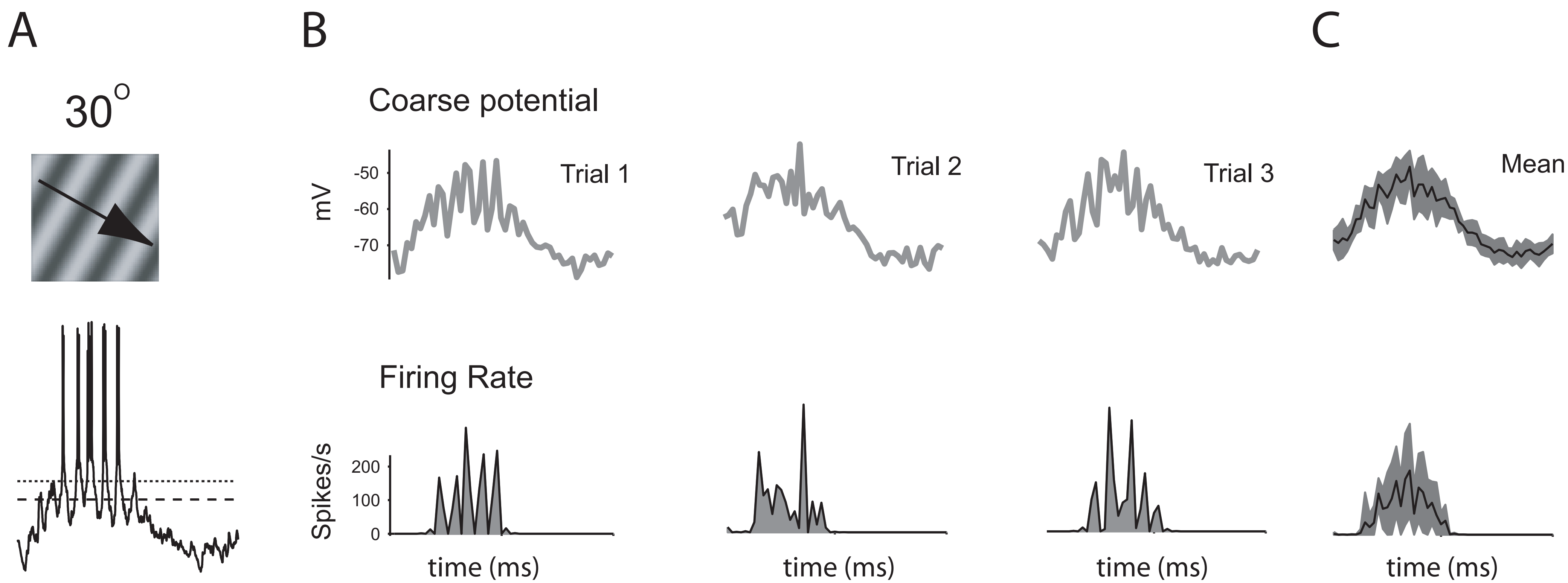
Covariability in Coupled Neural Systems

Cyrus Omar
Advisor: Brent Doiron
Program in Neural Computation
First Year Project
Aug. 22, 2009

Outline

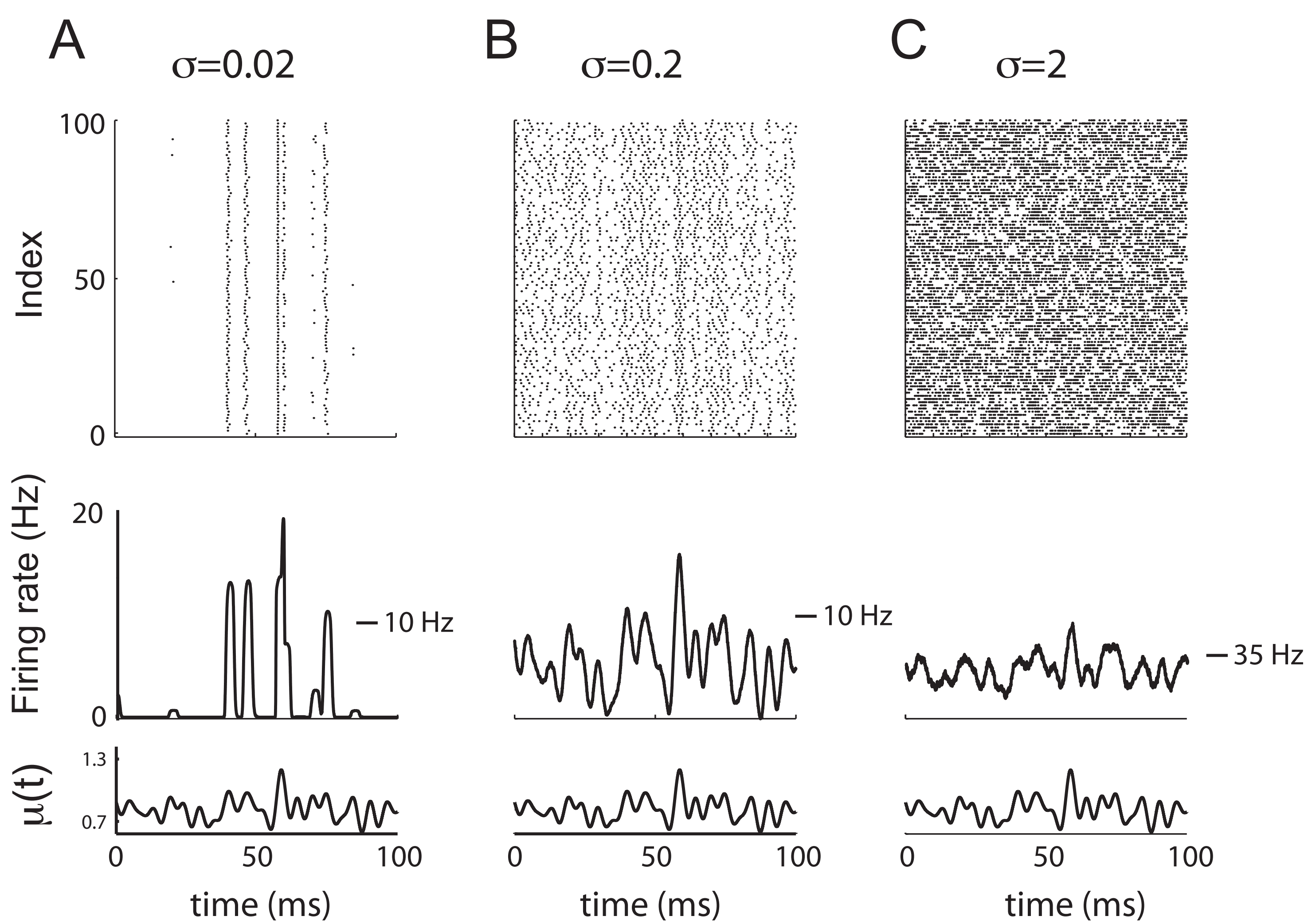
- Background: variability and covariability
    - What is it? Why is it important?
- The effect of coupling on covariability
    - A simple model of coupled units, explaining some curious somatosensory data
    - A first glance at a fully coupled, realistic model
        - Efficient implementation and analysis of these models

Neural responses are variable



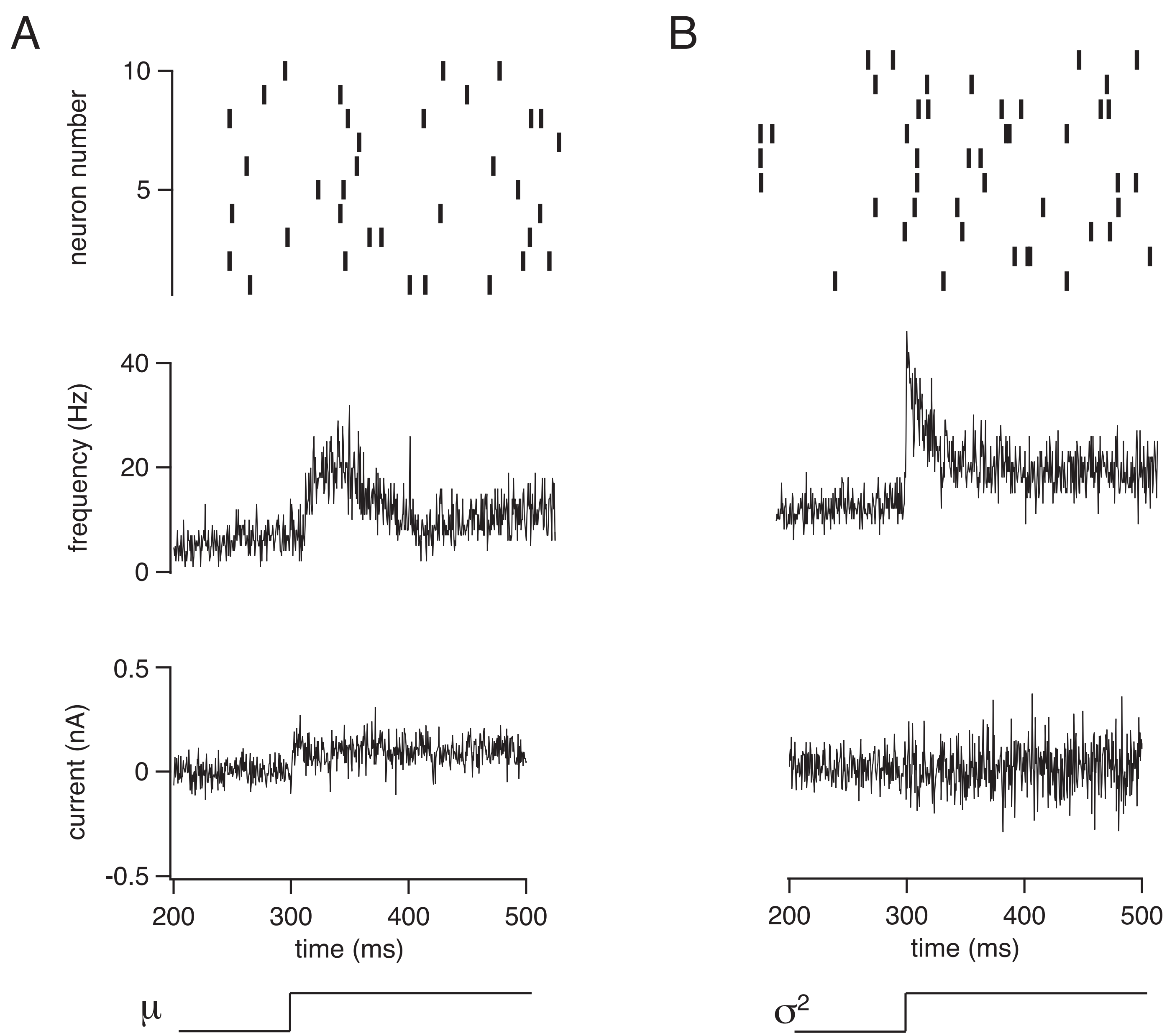
Carrandini, 2004

The effect of variability on coding



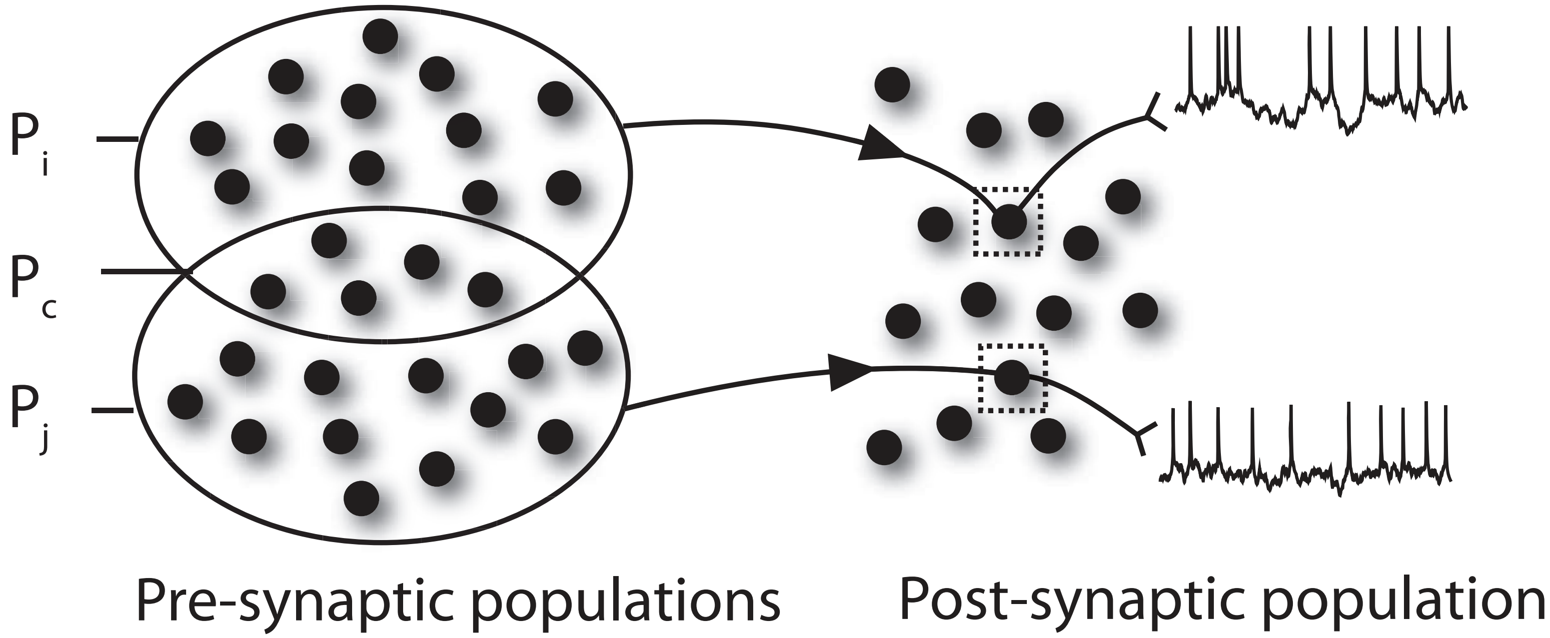
Motivated by Stein (1967) and Knight (1972). Figure from Doiron (2009).

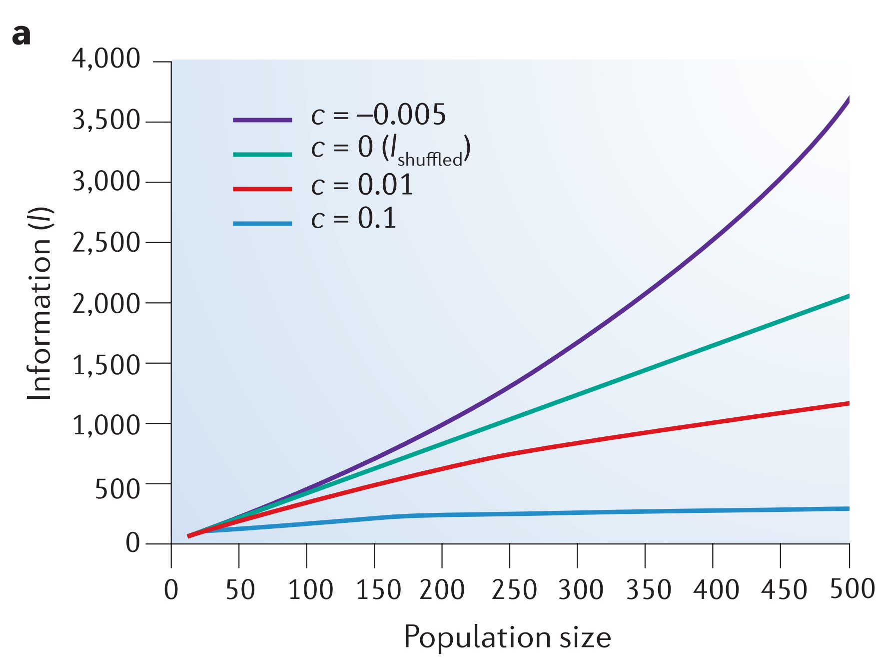
Using variability for coding

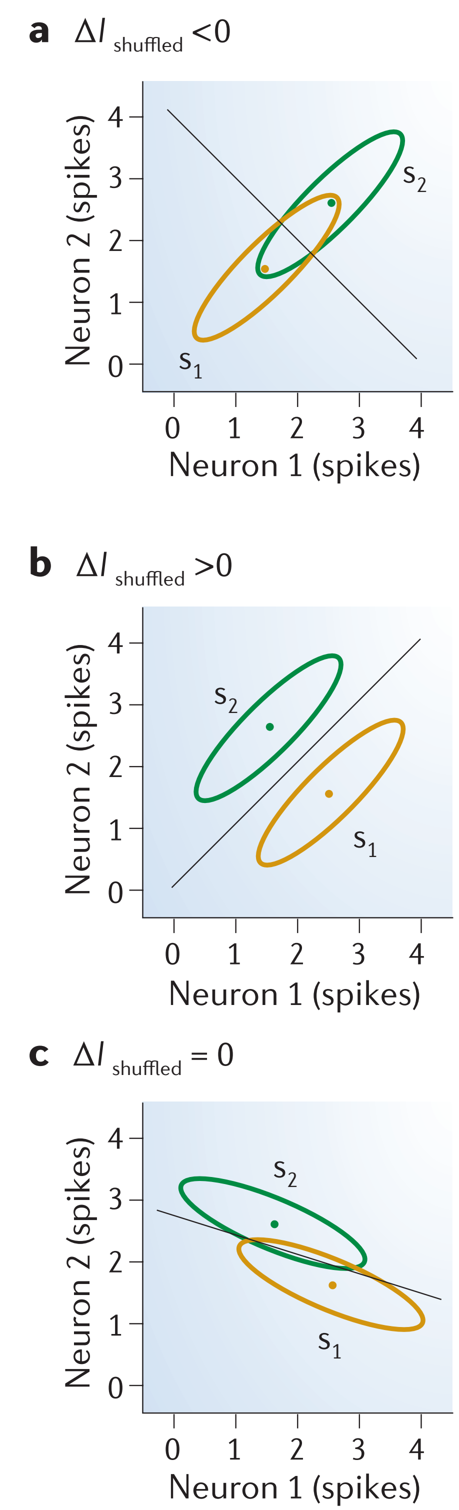


Silberberg et al, 2004

Neural **co**-variability is important

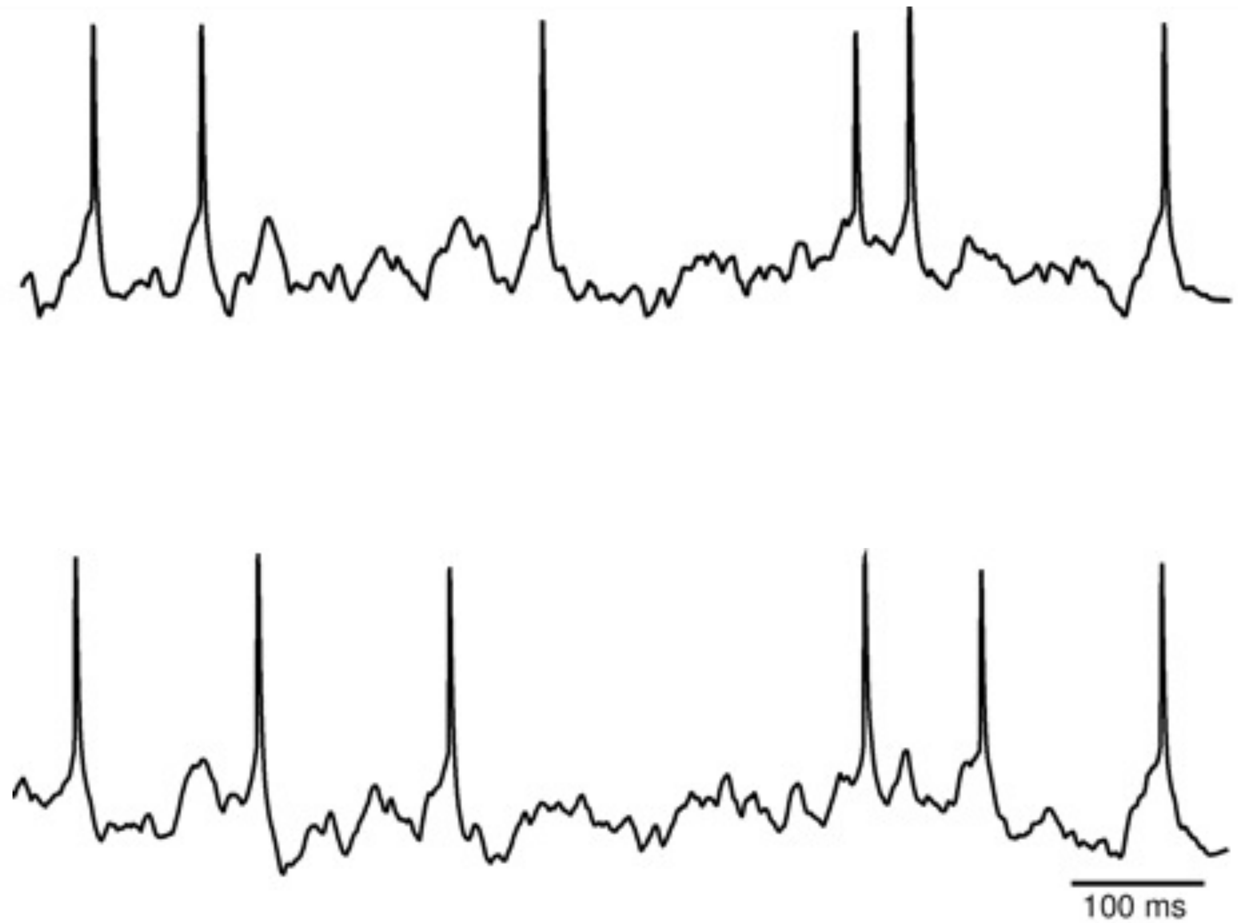




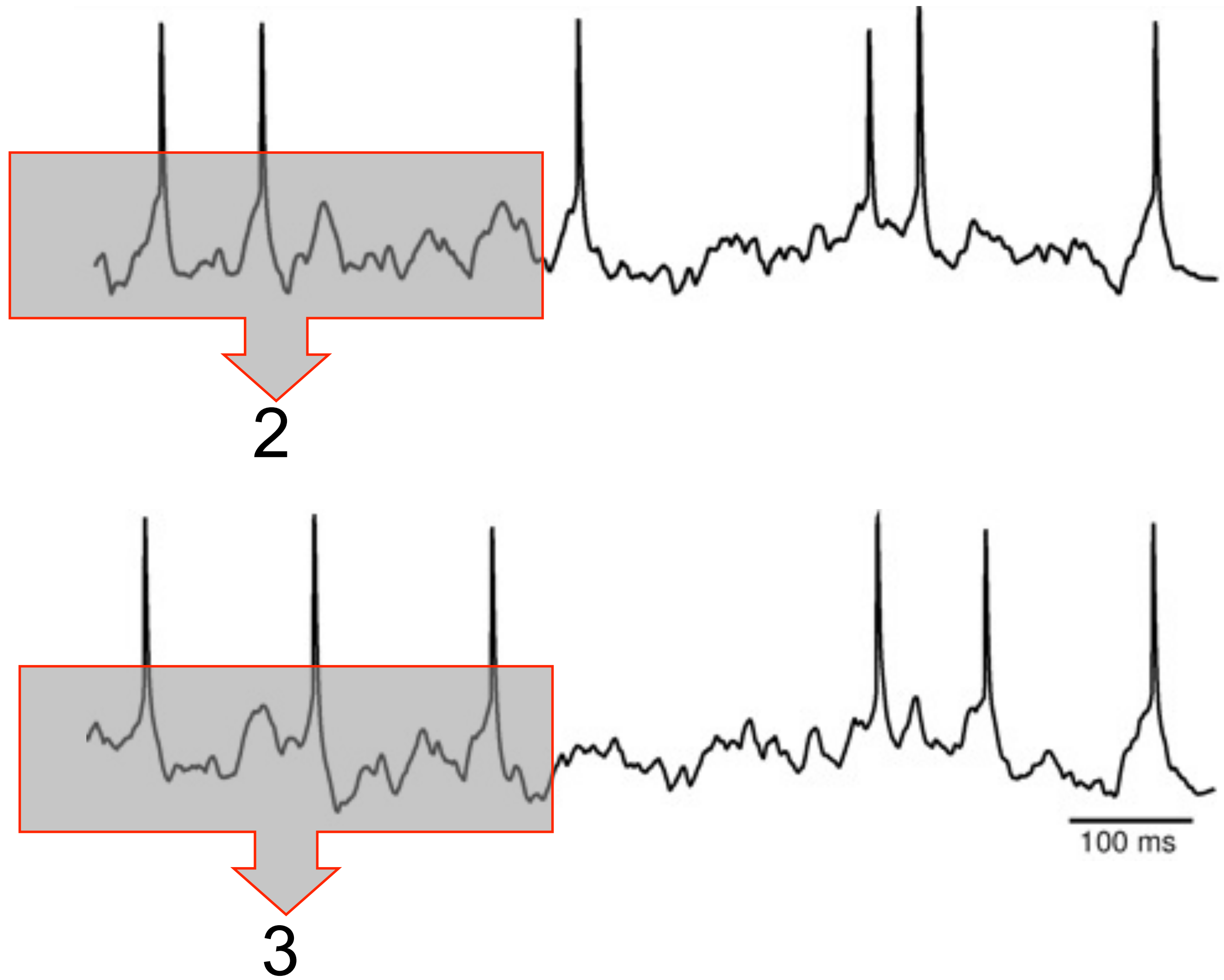


Averbeck et al, 2006

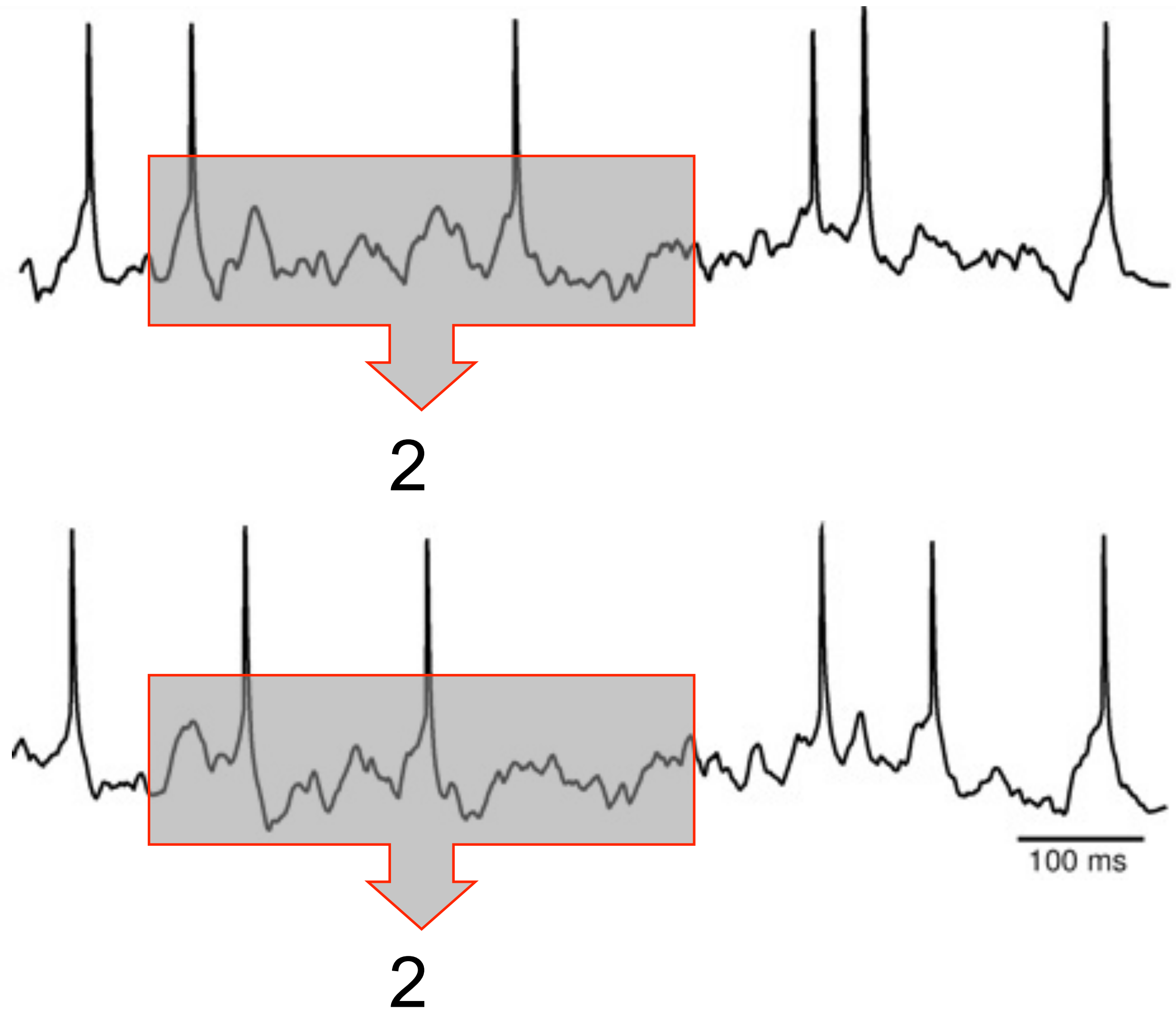
Calculating Spike Count Correlation



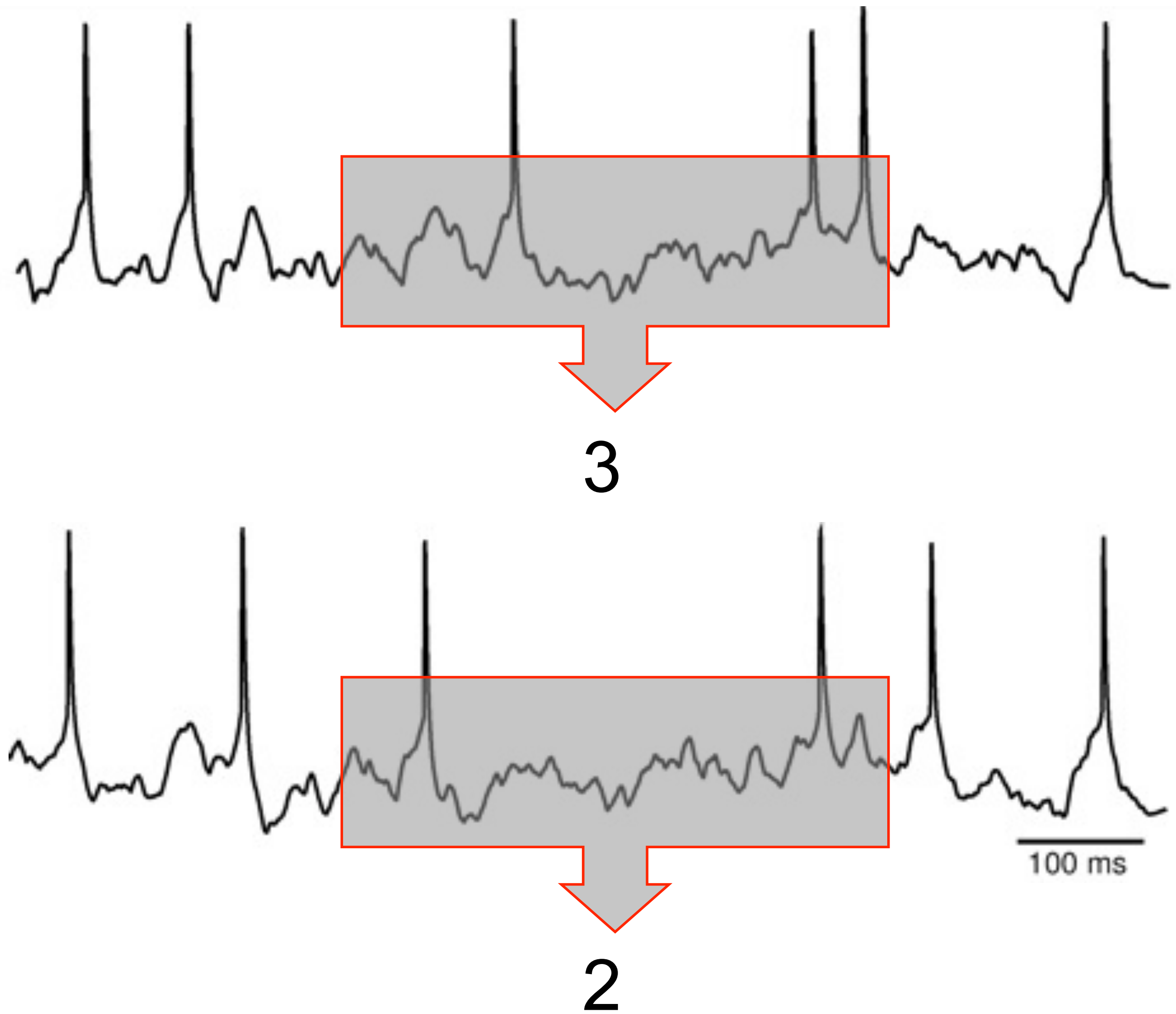
Calculating Spike Count Correlation



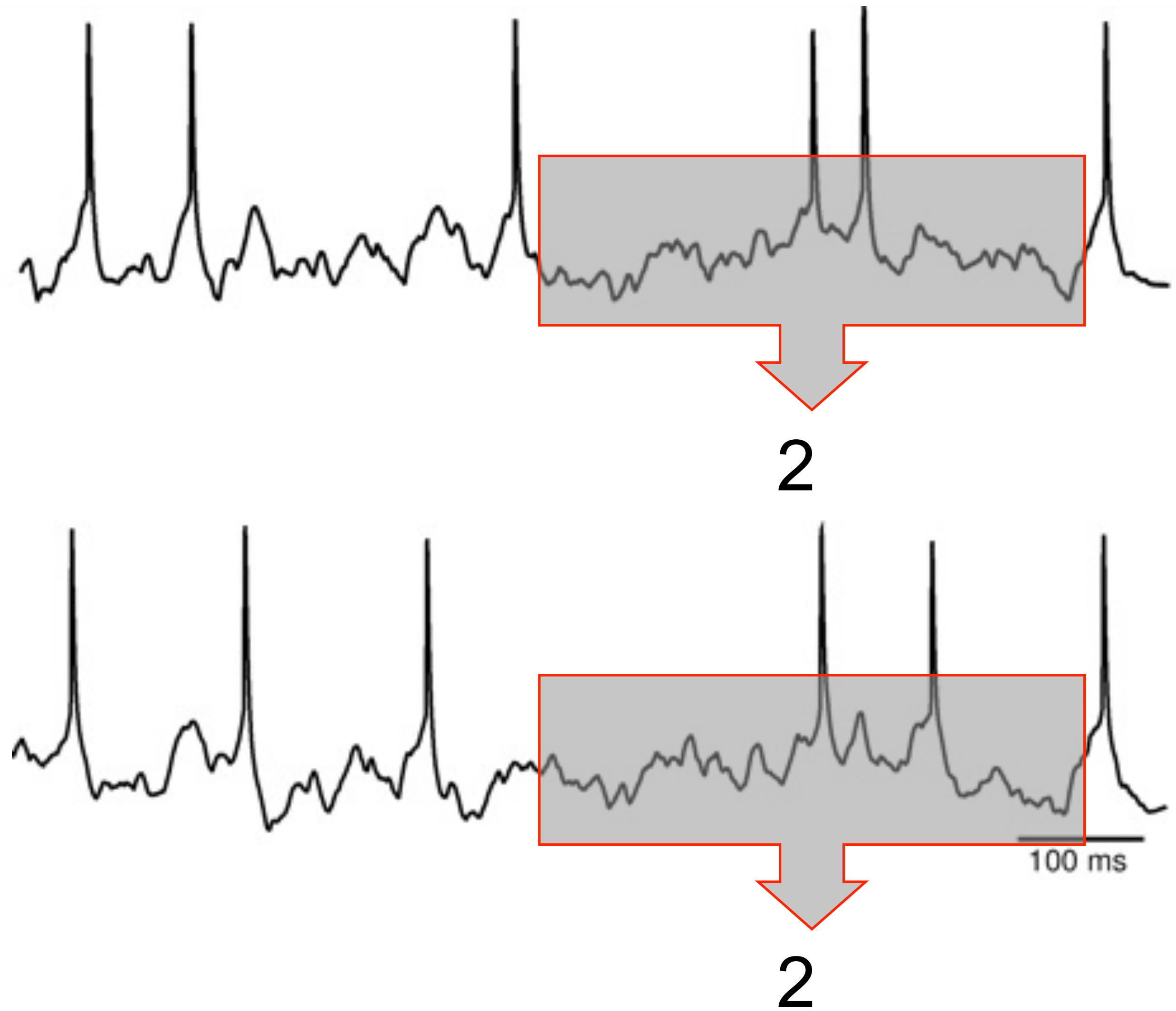
Calculating Spike Count Correlation



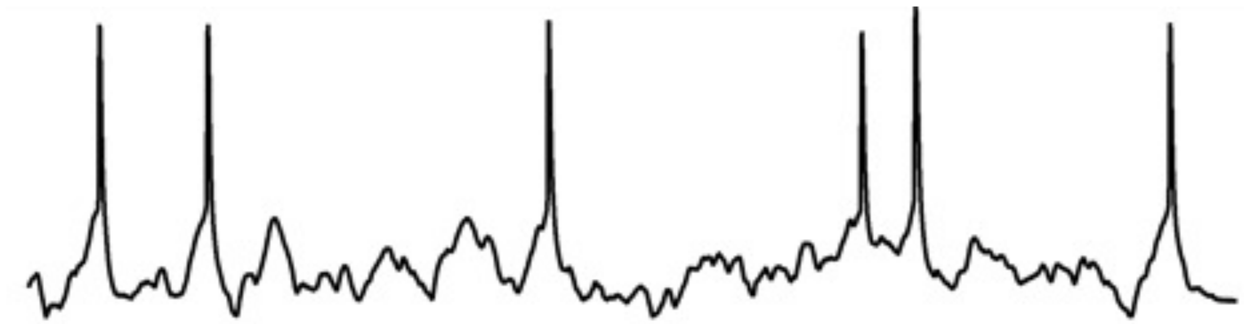
Calculating Spike Count Correlation



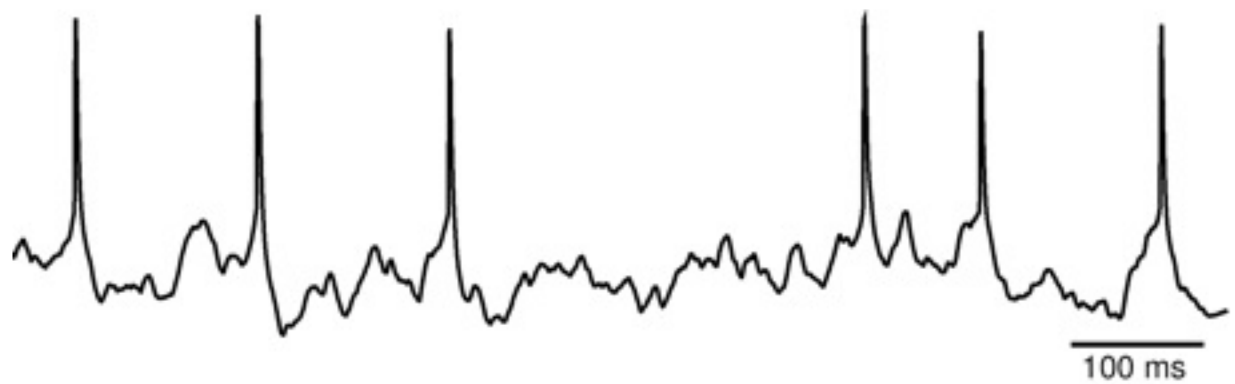
Calculating Spike Count Correlation



Calculating Spike Count Correlation



n_1 = … 2, 2, 3, 2, …



n_2 = … 3, 2, 2, 2, …

Calculating Spike Count Correlation

$$Cov_{12,T} = E_T(n_1 n_2) - E_T(n_1)E_T(n_2)$$

$$\rho_T = \frac{Cov_{12,T}}{\sqrt{Var_{1,T}Var_{2,T}}} \qquad \rho = \lim_{T \to \infty} \rho_T$$

Correlation coefficient

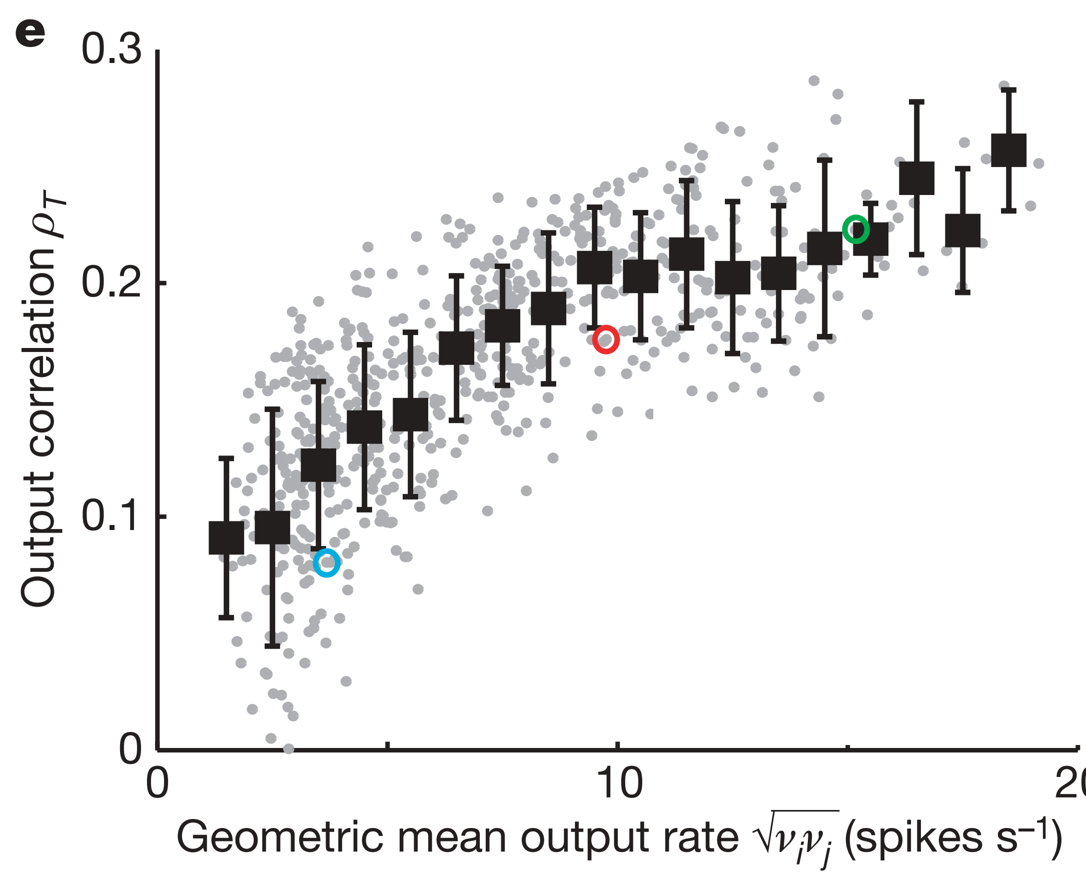
$\rho = -1$ - completely anti-correlated

$\rho = 0$ - uncorrelated

$\rho = 1$ - completely correlated

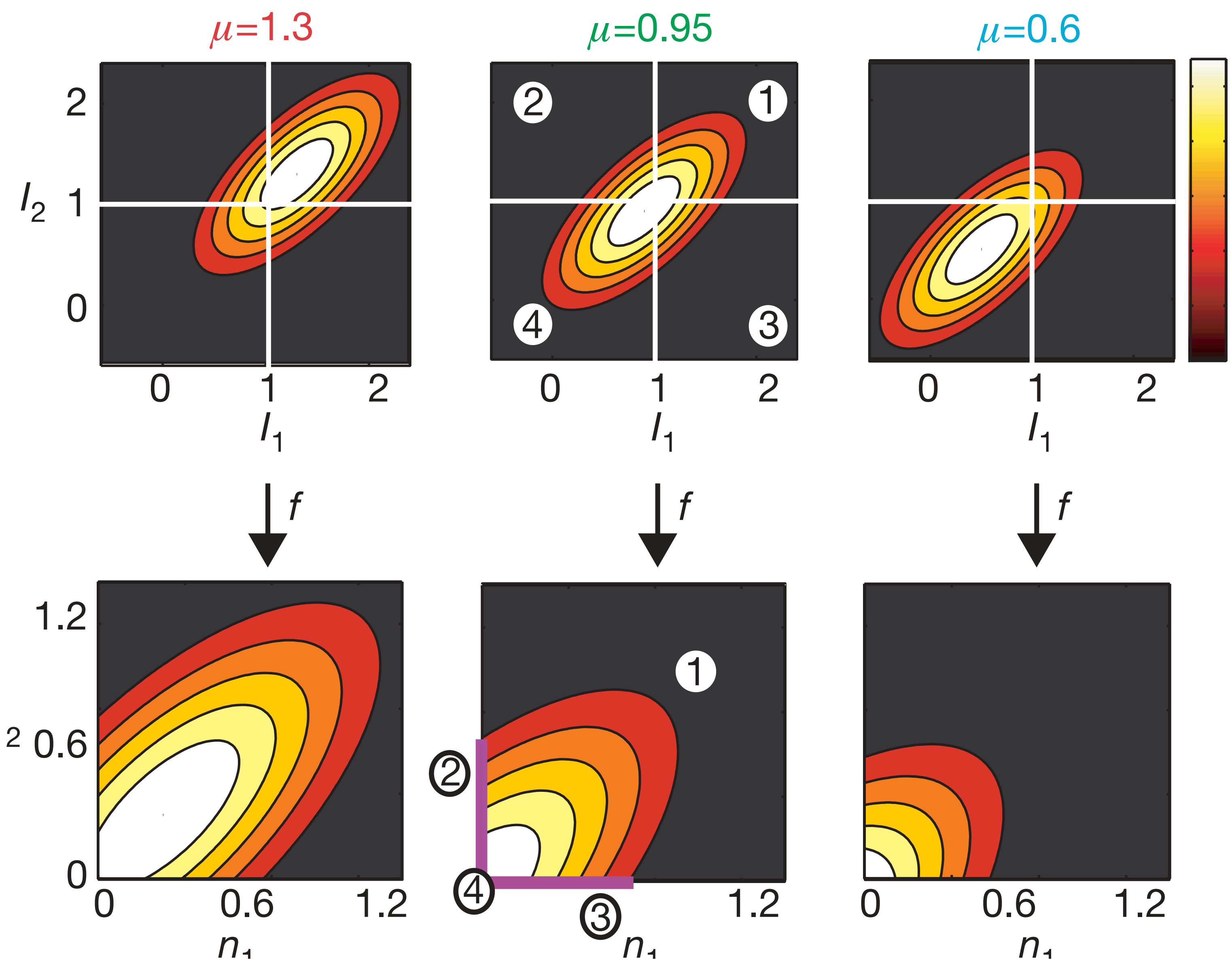
Predicting Output Correlation Given Input Statistics

A correlation rate relationship?



de la Rocha, Doiron, et al, 2007

Why?

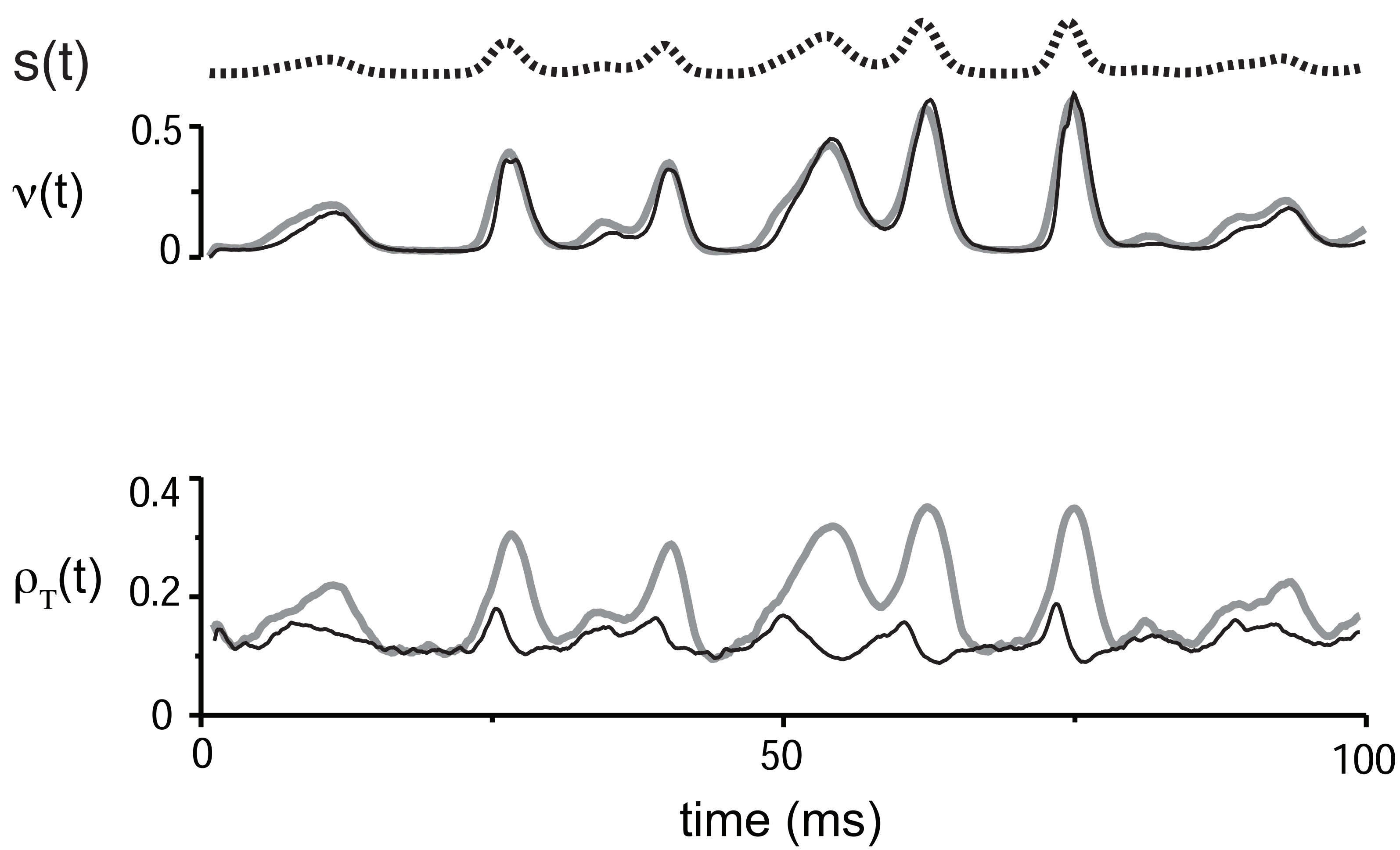


oval → circle → point

de la Rocha, Doiron, et al, 2007

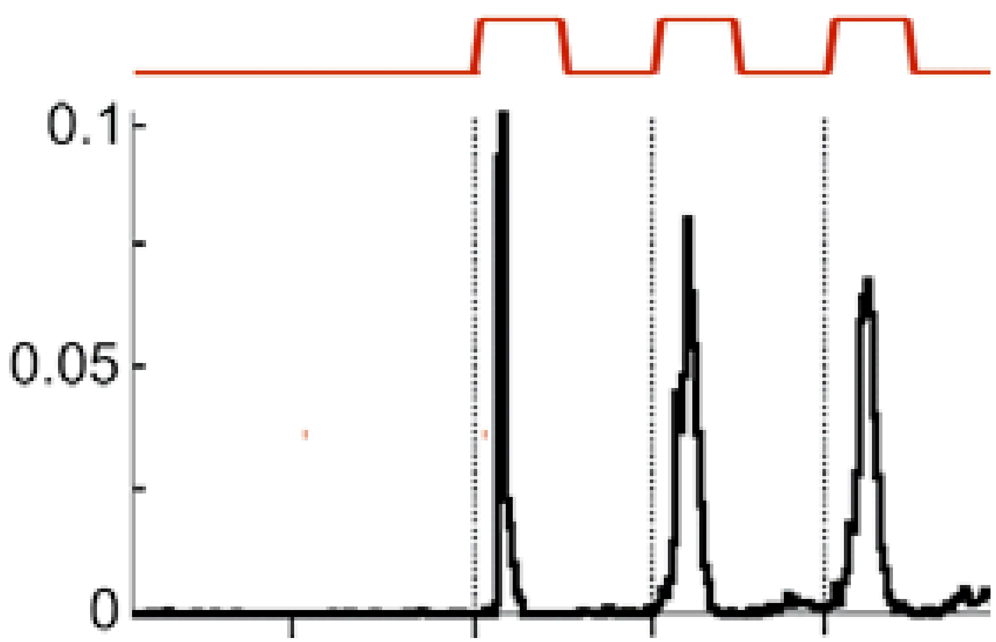
Predicting Output Correlation Given Input Statistics

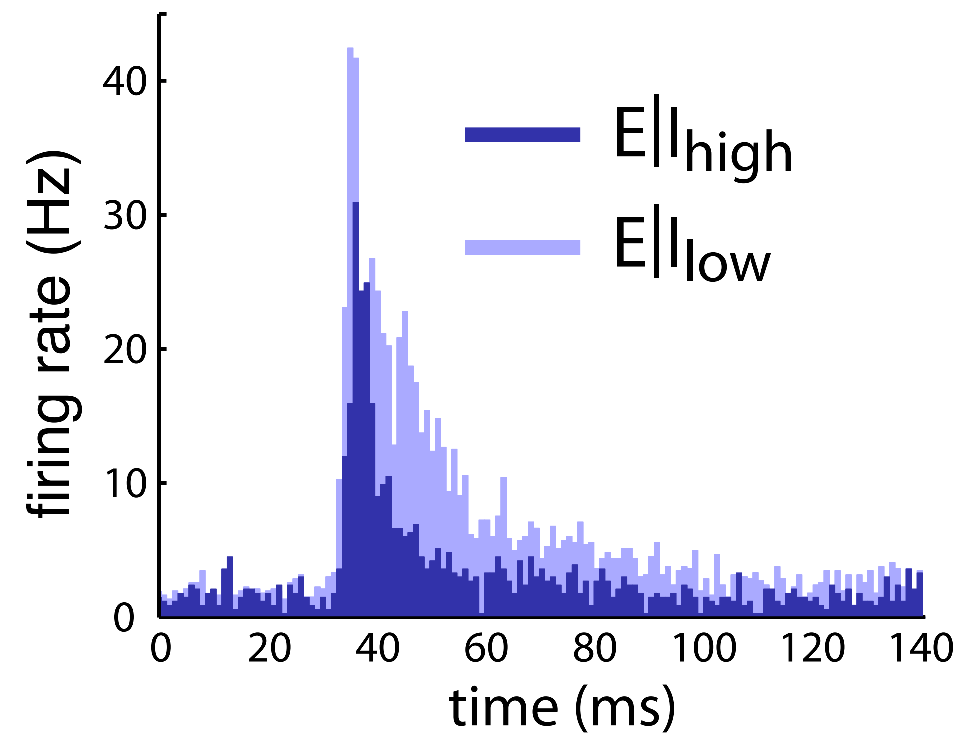
A correlation rate relationship?

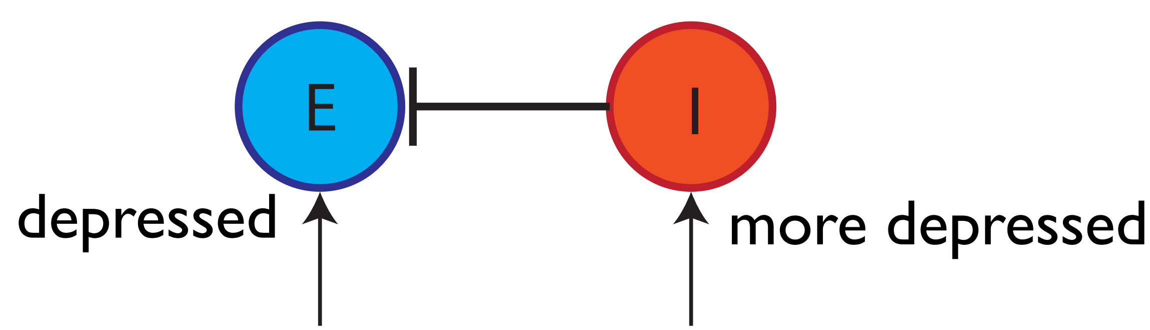


Shea-Brown, et al, 2008

E-I correlations

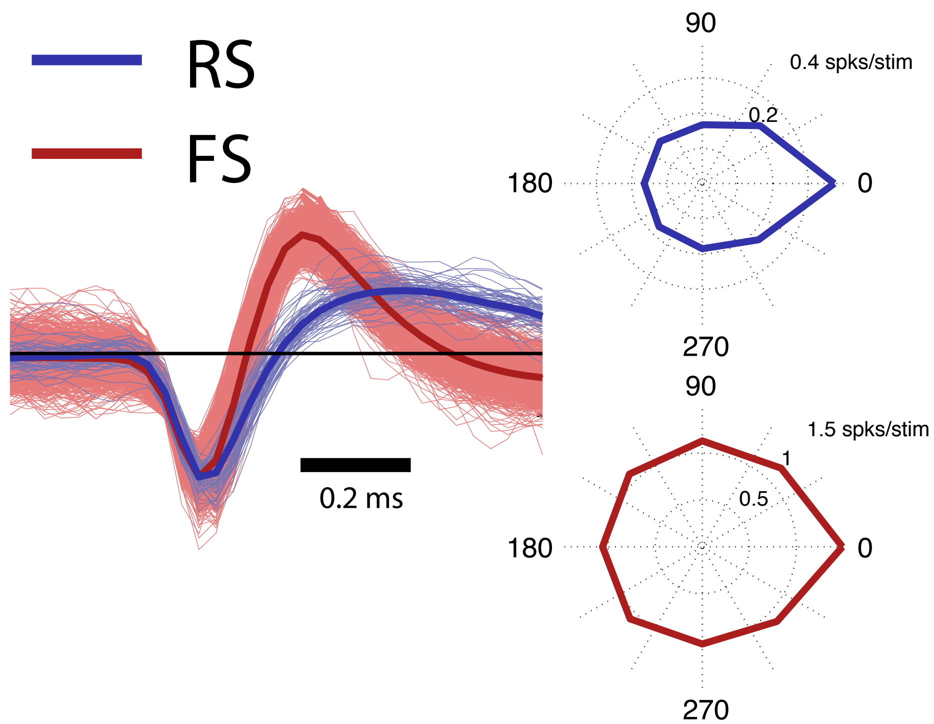


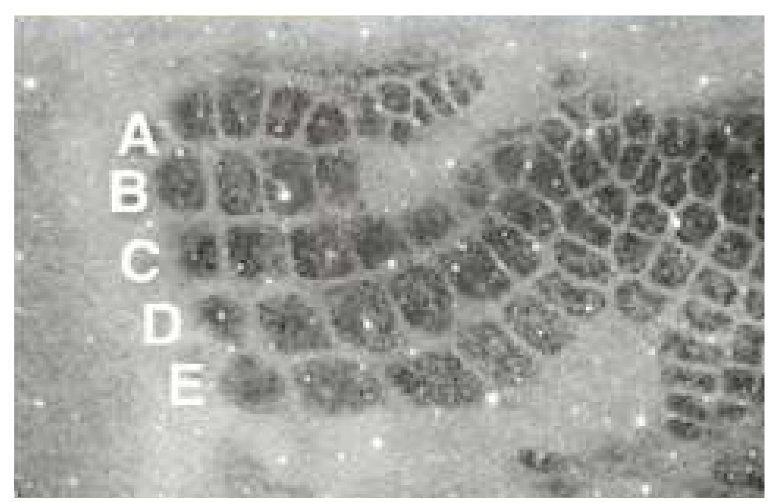




Gabernet, Jadhav, Feldman, Carandini, Scanziani, 2005

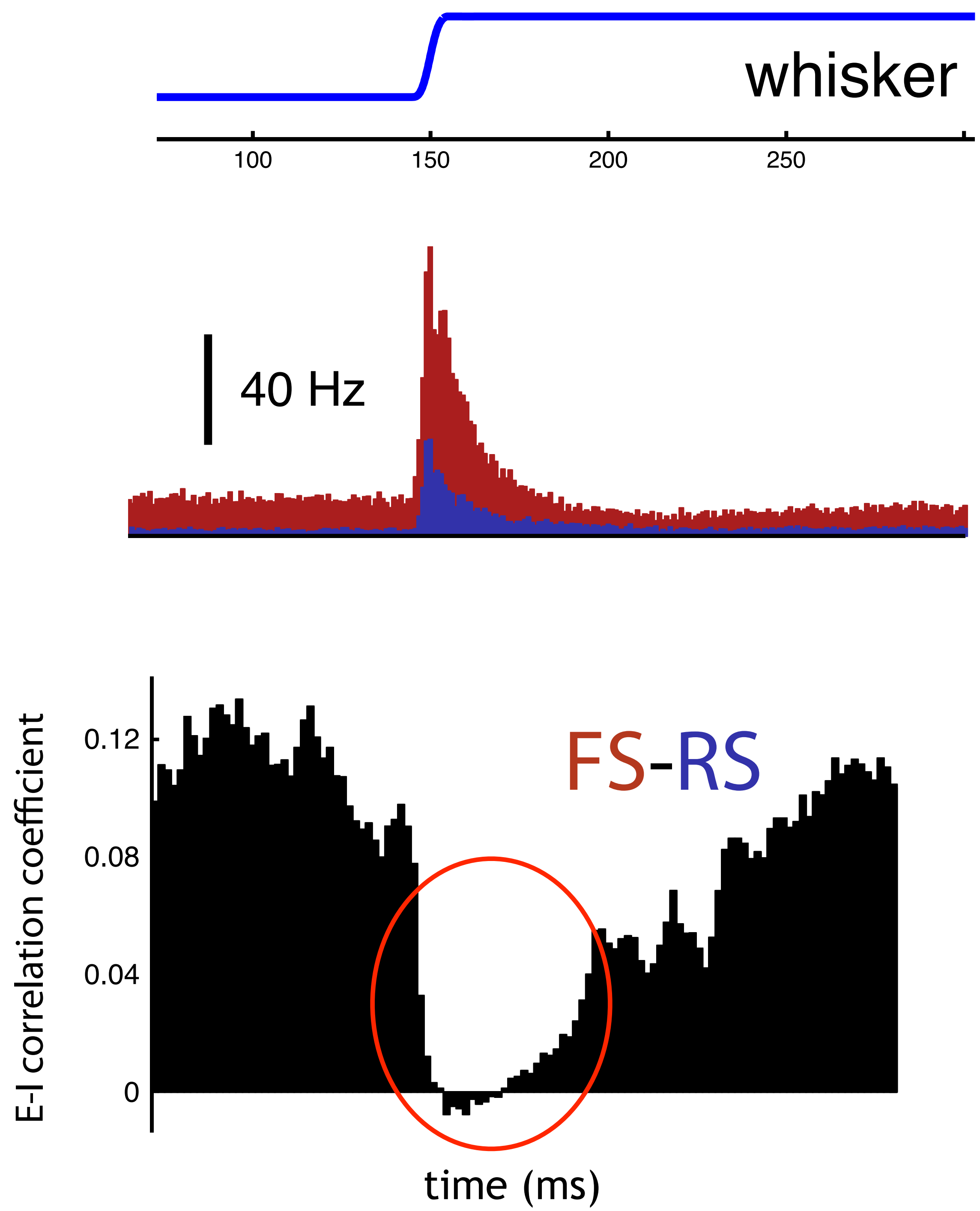
Measuring E-I correlation *in vivo*





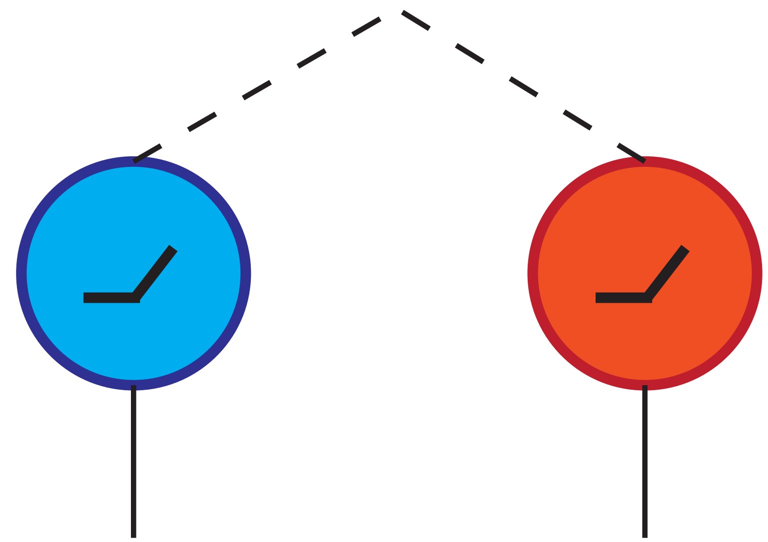
Middleton et al, in prep

Measuring E-I correlation *in vivo*

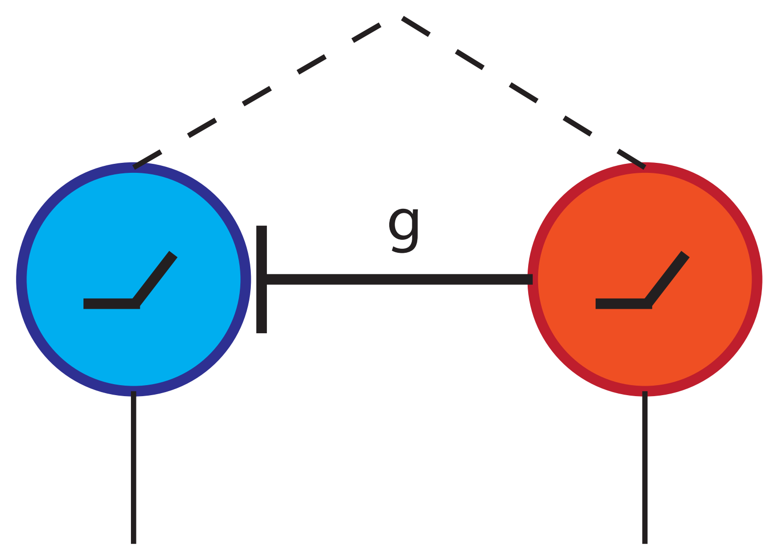


Middleton et al, in prep

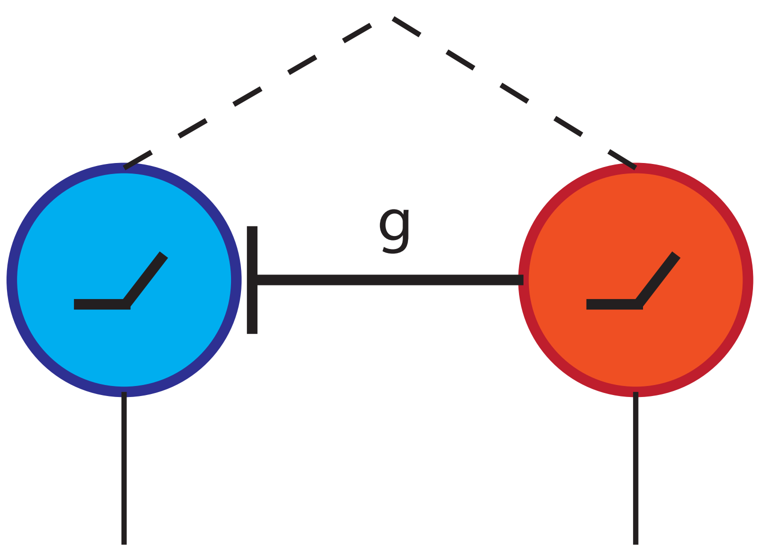
What's the problem?



What's the problem?

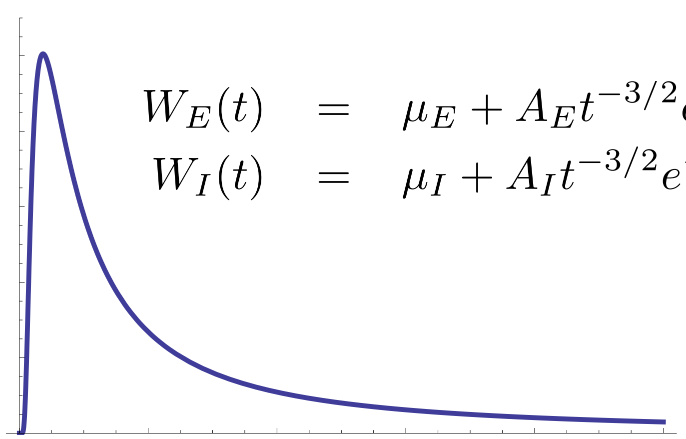


A Coupled Model



$$\begin{aligned} r_I(t) &= \max\{0, W_I(t) + \eta_I(t)\} \\ \tau_I \frac{d\eta_I}{dt} &= -\eta_I(t) + \sigma_I \xi_I(t) \\ \tau_s \frac{dI}{dt} &= -I(t) + r_I(t) \end{aligned}$$

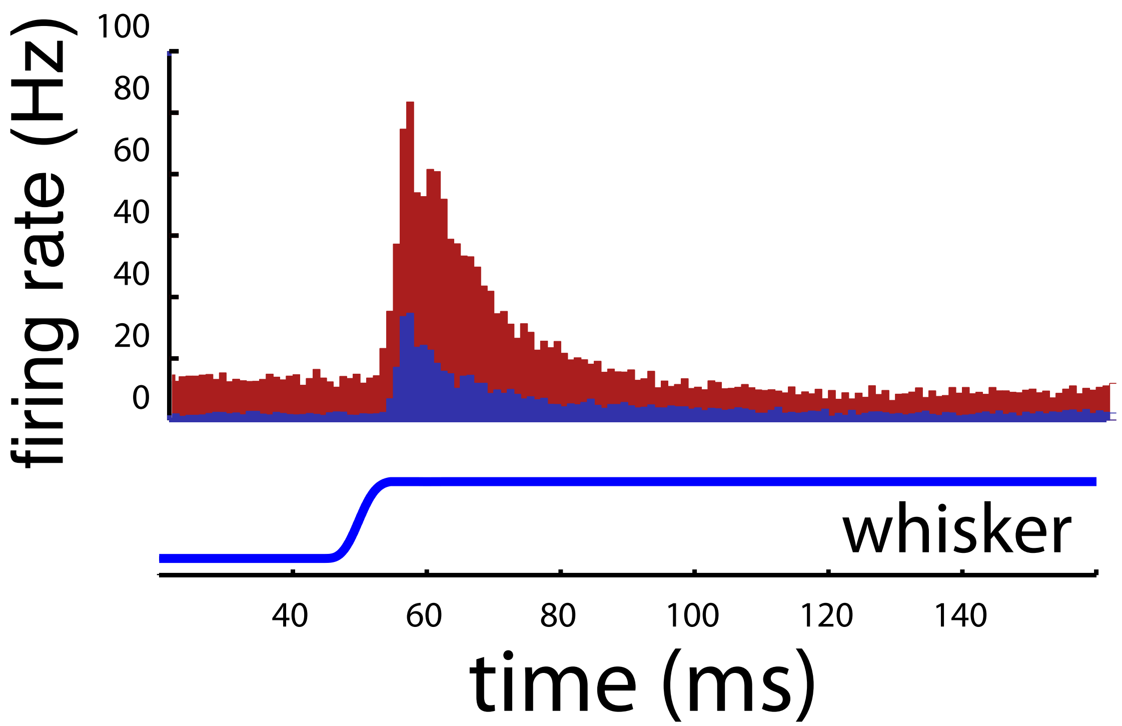
$$\begin{aligned} r_E(t) &= \max\{0, W_E(t) - gI(t) + \eta_E(t)\} \\ \tau_E \frac{d\eta_E}{dt} &= -\eta_E(t) + \sigma_E \xi_E(t) \end{aligned}$$

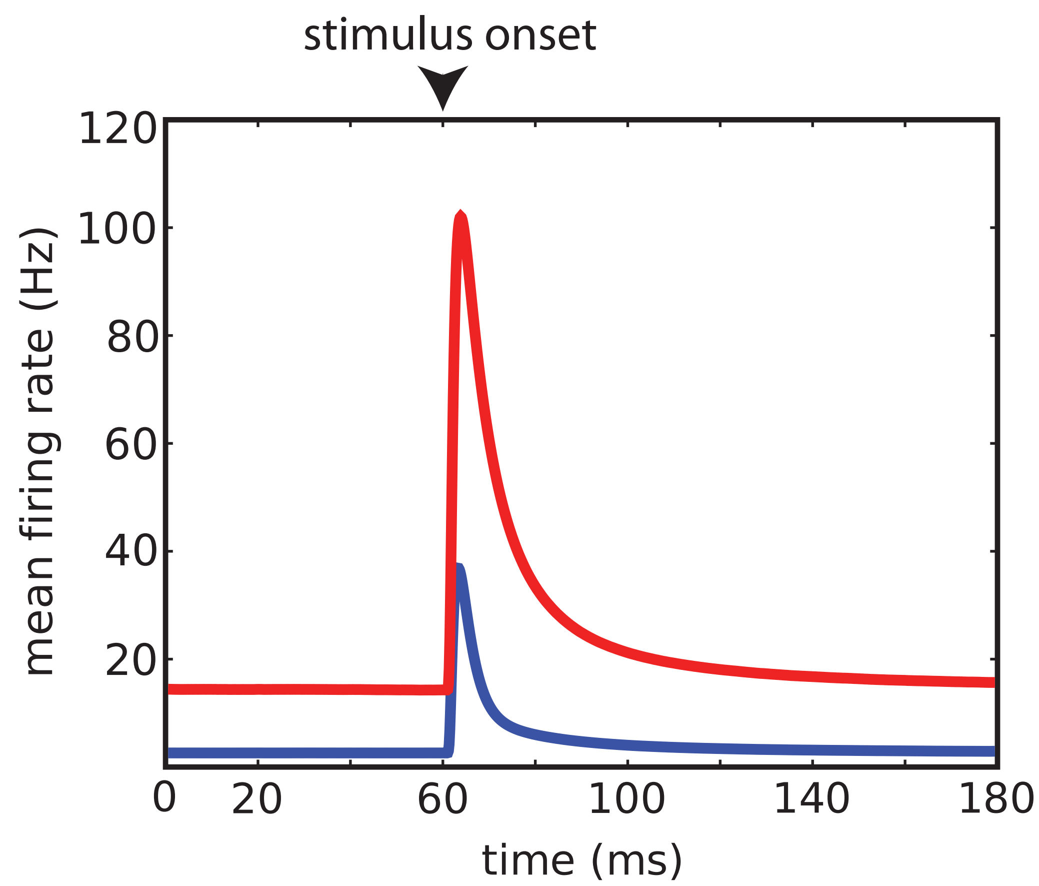


$$\begin{aligned} W_E(t) &= \mu_E + A_E t^{-3/2} e^{-0.55/t} \\ W_I(t) &= \mu_I + A_I t^{-3/2} e^{-0.55/t} \end{aligned}$$

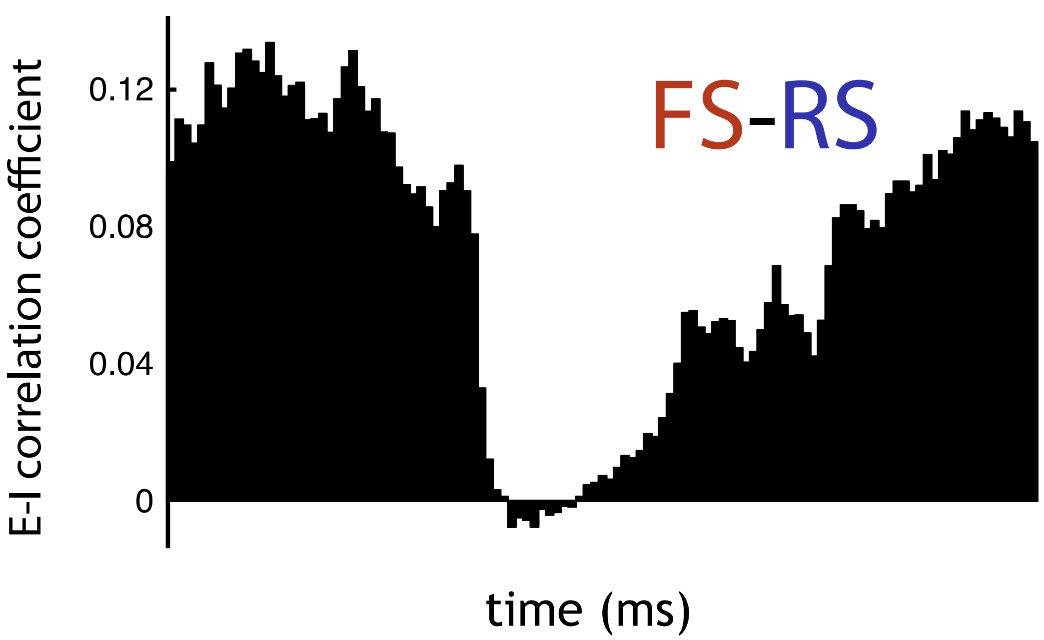
$$\langle \xi_E(t) \rangle = \langle \xi_I(t) \rangle = 0$$
$$\langle \xi_E^2(t) \rangle = \langle \xi_I^2(t) \rangle = 1$$
$$\langle \xi_E(t) \xi_I(t) \rangle = c \in [0, 1]$$

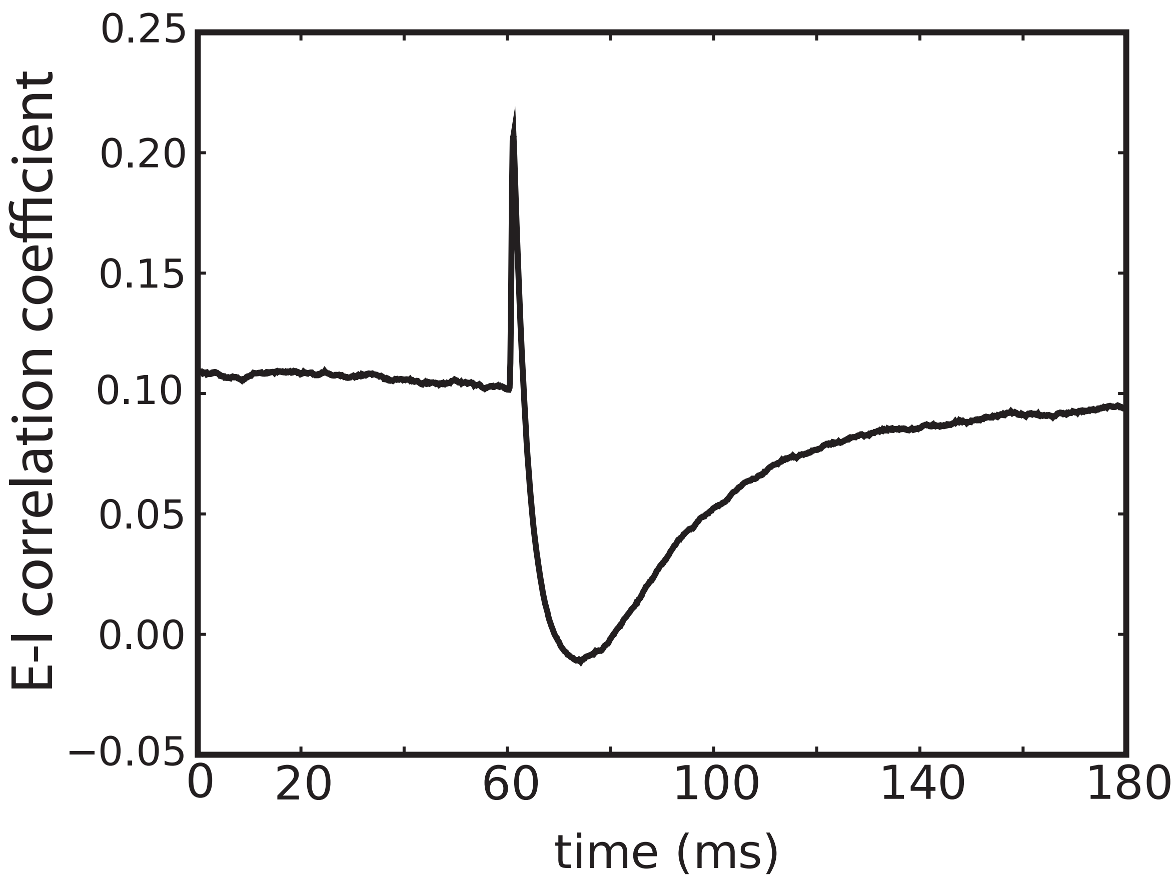
Fitting the model



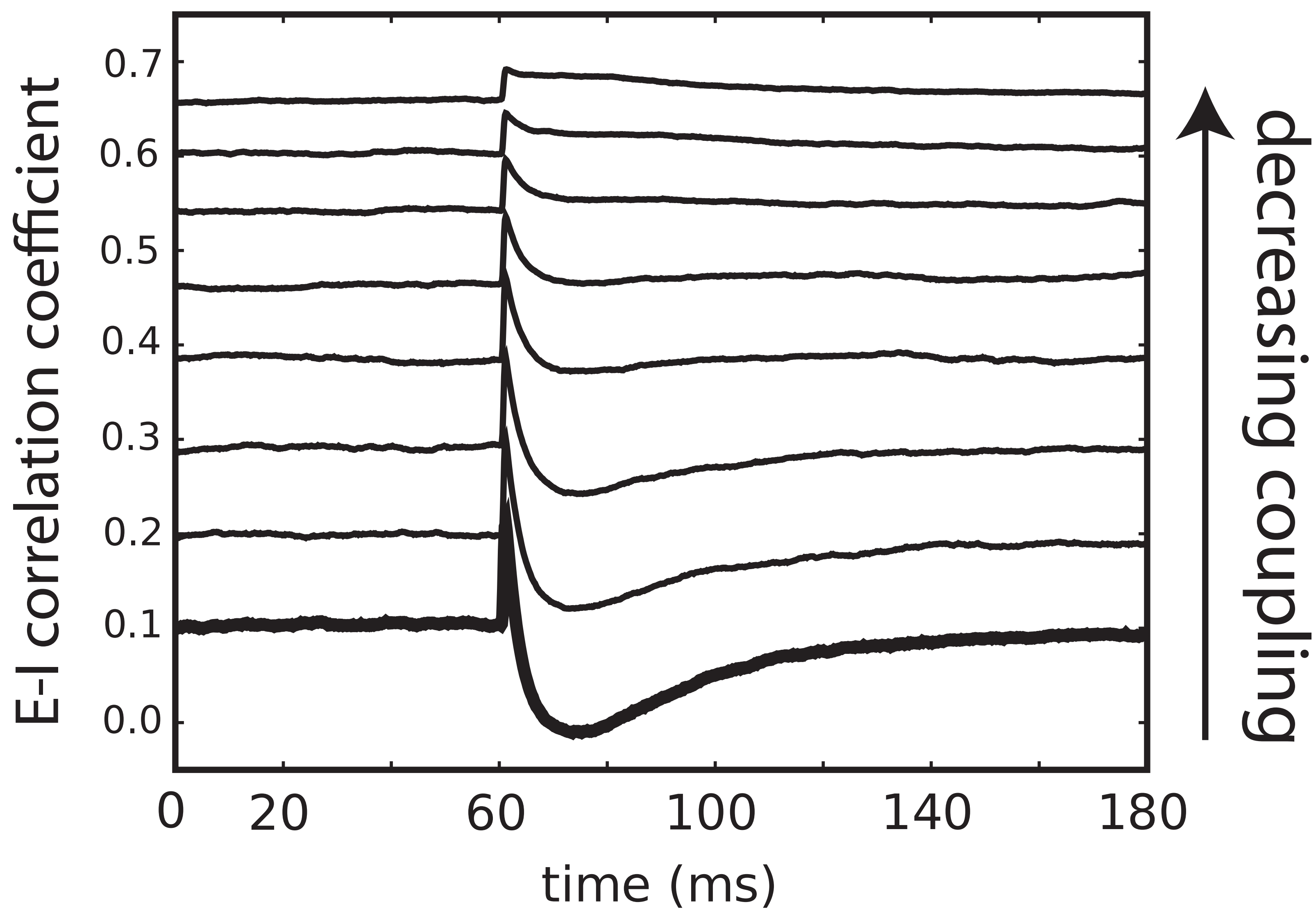


Fitting the model



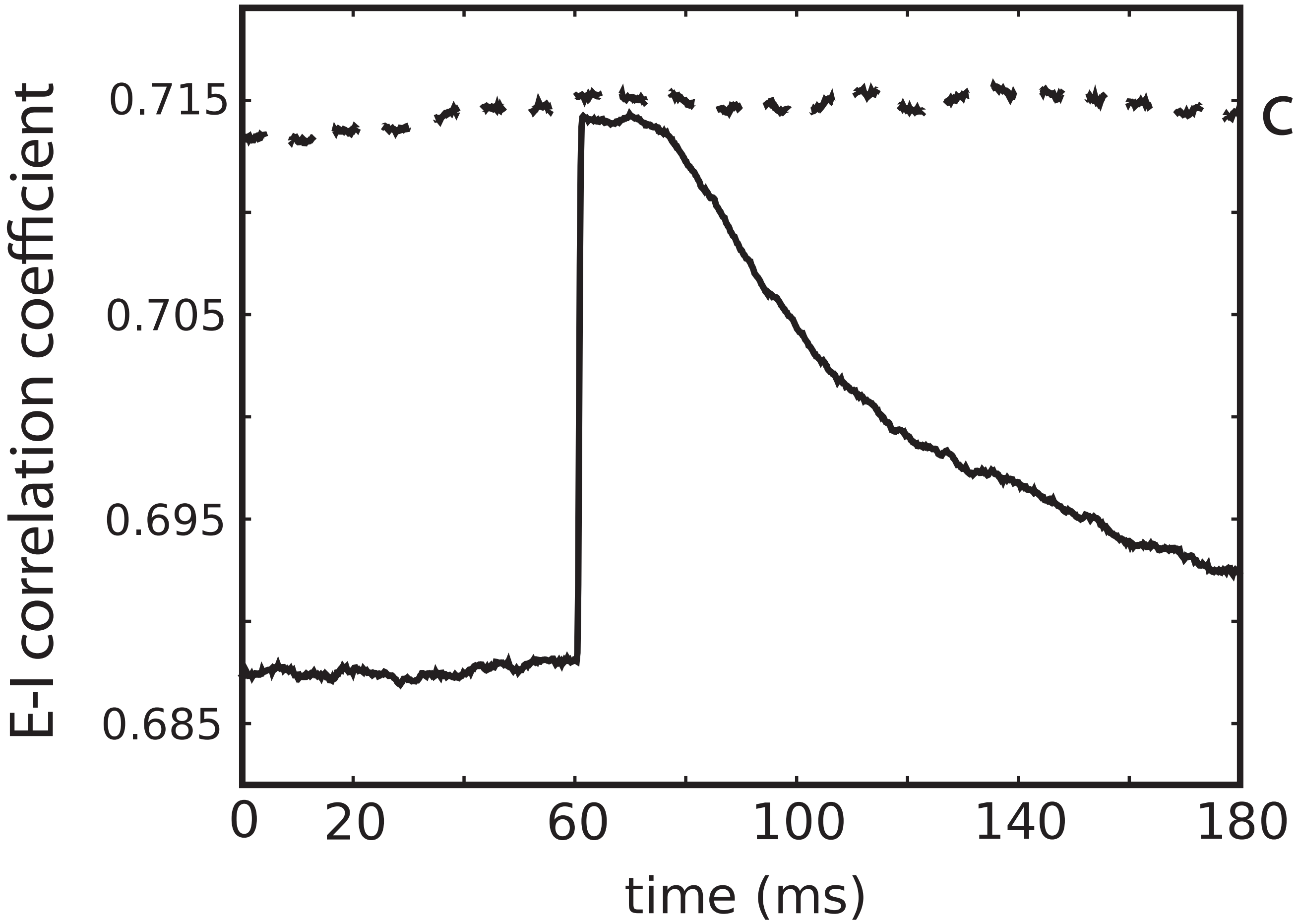


The effect of feedforward inhibition

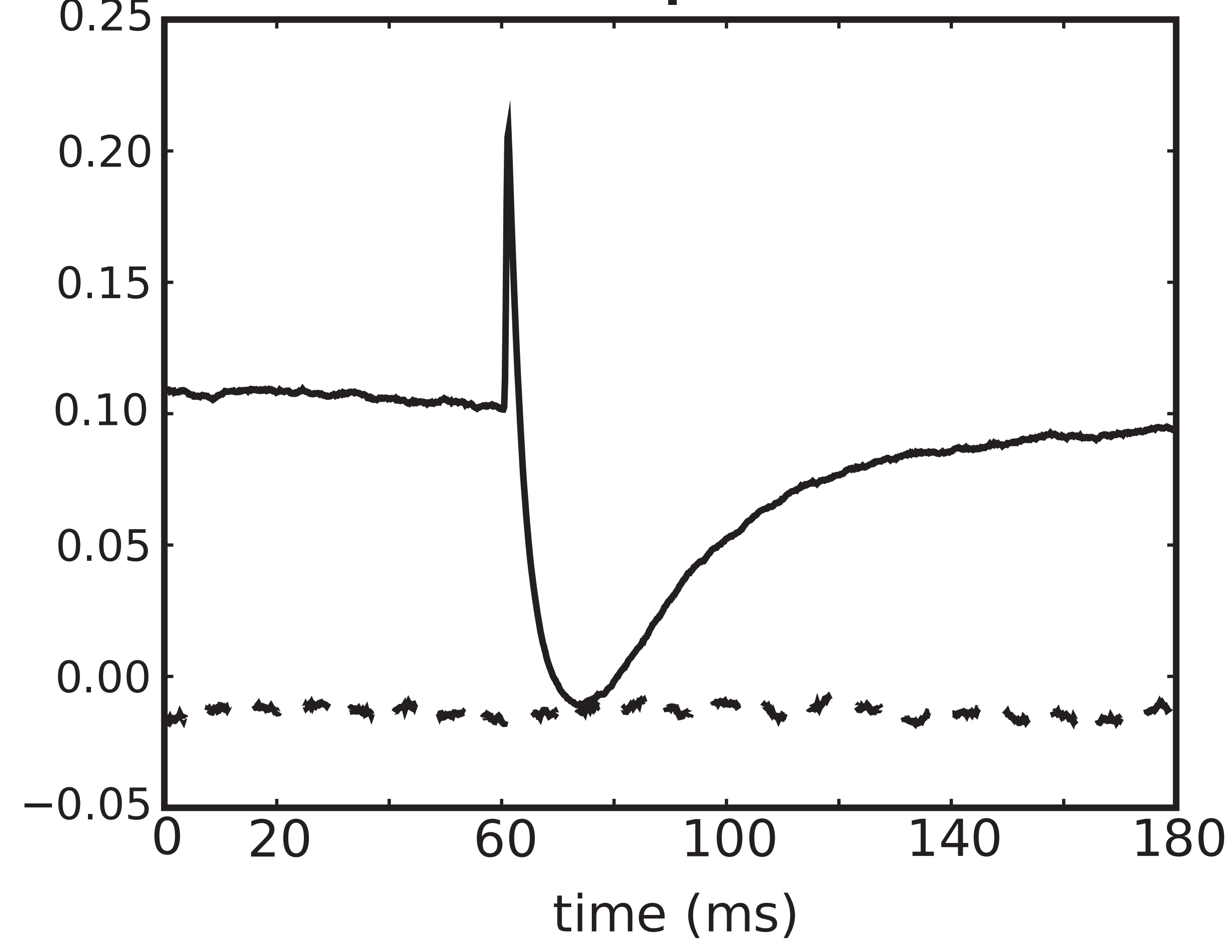


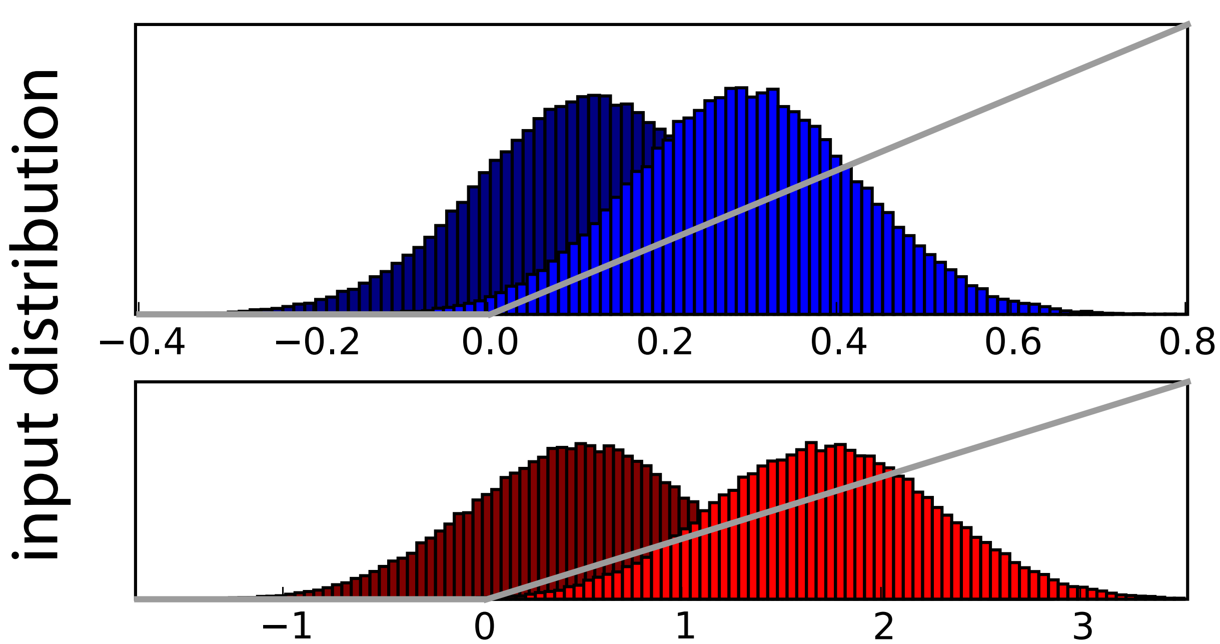
Mechanism

uncoupled



coupled





transfer function:
— non linear
– linear

Summary so far

- Inhibitory coupling anti-correlates neural populations
- Non-linearities dilute this effect
- The evoked state moves you toward the linear part of the transfer function, unlocking the anti-correlating effect of inhibition

Issues with this model

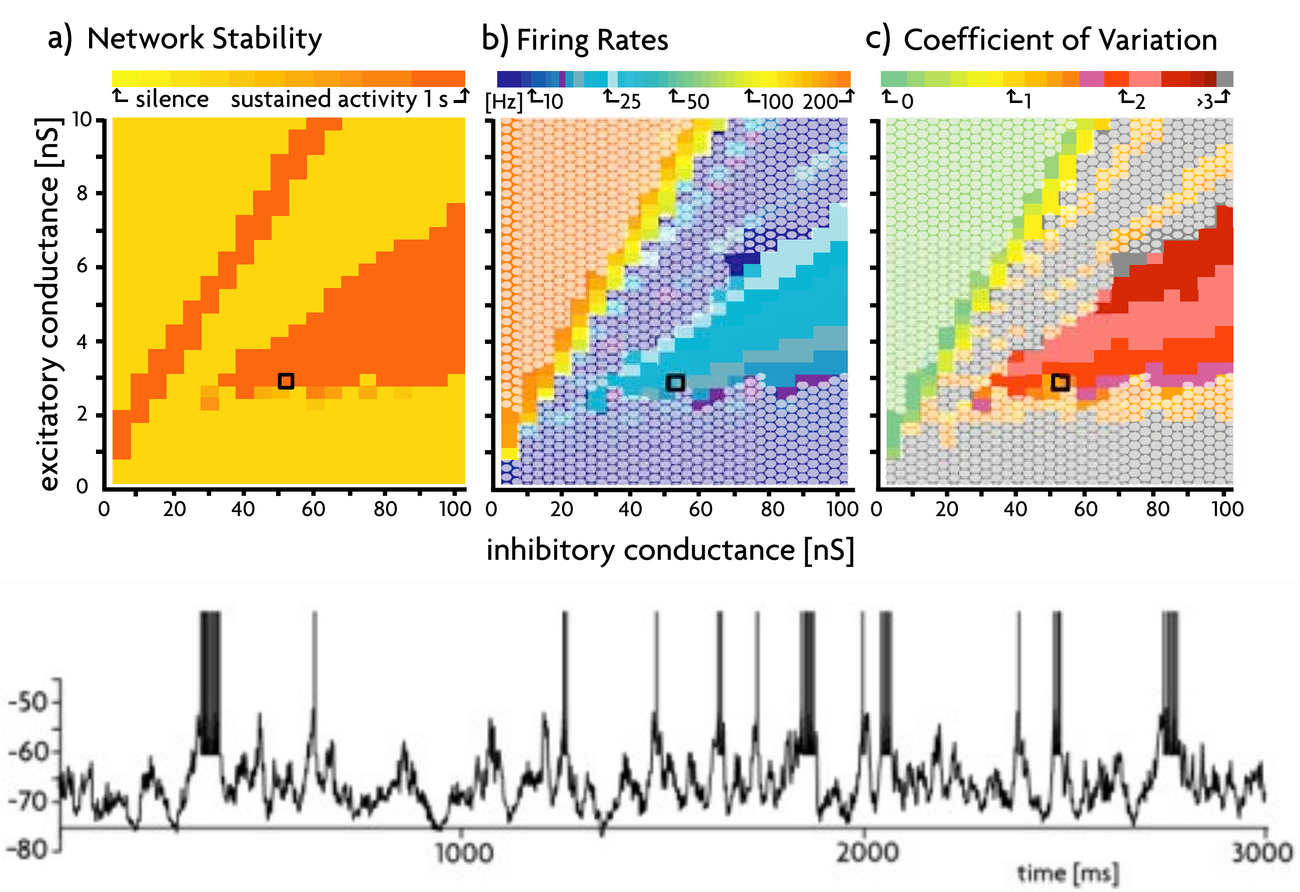
- Missing full set of connections
- Not-quite-realistic non-linearities
- Externally-imposed noise with a fixed and fitted input correlation

A Spiking Model with Internally Generated Variability

$$\tau \frac{dV}{dt} = (V_{\text{rest}} - V) + g_{\text{ex}}(E_{\text{ex}} - V) + g_{\text{inh}}(E_{\text{inh}} - V)$$

$$\tau_{\text{ex}} \frac{dg_{\text{ex}}}{dt} = -g_{\text{ex}} \qquad g_{\text{ex}} \rightarrow g_{\text{ex}} + \Delta g_{\text{ex}}$$

$$\tau_{\text{inh}} \frac{dg_{\text{inh}}}{dt} = -g_{\text{inh}} \qquad g_{\text{inh}} \rightarrow g_{\text{inh}} + \Delta g_{\text{inh}}$$



(Vogels & Abbott, 2005)

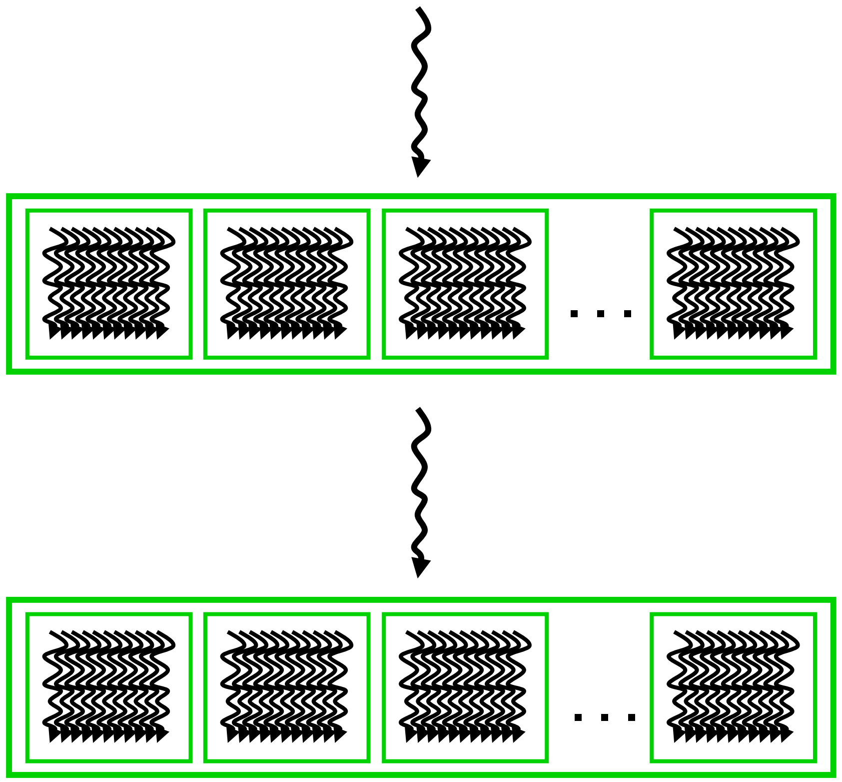
Efficiently simulating and analyzing thousands of trials with a large spiking network

- 4000 neurons, 1000 trials
- 8 million pairwise correlations to calculate per bin, 50 bins
- Conventional code: 1.5 hours for simulations, 8 hours to calculate correlations

GPUs

(Graphics Processing Units)

- Massively parallel single precision floating point
- Have to program in SPMD (single program multiple data) style - thousands of threads all running the same code on different parts of memory



SpikeStream

- Python framework for simulation and spike train analysis
- Specify models and computations in Python (a very nice language!)
- Code generation techniques produce underlying CUDA code for the device

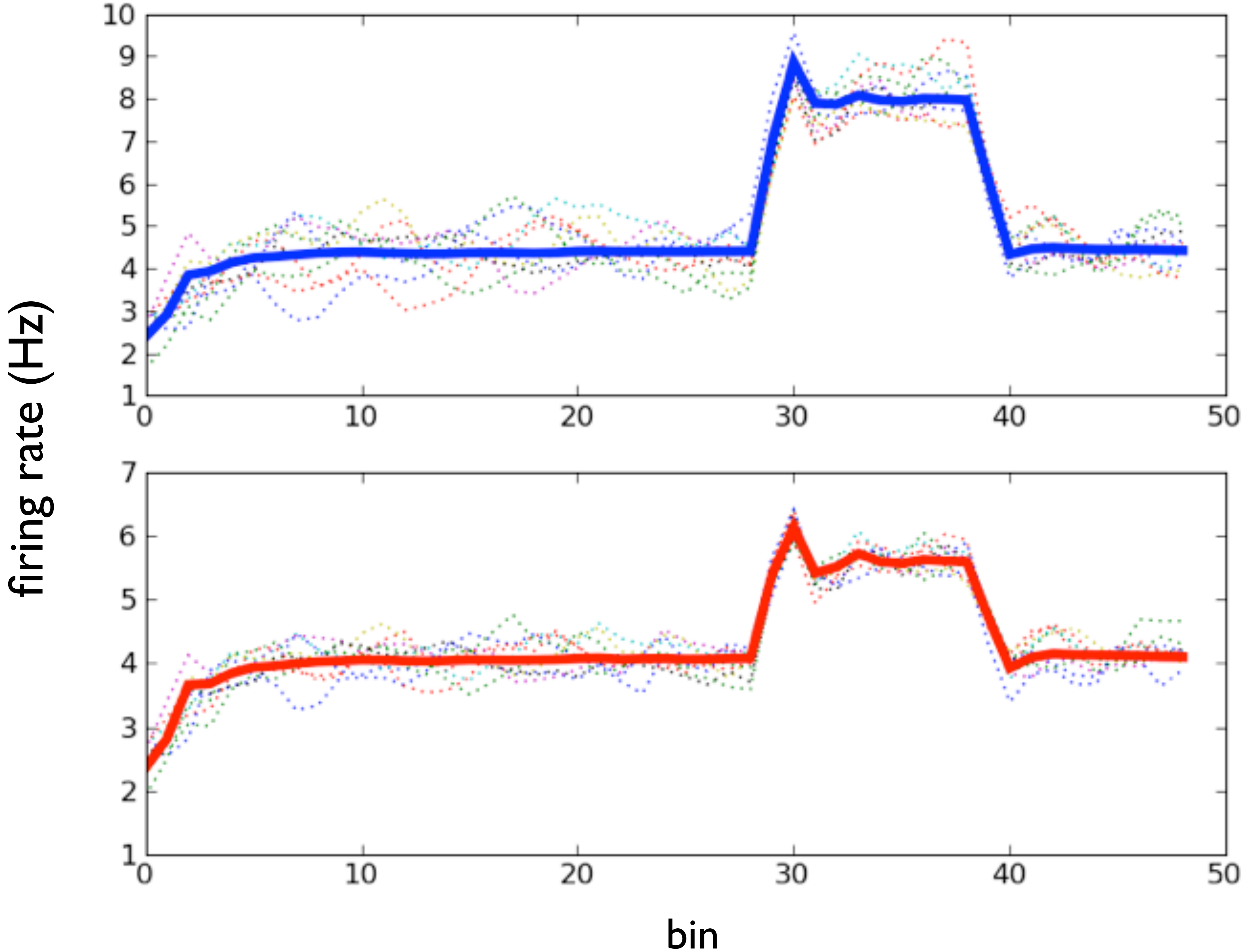
Example

- (show Python code and generated CUDA code)

SpikeStream

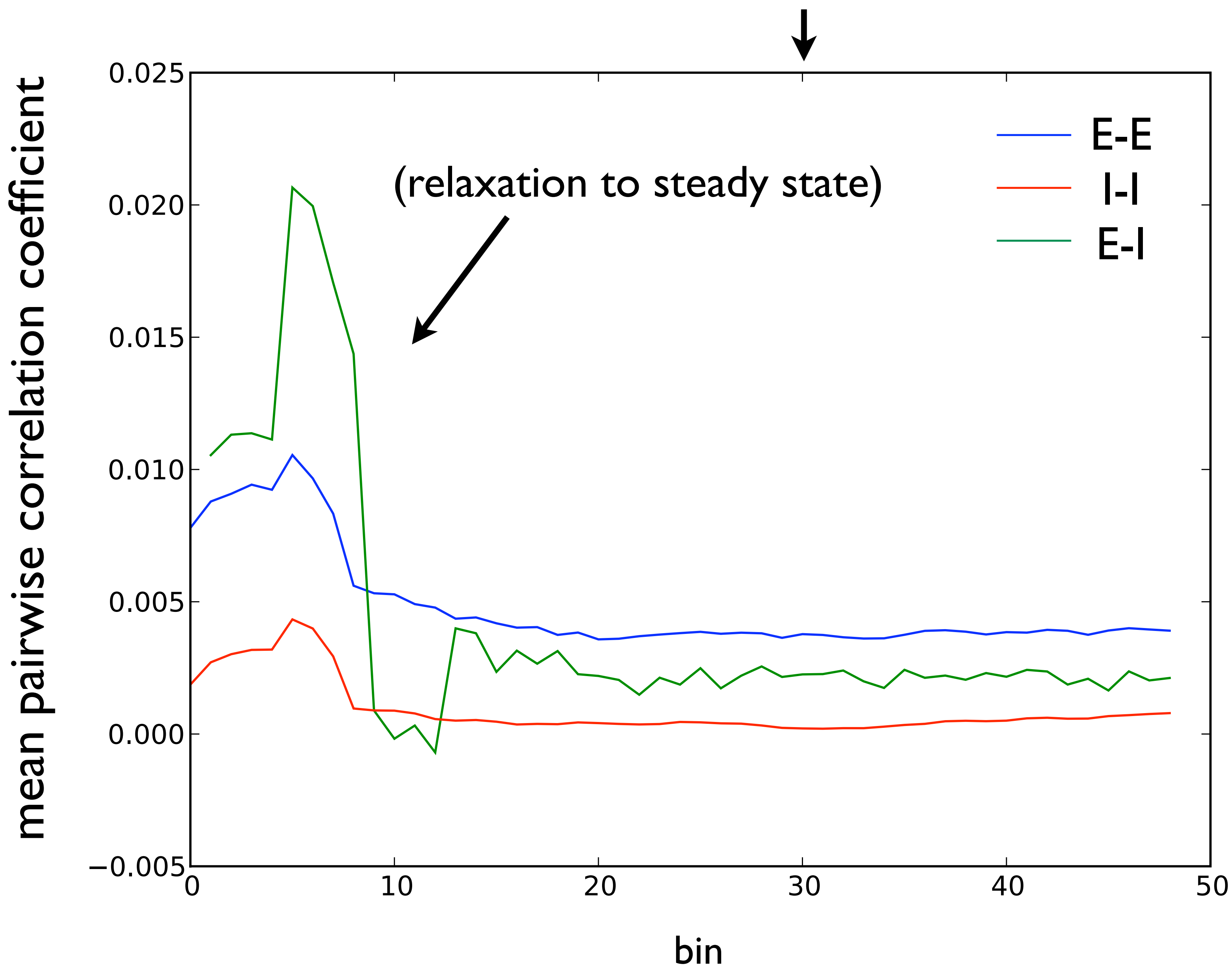
- Interesting performance characteristics:
  - More complex models not much slower (rate-limiting step: spike propagation)
    - Large memory access latency, hundreds of accesses per spike
  - Memory limits: 4GB per card, so millions of synapses (or thousands of repeats of a smaller sim), but not billions
    - Multiple cards can be used for repeats, but not easy to extend one sim over multiple cards

A Spiking Model with Internally Generated Variability



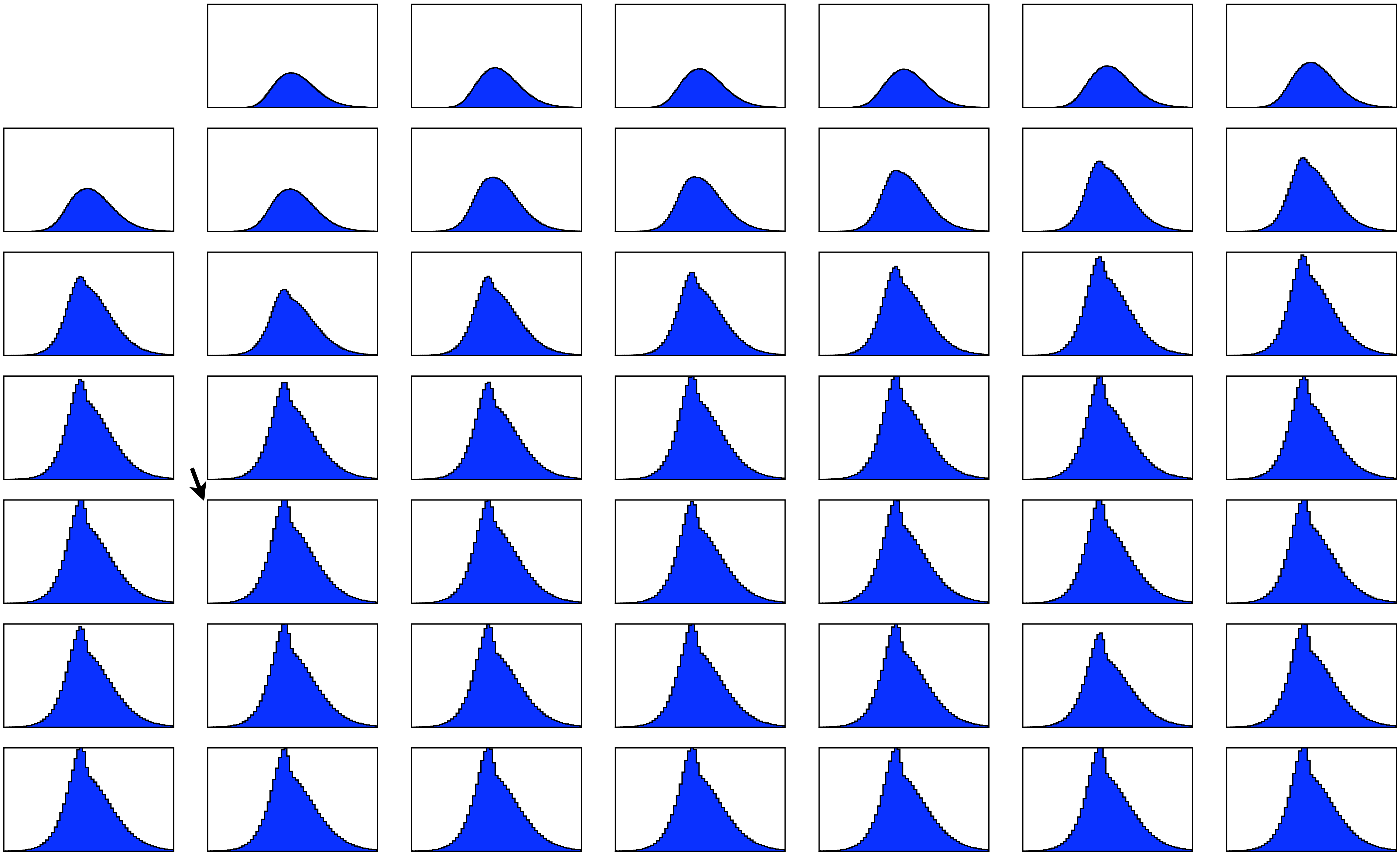
(1.2 hours on CPU, 30 seconds on GPU = 130x speedup)

The system is acting linearly

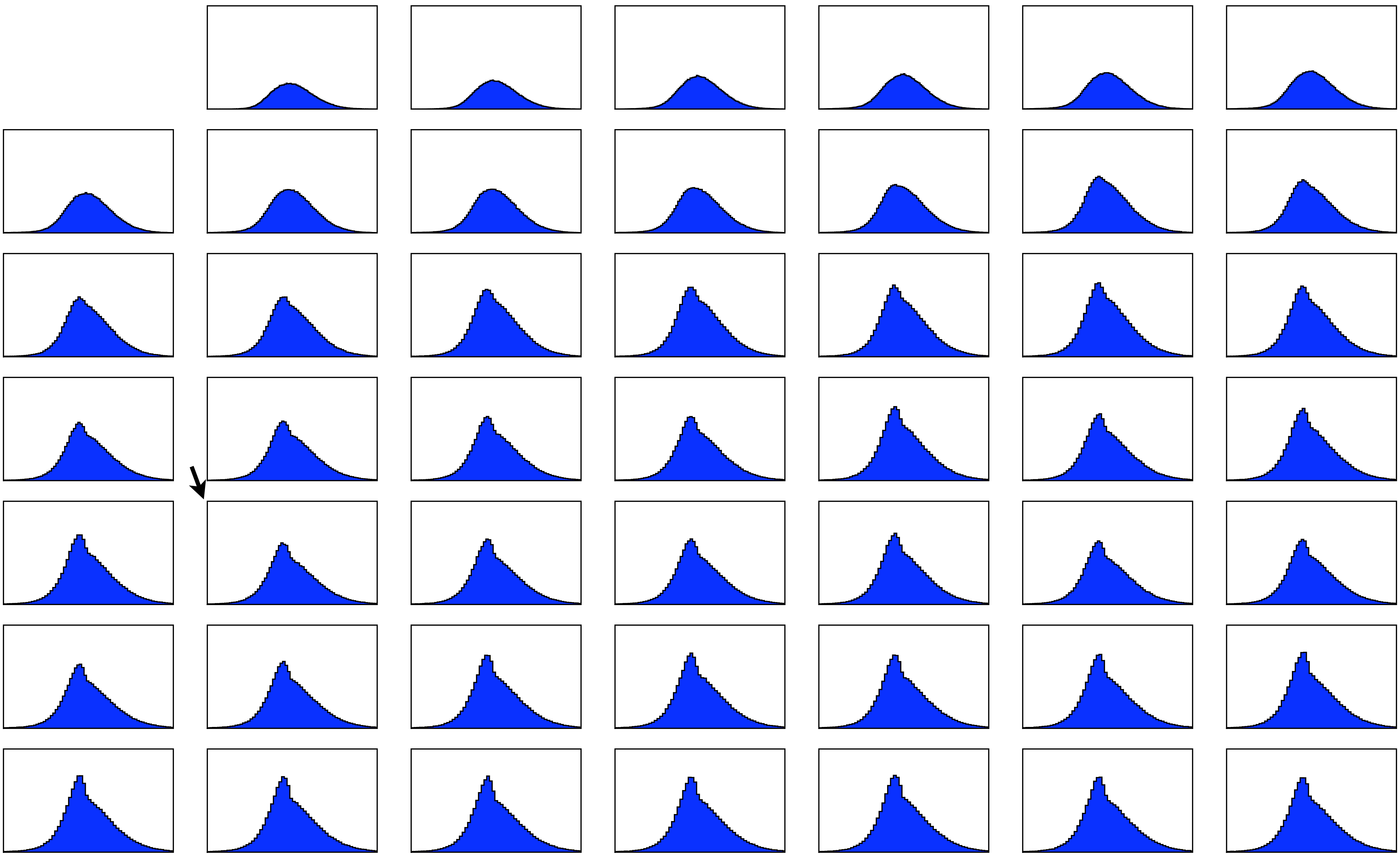


(8 hours on CPU, 2 minutes on GPU = 240x speedup)

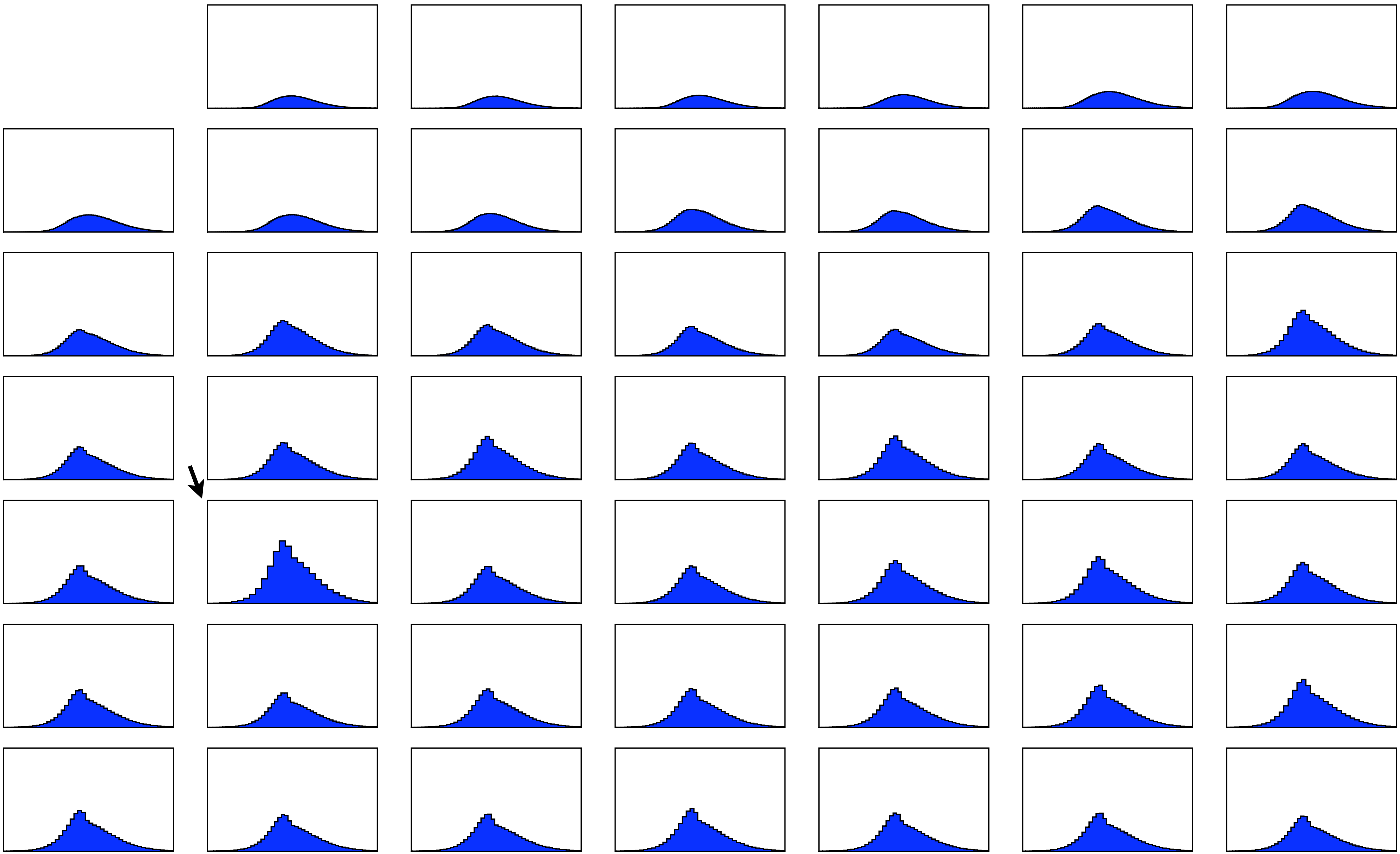
E-E Correlation Distribution For Each Bin



I-I Correlation Distribution For Each Bin



E-I Correlation Distribution For Each Bin



Questions

- Can we understand the peculiar shape of the correlation distribution in this network?
- Can we modify the network to behave like Jay's data?
  - More realistic connectivity (sparse vs. dense)
  - More realistic coupling (strong feedforward inhibition)
  - Different non-linearities in the neurons
  - More realistic input

Summary

- Correlations in neural systems are affected by connectivity and non-linearities in complex ways. The details matter.
- GPU computing opens up new avenues for approaching this problem with larger-scale models with more realistic characteristics.

Thank You!