Spike-triggered Average, Spike-triggered Covariance, LNP models

Jonathan Pillow

Methods in Computational Neuroscience

(NEU 394P, PSY 394U)

Spring 2010

2^{nd} idea: polynomial model (Volterra/ Wiener Kernels)

Taylor series expansion of a function f(x) in *n* dimensions

$$y \;=\; k_0 + \vec{k}_1 \cdot \vec{x} + \vec{x}^t K_2 \vec{x} + K_3 \cdot \vec{x}^3 + \ldots$$

	const	vector	matrix	3-tensor

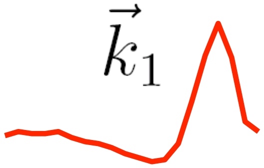
# pars:	1	n (20)	n^2 (400)	n^3 (8000)

- estimate kernels using moments of spike-triggered stimuli
- in practice, rarely have enough data to go beyond 2^{nd} order.

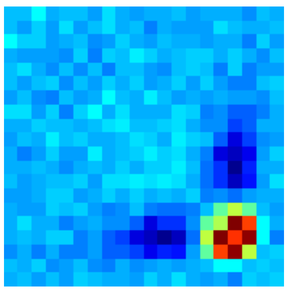
Quadratic Model

$$y \;=\; k_0 + \vec{k}_1 \cdot \vec{x} + \vec{x}^t K_2 \vec{x}$$

$\vec{k}_1$



K_2



<u>HW problem:</u>

Show that if: $x \sim \mathcal{N}(0, I)$ (Gaussian white noise stimuli)

$$\frac{1}{n_{sp}} \sum_{i=1}^{N} y_i x_i$$, the spike-triggered average

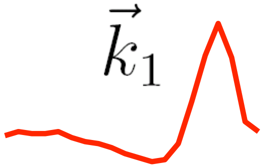
and $$\frac{1}{n_{sp}} \sum_{i=1}^{N} y_i (x_i x_i^T)$$, the spike-triggered covariance

provide a consistent estimators for k_1 and K_2, respectively.

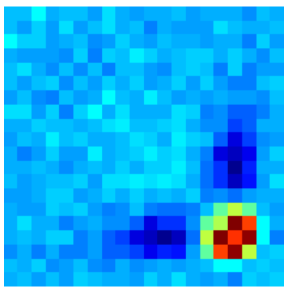
Test of Quadratic Model

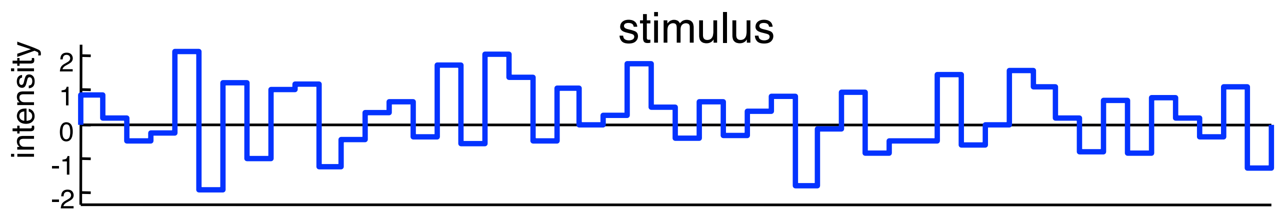
$$y \quad = \quad k_0 + \vec{k}_1 \cdot \vec{x} + \vec{x}^t K_2 \vec{x}$$

$\vec{k}_1$



K_2

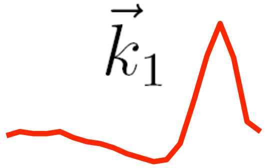




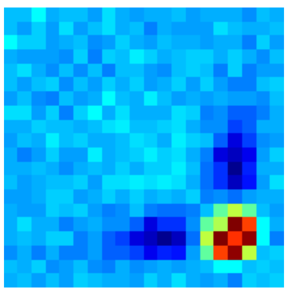
Test of Quadratic Model

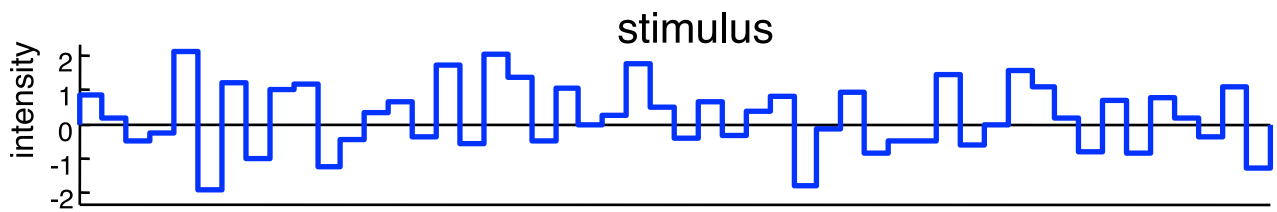
$$y \quad = \quad k_0 + \vec{k}_1 \cdot \vec{x} + \vec{x}^t K_2 \vec{x}$$

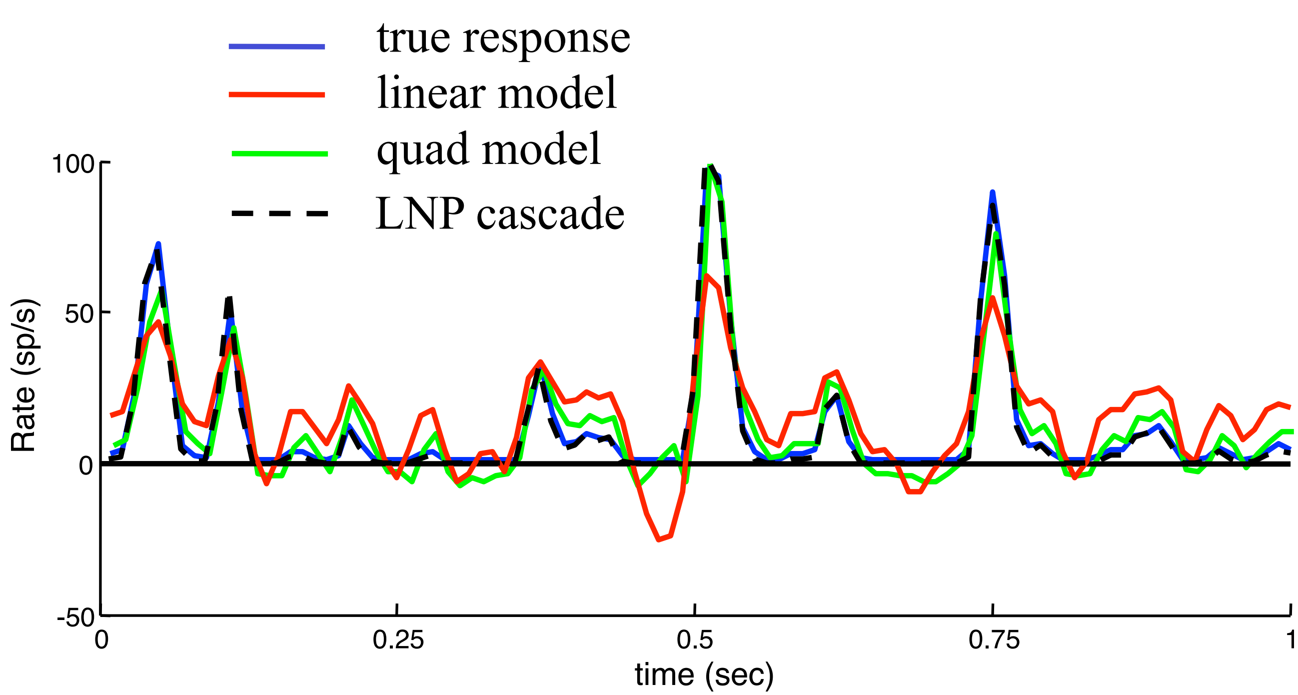
$\vec{k}_1$



K_2

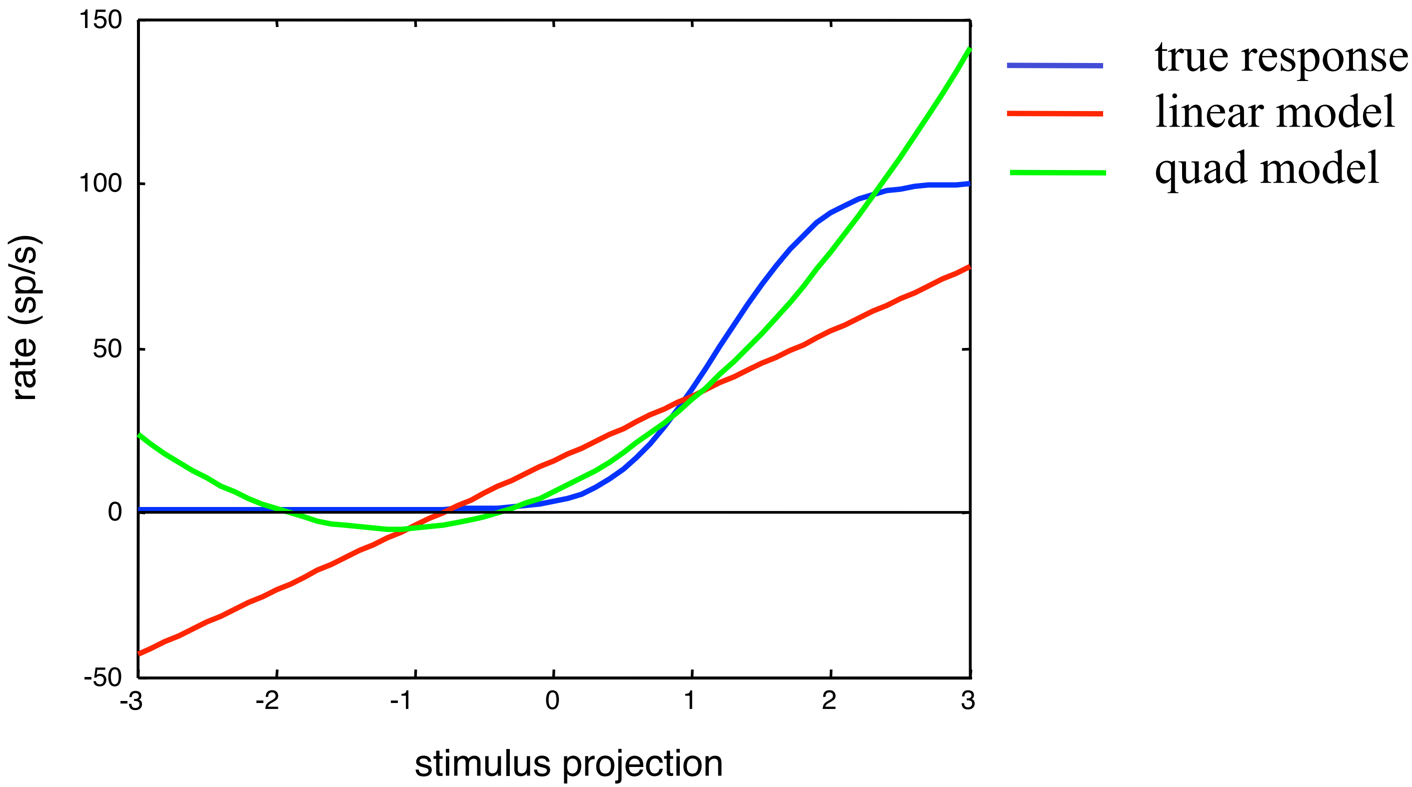






Why do Volterra/Wiener models perform poorly?

Polynomials do a poor job of representing the nonlinearities found in neurons.



Volterra / Wiener expansion for functionals

$$f(t) = \sum_i K_i \otimes x_t^{(i)}$$

i'th order tensor

$$K_i \otimes x_t^{(i)} = \sum_{j_1,\ldots,j_i} K(j_1,\ldots,j_i) \cdot \Big(x(t-j_1) \cdot \ldots \cdot x(t-j_i) \Big)$$

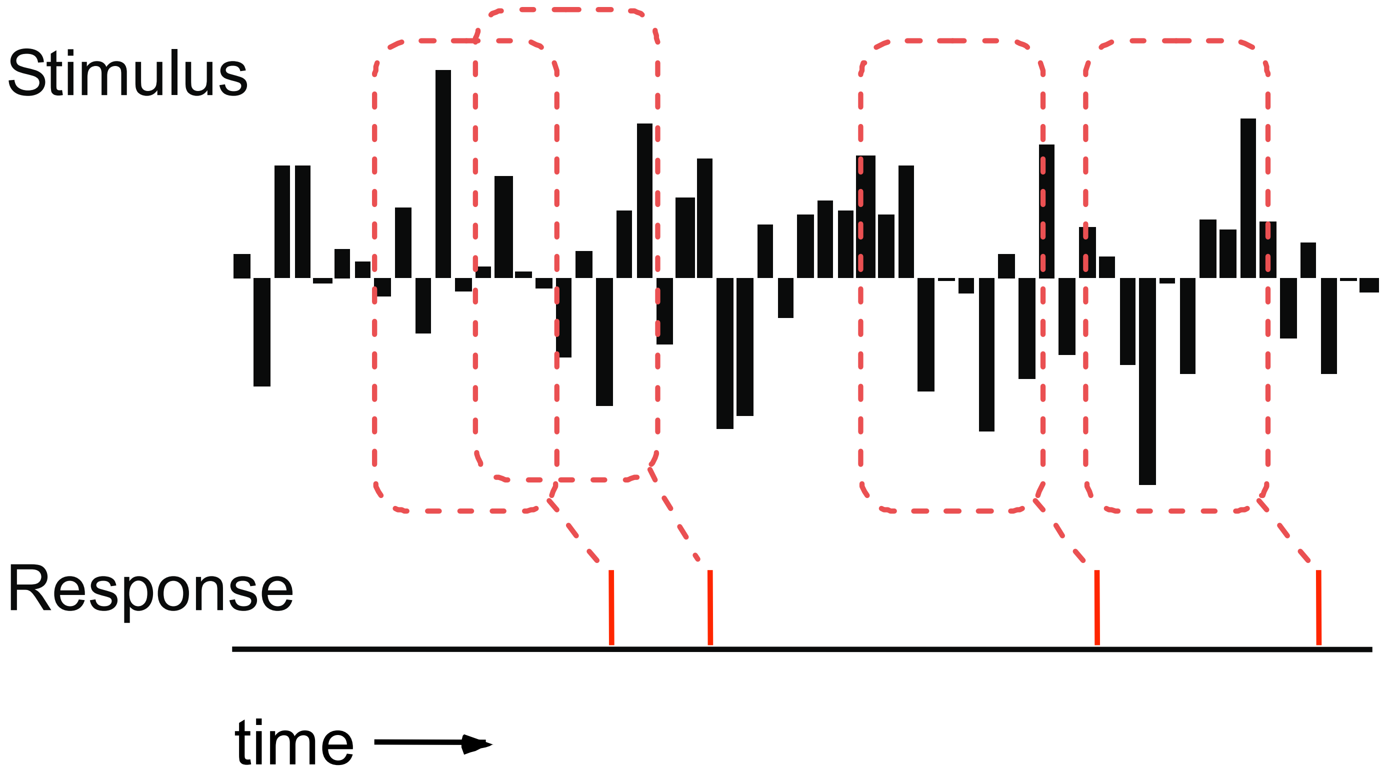
Summary so far:

- Linear Models, Volterra/Wiener Kernels:
  fit a polynomial to $E(y|x)$

Next Up:

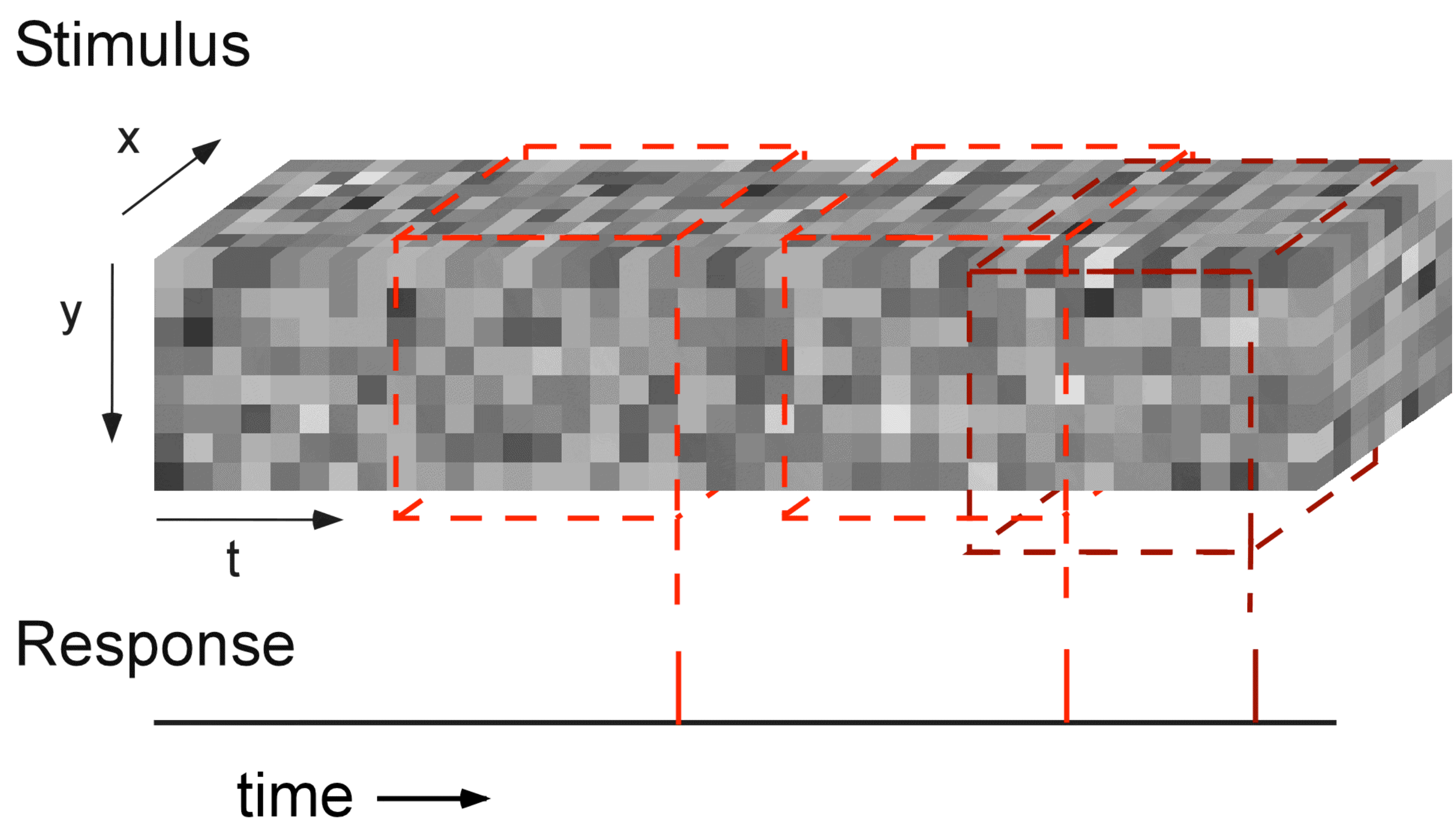
- cascade models (Linear-nonlinear-Poisson)

Spike-triggered ensemble



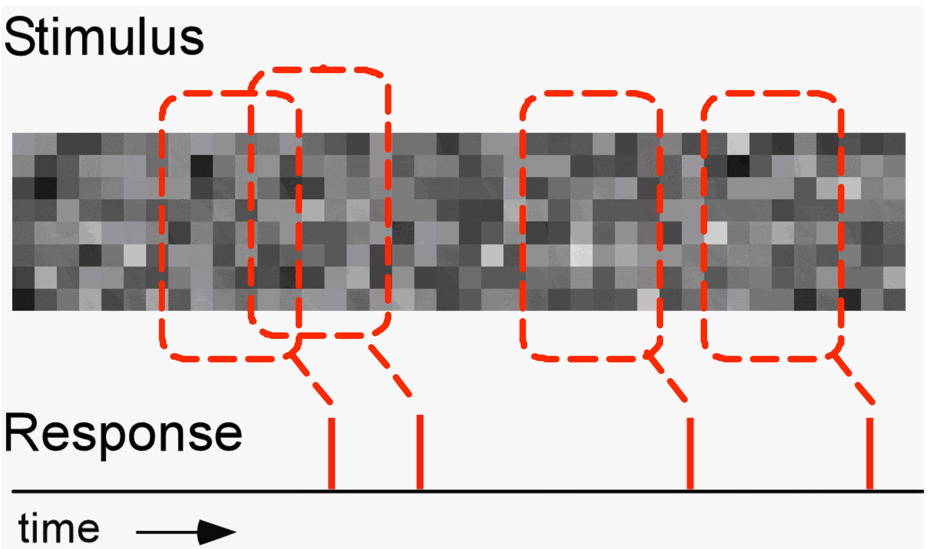
- 9-sample stimulus block

Spike-triggered ensemble (3D stimulus)



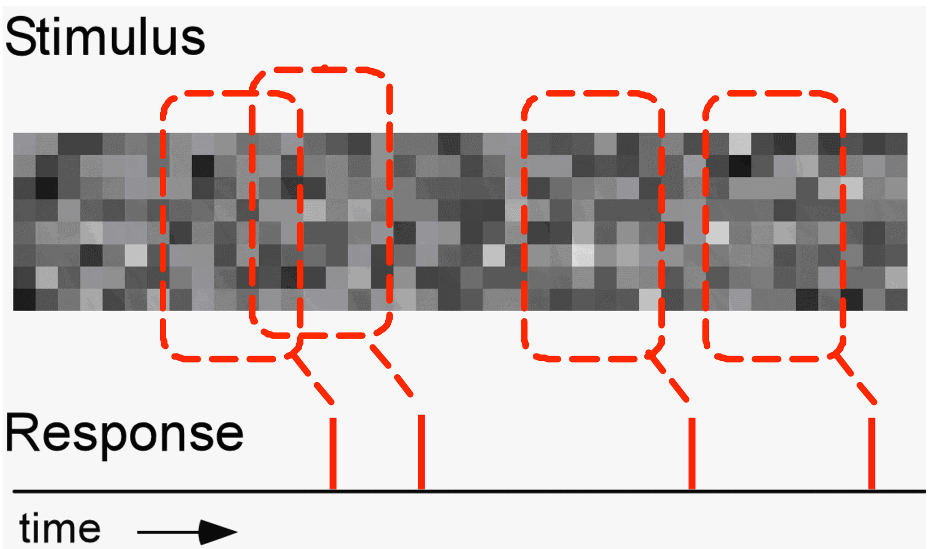
- 8 x 8 x 10 stimulus block

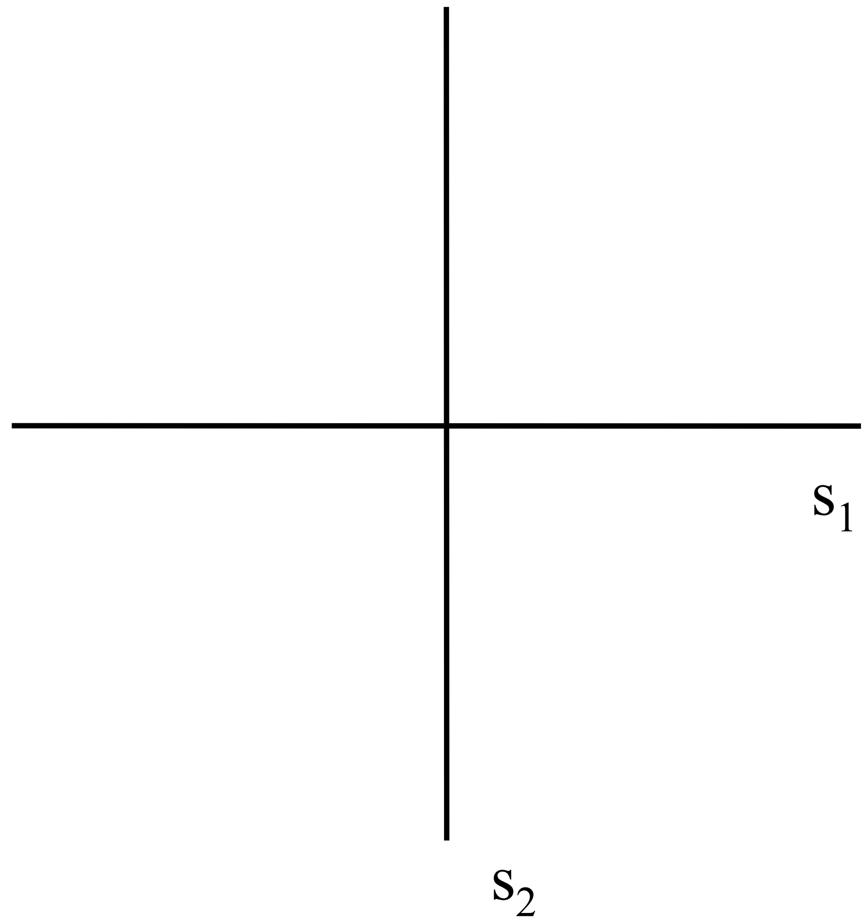
2D stimulus (flickering bars)



- 8 x 6 stimulus block
  = 48-dimensional vector

Geometric picture



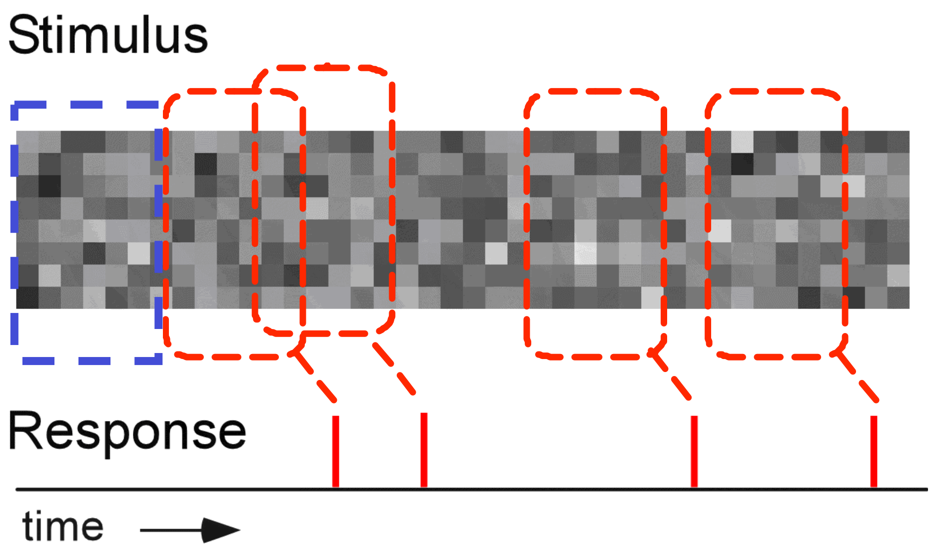


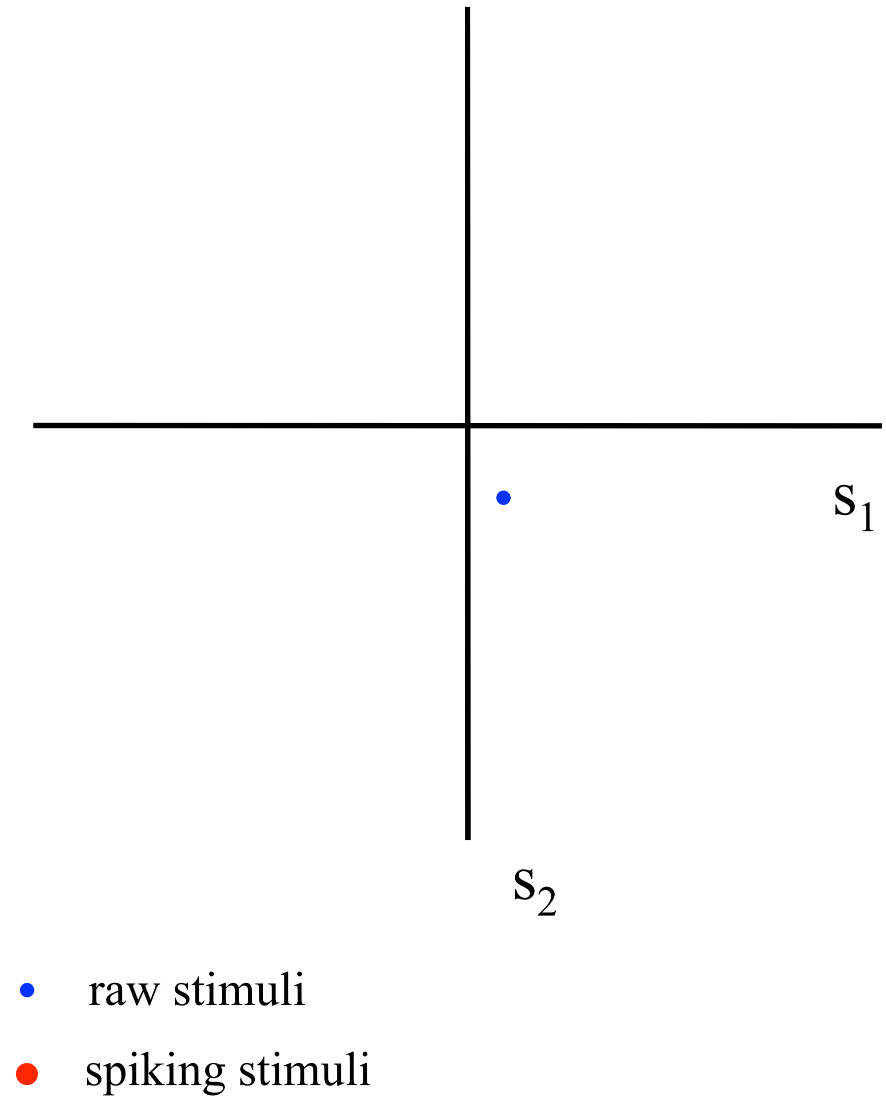
- 8 x 6 stimulus block
  = 48-dimensional vector

• raw stimuli

• spiking stimuli

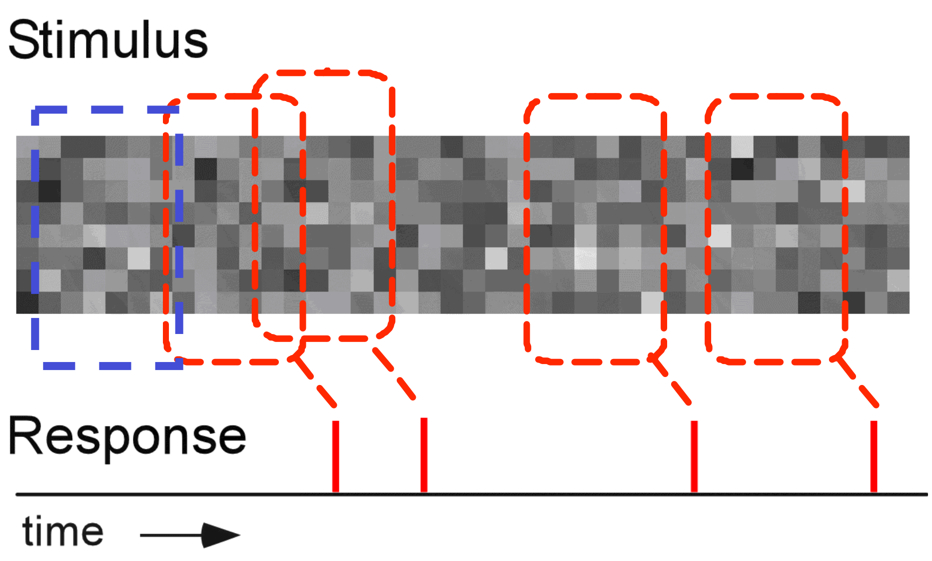
Geometric picture

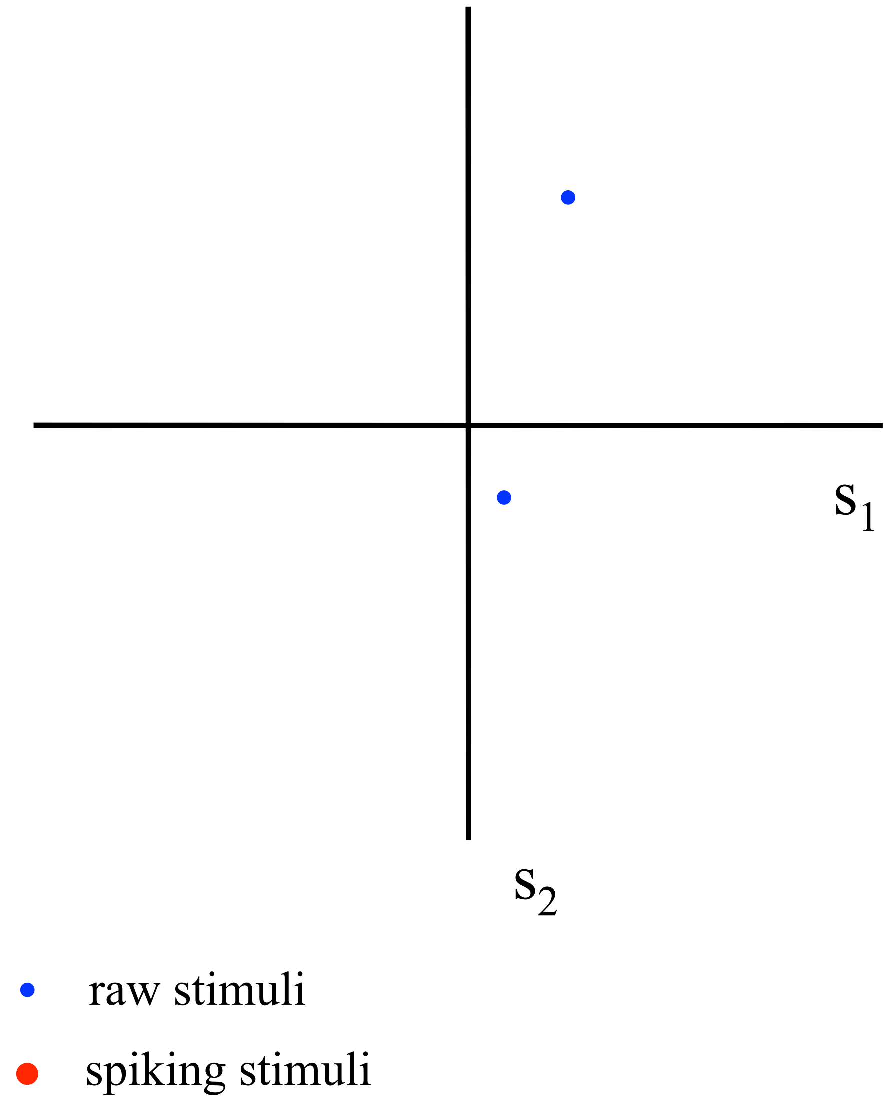




- 8 x 6 stimulus block
  = 48-dimensional vector

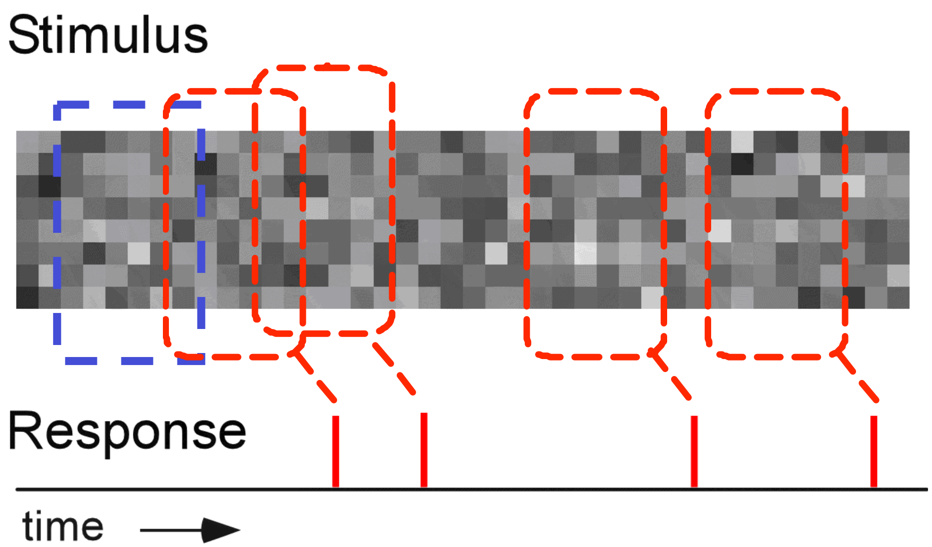
Geometric picture

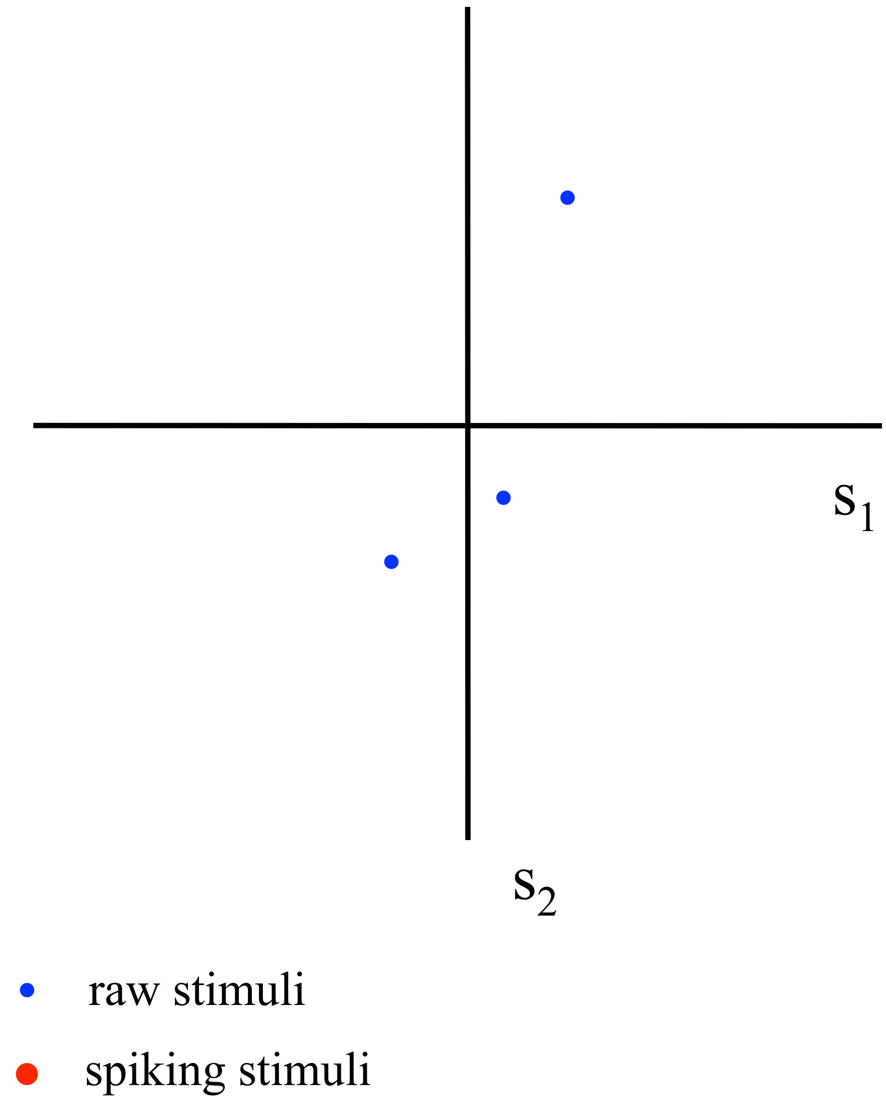




- 8 x 6 stimulus block
  = 48-dimensional vector

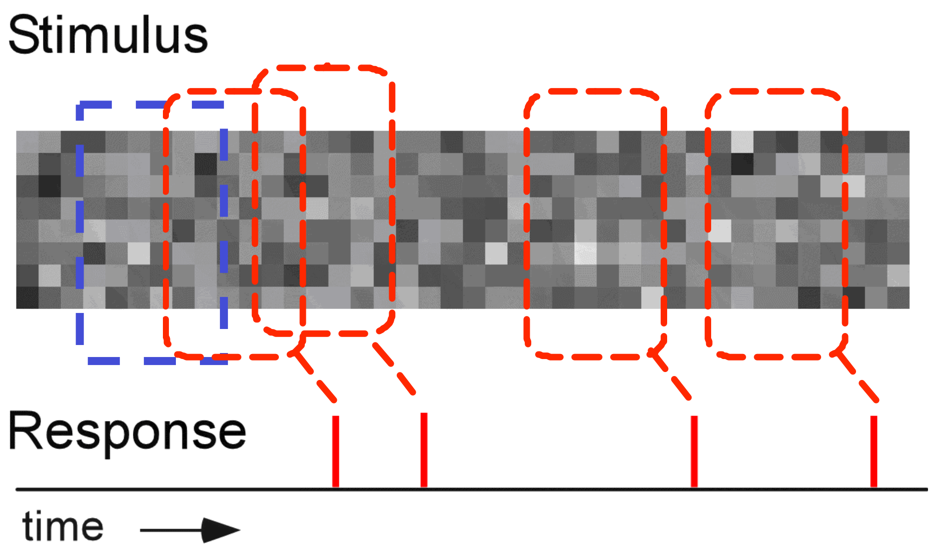
Geometric picture

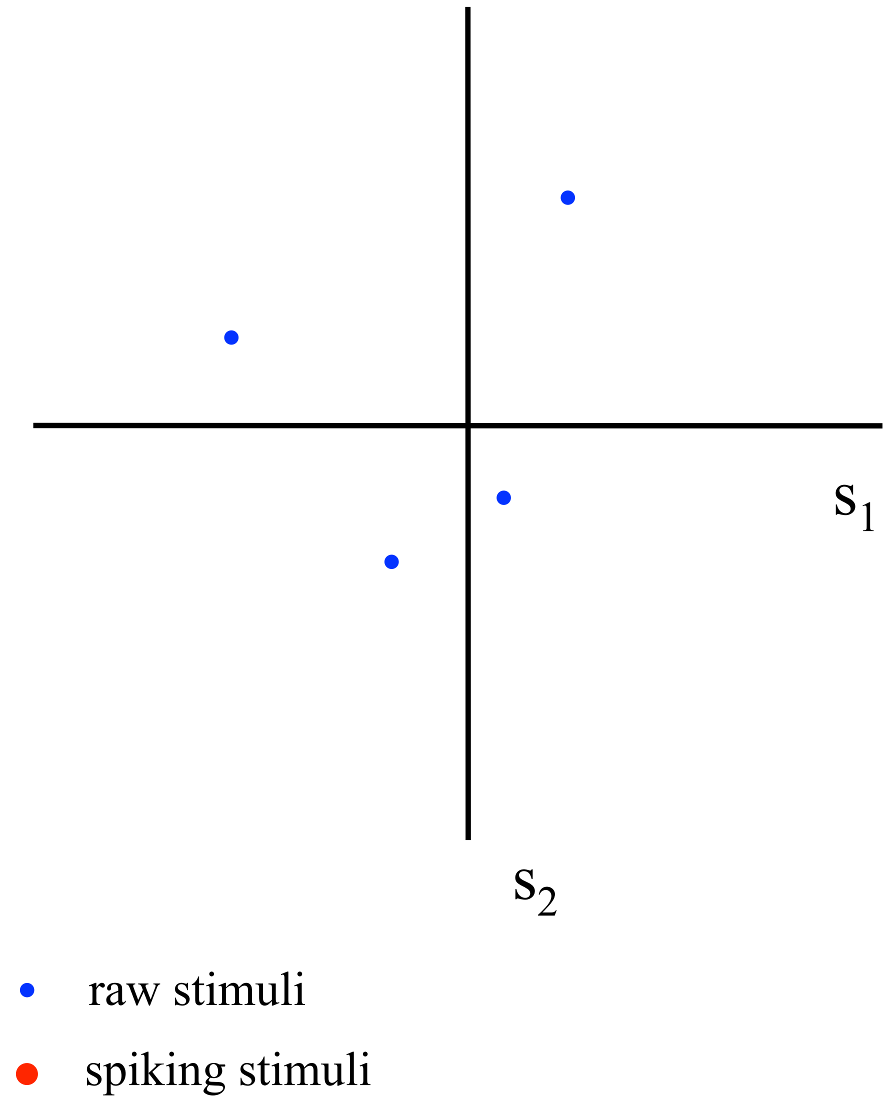




- 8 x 6 stimulus block
  = 48-dimensional vector

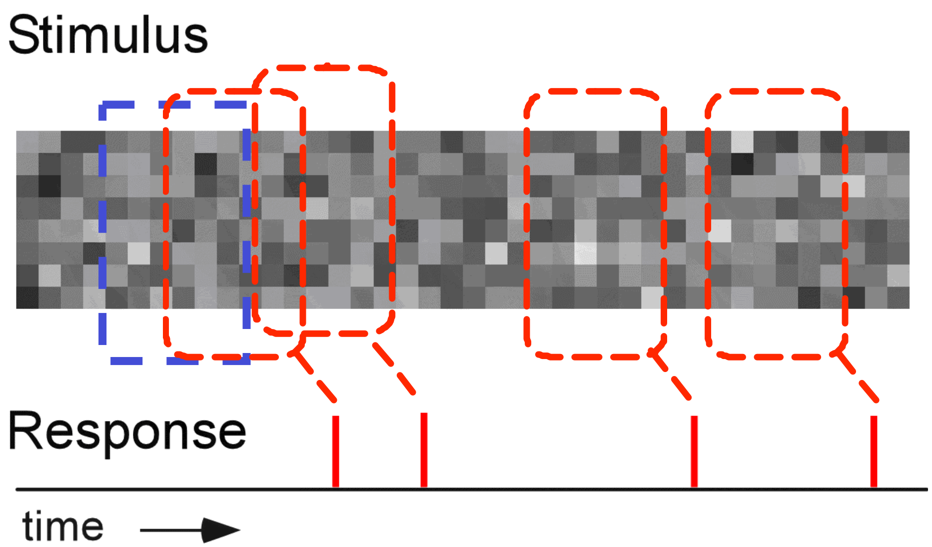
Geometric picture

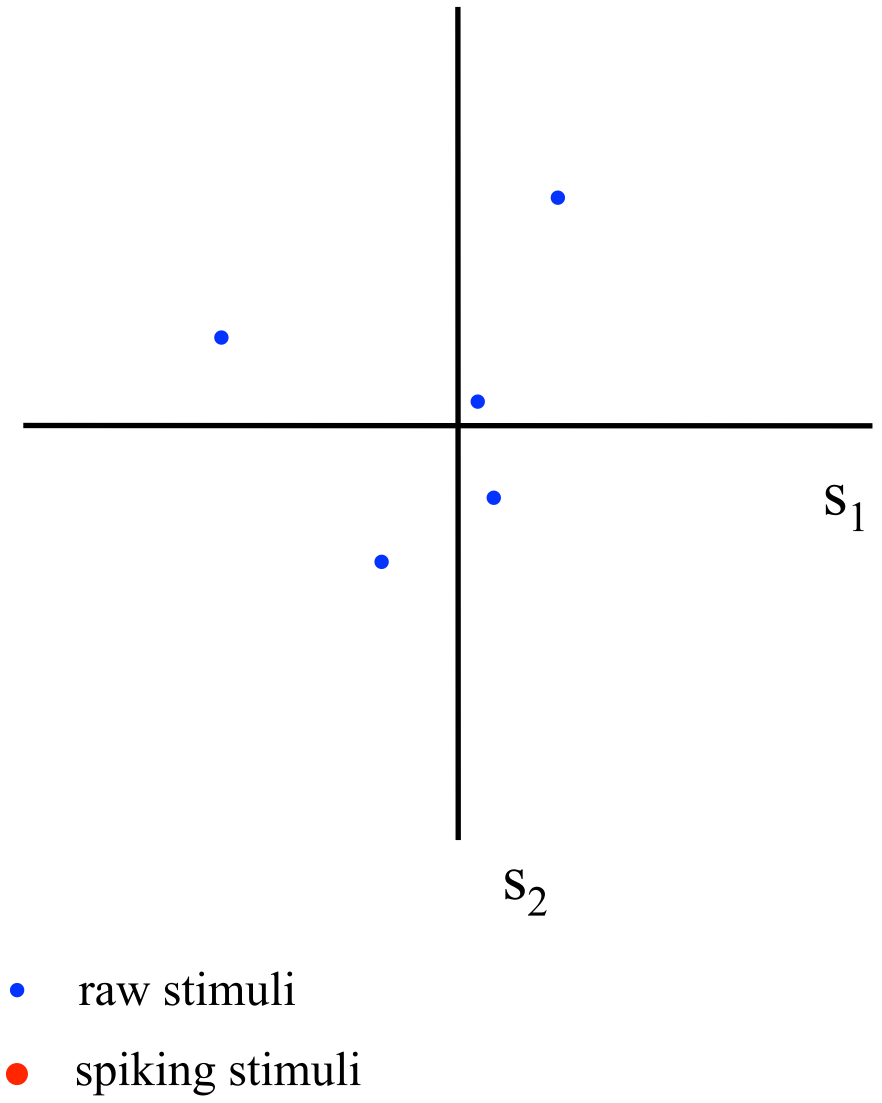




- 8 x 6 stimulus block
  = 48-dimensional vector

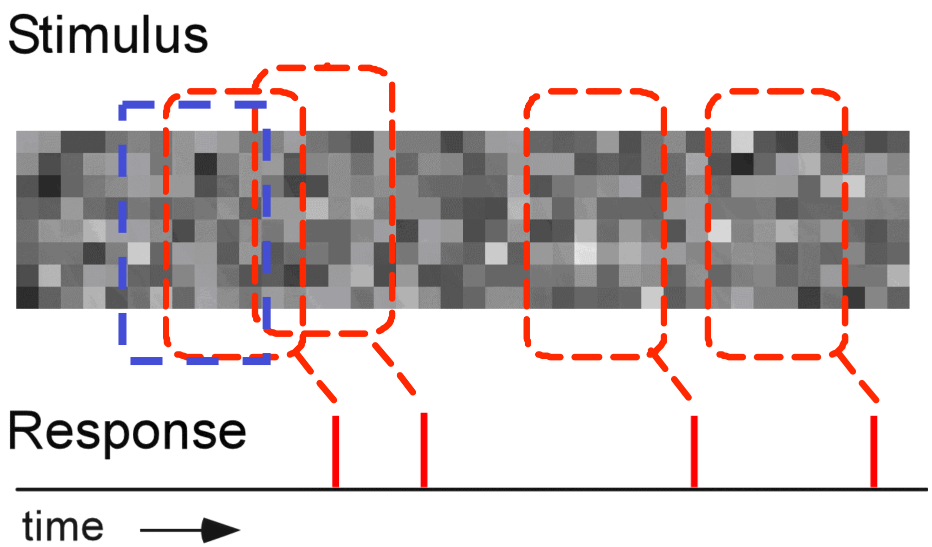
Geometric picture

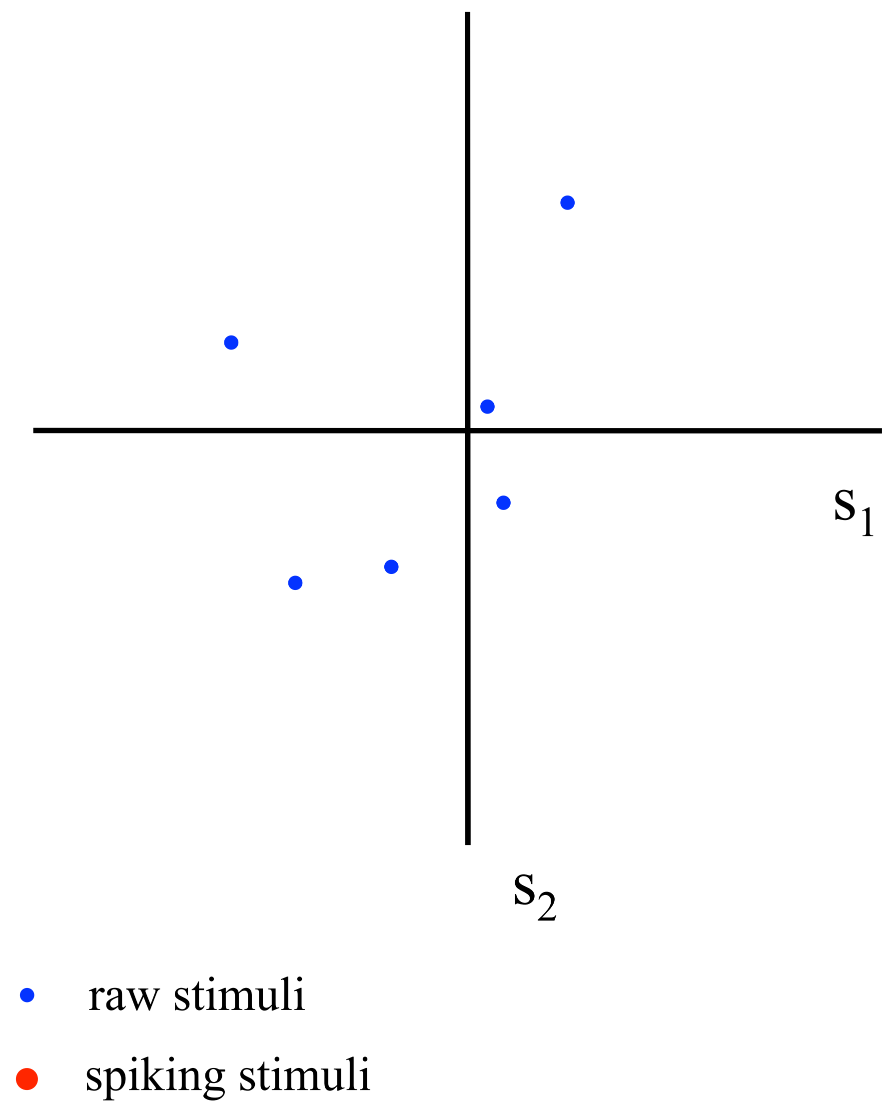




- 8 x 6 stimulus block
  = 48-dimensional vector

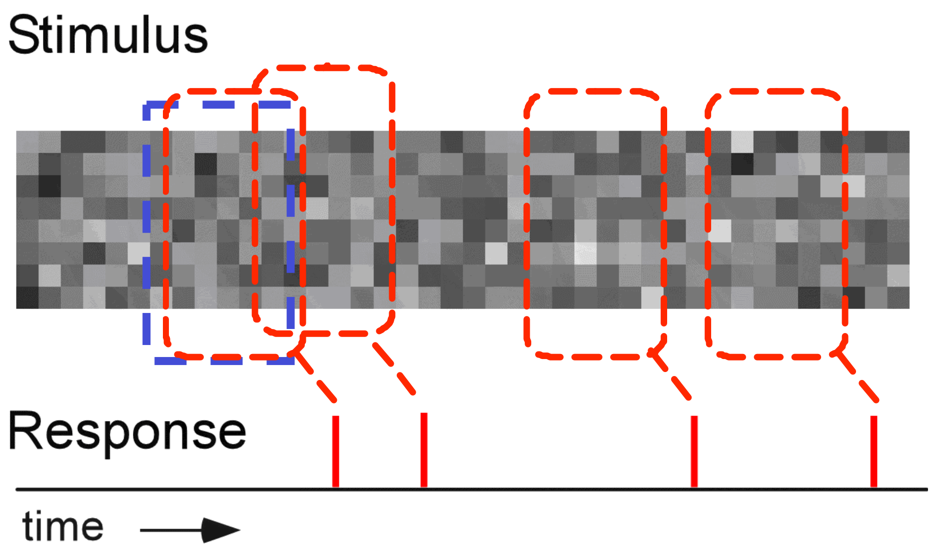
Geometric picture

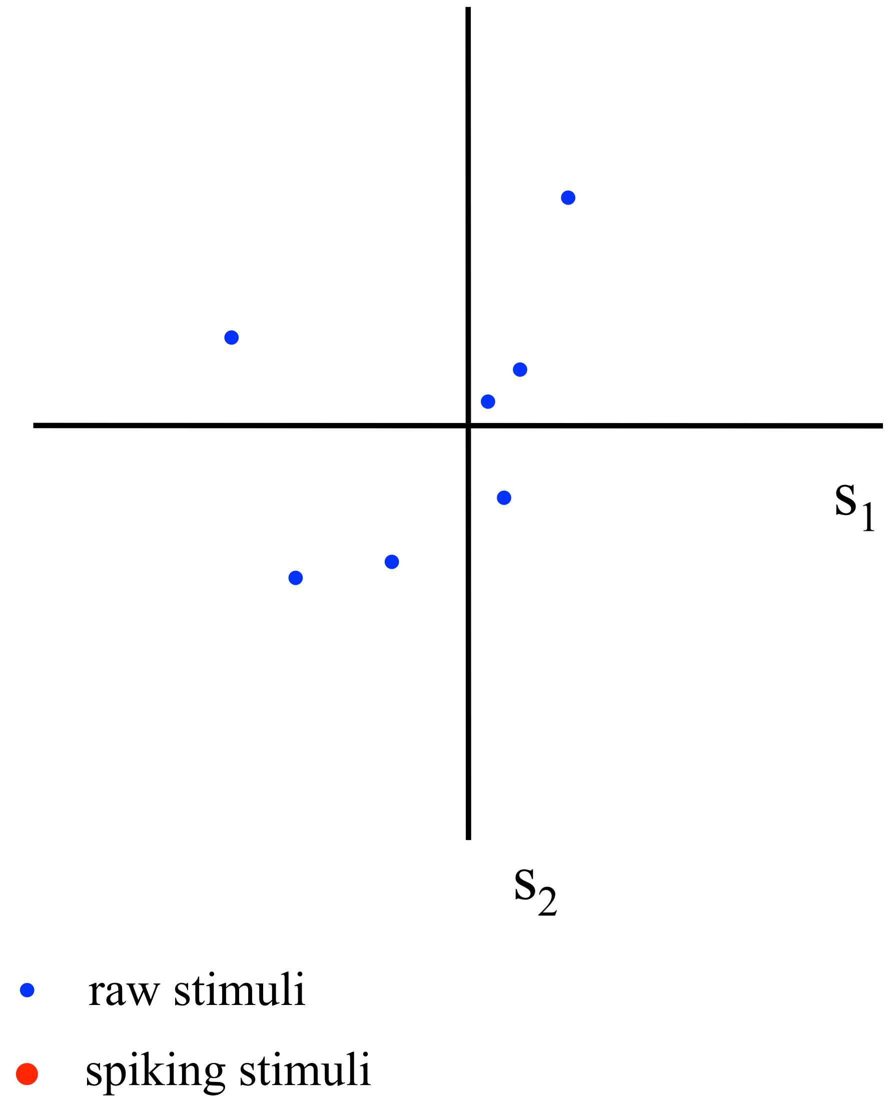




- 8 x 6 stimulus block
  = 48-dimensional vector

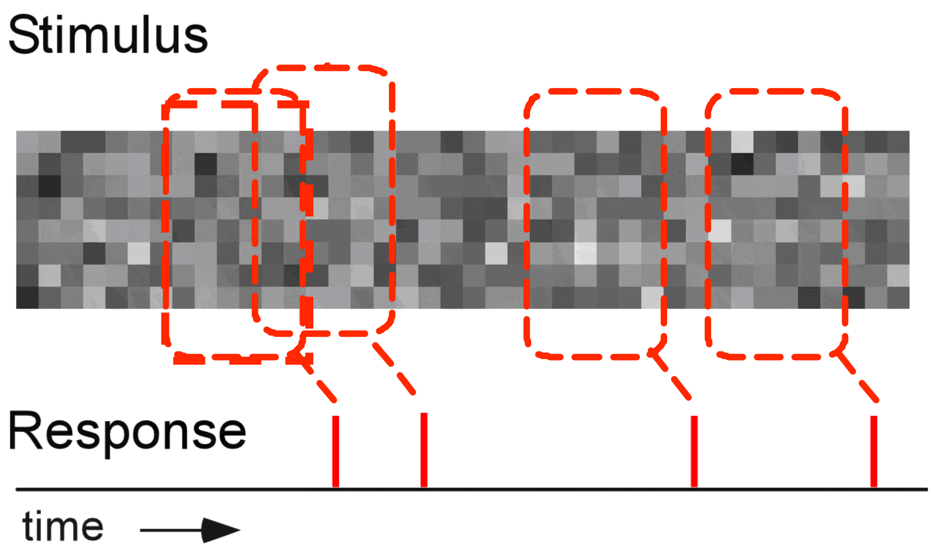
Geometric picture

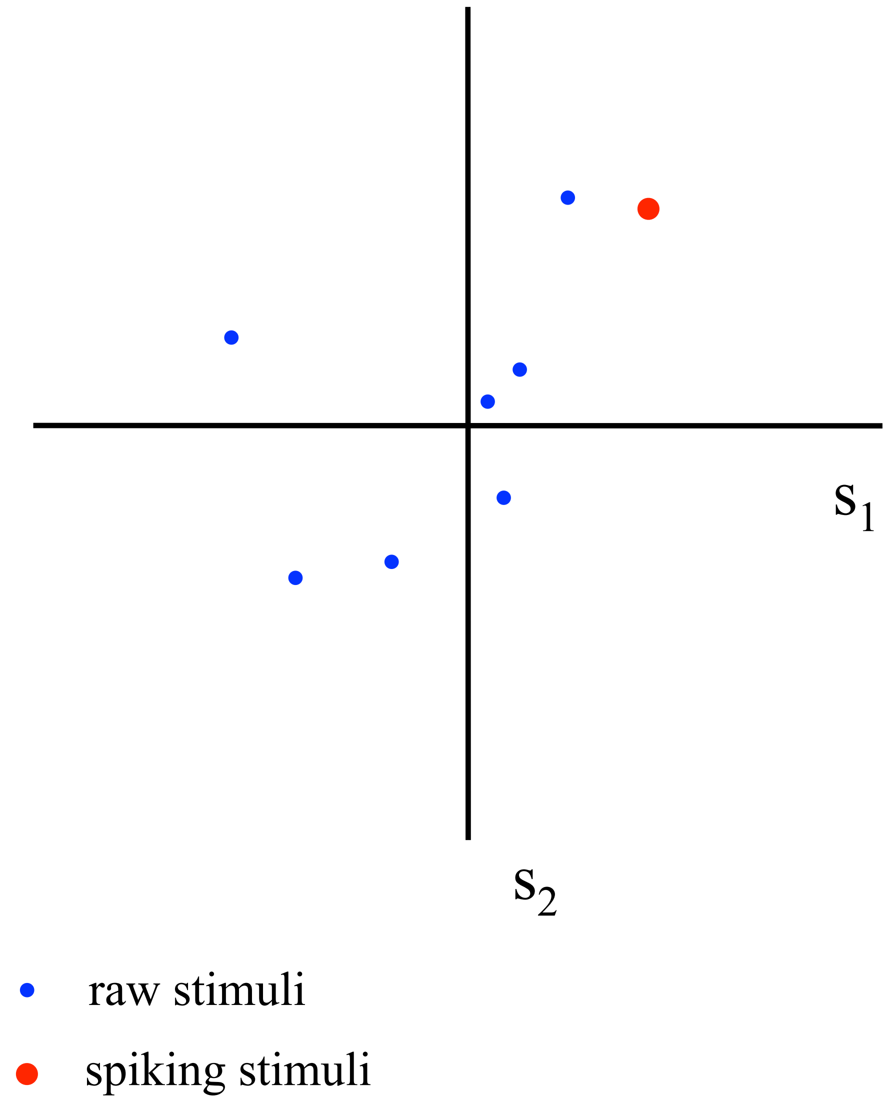




- 8 x 6 stimulus block
  = 48-dimensional vector

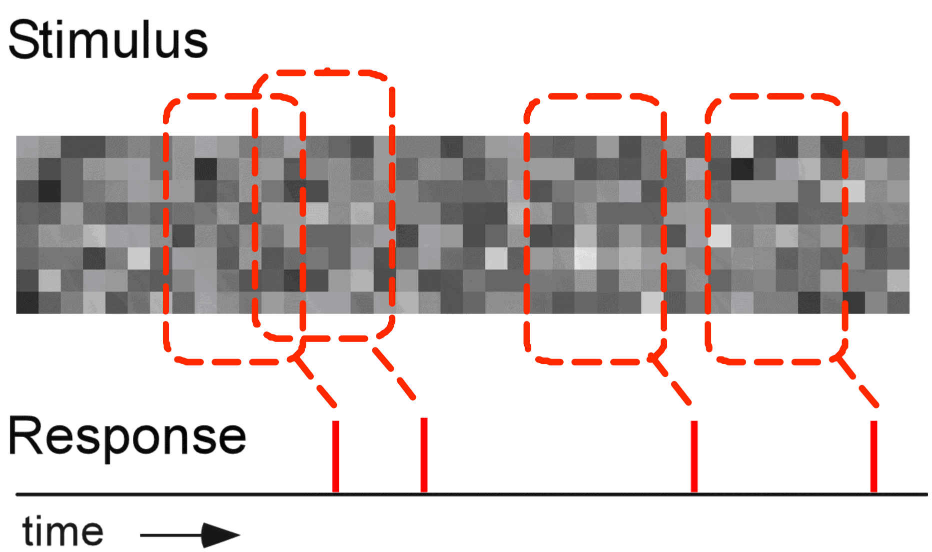
Geometric picture

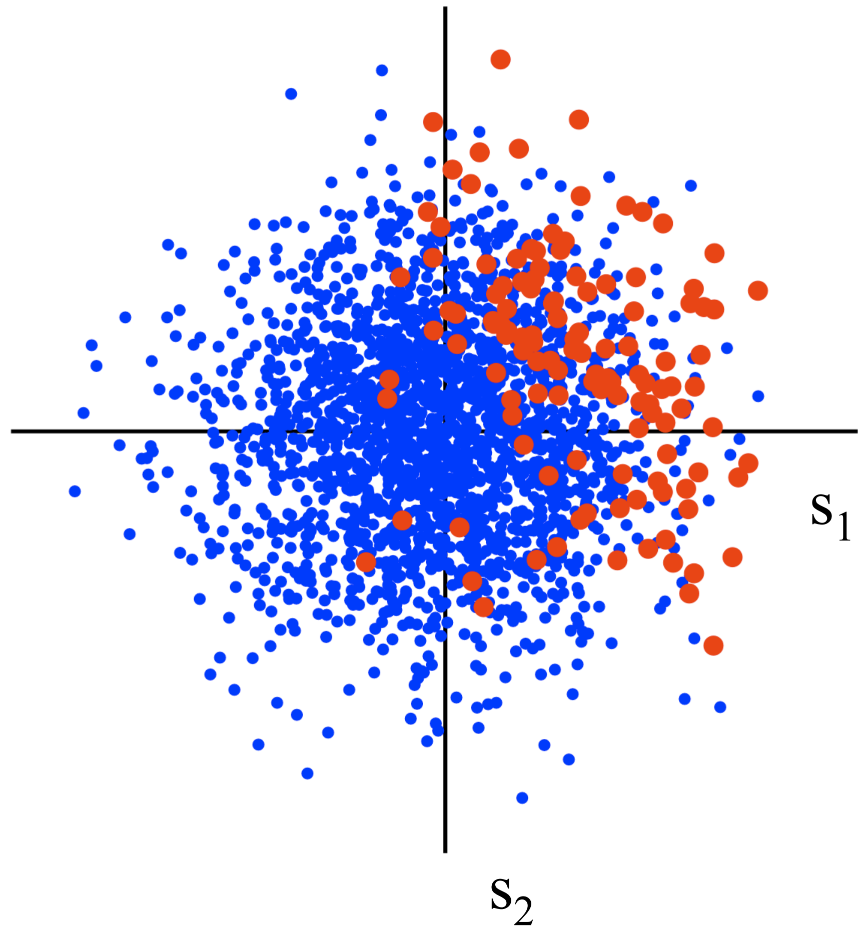




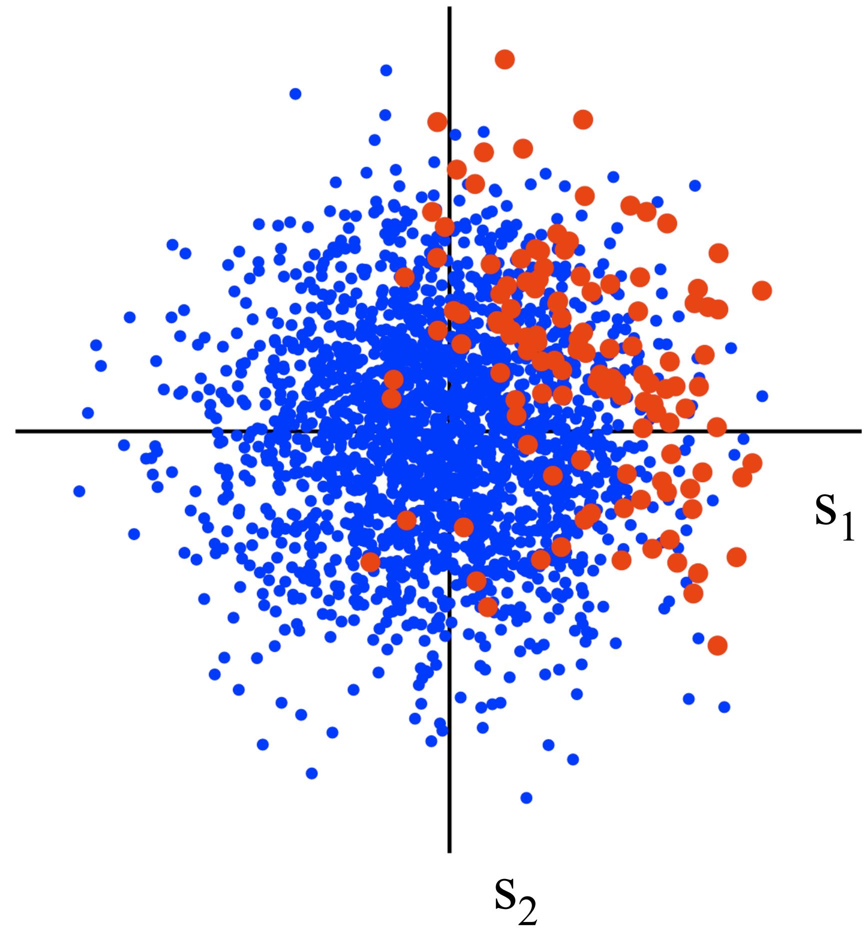
- 8 x 6 stimulus block
  = 48-dimensional vector

Geometric picture





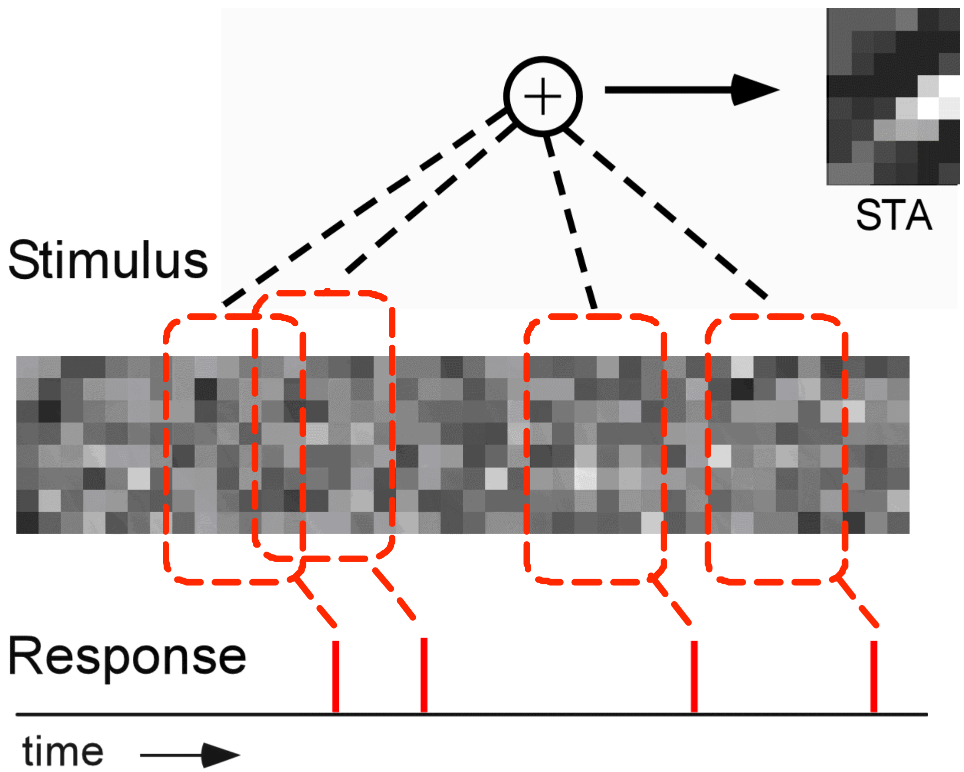
General Idea:
look for ways to capture the difference between the distribution of red dots, P(x,y=spike), and blue dots, P(x).

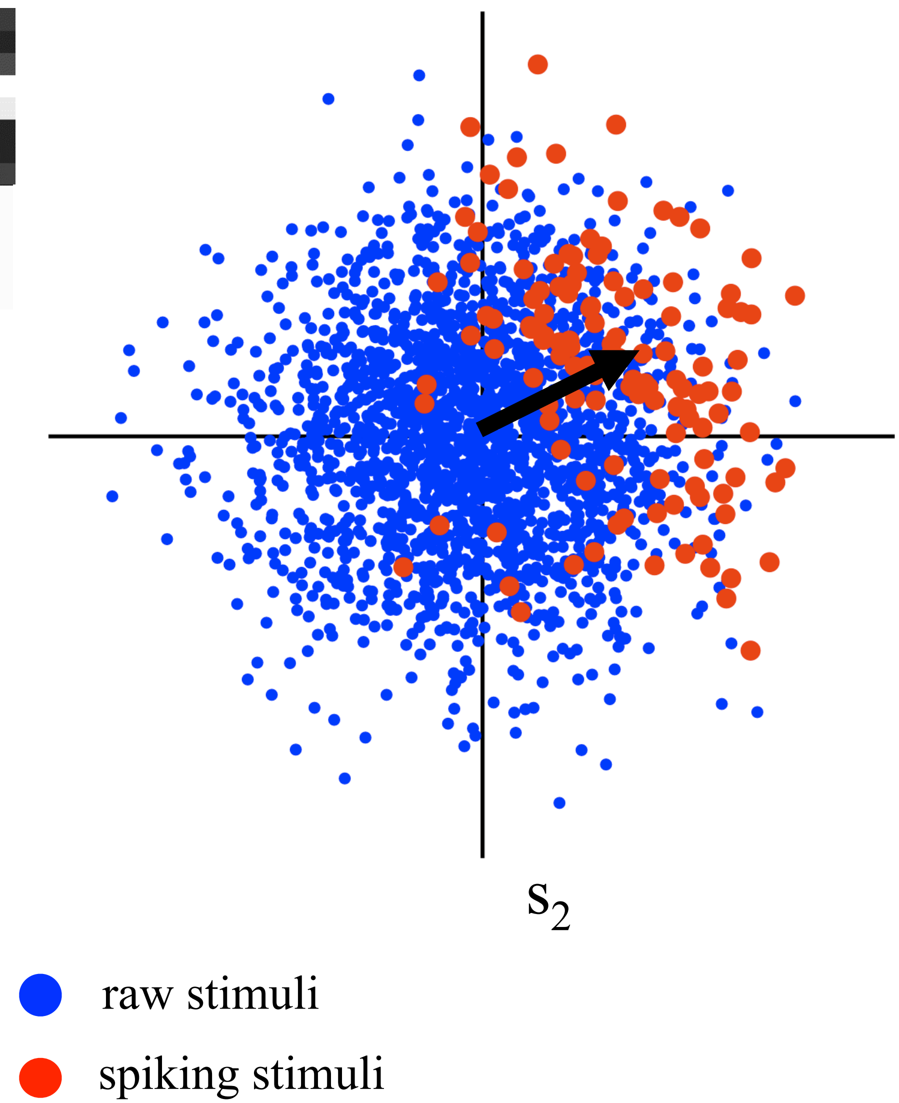


$$P(\text{spike}|x) = \frac{P(x, \text{spike})}{P(x)}$$

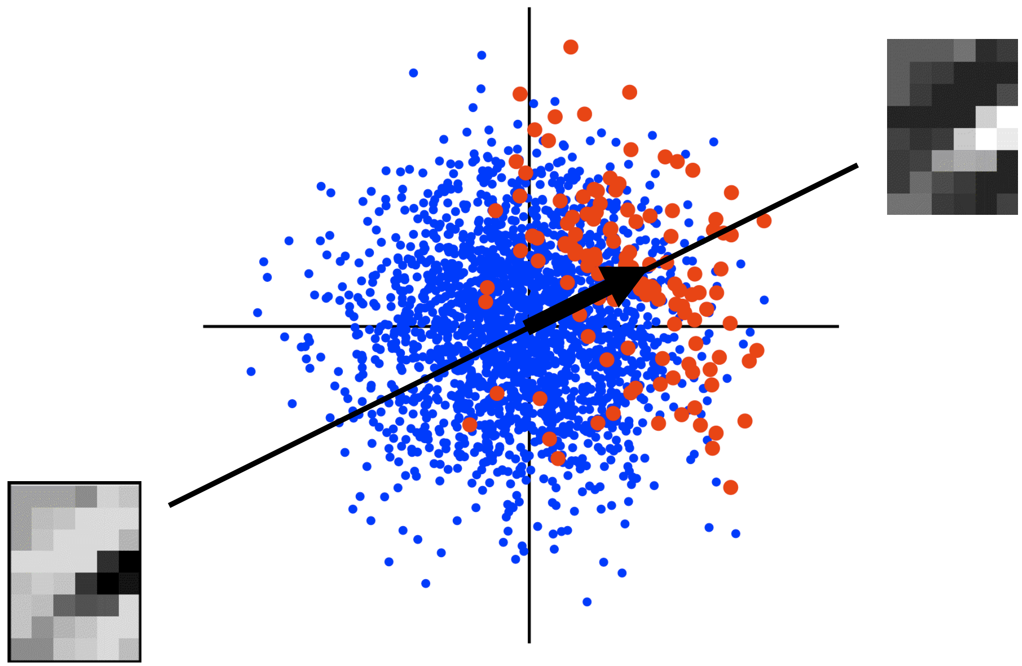
(Bayes' rule)

Computing the STA

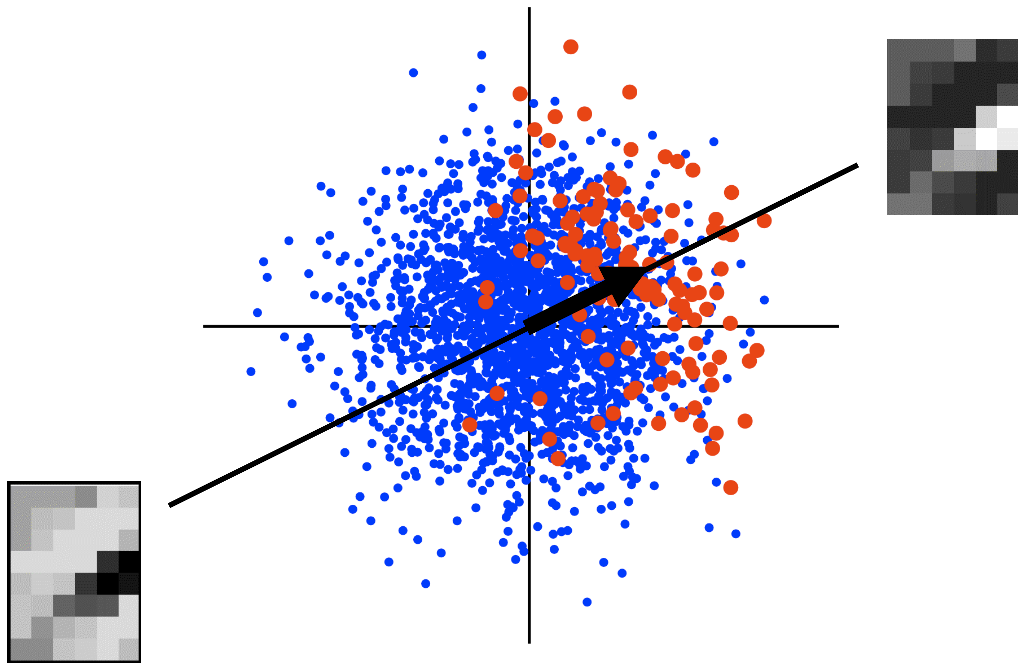




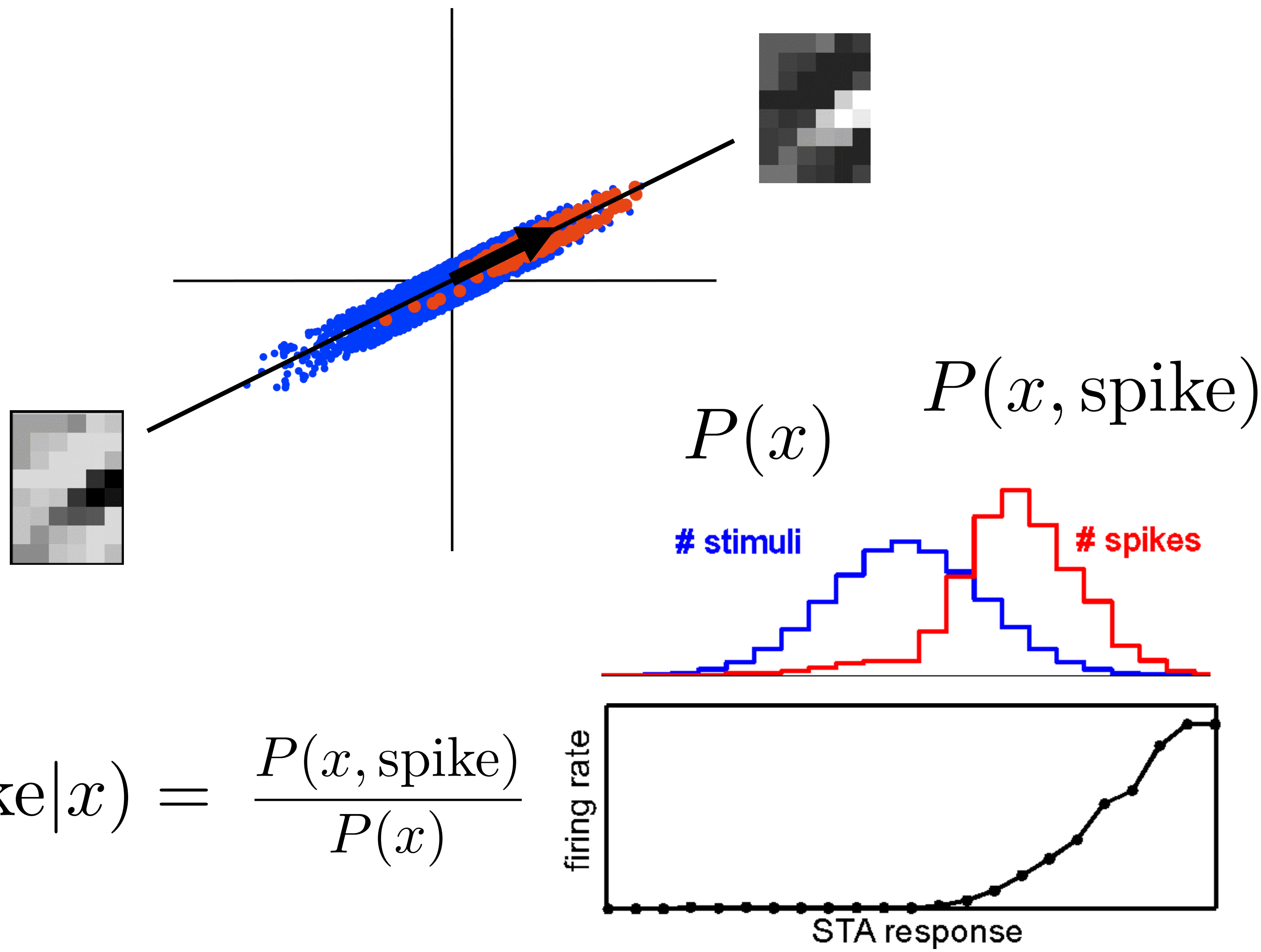
STA corresponds to a “direction” in stimulus space



Projecting onto the STA

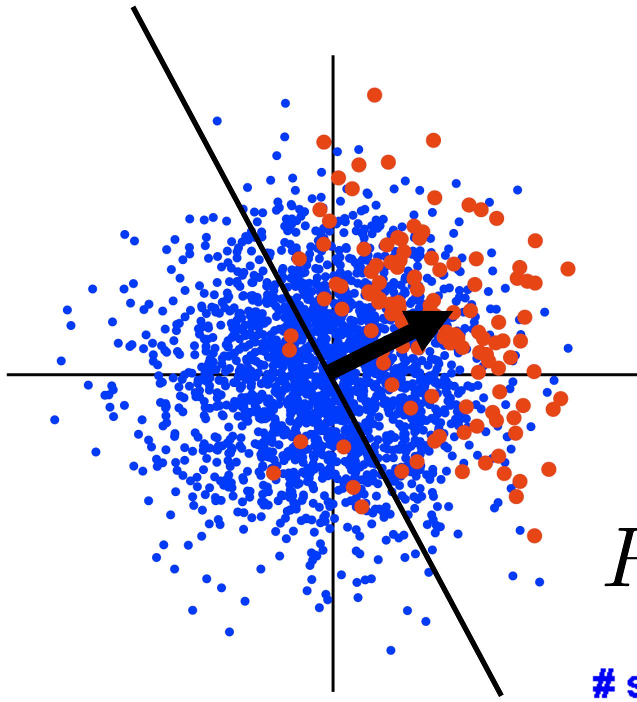


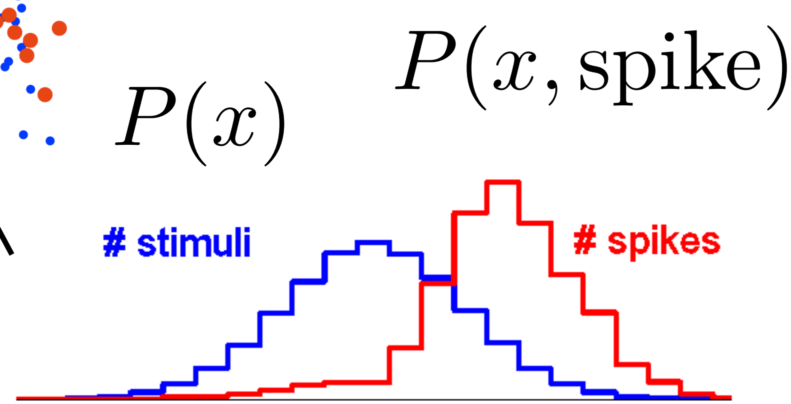
Projecting onto the STA

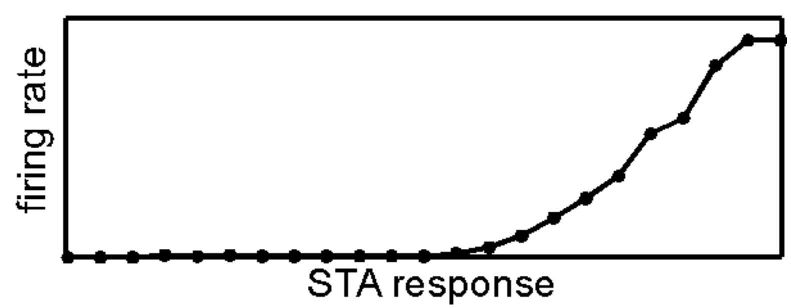


$$P(\text{spike}|x) = \frac{P(x, \text{spike})}{P(x)}$$

Projecting onto an axis orthogonal to the STA

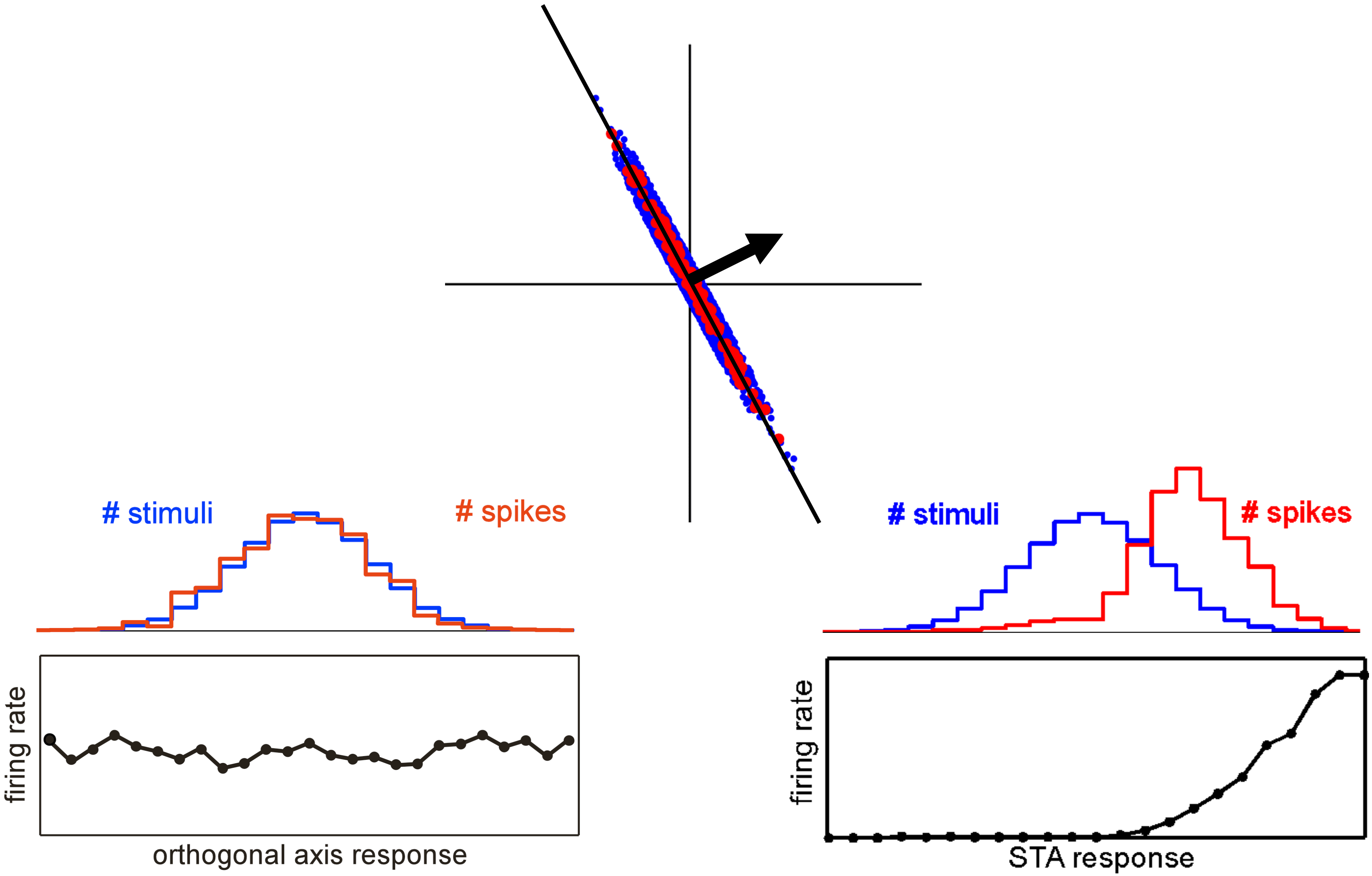




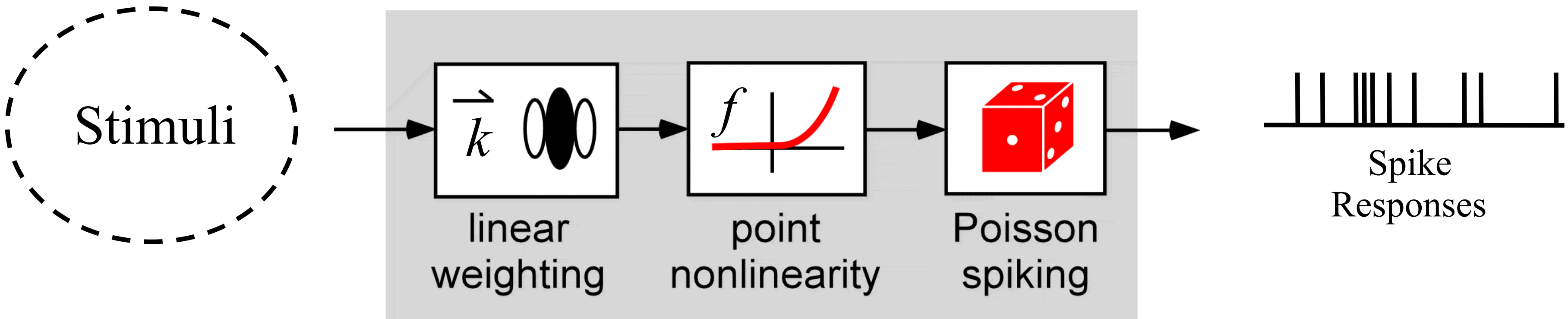


$$P(\text{spike}|x) = \frac{P(x, \text{spike})}{P(x)}$$

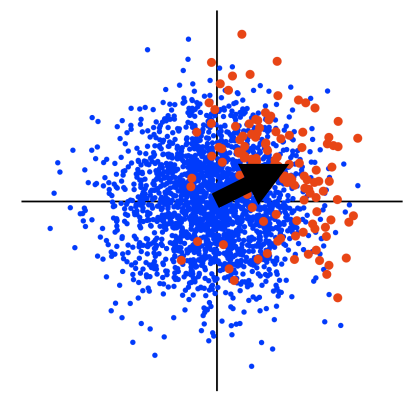
Projecting onto an axis orthogonal to the STA



LNP (Linear-Nonlinear-Poisson) cascade model

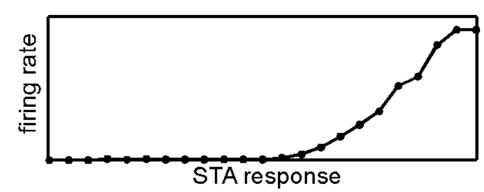


Characterization Procedure:

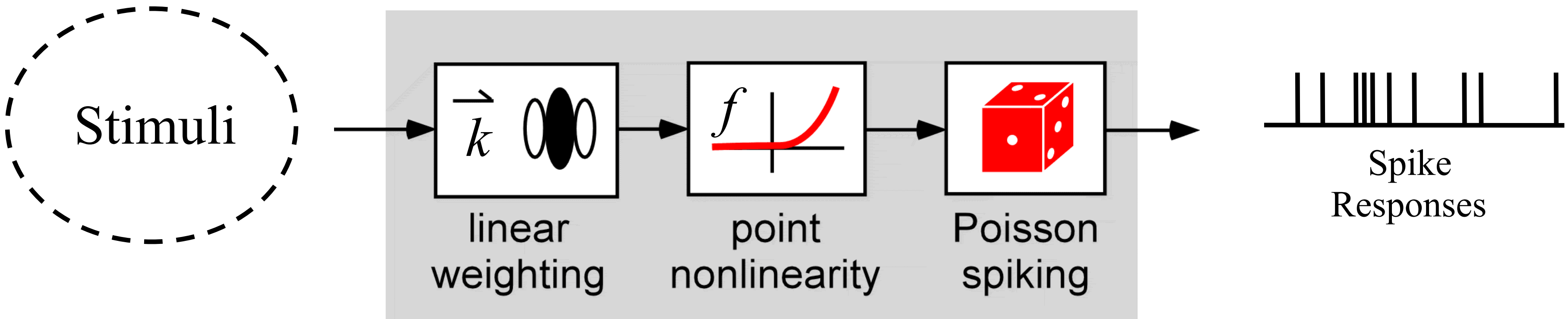


1. Identify a single dimension of stimulus space that drives the neuron's response. (STA)

2. Project stimuli onto STA, estimate pdf of spiking stimuli, and compute *f* via Bayes' rule.



LNP (Linear-Nonlinear-Poisson) cascade model



encoding distribution:

(Bernoulli)

$$P(\text{spike}|\vec{x}_t) = f(\vec{k} \cdot \vec{x}_t)$$

parameters: $\vec{k}$ stimulus filter ("receptive field")

f nonlinearity

Bussgang's Theorem

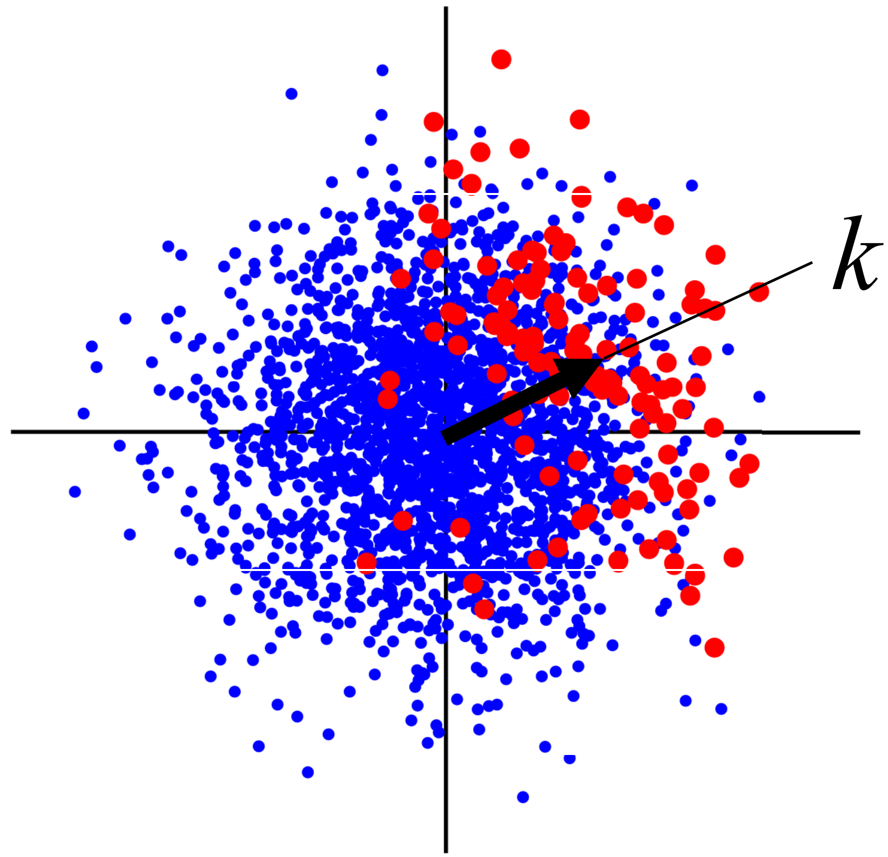
If $P(y|x) = f(k \cdot x)$, the STA is an unbiased estimator for k, if:

1. P(x) is spherically symmetric.
2. f introduces a change of mean in P(y|x).

Bussgang's Theorem

<u>Proof Intuition</u>

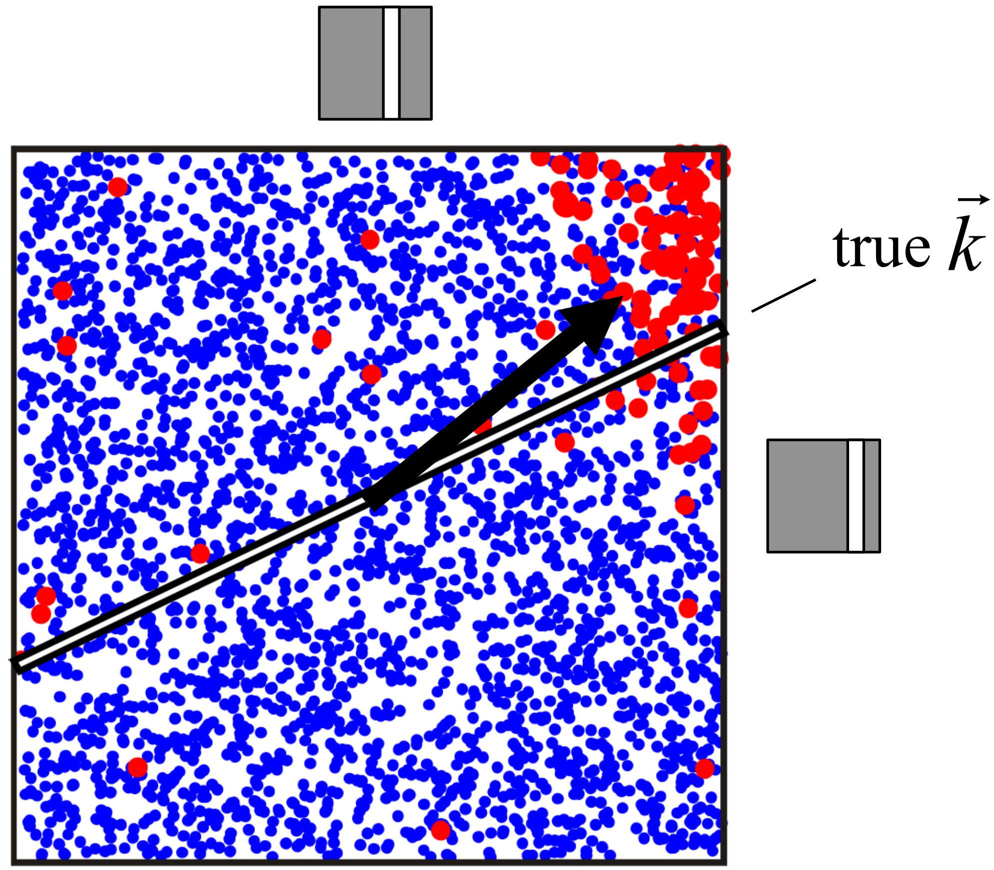
- For circularly symmetric distribution, P(stimulus) on either side of k is equal.
- These equal contributions will average out, leaving only k



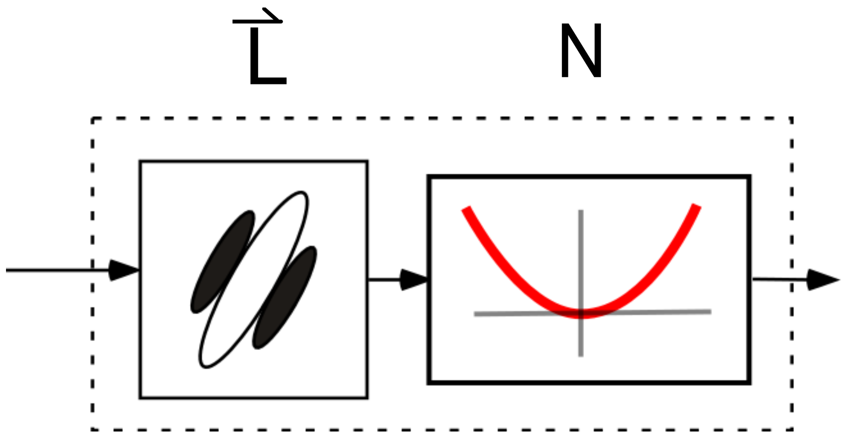
Caveat: stimulus choice is important

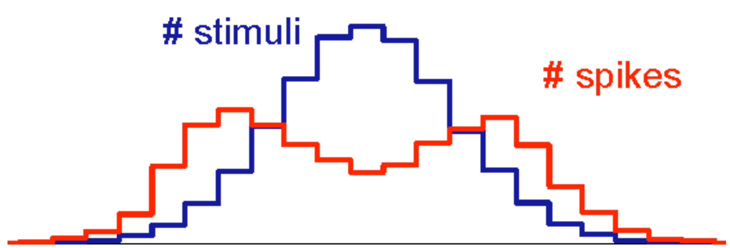
- STA requires spherical stimulus distribution

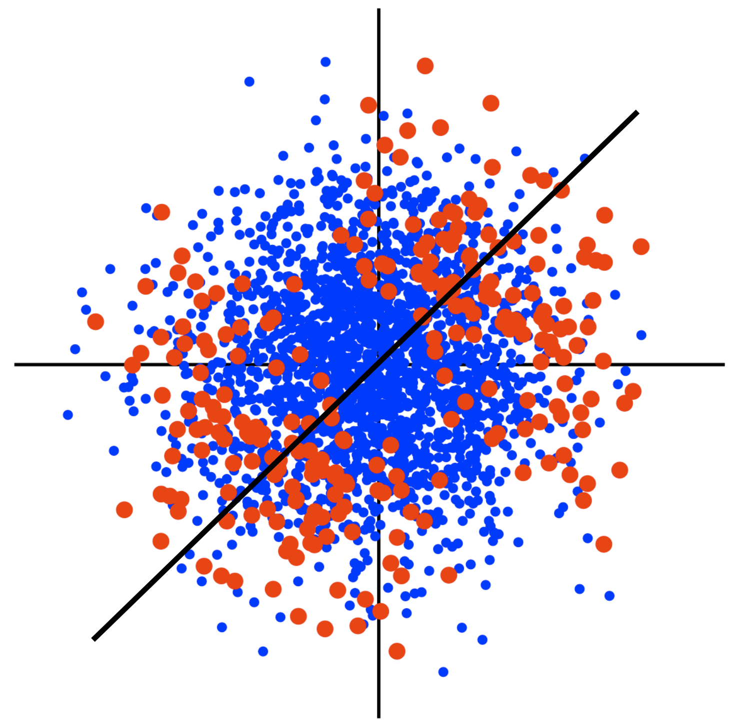
Example:
uniform noise



How else can the STA fail?

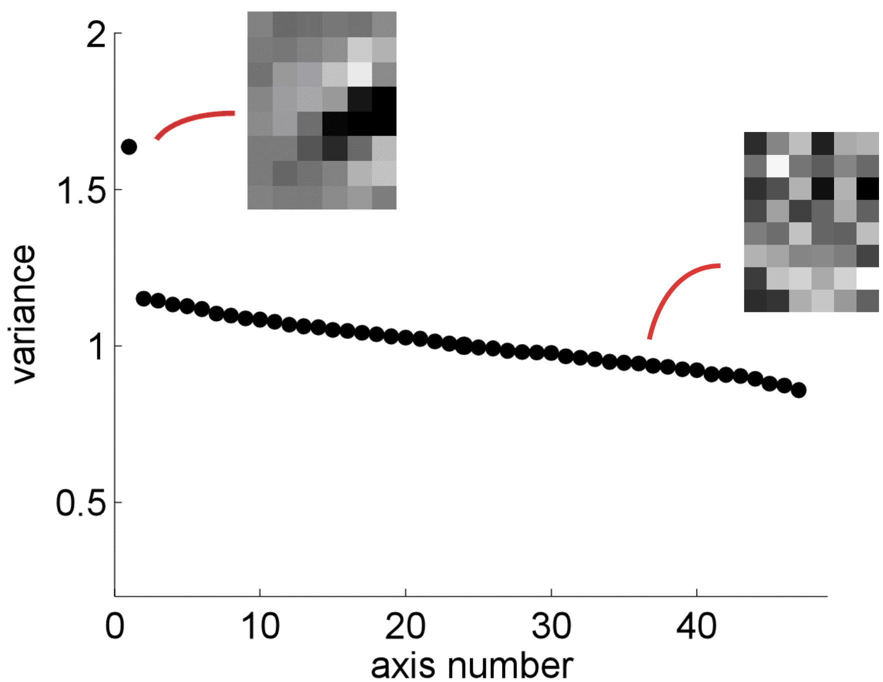


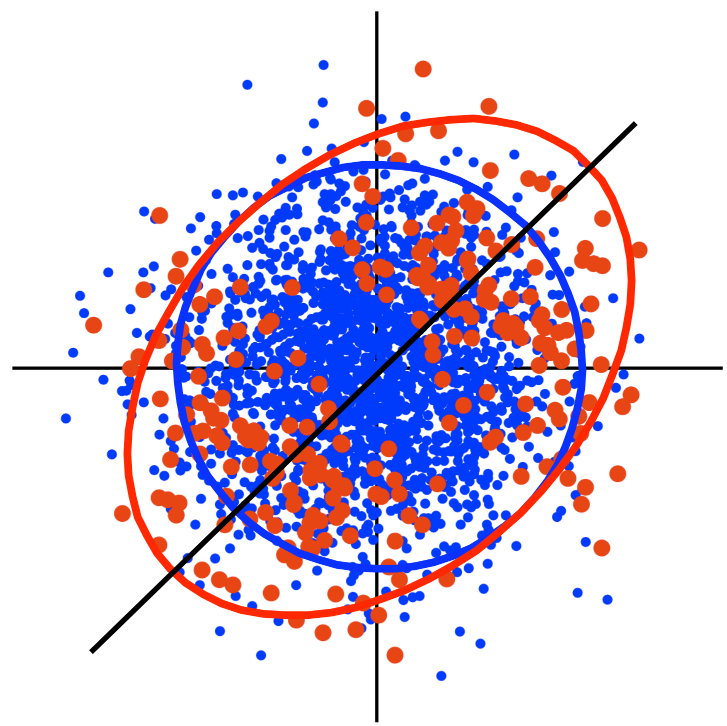




idea: look for axes with a change in variance!

Looking for changes in variance



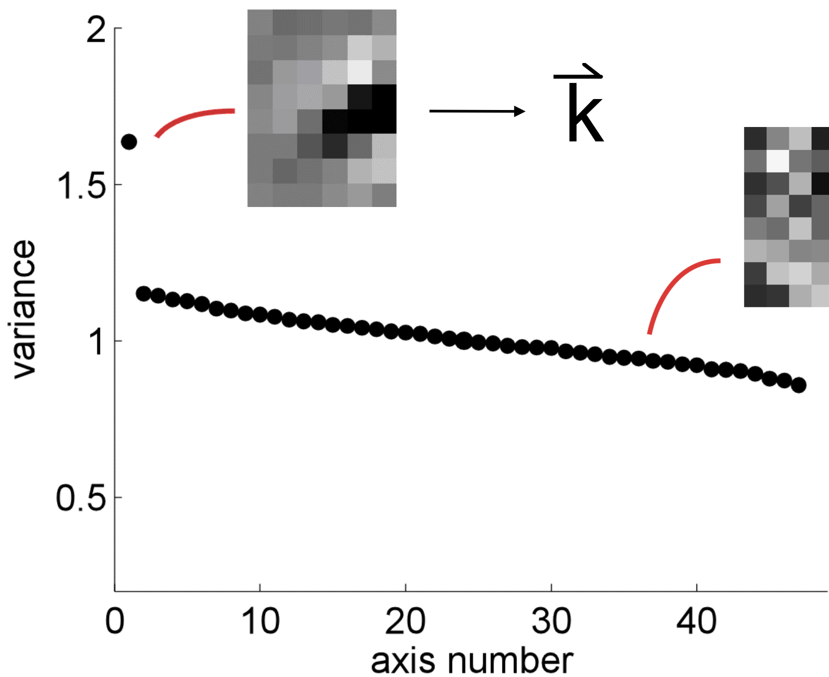


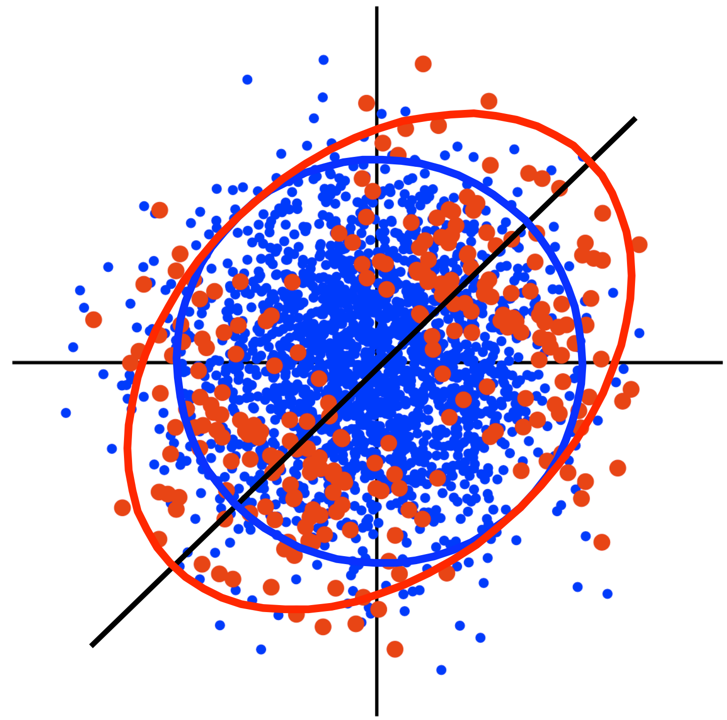
Two facts from linear algebra:

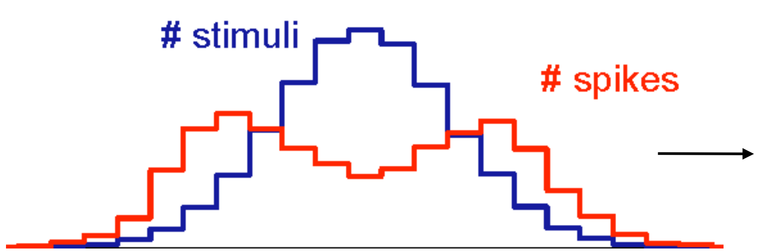
1) variance always traces out an ellipse
2) there is a readymade tool for solving this problem

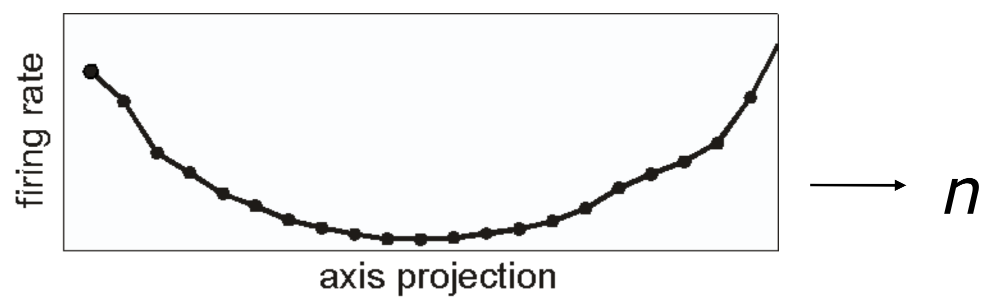
   (PCA, eigenvector decomposition, Hotelling transform)

Spike-triggered covariance (STC)

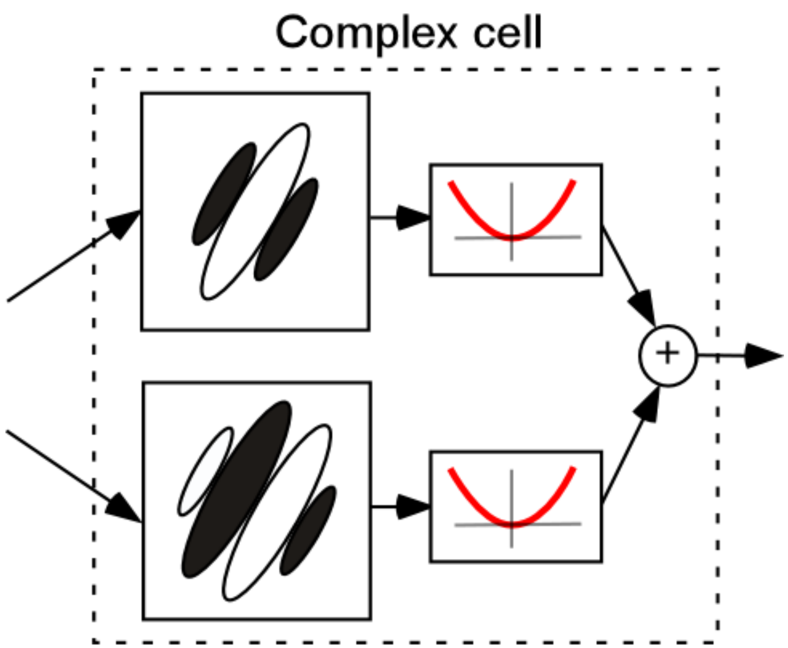




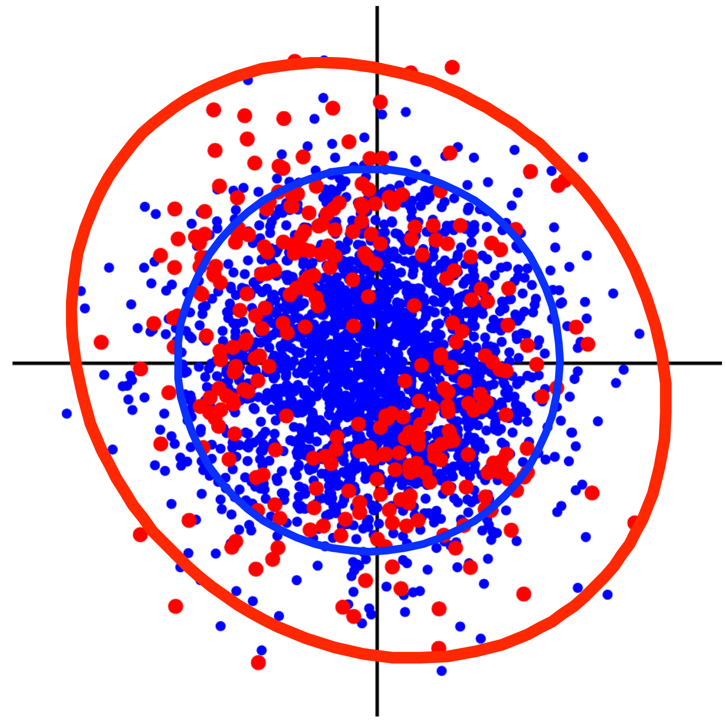




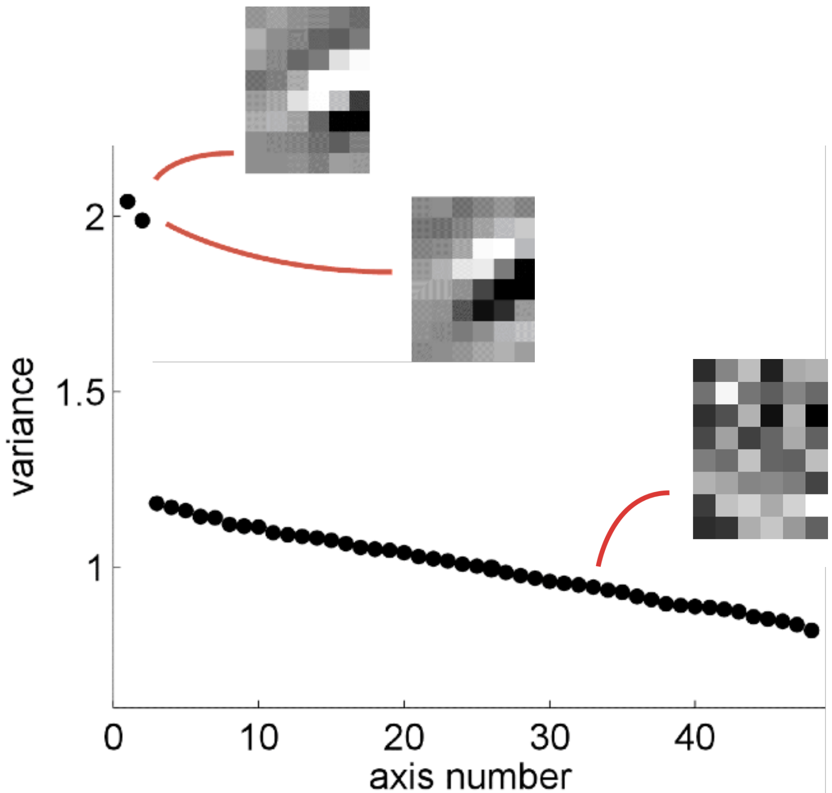
Multiple linear filters

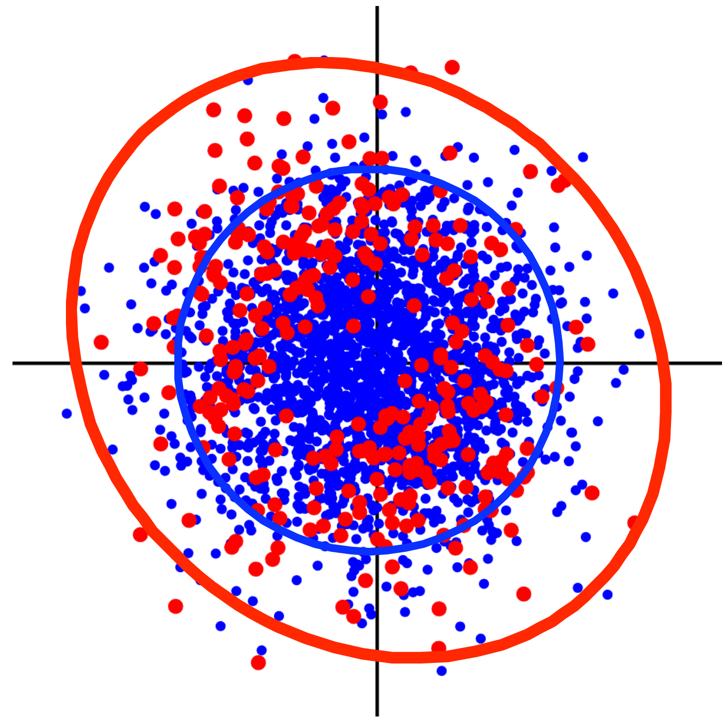


Adelson & Bergen 1985



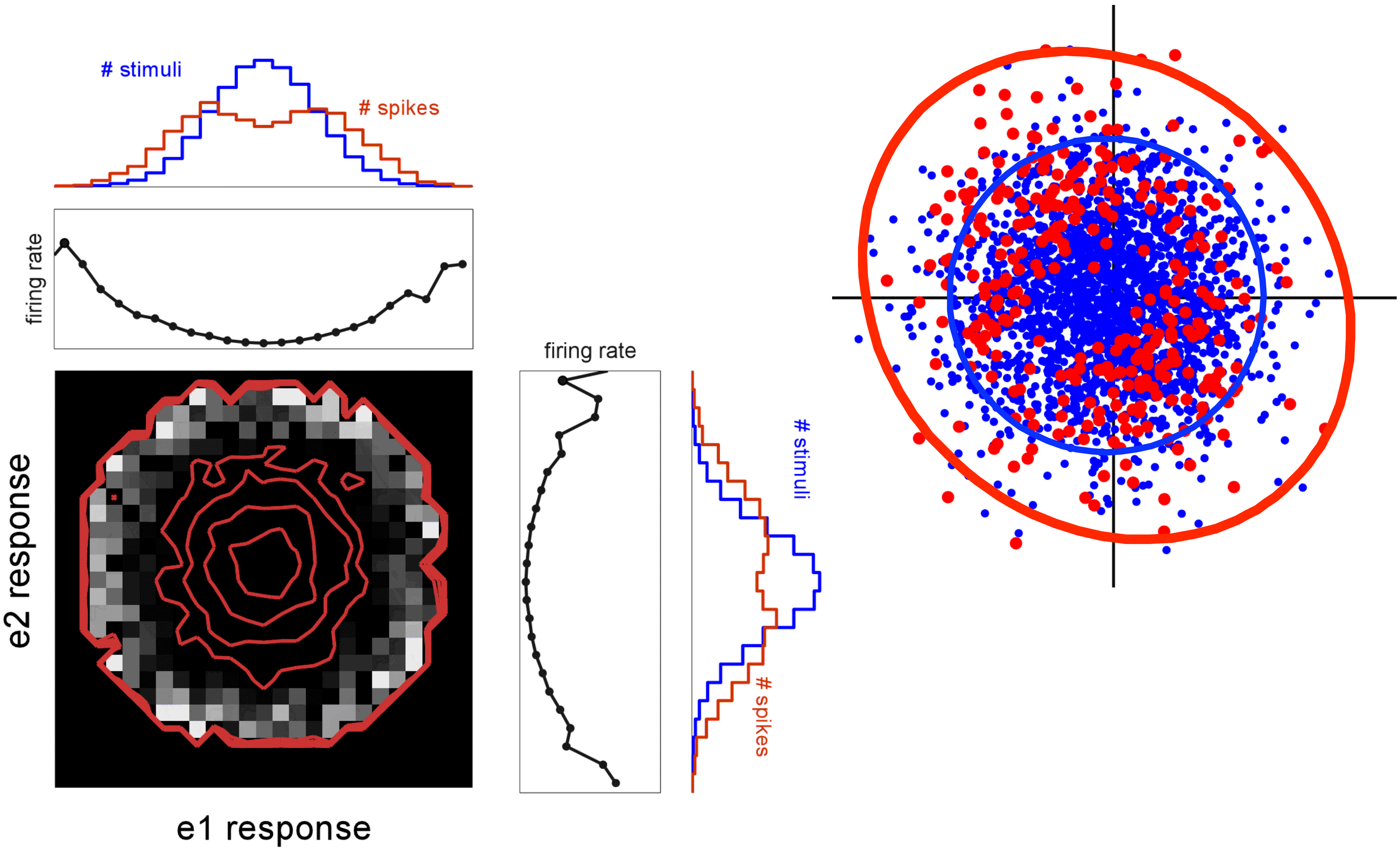
Multiple linear filters



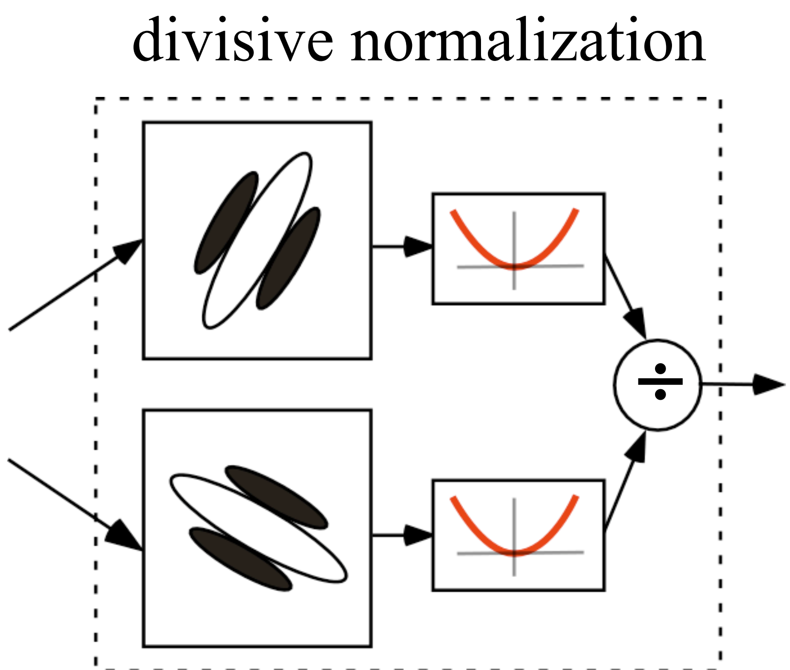


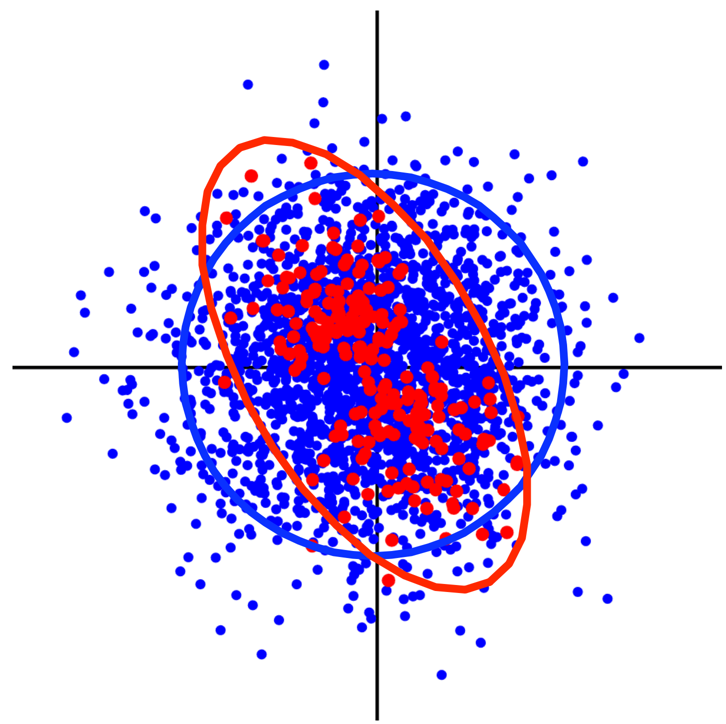
Constructing the 2D nonlinearity

$$P(\text{spike}|x) = f(k_1 \cdot x,\ k_2 \cdot x)$$

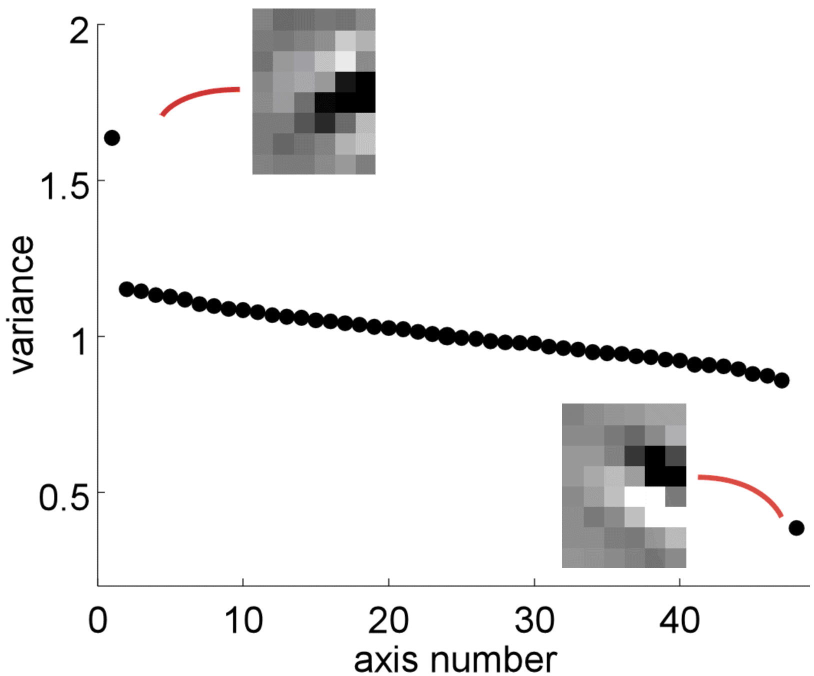


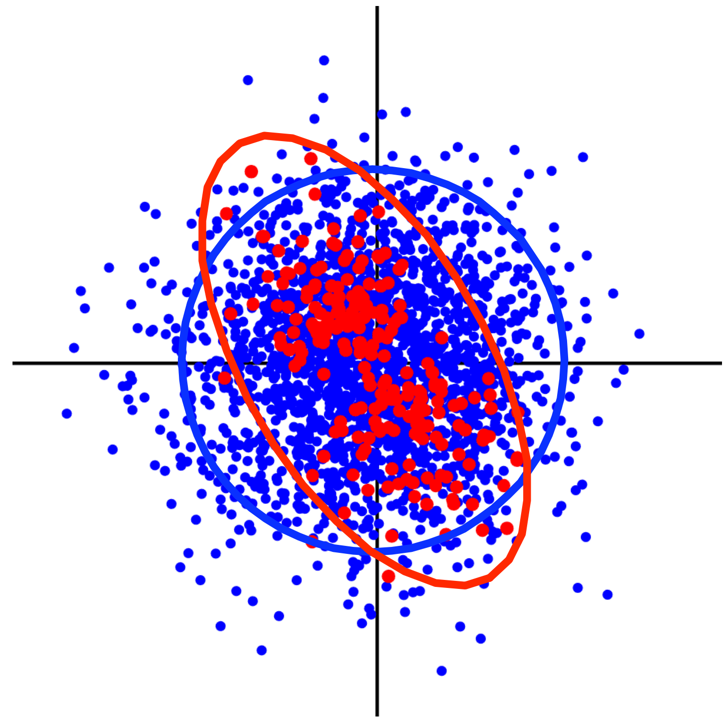
Suppressive interactions



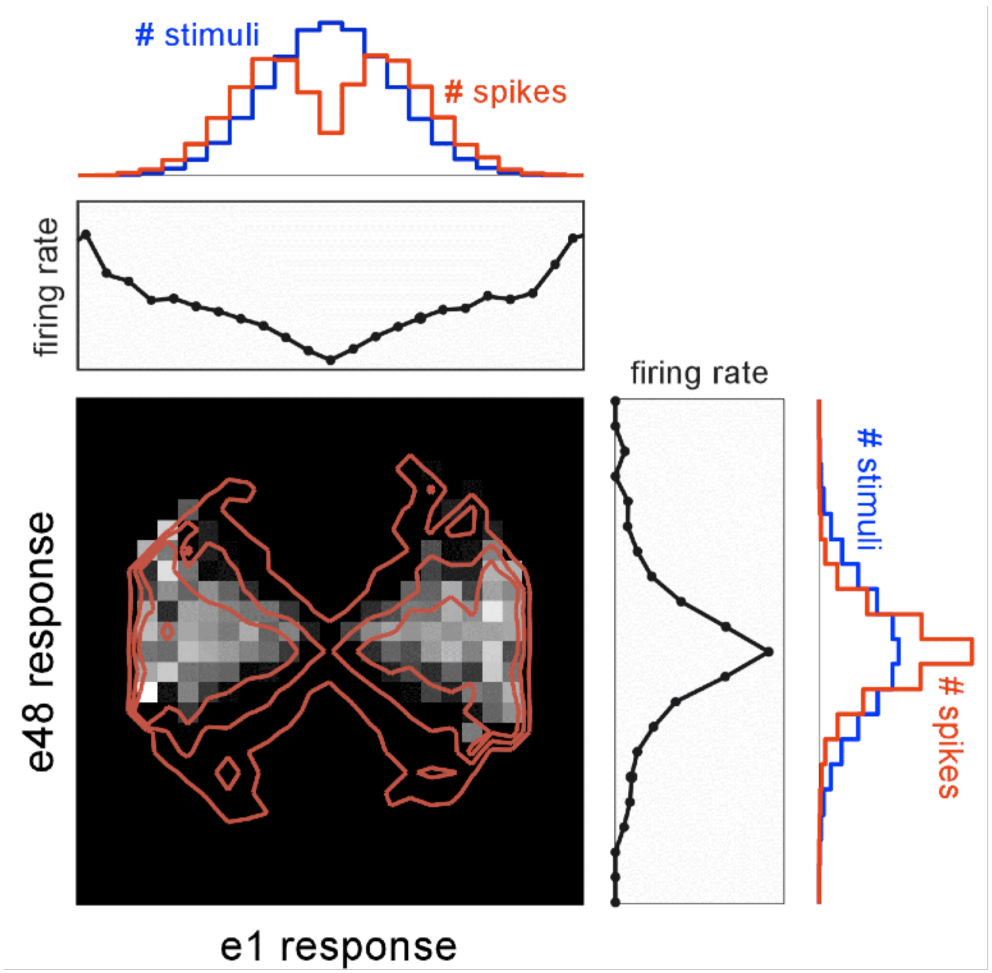


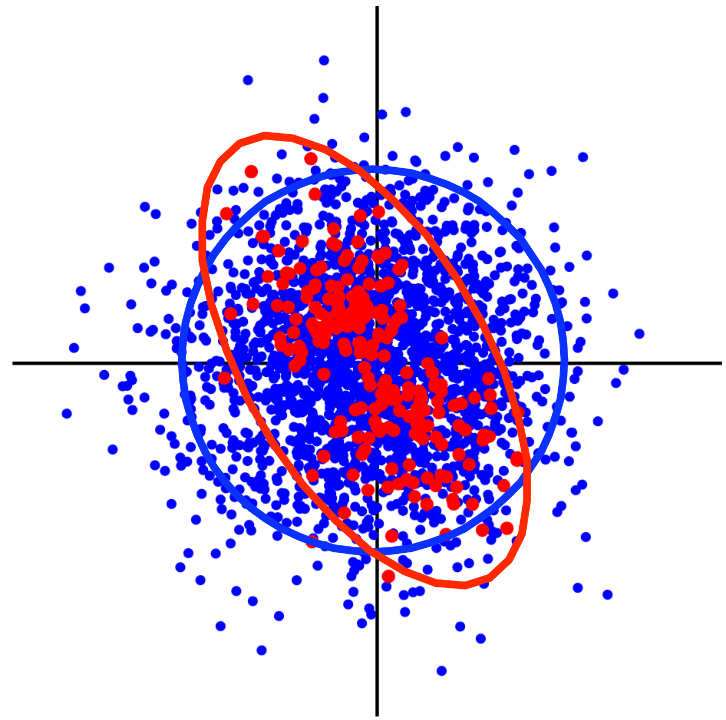
STC: suppressive axes





STC: suppressive axes





Summary: spike-triggered covariance analysis

1. Compute covariance of spike triggered stimuli

$$A = \frac{1}{n} \sum x_i x_i^T$$

2. Compute eigenvectors/eigenvalues of A

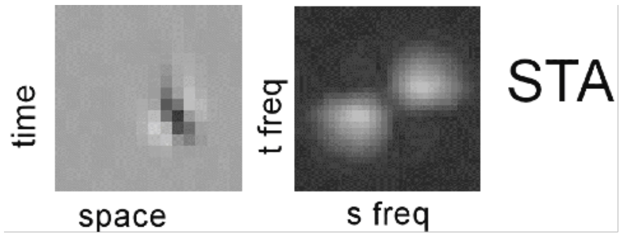
3. Eigenvalues bigger/smaller than 1 indicate stimulus axes along which the response is excited/suppressed

4. Construct model of multi-dim nonlinearity f

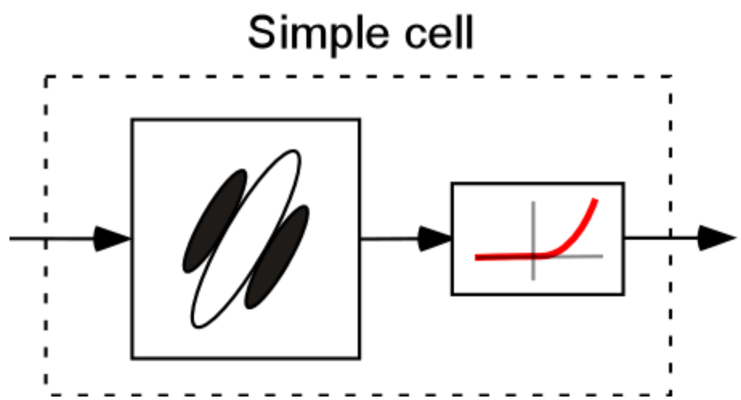
Examples:

STC applied to neural data

V1 simple cell



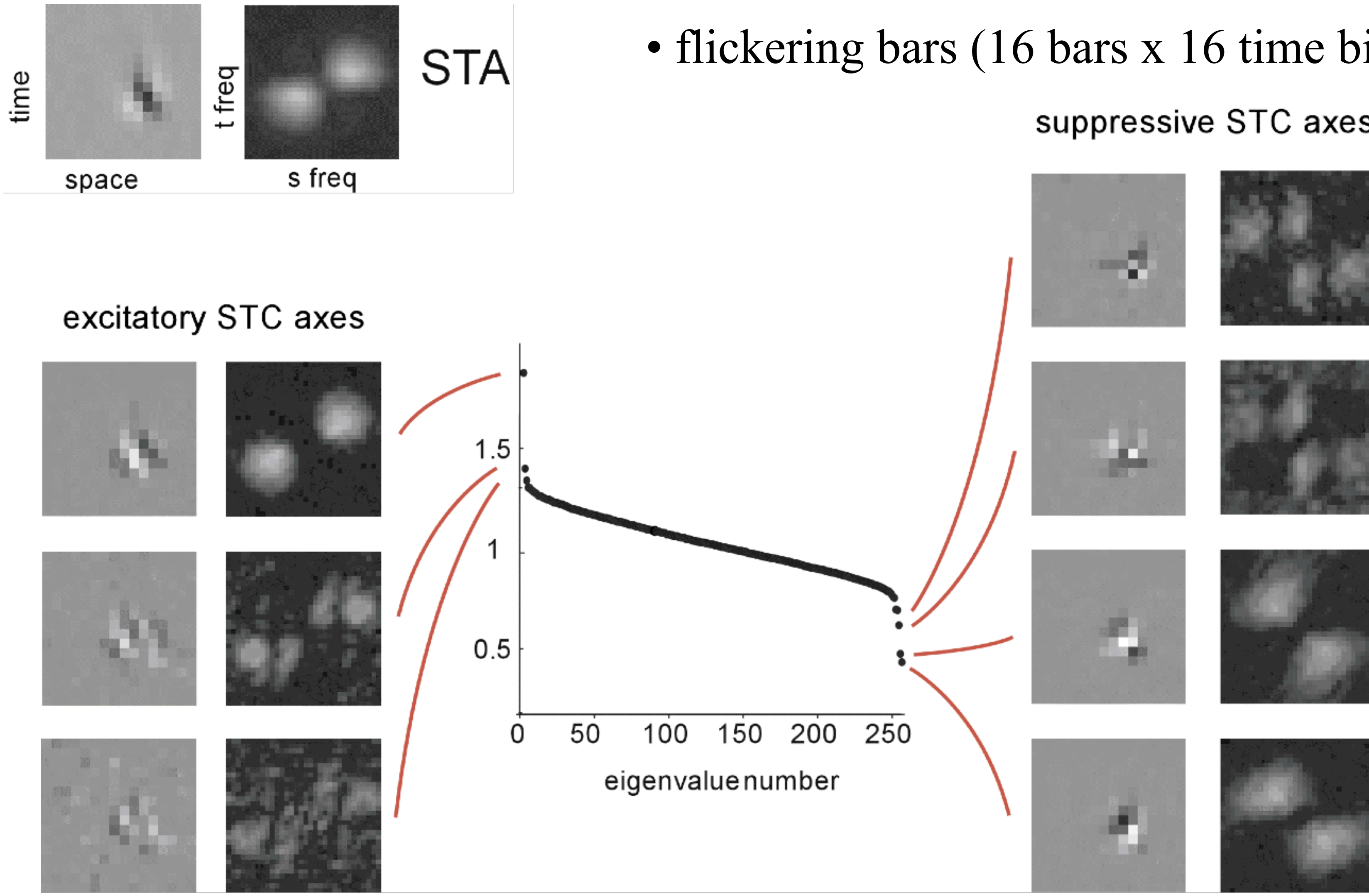
• flickering bars (16 bars x 16 time bins)



Rust, Schwartz, Movshon, Simoncelli (Neuron 2005)

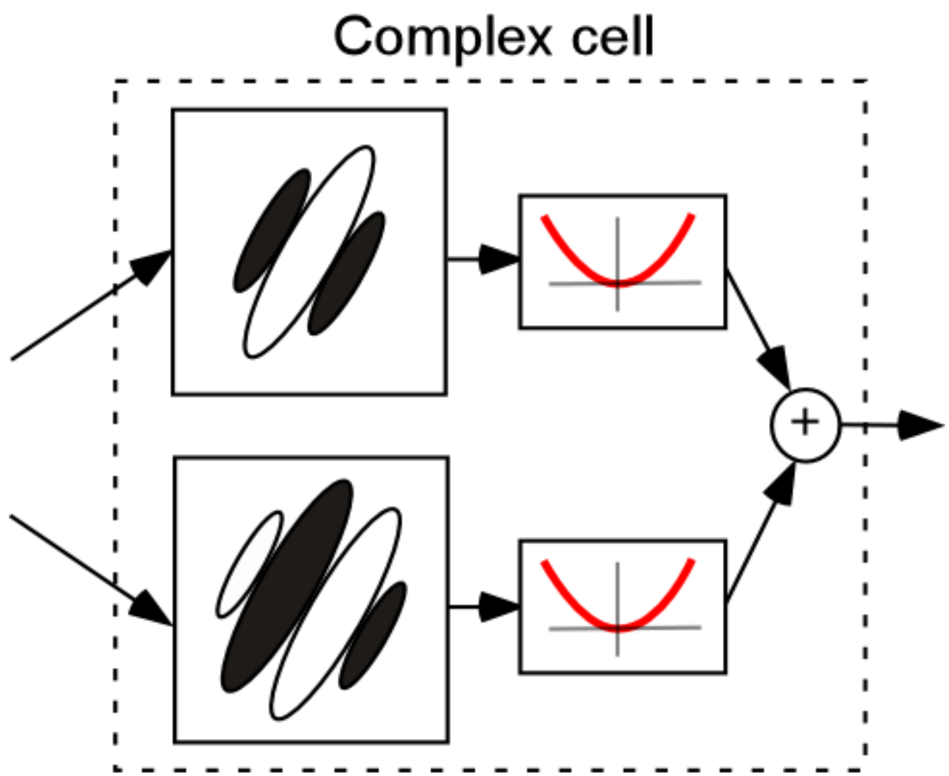
V1 simple cell

• flickering bars (16 bars x 16 time bins)



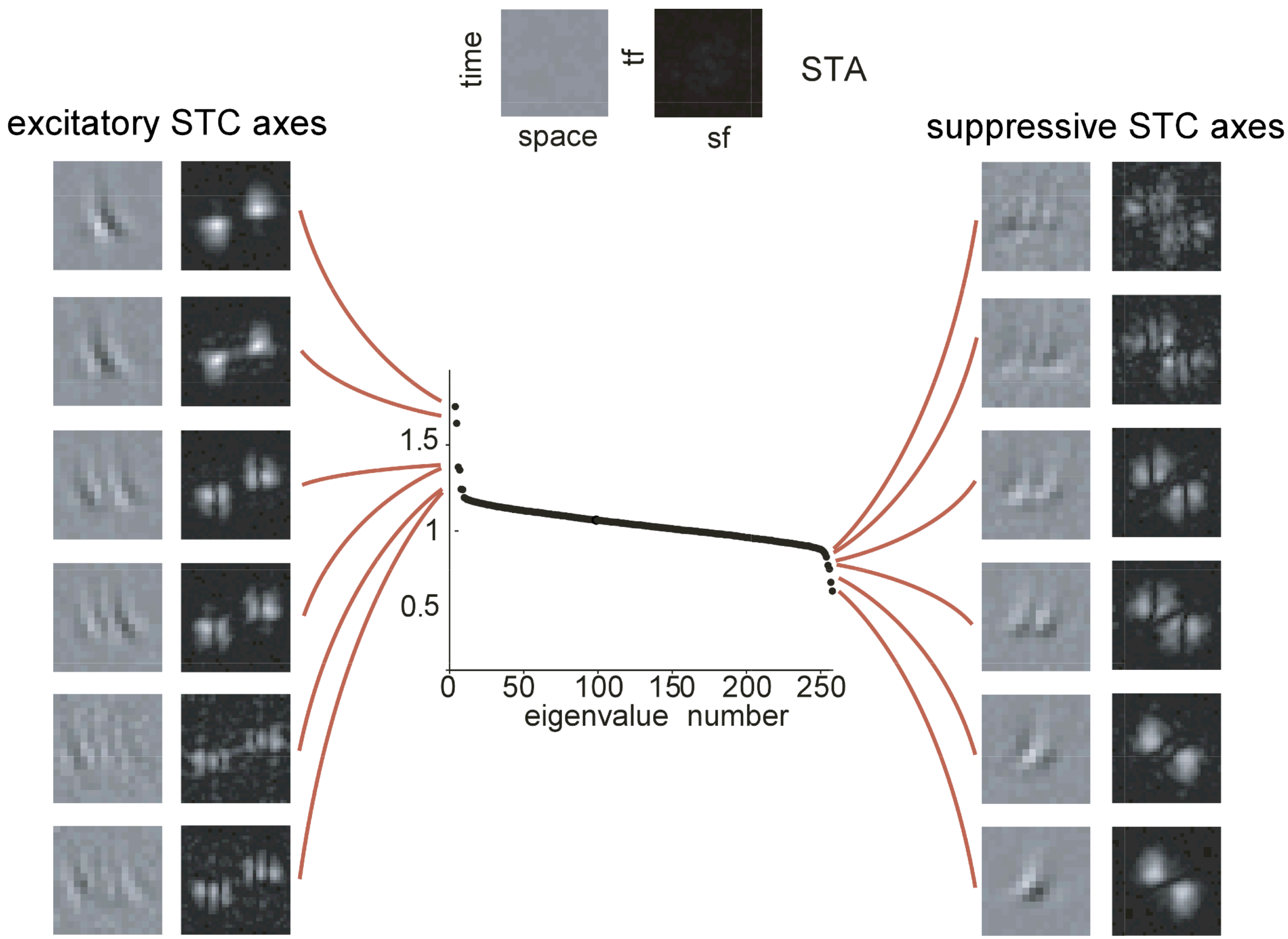
Rust, Schwartz, Movshon, Simoncelli (Neuron 2005)

V1 complex cell, standard model



Adelson & Bergen 1985

V1 complex cell

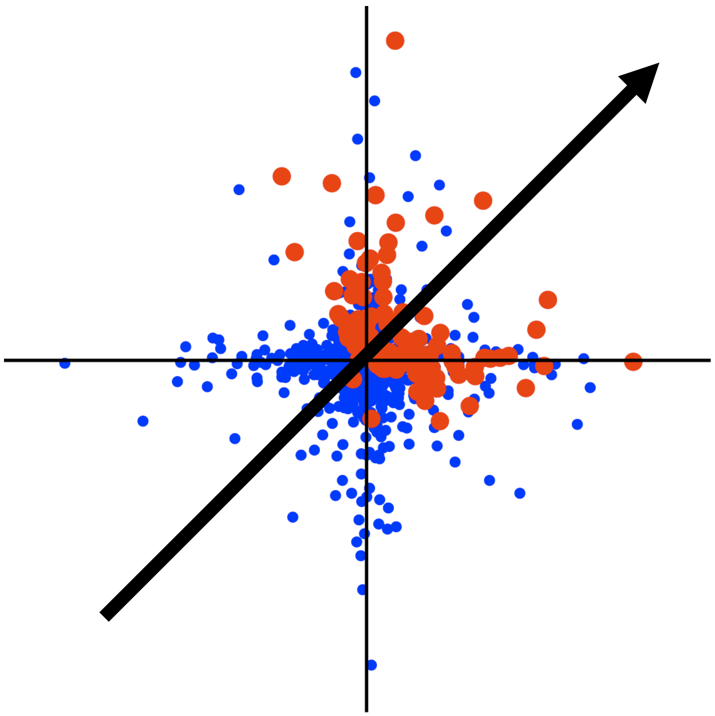


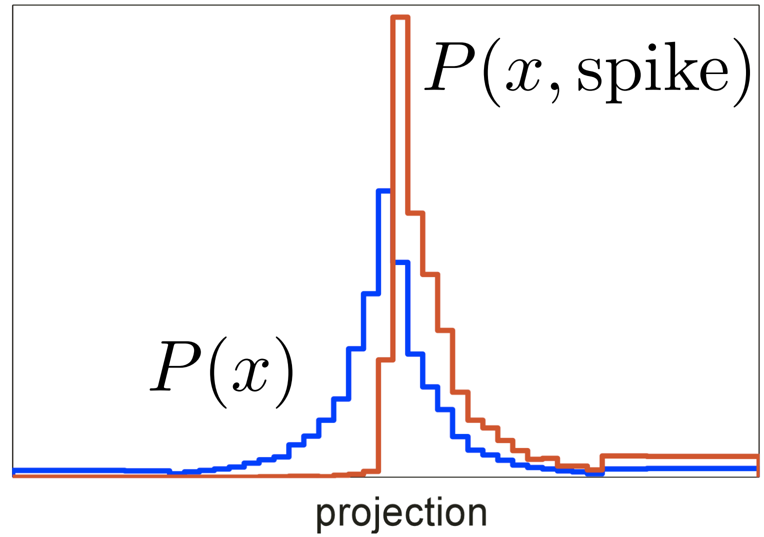
Rust et al 05

conclusion:

Standard models of neural response may underestimate the number of dimensions in which neurons compute their responses.

Beyond mean and variance: other techniques for finding stimulus features that affect response



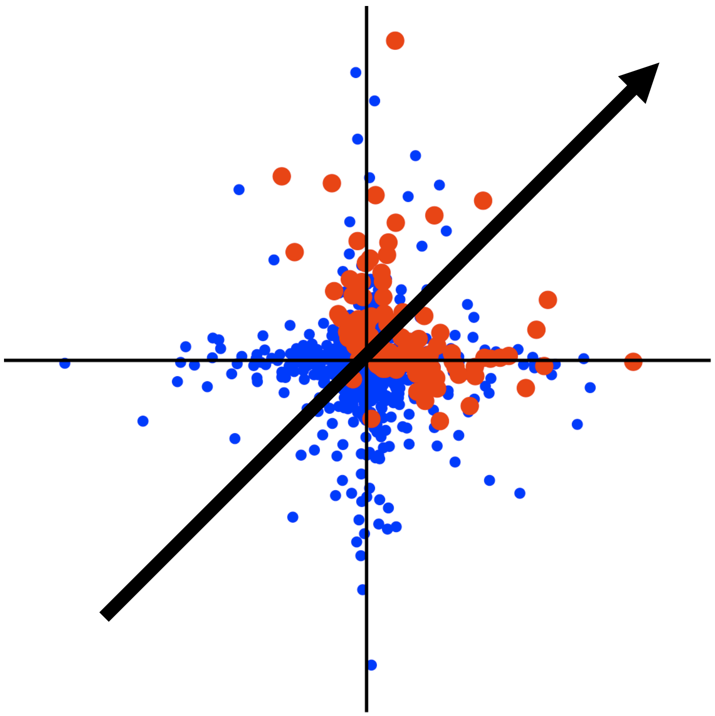


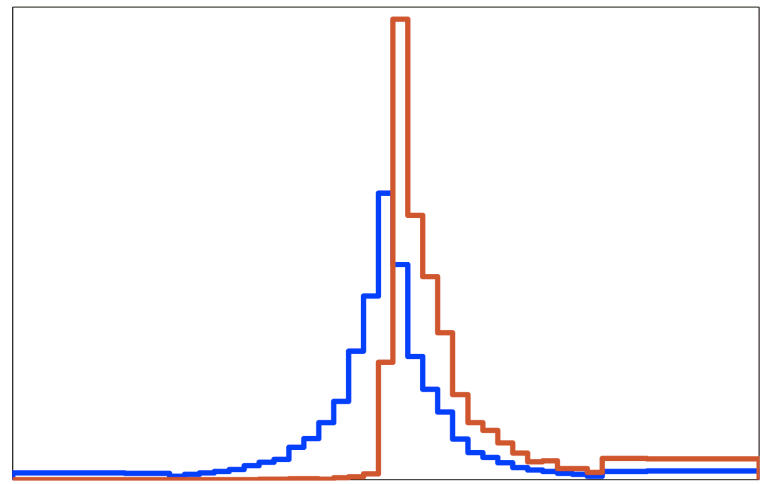
<u>Information-theoretic approach</u>: Search for axis maximizing the diff. between the two distributions

$$\text{rate} = \frac{P(x, \text{spike})}{P(x)}$$

$$\underset{k}{\text{maximize}} \quad KL[\ P(k \cdot x, \text{spike}) \mid P(k \cdot x)\ ]$$

Beyond mean and variance: other techniques for finding stimulus features that affect response





<u>Information-theoretic approach</u>: Search for axis maximizing the diff. between the two distributions

- computationally intensive, but
- doesn't require spherical symmetry

- Paninski '03
- Sharpee, Rust & Bialek '03

Summary (last 2 lectures):

1. Neural encoding problem: $P(y|x)$
2. Classical approach: parametric stimuli
3. Wiener Kernels: polynomial models
4. Linear-Nonlinear-Poisson cascade models: fit using dimensionality-reduction techniques (STA, STC)

Open problems:

1. Better models of the nonlinearity.
2. Characterizing neurons deeper in sensory processing pathway
3. Incorporating adaptative effects