Champalimaud Foundation

Christian Machens

ENS ÉCOLE NORMALE SUPÉRIEURE

# Efficient coding in spiking networks

## A normative account for E/I balance


Sophie Deneve

Ecole Normale Supérieure, Paris



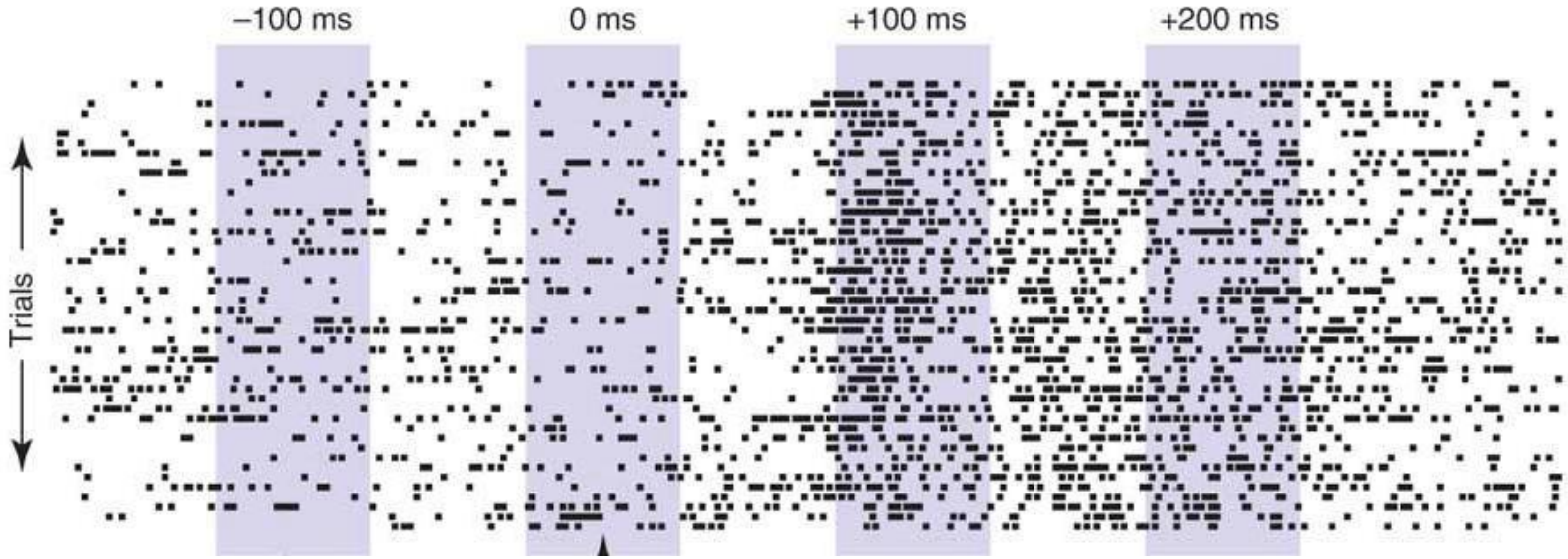
Agence Nationale pour la Recherche (ANR)
James S. McDonnell Foundation
European research council (ERC)


Agence Nationale de la Recherche ANR

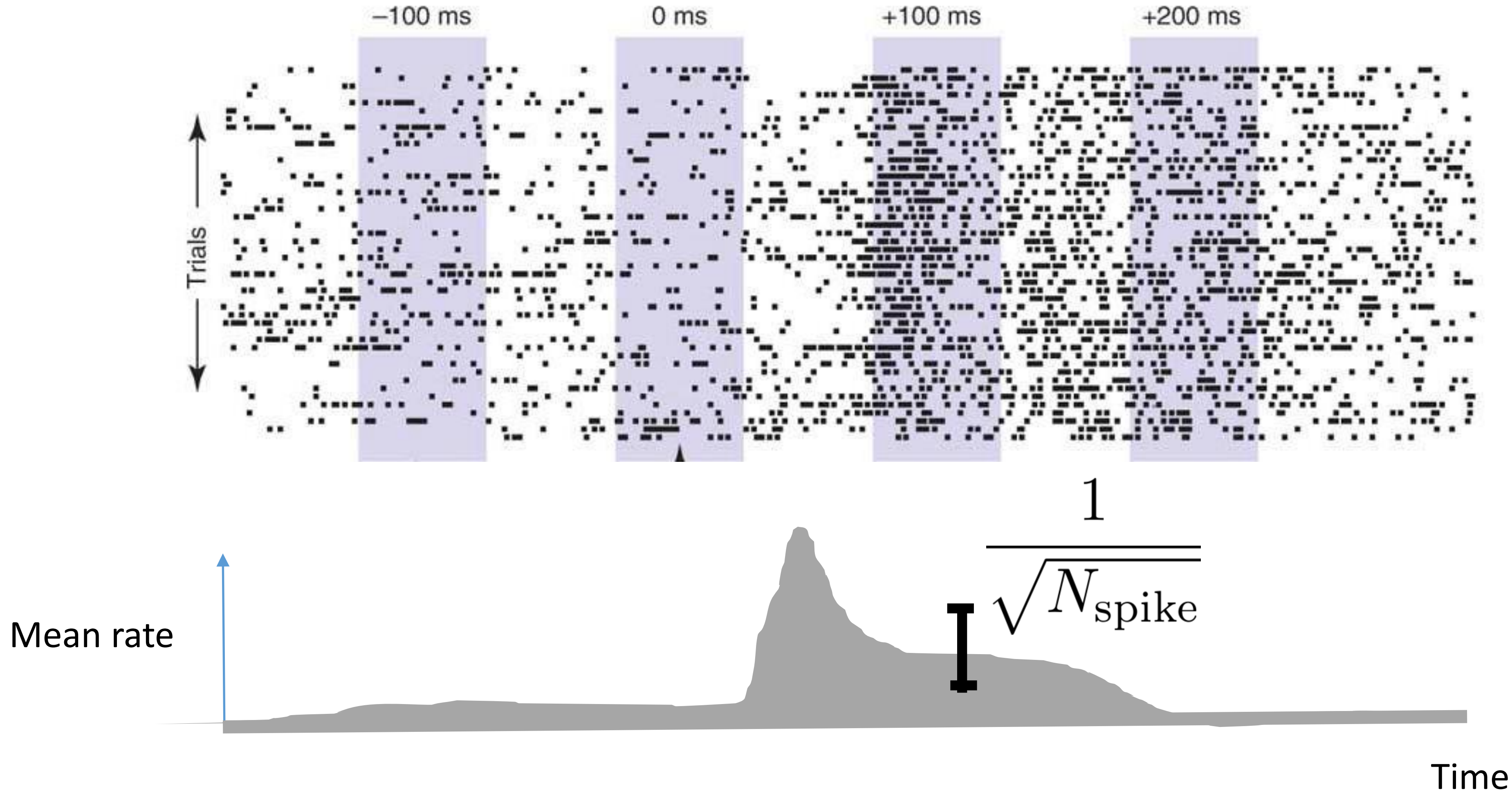
James S. McDonnell Foundation

erc European Research Council

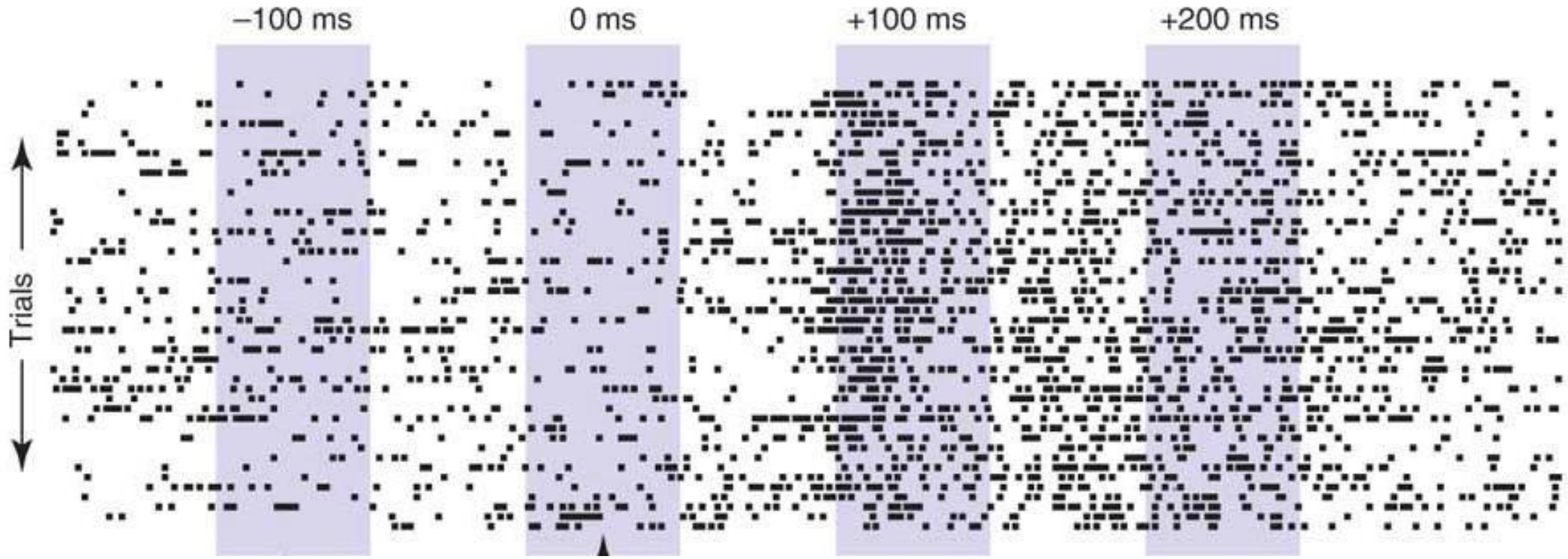
# Cortical neural responses are extremely variable



# Cortical neural responses are extremely variable



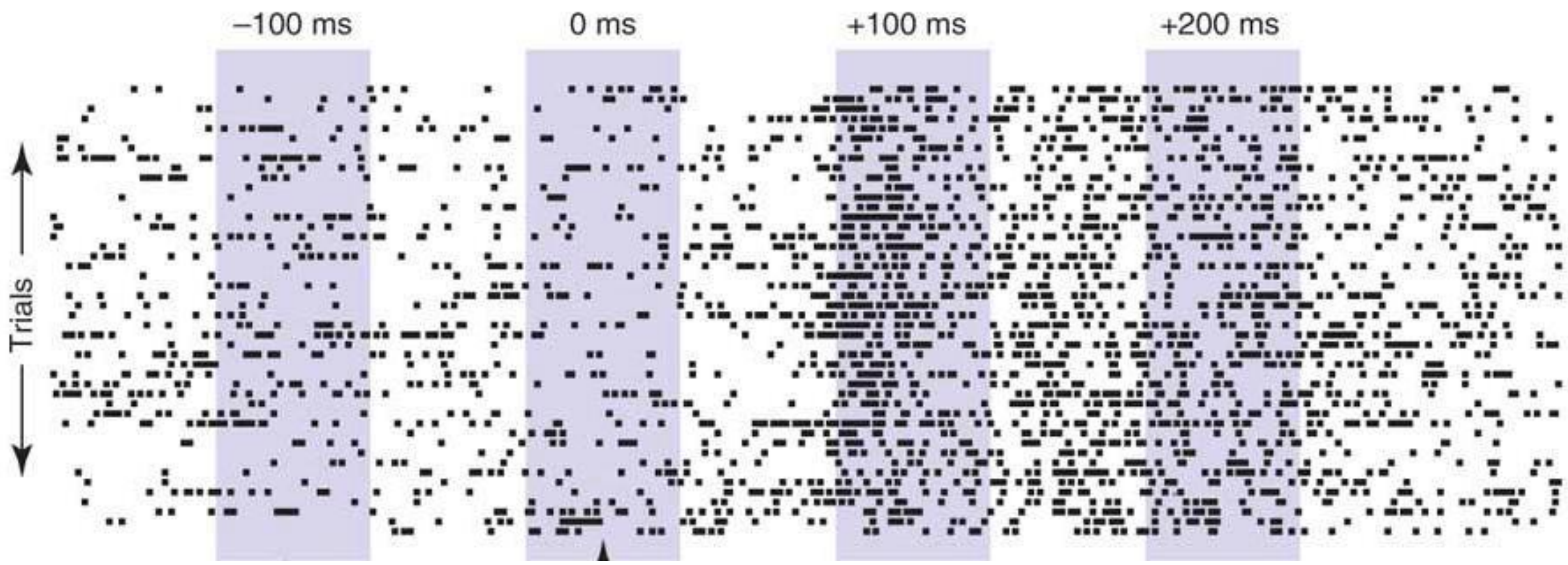
# Cortical neural responses are extremely variable




How?

Excitatory – Inhibitory balance

# Cortical neural responses are extremely variable



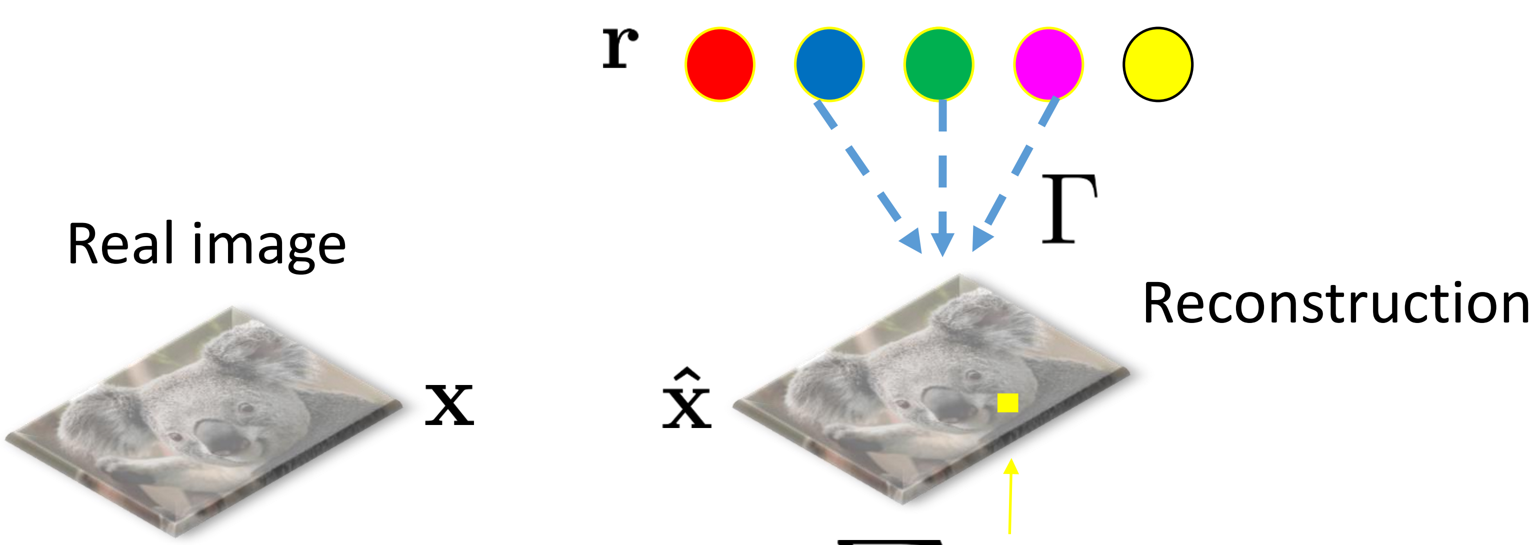

WHY?????

# Efficient neural coding

Real image

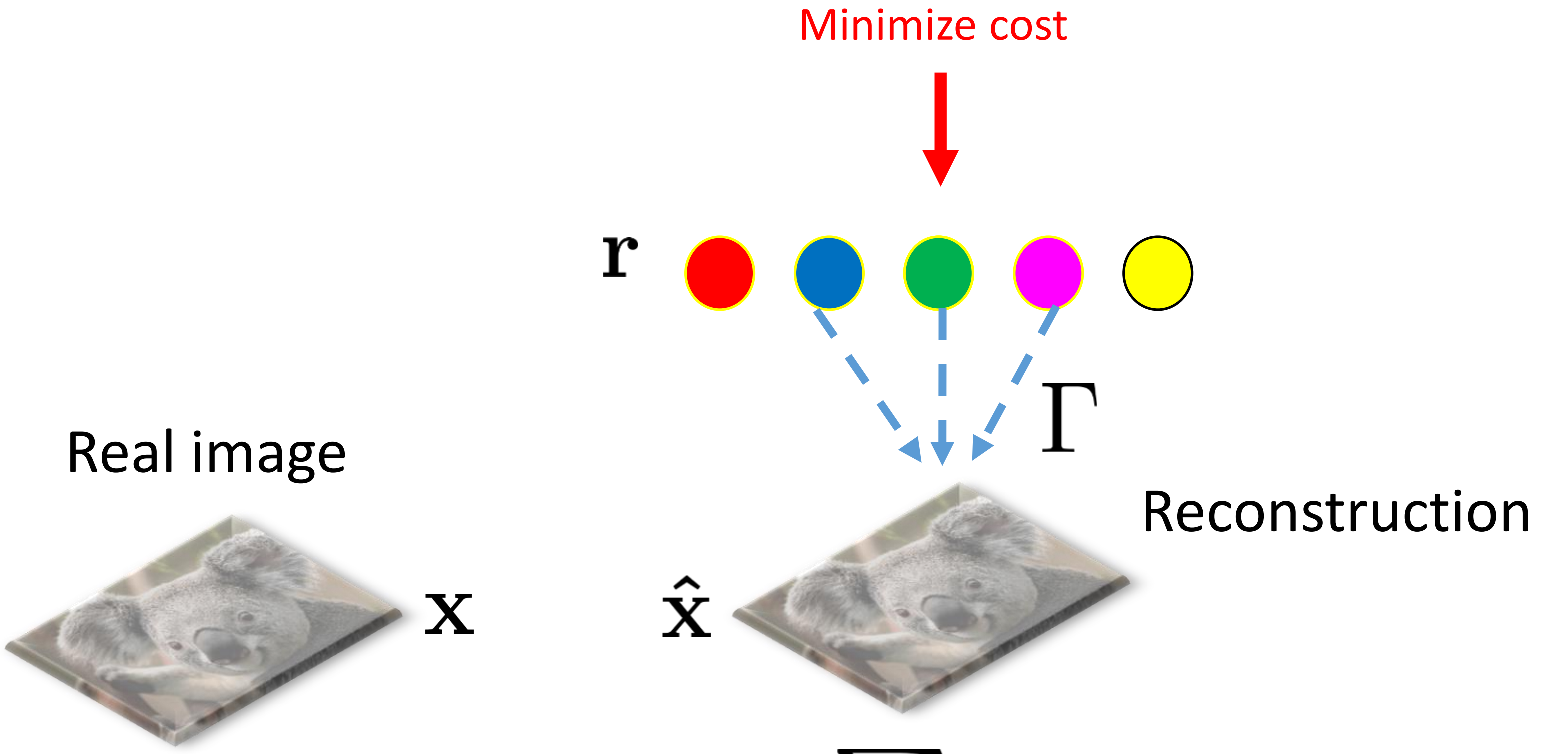
$\mathbf{x}$

# Efficient neural coding



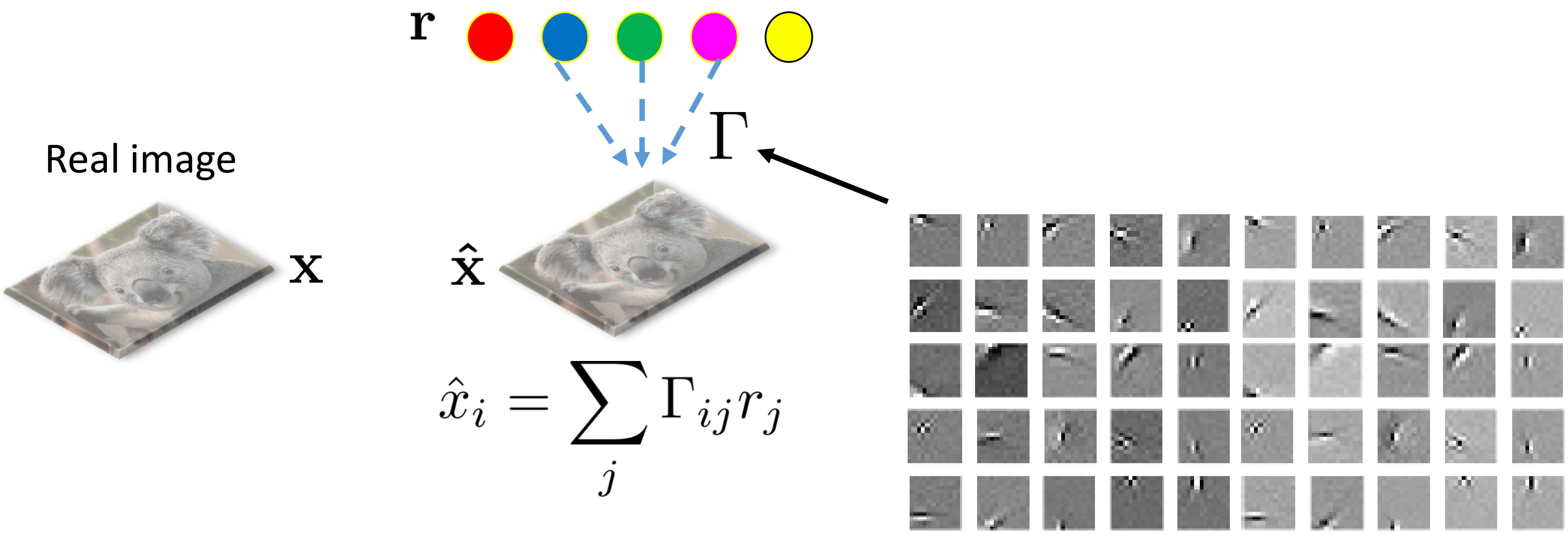

$$\hat{x}_i = \sum_j \Gamma_{ij} r_j$$

# Efficient neural coding



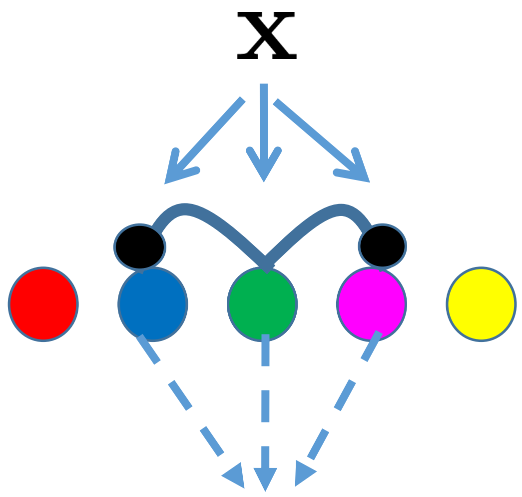

$$\hat{x}_i = \sum_j \Gamma_{ij} r_j$$

# Efficient neural coding



# Efficient **population** coding

$$(\mathbf{r}, \Gamma) = \underset{\mathbf{r}^*, \mathbf{\Gamma}^*}{\arg\min} \left( \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \mathrm{Cost}(\mathbf{r}^*) \right)$$

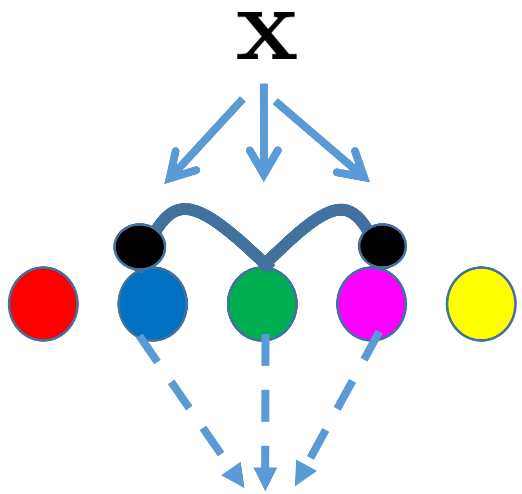


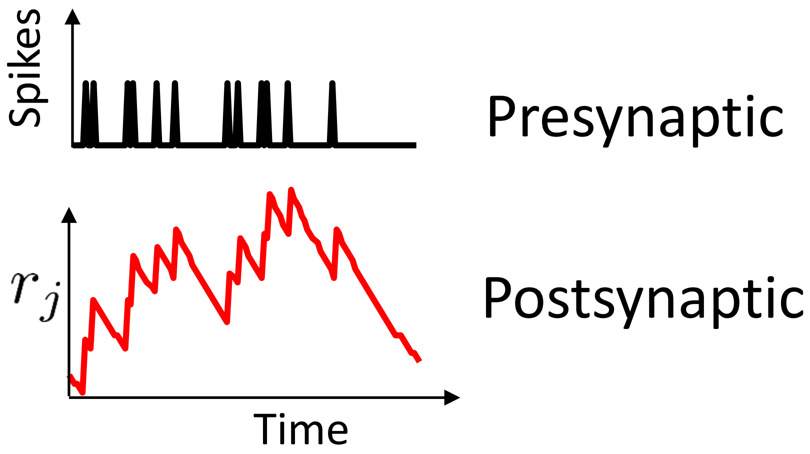

$$\hat{x}_i = \sum_j \Gamma_{ij} r_j$$

**Linear Decoder**

# Efficient **population** coding **WITH SPIKES**

$$(\mathbf{r}, \Gamma) = \underset{\mathbf{r}^*, \mathbf{\Gamma}^*}{\arg\min} \left( \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \mathrm{Cost}(\mathbf{r}^*) \right)$$

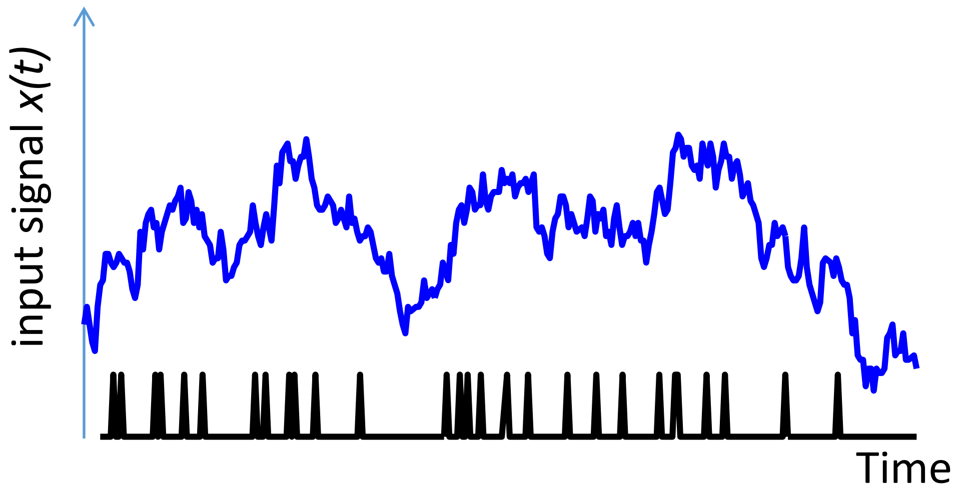


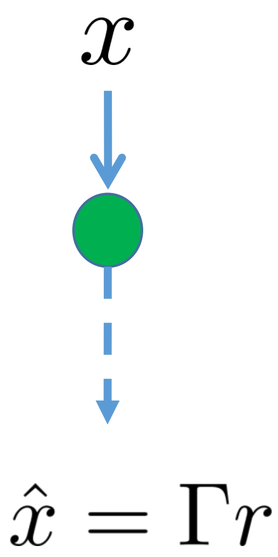




$$\hat{x}_i = \sum_j \Gamma_{ij} r_j$$

**Linear Decoder**

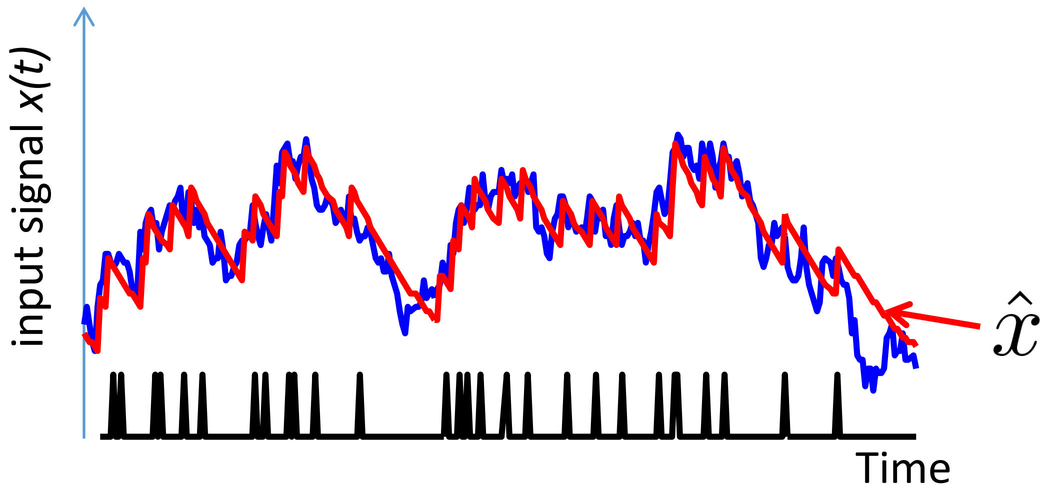
# Single Neuron



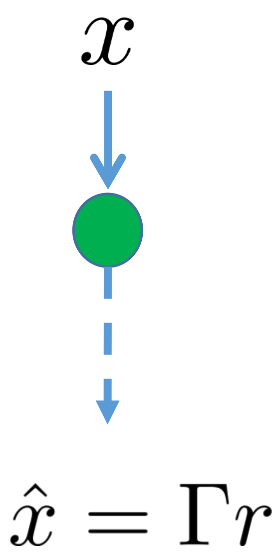




When to spike?

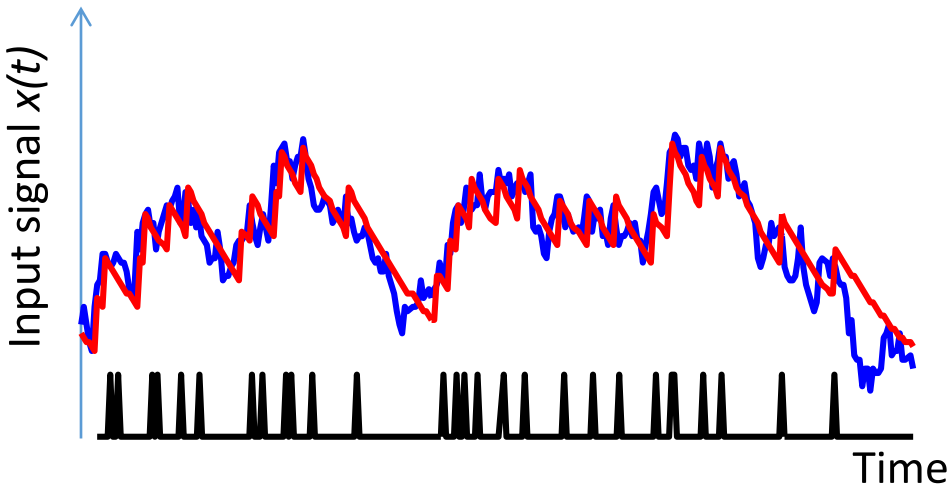
# Single Neuron




When to spike?



# Single Neuron

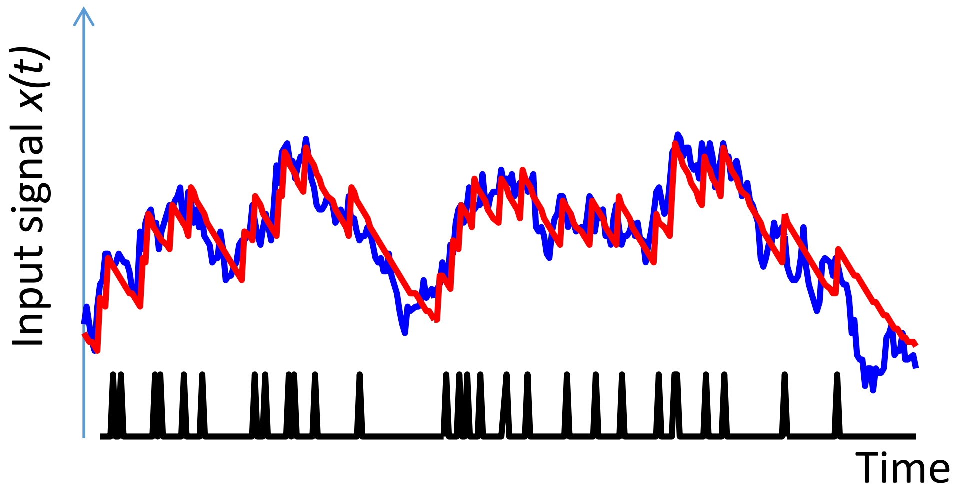



When to spike?

Minimize:

$$E = (x - \hat{x})^2$$

# Single Neuron




Minimize:

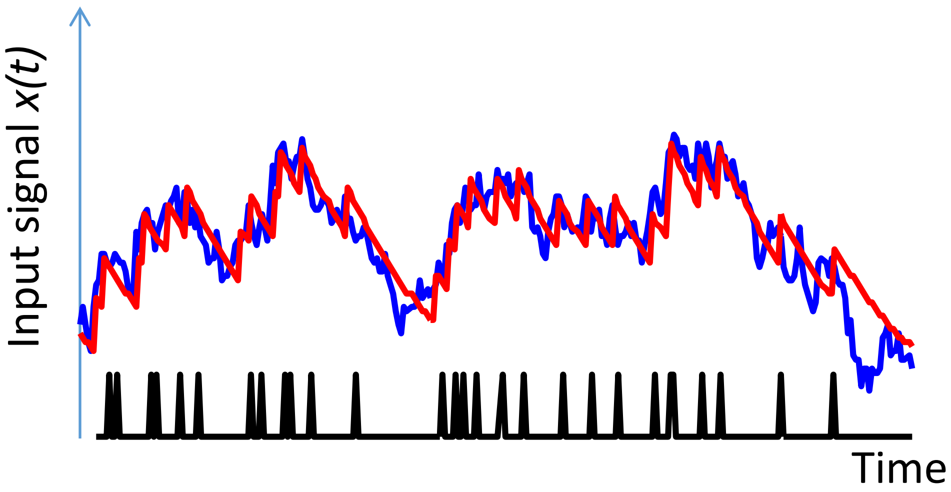
$$E = (x - \hat{x})^2$$

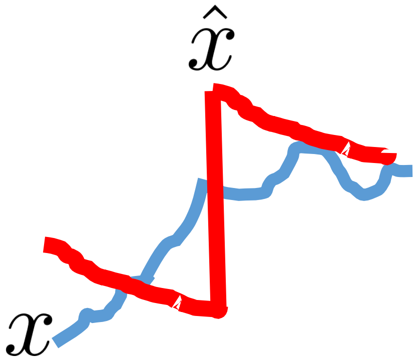
Greedily spike when

$$E^{\text{spike}} < E^{\text{nospike}}$$

$$(x - \hat{x} - \Gamma)^2 - (x - \hat{x})^2 < 0$$

# Single Neuron




Minimize:

$$E = (x - \hat{x})^2$$

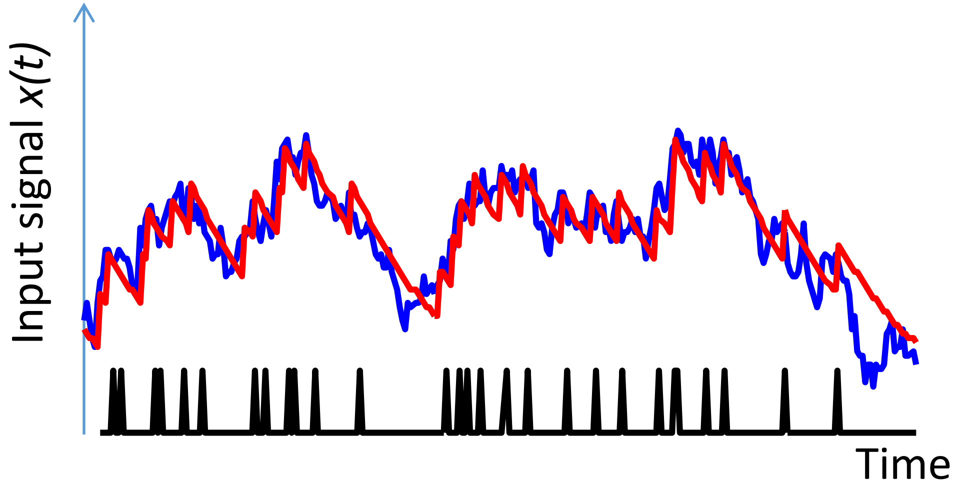
Greedily spike when

$$E^{\text{spike}} < E^{\text{nospike}}$$

$$x - \hat{x} > \frac{\Gamma}{2}$$

Decoding error Threshold

# Single Neuron

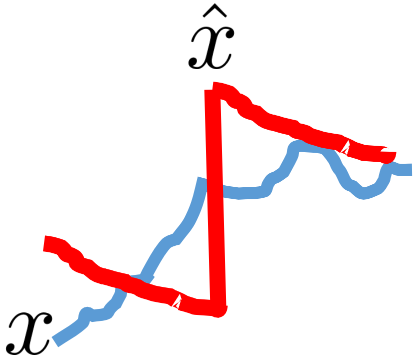


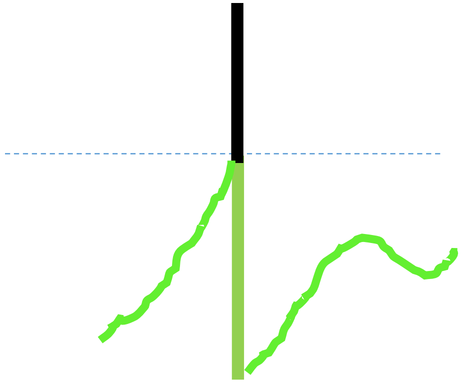

Minimize:

$$E = (x - \hat{x})^2$$

Greedily spike when

$$E^{\text{spike}} < E^{\text{nospike}}$$



$$V = x - \hat{x}$$

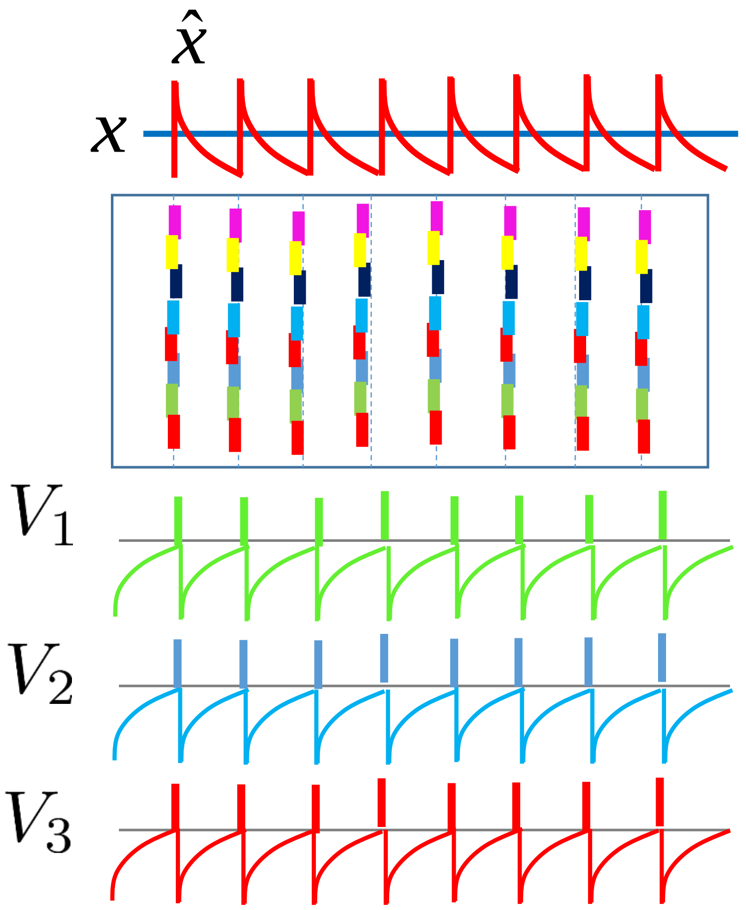
$$V = x - \hat{x} > \frac{\Gamma}{2}$$

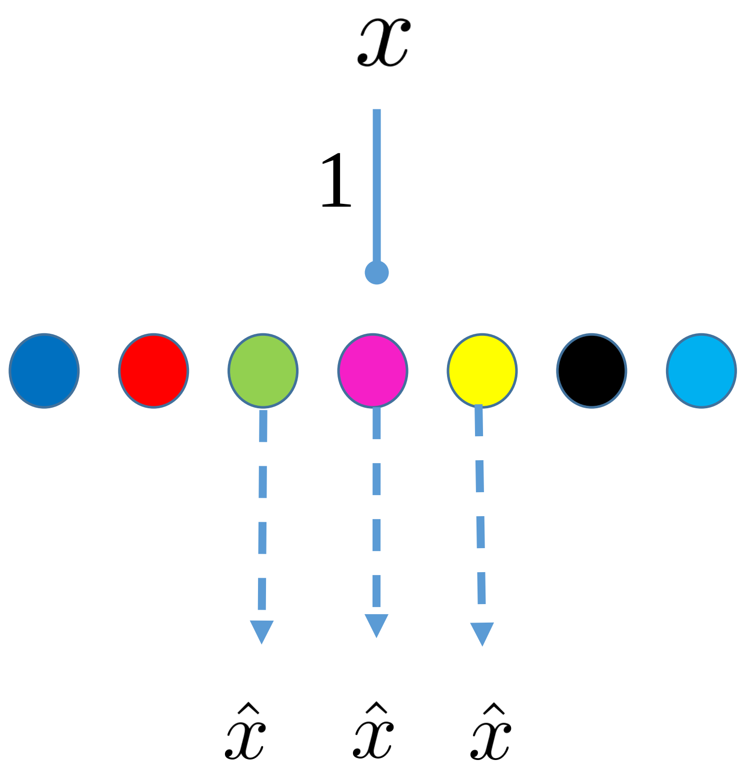
Membrane potential

Decoding error

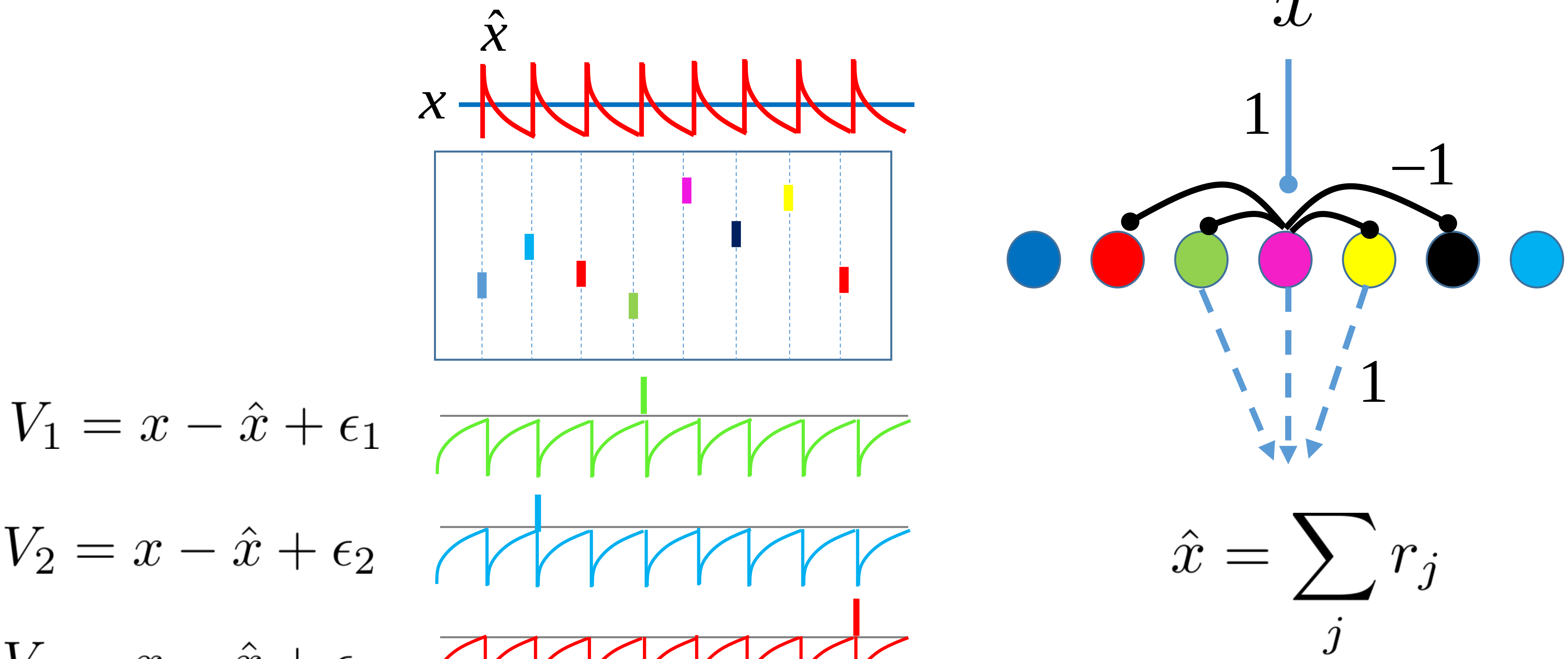
Threshold

# Homogeneous Network



# Homogeneous Network

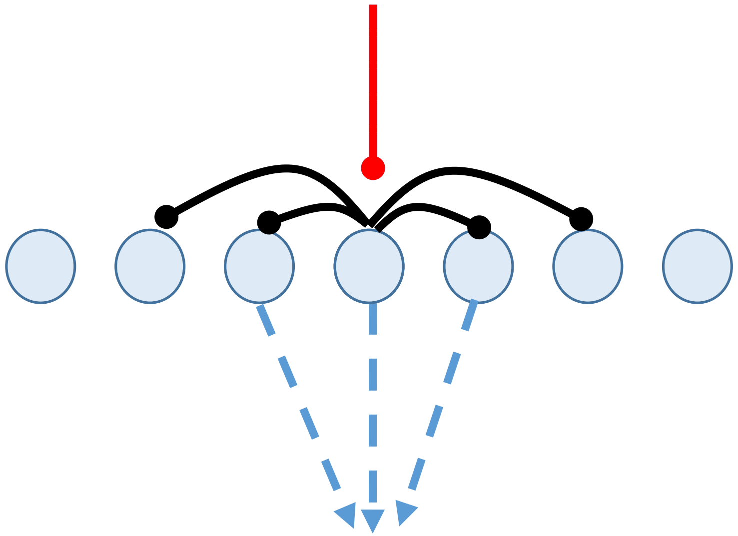


# General case

Minimize:

$$E = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \mathrm{Cost}(\mathbf{r})$$

$$\mathbf{x} = [x_1, \ldots, x_J]$$



$$\hat{x}_i = \sum_j \Gamma_{ij} r_j$$

# General case

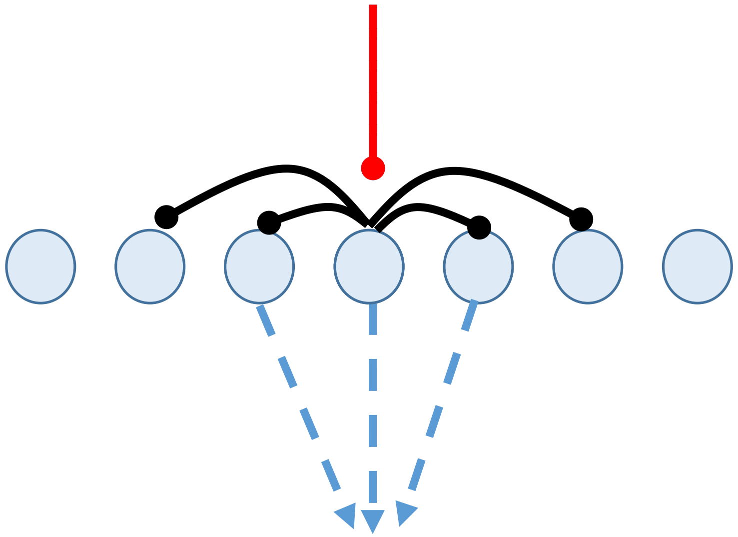
Minimize:

$$E = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \mathrm{Cost}(\mathbf{r})$$

Greedy spike rule:

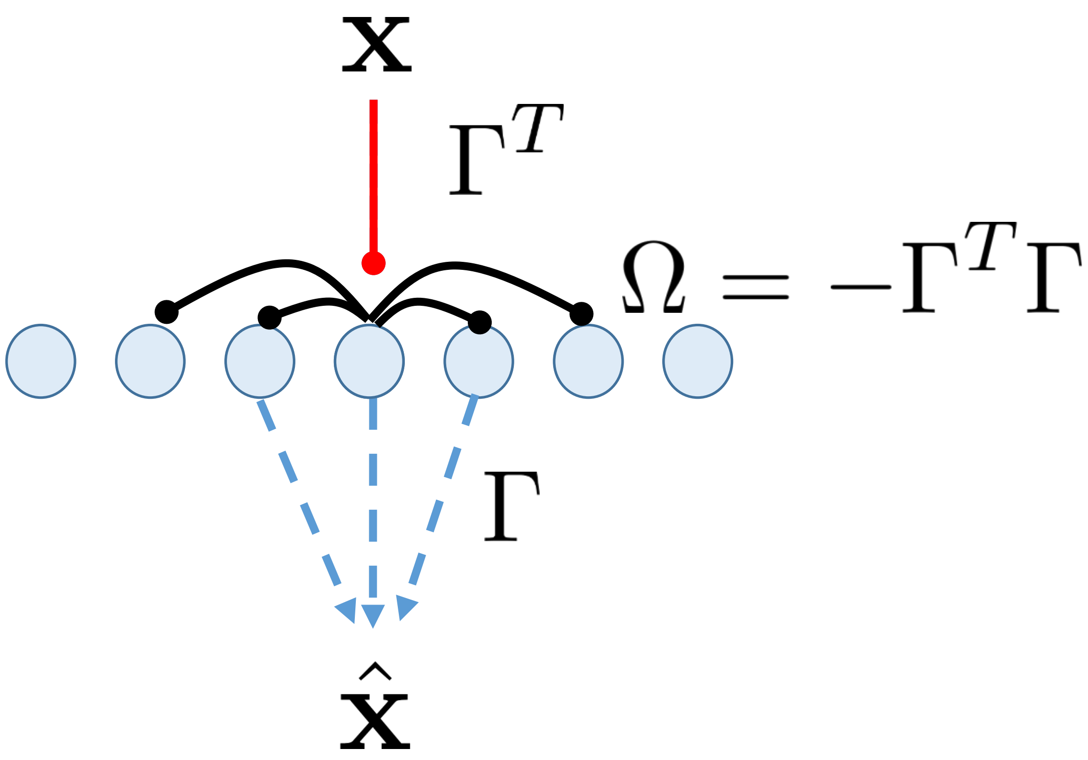
$$E^{\mathrm{spike}\, j} < E^{\mathrm{no\, spike\, j}}$$

$$\mathbf{x} = [x_1, \ldots, x_J]$$

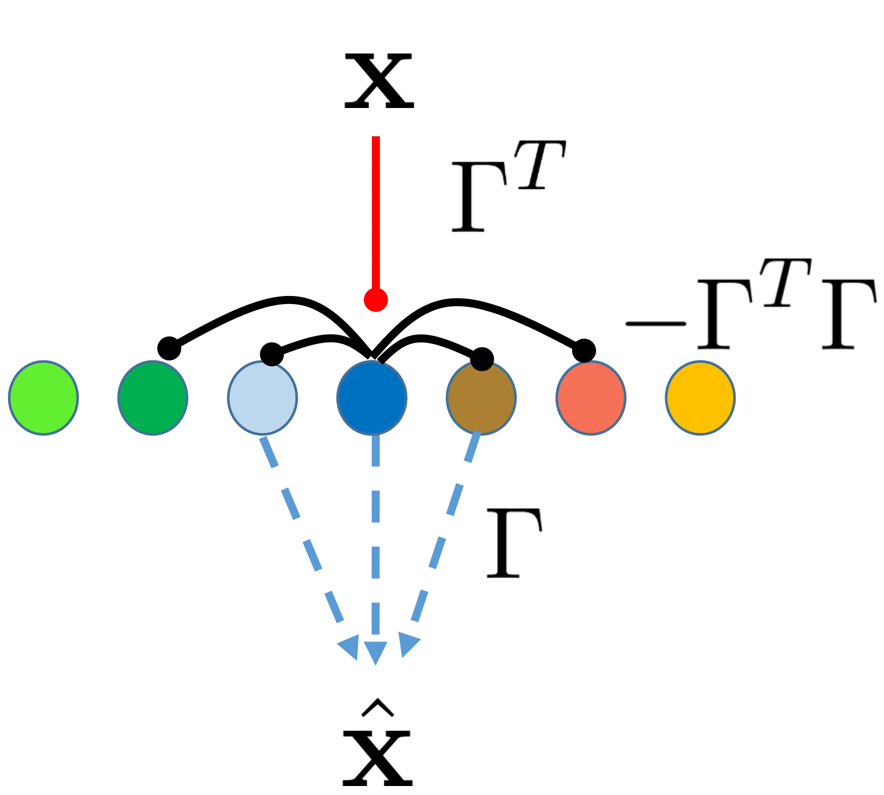


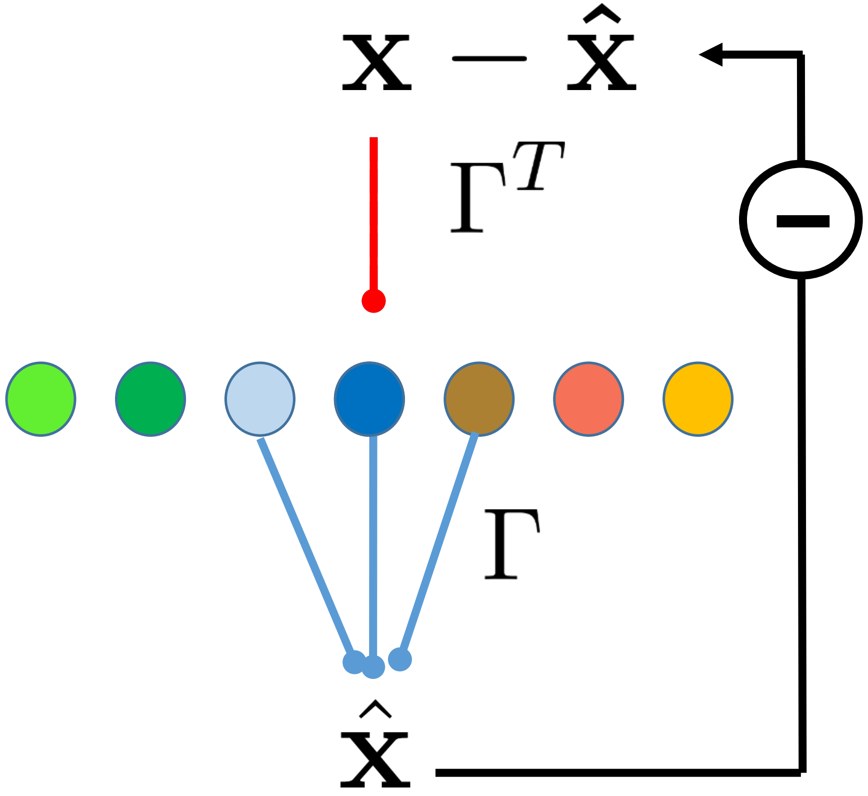
$$\hat{x}_i = \sum_j \Gamma_{ij} r_j$$

# General case: efficient LIF network

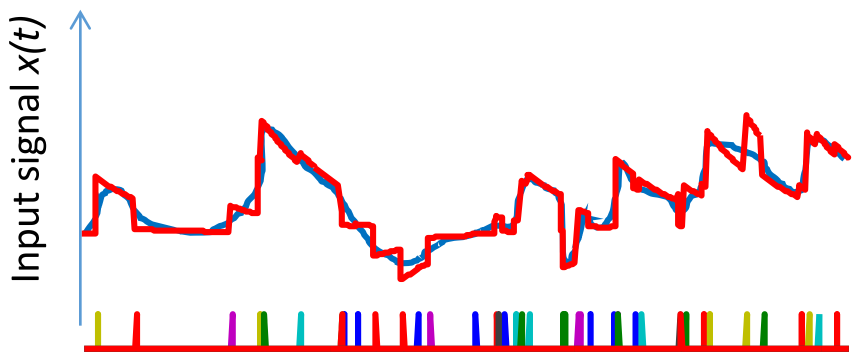


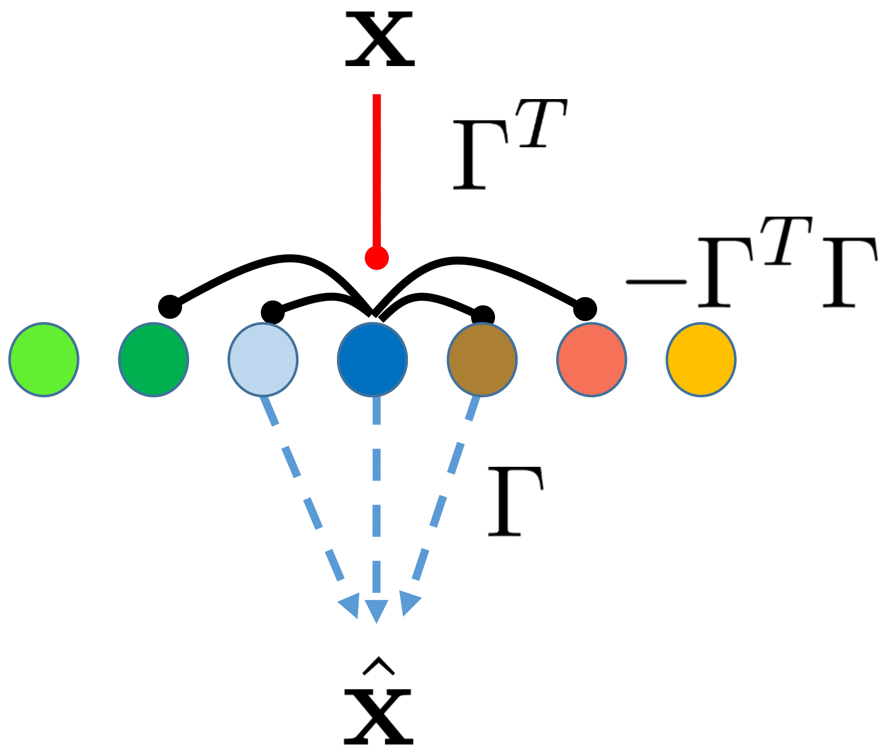
# Equivalent to predictive encoder



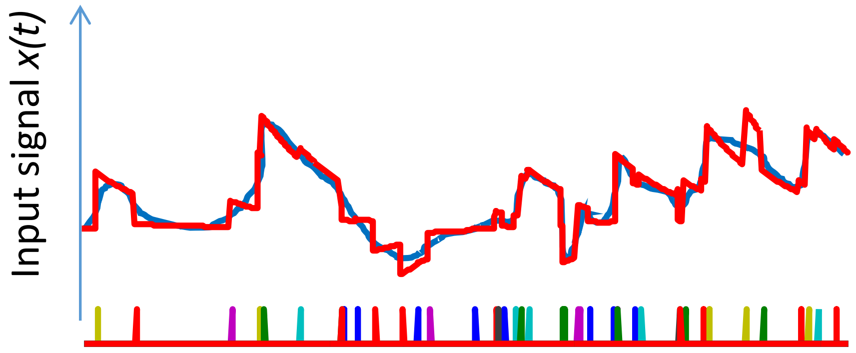



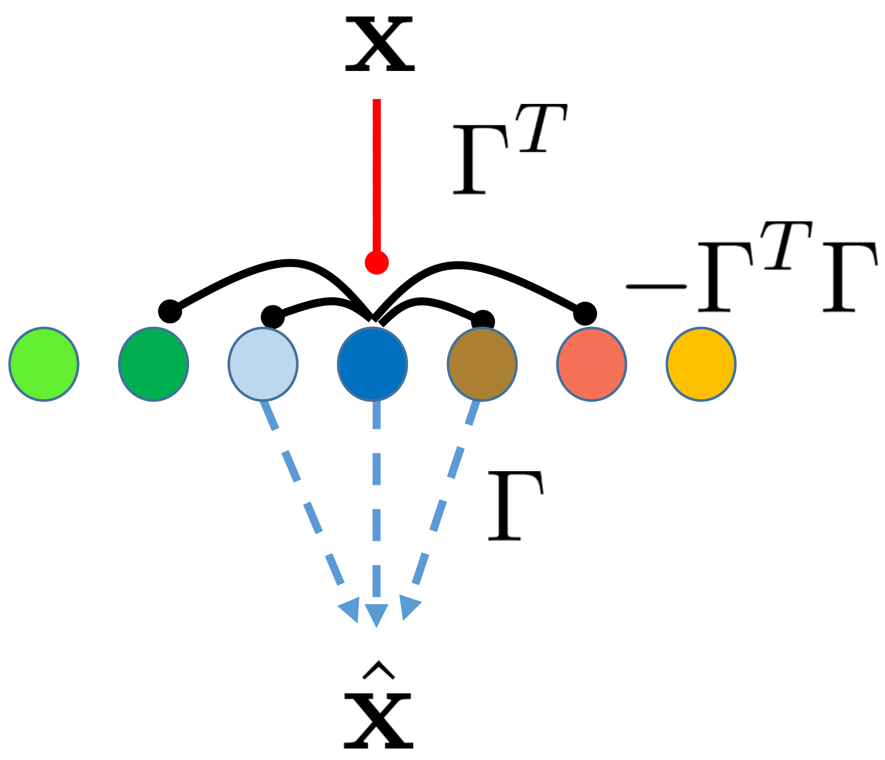
# Spike-based population coding

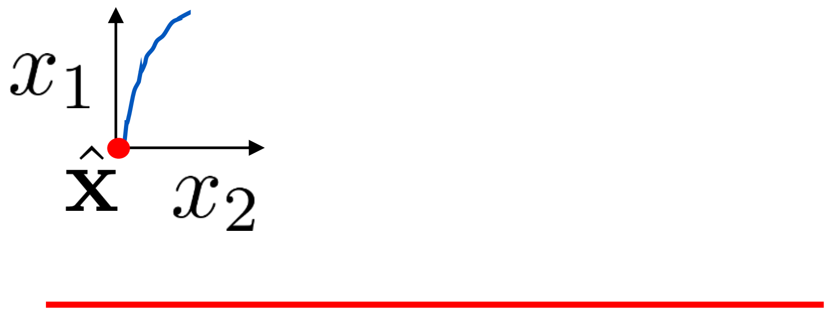


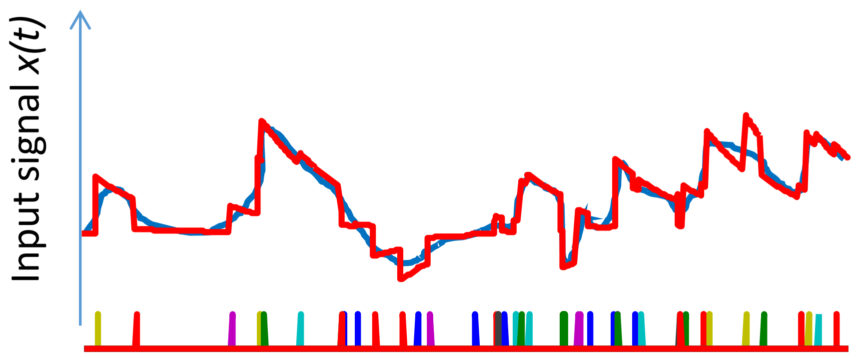
# Spike-based population coding

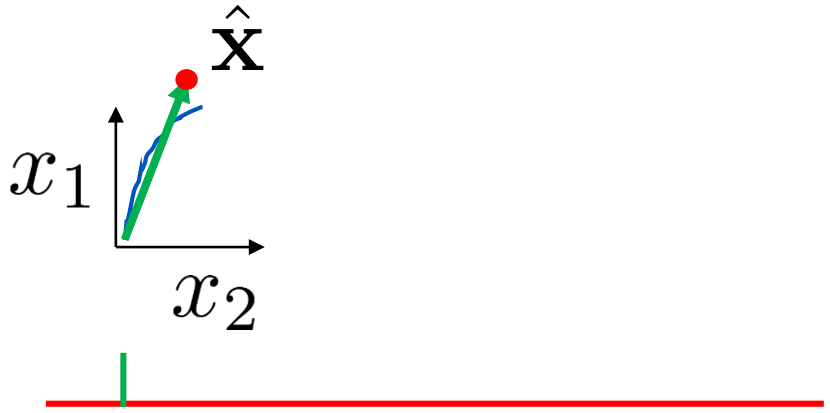


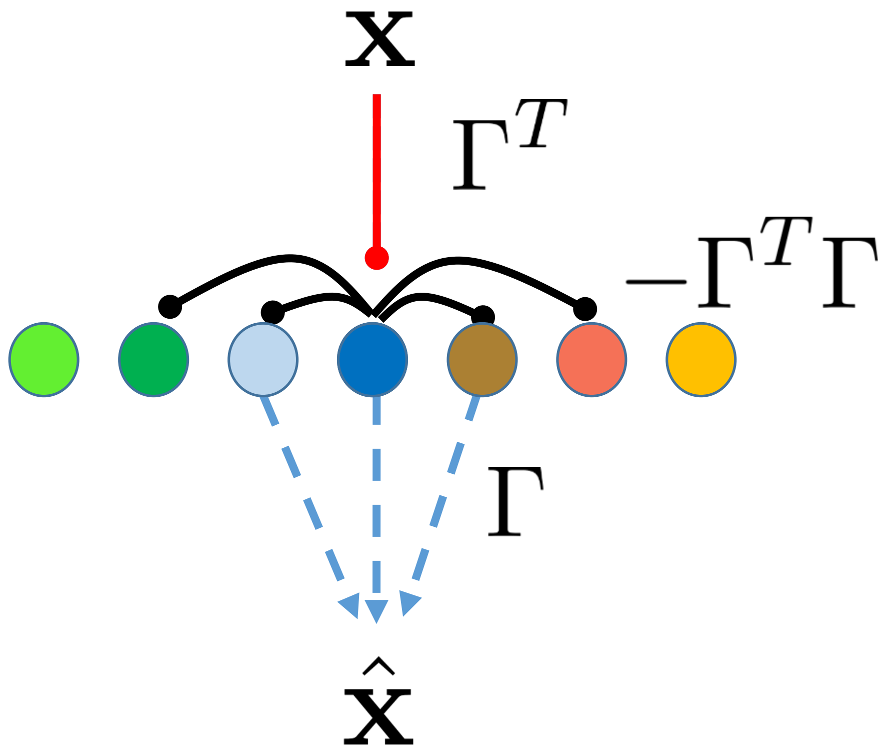



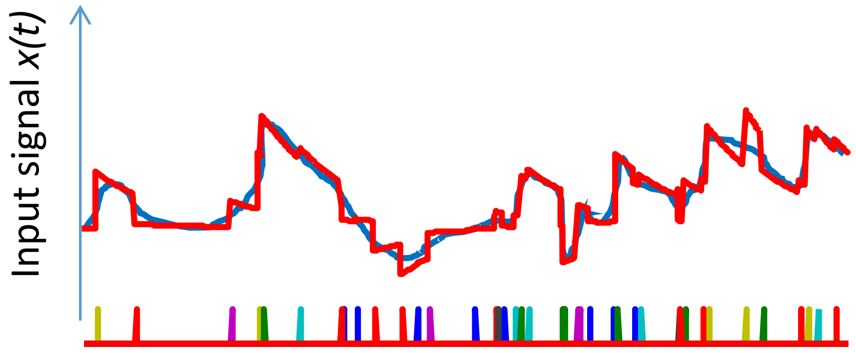
# Spike-based population coding

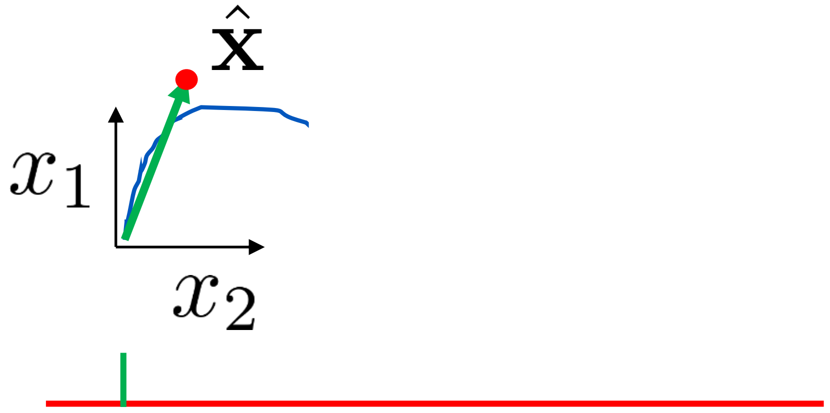


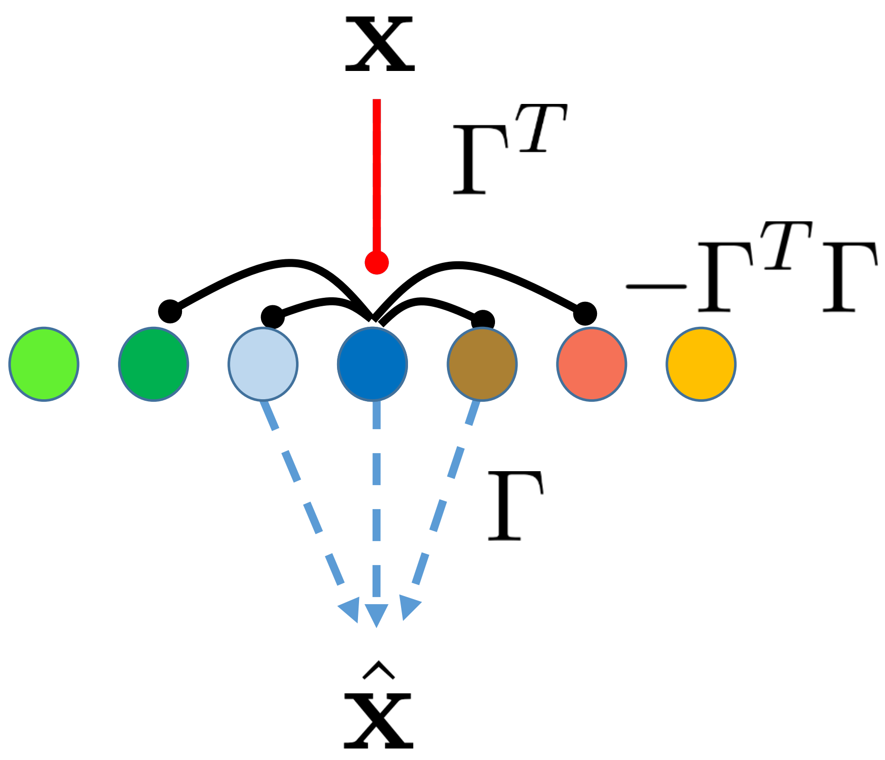



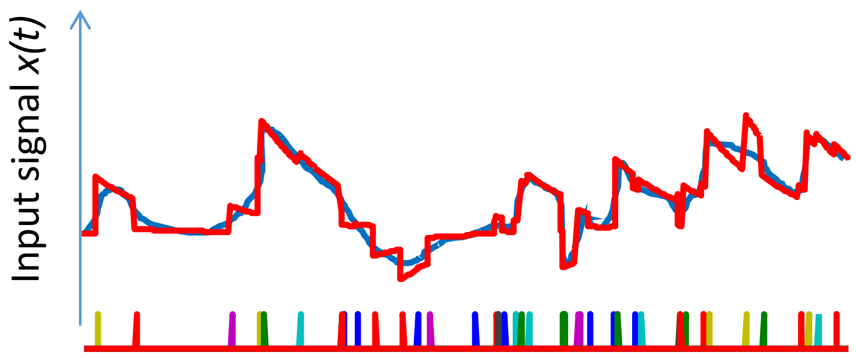
# Spike-based population coding

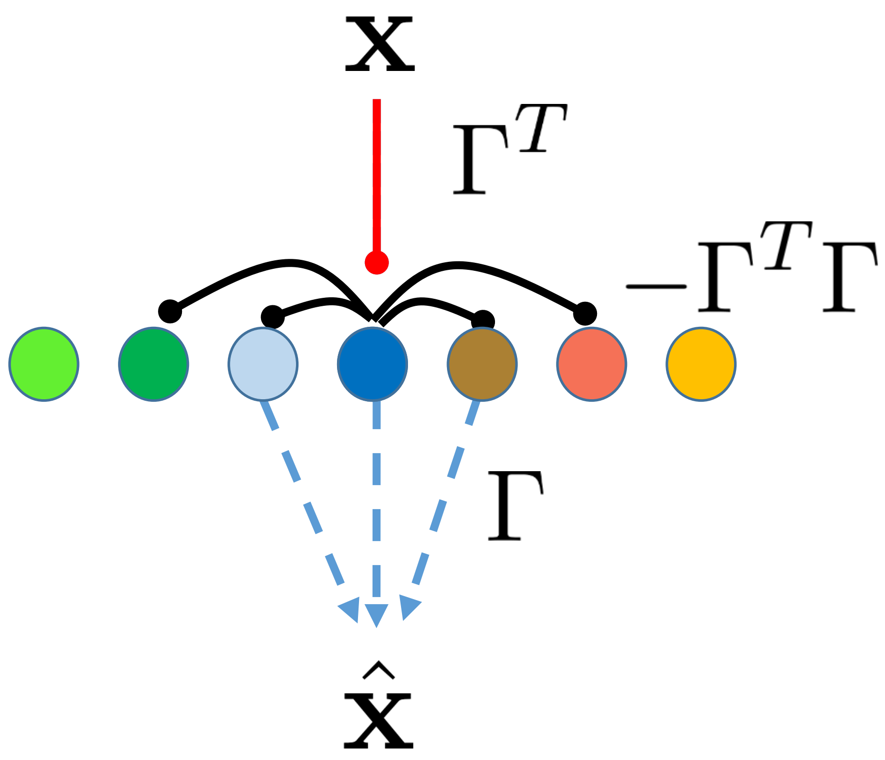


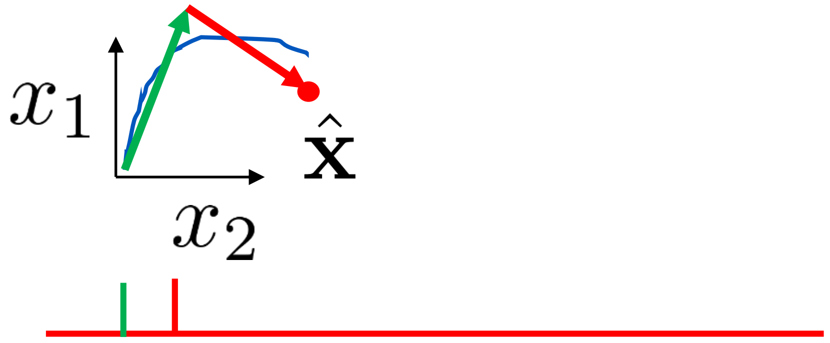



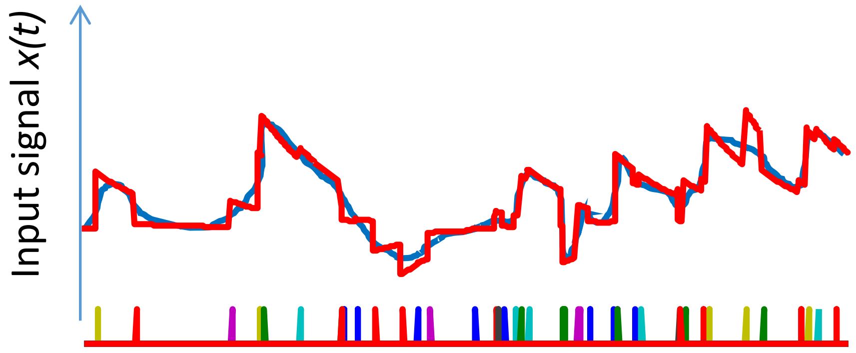
# Spike-based population coding

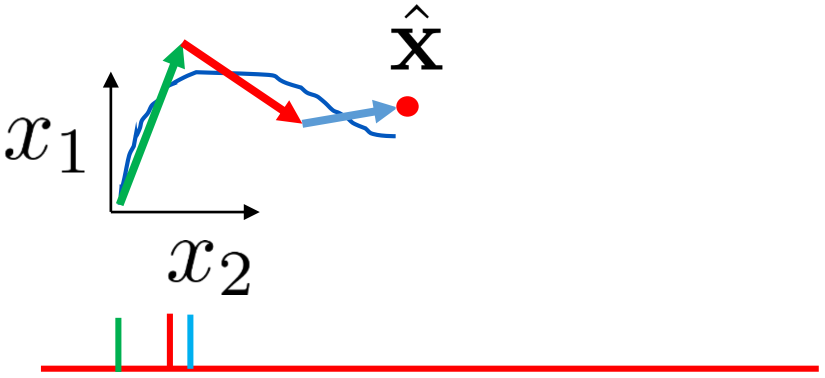


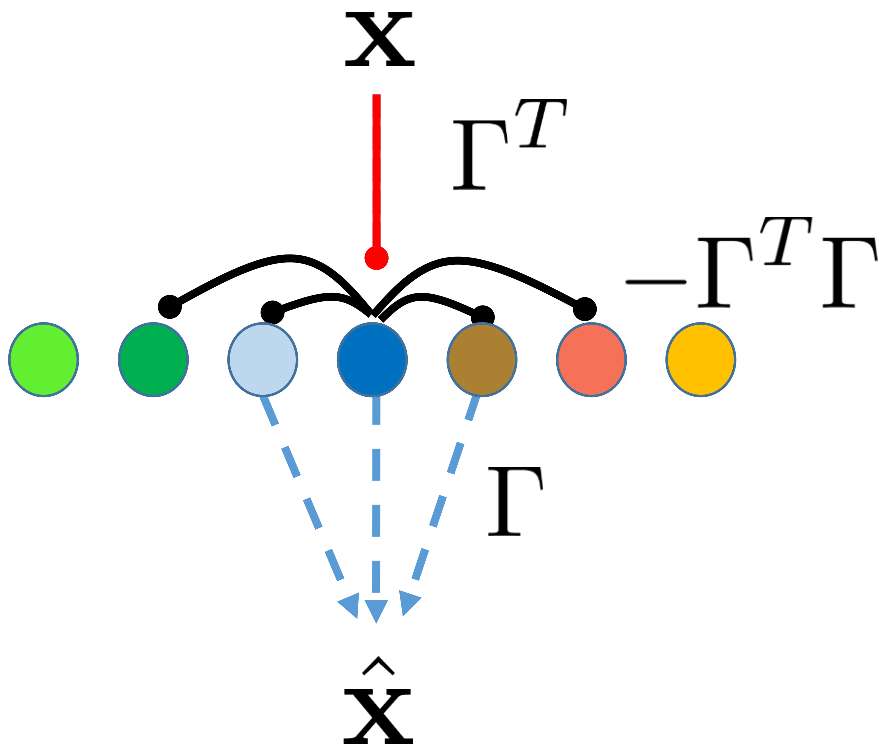



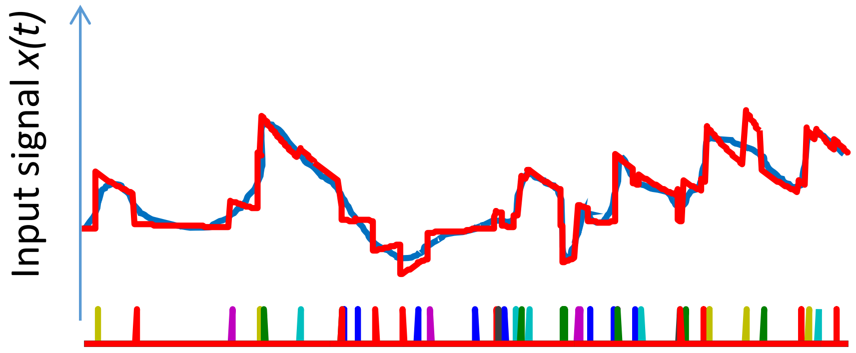
# Spike-based population coding

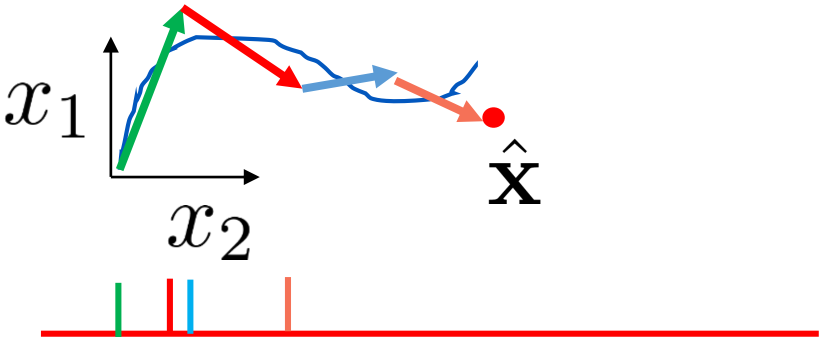


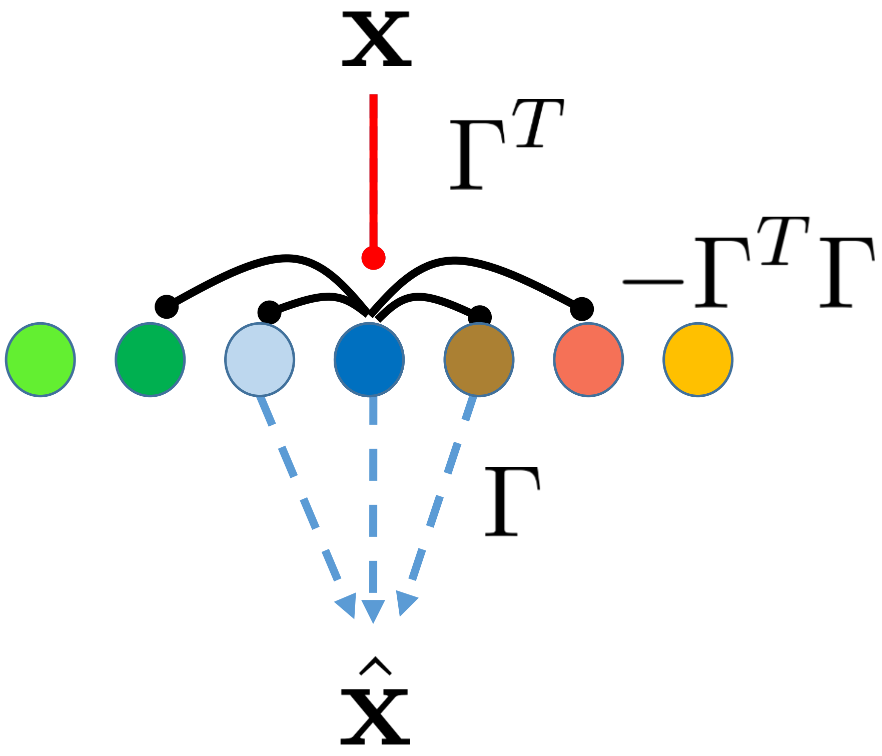



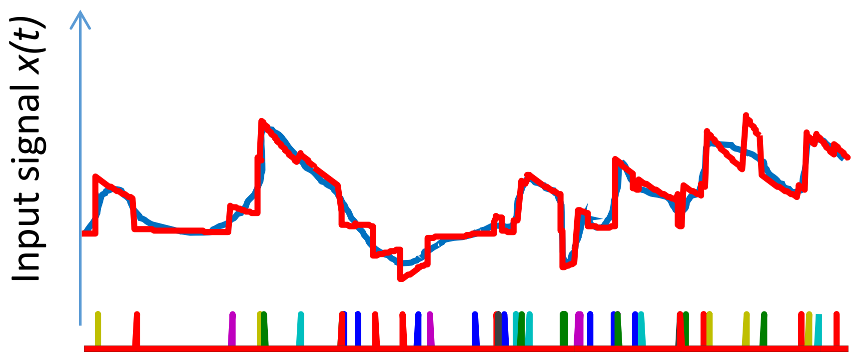
# Spike-based population coding

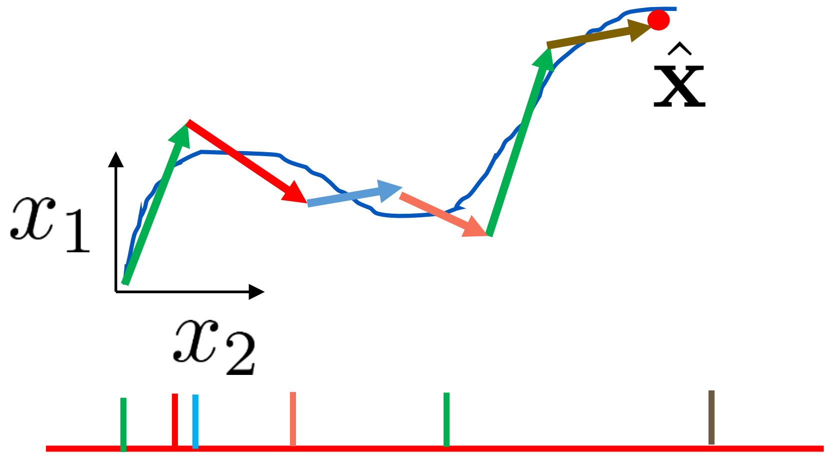


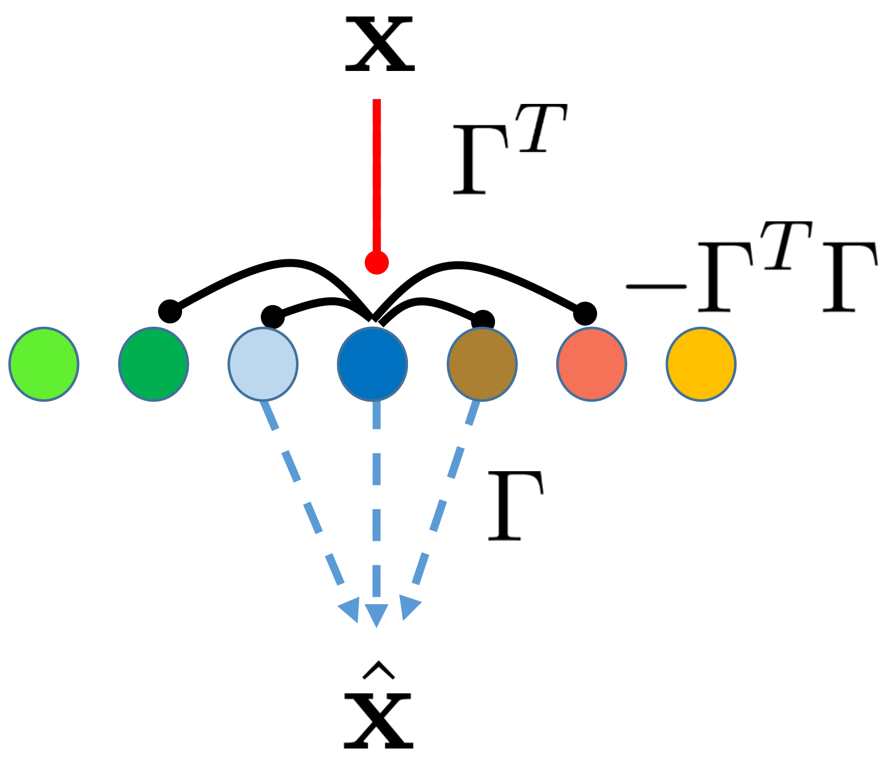



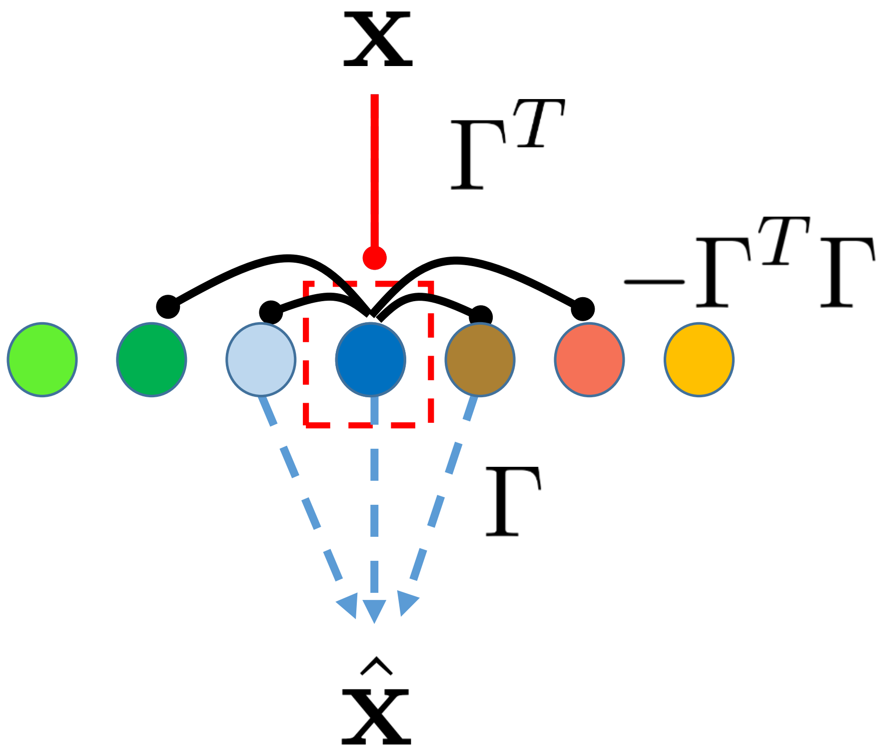
# Spike-based population coding



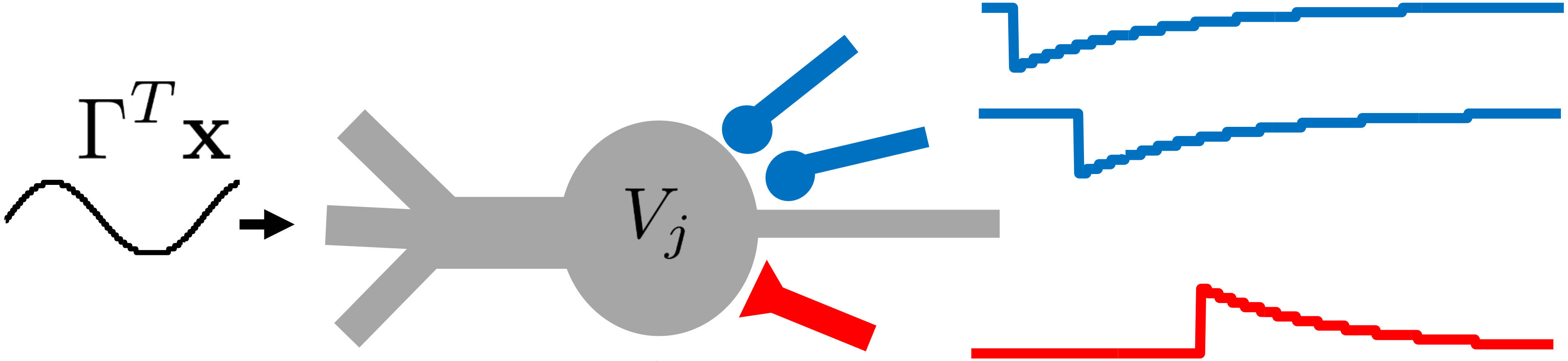
# Learning the connections



Ralph Bourdoukan

# Learning

$$-\Gamma^T \Gamma \mathbf{r} = -\Gamma^T \hat{\mathbf{x}}$$

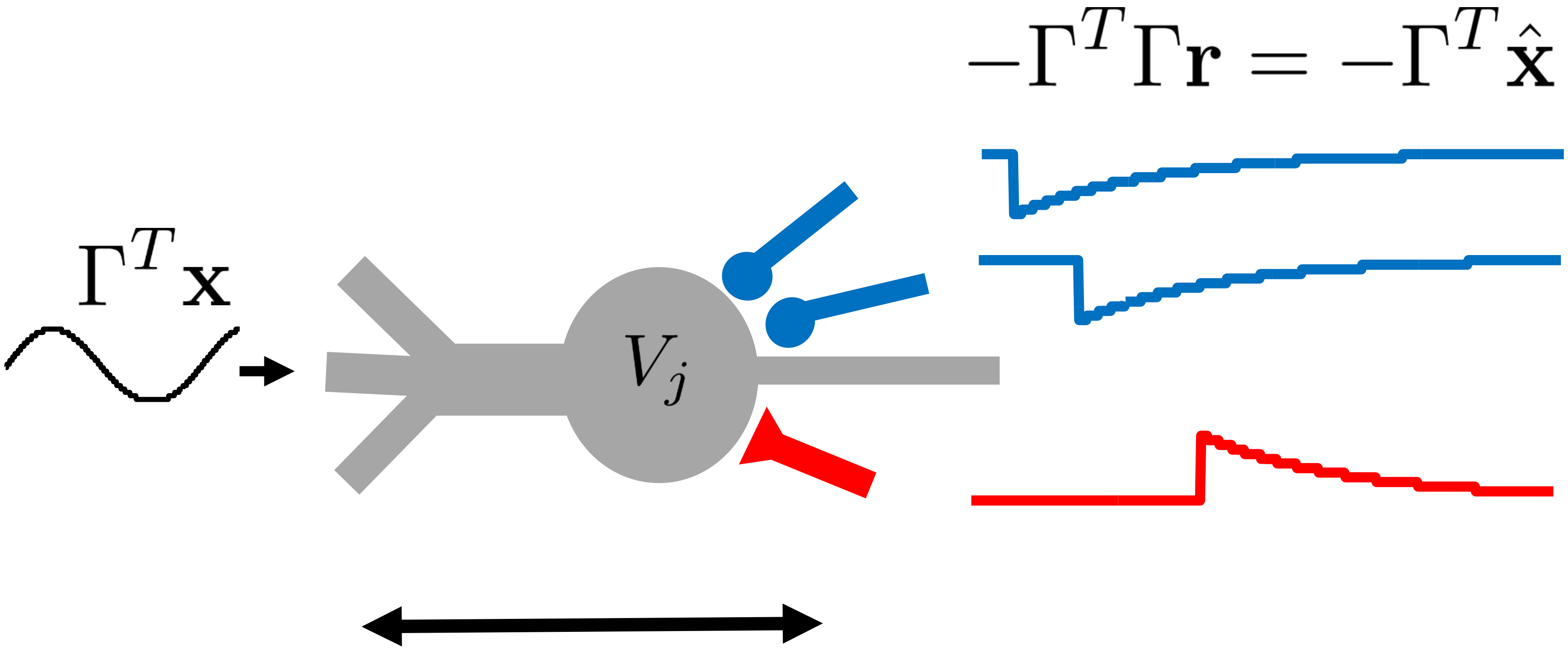



$$V = \Gamma^T (\mathbf{x} - \hat{\mathbf{x}})$$

Membrane potential = prediction error

Ralph Bourdoukan

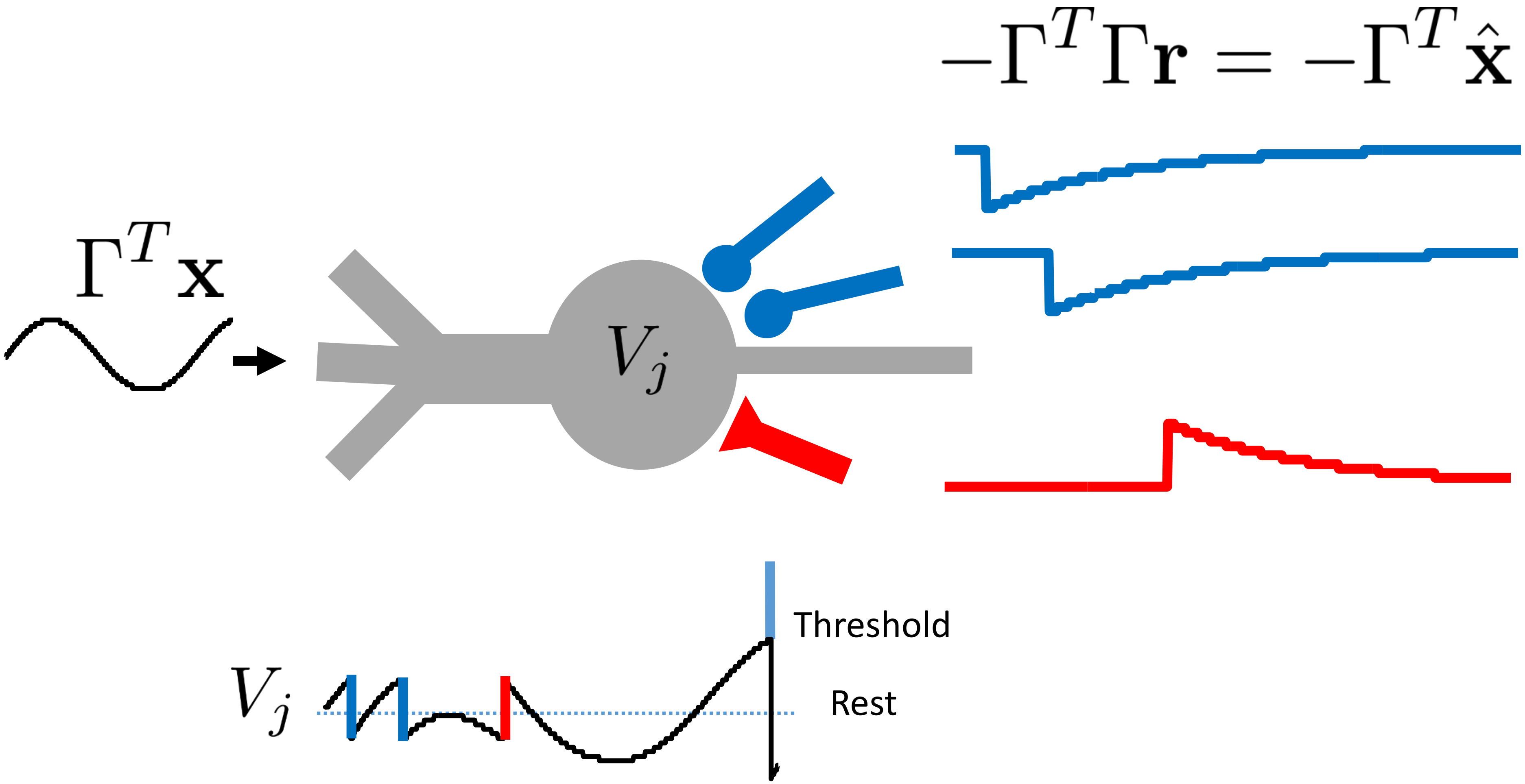
# Learning




E/I Balance = Minimize V

Ralph
Bourdoukan

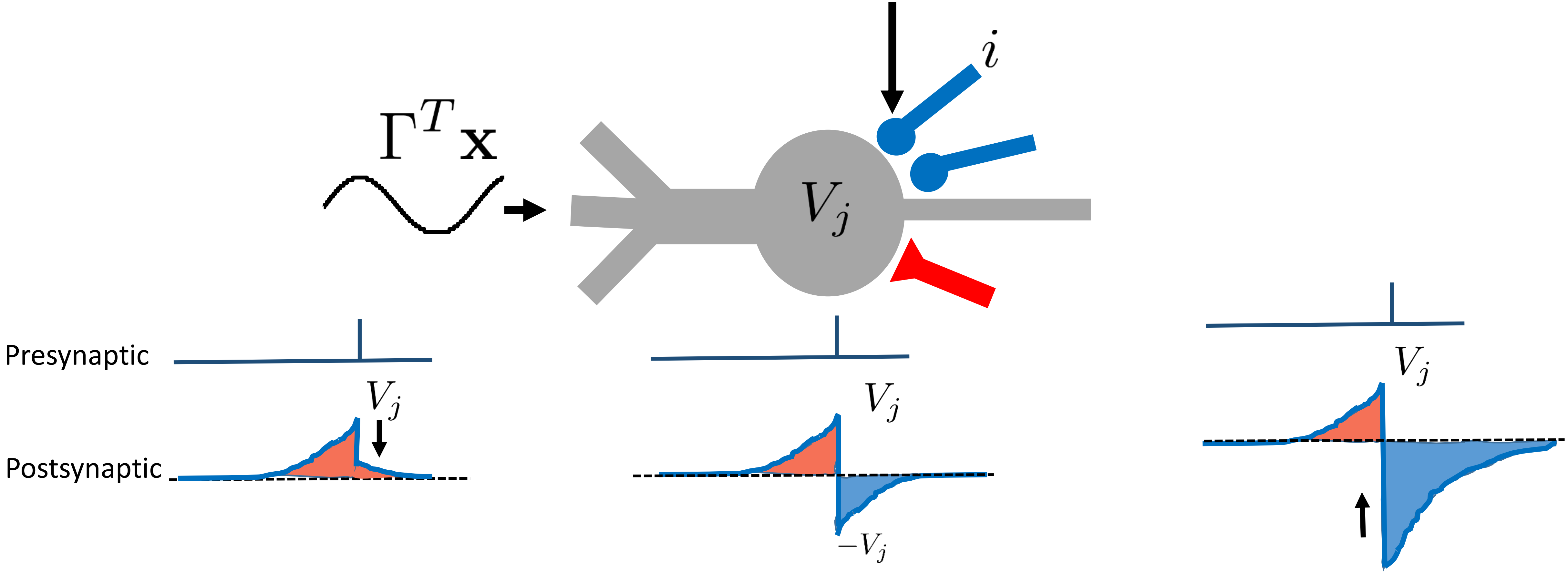
# Learning



Ralph
Bourdoukan

# Learning the recurrent connections

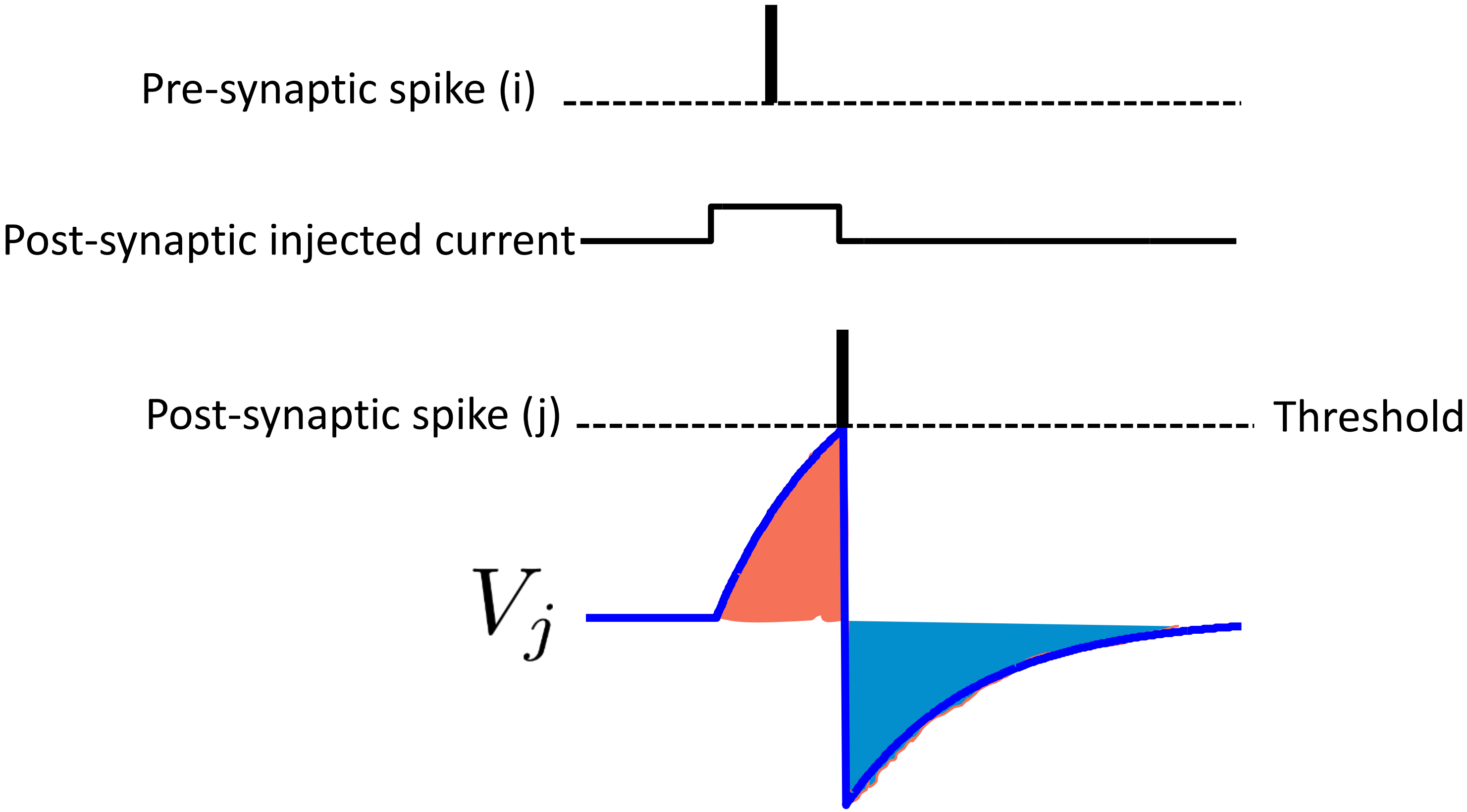
Whenever $i$ spikes: $\Delta W_{ij} \propto -2V_j + W_{ij}$



Ralph
Bourdoukan

# Spike time dependent plasticity

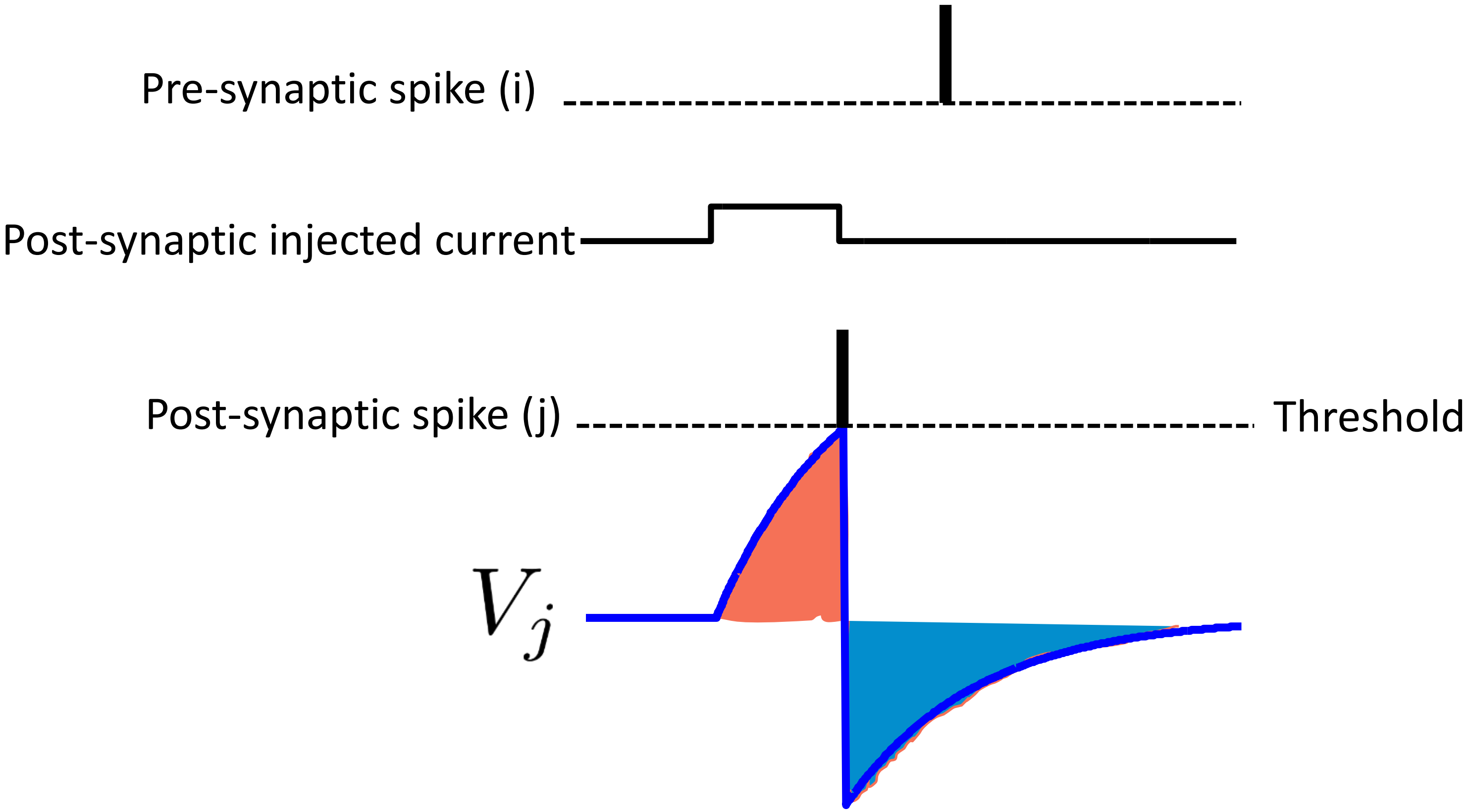
$$\Delta W_{ij} \propto -2V_j + W_{ij}$$



Ralph
Bourdoukan

# Spike time dependent plasticity

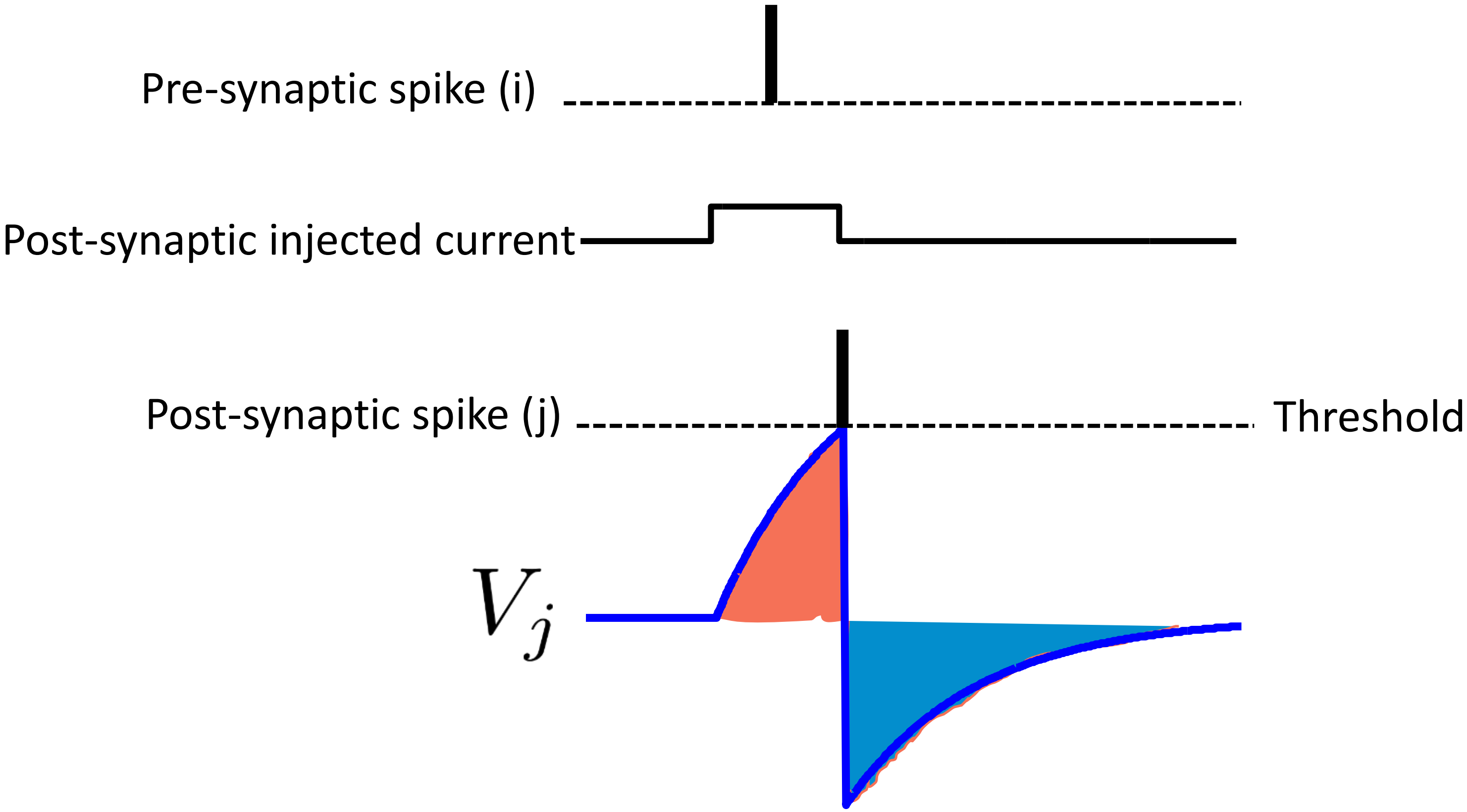
$$\Delta W_{ij} \propto -2V_j + W_{ij}$$

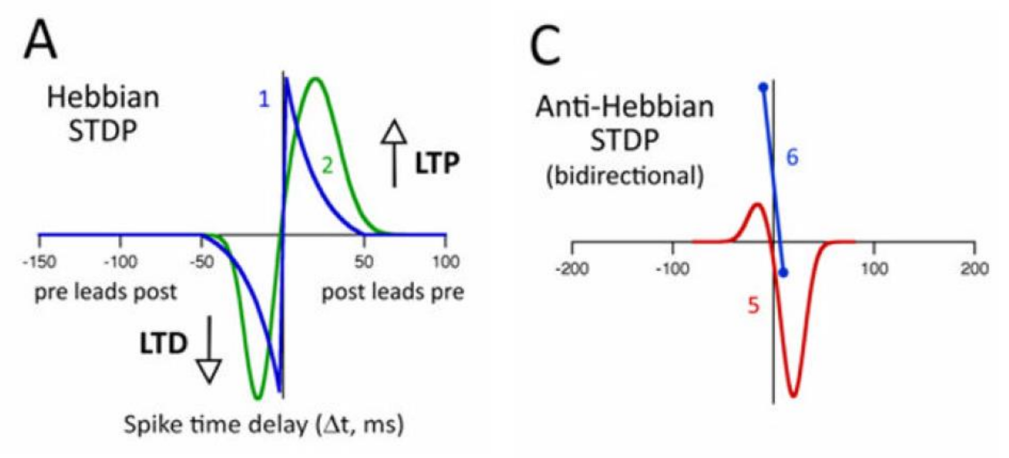


Ralph
Bourdoukan

# Spike time dependent plasticity

$$\Delta W_{ij} \propto -2V_j + W_{ij}$$



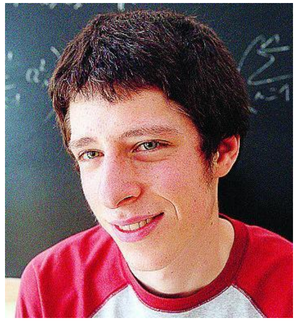



# Learning the optimal weights

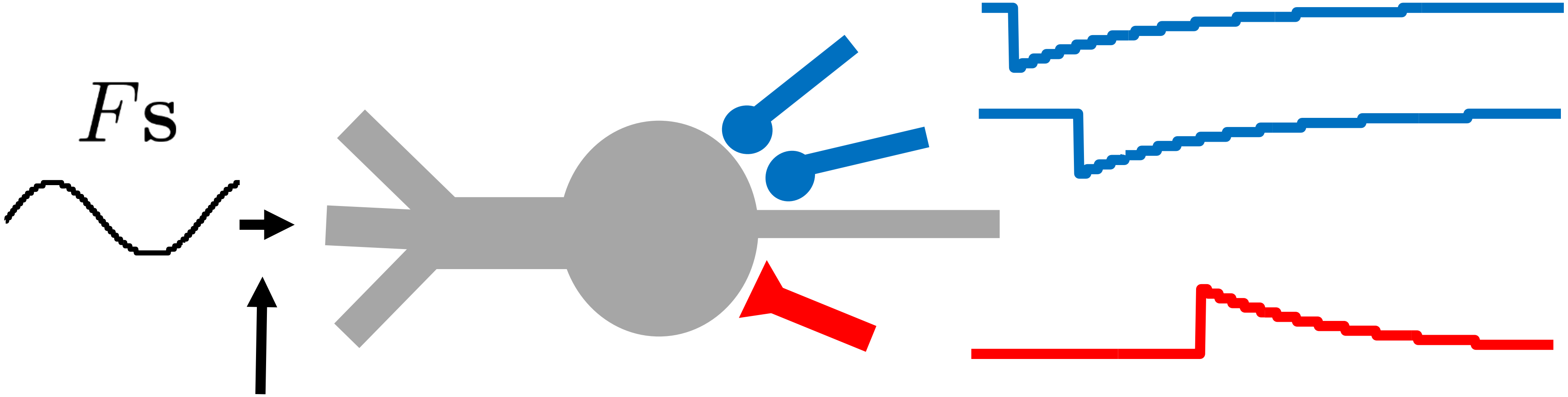
$$(\mathbf{r}, \Gamma) = \underset{\mathbf{r},\Gamma}{\arg\min} \left( \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \mathrm{Cost}(\mathbf{r}) \right)$$

# Learning the feedforward connections

Wieland Brendel



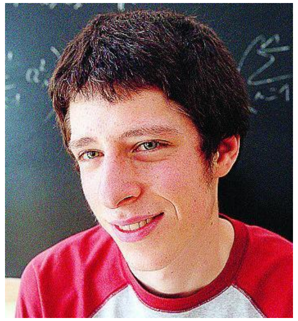
Pietro Vertechi



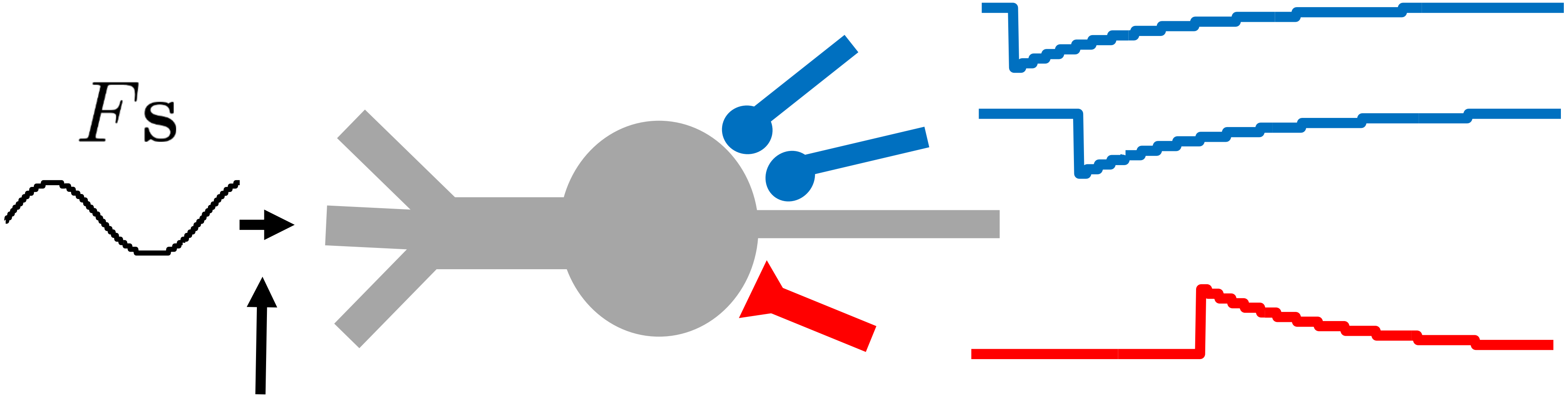
Wieland Brendel

# Learning the feedforward connections

## For uncorrelated inputs



Pietro Vertechi

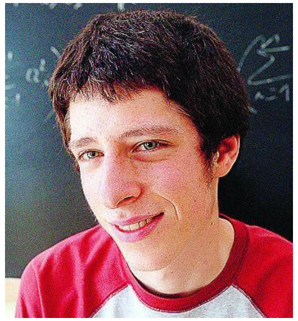



When j spikes:

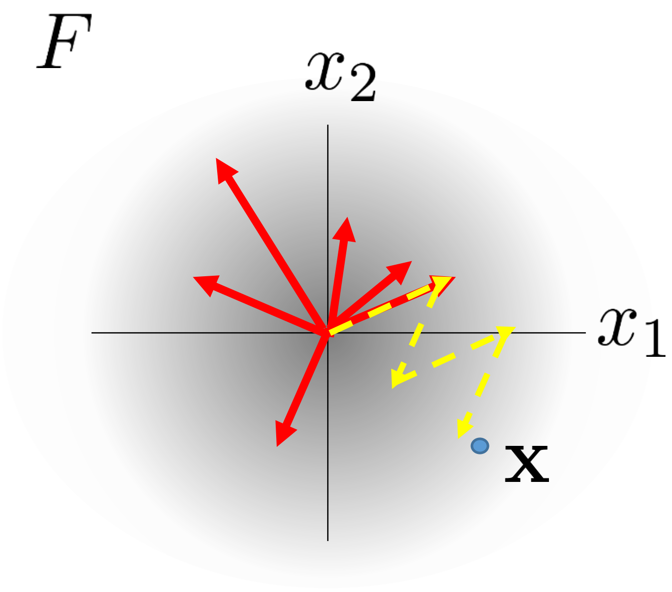
$$\Delta F_{kj} \propto \tilde{x}_k - F_{kj}$$

# Learning the feedforward connections

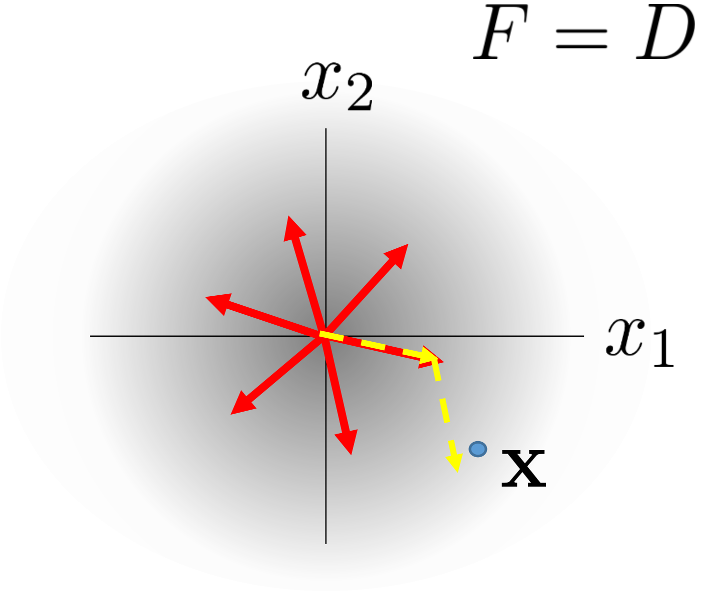
Wieland Brendel



Pietro Vertechi

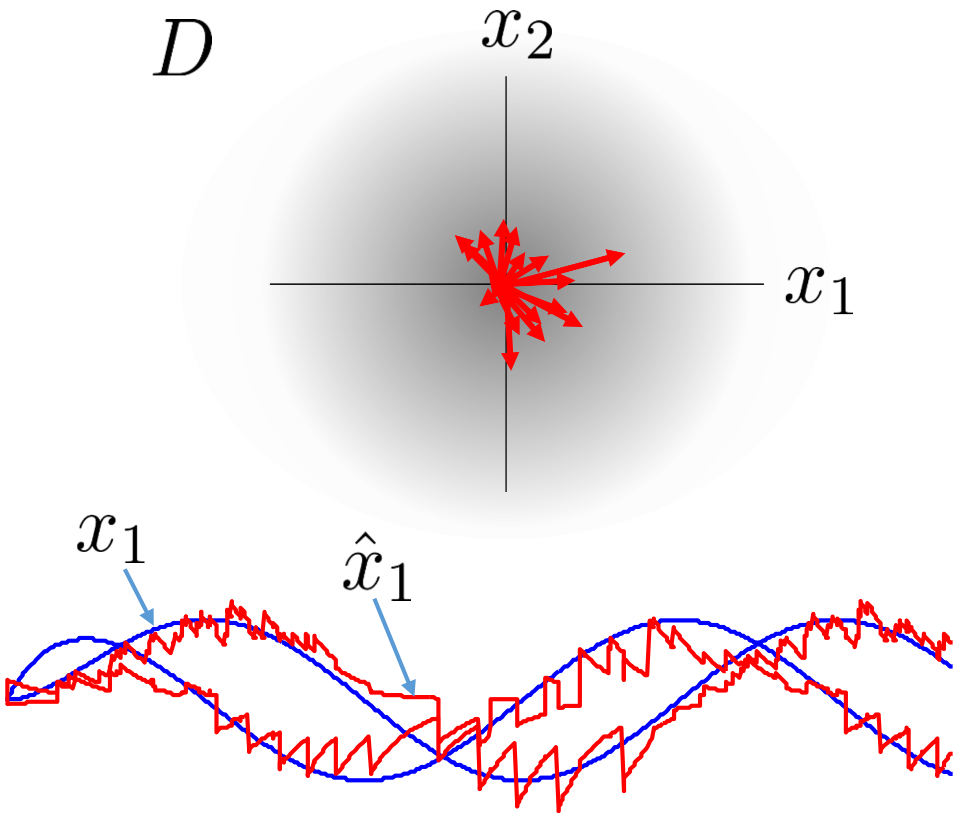



Sub-optimal

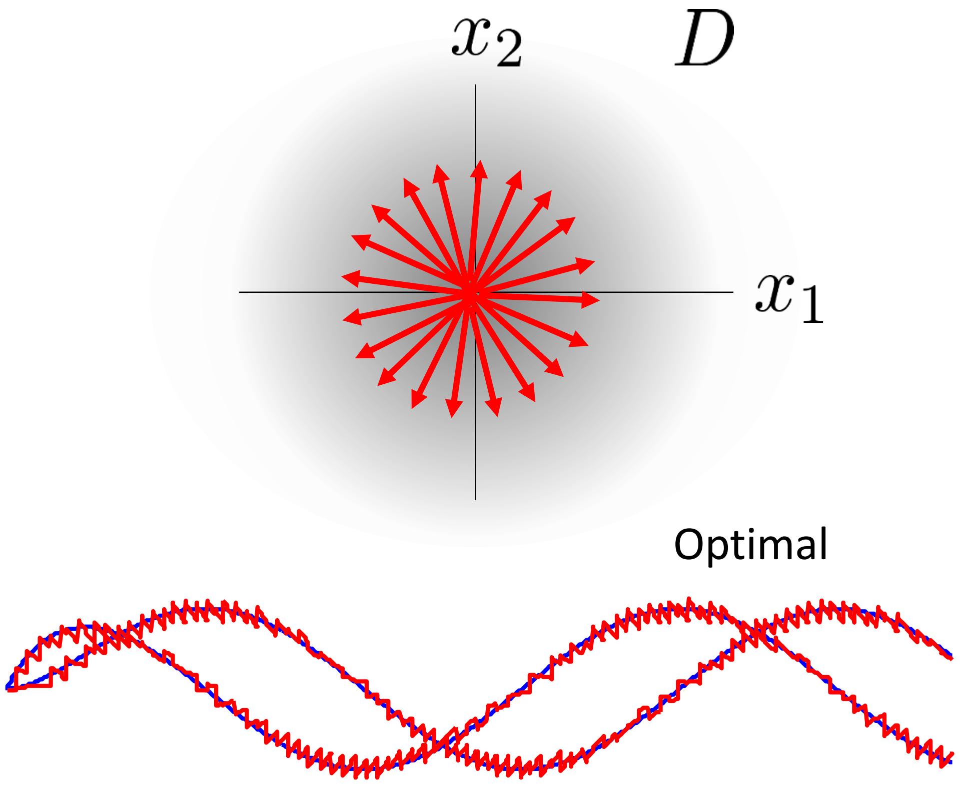


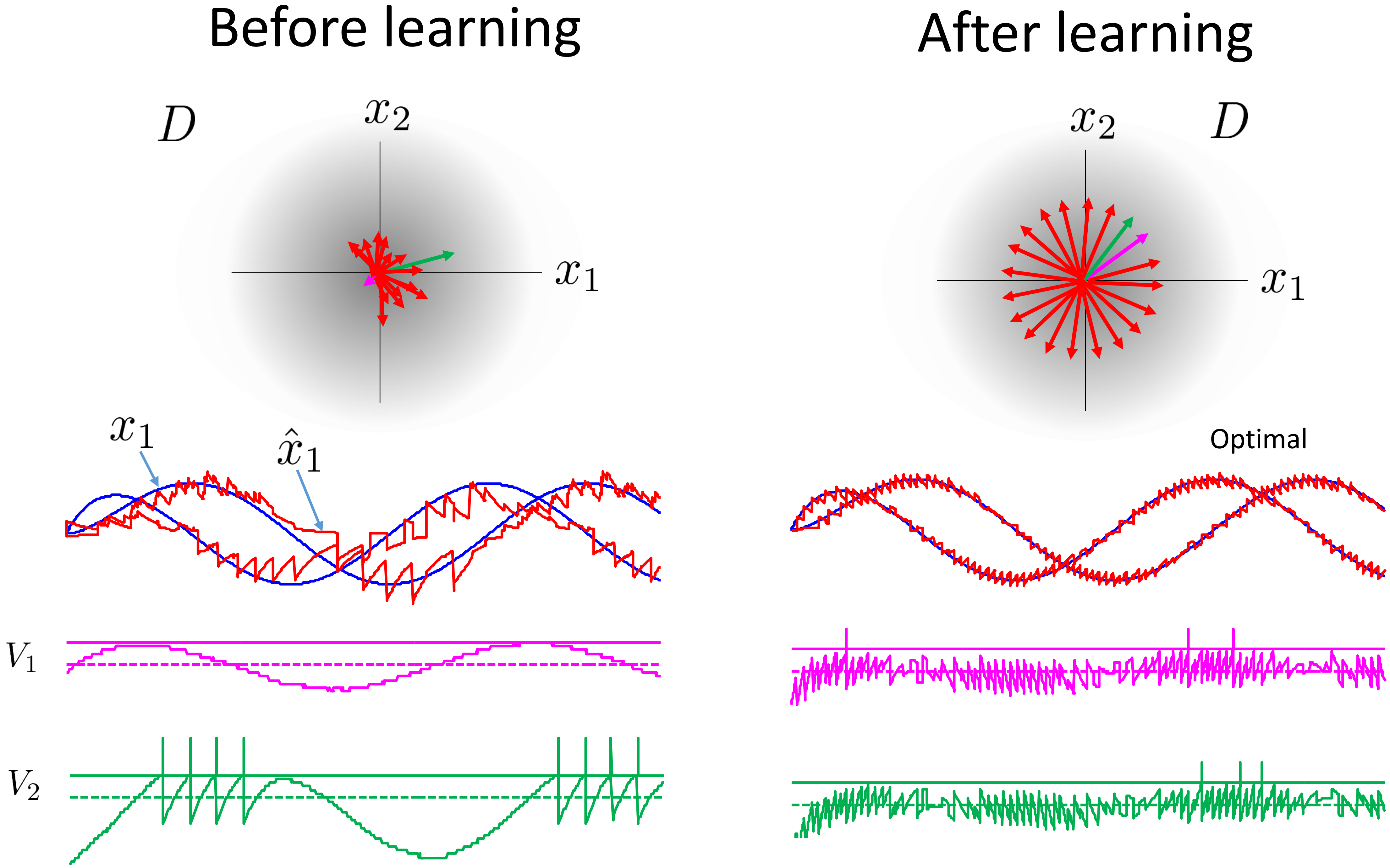

Optimal

## Before learning

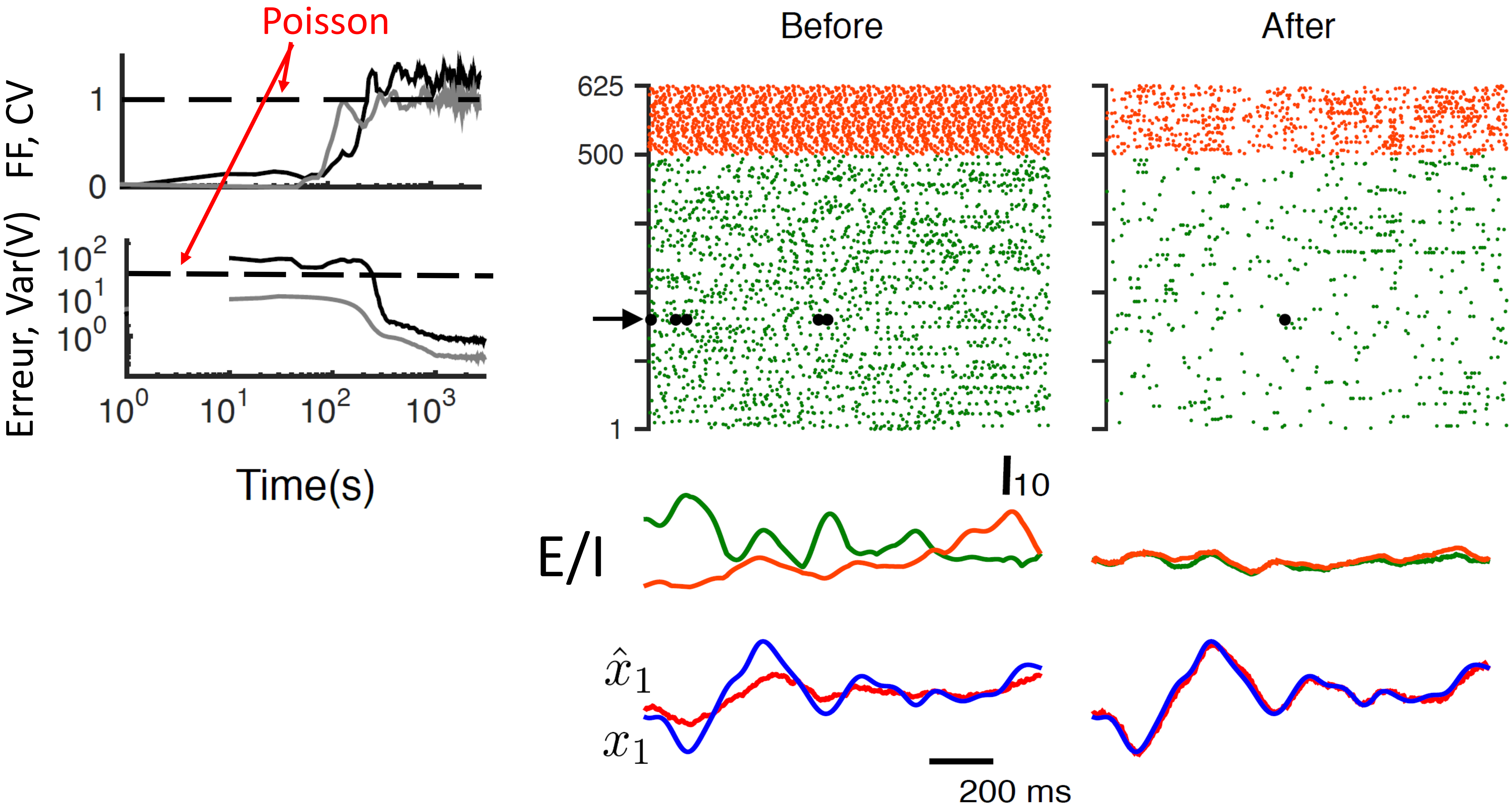



## After learning





Before learning
After learning
D
$x_2$
$x_1$
D
$x_2$
$x_1$
$x_1$
$\hat{x}_1$
Optimal
$V_1$
$V_2$

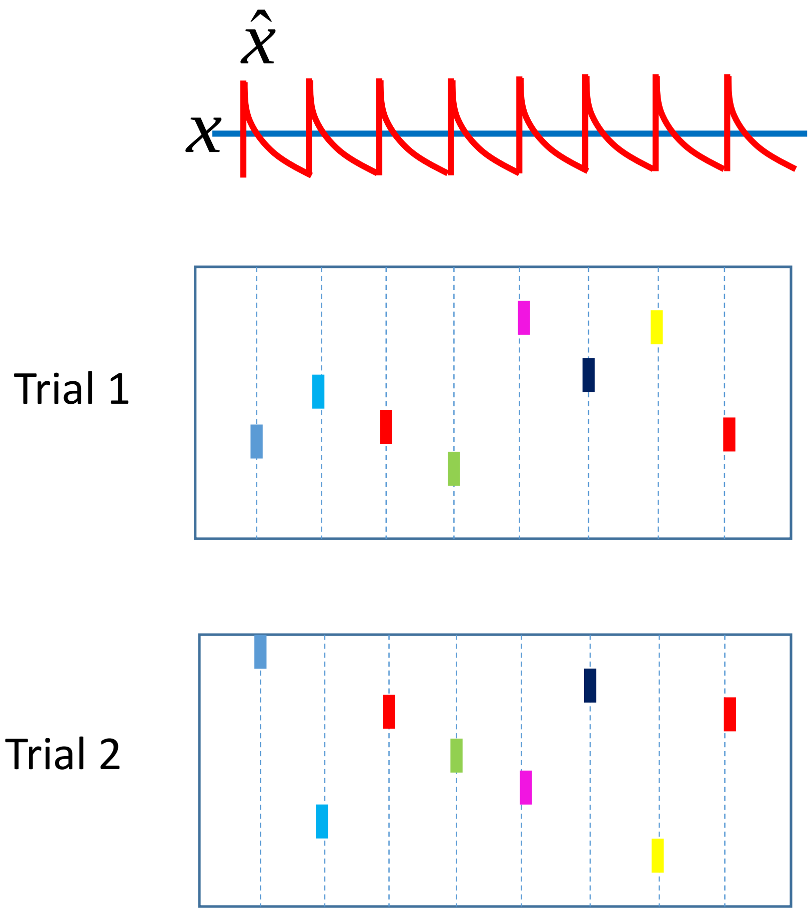


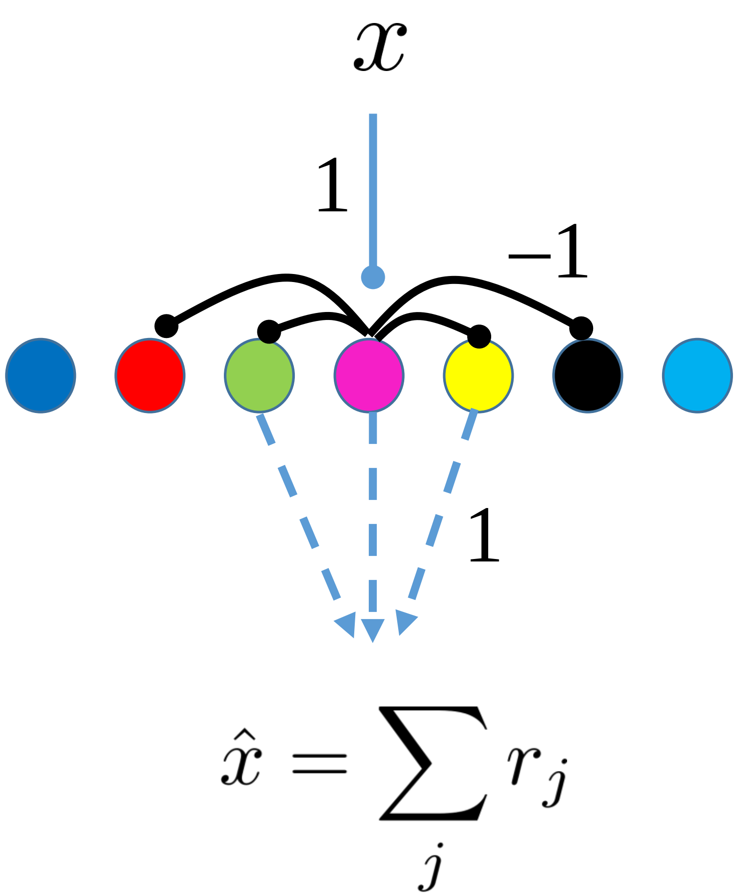
Poisson
FF, CV
1
0
Erreur, Var(V)
$10^2$
$10^1$
$10^0$
$10^0$ $10^1$ $10^2$ $10^3$
Time(s)
Before
After
625
500
1
$I_{10}$
E/I
$\hat{x}_1$
$x_1$
200 ms

# Implications

- Learning E/I balance = learning maximally efficient neural coding.
- From input statistics, model-free predictions for **plasticity, firing statistics** and **tuning properties of** spiking network**.**

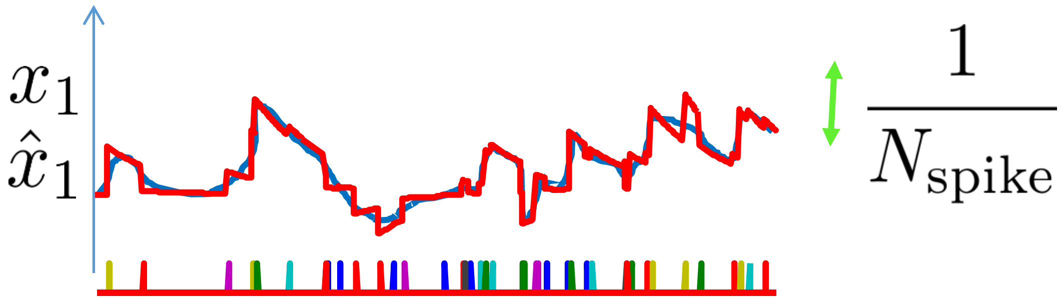
# Neural variability = Degeneracy, not noise



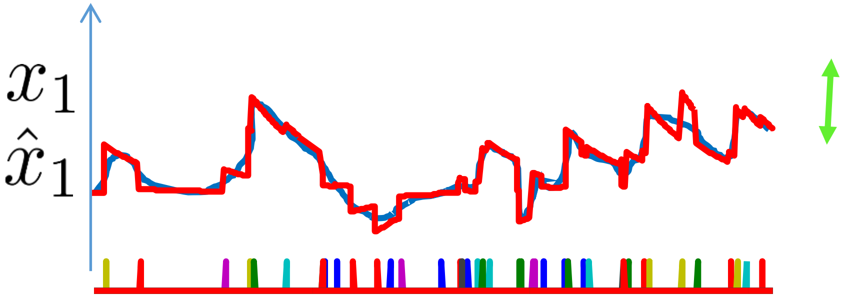
Martin Boerlin

# Neural variability = Degeneracy, not noise

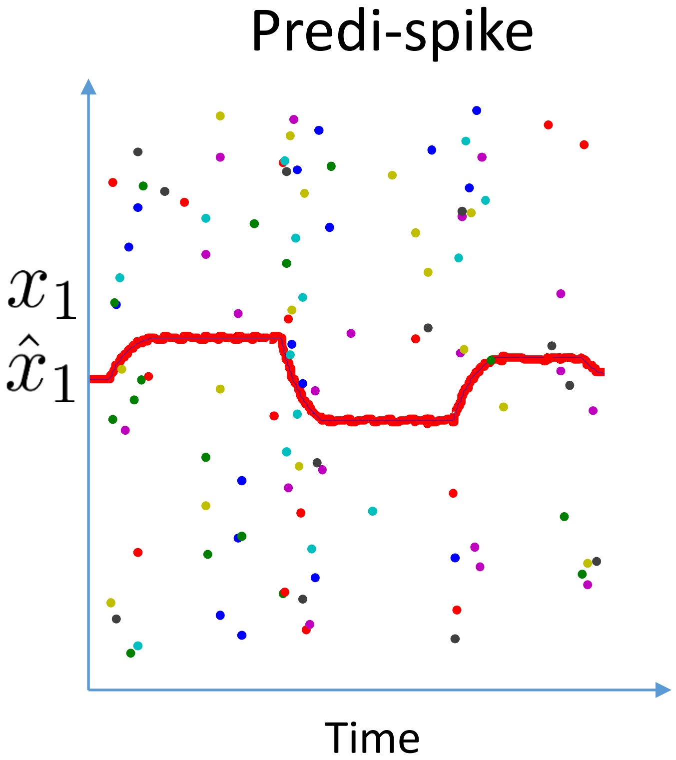


# Neural variability = Degeneracy, not noise

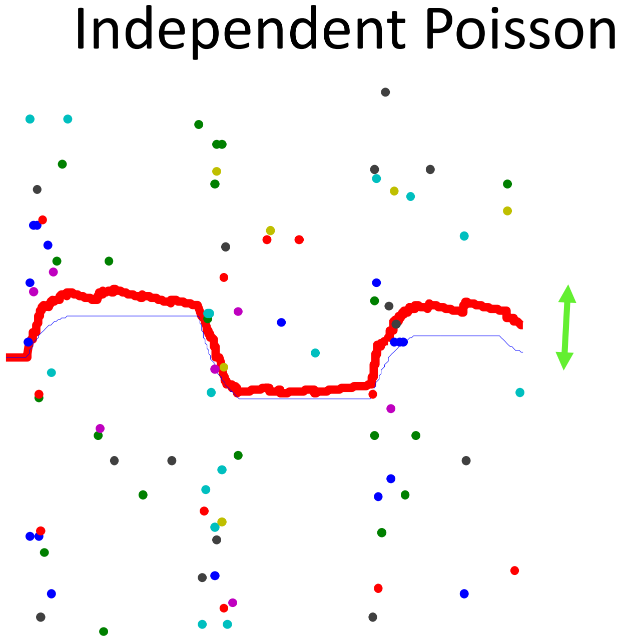
Martin
Boerlin




$$\frac{1}{N_{\text{spike}}}$$




Shuffle




$$\frac{1}{\sqrt{N_{\text{spike}}}}$$

# E/I balance = enforcing an efficient neural code.

Neural variability = degeneracy, not noise.
Degeneracy = robustness, not redundancy.
Single spikes are **RELEVANT**.

➡ Cortical networks might be much more precise than previously thought.

# Tuning curves = network solution

$$\mathbf{r} = \arg\min_{\mathbf{r}^*>\mathbf{0}} \left( \|\mathbf{x} - \Gamma \mathbf{r}^*\|^2 + \mathrm{Cost}(\mathbf{r}) \right)$$

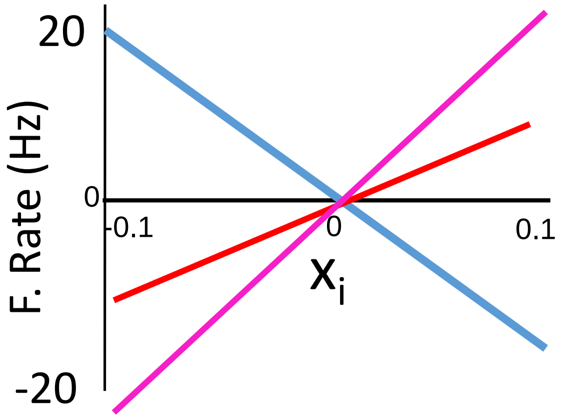
Activity of one neuron
depends on all other neurons

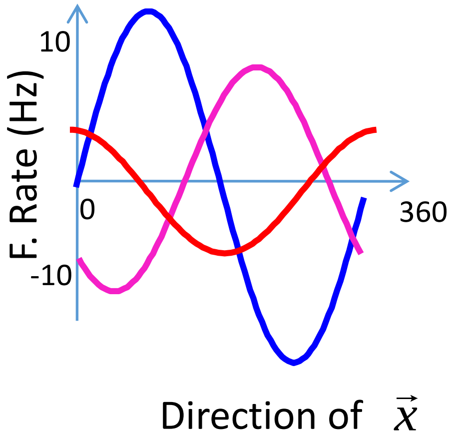
David Barrett

# Tuning curves = network solution

$$\mathbf{r} = \arg\min_{\mathbf{r}^* > \mathbf{0}} \left( \|\mathbf{x} - \Gamma \mathbf{r}^*\|^2 + \mathrm{Cost}(\mathbf{r}) \right)$$

## If firing rate could be negative…



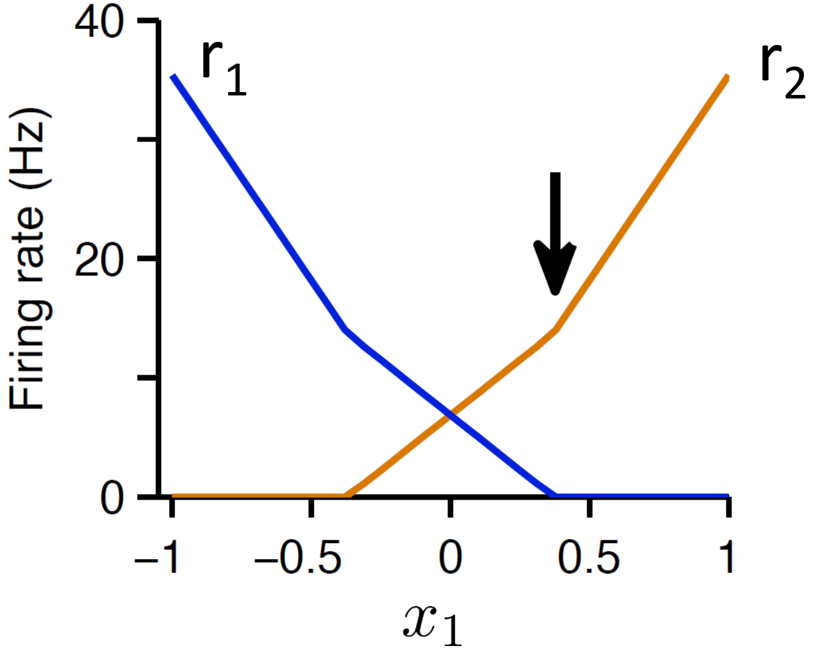



David Barrett

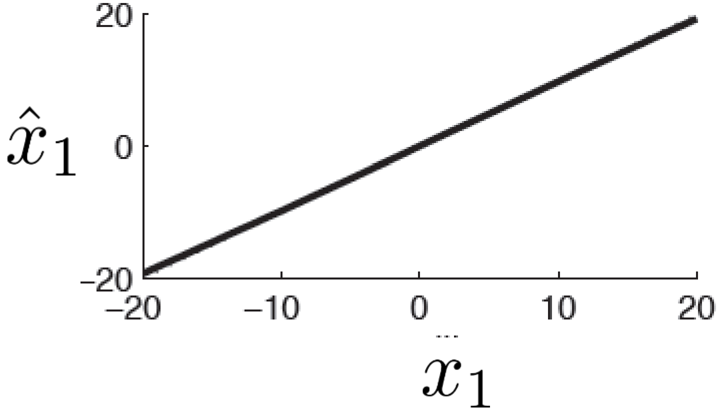
# Tuning curves = network solution

$$\mathbf{r} = \arg\min_{\mathbf{r}^* > \mathbf{0}} \left( \|\mathbf{x} - \Gamma \mathbf{r}^*\|^2 + \mathrm{Cost}(\mathbf{r}) \right)$$

But firing rates can't be negative …




$\hat{x}_1$ = $r_2 - r_1$

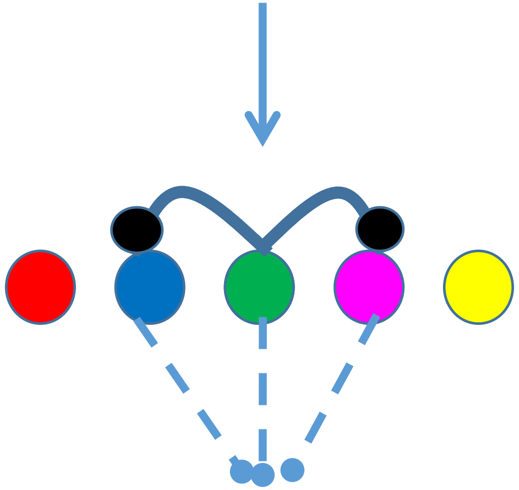


David Barrett

# Tuning curves = network solution

$$\mathbf{r} = \arg\min_{\mathbf{r}^* > \mathbf{0}} \left( \|\mathbf{x} - \Gamma \mathbf{r}^*\|^2 + \mathrm{Cost}(\mathbf{r}) \right)$$

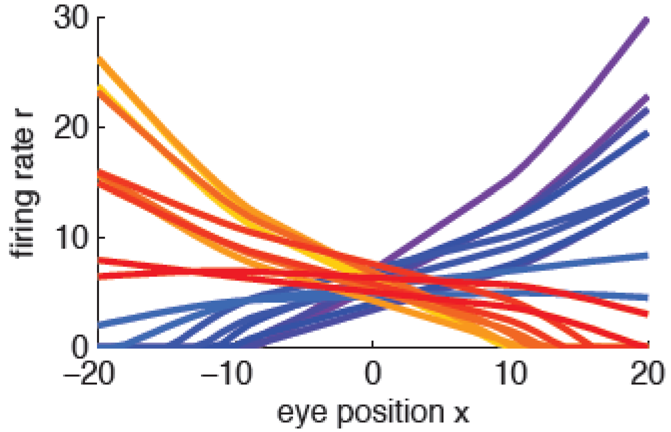
$x$



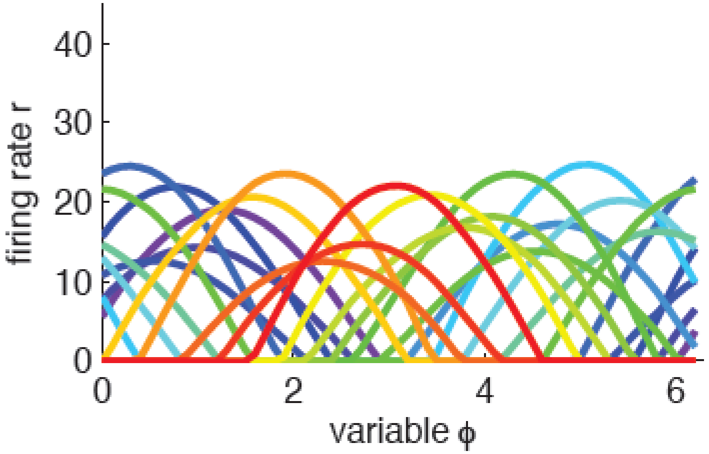
$$\hat{x} = \sum_j \Gamma_j r_j$$

## Non linear, heterogeneous tuning curves

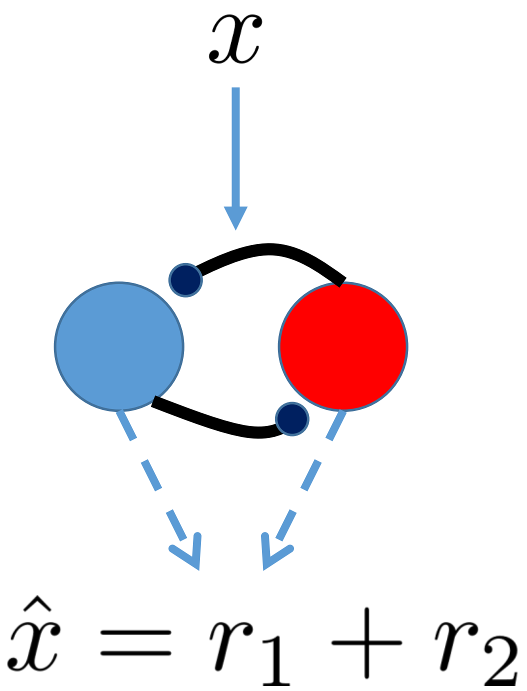
Amplitude tuning:

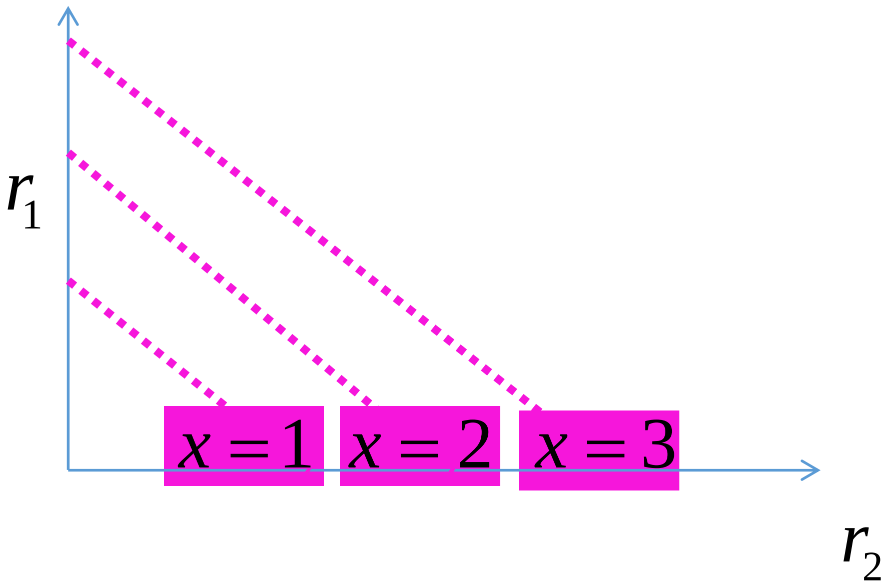



Direction tuning:

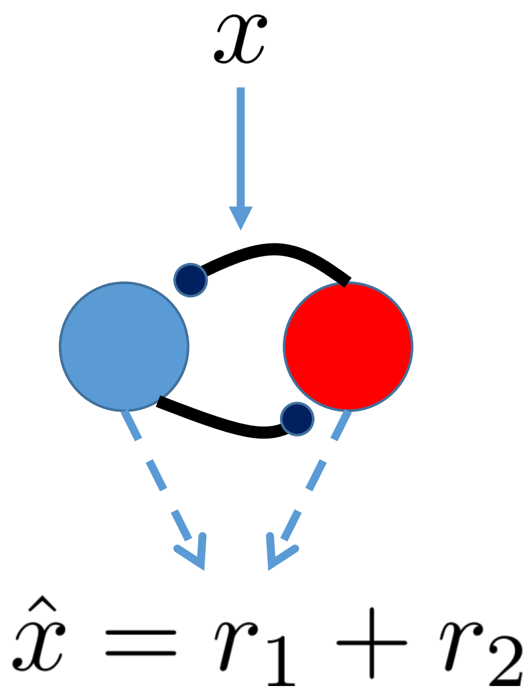


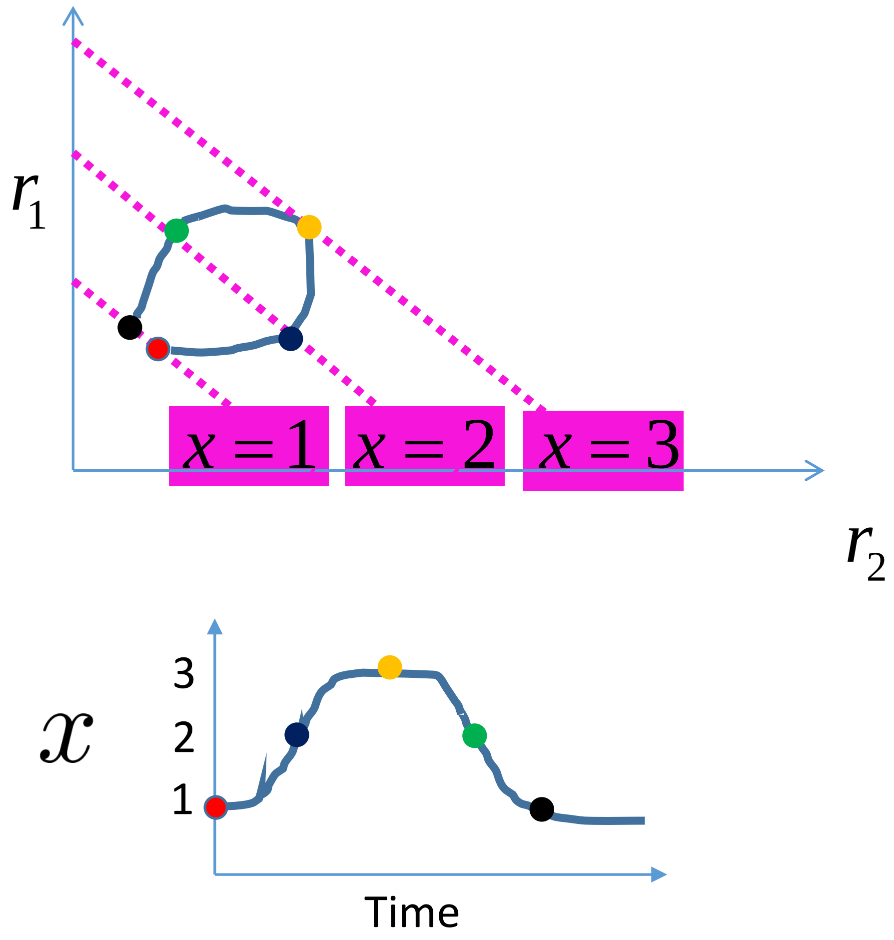
# Tuning curves = network solution



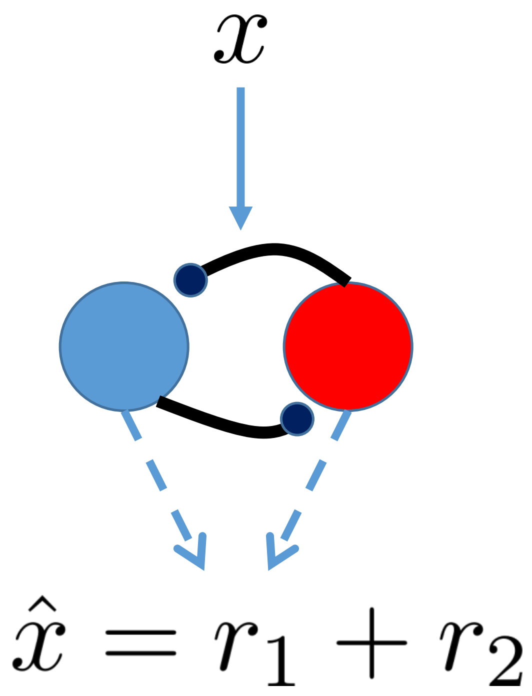



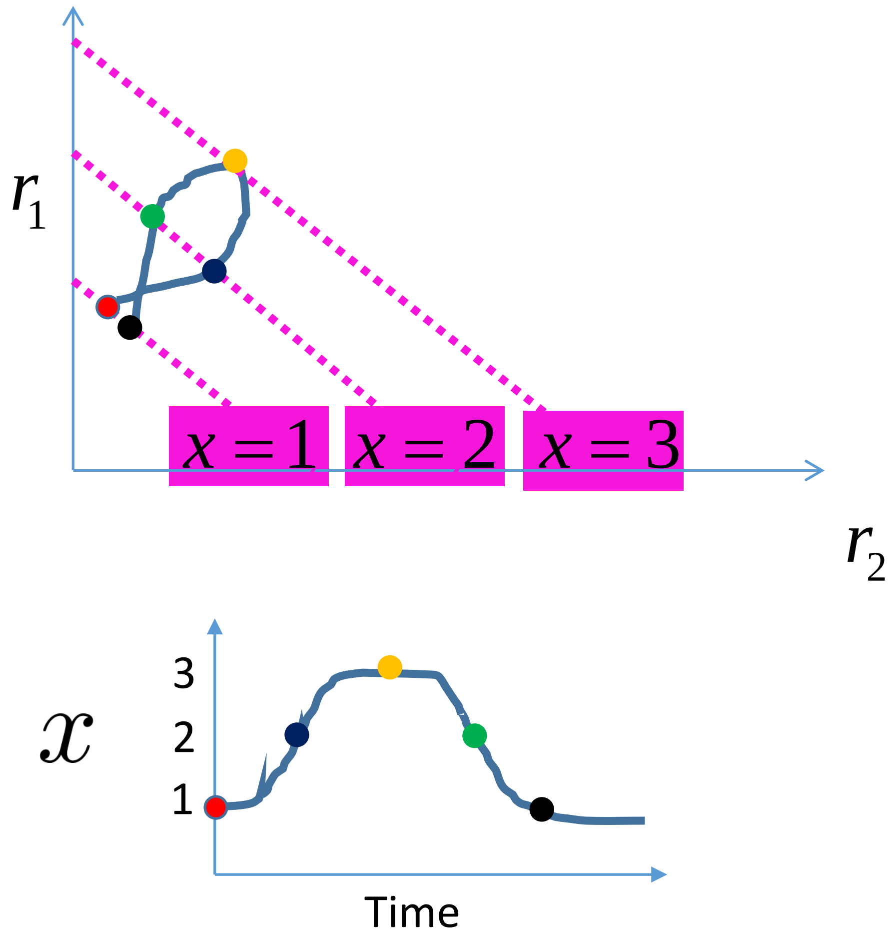
# Neural variability = Degeneracy, not noise



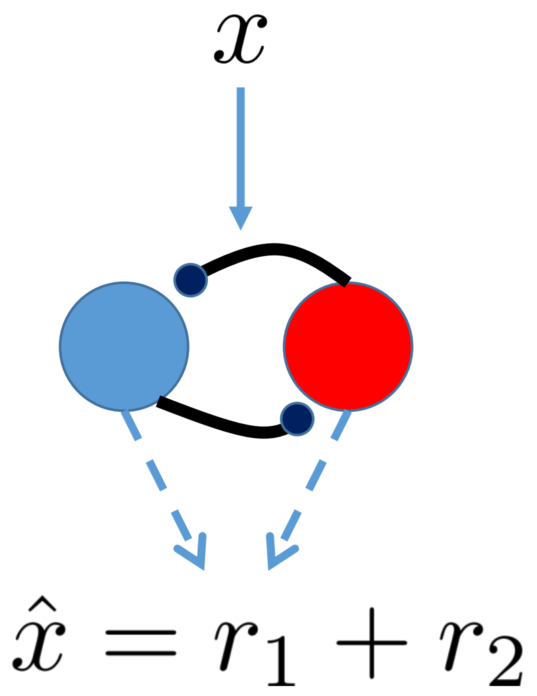



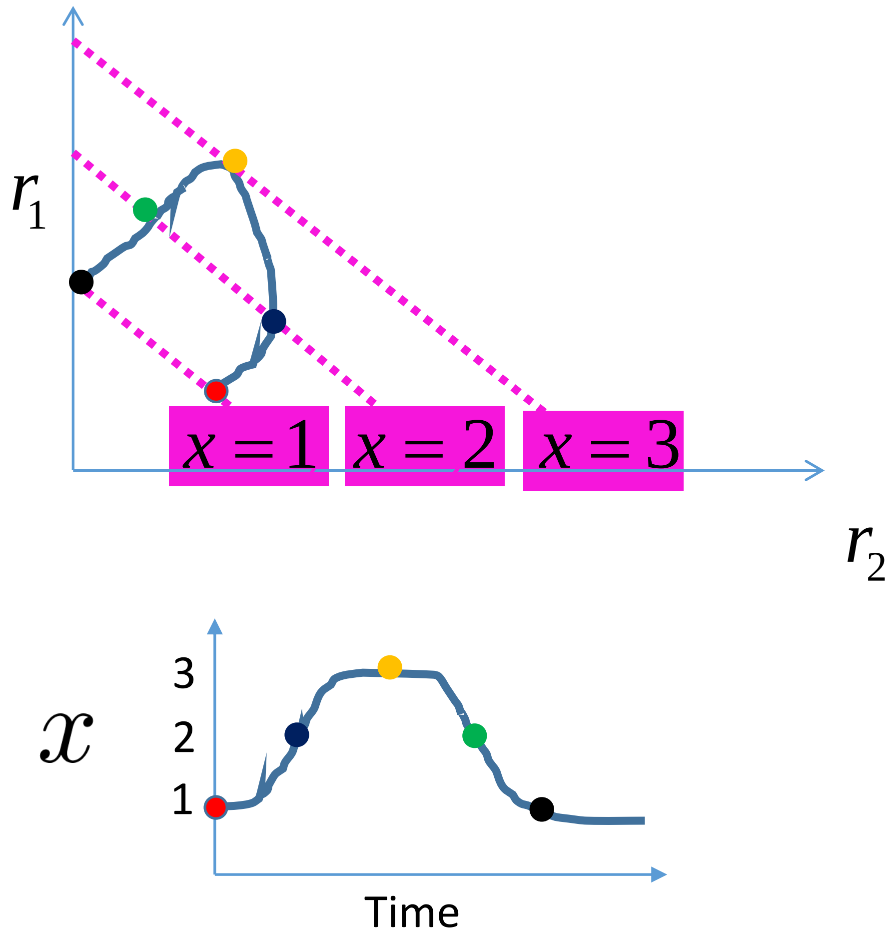
# Neural variability = Degeneracy, not noise



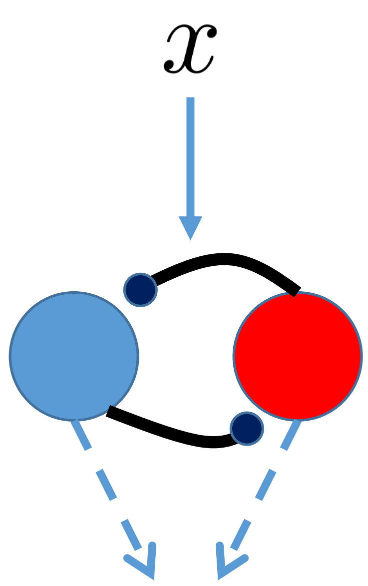


# Neural variability = Degeneracy, not noise



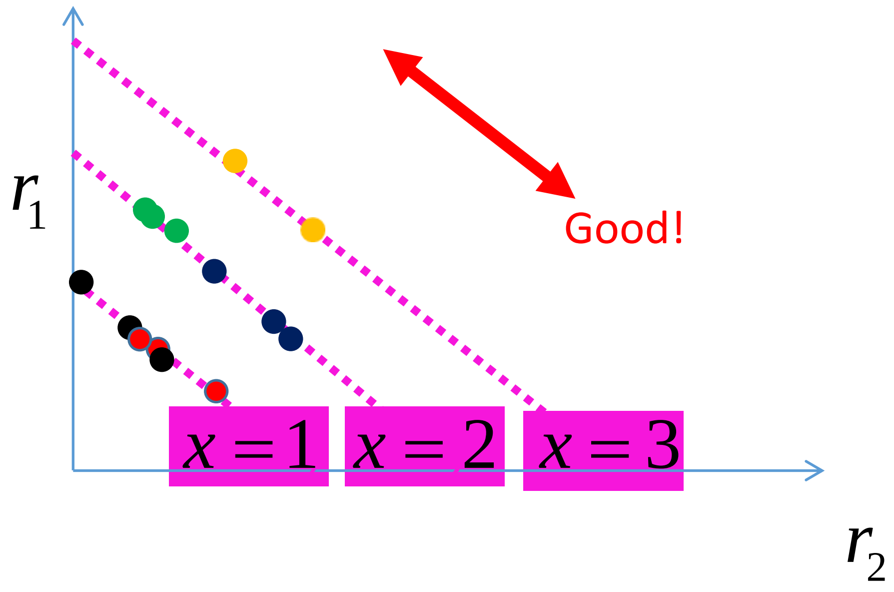



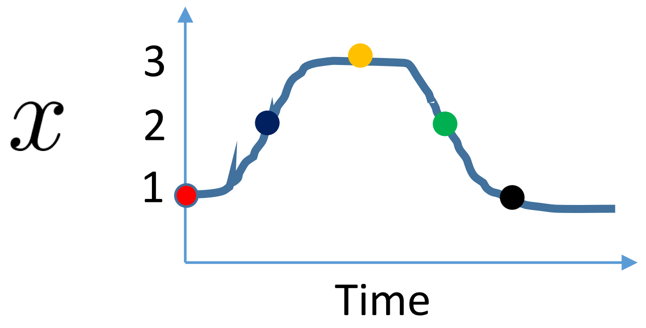
# Neural variability = Degeneracy, not noise



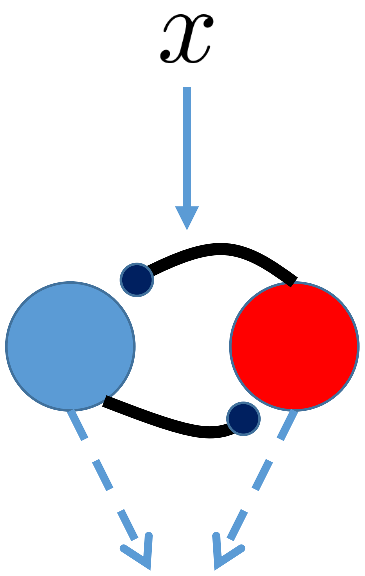

$$\hat{x} = r_1 + r_2$$

« Noise » correlations:

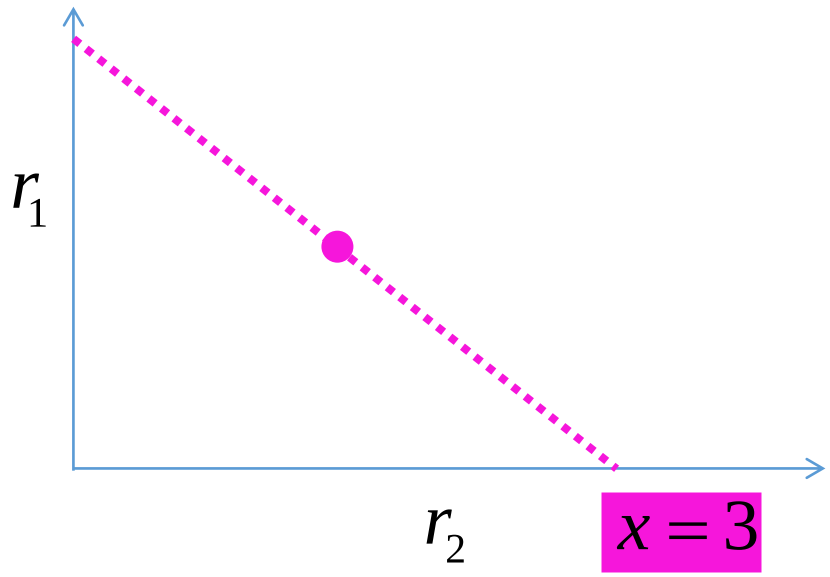


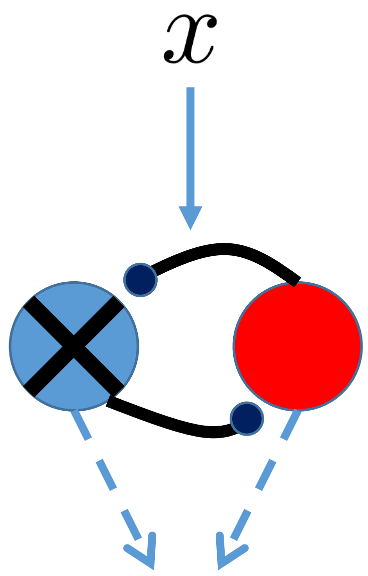
# Tuning curves = network solution



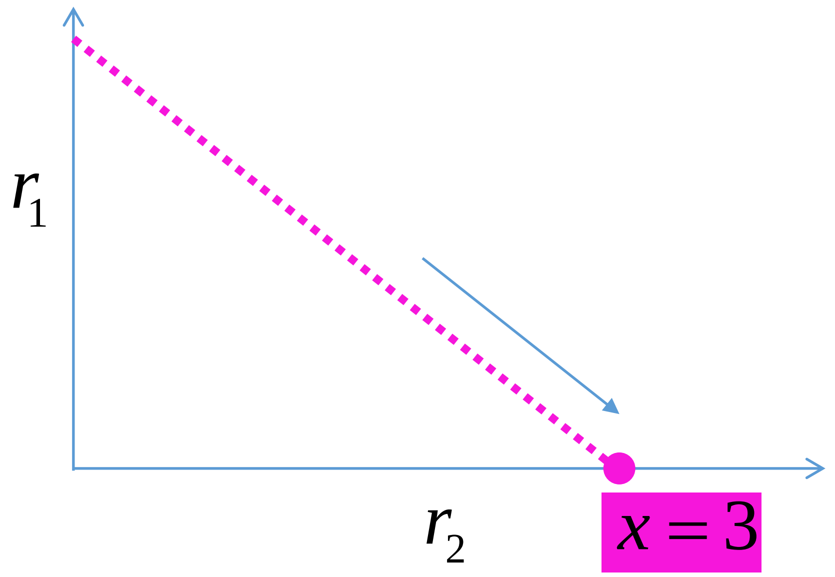

$$\hat{x} = r_1 + r_2$$

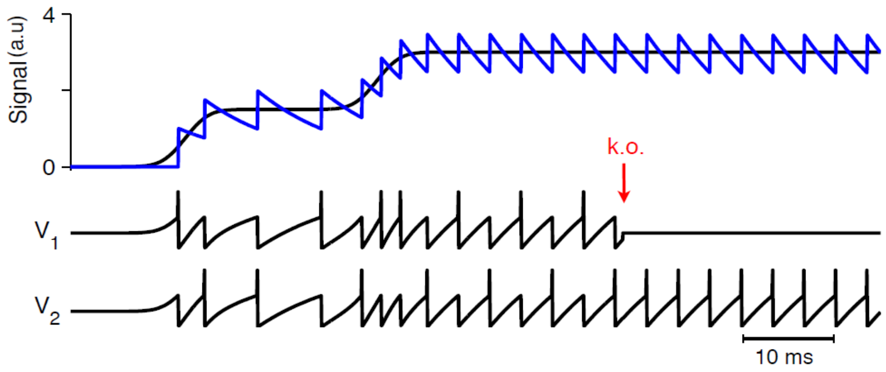


# Tuning curves = network solution

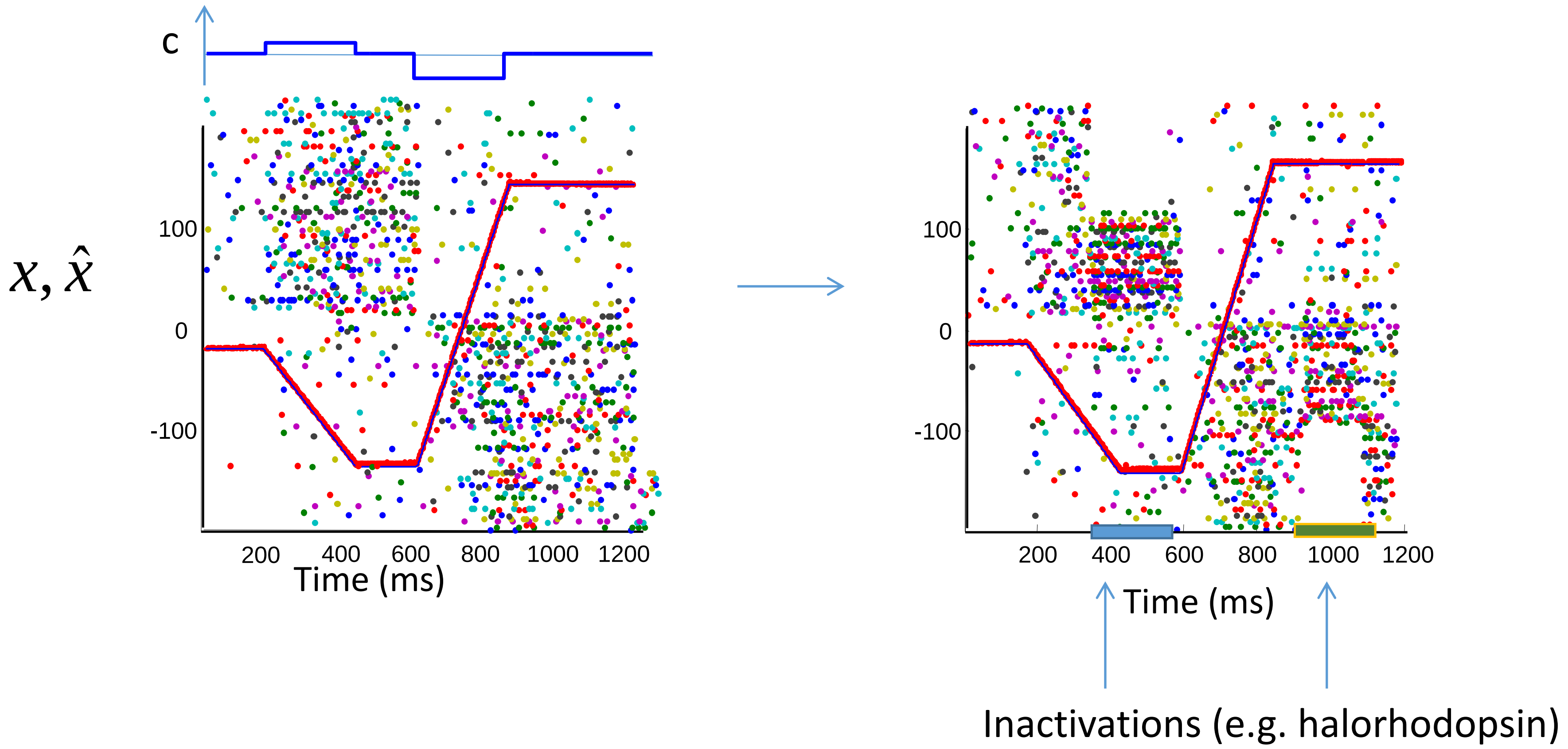



$$\hat{x} = r_1 + r_2$$



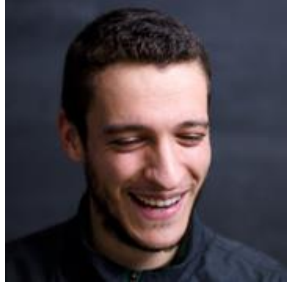



# The network is extremely robust

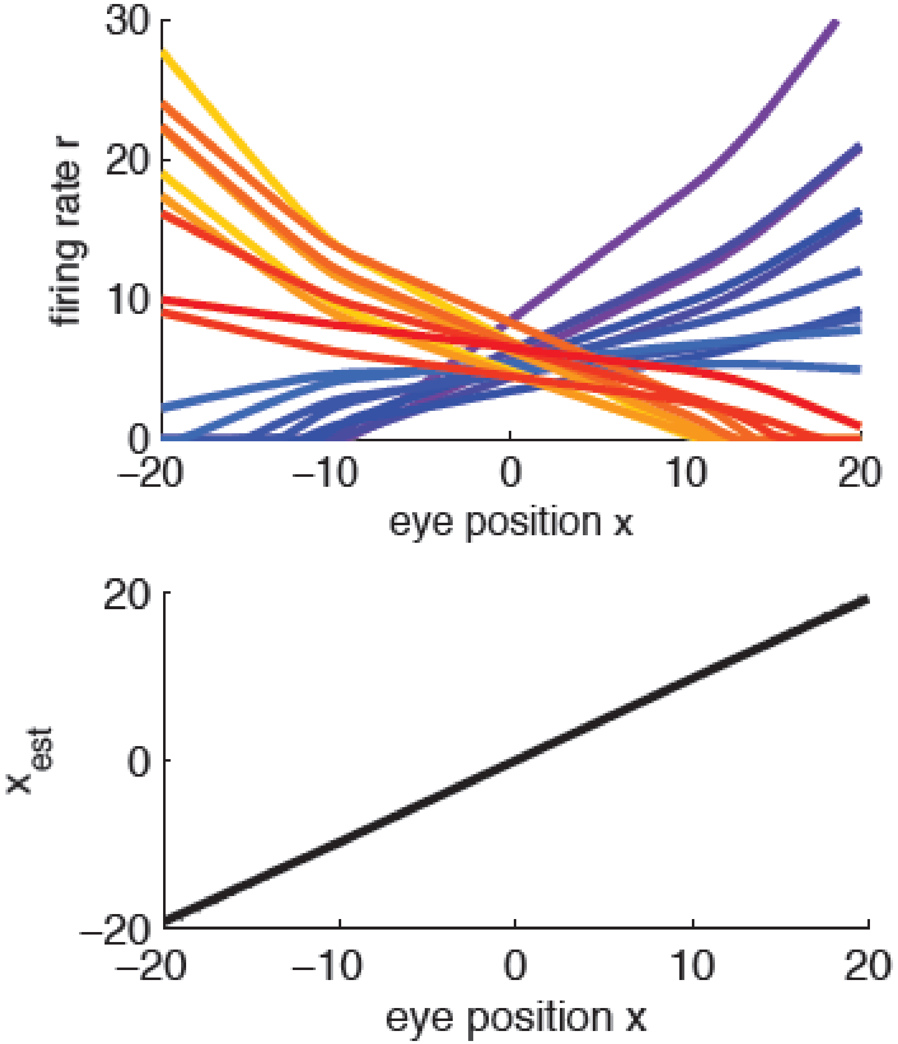



Robust to neural death, connection noise, background noise, synaptic failure…

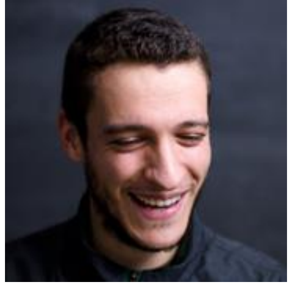
David Barrett



Nuno Calaim

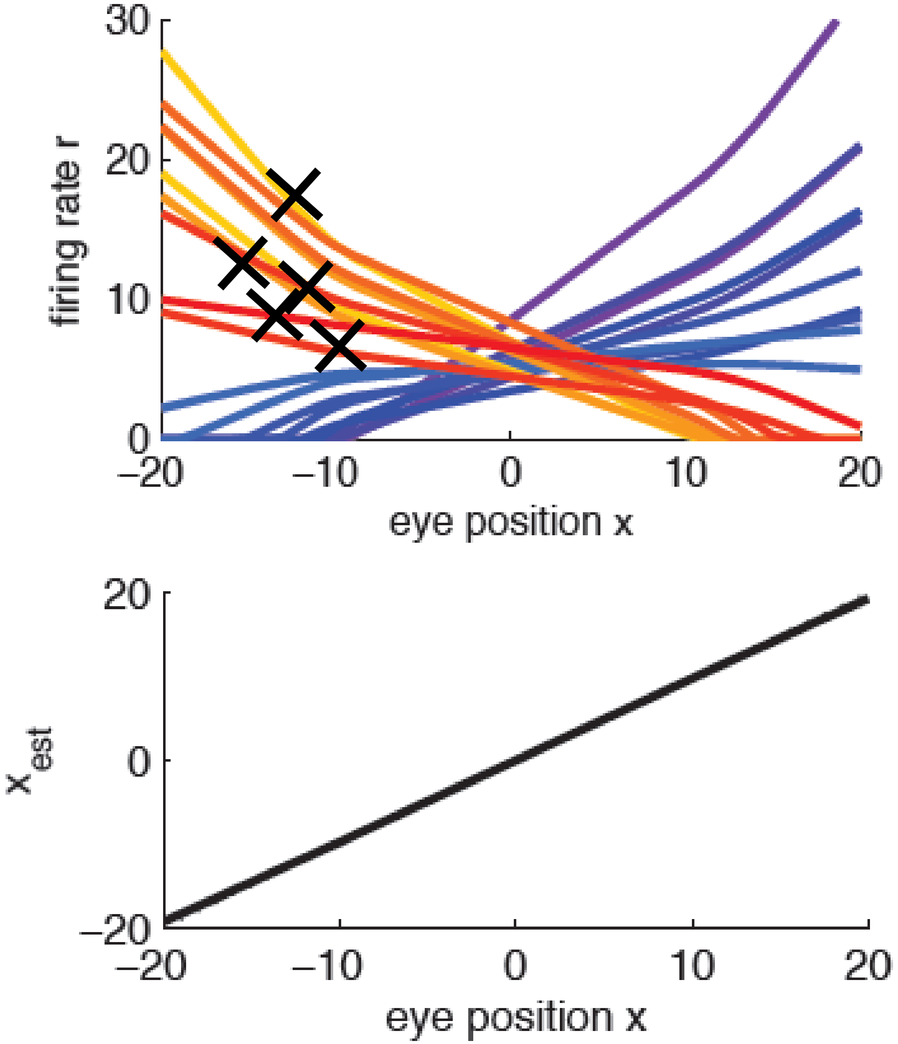


David Barrett

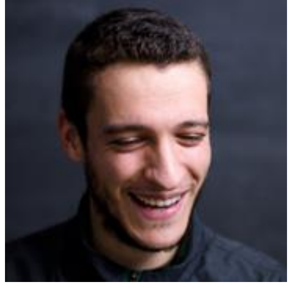


Nuno Calaim

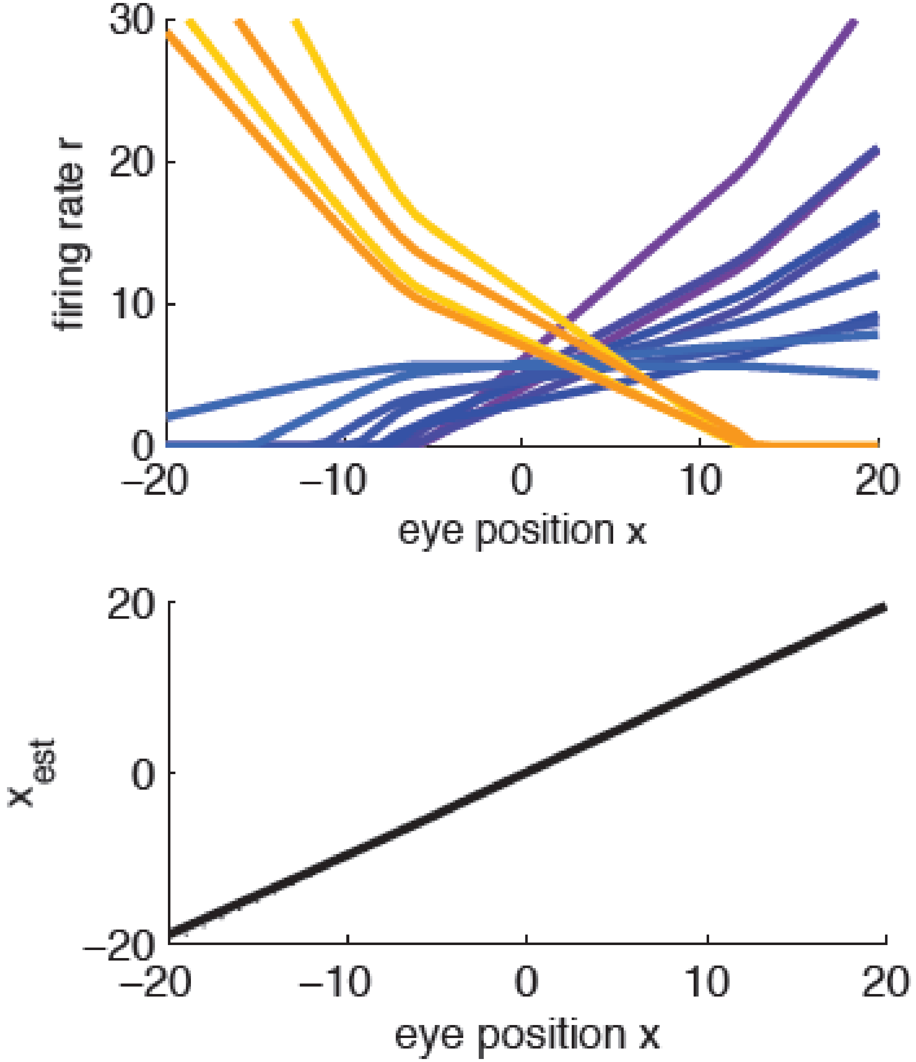
Ablate those neurons



David Barrett

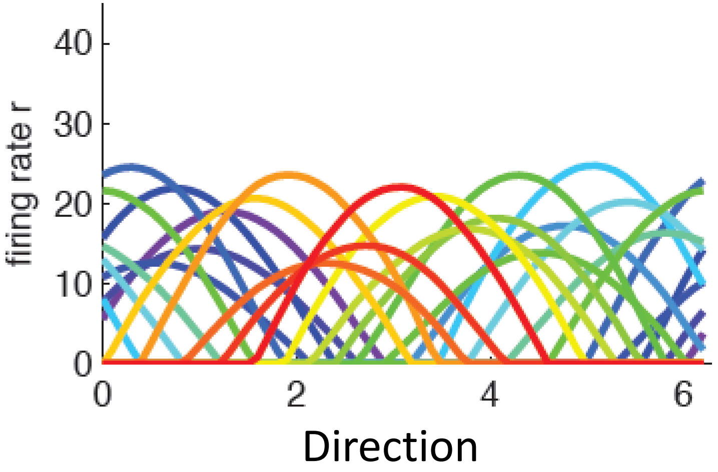


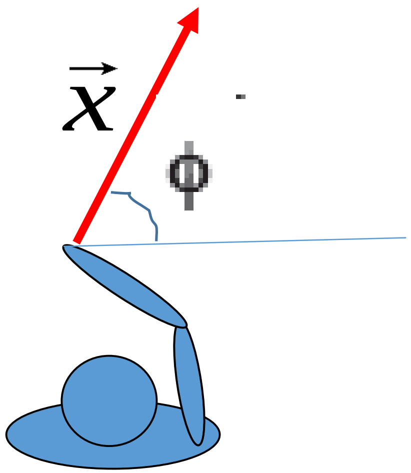
Nuno Calaim



# Direction tuning

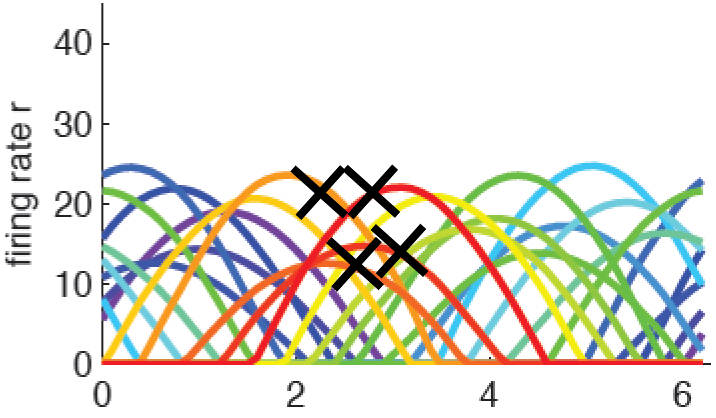
David Barrett

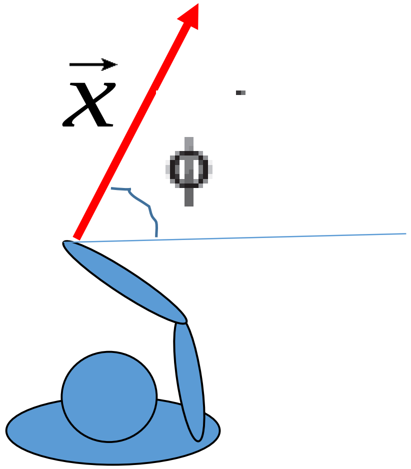


# Direction tuning

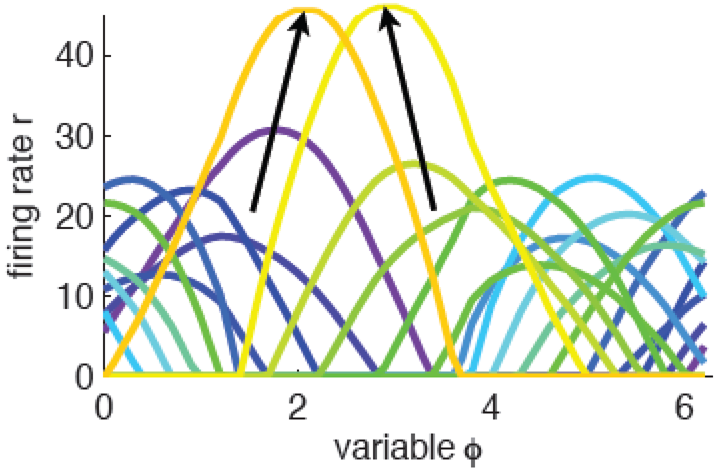
David Barrett

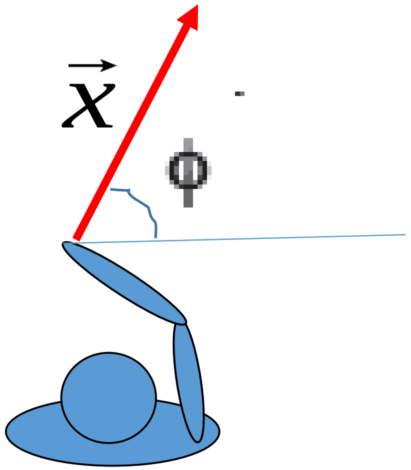


David Barrett

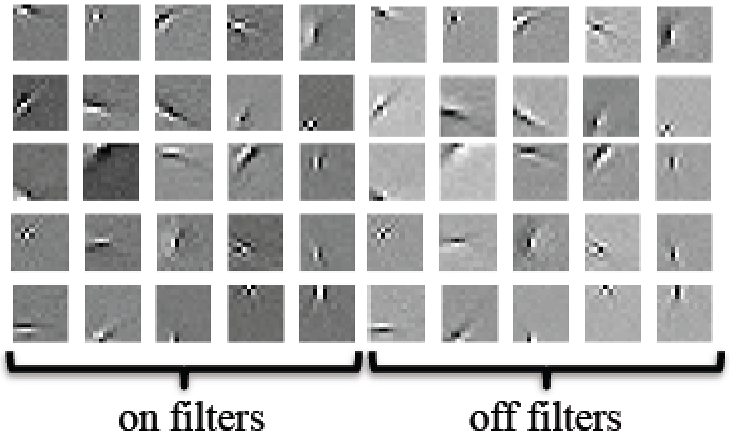
# Direction tuning

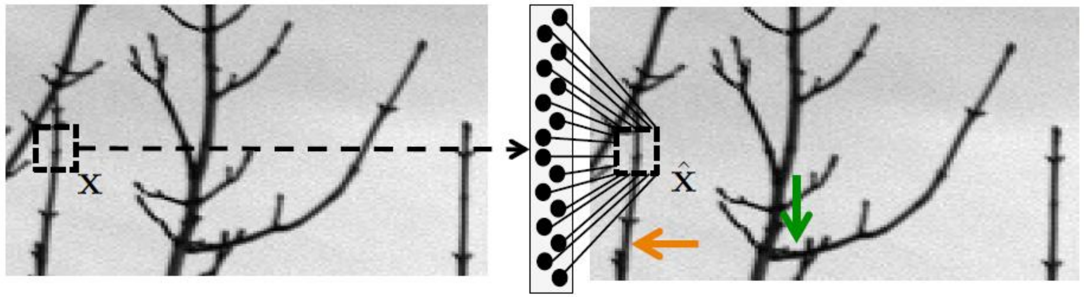


David Barrett

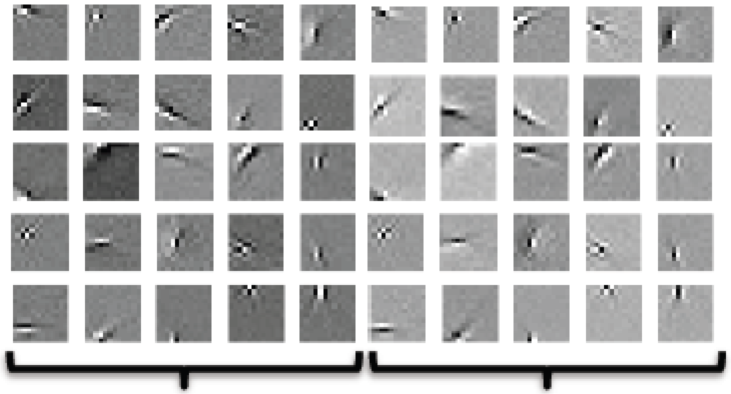
# Visual orientation tuning

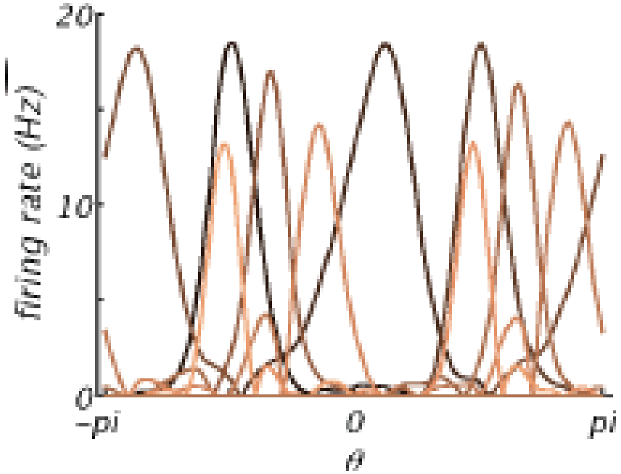


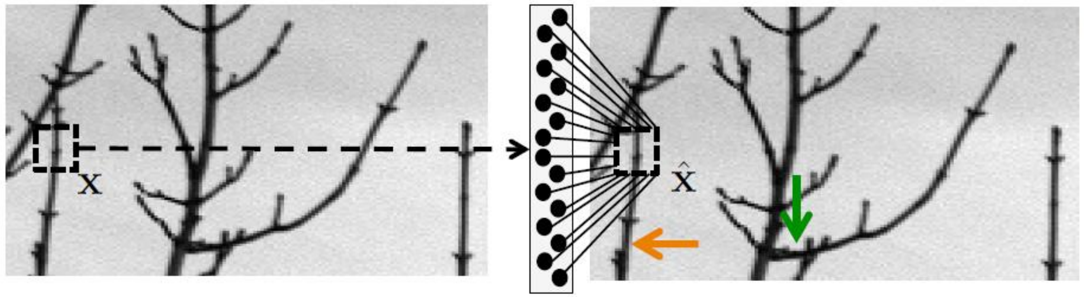



David Barrett

# Visual orientation tuning

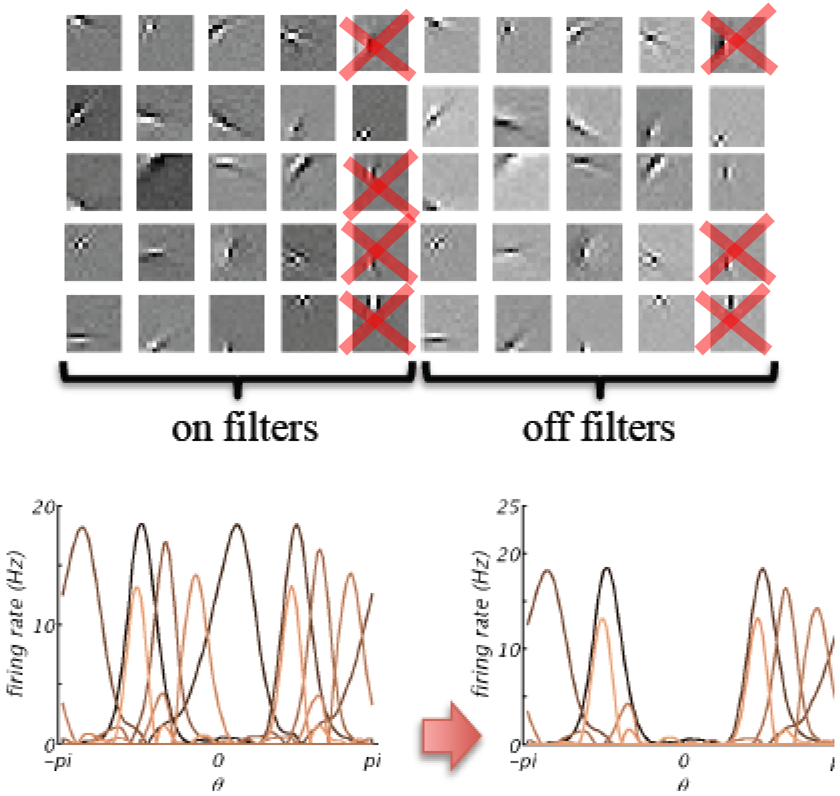


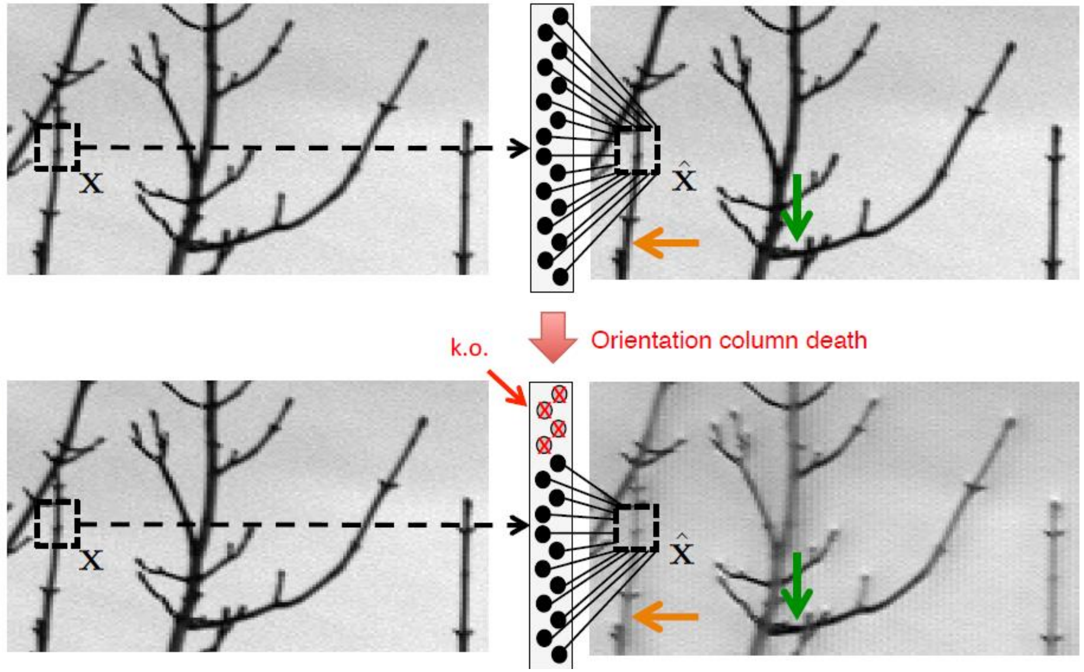



David Barrett

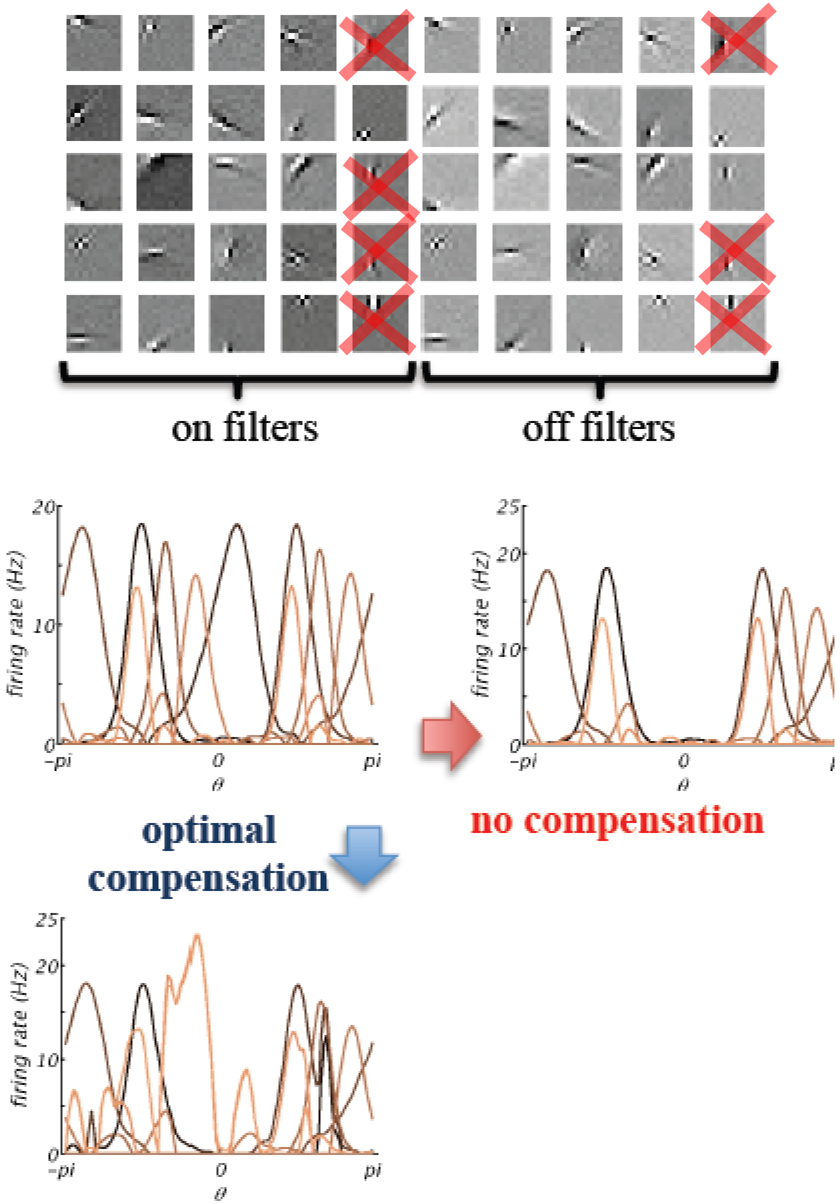
# Visual orientation tuning

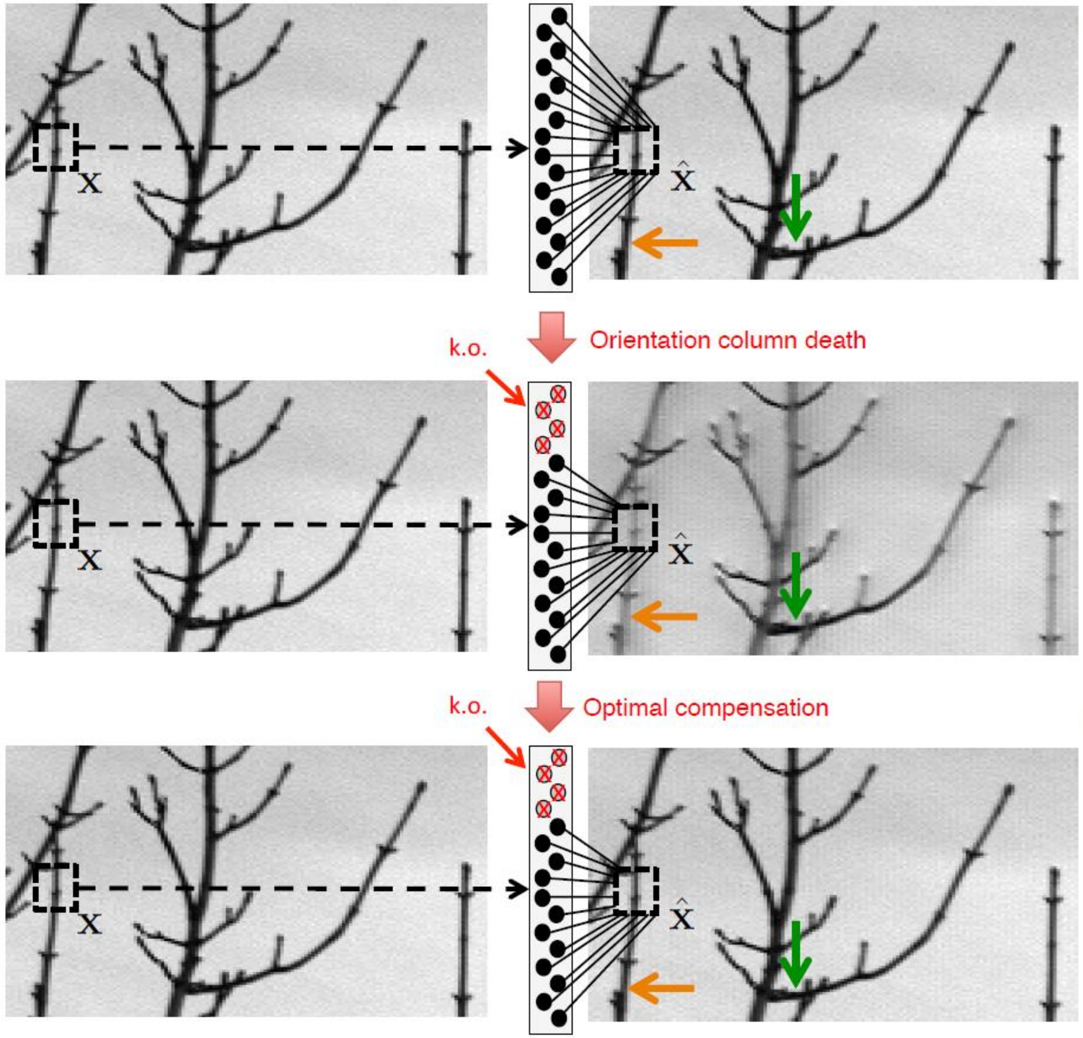


David Barrett

# Visual orientation tuning

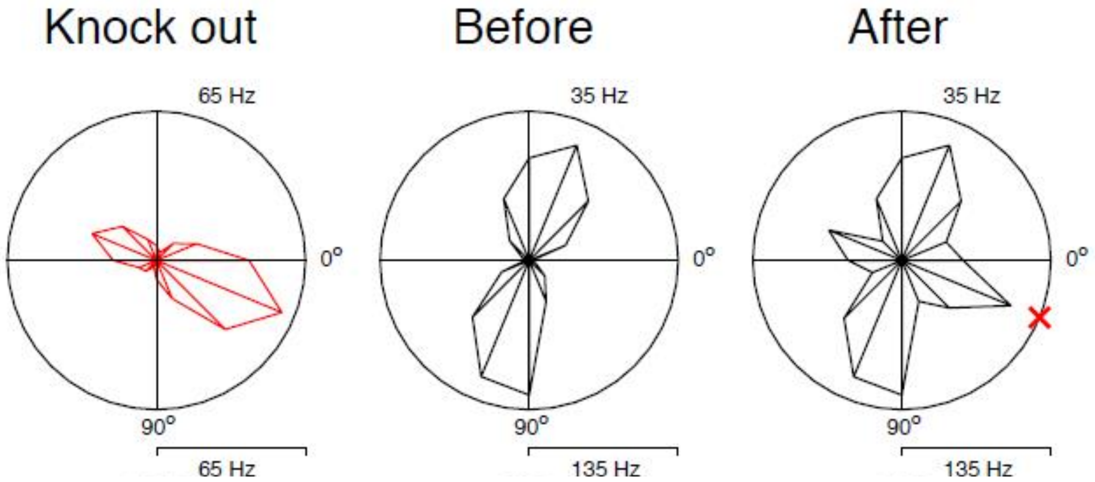


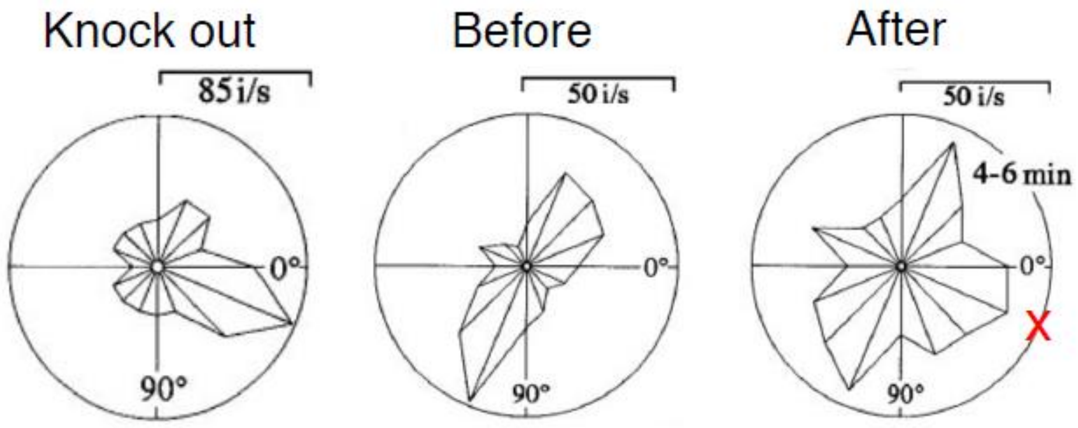
David Barrett

# Visual orientation tuning

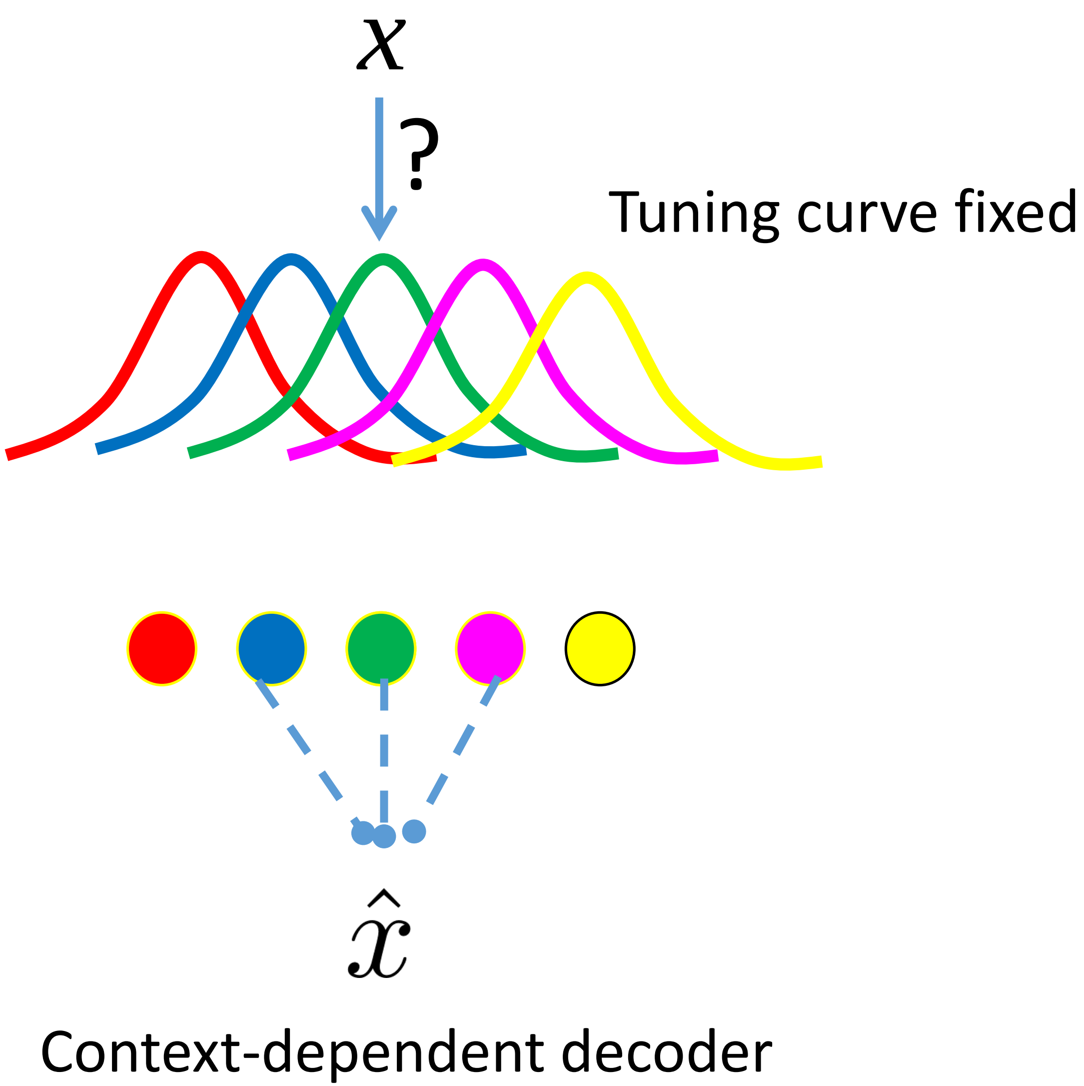
## Model



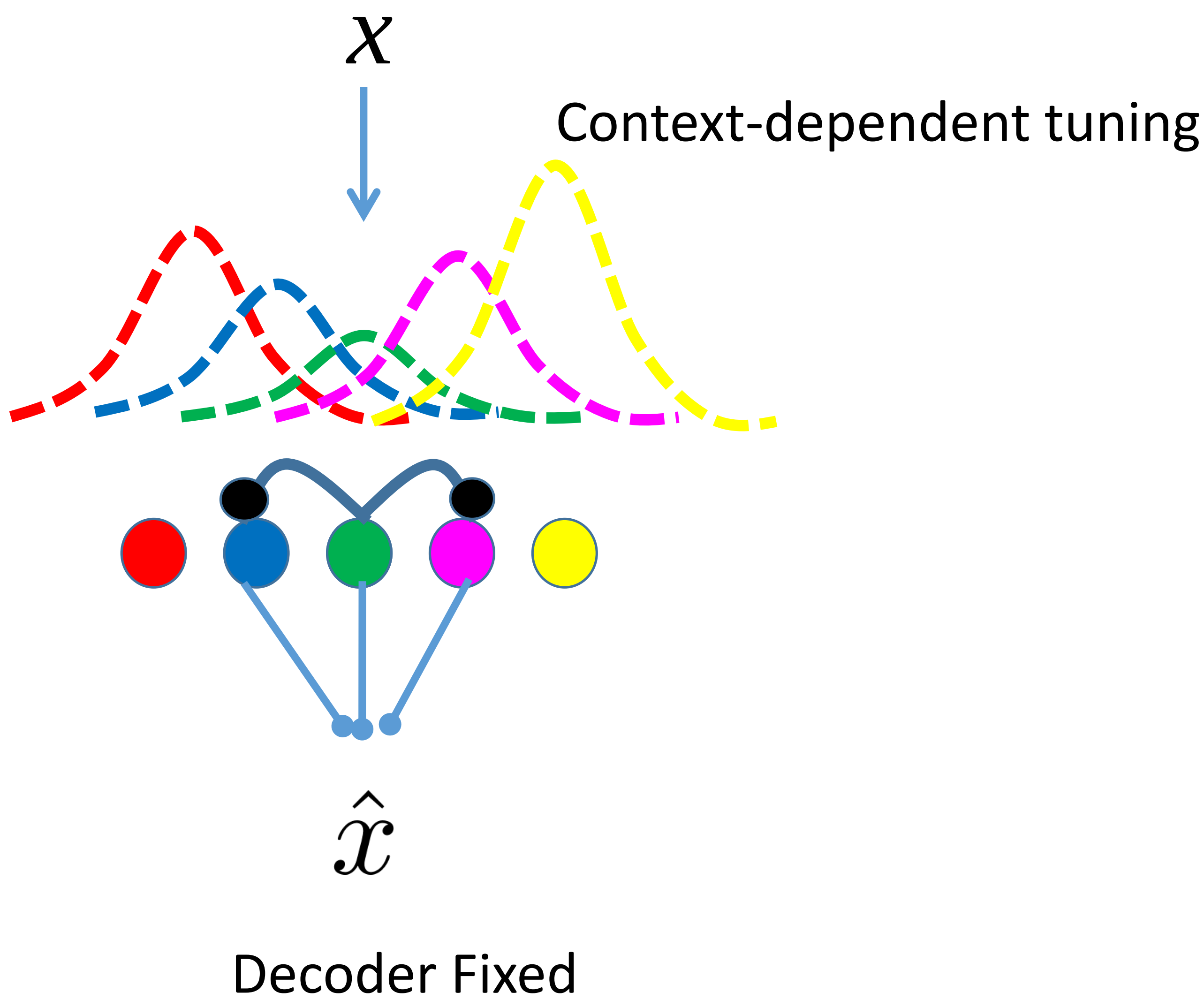

## Data (Crook and Eysel, 1992)



# Population coding




# Efficient population coding



# Neural responses= network-level solutions

- Tuning curves are highly context dependent.
- E/I balance = maximal robustness.
- Albeit encoding is complex, decoding is simple.
  Larger networks may be easier to characterize than single cells.

→ The brain might not be as “complicated” as it looks.

Wieland Brendel

Gabrielle Gutierrez

Matty Chalk

**Christian Machens**

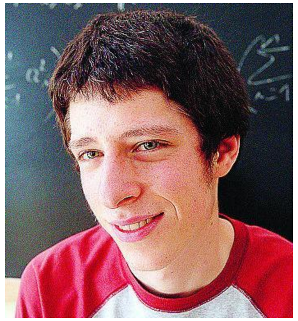
Ralph Bourdoukan

Fleur Zeldenrust

David Barrett

Veronika Koren

# Thanks for your attention!

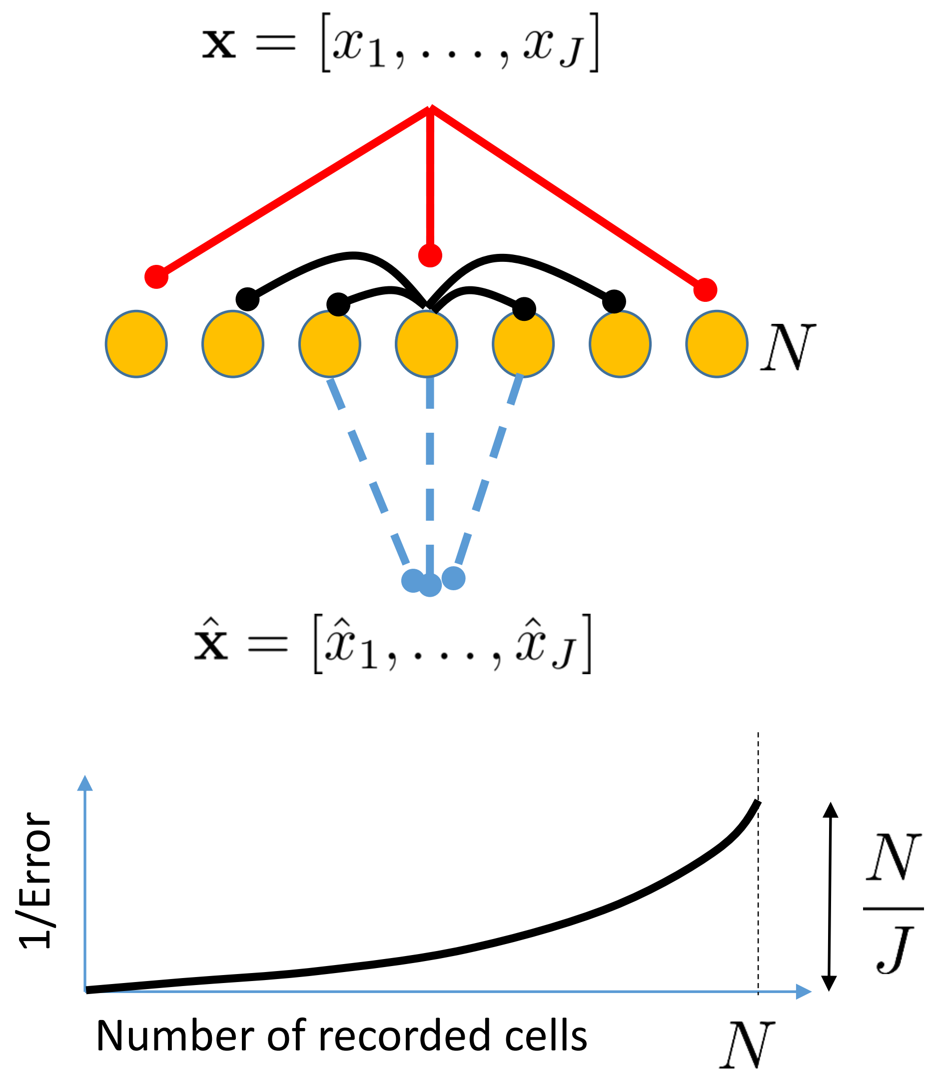
Pietro Vertechi

Martin Boerlin

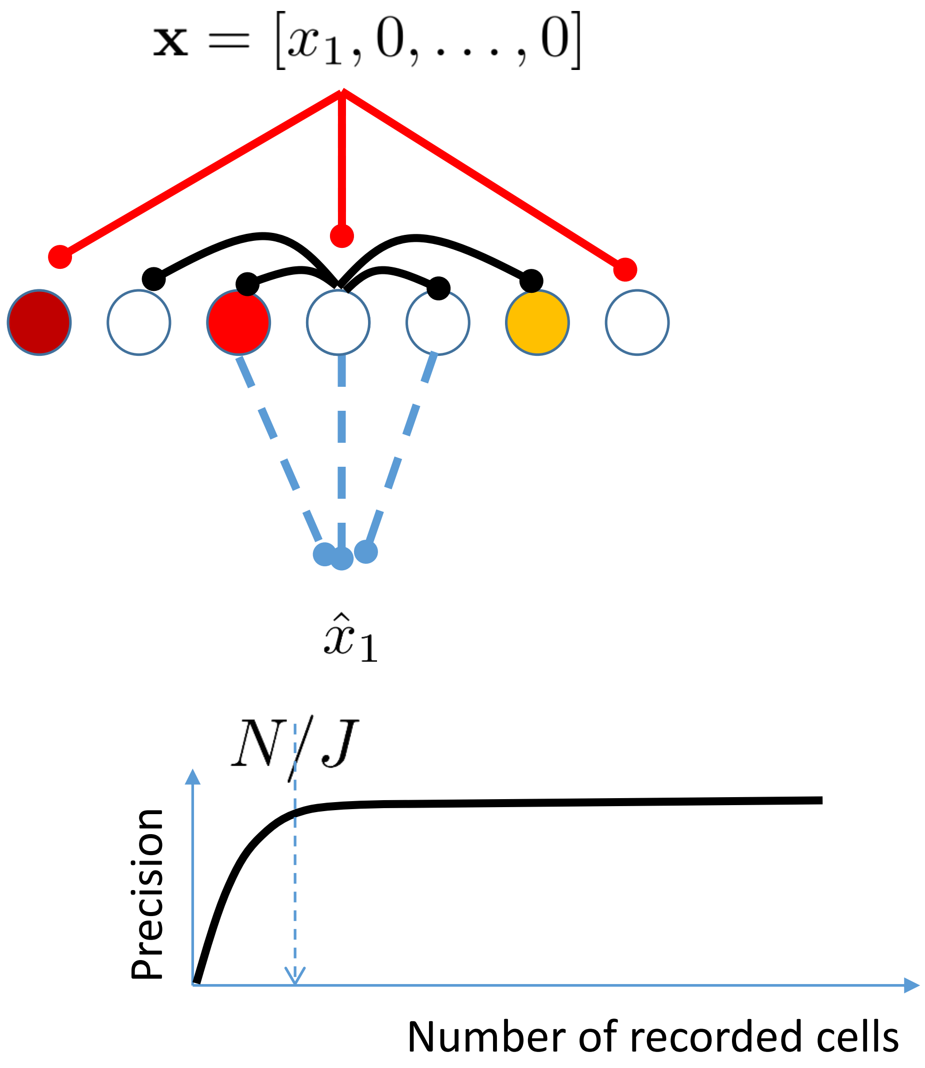
Erwan Ledoux

# Why so many neurons?

## Real life



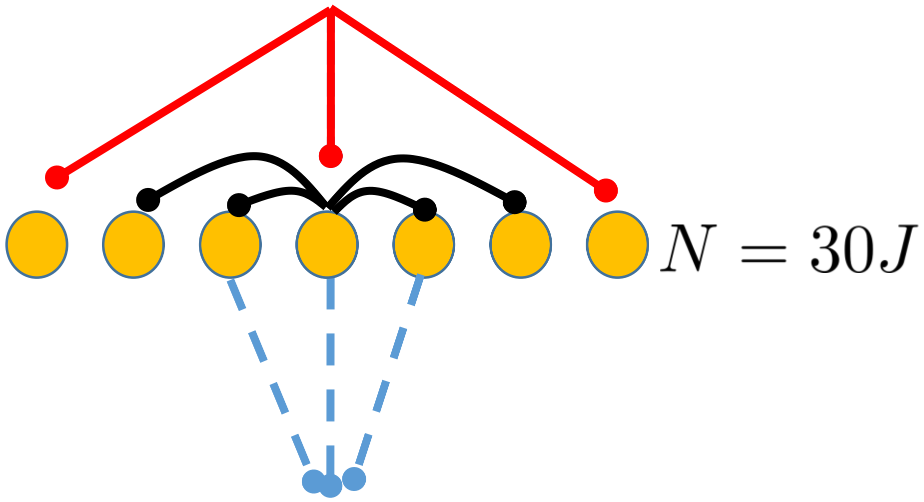

## Simple, low D stimuli



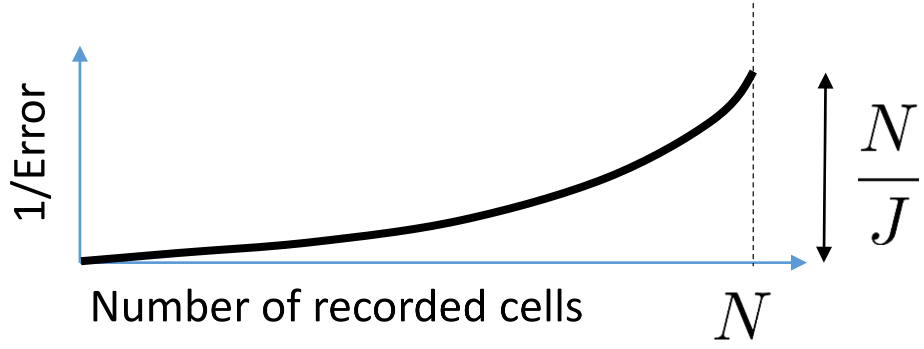
# Level of degeneracy?

## Real life

$$\mathbf{x} = [x_1, \ldots, x_J]$$

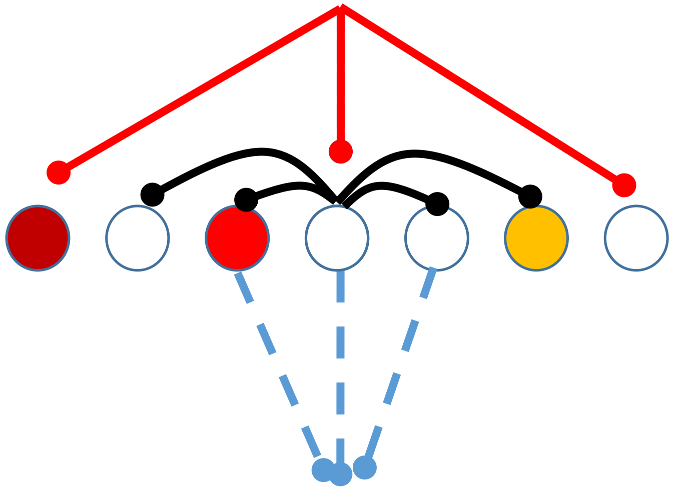



$$\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_J]$$

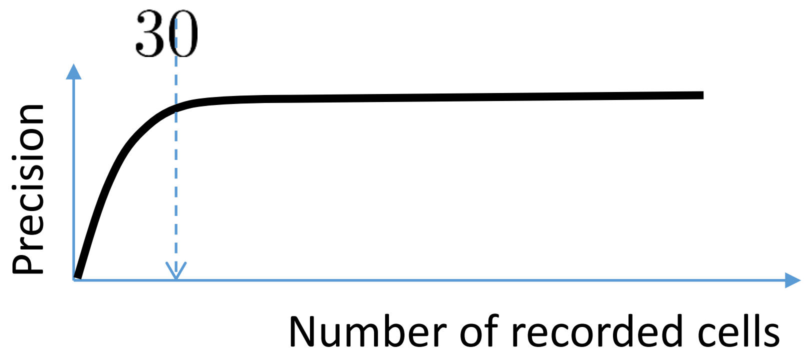



## Simple, low D stimuli

$$\mathbf{x} = [x_1, 0, \ldots, 0]$$

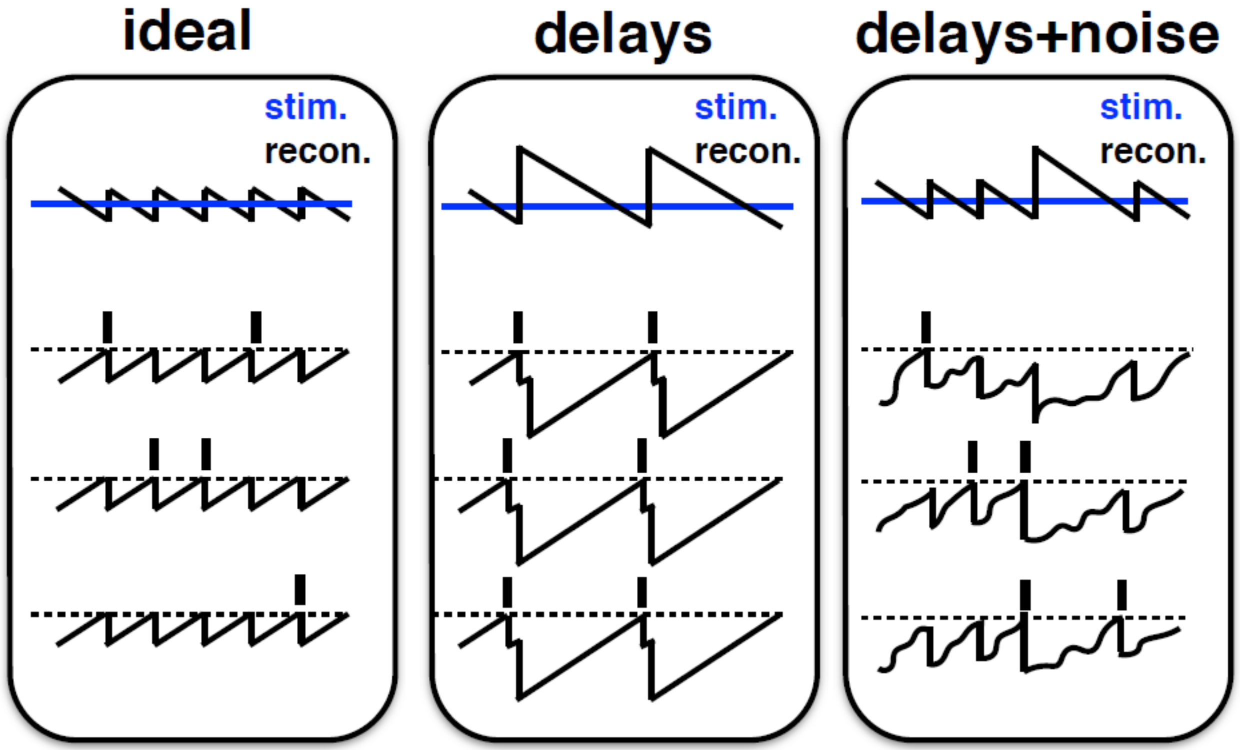


$$\hat{x}_1$$



Matty Chalk

# What if there are synaptic delays?




**Or…**

**More inputs**

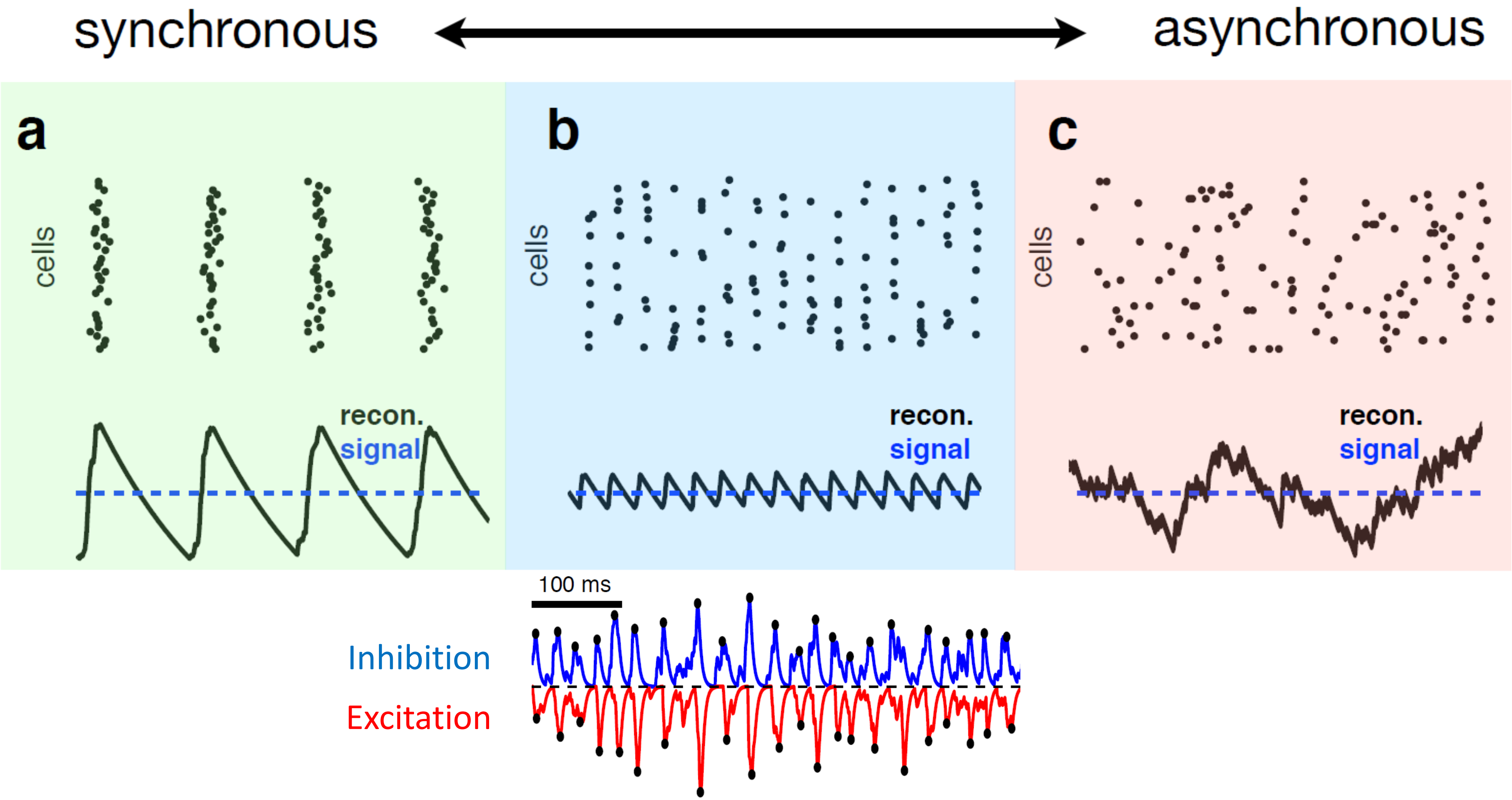
**Sparse connections**

**Synaptic failure**

**…ect….**

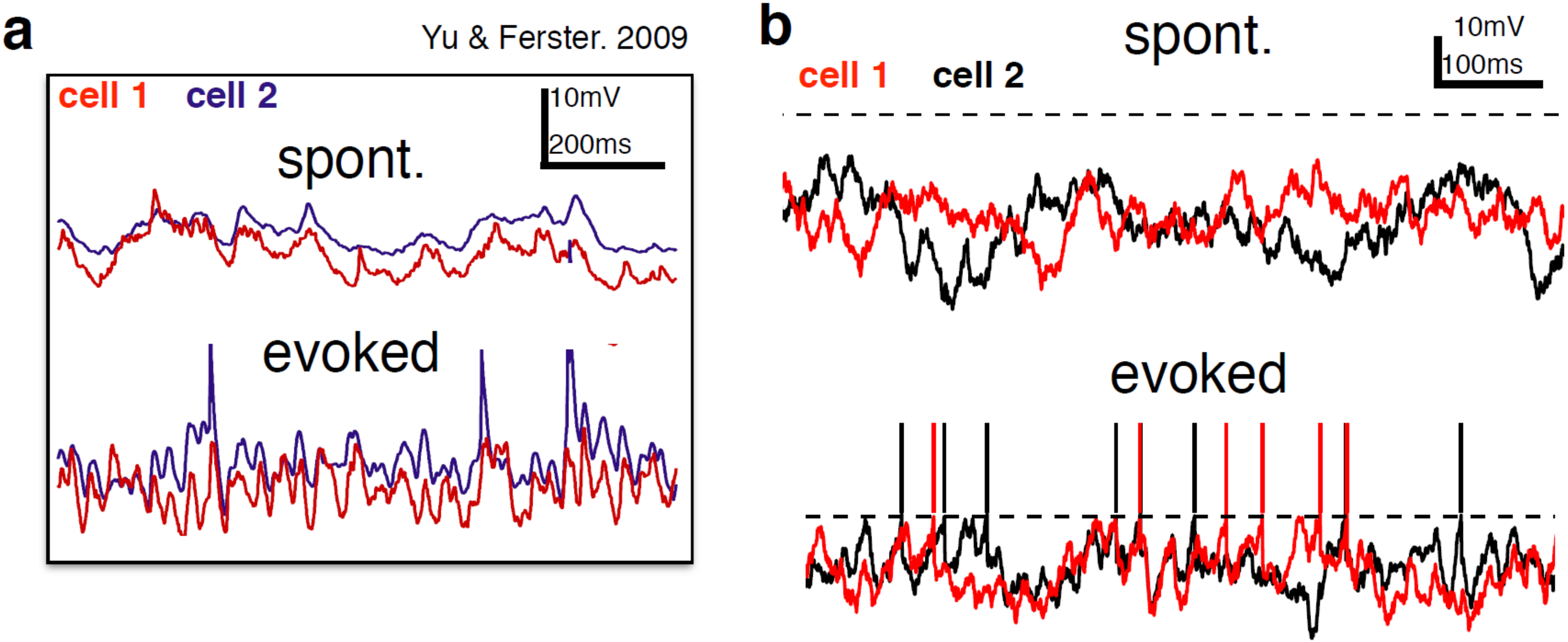
Matty Chalk

# Oscillations and predictive coding



Matty Chalk

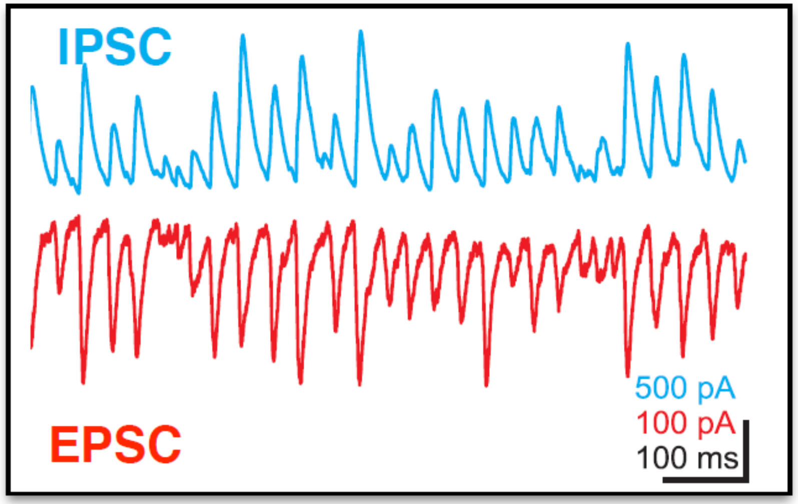
# Correlated oscillations in membrane potential

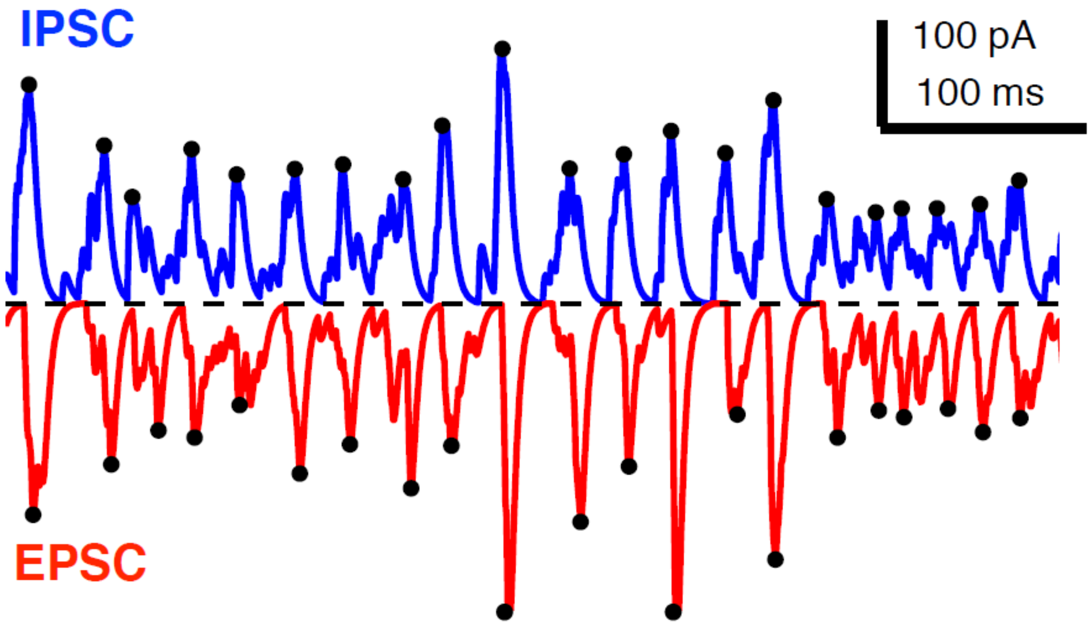


Matty Chalk

# Synchroneous oscillations in excitation and inhibition

Atallah & Scanziani. 2009

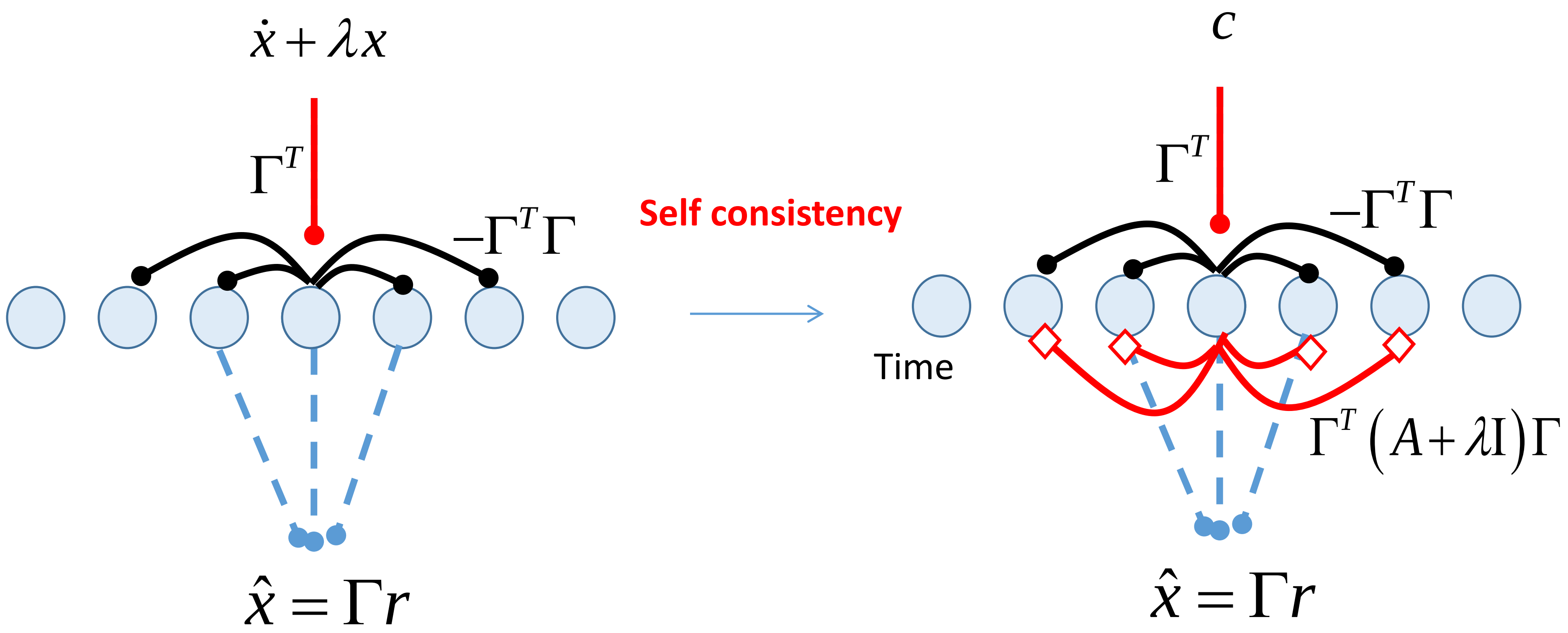


# Implementing a dynamical system

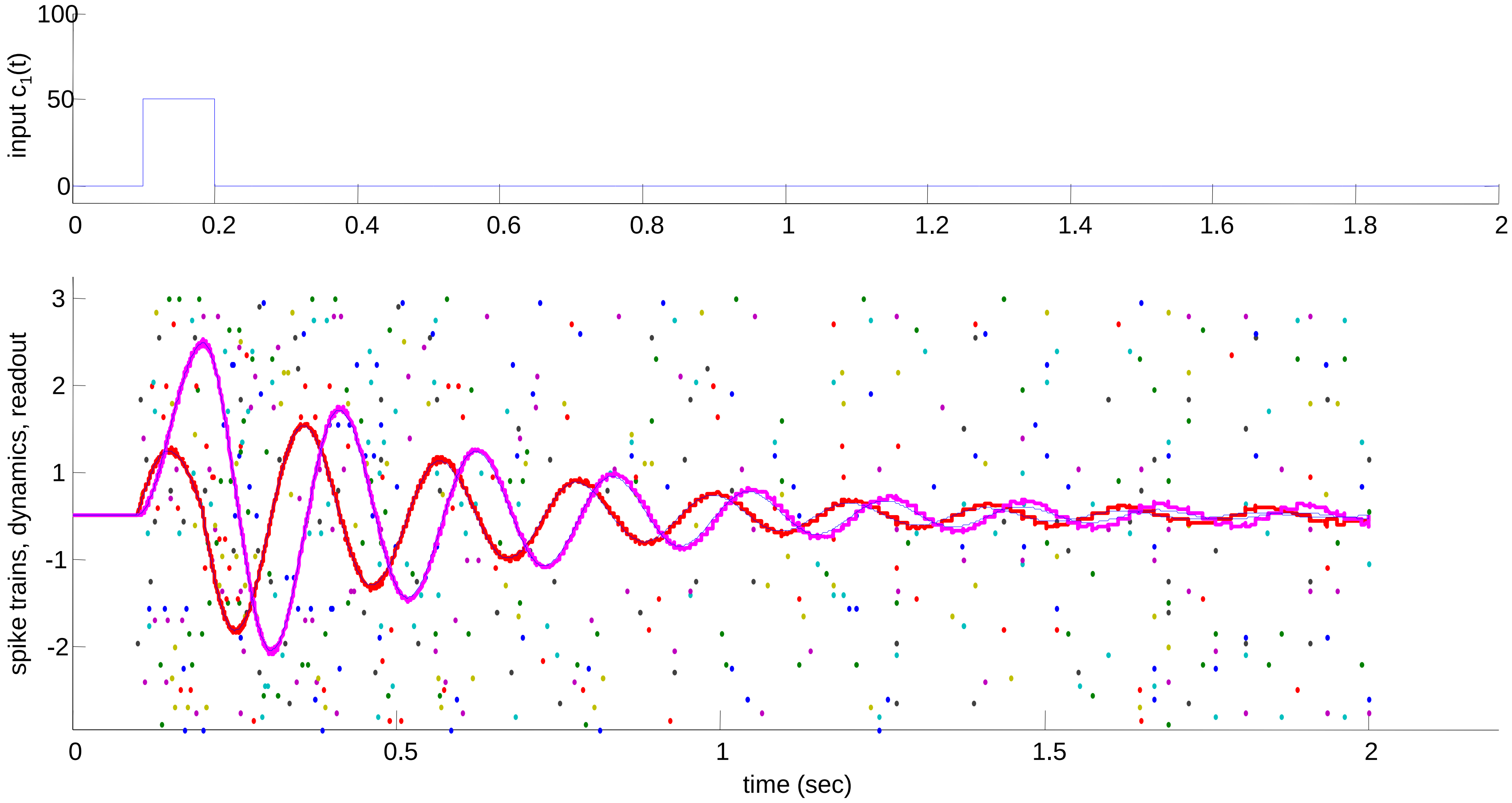
$$\dot{x} = Ax + c$$

Martin Boerlin

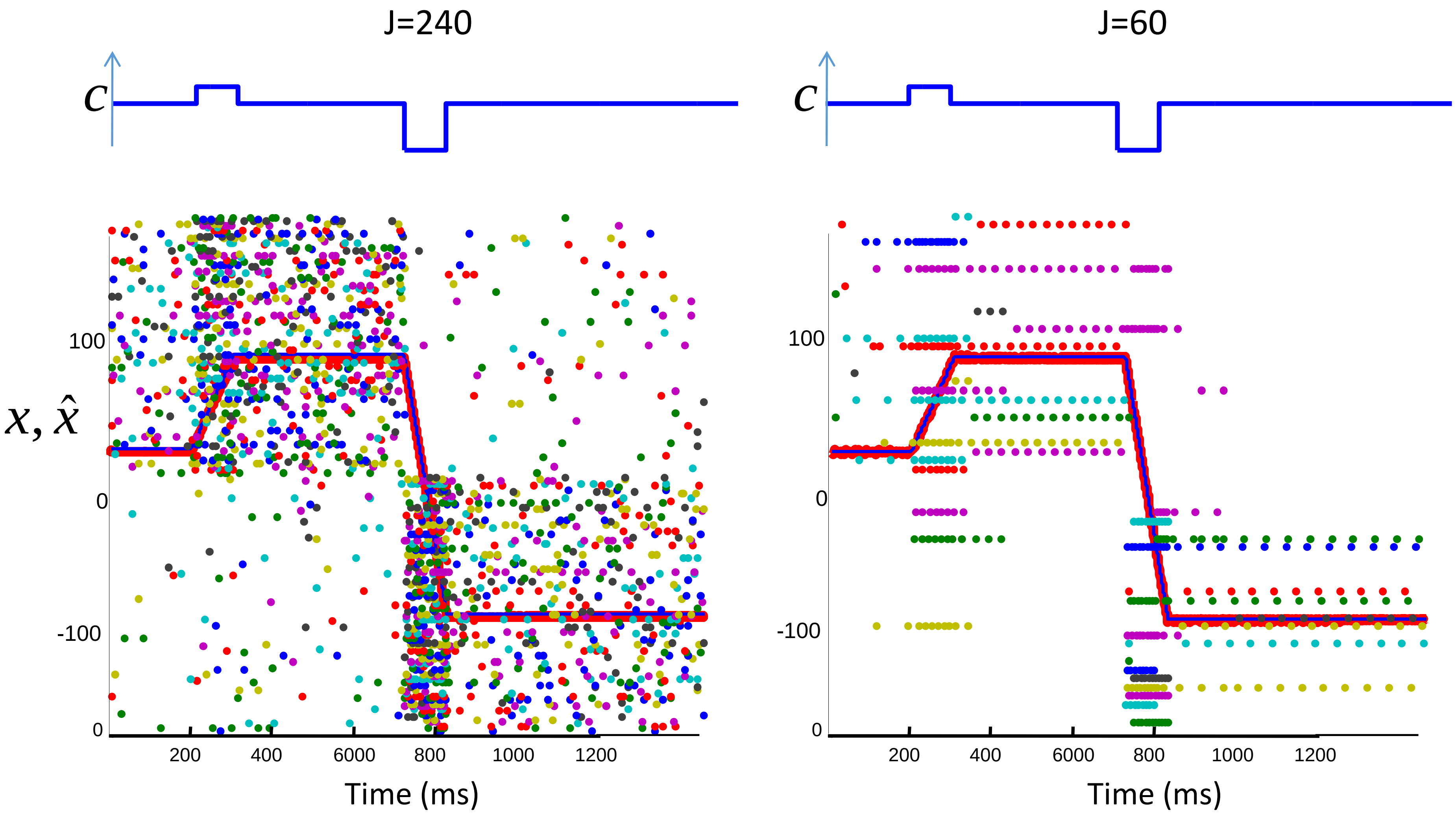


# Harmonic damped oscillator

Martin Boerlin

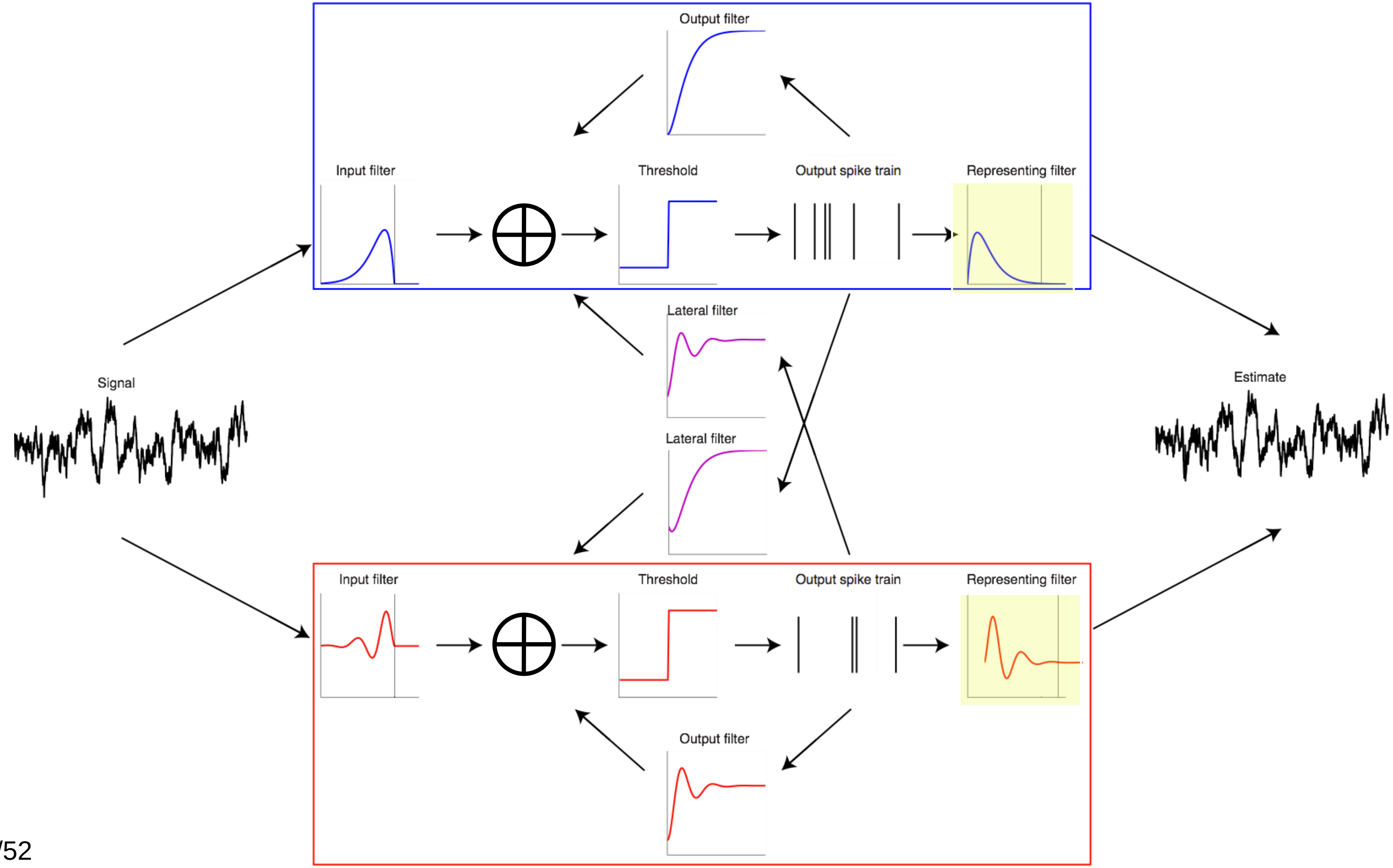


# Effect of network size



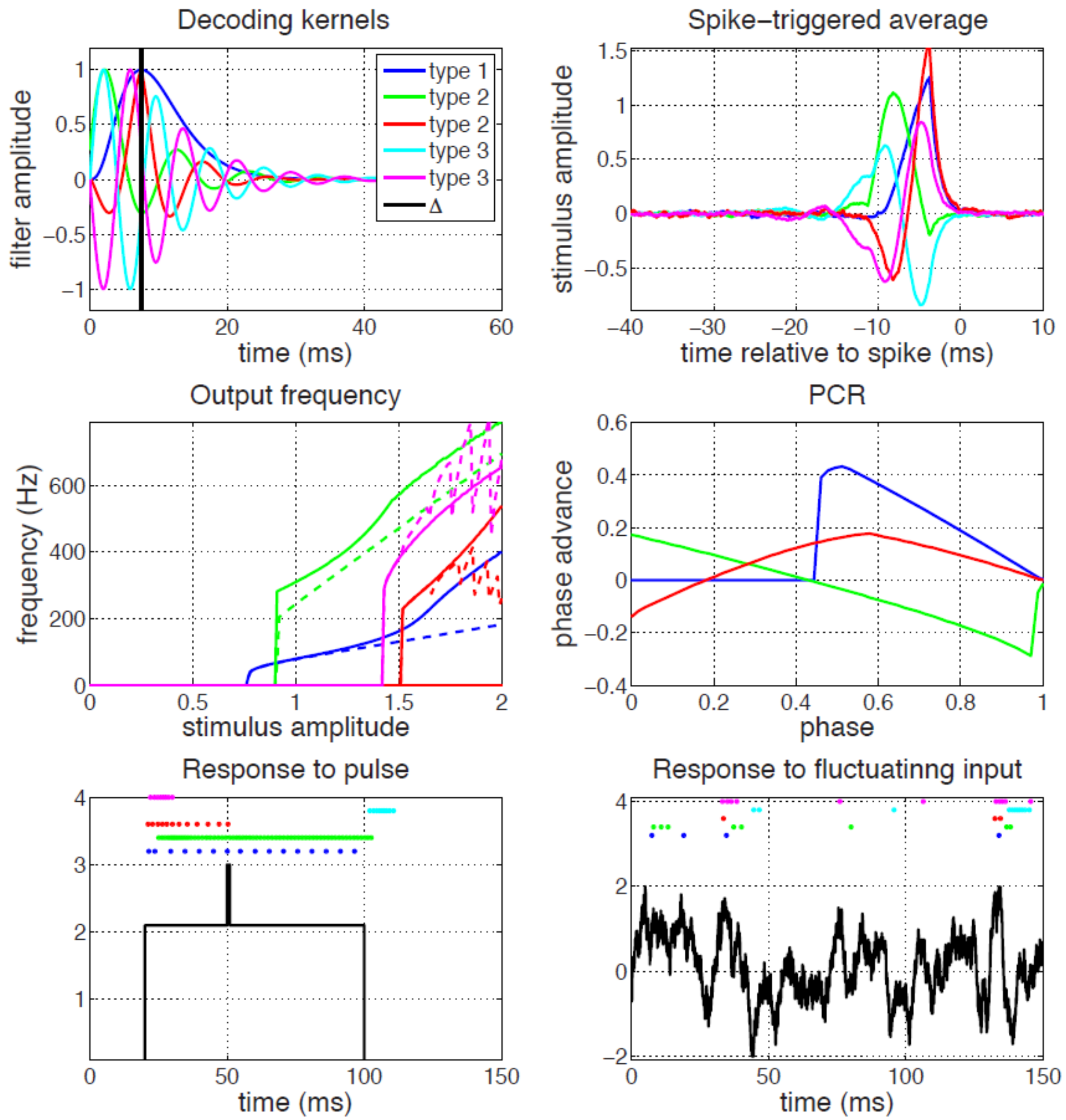
Fleur Zeldenrust

# Extension to temporal kernels: functional GLMs



Fleur Zeldenrust

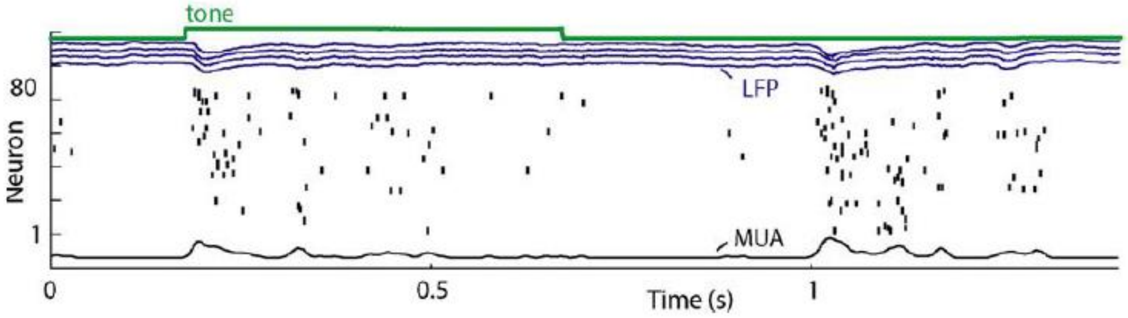
# Type I and Type II cells = different filter types



Veronika Koren

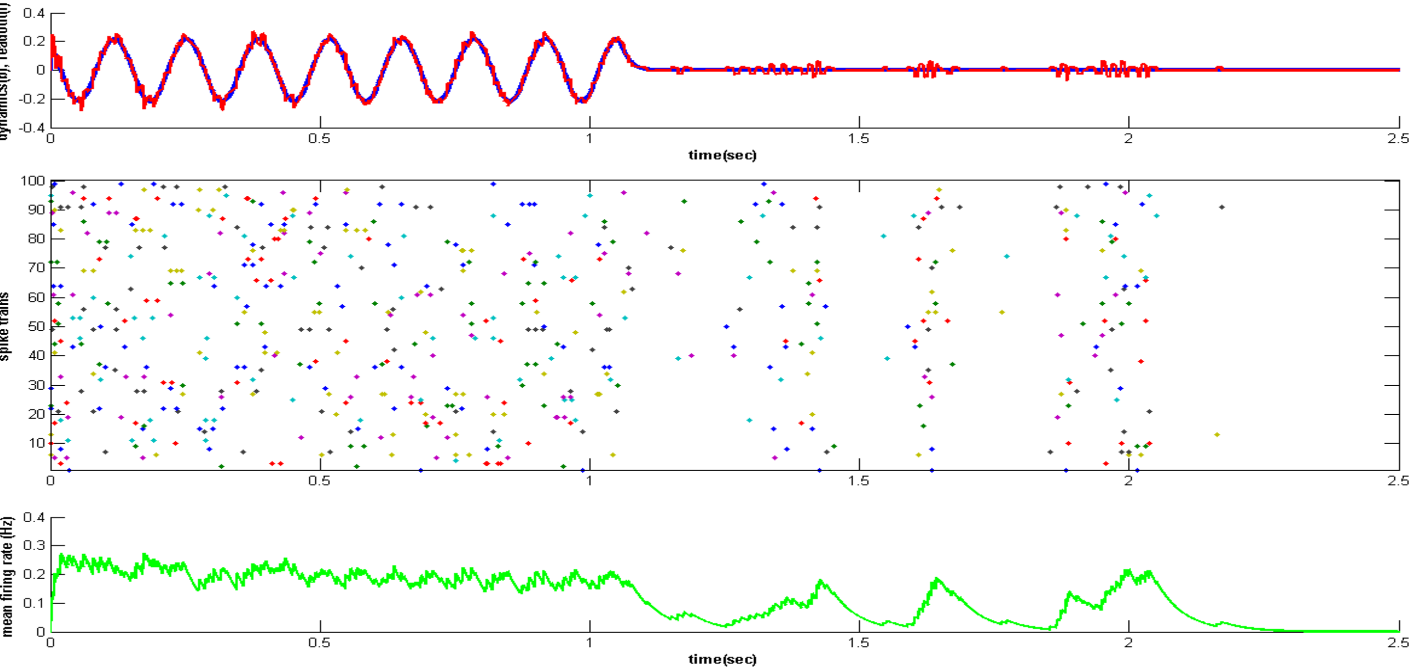
# Linear cost (threshold): Dompting the noise

## Quiescent “up” states



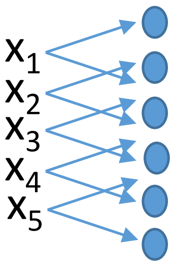

Luczak, Bartho, Harris, *Neuron*, 2009

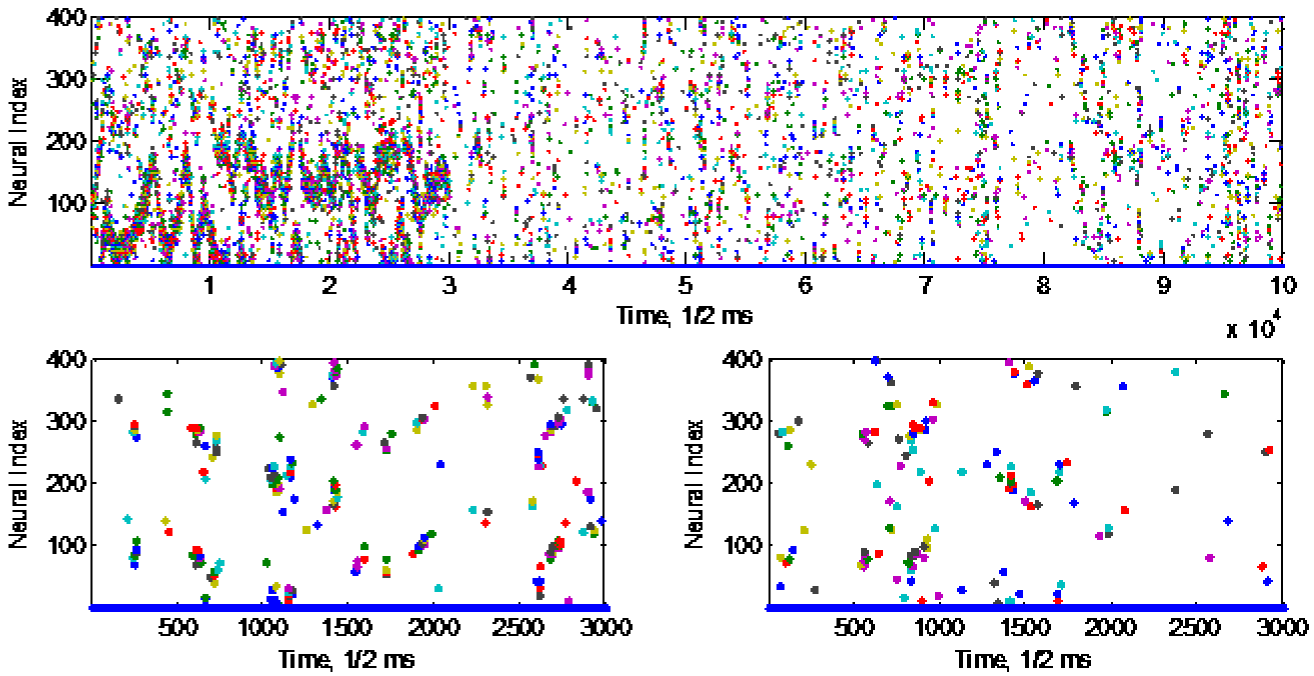
## In the model:

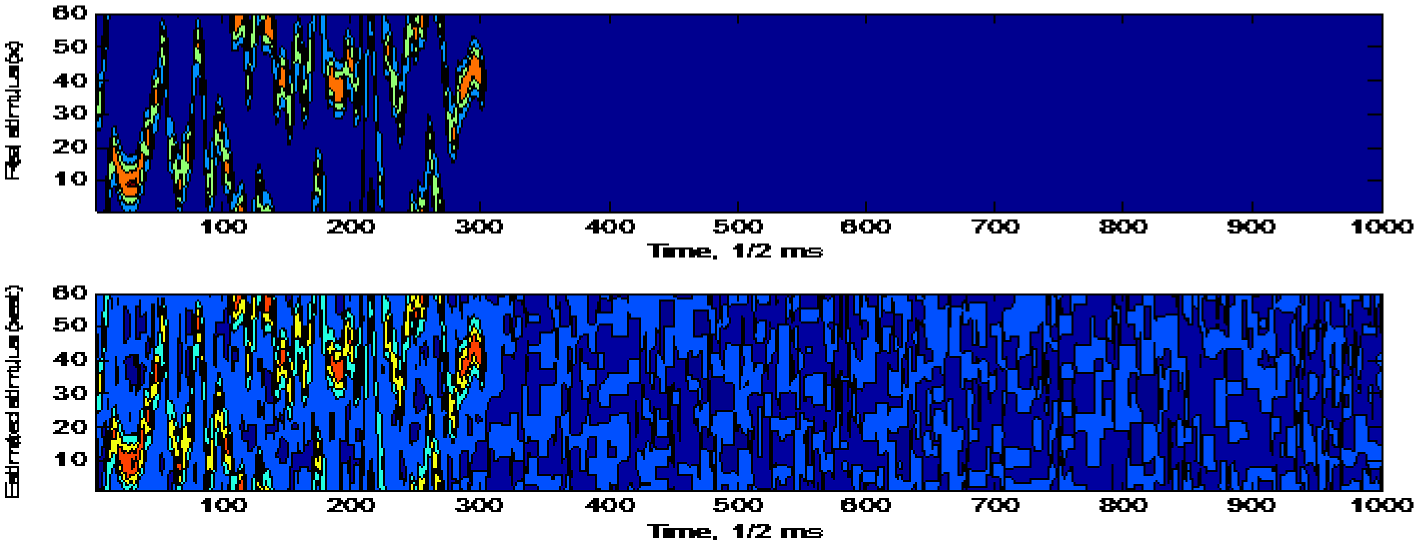


# With synaptic delays, topology

Veronika Koren

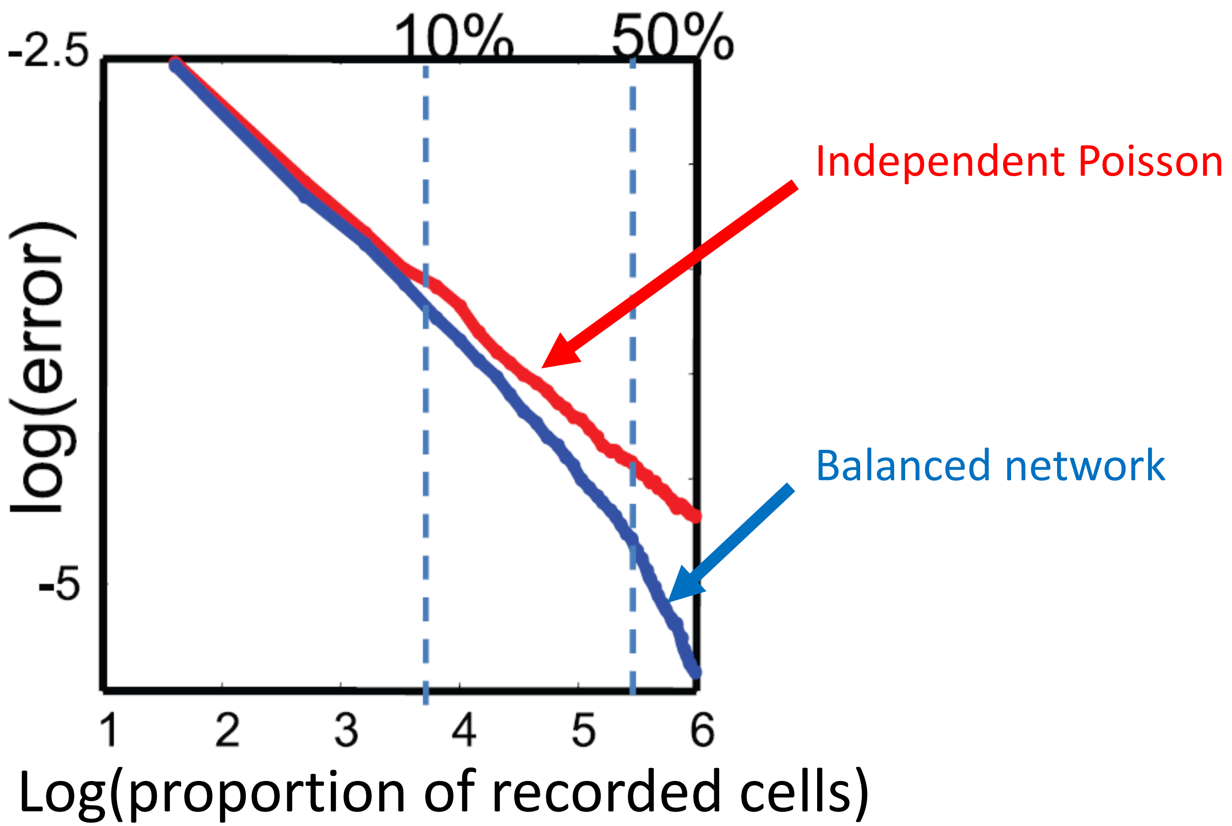


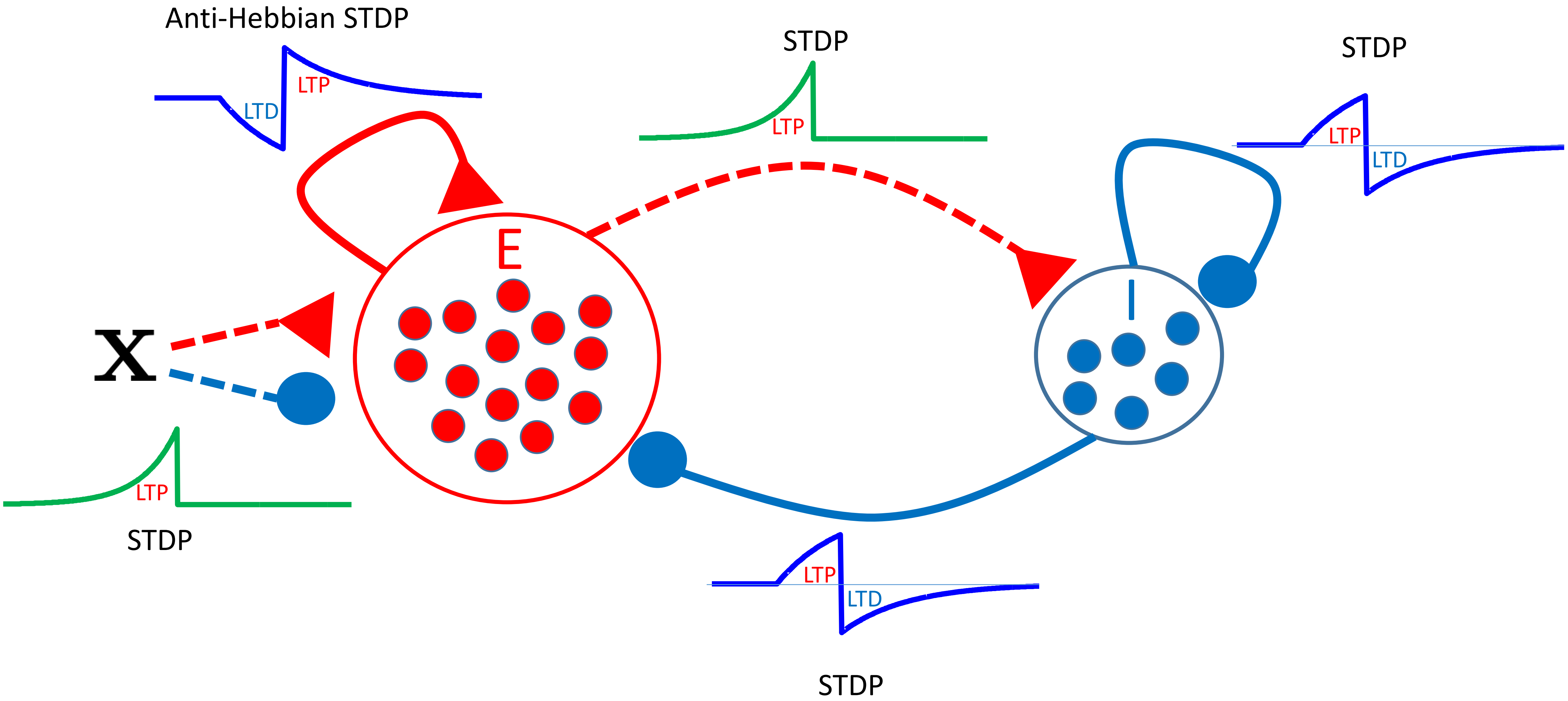



Martin
Boerlin

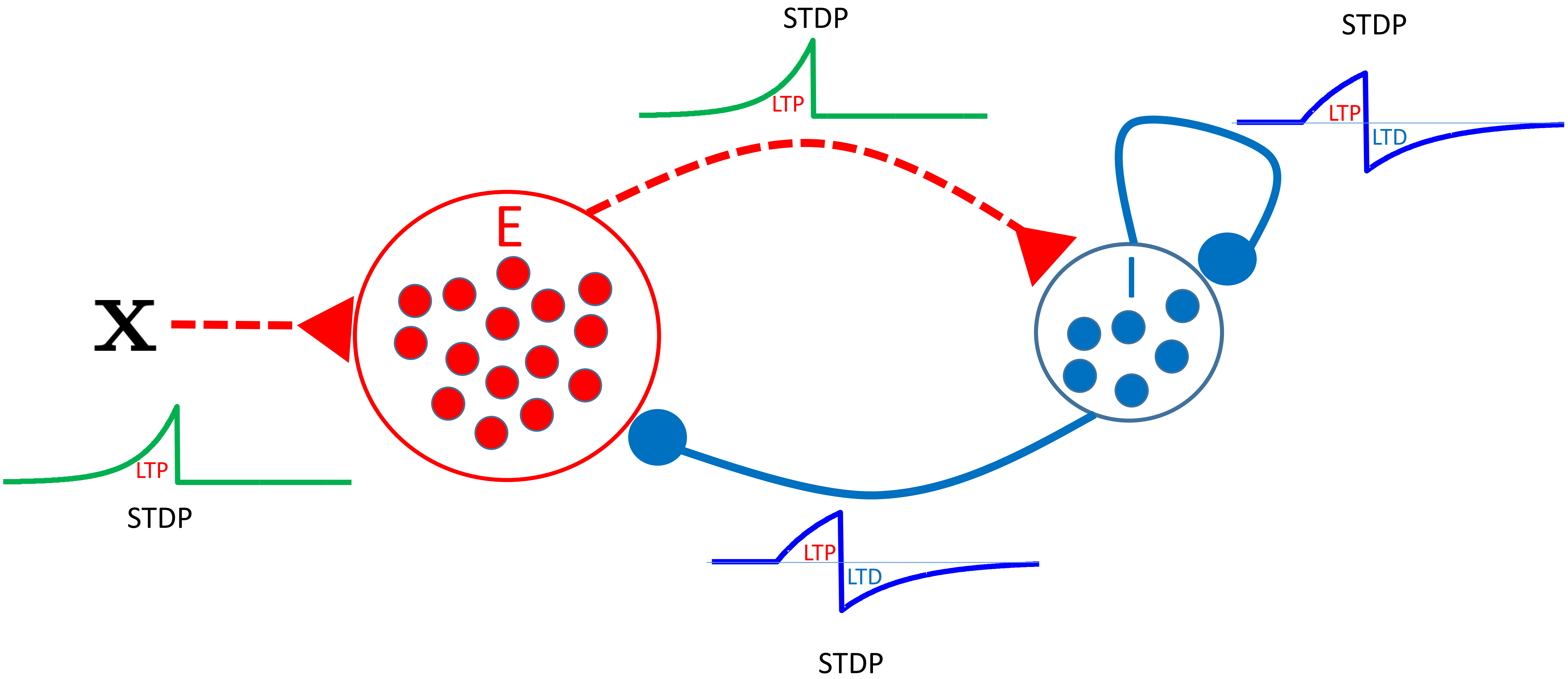
# Neural variability = Degeneracy, not noise



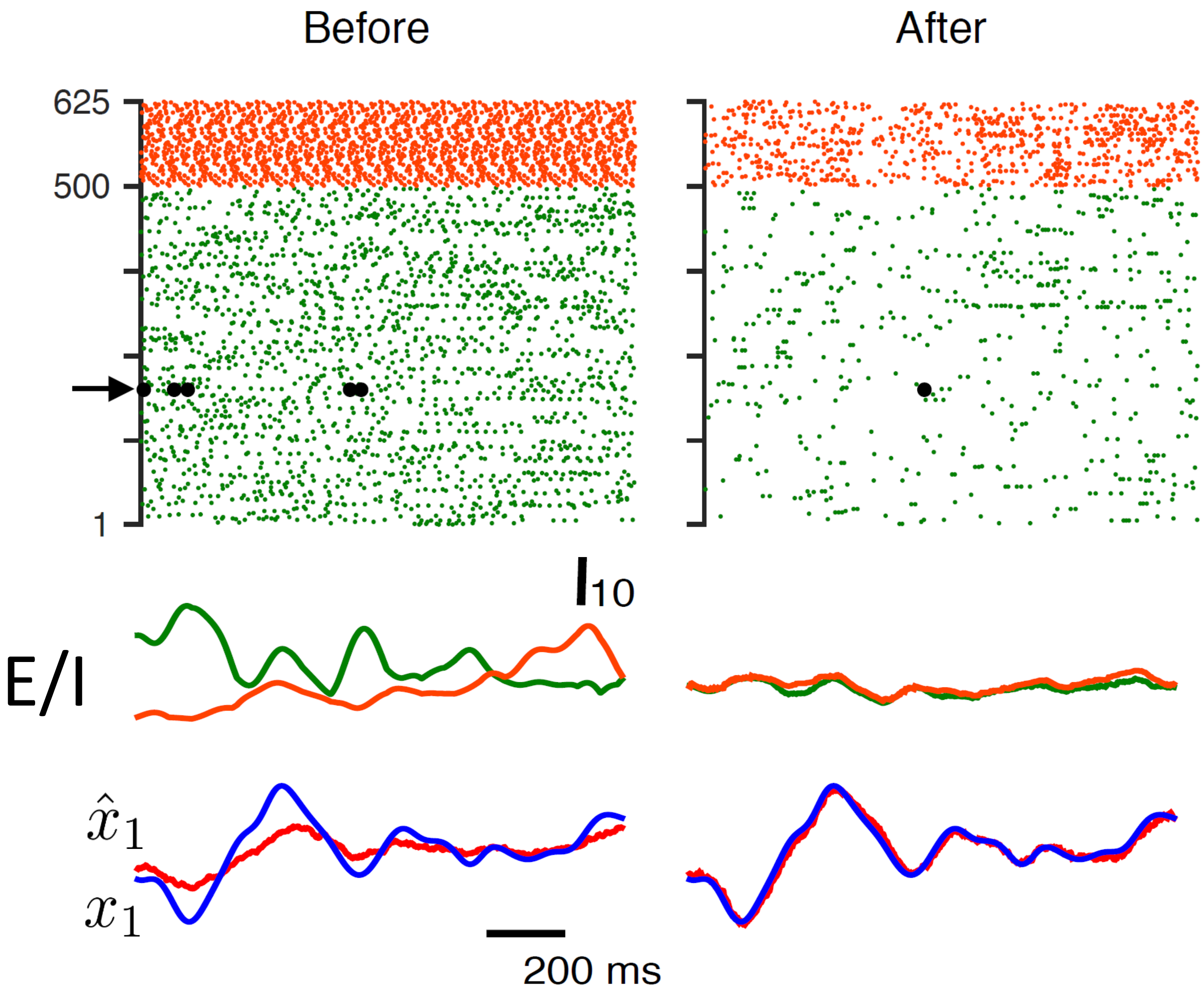
# Respecting Dale's Law



# Purely excitatory feedforward inputs

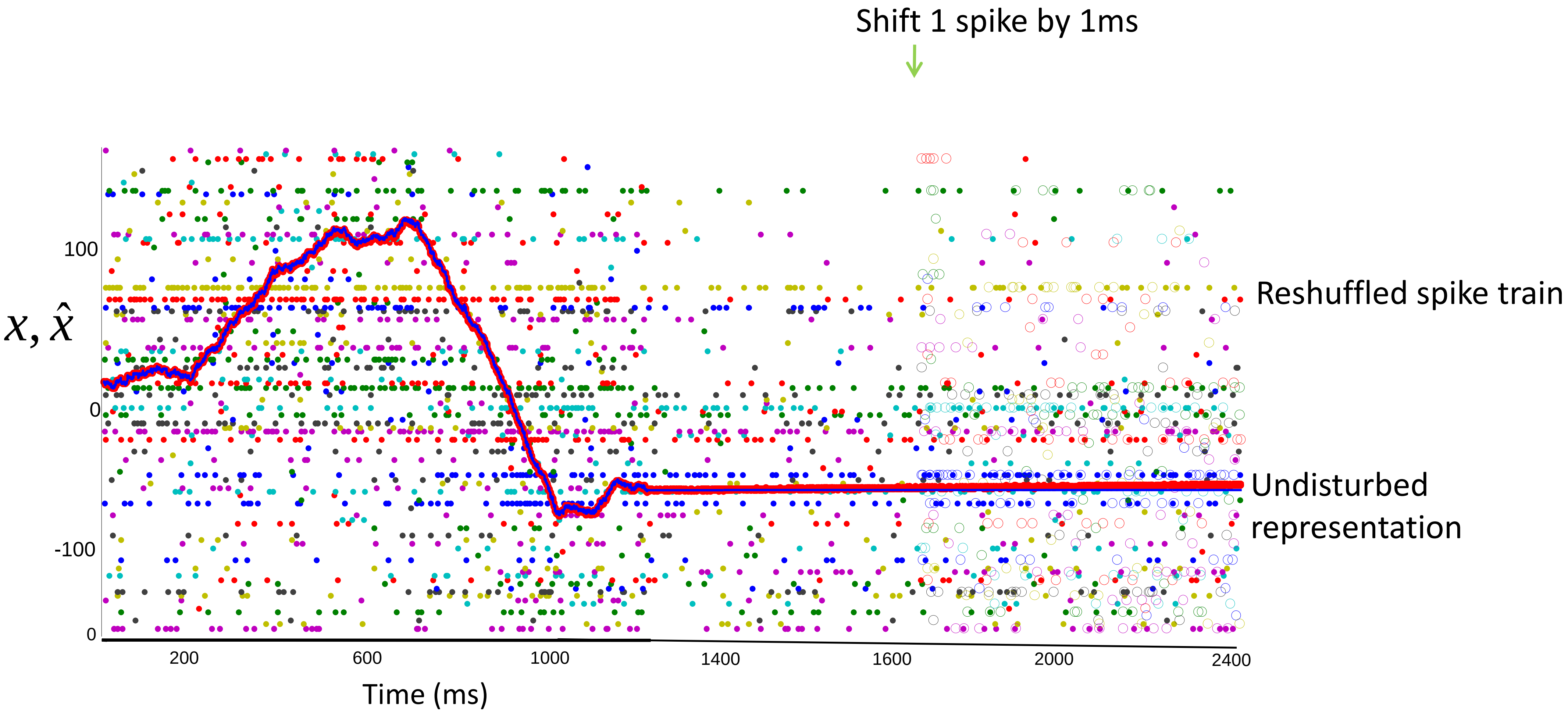


Ralph
Bourdoukan



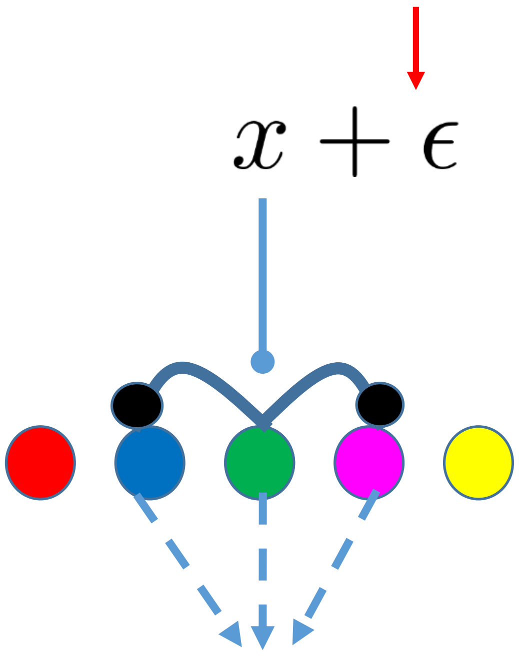
Martin Boerlin

# Neural variability = Degeneracy, not noise

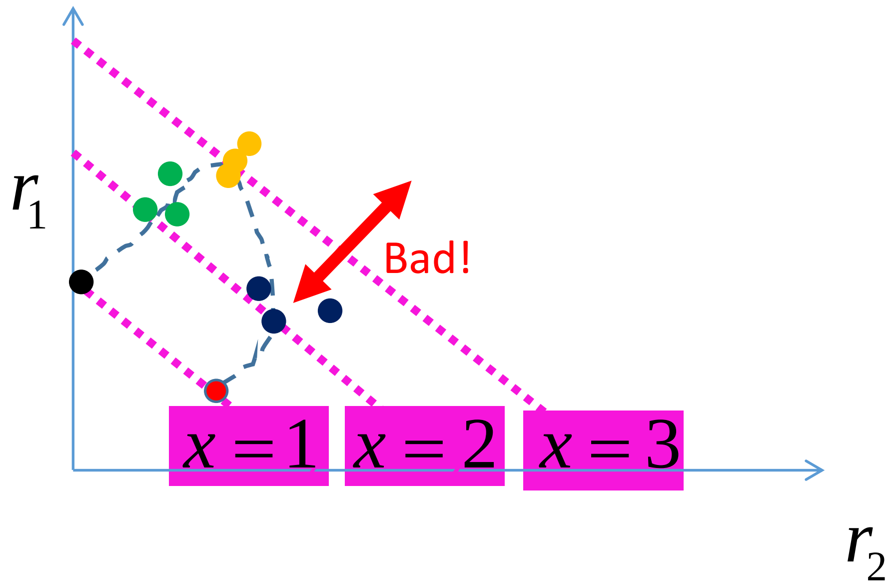


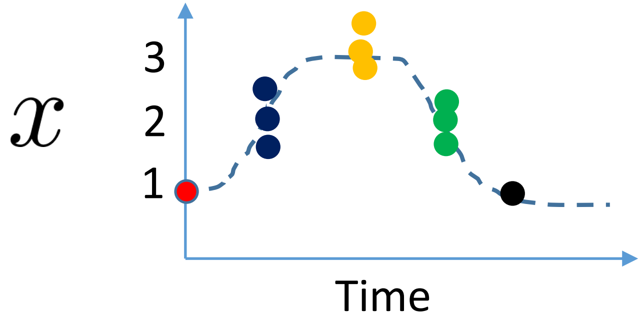
# Neural variability = Degeneracy, not noise

Uncontrolled variables




$$\hat{x} = \sum_j \Gamma_j r_j$$



# Adaptation as cost optimization

$\tau \qquad \tau_a > \tau$

$$\mathbf{r} = \arg\min_{\mathbf{r}^* > \mathbf{0}} \left( \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \mu \sum_i {r_i^a}^2 \right)$$

Cumulative cost for high firing rates

# Adaptation as cost optimization

$$\tau \qquad\qquad \tau_a > \tau$$

$$\mathbf{r} = \underset{\mathbf{r}^* > \mathbf{0}}{\arg\min} \left( \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \mu \sum_i {r_i^a}^2 \right)$$

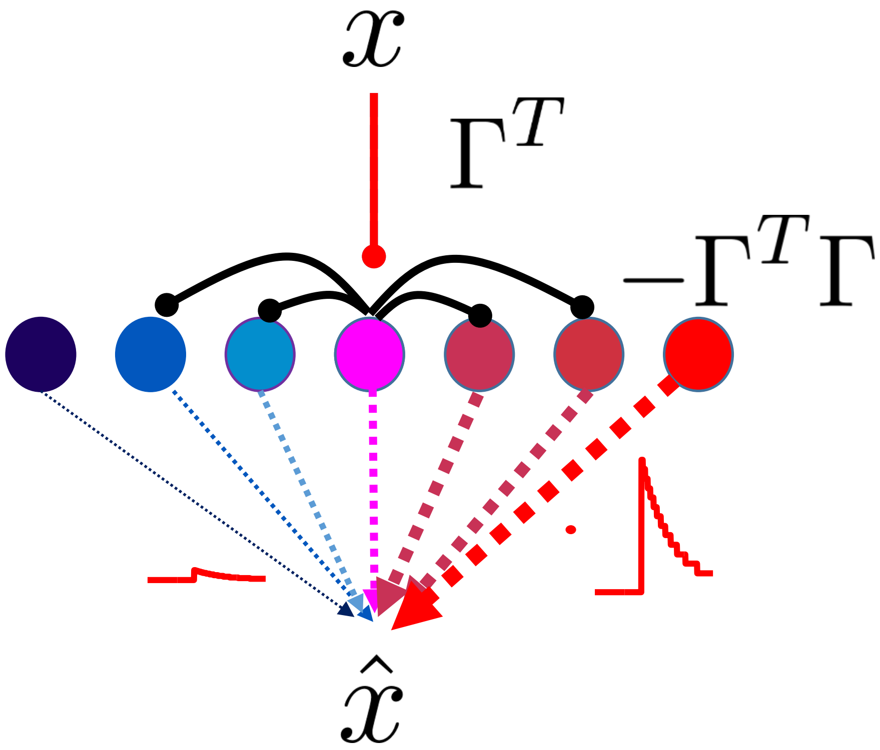
Cumulative cost of firing

$$V_i = \sum_i \Gamma_{ij} x_j - \sum_k \Omega_{ik} r_k - \boxed{\mu r_i^a}$$

Feedforward Input    Recurrent inhibition    Activity dependent suppression

# Adaptation as cost optimization

Gabrielle Gutierrez

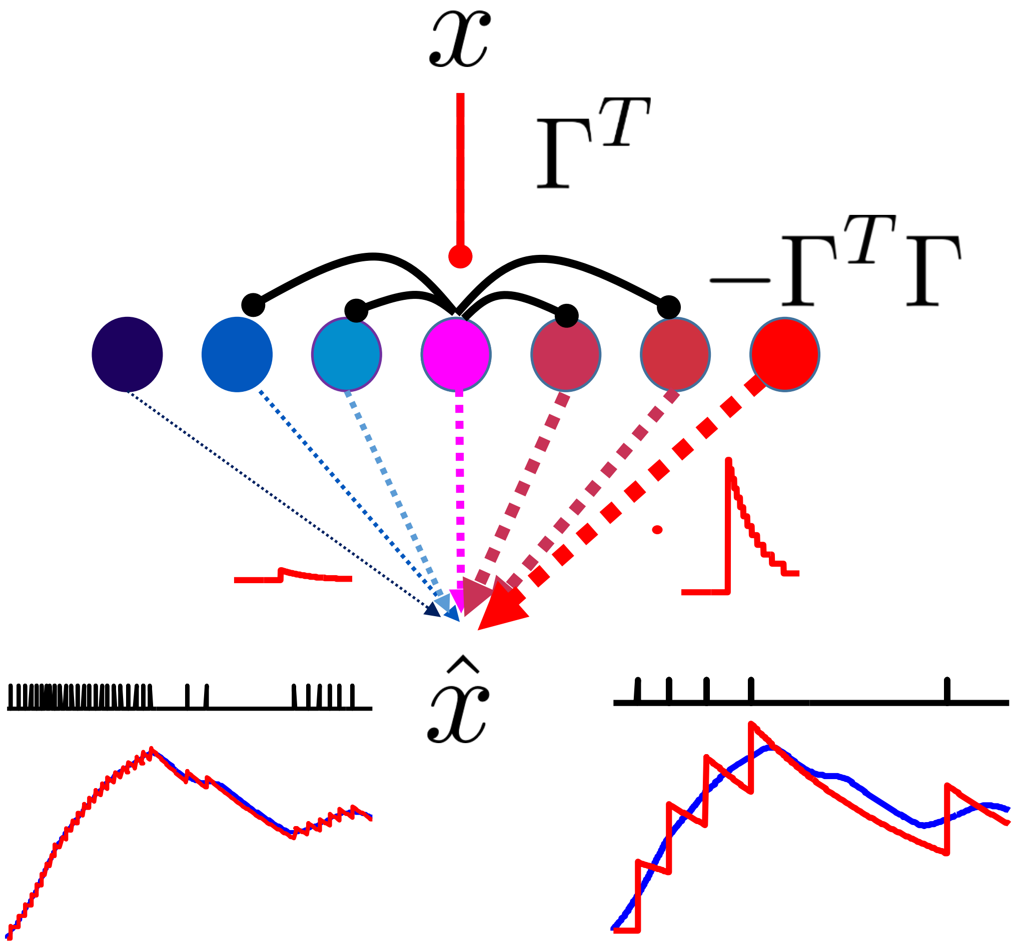



Precise, costly

Imprecise, cheap

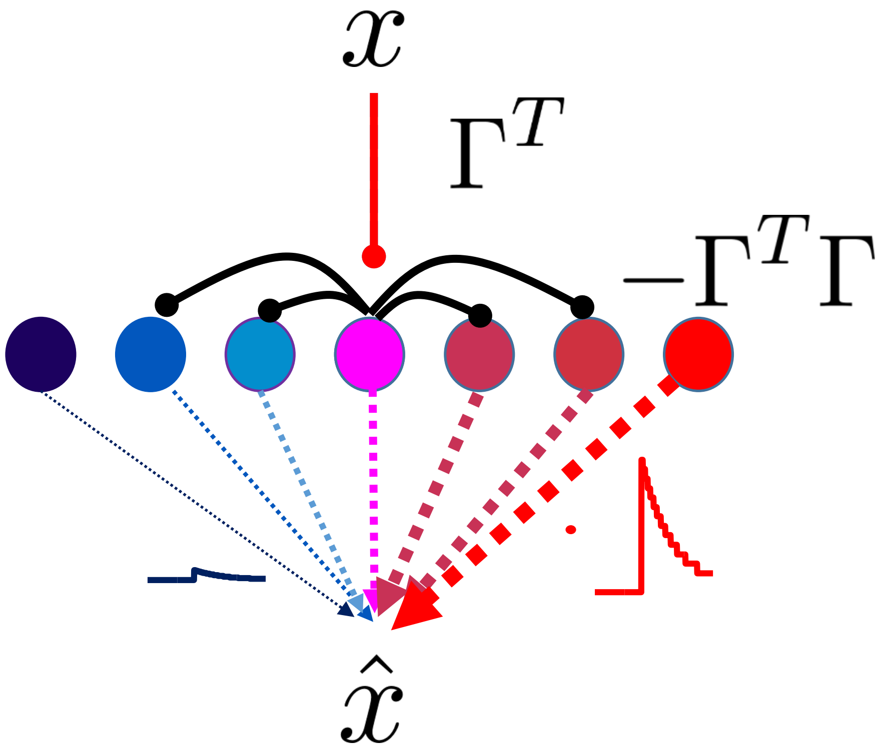
# Adaptation as cost optimization

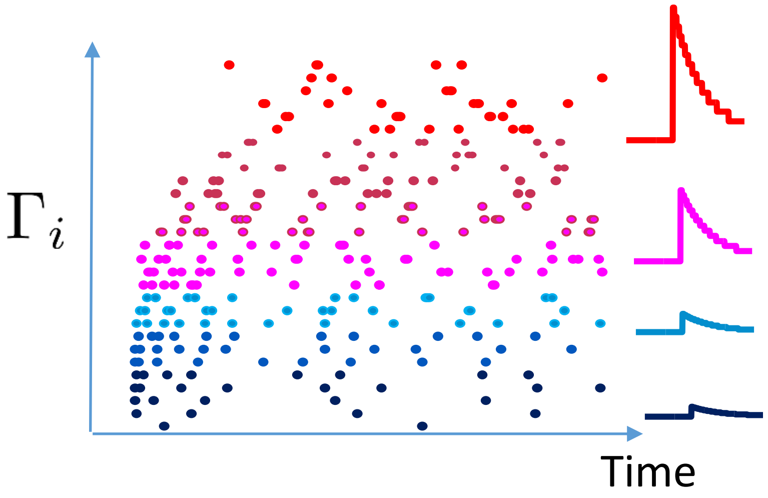
Gabrielle Gutierrez

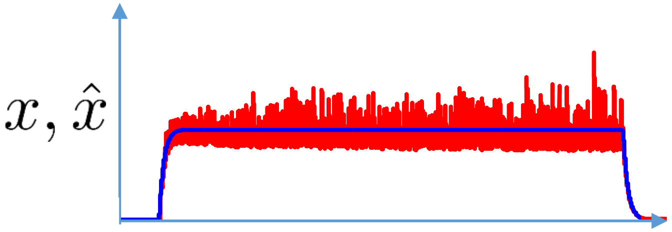


Gabrielle Gutierrez

# Adaptation as cost optimization

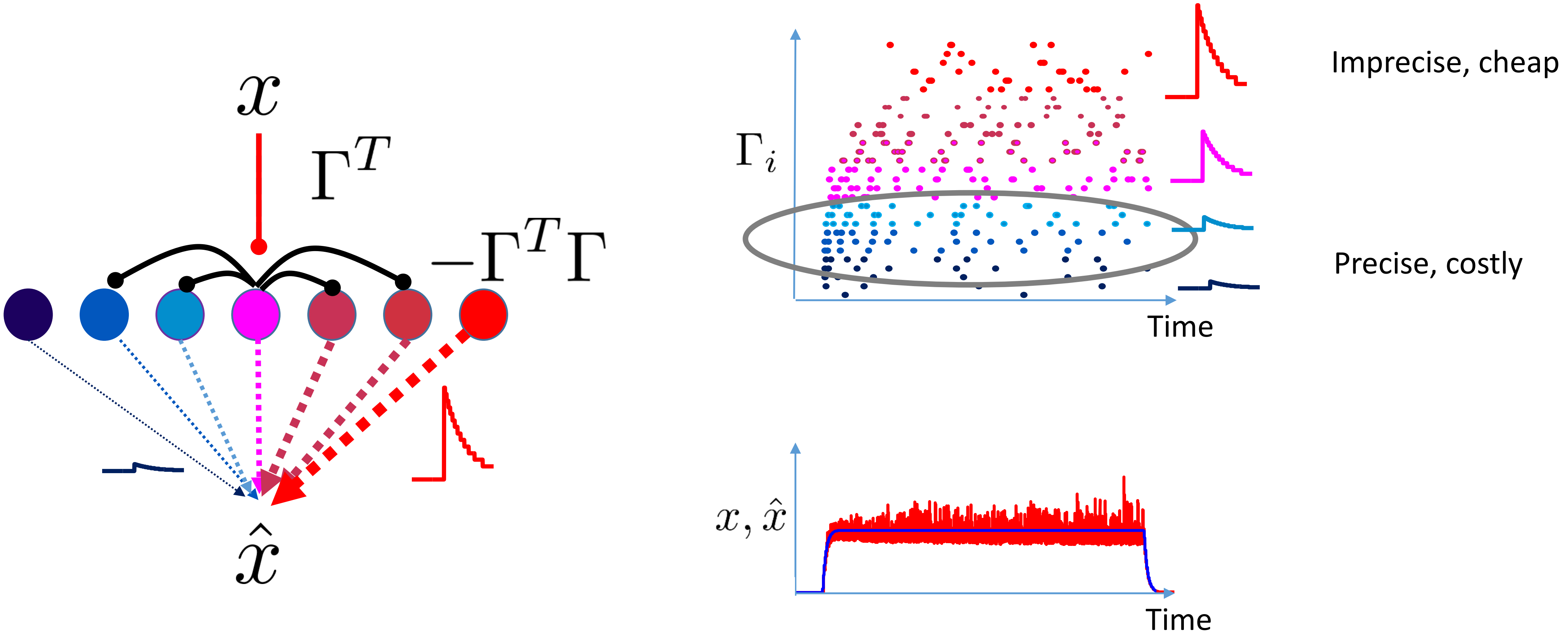


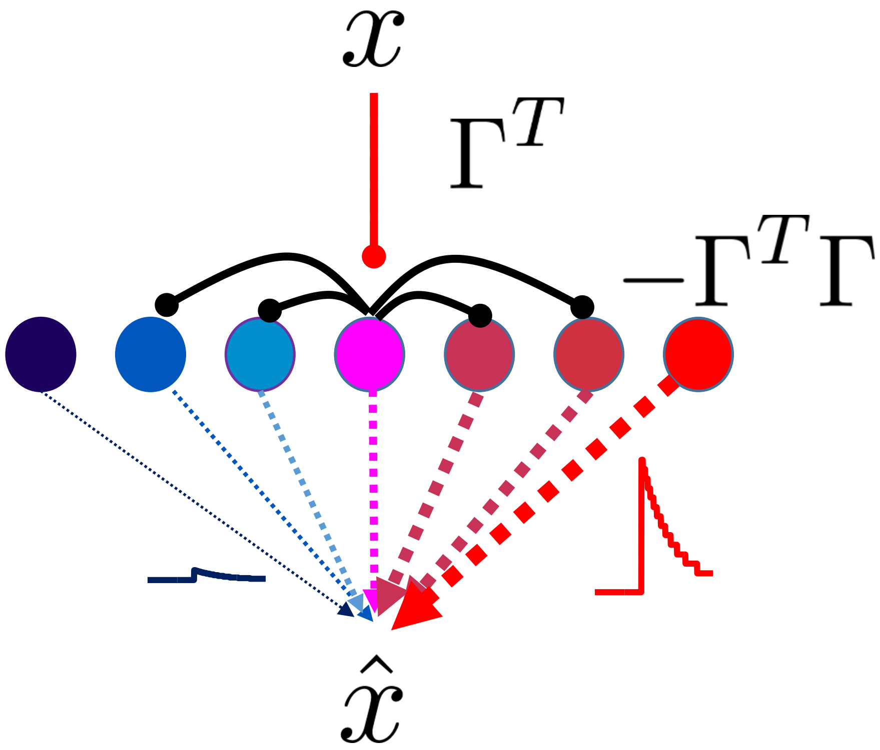
# Adaptation as cost optimization

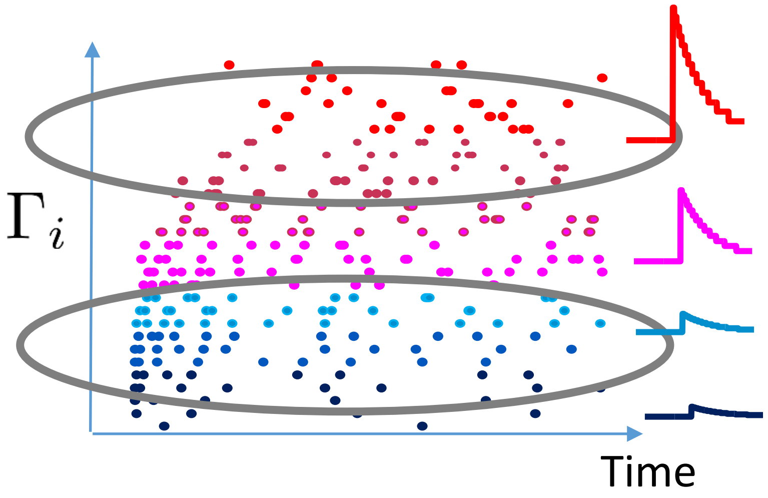
Gabrielle Gutierrez

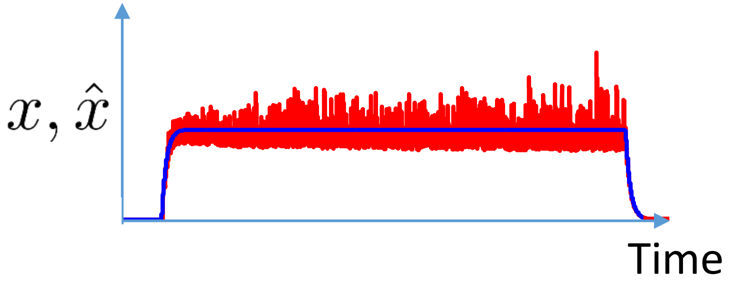


# Adaptation as cost optimization

Gabrielle Gutierrez

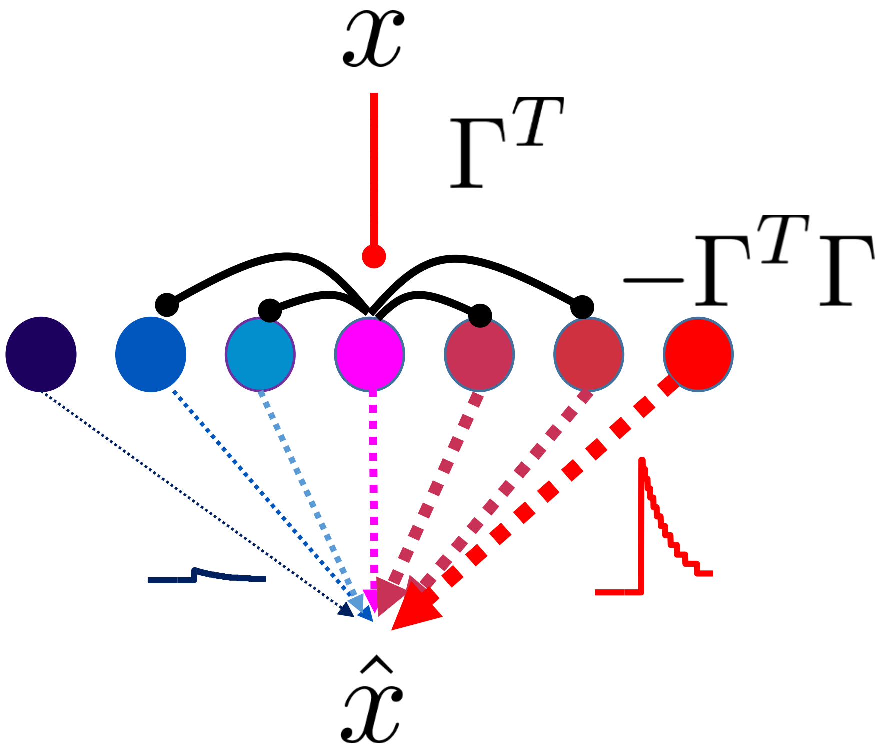


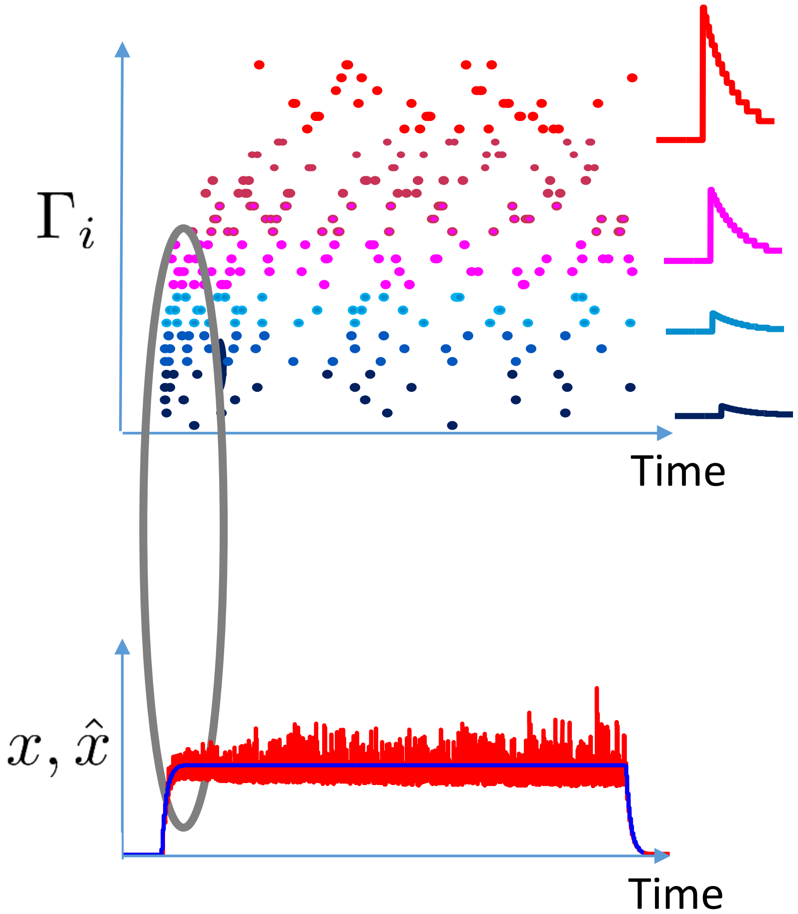



Gabrielle Gutierrez

# Adaptation as cost optimization



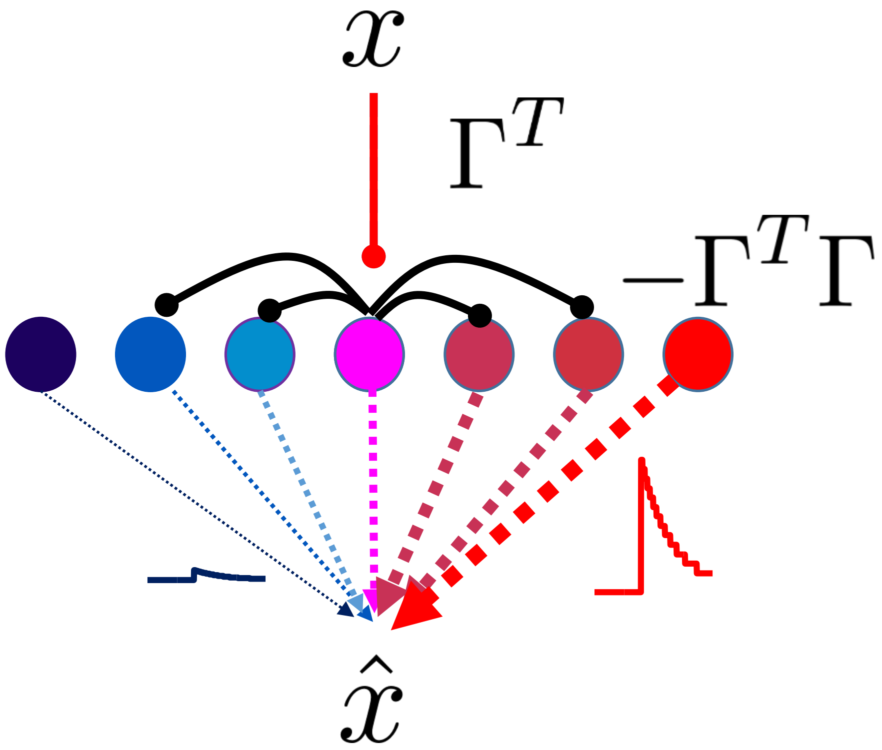



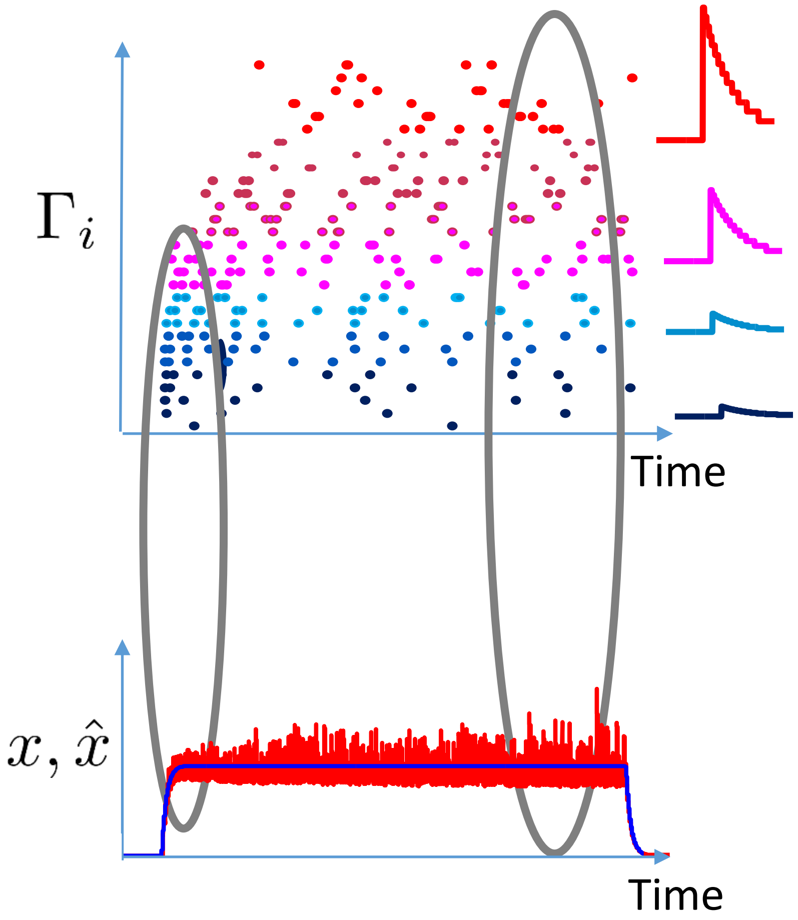

Imprecise, cheap

Precise, costly

Gabrielle Gutierrez

# Adaptation as cost optimization



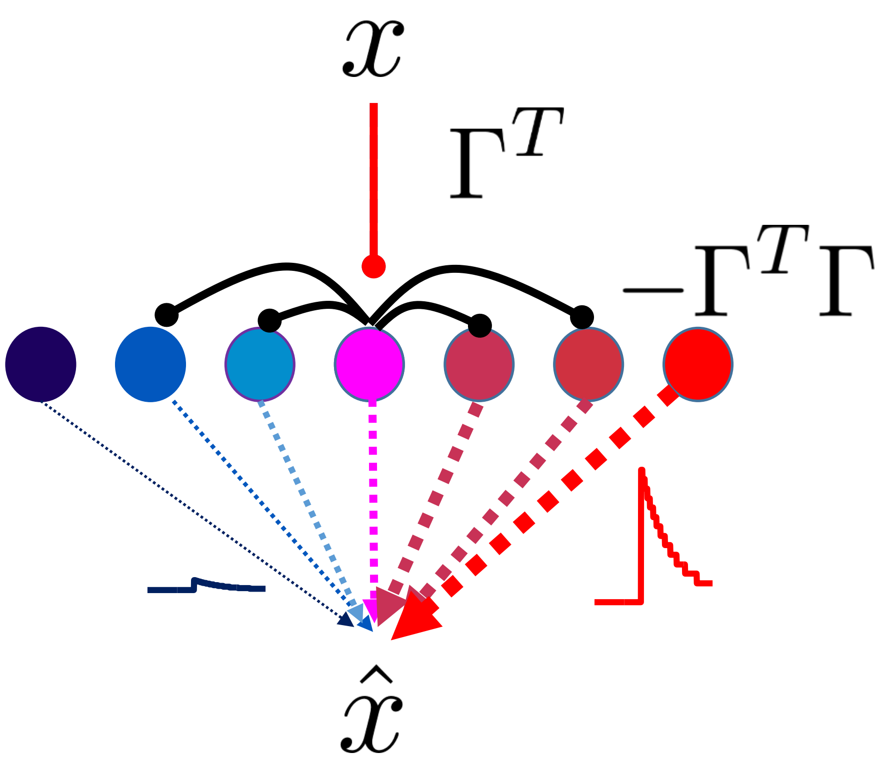



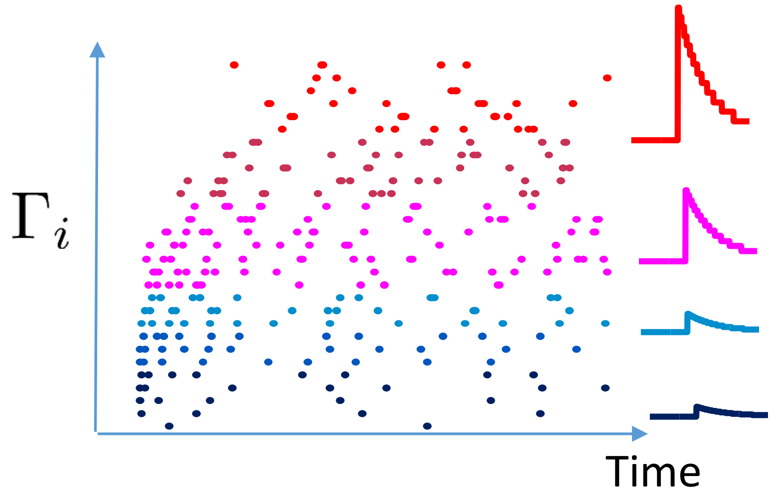

Imprecise, cheap

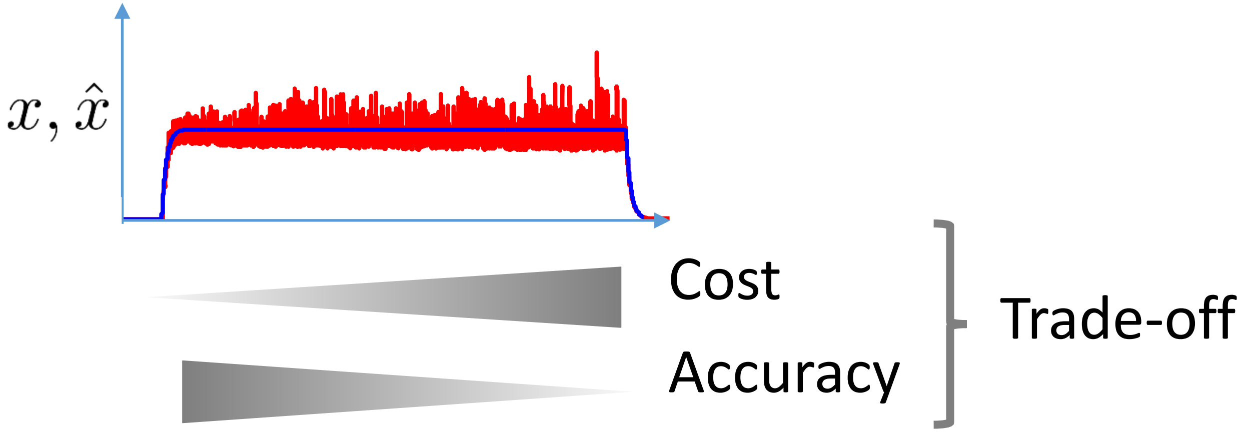
Precise, costly

# Cost, adaptation and homeostasis

Gabrielle Gutierrez

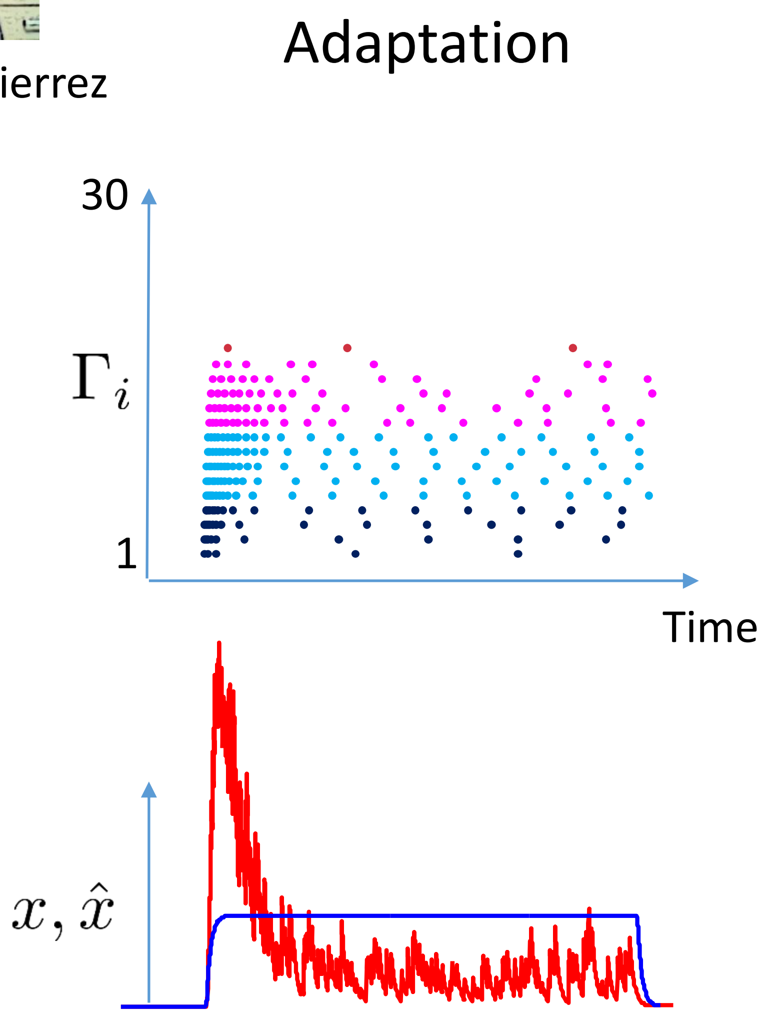


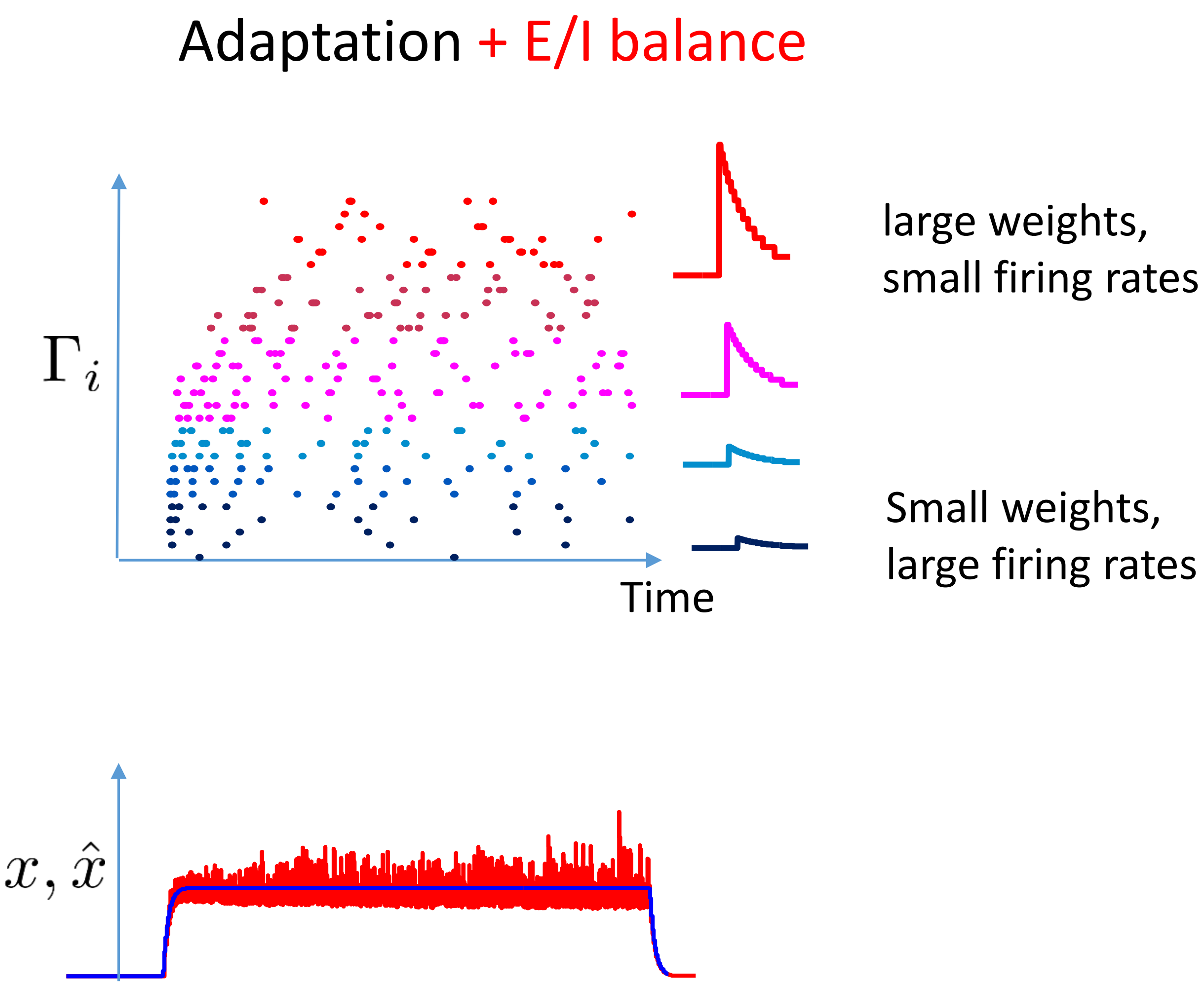



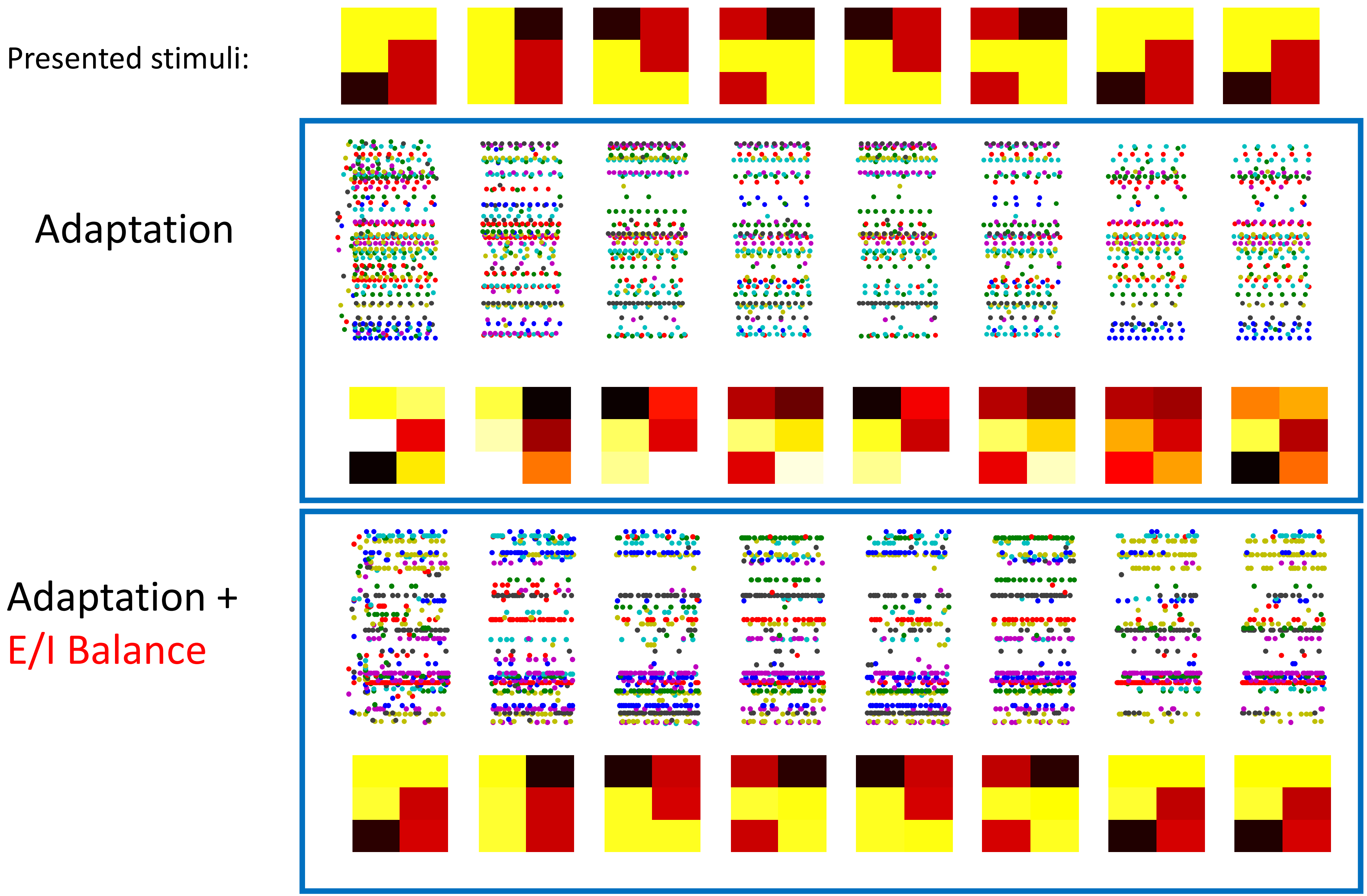



# Cost, adaptation and homeostasis

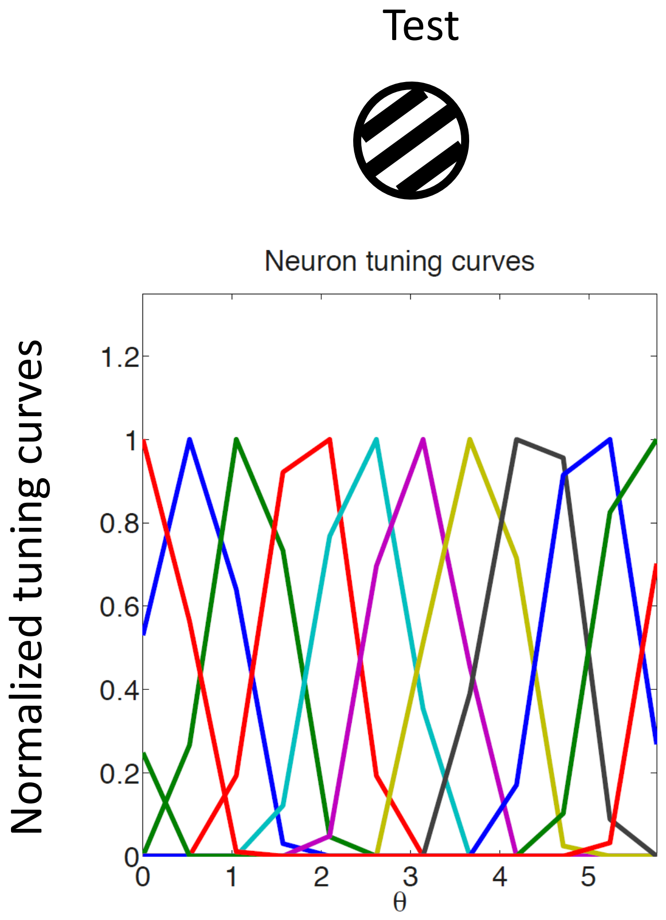
Gabrielle Gutierrez

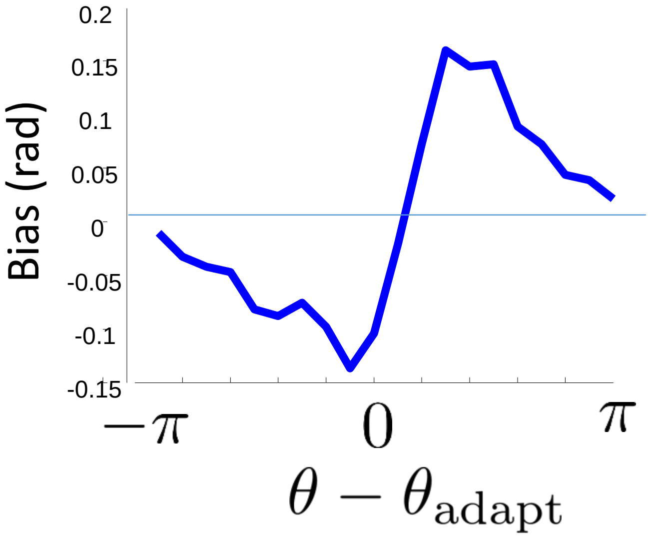


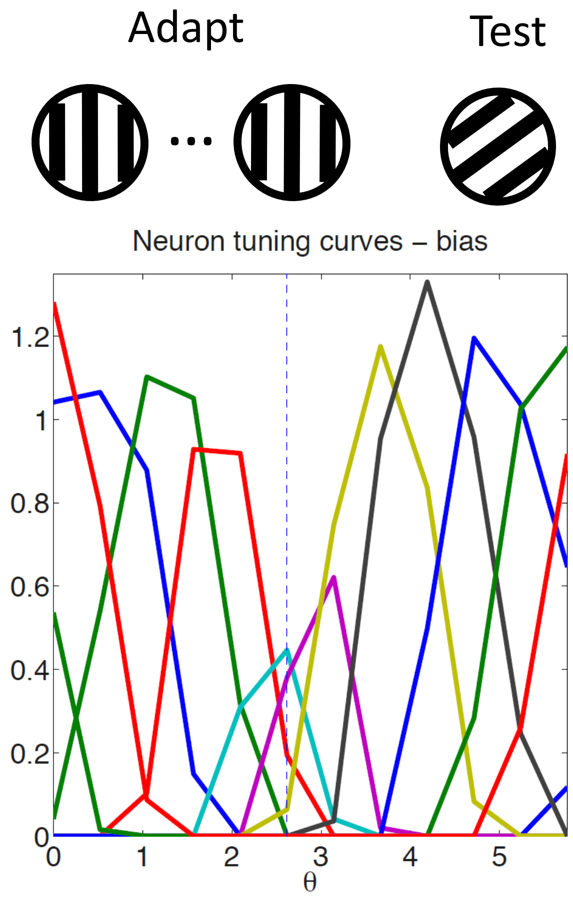




Presented stimuli:
Adaptation
Adaptation +
E/I Balance

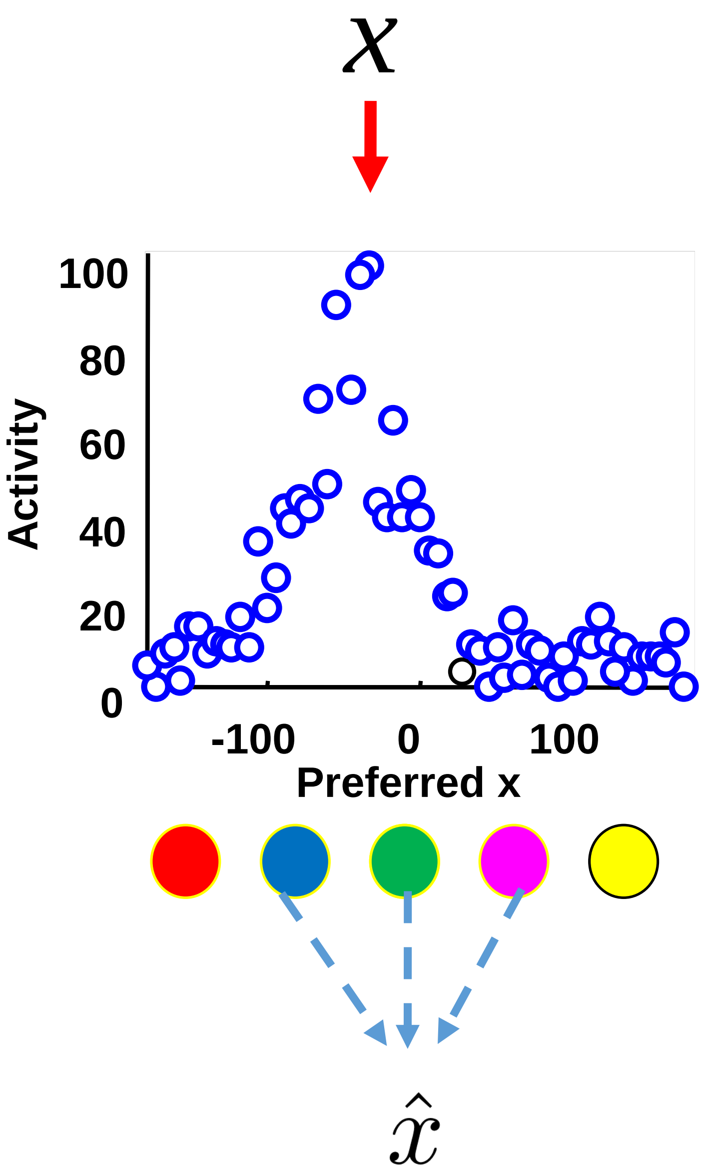
# Orientation Adaptation (tilt illusion)



# Population coding




Many unreliable neurons…

# Population coding

Many unreliable neurons…