subject to plasticity. Consequently, the firing rate y of the excitatory population is determined by the difference of excitatory and inhibitory currents.

2.1 Assumptions

2.1.1 Network Architecture

We analyze a network consisting of a population of inhibitory interneurons connected to a single output neuron through plastic synapses with a weight vector $\mathbf{W}^{I \rightarrow E}$. All neurons are linear

Poisson neurons. The input to this network consists of a set of time-dependent analog input signals $\mathbf{x}(t)$, which are received by both the output neuron (with a weight vector $\mathbf{W}^{E \rightarrow E}$) and the interneurons (weight matrix $\mathbf{W}^{E \rightarrow I}$). To allow for the possibility that the input signals arrive at the output neuron and the interneuron with different delays, we introduce a (negative or positive) delay $\tau^{E \rightarrow E}$ in the direct excitatory projection from the input to the output neuron that summarizes the difference in the delays along the two pathways. The main simplifications of the analytical model compared to the simulations are (i) the input $\mathbf{x}(t)$ is an analog signal rather than a spike train; (ii) synaptic events are treated as currents rather than as conductance changes; and (iii) the neuron model is linear and Poisson rather than integrate-and-fire. In addition to the simulation results, we show that linear transformations $\mathbf{W}^{E \rightarrow I}$ along the inhibitory feedforward branch of the circuit (Fig. S1) have little impact on the establishment of the balance.

2.1.2 Neuron Model

The interneurons and the output neuron are linear Poisson neurons, i.e. they emit action potentials with inhomogeneous firing rates that are given by a linear weighted sum of their input signals. We denote the spike trains of the output neuron and the interneurons by Y and Z_j , respectively. The spike trains are modeled as sums of δ -pulses: $Z_j(t) = \sum_f \delta(t - t_j^f)$, where t_j^f denote the time of the f -th spike emitted by the j -th interneuron.

We denote the firing rate of the output neuron by ρ^Y and those of the interneurons by z_j :

$$z_j(t) = \sum_i W_{ji}^{E \rightarrow I} x_i(t) \quad (11)$$

$$\begin{aligned} \rho^Y(t) &= \sum_i W_i^{E \rightarrow E} x_i(t - \tau^{E \rightarrow E}) - \sum_{j,f} W_j^{I \rightarrow E} \epsilon(t - t_j^f) \\ &= \sum_i W_i^{E \rightarrow E} x_i(t - \tau^{E \rightarrow E}) - \sum_j W_j^{I \rightarrow E} \int \epsilon(t - t') Z_j(t') dt', \end{aligned} \quad (12)$$

where $\epsilon(t)$ is a kernel (e.g. the inhibitory postsynaptic potential) that describes the time course of the output rate change caused by spike arrival at the synapse from an interneuron. For simplicity, we assume that ϵ is normalized such that its integral is equal to one: $\int \epsilon(t) dt = 1$.

For later use, let us also introduce the firing rate of the output neuron that arises after taking the ensemble average over the activity of the interneurons for a given set of input signals:

$$y(t) := \sum_i W_i^{E \rightarrow E} x_i(t - \tau^{E \rightarrow E}) - \sum_j W_j^{I \rightarrow E} \int \epsilon(t - t') z_j(t') dt' \quad (13)$$

and the correlation between the output neuron and a spike at the interneuron j :

$$c(t|t_j^f) = y(t|t_j^f) - y(t) = -W_j^{I \rightarrow E} \epsilon(t - t_j^f). \quad (14)$$

2.1.3 STDP Model

As shown in electrophysiological work in cultures and acute slice (17–19, 31–33), GABAergic synapses can be modified in a Hebbian manner by near-coincident activation of the pre- and postsynaptic neuron, with a coincidence time window of about 20ms duration. Moreover, presynaptic spikes alone induce a reduction of synaptic efficacy. We model this behavior by a spike-timing dependent (STDP) learning rule, in which the weight change of an inhibitory synapse from interneuron j to an excitatory postsynaptic neuron within a given time window of duration T is determined by:

$$\Delta W_j^{I \rightarrow E} = \eta \int_0^T \int_0^T L(t - t') Y(t') Z_j(t) dt dt' - \eta \rho_0 \int_0^T Z_j(t) dt, \quad (15)$$

where $L(t) = [2\tau_{\text{STDP}}]^{-1} e^{-|t|/\tau_{\text{STDP}}}$ denotes a symmetric learning window with a coincidence time τ_{STDP} , η is a learning rate and ρ_0 is a constant that controls the relative strength of the non-Hebbian weight decrease in relation to the Hebbian weight increase. The different normalization of the learning window in eq. 15 to that in eq. 4 and 5 is compensated by a rescaling of the learning rate η .

2.2 From STDP to Rate-Based Learning

To derive a rate-based learning rule from the STDP rule we consider the weight change in Eq. 15 and take the ensemble average over the activity of the output neuron Y and the interneurons Z , given the input signals \mathbf{x} (42):

$$\langle \Delta W_j^{I \rightarrow E} \rangle_{Y,Z|X} = \eta \iint L(t - t') \langle \langle Y(t) \rangle_{Y|X, Z} Z_j(t') \rangle_{Z|X} dt dt' \quad (16)$$

$$= \eta \iint L(t - t') \langle \rho^y(t) Z_j(t') \rangle_{Z|X} dt dt' - \eta \rho_0 \int z_j(t) dt. \quad (17)$$

Using that the spike trains Z_j arise from inhomogeneous Poisson processes with a correlation function $\langle Z_j(t) Z_k(t') \rangle_{Z|X} = z_j(t) z_k(t') + \delta(t - t') \delta_{jk} z_j(t)$, Eq. 17 can be simplified to

$$\langle \Delta W_j^{I \rightarrow E} \rangle_{Z,Y|X} = \eta \iint L(t - t') y(t) z_j(t') dt dt' - \eta(\rho_0 + W_j^{I \rightarrow E} \rho_s) \int z_j(t) dt, \quad (18)$$

with $\rho_s := \int L(\tau) \epsilon(\tau) d\tau$.

To simplify this expression further, we assume that the characteristic time scale of the correlation functions $c_j(s) = \frac{1}{T} \int y(t) z_j(t + s) dt$ is slower than the coincidence time τ_{STDP} of the learning window. In this case, we can perform a Taylor expansion of the correlation function around

$s = 0$ and neglect terms of order s or higher:

$$\iint L(t-t')y(t)z_j(t')dt dt' = T \int L(s)c_j(s)ds \quad (19)$$

$$= T \int L(s) (c_j(0) + c'_j(0)s + \dots) ds \quad (20)$$

$$\approx \int L(s)ds \int y(t)z_j(t)dt \quad (21)$$

$$= \int y(t)z_j(t)dt, \quad (22)$$

where in the last line we used that the integral over the double-exponential learning window is normalized.

Inserting this expression into Eq. 18 leads to the following simplified expression for the weight dynamics:

$$\langle \Delta W_j^{I \rightarrow E} \rangle_{Z,Y|X} = \eta \int (y(t)z_j(t) - (\rho_0 + \rho_s W_j^{I \rightarrow E})z_j(t)) dt. \quad (23)$$

2.3 Energy Function

The simulations suggest that the learning rule tries to balance excitation and inhibition at any given moment in time and that the neuron aims to maintain a given target firing rate. These findings can be substantiated by a mathematical analysis.

Let us for a moment neglect the delay that is introduced by the synaptic kernel at the inhibitory synapses and set $\epsilon(t) = \delta(t)$. In this case, it can be shown that the learning rule Eq. 23 performs a gradient descent on the following energy function:

$$\Psi(\mathbf{W}^{I \rightarrow E}) = \frac{1}{2} \langle (y(t) - \rho_0)^2 \rangle_t + \frac{1}{2} \rho_s \sum_j \bar{z}_j(t) (W_j^{I \rightarrow E})^2, \quad (24)$$

where $\langle \cdot \rangle_t$ denotes a temporal average and $\bar{z}_j = \langle z_j(t) \rangle_t$ is the mean firing rate of interneuron j .

We defer the proof that a gradient descent on this energy function reproduces the learning rule Eq. 23 to section 2.9. Instead we discuss the two terms of the objective function Ψ . The second term, which we denote by Ψ_{spike} in following, arises from spike-spike correlations and plays the role of a cost term that punishes large inhibitory weights for synapses that are active. The effect of the first term is discussed in the following.

2.4 Rate Stabilization

The first term of the energy function measures the quadratic deviation of the output firing rate from ρ_0 . Therefore, the constant ρ_0 acts as a target firing rate. The learning rule thus implements a form of homeostatic plasticity that stabilizes the postsynaptic firing rate (43). This is reflected by the simulations, which show that the postsynaptic firing rate after convergence depends linearly on the strength ρ_0 of the non-Hebbian synaptic depression.

For large numbers of inhibitory synapses and/or high firing rates of the inhibitory neurons, spike-spike correlations can be neglected. The output firing rate is then simply given by $y = \rho_0$. This theoretical prediction of the firing rate fits the simulations well (see Fig. 1 G).

The fact that the firing rate is controlled by a single parameter is advantageous for simulation studies, because it allows to automatically tune a recurrent network to a desired firing rate by simply choosing the appropriate parameters for the learning rule.

2.5 Current Balance

The firing rate of the linear output neurons is given by the difference between excitatory input $E(t) = \sum_i W_i^{E \rightarrow E} x_i(t - \tau^{E \rightarrow E}) = \mathbf{W}^{E \rightarrow E} \cdot \mathbf{x}(t - \tau^{E \rightarrow E})$ and inhibitory input $\sum_j W_j^{I \rightarrow E} z_j = \mathbf{W}^{I \rightarrow E} \cdot \mathbf{z}(t)$. Therefore, the first term of the energy function Ψ measures a quadratic error in the balance between excitation and inhibition, corrected by the target firing rate ρ_0 :

$$\Psi = \underbrace{\langle (E(t) - \rho_0)^2 \rangle_t}_{\tilde{E}(t)} + \Psi_{spike} \quad (25)$$

$$= \langle (\tilde{E}(t) - \mathbf{W}^{I \rightarrow E} \cdot \mathbf{z}(t))^2 \rangle_t + \Psi_{spike} \quad (26)$$

In our simulations, the target firing rate is smaller than the excitatory drive: $\rho_0 < E(t)$ (i.e. in the absence of inhibition, the neurons fire at much higher rates than the target rate ρ_0). Therefore, the subtraction of the target rate can be seen as a relatively small correction of the excitatory drive: $\tilde{E}(t) \approx E(t)$. Then, the first term of the energy function measures the mean square difference between the excitatory and the inhibitory input to the output cell. Minimizing this term corresponds to balancing excitation and inhibition for any given moment in time. Moreover, because the inhibitory input is linear in the inhibitory weights, minimizing the first part of the objective function is mathematically equivalent to a linear regression with the inhibitory weights as parameters and the excitatory input \tilde{E} as a target.

2.6 Stimulus Co-Tuning

Intuitively, it is clear that a detailed balance between excitation and inhibition can only be reached if stimulus-evoked excitatory and inhibitory currents are balanced on the level of individual stimuli, i.e. that excitation and inhibition are co-tuned in terms of stimulus selectivity.

To find a mathematical formalization of this intuition, let us assume that the input neurons are ordered according to their stimulus preference along an arbitrary stimulus dimension (e.g. auditory frequency, visual orientation). The excitatory weights $\mathbf{W}^{E \rightarrow E}$ determine the stimulus tuning curve for the excitatory input the output neuron receives.

The expected inhibitory input $I(t)$ the output neuron receives is determined by the indirect propagation of the input activity $x_i(t)$ via the interneurons:

$$I(t) = \mathbf{W}^{I \rightarrow E} \cdot \mathbf{z}(t) = \mathbf{W}^{I \rightarrow E} \cdot \mathbf{W}^{E \rightarrow I} \mathbf{x}(t) \quad (27)$$

$$= ((\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E}) \cdot \mathbf{x}(t). \quad (28)$$

The stimulus tuning of the inhibitory input is thus determined by the product $(\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E}$ of the weights along the indirect inhibitory pathway. Whether excitation and inhibition are co-tuned is therefore determined by the relation of the excitatory weights $\mathbf{W}^{E \rightarrow E}$ and the effective inhibitory weight vector $(\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E}$.

A perfect balance can only be reached if the information that is propagated along the direct excitatory pathway is available in the activity of the interneurons. If this is not the case, the balance must remain approximate. A way to see if and how the learning rule Eq. 23 tries to approximate the perfect co-tuning is to rewrite the energy function Eq. 24 in terms of the difference of the excitatory and the inhibitory weights. To this end, let us first split the first term of the energy into mean and variance:

$$\Psi = \langle (y - \bar{y})^2 \rangle_t + (\bar{y} - \rho_0)^2 + \Psi_{spike}. \quad (29)$$

If we neglect the delay $\tau^{E \rightarrow E}$ on the direct excitatory pathway for a moment, the output rate is linear in the input and depends on the difference $\Delta := \mathbf{W}^{E \rightarrow E} - (\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E}$ between the excitatory and the cumulative inhibitory weights:

$$y(t) = \sum_i (W_i^{E \rightarrow E} - \sum_j W_j^{I \rightarrow E} W_{ji}^{E \rightarrow I}) x_i(t) = \Delta \cdot \mathbf{x}(t). \quad (30)$$

Inserting this into the energy function yields:

$$\Psi = \Delta^T \mathbf{C} \Delta + (\Delta \cdot \bar{\mathbf{x}} - \rho_0)^2 + \Psi_{spike}, \quad (31)$$

where $\bar{\mathbf{x}} = \langle \mathbf{x} \rangle_t$ denotes the mean input rate and $\mathbf{C} := \langle (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \rangle_t$ is the covariance matrix of the input.

The second term of this energy function punishes deviations from the *global* balance, i.e. it aims at a balance of excitation and inhibition *on average*. The first term is a positive semi-definite quadratic form in the difference of excitation and inhibition. It therefore punishes deviations from a more precise *detailed* balance, i.e. imbalances between excitation and inhibition on the level of individual inputs. The covariance matrix \mathbf{C} of the input introduces a weighting: imbalances along dimensions of high input variance are punished more severely than along low input variance.

The reformulation of the energy function in terms of the excitatory and inhibitory weights shows that inhibitory plasticity seeks to establish a co-tuning of excitation and inhibition. If information is lost along the indirect pathway, the system will establish an approximation that minimizes the squared deviation from the balance, weighted according to the variance in the input.

2.7 Effect of Delays

The analysis presented in the last paragraphs neglects transmission and conduction delays. The effect of delays is twofold. Firstly, they can make a balance between of excitatory and inhibitory input to the output neuron impossible, because the two signals arrive at different moments in time. Secondly, they can disturb Hebbian learning, because the correlations between pre- and postsynaptic activity are weakened.

In the presence of delays, there is no energy function for the system dynamics. To identify under which conditions inhibitory plasticity can establish a balance between excitation and inhibition in the presence of delays, we have to consider the stationary solution of the learning dynamics Eq. 23. For clarity, we neglect the term that arises from spike-spike correlations, i.e. we assume that $\rho_s W_j^{I \rightarrow E} \ll \rho_0$. The fixed point of eq. 23 is then:

$$\Delta \mathbf{W}^{I \rightarrow E} = \eta \int [y(t) \mathbf{z}(t) - \rho_0 \mathbf{z}(t)] dt = 0, \quad (32)$$

and summarize the effect of the synaptic kernel ϵ at the inhibitory synapse into a synaptic delay:

$$y(t) = \mathbf{W}^{E \rightarrow E} \cdot \mathbf{x}(t - \tau^{E \rightarrow E}) - \mathbf{W}^{I \rightarrow E} \cdot \mathbf{z}(t - \tau^{I \rightarrow E}). \quad (33)$$

By inserting Eqs. 11 and 33 into the stationarity condition Eq. 32, we get

$$\mathbf{W}^{E \rightarrow I} \mathbf{C}(\tau^{E \rightarrow E}) \mathbf{W}^{E \rightarrow E} - \mathbf{W}^{E \rightarrow I} \mathbf{C}(\tau^{I \rightarrow E}) (\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E} - \rho_0 \mathbf{W}^{E \rightarrow I} \bar{\mathbf{x}} = 0, \quad (34)$$

where $\mathbf{C}(\tau) := \langle \mathbf{x}(t) \mathbf{x}^T(t - \tau) \rangle_t$ denotes the time-delayed correlation matrix of the input signals.

Because we are now interested merely in whether delays can disrupt the balance, we assume that the full input information is present in the inhibitory population, i.e. that the weight matrix $\mathbf{W}^{E \rightarrow I}$ has full rank. Then, Eq. 34 can be resolved for the effective inhibitory weights:

$$(\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E} = \mathbf{C}^{-1}(\tau^{I \rightarrow E}) \mathbf{C}(\tau^{E \rightarrow E}) \mathbf{W}^{E \rightarrow E} - \rho_0 \mathbf{C}^{-1}(\tau^{I \rightarrow E}) \bar{\mathbf{x}}. \quad (35)$$

The first observation that is consistent with our previous statements on the stimulus co-tuning is that when the delay along the indirect inhibitory pathway is the same as on the direct excitatory pathway $\tau^{I \rightarrow E} = \tau^{E \rightarrow E}$, the product $\mathbf{C}^{-1}(\tau^{I \rightarrow E}) \mathbf{C}(\tau^{E \rightarrow E})$ is reduced to the unit matrix. The effective inhibitory weights $(\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E}$ are then the same as the excitatory weights $\mathbf{W}^{E \rightarrow E}$, apart from a correction that depends on ρ_0 and maintains the target firing rate.

To get a clearer picture on the effect of delays, let us assume that different input signals are decorrelated and that their autocorrelation function decays exponentially, i.e. that the time-delayed correlation matrix $\mathbf{C}(\tau)$ of the input signals has the following simple structure:

$$\mathbf{C}(\tau) = \bar{\mathbf{x}} \bar{\mathbf{x}}^T + \sigma^2 e^{-|\tau|/\tau_c} \mathbf{E}, \quad (36)$$

where σ and τ_c denote the variance and autocorrelation time of the inputs and \mathbf{E} is the unit matrix.

With this assumption, the matrix $\mathbf{C}(\tau)$ can be inverted analytically and the effective inhibitory weights can be written as a linear combination of the excitatory weights and the (untuned) mean firing rates of the input neurons:

$$(\mathbf{W}^{E \rightarrow I})^T \mathbf{W}^{I \rightarrow E} = A \mathbf{W}^{E \rightarrow E} + B \bar{\mathbf{x}}, \quad (37)$$

where A and B are the following expressions that depend on the delays in the system

$$A = e^{\frac{|\tau^{I \rightarrow E}| - |\tau^{E \rightarrow E}|}{\tau_c}} \quad (38)$$

$$B = \frac{(1 - A) \mathbf{W}^{E \rightarrow E} \cdot \bar{\mathbf{x}}}{|\bar{\mathbf{x}}|^2 + \sigma^2 e^{-|\tau^{I \rightarrow E}|/\tau_c}} - \frac{\rho_0}{|\bar{\mathbf{x}}|^2 + \sigma^2 e^{-|\tau^{I \rightarrow E}|/\tau_c}}. \quad (39)$$

The dependence of the factor A on the delays is worth discussing, because it determines the strength of the inhibitory tuning and because it qualitatively captures the effects observed in the simulations (see also Additional Simulation Results, below).

The key to understanding how A depends on the delays lies in the Hebbian learning rule that controls the inhibitory weights. Because the activity of the output neuron is the difference between excitation and inhibitory inputs, inhibition is strengthened by correlations between the activity of the interneurons and the excitatory drive to the output neuron, while correlations between the activity of the interneurons and the inhibitory drive to the output neuron decrease the weights. The correlation between the activity of the interneurons and the excitatory drive to the output neuron decreases with the difference in the delays along the two excitatory pathways. Therefore, inhibition is weakened with increasing $|\tau^{E \rightarrow E}|$. Conversely, the delay along the inhibitory pathway decreases the correlation between the activity of the interneurons and the inhibitory drive to the output neuron. Because this correlation limits the growth of the inhibitory weights, the inhibitory weights grow as the delay $|\tau^{I \rightarrow E}|$ increases. The sensitivity of the effective inhibitory weights on the delays depends on the autocorrelation time of the input signals, because this is the time scale that controls by how much the signals can be delayed in time, before the correlation along the respective pathways is lost. Note here also that changing the inhibitory delay between presynaptic spike firing and spike arrival at the synapse for a learning rule that depends on the presynaptic firing time is the same as changing the peak of the learning window L in eq. 15 for a learning rule that depends on presynaptic spike arrival time.

The factor B that controls the untuned contribution to the effective inhibitory weights increases as the inhibitory tuning strength A decreases. This effect is qualitatively confirmed by the simulations (Fig. S2 D), although the increase in the untuned inhibition is stronger than theoretically predicted. This is due to the limitations of the rate picture that are discussed below.

2.8 Discussion

The present theoretical treatment shows that the Hebbian learning rule in inhibitory synapses can be interpreted as a gradient descent on the mean square difference between excitation and inhibition. Although the theory is based on a simple linear picture of a network with feedforward inhibition, it is able to capture the key effects that are observed in the simulations: rate homeostasis, current balance, stimulus co-tuning and the gradual loss of co-tuning with increasing delays along the excitatory pathway.

In a system with balanced excitation and inhibition, the difference between excitatory and inhibitory drive to the output neuron fluctuates around zero. Therefore, the output neuron will frequently encounter a net negative drive. The present linear treatment of the system will therefore often assume biologically implausible negative rates of the output neuron. From this perspective, it is surprising that such a simplified picture captures the key effects present in the simulations. Because the inhibitory weights are subject to a Hebbian learning rule, negative output rates decrease the inhibitory weights, while the more realistic integrate-and-fire neuron used in the simulations simply stops firing and thereby evades Hebbian changes in the inhibitory weights. Thus, the theory systematically underestimates the inhibitory weights. This is most evident in the simulations with delays: If inhibition lags behind excitation, the output neuron receives a period of unbalanced positive input whenever the input neurons start firing (onset transient; (5, 11, 44)) and, conversely, a period of negative input every time the input neurons

stop firing. Because of the output rectification of the integrate-and-fire neuron, the Hebbian learning rule “sees” only the onset transient and increases the inhibitory weights. This effect becomes more prominent with increasing delay (Fig. S2 C, D).

The mathematical analysis suggests that the effects that are observed in the simulations are not sensitive to details of the STDP learning rule that is used. As long as the integral of the learning window is positive, the rate-based description of the learning dynamics remains the same. Therefore, asymmetric learning windows, as observed for most excitatory synapses, would most likely not change the results qualitatively, as long as the LTP component dominates over LTD.

2.9 Theory Appendix: Proof that the learning rule is a gradient descent.

We assume that synaptic transmission at the inhibitory synapse is fast compared to the auto-correlation time of the inhibitory rate variations, so that we can replace the synaptic kernel ϵ by a δ -function. Then, the output firing rate $y(t)$ is given by

$$y(t) = \sum_i W_i^{E \rightarrow E} x_i(t - \tau^{E \rightarrow E}) - \sum_j W_j^{I \rightarrow E} z_j(t). \quad (40)$$

Consequently, the derivative of the output rate y with respect to the inhibitory weights $W_j^{I \rightarrow E}$ is simply the negative firing rate z_j of the inhibitory neuron j :

$$\frac{\partial}{\partial W_j^{I \rightarrow E}} y(t) = -z_j(t). \quad (41)$$

We can now calculate the partial derivative of the energy function Eq. 24 with respect to the inhibitory weights:

$$\frac{\partial}{\partial W_j^{I \rightarrow E}} \Psi = \langle (y(t) - \rho_0) \frac{\partial}{\partial W_j^{I \rightarrow E}} y(t) \rangle_t + \rho_s \langle z_j(t) \rangle_t W_j^{I \rightarrow E} \quad (42)$$

$$= -\langle y(t) z_j(t) - (\rho_0 + \rho_s W_j^{I \rightarrow E}) z_j(t) \rangle_t \quad (43)$$

$$= -\frac{1}{T} \int y(t) z_j(t) - (\rho_0 + \rho_s W_j^{I \rightarrow E}) z_j(t) dt \quad (44)$$

A comparison with Eq. 23 shows that the inhibitory plasticity rule is indeed a gradient descent on the energy function Ψ :

$$\langle \Delta W_j^{I \rightarrow E} \rangle_{Z,Y|X} = -\eta T \frac{\partial}{\partial W_j^{I \rightarrow E}} \Psi. \quad (45)$$

3 Supplementary Simulation Results & Discussion

3.1 Single Cell Simulations

3.1.1 Uncorrelated Signals

To investigate the effect of uncorrelated signals on inhibitory tuning we stimulated the single integrate-and-fire neuron we used for Fig. 1, 2, & 3 with two additional protocols. In the absence of any temporal structure, i.e. if each synapse received a Poisson process with a constant rate, the plasticity rule rapidly established a global balance with identical inhibitory weights for all

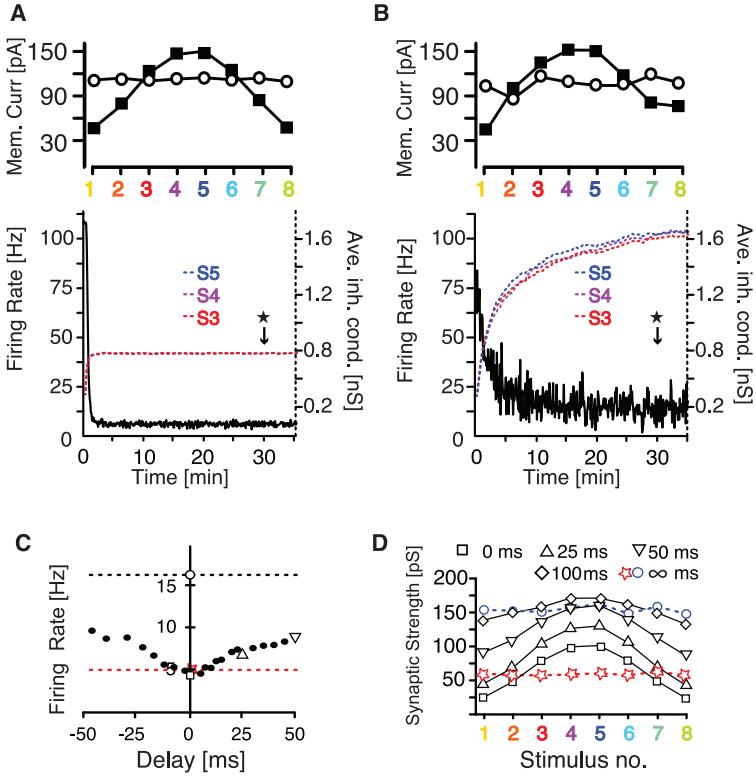


Fig. S2: The effect of uncorrelated or delayed signals on inhibitory tuning.

Tuning solutions for (A) unstructured input noise (on all synapses) and (B) structured but uncorrelated input signals (i.e. different signals for excitatory and inhibitory synapses): Upper panel: Excitatory and inhibitory membrane currents (black and white symbols, respectively) evoked by each signal channel, averaged over 8 s after inhibitory synaptic plasticity dynamics reached steady state. Lower panel: Temporal evolution of the postsynaptic firing rate (solid line) and the average synaptic weights of the inhibitory synapses associated with 3 representative signals (dotted lines). The \star symbols indicate the time at which the respective upper panels were recorded. (C) Output firing rate as a function of the delay between the excitatory and inhibitory signal stream. All other parameters remained as in Fig. 1. The dashed lines show the firing rates for unstructured noisy inputs (red) and structured but uncorrelated inputs (blue, see also panels A & B). Open symbols mark the delays investigated in (D). (D) Shape of the inhibitory synaptic weight tuning as a function of different delays.

channels and firing rates close to the target rate ρ_0 (Fig. S2 A). When we stimulated the cell with temporally structured input as in the main part of the paper, but removed the correlation between excitatory and inhibitory signals, the learning rule could still establish a global balance of all input currents (albeit with higher inhibitory synaptic weights) (Fig. S2 B), but failed to bring the firing rates to the target rate ρ_0 , because in this scenario some excitatory spikes cannot be balanced since they lack inhibitory partner spikes.

3.1.2 Delayed Signals

Similarly, but much less pronounced, a delay between the excitatory and the inhibitory component of the signals also caused a deviation of the postsynaptic firing rate from the target rate ρ_0 (Fig. S2 C). The detailed balance of each input channel on the other hand was maintained for delays smaller than or equal to the autocorrelation time of the input signals (50 ms in our simulations): Although the learning rule compensated for the delay between excitation and inhibition through up-regulating the weights of all inhibitory synapses equally, the tuning shape of the excitatory synapse population was maintained (Fig. S2 D). For delays much larger than the autocorrelation time of the input signals, the correlation between excitation and inhibition is lost, leading to the above case of structured, but uncorrelated (on timescales ≤ 100 ms) input signals (Fig. S2 B)

3.1.3 Convergence Times

The convergence time, i.e. the time from the moment the excitation-inhibition balance is disturbed, to when ISP has re-established a detailed balance, is affected by various parameters. Apart from the dependence on the learning rate η , it depends on the pre- and postsynaptic firing rates, the target firing rate ρ_0 as well as on the number of input signals and how often they are co-active. Firstly, pre- and postsynaptic firing rates influence the speed of convergence, simply because learning is spike-based. It is worth noting, moreover, that depending on whether a synapse has to be strengthened or weakened, plasticity is driven by either both pre- and postsynaptic activity, or by presynaptic activity alone. Therefore learning speed depends on whether the disturbance of the balance introduces an overshoot of excitation or inhibition. Secondly, the target rate ρ_0 plays a role, because the learning rule first establishes and then maintains a global balance with an average firing rate equal to the target rate. The firing rates during the subsequent establishment of the detailed balance thus depend on the target rate, which thereby indirectly influences learning speed. Finally, the maintenance of the global balance during the refinement to the detailed balance implies that the input channels cannot learn independently. The degree of cross-talk between channels depends on how often two or more channels are co-active, which in turn depends on (i) the total number of channels and (ii) the temporal sparseness of their activity.

Because the speed of the balancing process depends on all these parameters, we emphasize that in our fit of the experimental data in Fig. 3 the agreement in the time constant of the rebalancing (which can also be altered by the executing experimenter (11)) is not as important as the agreement in the shape of the rebalancing dynamics: Both in the experimental observations and in the model, the time course of the inhibition-dominated channel is roughly linear (presumably because it is driven by presynaptic spikes alone, and their rate does not change during learning), while the time course of the excitation-dominated channel is roughly exponential (presumably

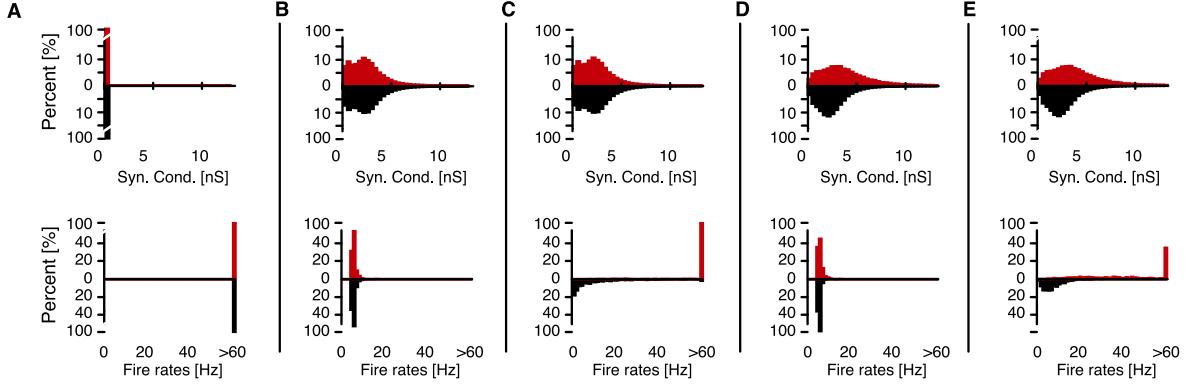


Fig. S3: Additional statistics for Fig. 4.

Five consecutive snapshots of network states as in Fig. 4. Each snapshot shows: The distributions of inhibitory synaptic weights $\mathbf{W}^{I \rightarrow E}$ onto all cells of the red cell assembly and the unstrengthened (black) control assembly (upper panels) and the distribution of average firing rates recorded for 1 s (lower panels). Temporal development identical to Fig. 4: (A) Synchronous regular network dynamics at the beginning of the simulation with weak inhibitory-synapses. (B) Establishment of the AI (steady) state through up-regulated inhibitory synaptic weights. (C) Introduction of an excitatory synaptic memory as strongly enhanced excitatory synaptic strengths between neurons of the red designated neuron group in Fig. 4 A leads to high regular firing of these neurons. (D) Recovery of the AI state. (E) Memory recall as pattern completion by means of externally driving one quarter of the memory pattern with additional excitatory stimulus.

through the dependence on postsynaptic spiking, which decreases as inhibition increases).

3.1.4 Sparse Responses

After inhibitory plasticity has established a detailed balance of excitation and inhibition, neuronal responses to sensory stimulation are sparse, i.e. neurons emit relatively few, and isolated action potentials (Fig. 1). The main reason for this sparseness is that the inhibitory plasticity rule enforces the neuron to fire at a target rate, which is determined by the parameters of the learning rule (see theory above) and which we choose to be small.

For balanced inputs, the few spikes that are emitted are caused by two effects. Firstly, rapid transients in the input signals cannot be balanced because of the slower response of the inhibitory synapses. This effect is important for rapidly changing input signals. It causes reliable spikes that are locked to input transients and show little trial-to-trial variability in their timing (cf. the onset transient in Fig. 2F of the main text). Secondly, random fluctuations around the balance of excitation and inhibition can sometimes cause sufficiently large overshoots of excitation to drive the neuron to spike. This effect dominates over the effect of transients when the input changes slowly. Moreover, fluctuation-driven spiking occurs primarily when both excitatory and inhibitory inputs are large, because large inputs cause large fluctuations, and spike times are relatively unreliable, because large fluctuations occur at random times. In our simulations, the input signals change relatively slowly compared to the synaptic time constants (50 ms autocorrelation time vs. 5/10 ms synaptic time constants for excitation/inhibition), so we expect that most spikes are fluctuation-driven.

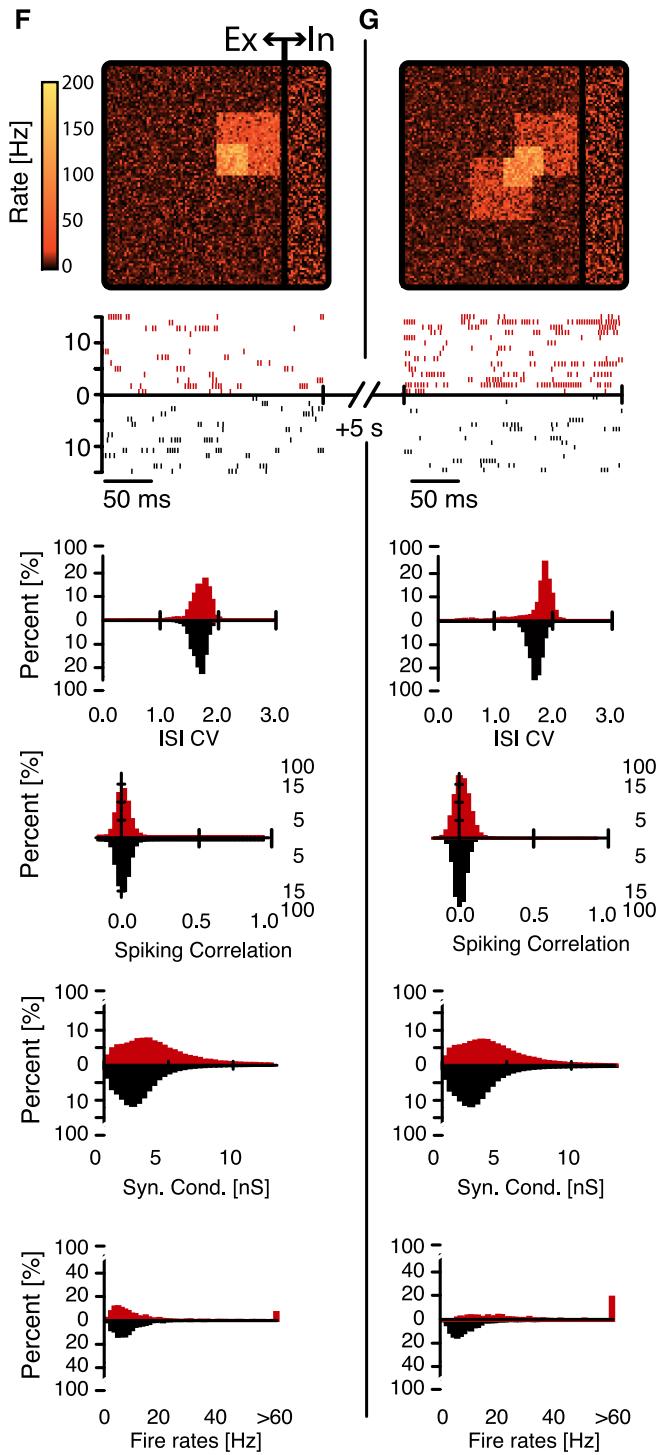


Fig. S4: Two additional instances of recall similar to Fig. 4 E. Each snapshot, as in Fig. 4, contains (from top to bottom): - The momentary (1s) average firing rate of each neuron - A raster plot of 30 randomly drawn neurons from one (red) cell assembly and the control group - The distributions of coefficients of variation of interspike intervals (ISI CVs) recorded from all neurons in the red and black groups. - The distributions of spiking correlations between spike trains from neurons in the same designated groups - The distributions of inhibitory synaptic weights $\mathbf{W}^{I \rightarrow E}$ in the network. - The distribution of average firing rates recorded for 1 s. Temporal development continuing from to Fig. 4 E: (F) Memory recall of the cell assembly as pattern completion by means of externally driving cells in the second memory pattern (marked blue in Fig. 4) with additional excitatory current. (G) Memory recall of both patterns as pattern completion by means of externally driving cells in both patterns symmetrically with additional excitatory stimulus.

These arguments hold for a detailed balance, which – according to the theory and our simulations – is the target state of the inhibitory learning rule. Whether a detailed balance can be reached, however, depends on whether the information in the excitatory inputs of the neuron is also available in its inhibitory inputs. If the excitatory input current shows a stimulus tuning, for example, the presynaptic inhibitory neurons need to be suitably tuned, as well, in order to allow a detailed balance of excitation and inhibition. If the inhibitory input neurons have no

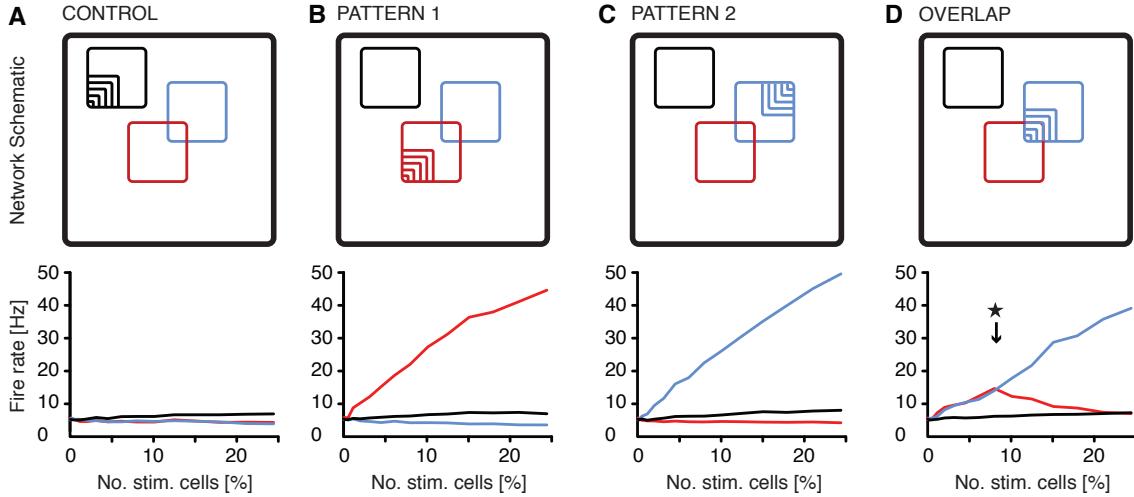


Figure S5: Recall firing rates as a function of the number of externally driven neurons. Each subfigure shows a schematic of the network, indicating which cells in which pattern are driven (upper panel), and a plot of the resulting firing rates in each group as a function of the number of cells stimulated, following the same color code as above. (A) Different numbers of cells of an arbitrarily chosen control group are driven with an external stimulus. (B, C) Different numbers of cells belonging exclusively to either (red) pattern 1, or (blue) pattern 2 are driven with an external stimulus. (D) Cells belonging to both patterns are driven with an additional stimulus. The \star symbols indicates the maximum number of shared cells that can be driven. Beyond this point the cell populations are driven asymmetrically, i.e. additionally cells belong to (blue) pattern 2 but not to (red) pattern 1.

or weaker stimulus tuning (45), stimulus-dependent changes in the excitatory inputs cannot be balanced out, and the optimal state of the learning rule is a mere global balance. Variability in the tuning of inhibitory neurons could thus be an explanation for the variability in the co-tuning of excitation and inhibition that has been observed in sensory cortices (46).

The co-tuning precision of excitation and inhibition also has a strong influence on the response patterns of the neurons. In the globally balanced state, neuronal firing is not as sparse as in the case of the detailed balance, because inputs with strong synapses can cause large, unbalanced excitatory currents that dominate postsynaptic activity and can cause bursts of action potentials that last as long as these inputs are activated (Fig. 2B).

3.2 Network Simulations

3.2.1 Additional Network Statistics

To complete the statistics for the network simulations (Fig. 4), we additionally supply the distributions of inhibitory synaptic weights onto all neurons in the red cell assembly and the black control assembly (see Fig. 4), as well as their distributions of firing rates, averaged over 1 s, for each snapshot (Fig. S3).

3.2.2 Additional Recall Protocols

Recall of an imprinted pattern in recurrent networks as shown in Fig. 4 was not limited to one assembly, but could also be achieved in the other assembly (Fig. S4F), or even in both patterns simultaneously, if cells in both patterns were driven in equal numbers (Fig. S4G). Elevated firing rates during recall are a direct consequence of the strengthened excitatory synapses within each assembly (Fig. S5). When cells that are part of the un-strengthened control assembly are stimulated, the background firing rate of (the other) cells in the control assembly is raised only slightly. The firing rate of the two cell assemblies remains low, decreases slightly even, because these cells receive greater amounts of inhibitory currents through their strengthened synapses. When cells that are part of a cell assembly are stimulated, the average firing rate of the (other) cells in the stimulated assembly increases as a function of the number of stimulated cells. The firing rate in the other pattern tends to decrease because patterns inhibit each other slightly through inhibition that originates from inhibitory cells servicing both patterns (Fig. S5B, C). When equal numbers of cells in two separate patterns are driven simultaneously, both assemblies can be activated (Fig. S5D).

Driving a subset of cells with an external stimulus is one possibility of changing the balance of excitatory and inhibitory input currents a cell receives. Other possibilities, such as population-selective disinhibition (11) or global changes in synaptic efficacies through the application of a neuromodulator (47–49) are also conceivable.

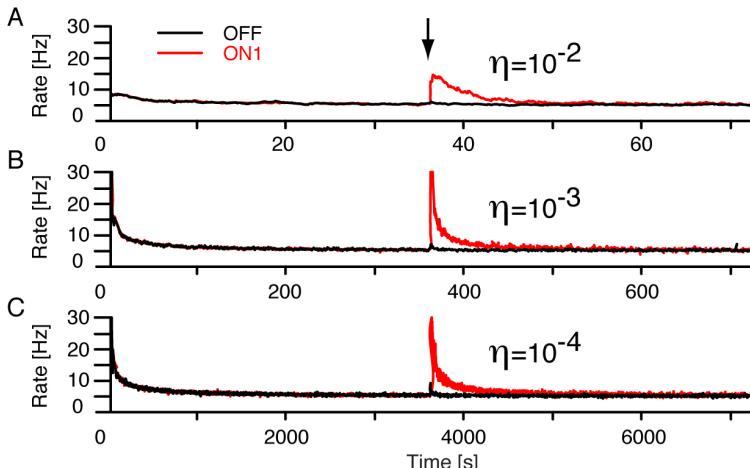


Fig. S6: Robustness to changes in the learning rate η . (**A, B, C**) Average population firing rate (1 s filter time constant) recorded from a set of background neurons (black) and from one of the cell assemblies (red) (as in Fig. 4) for different learning rates η . Black arrow indicates time of strengthening of synapses within the Hebbian assemblies. Take note of the rescaled time axes.

3.2.3 Stability and Parameter Robustness

To investigate how robustly the AI state emerges in recurrent networks, we repeated the simulation protocol used to create Fig. 4 of the main text for a wide range of network parameters (\bar{g}^{EE} , \bar{g}^{EI} , ρ_0). We characterized the network dynamics after the inhibitory synapses converged to their steady state values with various measures extracted from the following 30 s simulation, with plasticity turned off. We used the ISI CV and the standard deviation of the population rate (filter time constant 5 ms), σ_{Rate} , to evaluate the regularity and synchrony of the dynamics, and also recorded the average population firing rate to compare the real output firing rate with the target firing rate ρ_0 .

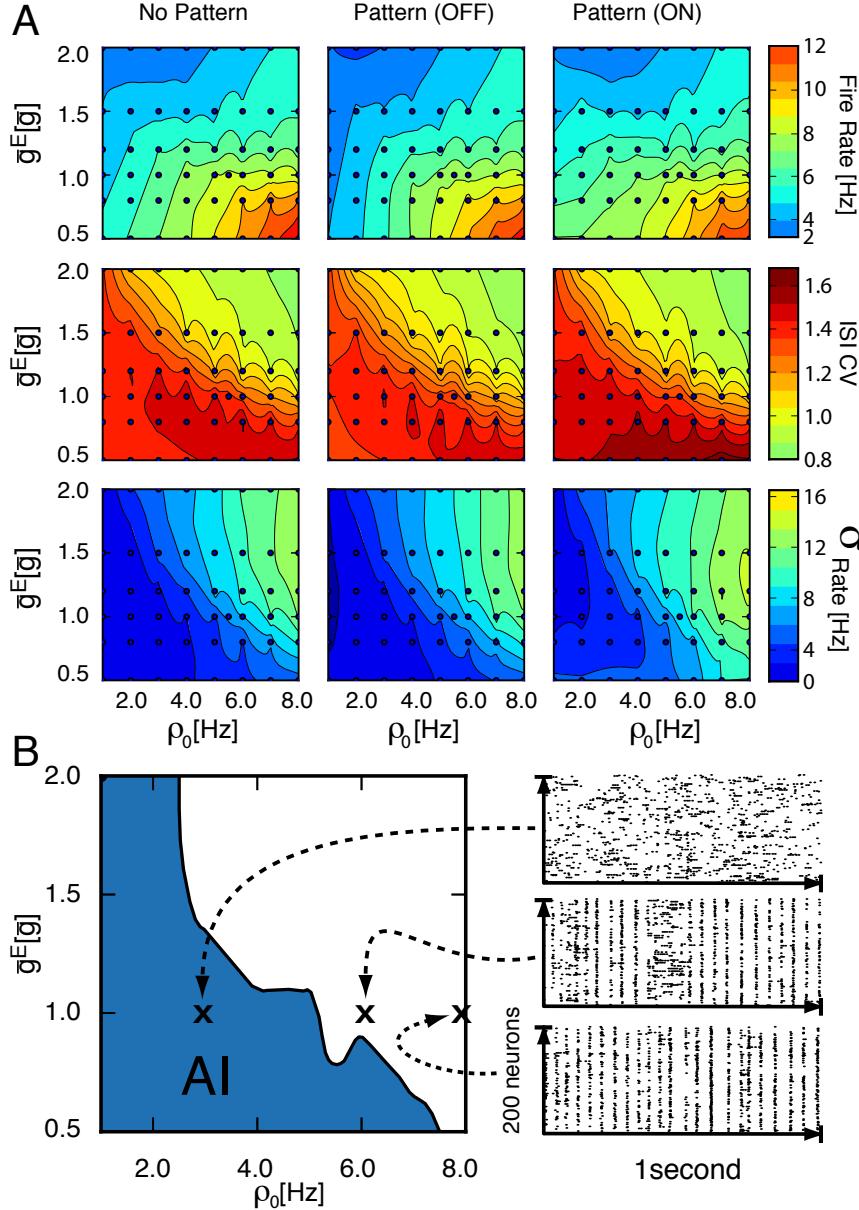


Fig. S7: Asynchronous irregular activity emerges over a wide parameter range.

(A) Network statistics at the end of the learning protocol of Fig. 4 in the main text, as a function of the excitatory synaptic conductance \bar{g}^E , scaled in units of the standard value $\bar{g} = 3\text{nS}$, and the target firing rate ρ_0 . Excitatory conductances onto excitatory and inhibitory cells are kept equal ($\bar{g}^E = \bar{g}^{EE} = \bar{g}^{EI}$). Top row: Mean firing rate of excitatory subpopulations (filtered population spike count; exponential filter with 1 s time constant). Middle row: Mean ISI CV. Bottom row: Standard deviation of the population rate (5 ms filter time constant). Columns signify different subpopulations. No Pattern: Random network without memory pattern as in Fig. 4B. Pattern (OFF): Network after the introduction and balancing of a memory as in Fig. 4D. Recorded neurons belong to the (black) un-strengthened control group. Pattern (ON): As in Fig. 4D, recorded neurons belong to the red cell assembly. Simulated parameter combinations are represented by black dots. For visualization, heat maps were interpolated using natural neighbor interpolation. (B) Left: Parameter region with AI activity. Right panel: Spike raster plots for three representative parameter combinations as indicated on the graph.

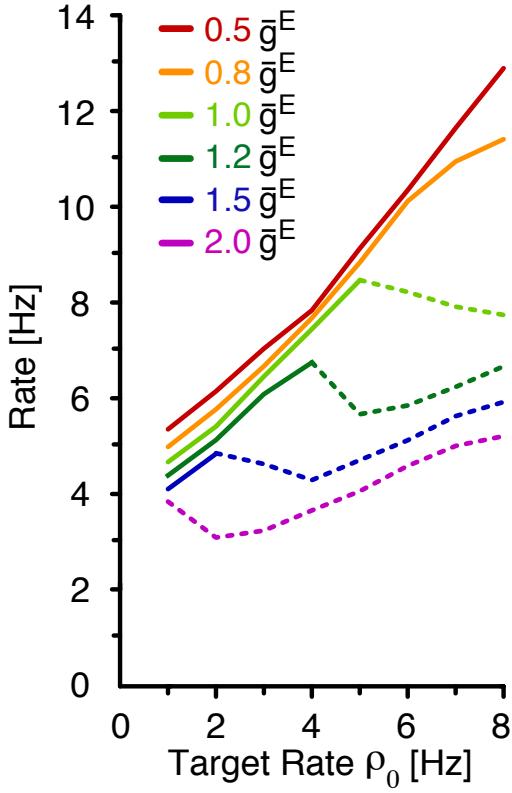


Fig. S8: Robustness in regards to target rate ρ_0 . **(A)** Mean output rate of background neurons as a function of the target rate ρ_0 , for different values of the excitatory conductances $\bar{g}^{EE} = \bar{g}^{EI}$ (after convergence, 1 h simulated time). Solid lines correspond to network states that were classified as AI. Dashed lines represent states classified as non-AI.

We initially verified that the learning rate η had no de-stabilizing effect on the population dynamics of the network in Fig. 4 (Fig. S6). To restrict the dimensionality of our parameter scan, we subsequently kept the learning rate fixed at $\eta = 1 \times 10^{-4}$ and studied the network dynamics as a function of ρ_0 and the excitatory synaptic weights $\bar{g}^E = \bar{g}^{EI} = \bar{g}^{EE}$, in the range of $0.5 \bar{g}$ to $2.0 \bar{g}$ ($1.5 - 6$ nS). As illustrated in Fig. S7, ISI CV and variability of the population rate are robust to changes in \bar{g}^E and ρ_0 , and only leave the asynchronous irregular regime when both \bar{g}^E and ρ_0 are large.

To summarize and quantify the results across conditions (naive network or with imprinted assemblies) we used the criterion

$$(\text{ISI CV} > 1) \wedge (\sigma_{\text{Rate}} < 5 \text{ Hz}) \quad (46)$$

as a binary measure of whether or not a respective parameter combination led to AI activity (Fig. S7 B). The results illustrate that AI activity emerges over a wide parameter range of \bar{g}^E and ρ_0 and only gives way to synchronous activity when both parameters are large (see representative spike rasters in Fig. S7 B).

Unlike in the feed-forward case, the target rate ρ_0 does not accurately predict the final output firing rate of the network. The observed firing rate is consistently higher than theoretically predicted (Fig. S8) by the simplified linear model. Potential reasons for this are discussed in section 2.8.

To investigate whether instability of the AI state for large excitatory conductances is caused by the increase in excitatory drive to inhibitory population or by the increase in recurrent excitation, we also studied the effects of individual changes to the connection weights \bar{g}^{EI} and \bar{g}^{EE} (Fig. S9).

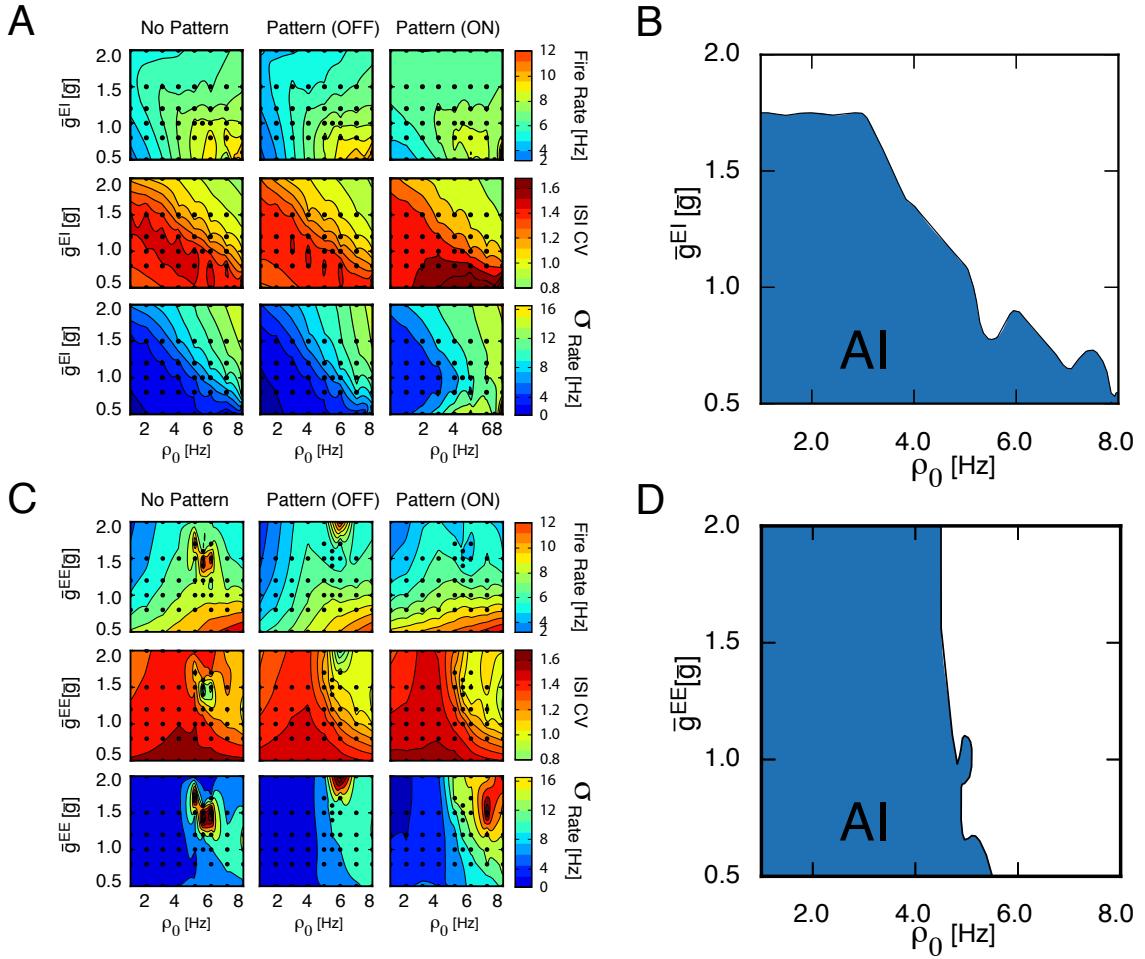


Fig. S9: Varying excitatory conductances onto excitatory (\bar{g}^{EE}) and inhibitory neurons (\bar{g}^{EI}) independently does not disrupt emergence of AI activity for a wide range of parameters. **(A)** Same as Fig. S3 but keeping \bar{g}^{EE} fixed. **(B)** Parameter region for AI activity as in Fig. S3 outlining AI-producing parameter space B. **(C)** Same as Fig. S3 but keeping g^{EI} fixed. **(D)** Parameter region with AI activity as in Fig. S3.

While an increase in \bar{g}^{EI} led to similar network instabilities as observed when both conductances are changed simultaneously (cp. Fig. S7 & S9 A,B), increases in \bar{g}^{EE} did not strongly affect the dynamical state of the network (Fig. S9 C,D), indicating that the instability of the AI state arises from a synchronization of the inhibitory population caused by large excitatory drive.

In summary, the ability of inhibitory synaptic plasticity to restore balanced network states was shown to be robust to changes in learning rate as well as excitatory synaptic weights. The ultimately resulting network activity we observed was mainly controlled by the target rate ρ_0 and the excitatory conductance g^{EI} onto the inhibitory population. We expect that the precise transition between AI and non-AI activity states also depends on the external drive, but this was not systematically explored. We conclude that the plasticity mechanism at the inhibitory synapse we put forward provides a powerful mechanism to regulate firing rates and robustly drive spiking recurrent neural networks to the balanced regime where activity is irregular and asynchronous.

4 References

1. N. Brunel, *J. Comp. Neurosci.* **8**, 183 (2000).
2. C. Van Vreeswijk, H. Sompolinsky, *Science* **274**, 1724 (1996).
3. M. Tsodyks, T. Sejnowski, *Network: Comput. Neural Syst.* **6**, 111 (1995).
4. A. Renart, *et al.*, *Science* **327**, 587 (2010).
5. M. Wehr, A. M. Zador, *Nature* **426**, 442 (2003).
6. M. Okun, I. Lampl, *Nature Neurosci.* **11**, 535 (2008).
7. R. C. Froemke, M. M. Merzenich, C. E. Schreiner, *Nature* **450**, 425 (2007).
8. J. de la Rocha, C. Marchetti, M. Schiff, A. D. Reyes, *J. Neurosci.* **28**, 9151 (2008).
9. B. K. Murphy, K. D. Miller, *Neuron* **61**, 635 (2009).
10. Y. Shu, A. Hasenstaub, D. A. McCormick, *Nature* **423** (2003).
11. T. P. Vogels, L. F. Abbott, *Nature Neurosci.* **12**, 483 (2009).
12. W. Gerstner, *Neural Comput.* **12**, 43 (2000).
13. T. P. Vogels, L. F. Abbott, *J. Neurosci.* **25**, 10786 (2005).
14. A. Kumar, S. Rotter, A. Aertsen, *Nature Rev. Neurosci.* **11**, 615 (2010).
15. J. Cafaro, F. Rieke, *Nature* **468**, 964 (2010).
16. T. Hromádka, M. DeWeese, *PLoS Biol.* **6** (2008).
17. M. A. Woodin, K. Ganguly, M. M. Poo, *Neuron* **39**, 807 (2003).
18. T. Balena, M. A. Woodin, *Europ. J. Neurosci.* **27**, 2402 (2008).
19. M. A. Woodin, A. Maffei, *Inhibitory Plasticity* (Springer, 2010).
20. M. R. DeWeese, M. Wehr, A. M. Zador, *J. Neurosci.* **23**, 7940 (2003).
21. H. Yao, L. Shi, F. Han, H. Gao, Y. Dan, *Nature Neurosci.* **10**, 772 (2007).
22. S. Crochet, J. Poulet, Y. Kremer, C. Petersen, *Neuron* **69**, 1160 (2011).
23. L. M. Aitkin, D. J. Anderson, J. F. Brugge, *J. Neurophysiol.* **33**, 421 (1970).
24. I. O. Volkov, a. V. Galazjuk, *Neuroscience* **43**, 307 (1991).
25. P. Series, P. E. Latham, A. Pouget, *Nature Neurosci.* **7**, 1129 (2004).
26. J. Beck, V. R. Bejjanki, A. Pouget, *Neural Comput.* **23**, 1484 (2011).
27. D. Hebb, *The organization of behavior; a neuropsychological theory.* (Wiley-Interscience, New York, 1949).
28. W. Gerstner, R. Ritz, J. L. van Hemmen, *Biol. Cybern.* **69**, 503 (1993).
29. D. J. Amit, N. Brunel, *Cereb. Cortex* **7**, 237 (1997).

30. A. Renart, R. Moreno-Bote, X.-J. Wang, N. Parga, *Neural Comput.* **19**, 1 (2007).
31. V. Kilman, M. C. W. van Rossum, G. G. Turrigiano, *J. Neurosci.* **22**, 1328 (2002).
32. K. Hartmann, C. Bruehl, T. Golovko, A. Draguhn, *PLoS One* **3**, e2979 (2008).
33. A. Maffei, A. Fontanini, *Curr. Opin. Neurobiol.* **19**, 168 (2009).
34. M. C. W. van Rossum, G. G. Turrigiano, S. B. Nelson, *J. Neurosci.* **22**, 1956 (2002).
35. MATLAB, (*v7.10.0, R2010a*) (The MathWorks Inc., Natick, Massachusetts, 2010).
36. R. Brette, *et al.*, *J. Comp. Neurosci.* **23**, 349 (2007).
37. A. Kumar, S. Rotter, A. Aertsen, *J. Neurosci.* **28**, 5268 (2008).
38. M. Diesmann, M. O. Gewaltig, A. Aertsen, *Nature* **402**, 529 (1999).
39. E. Nordlie, M.-O. Gewaltig, H. E. Plessner, *PLoS Comp. Biol.* **5**, e1000456 (2009).
40. S. Kunkel, M. Diesmann, A. Morrison, *Frontiers Comp. Neurosci.* **4**, 160 (2011).
41. T. P. Vogels, K. Rajan, L. F. Abbott, *Annu. Rev. Neurosci.* **28**, 357 (2005).
42. R. Kempter, W. Gerstner, J. van Hemmen, *Physical Review E* **59**, 4498 (1999).
43. R. Kempter, W. Gerstner, J. L. van Hemmen, *Neural Comput.* **13**, 2709 (2001).
44. J. Kremkow, A. Aertsen, A. Kumar, *J. Neurosci.* **30**, 15760 (2010).
45. S. B. Hofer, *et al.*, *Nature Neurosci.* **14** (2011).
46. G. K. Wu, R. Arbuckle, B.-H. Liu, H. W. Tao, L. I. Zhang, *Neuron* **58**, 132 (2008).
47. Z. Gil, B. W. Connors, Y. Amitai, *Neuron* **19**, 679 (1997).
48. Z. Xiang, *Science* **281**, 985 (1998).
49. A. A. Disney, C. Aoki, M. J. Hawken, *Neuron* **56**, 701 (2007).

5 Technical Appendix

5.1 Tabular Parameter Summaries for the Network Simulations

A		Model Summary
Populations		Two: excitatory, inhibitory
Topology		—
Connectivity		Random all-to-all connections
Neuron model		Leaky integrate-and-fire, fixed voltage threshold, fixed absolute refractory time
Synapse model		conductance based inputs (exponentially decaying PSC)
Plasticity		Inhibitory plasticity
Input		Fixed input current to all units
Input (Recall)		Poisson spike trains to sub-population
Measurements		Spike activity

B			Populations
Name	Elements	Size	
E	Iaf neuron	$N_E = 4N_I$	
I	Iaf neuron	N_I	

C				Connectivity
Name	Source	Target	Pattern	
EE	E	E	Random with sparseness ϵ , weight \bar{g}^E ($\chi\bar{g}^E$ for connections in a pattern)	
EI	E	I	Random with sparseness ϵ , weight \bar{g}^E	
IE	I	E	Random with sparseness ϵ , weight plastic \bar{g}_{ij}^{IE}	
II	I	I	Random with sparseness ϵ , weight \bar{g}^{II}	

D1		Neuron and Synapse Model
Name	Type	
	Iaf neuron	
Type	Leaky integrate-and-fire, exponential conductance based input	
Subthreshold dynamics	$\tau \frac{dV_i}{dt} = (V^{\text{rest}} - V_i) + (g_i^E(V^E - V_i) + g_i^I(V^I - V_i) + I_b) \times \frac{1}{g^{\text{leak}}}$ $V(t) = V_{\text{rest}}$ otherwise	
Synaptic dynamics	$\tau_E \frac{dg_i^E(t)}{dt} = -g_i^E(t) + \bar{g}^E \delta(t - t_i^*)$ and $\tau_I \frac{g_i^I(t)}{dt} = -g_i^I(t) + \bar{g}_{ij}^{Ix} \delta(t - t_i^*)$	
Spiking	If $V(t - dt) < \theta \wedge V(t) \geq \theta$ 1. set $t^* = t$ 2. emit spike with time-stamp t^*	

D2		Plasticity Model
Name	Type	
	Inhibitory Spike Timing Dependent Plasticity (iSTDP)	
Type	Symmetric iSTDP with a constant offset for presynaptic spikes	
Acts on	IE	
Synaptic traces		$\tau_{\text{STDP}} \frac{dx_i}{dt} = -x_i + \delta(t - t_i^*)$
Online rule		$W_{ij} = W_{ij} + \eta(x_i - \alpha)$ for presynaptic spikes of neuron j at time t_f^j and $W_{ij} = W_{ij} + \eta x_j$ for postsynaptic spikes at time t_f^i

E		Input
Type	Description	
Current input	Fixed current I to all neurons	

Table 1: Tabular description of network model (Fig. 4) following (39).

Populations		
Name	Value	Description
N_E	8000	Size of excitatory population E
N_I	2000	Size of inhibitory population I
Connectivity		
Name	Value	Description
ϵ	0.02	Probability of any connection (EE,EI,IE,II)
\bar{g}	3 nS	Basic weight unit
\bar{g}^E	\bar{g}	Weight of all excitatory synapses (= 3 nS)
γ	10	Scaling factor for inhibitory weights
\bar{g}^{II}	$\gamma \bar{g}$	Weight of inhibitory to inhibitory synapses (= 30 nS)
\bar{g}_{ij}^{IE}	$W_{ij}\bar{g}^{II}$	Weight of inhibitory to excitatory synapses (= $W_{ij} \times 30$ nS)
χ	5	Potentiation factor of excitatory weights belonging to one or more patterns
Neuron Model		
Name	Value	Description
τ	20 ms	Membrane time constant
Θ	-50 mV	Spiking threshold
V_{rest}	-60 mV	Resting potential
V^E	0 mV	Excitatory reversal potential
V^I	-80 mV	Inhibitory reversal potential
g_{leak}	10 nS	Leak conductance
I_b	200 pA	Background current to each cell (unless stated otherwise)
τ_{ref}	5 ms	Absolute refractory period
Synapse Model		
Name	Value	Description
τ_E	5 ms	Decay constant of AMPA-type conductance
τ_I	10 ms	Decay constant of GABA-type conductance
Plasticity Model		
Name	Value	Description
τ_{STDP}	20 ms	Decay constant of (pre and post) synaptic trace
η	1×10^{-4}	Learning rate
α	0.12	Presynaptic offset
W_{min}	0	Minimum inhibitory synaptic weight
W_{max}	$10\bar{g}^{II}$	Maximum inhibitory synaptic weight

Table 2: Simulation parameter summary for the network model (Fig. 4) following (40).

5.2 MATLAB® Code

5.2.1 Input Signals

Annotated MATLAB® code to create input spike trains as used for Fig. 1, 2, & 3.

```
clear all  
close all
```

%% Figure dimensions

```
figure(1)  
set(gcf,'PaperUnits','centimeters')  
%Setting the units of the figure on paper to centimeters.  
xSize = 13; ySize = 13;  
%Size of the figure  
xLeft = (21-xSize)/2; yTop = (30-ySize)/2;  
%Coordinates to center the figure on A4-paper  
set(gcf,'PaperPosition',[xLeft yTop xSize ySize])  
%This command sets the position and size of the figure on the paper to the desired values.  
set(gcf,'Position',[0.5 0.5 xSize*50 ySize*50])  
set(gcf, 'Color', 'w');
```

%% Parameter Initialization:

```
duration=5000; % duration in ms  
dt=0.1; % simulation time step.  
tau=50; % Filter time for the input.  
tRef=5; % Refractory period for the spike trains.  
nn=100; % Number of spiketrains we seek to create later.  
spikebin=5; % Number of ms per Psth bin.  
  
Timevector=(0.1:dt:duration);  
% A vector of time in ms, in steps of dt.  
WhiteNoise=rand(1,length(Timevector))-0.5;  
% uniform white noise drawn from +/- 0.5 as long as the time vector.  
FilteredWhiteNoise=WhiteNoise.*0;  
% an empty vector which we will use to create the time-filtered input.  
SpikeTrains=zeros(nn,length(Timevector));  
%A Matrix that will hold all spiketrains.  
PlotTrains=SpikeTrains;  
% This is just a plotting variable to overcome a screen resolution problem in matlab.  
avetrain=0;  
% A counter to calculate the average firing rate.  
tslt=0;  
% (== t(me)s(ince)l(ast)(t)oggle (this serves as a Boolean for the sparsification of the input  
signal.
```

```
tsls=zeros(nn,1);
% (== t(ime)s(ince)l(ast)(s)pike (to keep track of the refractory period of each spike train)
BinnedSpikeTrains=zeros(1,duration/spikebin);
% a vector to create a PSTH with binwidth "spikebin" from the spike trains.
```

%% Making the time-filtered white noise signal:

```
for t=2:duration/dt
    FilteredWhiteNoise(t) = WhiteNoise (t) - ...
        (WhiteNoise (t) - FilteredWhiteNoise(t-1))*exp(-dt/tau);
end
%% This routine changes the signal trace "FilteredWhiteNoise" by a "exp(-dt/tau)" fraction of
the difference between the signal and a random number.

FilteredWhiteNoise=FilteredWhiteNoise./max(FilteredWhiteNoise);
%Normalize to a maximum value of 1.
```

%% Plotting:

```
figure(1)
subplot(4,1,1)
plot(Timevector, FilteredWhiteNoise)
axis([0 duration -1 1])
x=sprintf('Time Filtered White Noise (FWN)');
title (x)
```

%% Normalize and Rectify:

```
FilteredWhiteNoise=FilteredWhiteNoise.*((500*dt/1000));
% Normalizes the trace to a peak value of 500Hz*dt (=0.05).
FilteredWhiteNoise(FilteredWhiteNoise<0)=0;
%Sets all negative values of "FilteredWhiteNoise" to 0.
```

%% Plotting:

```
subplot(4,1,2)
hold on
plot(Timevector,FilteredWhiteNoise, 'b', 'LineWidth', 1.1)
```

%% Sparsifieing the Input Signals:

```
% This routine goes through the signal trace and deletes entire "activity bumps" if certain
conditions are fulfilled:
toggle = 0;
tslt=0;
for d=1:duration/dt-1
```

```

% Routine becomes active (sets toggle == 1) if the signal is ==0, and the toggle is off (==0)
% and has been off for at least 1 ms:
if(FilteredWhiteNoise(d)==0 && toggle==0 && (d-tsLT>10))
    toggle=1; % toggle set
    tsLT=d; % "refractory" for toggle is set
end
% If routine active, every signal value is set to zero:
if (toggle==1)
    FilteredWhiteNoise(d) = 0;
% If routine has been active for longer than 0.5 ms, and the signal is 0, routine becomes inactive:
if (FilteredWhiteNoise(d+1)==0 && (d-tsLT>5))
    toggle=0;
end
end

```

%% Plotting:

```

subplot(4,1,2)
hold on
plot(Timevector, FilteredWhiteNoise, 'r')
axis([0 duration -0.005 0.05])
title ('Rectified & calibrated (blue) and sparsened (red) FWN')

```

%% Adding background firing rate:

```

FilteredWhiteNoise=FilteredWhiteNoise+(5*dt/1000);
% This is adjusted so that without any FilteredWhiteNoise the firing rate is 5 Hz*dt (0.0005).

```

%% Creating 100 spike trains:

```

for i=1:nn
    for t=1:duration/dt
        if (tsLS (i) <= 0) % Allows potential spike if refractory period has subsided
            if(rand<FilteredWhiteNoise(t))
                SpikeTrains (i,t) = 1;% Fires if random variable < "FilteredWhiteNoise".
                tsLS (i) = tRef;% Sets the absolute refractory period.
                avetrain=avetrain+1;% Counts the total number of spikes.

                if(duration/dt-t>25)% This is just a plotting routine.
                    PloTrains (i,t:t+25)=1;% (Spike is elongated for plotting.)
                end
            end
        else
            tsLS (i)=tsLS (i) -dt;% subtracts dt from refractory counter if it is still >0.
        end
    end
end

```

```
avetrain=avetrain/(nn*duration/1000); %Calculates the average firing rate in Hz.
```

%% Plotting:

```
subplot(4,1,3)
imagesc(~PloTrains)
colormap(gray)
title ('100 Spiketrains')
```

%% Recording a PSTH / Binned Input rate:

```
% This bins the spikes into bins and calculates the instantaneous firing rate in Hz.
for i=1:(duration/spikebin)-1
    BinnedSpikeTrains(i)=...
        sum(sum(SpikeTrains(:,((i-1)*(spikebin/dt))+1:(i*(spikebin/dt)))));
end
BinnedSpikeTrains= (BinnedSpikeTrains*(1000/spikebin))/nn;
```

%% Plotting:

```
subplot(4,1,4)
plot(BinnedSpikeTrains)
x=sprintf('Average input rate for 1 excitatory channel, \%3.2f Hz, ...
peak \%3.2f Hz', avetrain, max(BinnedSpikeTrains));
title (x)
```

End of code.

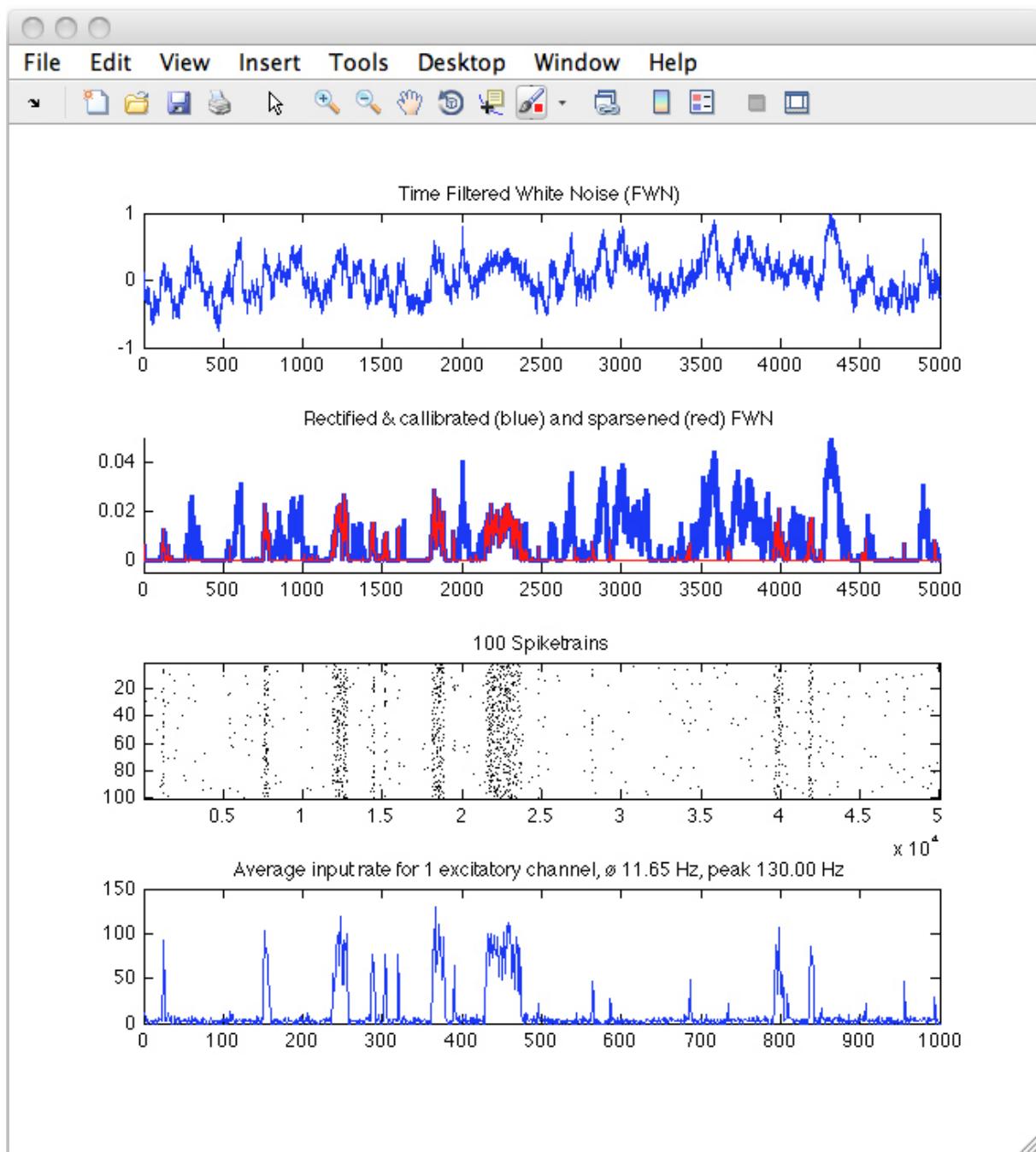


Figure S10: Output of the Input Signal Routine described in the technical appendix (5.2.1). Panels show (from top to bottom): Time filtered white noise signal \hat{s}_k over 5 seconds. - Rectified and calibrated signal \hat{s}_k (blue) and resulting sparsified signal s_k (red). - 100 Spike trains drawn from Poisson processes using $s_k + 5\text{Hz} \times dt$ as the spiking probability. - PSTH of the above spike trains in 5 ms bins.

5.2.2 Single Cell Simulations

Annotated matlab code to simulate a single cell with plastic inhibitory synapses as described in Figures 1,2, and 3. This is not the original source code. While the original model was written in C, this mode has been adjusted to run on matlab. In order to allow for reasonable run times, the structure of the code has been simplified for the sake of computational efficiency. Parameters may vary from the ones used in the paper, specifically η has been increased to accommodate slower computers.

The model is constructed as two nested time loops - the inner loop simulates `duration` ms of a single integrate-and-fire cell receiving spikes from `NCells` presynaptic spike trains. During the simulation, input signals are created *ad hoc* (for computational efficiency), and presynaptic spikes are drawn at every time step. The postsynaptic membrane potential $V(t)$ is calculated, and inhibitory synaptic weights `Synapse(i)` can change according to the above discussed plasticity rule. At the end of the simulation time ($t=duration \times dt$), the synaptic weights, as well as other variables are kept, (and also plotted) and the outer loop starts a new run of the simulation at $t=1$ with the previously saved variable values. The simulation is terminated when the number of runs reaches `NRuns`.

Beginning of MATLAB Code.

%% Initialization:

```
clear all  
close all
```

%% Figure dimensions

```
figure(1)  
set(gcf,'PaperUnits','centimeters')  
%Setting the units of the figure on paper to centimeters.  
xSize = 20; ySize = 8;  
%Size of the figure  
xLeft = (21-xSize)/2; yTop = (30-ySize)/2;  
%Coordinates to center the figure on A4-paper  
set(gcf,'PaperPosition',[xLeft yTop xSize ySize])  
%This command sets the position and size of the figure on the paper to the desired values.  
set(gcf,'Position',[0.5 0.5 xSize*50 ySize*50])  
set(gcf, 'Color', 'w');
```

%% Parameters

```
eta=0.001;  
%Learning Rate  
alpha=0.25*eta;  
%Decay term  
tauPlasticity=20;
```

```

%Time window of the learning rule.

duration=1500;
%DURATION of a single run in ms.
dt=0.1;
% Simulation time step in ms.
NRuns=100;
%Number of consecutive runs in one simulation.
%(The total duration of the simulation is thus "Duration*NRuns".
tRef=5;
% Refractory period for the spike trains.
gBarEx=0.014;
% Scaling factor of the excitatory synaptic conductance in units of the leak (gleak = 10 nS)
gBarIn=0.035;
% Scaling factor of the inhibitory synaptic conductance in units of the leak (gleak = 10 nS)
VRest=-60;
%Resting potential in mV.
Vth=-50;
% Threshold, in mV.
taumem=20;
% Membrane time constant, in ms.
EAMPA=0;
% Excitatory reversal potential, in mV.
EGABA=-80;
% Inhibitory reversal potential, in mV.
tauEx=5;
% Excitatory synaptic time constant, in ms.
tauIn=10;
% Inhibitory synaptic time constant, in ms.
stopnow=0;
% This is a flag to stop the while loop in case the EPSP routine below is uncommented

```

%% Input

```

noisetau=50;
% Filter time for the input in ms
Backgroundrate=5*dt/1000;
% Background spiking probability 5Hz*dt (= 0.0005) (see Technical Appendix 6.2.1).
ApproximateNormalizingFactor=0.03;
% This serves to normalize the trace to an approximate peak value of 1
Maxrate=500*dt/1000;
% Peak spiking probability 500Hz*dt (= 0.05))
NSigs=8;
% Number of input signals.
NCells=1000;
% Number of input spike trains.
ExFrac=0.8;
%Fraction of Excitatory spike trains.

```

```

ExGroupsize=(NCells*ExFrac)/NSigs;
% Number of spike trains in each excitatory input group
InGroupsize=(NCells*(1-ExFrac))/NSigs;
% Number of spike trains in each inhibitory input group

expGEx=exp(-dt/tauEx);
expGIn=exp(-dt/tauIn);
expPlasticity=exp(-dt/tauPlasticity);
expnoise=exp(-dt/noisetau);
% Pre-calculated exponential factors for speed of simulation

```

%% Vectors

```

Input=zeros(1,NSigs);
% A vector that holds the momentary input signal for each signal channel
Timevector=(0.1:dt:duration);
% A vector of time in ms, in steps of dt.
Exkeep=Timevector*0;
Inkeep=Timevector*0;
% Vectors to keep the synaptic currents.
FilteredWhiteNoise=zeros(1,8);
% A vector to create the time-filtered input
InputGroup=zeros(1,NCells);
% A Vector that keeps track of which spike train belongs to which input
InputSpikeRefr=zeros(1,NCells);
% To keep track of the input spike train refractory periods)
tolos=0;
% (== time)of(l)ast)o(utput (s)poke (Keeping track of the output cell's refractory period )
Synapse=ones(1,NCells);
%Synaptic weights
sgEx=zeros(1,NCells);
sgIn=zeros(1,NCells);
% Vectors to save the group-wise synaptic conductances the cell experiences.
AveExCurr=zeros(1,NSigs);
AveInCurr=zeros(1,NSigs);
% Vectors to save the group-wise synaptic currents the cell experiences.
pre=zeros(1,NCells);
% A Vector to save the presynaptic learning trace.
Time=(1:NRuns);
Rate=zeros(1,NRuns);
%Vectors for plotting

```

%% InputGroups

```

temptype=0;
for i=1:NCells
    if (i<=NCells*ExFrac)
        if (mod(i-1,ExGroupsize)==0)
            temptype=temptype+1;
    end

```

```

        InputGroup(i)= temptype;
    else
        if (mod(i,InGroupsize)==0)
            temptype=temptype-1;
        end
        InputGroup(i)= -temptype;
    end
end
InputGroup(1000)=InputGroup(999);
% This routine assigns every spike train to an input group, starting with group 1 and ending
with group NSIGs for the excitatory spike trains and then going back from NSIGs to 1 for the
inhibitory spiketrains. To label inhibitory spike trains uniquely, their group identity is assigned
as a negative number.

```

%% Synapse Tuning

```

for i=1:800
    Synapse(i) = 0.3 + (1.1/(1+(InputGroup(i)-5)^4))+rand*0.1;
end
for i=801:NCells
    Synapse(i) = 0.1;
end
% This routine assigns a synaptic weight to every synapse. Excitatory synapses are tuned
according to their group identity (plus a noise term) to resemble the tuning reported in (7).
Inhibitory synapses are uniformly weak.

```

```

%%%%%%%%%%%%%
%% Simulation%%%%%
%%%%%%%%%%%%%

```

%% Initialize values

```

gEx=0;
% Overall excitatory synaptic conductance.
gIn=0;
% Overall inhibitory synaptic conductance.
gLeak=1;
% Leak conductance (Everything else is normalized in respect to the leak.)
V(1)=VRest;
% Membrane potential is initially at VRest.
post=0;
% Postsynaptic learning trace.
InputSpikeCount=0;
OutputSpikeCount=0;
AveCurrCounter=0;
%Counters for rates and averaging.
tRunning=0;
runcount=0;
going=1;

```

```
% Time parameters are initialized
```

%% Start of the loops

```
while(going>0)
    runcount=runcount+1;
% A counter to keep track of consecutive runs.
    OutputSpikeCount=0;
    for i=1:NSigs
        AveExCurr(i)=0;
        AveInCurr(i)=0;
    end
    AveCurrCounter=0;
% Reset the counters.

% %Uncomment from here to %%XX to plot individual synapse strengths in [pS]
%     figure(2)
%     clf(2)
%     plot(gBarEx*Synapse(1:800)*10000, 'r.')
%     hold on
%     plot(-(-796:4:1),gBarIn*Synapse(801:NCells)*10000, 'g.')
% %%XX end of commented section
% Uncommenting this routine plots the synaptic weights in a second window.
```

```
for t=2:length(Timevector)
% The time loop begins here.
    tRunning=tRunning+dt;
    gEx = gEx*expGEx;
    gIn = gIn*expGIN;
    for i=1:NSigs
        sgEx(i) = sgEx(i)*expGEx;
        sgIn(i) = sgIn(i)*expGIN;
    end
% The synaptic conductances decay exponentially towards 0.

    for i=801:NCells
        pre(i)= pre(i)*expPlasticity;
    end
    post=post*expPlasticity;
% The learning traces decay exponentially towards 0.
```

%% Input Signals

```
for i=1:NSigs
    re=rand-0.5;
    FilteredWhiteNoise(i) = re -(re - FilteredWhiteNoise(i))*expnoise;
    Input(i)=Backgroundrate + ...
        max(0, Maxrate*FilteredWhiteNoise(i))/ApproximateNormalizingFactor;
```

```

    end
%
% At every time step the current input for each signal is calculated from a time-filtered
% input signal. Signals traces are not saved to increase simulation speed.
%
% For more details, please see 6.2.1

```

```

% % % Uncomment from here to %%%YY to plot single EPSPs at VRest
%
%     for i=1:NSigs
%         Input(i)=0;
%     end
%     if(t==300/dt)
%         gEx = gEx + gBarEx;
%     end
%     if(t==700/dt)
%         gIn = gIn + gBarIn;
%     end
%     stopnow=1;
%%%%%%YY

```

% Uncommenting this routine makes it possible to evaluate single EPSPs by switching off the input signals and instead injecting one excitatory and one inhibitory PSP (at 300 and 700 ms) into the cell. "stopnow" is a flag that stops the simulation after one run and rescales the axes of the voltage plot.

%% Presynaptic spike trains

```

for i=1:NCells
    if (rand < Input(abs(InputGroup(i))) && InputSpikeRefr(i)<=0)
%
% If a Spiketrain fired a spike, (see also 6.2.1) ...
    if(InputGroup(i)>0)
        .. and if it is excitatory...
        gEx = gEx + (gBarEx * Synapse(i));
        ... increase the excitatory synaptic conductance variable according to
        the strength of the synapse
        sgEx(abs(InputGroup(i)))=sgEx(abs(InputGroup(i))) + gBarEx*Synapse(i);
%
% (Keeping track of the synaptic conductances group-wise for plotting.)
    else
        otherwise (meaning the synapse is inhibitory)...
        gIn = gIn + gBarIn * Synapse(i);
        ... increase the synaptic conductance by the synapse strength
        sgIn(abs(InputGroup(i)))=sgIn(abs(InputGroup(i))) + gBarIn*Synapse(i);
%
% (To keep track of the synaptic conductances group-wise for plotting.)
        pre(i)= pre(i) + eta;
%
% Update the presynaptic learning trace.
        Synapse(i)=Synapse(i) + post - alpha;
%
% Update the synaptic strength according to the rule.
        → add the effect of proximate post synaptic spikes, and subtract  $\alpha$ .
        if(Synapse(i) <=0)
            Synapse(i)=0;

```

```

        end
%
        Ensure non-negative synaptic weights.
    end
    InputSpikeRefr(i)=tRef;
%
... Also: set the refractory time counter to the value of refractoriness
InputSpikeCount=InputSpikeCount+1;
.. and count the overall number of input spikes
else
%
meaning if no presynaptic spike was evoked:
    InputSpikeRefr(i)=InputSpikeRefr(i)-dt;
%
subtract dt from the refractory counter.
end
end

```

%% Membrane potential and postsynaptic spikes.

```

if ((tRunning - tolos) < tRef)
    V(t) = VRest;
%
If the Cell is refractory, keep V at Vrest
else
%
Meaning: if the cell is not refractory, ...
    gTot = gLeak + gEx + gIn;
%
calculate the total membrane conductance,
tauEff=taumem/gTot;
%
and the effective time constant, as well as...
    VInf = ((gLeak*VRest + gEx * EAMPA+ gIn*EGABA)/gTot);
%
the membrane potential that V strives towards.
    V(t) = VInf + (V(t-1) - VInf)*exp(-dt/tauEff);
%
Use the above to update the membrane potential
for i=1:NSigs
    AveExCurr(i)= AveExCurr(i) + sgEx(i)*(V(t)-EAMPA);
    AveInCurr(i)= AveInCurr(i) + ...
        sgIn(i)*(V(t)-EGABA) + (gLeak*(V(t)-VRest))/NSigs;
end
AveCurrCounter=AveCurrCounter+1;
%
The above routine keeps track for the group-wise input currents for plotting but
does not affect the behavior of the cell. We divide the (mostly inhibitory acting)
leak current evenly to all Groups,since each input signal causes the same absolute
amount of leak current (by deflecting the membrane potential away from rest with
identical statistics over time for each signal).
end

if (V(t)>Vth)
%
If the membrane potential hits threshold...
    tolos=tRunning;
%
... set the refractory counter to the current time step
    V(t-1)=0;
%
... set the last membrane potential before the spike to zero (for plotting)
    V(t)=VRest;

```

```

%
% ... reset the current membrane potential to Vrest.
% OutputSpikeCount=OutputSpikeCount+1;
%
% ... count the spike.
post = post + eta;
%
% ... update the postsynaptic learning trace
for i=801:NCells
    Synapse(i) = Synapse(i)+pre (i);
end
%
% update all synapses according to the rule
% → Add the effect of proximate presynaptic spikes.
end
%
% subtract dt from the refractory counter.
Exkeep(t)=gEx*(V(t)-EAMPA);
Inkeep(t)=gIn*(V(t)-EGABA);
%
% For plotting purposes, keeps track of the excitatory and inhibitory synaptic currents.
% Because everything is normalized by the leak conductance, the values are saved in
% units of  $10nS \times mV$  ( $= 10^{-11} \text{ Amp} = [10pA]$ ).
%
% To plot in units of nA, one has to divide this number by 100.
end
%
% End of the time loop
Rate(runcount)=OutputSpikeCount/duration*1000;
Time(runcount)=runcount*duration/1000;

```

```

%%%%%%%%%%%%%%%
% Plotting %%%%%%
%%%%%%%%%%%%%%%

```

```

figure(1)

subplot(3,14,[1 2 3 4 15 16 17 18])
hold off
plot(Timevector, Exkeep/100,'k', 'LineWidth', 1)
%
% plotting in units of nA hence the values need to be divided by 100, see above.
hold on
plot(Timevector,Inkeep/100,'Color', [0.6 0.6 0.6], 'LineWidth', 1)
plot(Timevector,(Inkeep+Exkeep)/100,'Color', [0 0.6 0])
axis([0 duration -3 3]);
ylabel('Synaptic Currents [nA]');

subplot(3,14,[29 30 31 32])
if (stopnow==0)
    plot(Timevector, V, 'k')
end
axis([0 duration -65 5]);
ylabel('Mem. Pot. [mV]');
xlabel('Time [ms]');

```

```

subplot(3,14, [6 7 8 9 20 21 22 23 34 35 36 37])
hold off
plot((1:8), (-AveExCurr/AveCurrCounter)/100,'-ks','LineWidth',2, ...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','k',...
      'MarkerSize',10)
hold on
plot((1:8), (AveInCurr/AveCurrCounter)/100,'-ks','LineWidth',2, ...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','w',...
      'MarkerSize',10)
axis([1 8 0 0.25]);
ylabel('Mean Synaptic Currents [nA]');
xlabel('Signal Number');

subplot(3,14,[11 12 13 14 25 26 27 28 39 40 41 42])
hold on
plot(runcount*duration/1000, mean(Synapse(925:949))*gBarIn*10000, ...
      '.', 'Color','r')
% (Plotted in units of pS)
plot(runcount*duration/1000, mean(Synapse(900:924))*gBarIn*10000,... ...
      '.', 'Color',[0.5 0.0 0.5])
% (Plotted in units of pS)
plot(runcount*duration/1000, mean(Synapse(875:899))*gBarIn*10000, ...
      '.', 'Color','b')
% (Plotted in units of pS)
ylabel('Mean Synaptic Strength [pS]');
axis([0 NRuns*duration/1000 0 800]);
ax1 = gca;
set(ax1,'XColor','k','YColor',[0.5 0.0 0.5], 'YAxisLocation','right')
ax2 = axes('Position',get(ax1,'Position'),...
           'XAxisLocation','bottom',...
           'YAxisLocation','left',...
           'Color','none',...
           'XColor','k' , 'YColor','k');
hold(ax2,'on')
plot(Time(1:runcount), Rate(1:runcount), '-.k','Parent',ax2)
ylabel('Mean Output Rate [Hz] ','Parent',ax2);
xlabel('Time [s]');
axis([0 NRuns*duration/1000 0 100]);

if(runcount>NRuns)
    going=0;
end
if (stopnow==1)
    going=0;
end
% (This is only ==1 when the EPSP routine is uncommented)

```

```

end
% End of the while loop

```

% End of Code

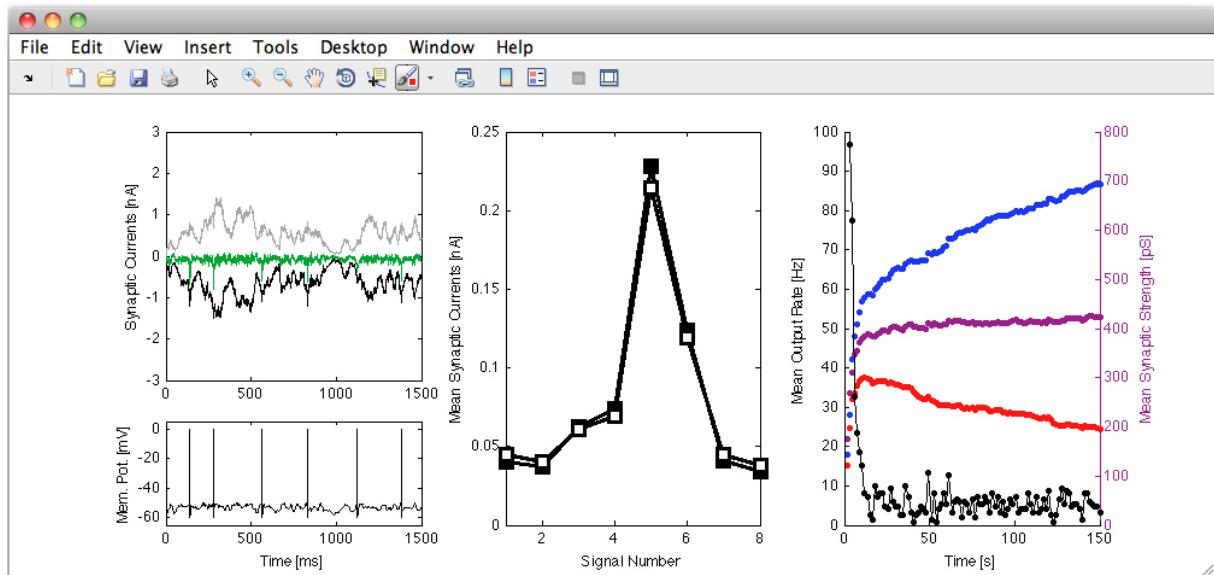


Figure S11: Screenshot of figure generated by the single cell simulation code in the technical appendix 5.2.2. Panels are similar to Fig. 1 and show qualitatively identical results.