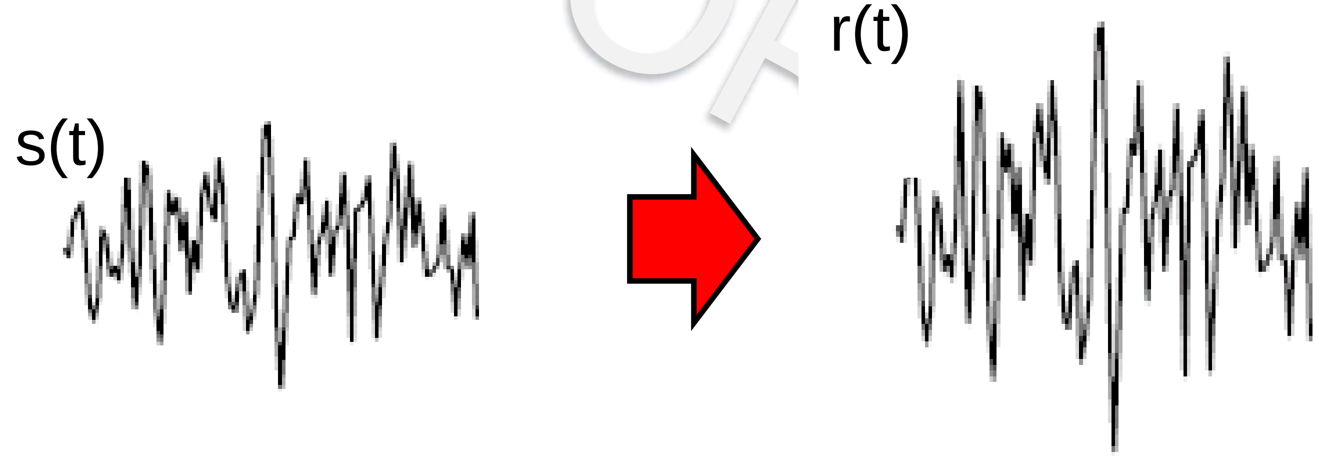
Constructing response models

$\boxed{\text{P(response | stimulus)} \rightarrow r(t) \text{ given a stimulus } s}$

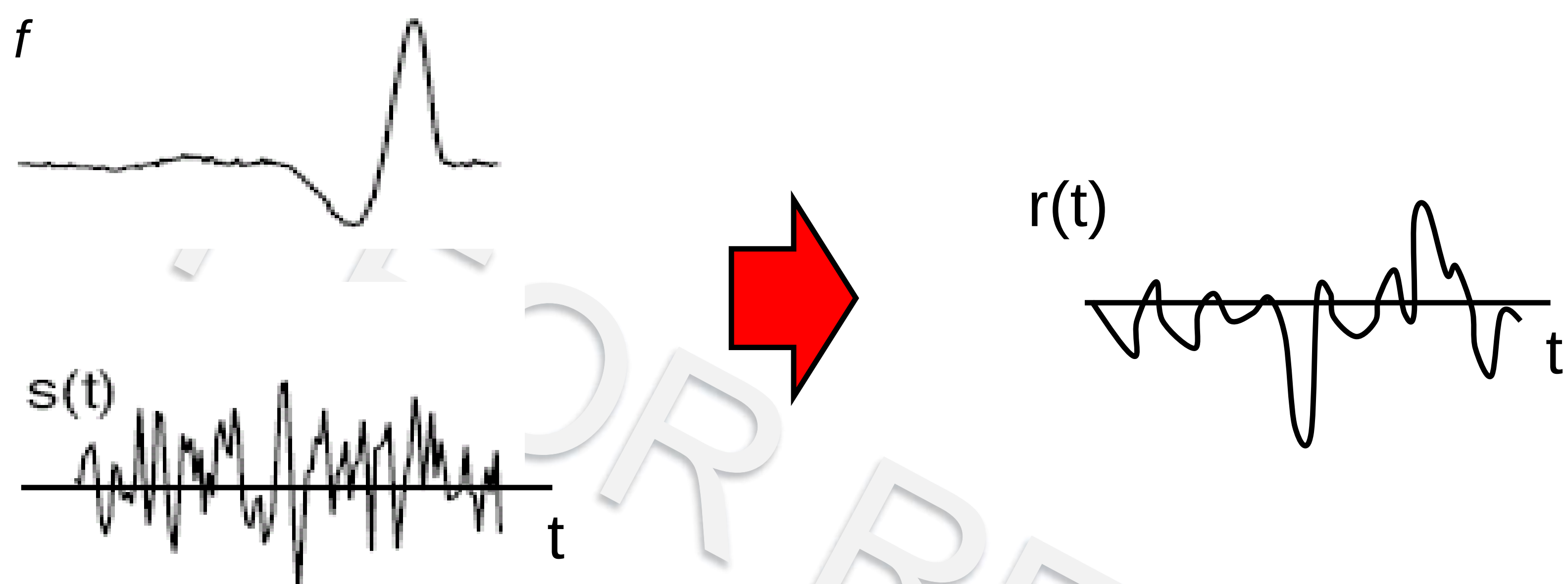
$\boxed{\text{P(response | stimulus)}}$

Basic coding model: linear response

$$r(t) = \phi\, s(t)$$



Basic coding model: temporal filtering

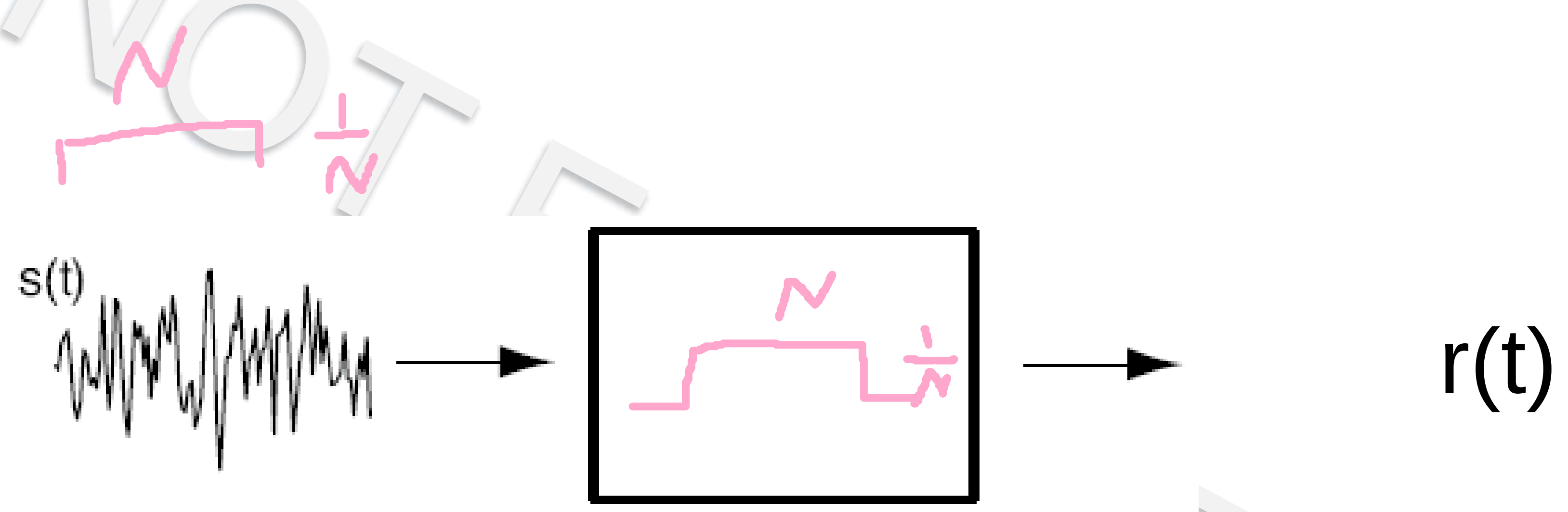


Linear filter:

$$r(t) = \sum_{k=0}^{n} s_{t-k} f_k$$

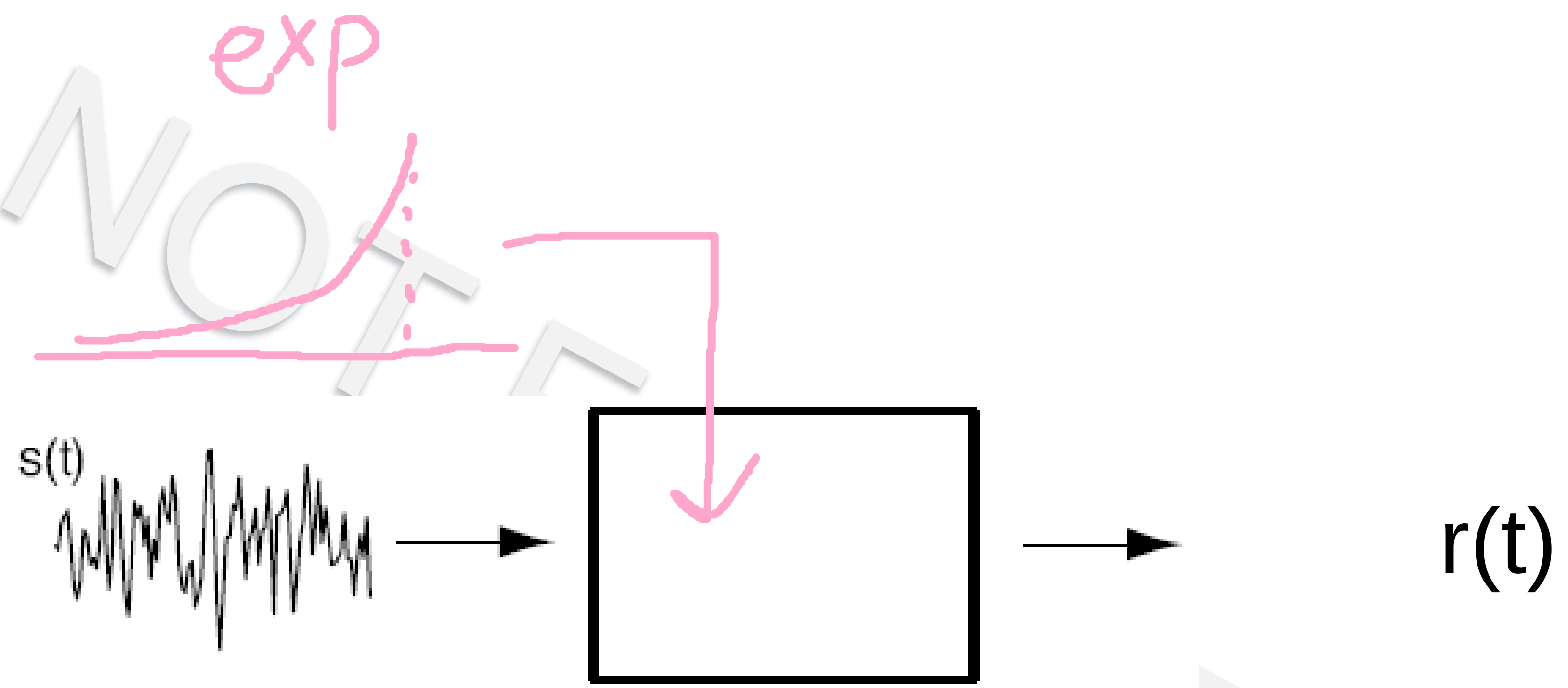
$$r(t) = \int_{-\infty}^{t} d\tau \, s(t-\tau) f(\tau)$$

Example I: running average



Linear filter: $$r(t) = \sum_{k=0}^{n} s_{t-k} f_k$$

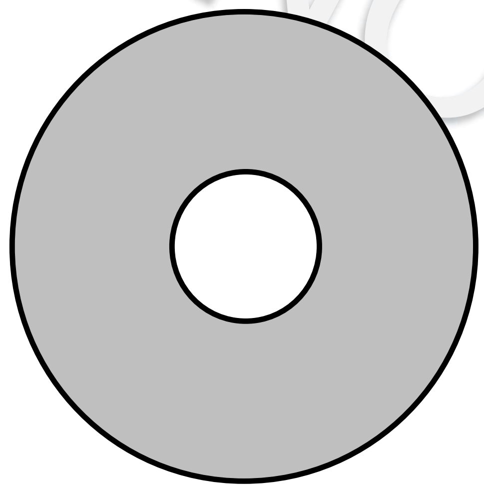
Example II: leaky average



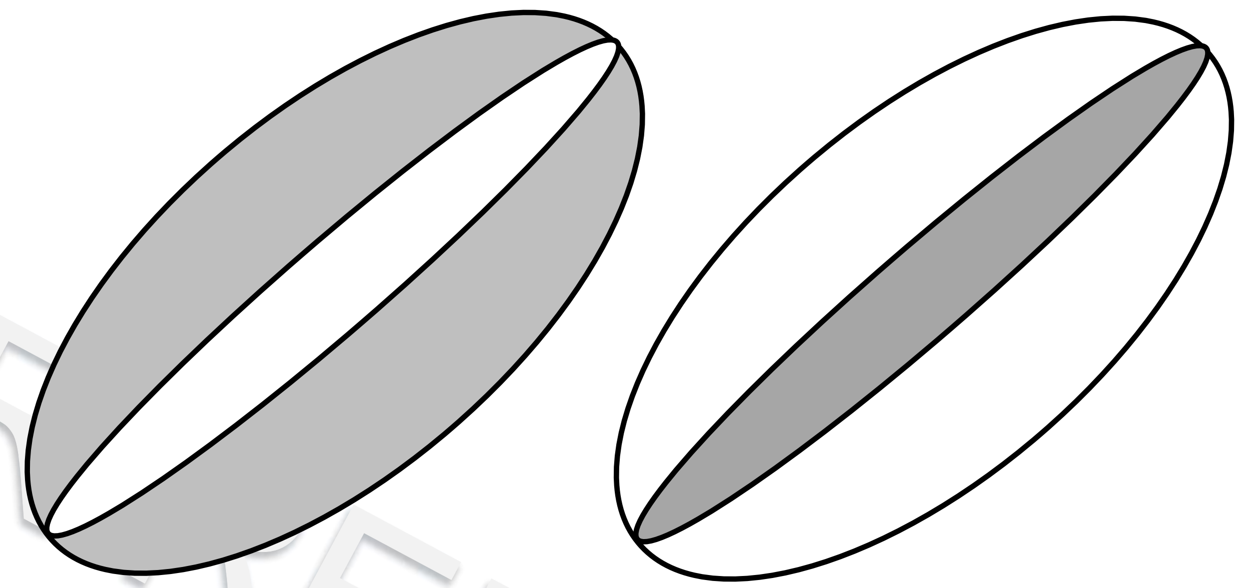
Linear filter: $$r(t) = \sum_{k=0}^{n} s_{t-k} f_k$$

Basic coding model: spatial filtering

Basic coding model: spatial filtering



retina



Visual cortex

Basic coding model: spatial filtering

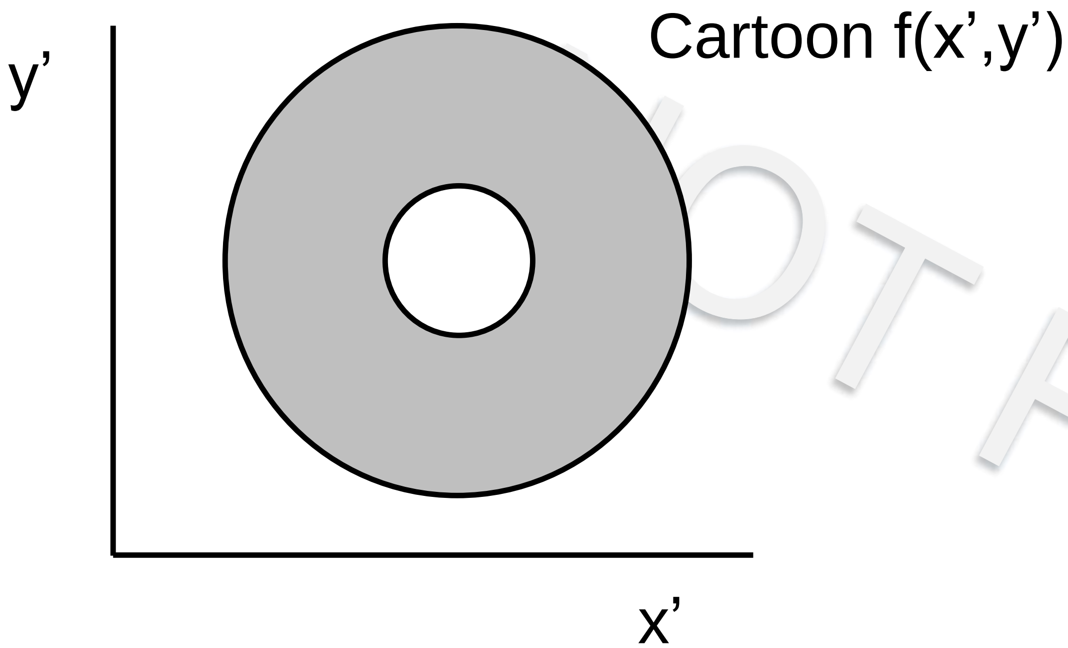
$$r(t) = \sum_{k=0}^{n} s_{t-k} f_k$$

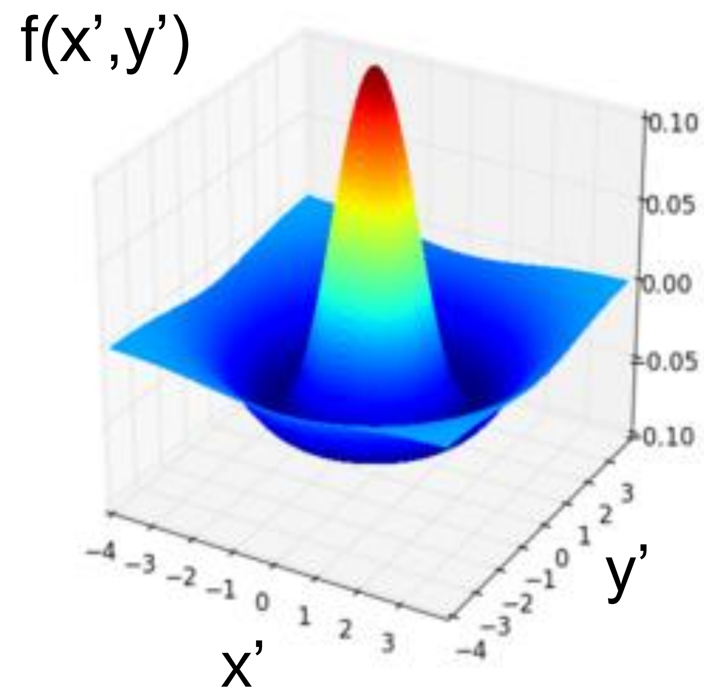
Temporal filter

$$r(x,y) = \sum_{x\prime=-n, y\prime=-n}^{n} s_{x-x\prime, y-y\prime} f_{x\prime, y\prime}$$

$$= \int_{-\infty}^{\infty} dx' dy' \, s(x-x', y-y') f(x', y')$$

Spatial filtering and retinal receptive fields

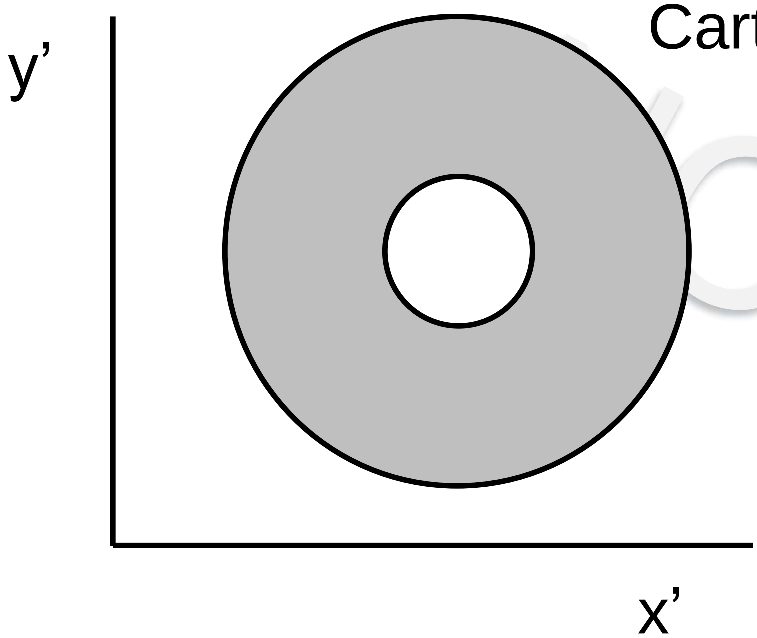


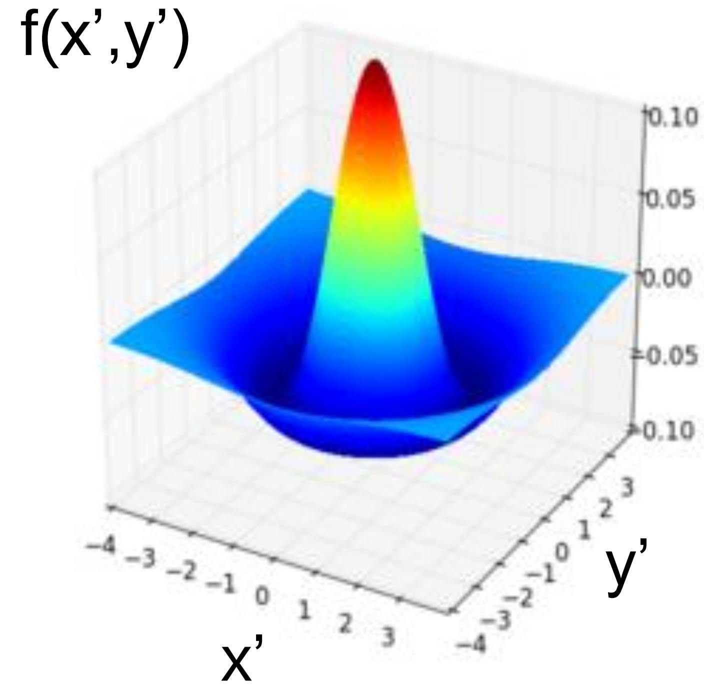


$$r(x,y) = \sum_{x'=-n,y'=-n}^{n} s_{x-x',y-y'} f_{x',y'}$$

http://en.wikibooks.org/wiki/Sensory_Systems/Visual_System

Spatial filtering and receptive fields

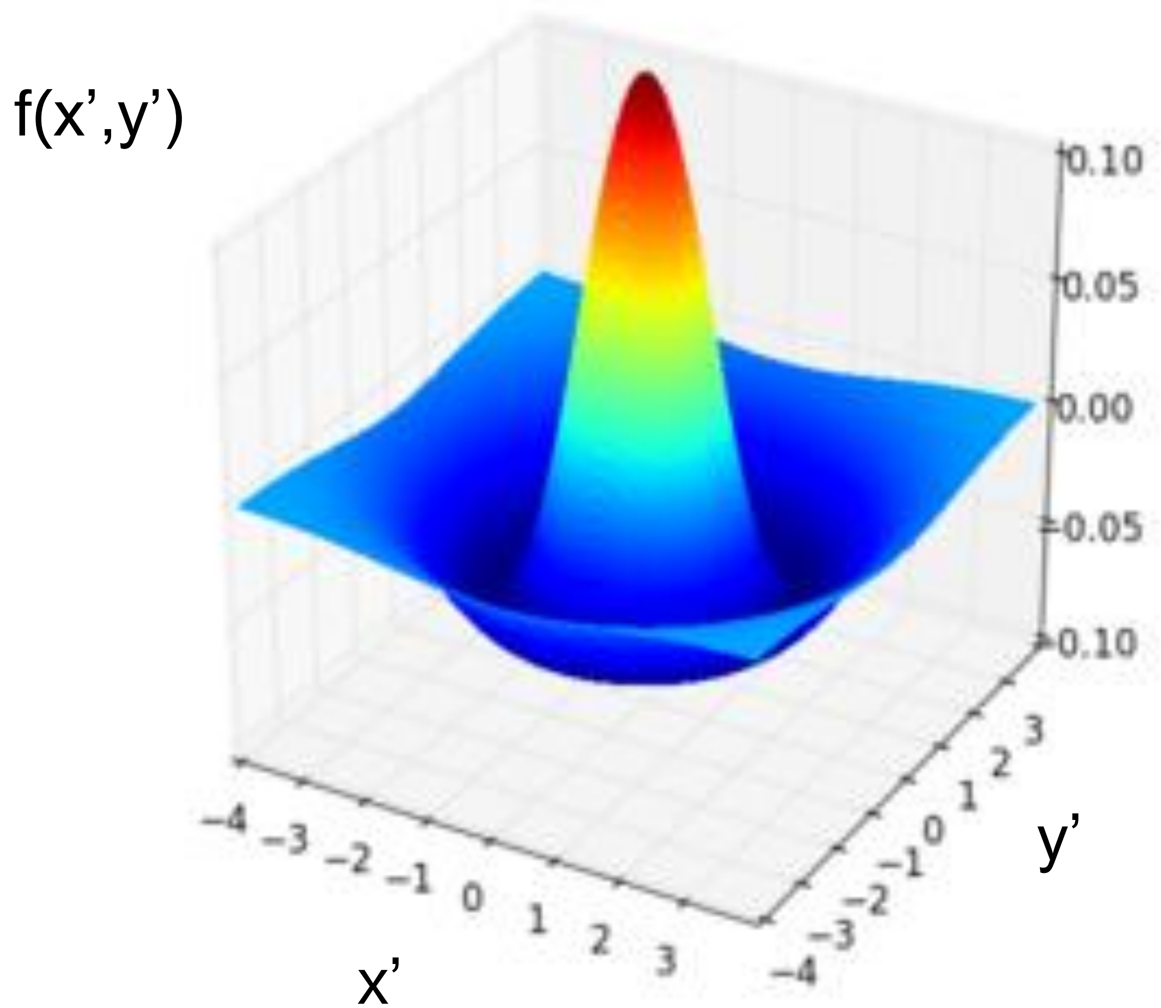




$$r(x,y) = \sum_{x'=-n,y'=-n}^{n} s_{x-x',y-y'} f_{x',y'}$$

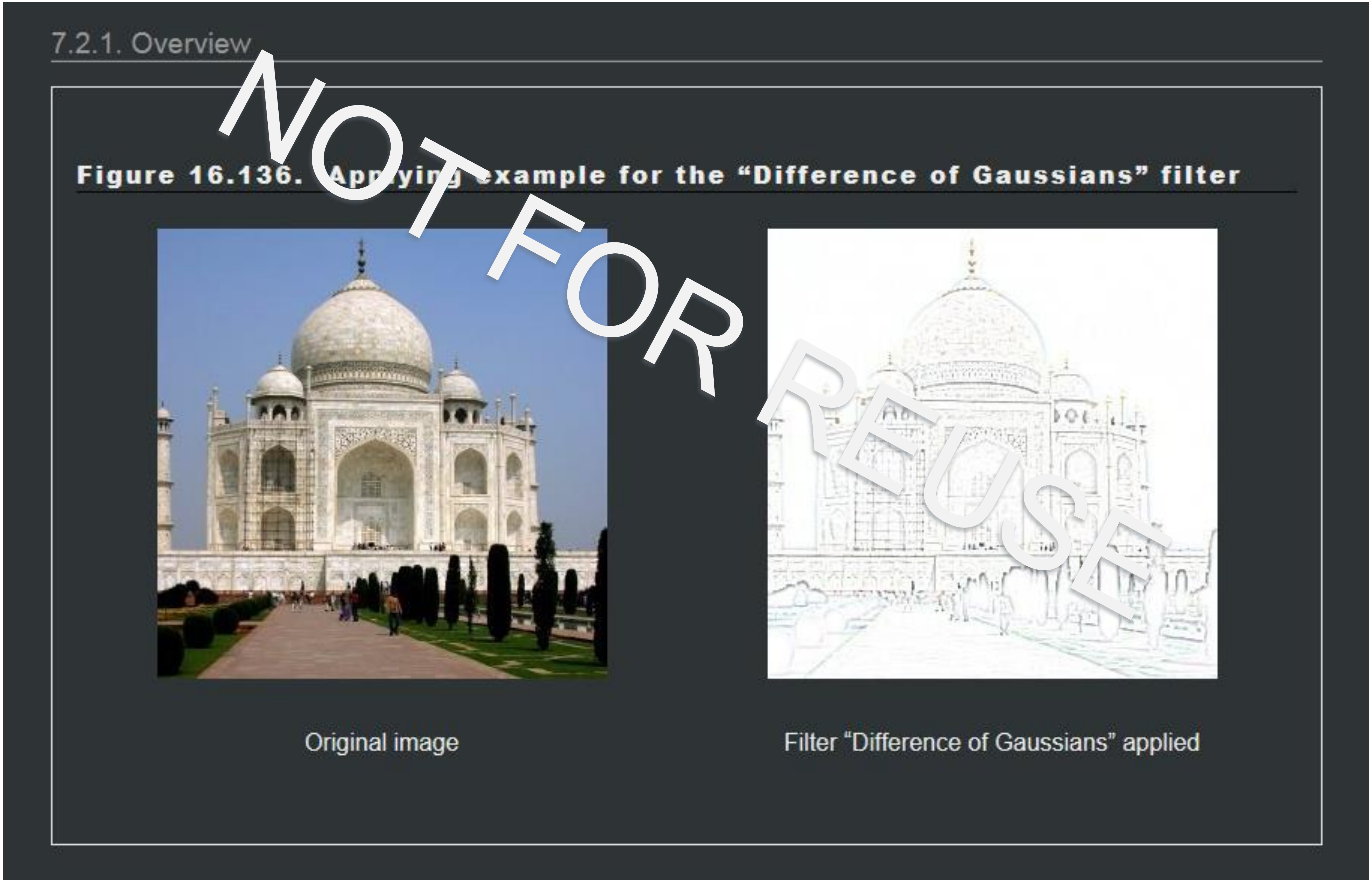
http://en.wikibooks.org/wiki/Sensory_Systems/Visual_System

Spatial filtering and receptive fields



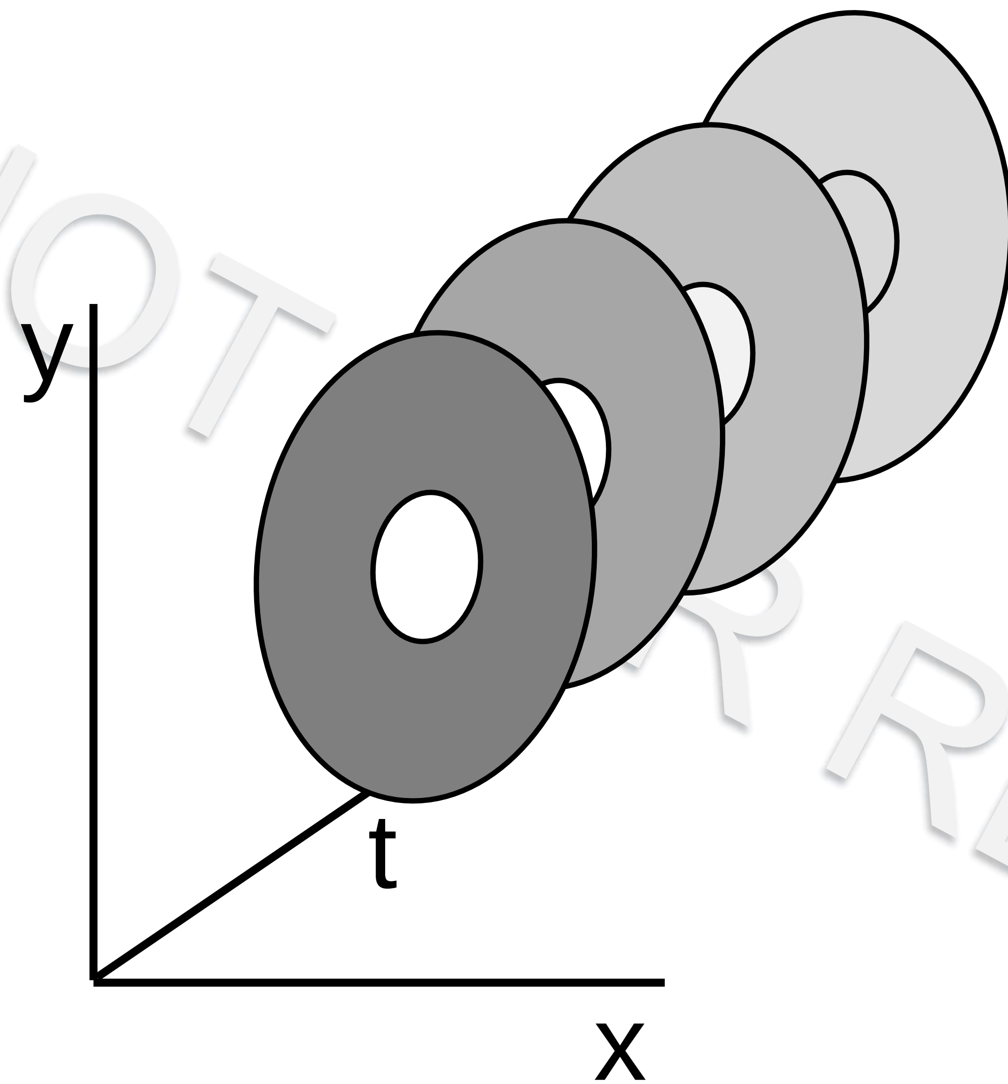
http://en.wikibooks.org/wiki/Sensory_Systems/Visual_System

Spatial filtering



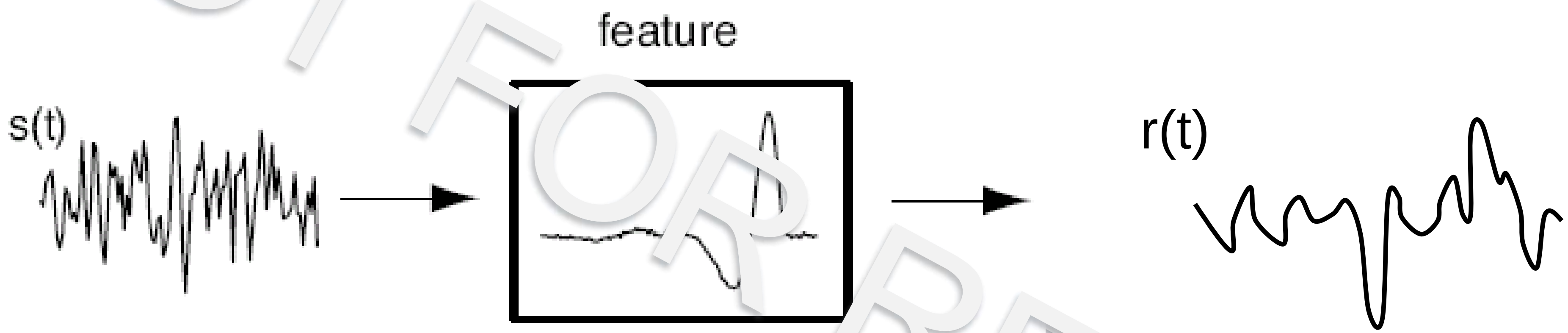
http://docs.gimp.org/2.6/en/plug-in-dog.html

Basic coding model: *spatiotemporal* filtering



$$r_{x,y}(t) = \iiint dx'\, dy'\, d\tau\ f(x',y',\tau)\ s(x-x',y-y',t-\tau)$$

Basic coding model: temporal filtering

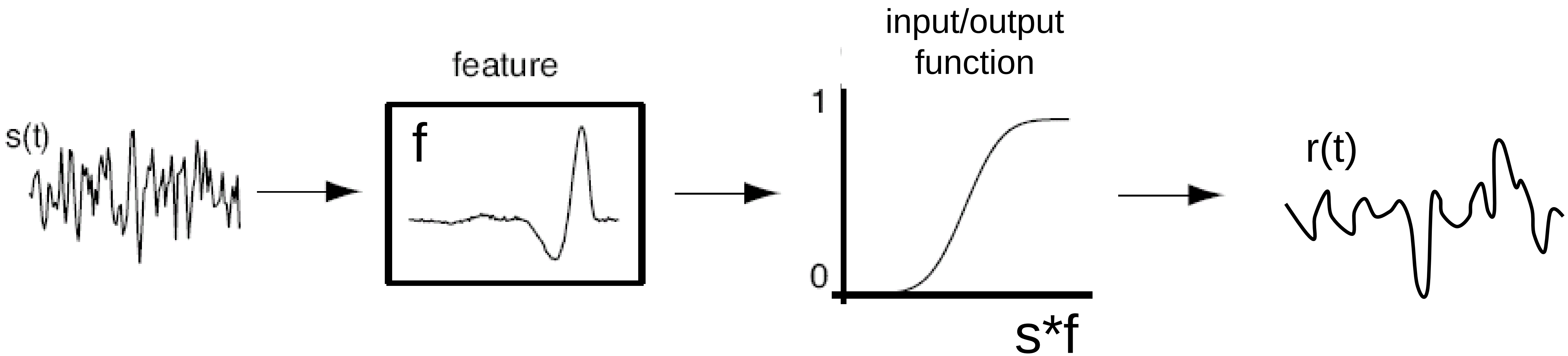


Linear filter: $r(t) = \int s(t-\tau)\, f(\tau)\, d\tau$

Can firing rates be negative? Can they increase indefinitely as the input increases?
Both of those are a possible result from a linear filtering operation like this.

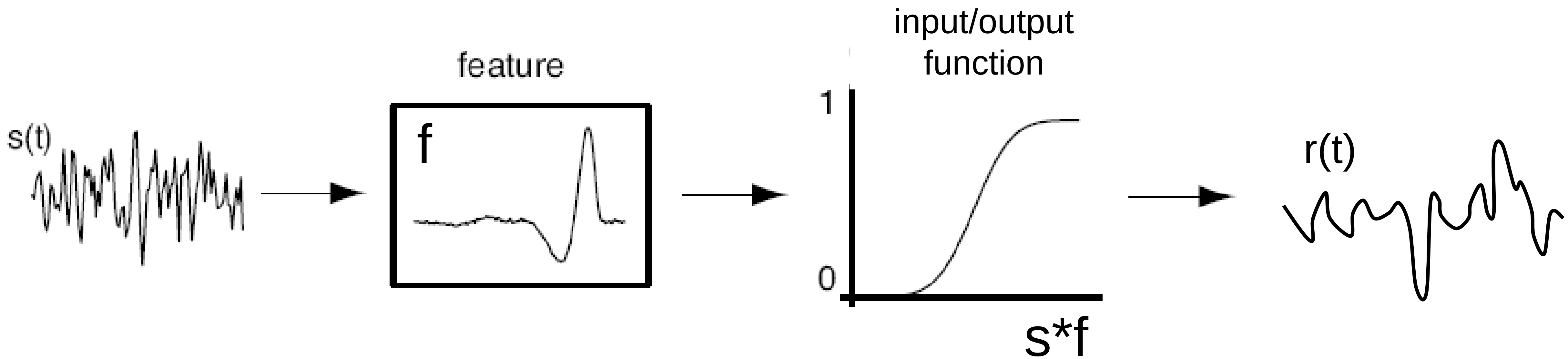
…shortcomings?

Next most basic coding model

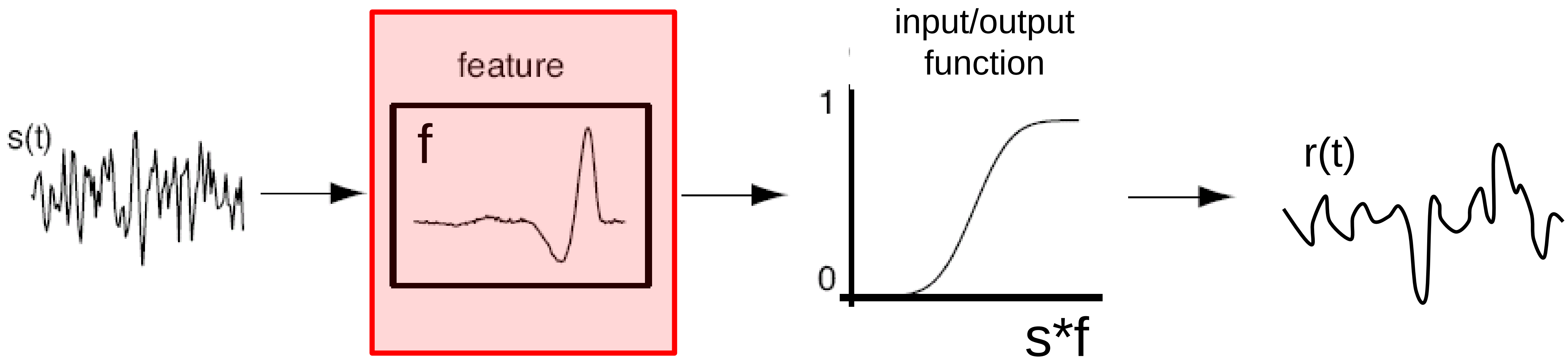


Linear filter & nonlinearity: $r(t) = g(\int s(t-\tau)\, f(\tau)\, d\tau)$

How to find the components of this model



How to find the components of this model



How to proceed?

P(response | stimulus)

Our problem is one of dimensionality!

Time points × pixels = 非常多的维度
导致没法sample出整个distribution
所以我们只能降维

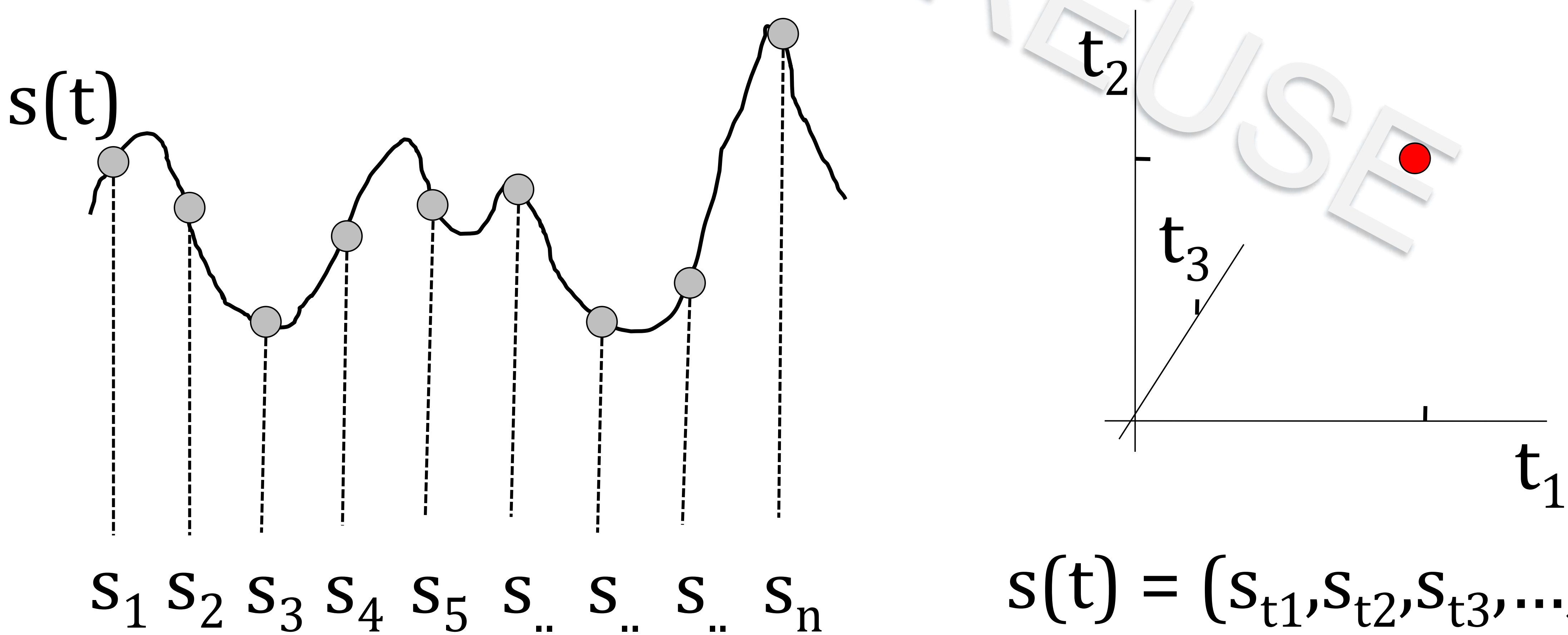
We want to sample the responses of the system to many stimuli so we can characterize what it is about the input that triggers responses.

P(response | stimulus) → P(response | s_1)

Dimensionality reduction

Start with a very high dimensional description
(eg. an image or a time-varying waveform)
and pick out a small set of relevant dimensions.

这里对不同时间的刺激s进行采样，得到了的刺激在不同时间的特征。
We discretize a stimulus waveform in time, we can represent it as a vector in some vector space.
The dimensionality of this vector space is the number of points used in the discretization.
我们想知道s是什么。我们可以用s随着时间的概率分布来刻画它。但是我们不知道s随着时间的概率分布，所以我们通过采样不同时间的刺激，来刻画这个刺激s。



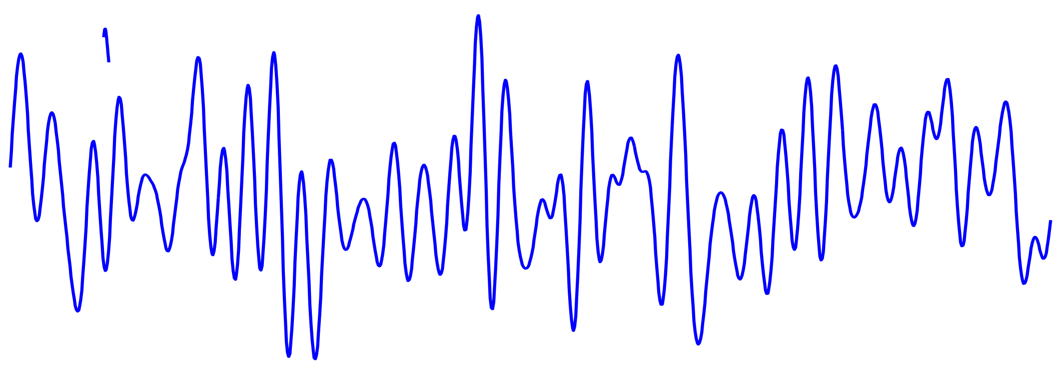
$$s(t) = (s_{t1}, s_{t2}, s_{t3}, \ldots, s_{tn})$$

What is the right stimulus to use?

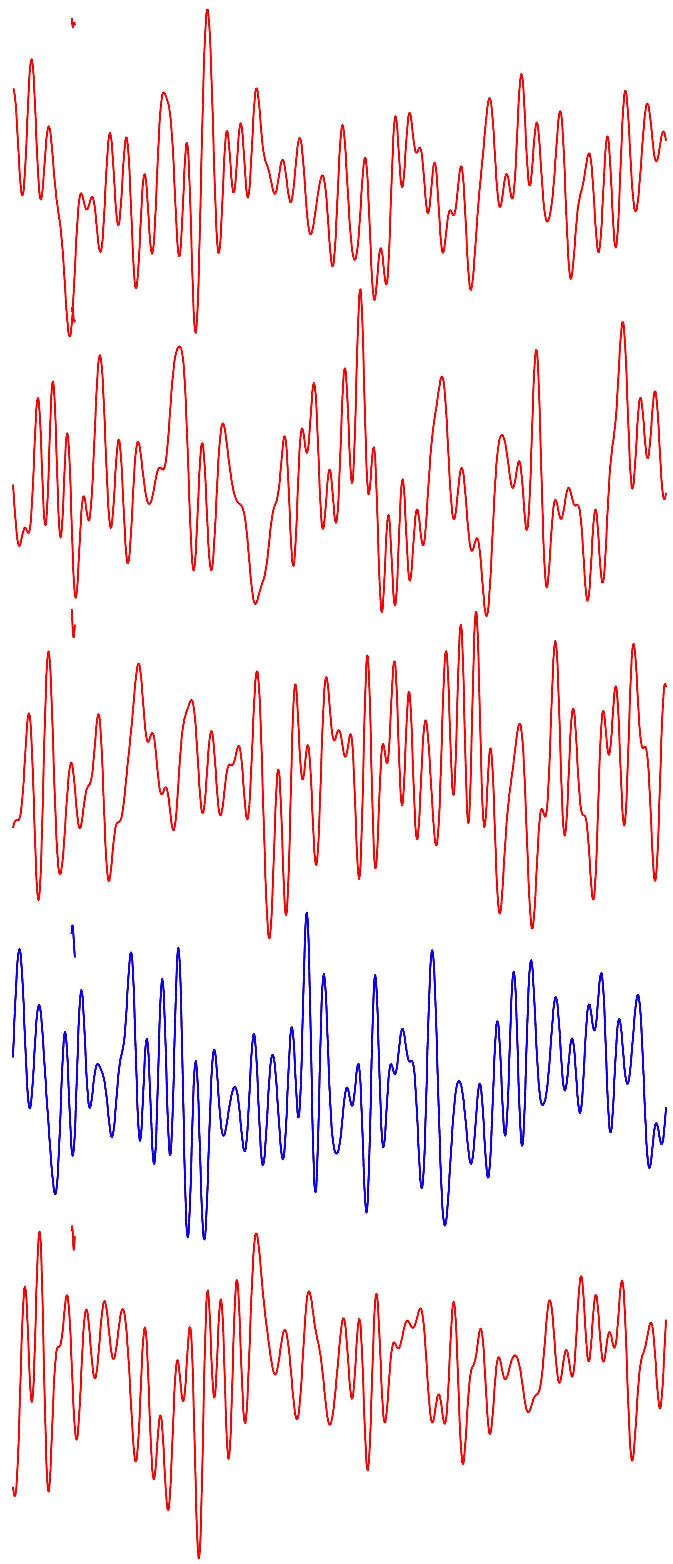
We want to sample the responses of the system to a variety of stimuli so we can characterize what it is about the input that triggers responses.

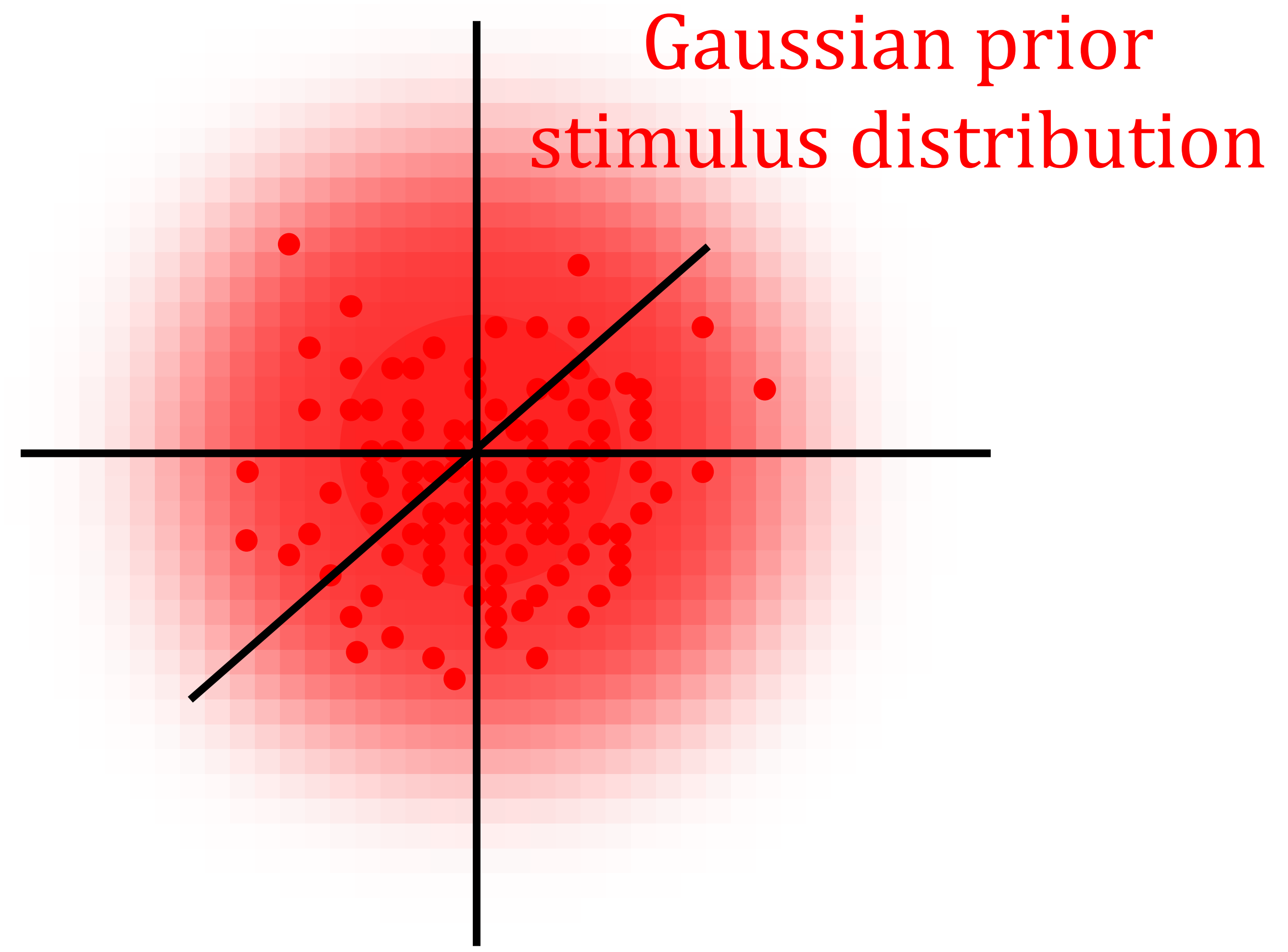
$$P(\text{response} \mid \text{stimulus}) \rightarrow P(\text{response} \mid s_1, s_2, .., s_n)$$

One common and useful method is to use
white noise

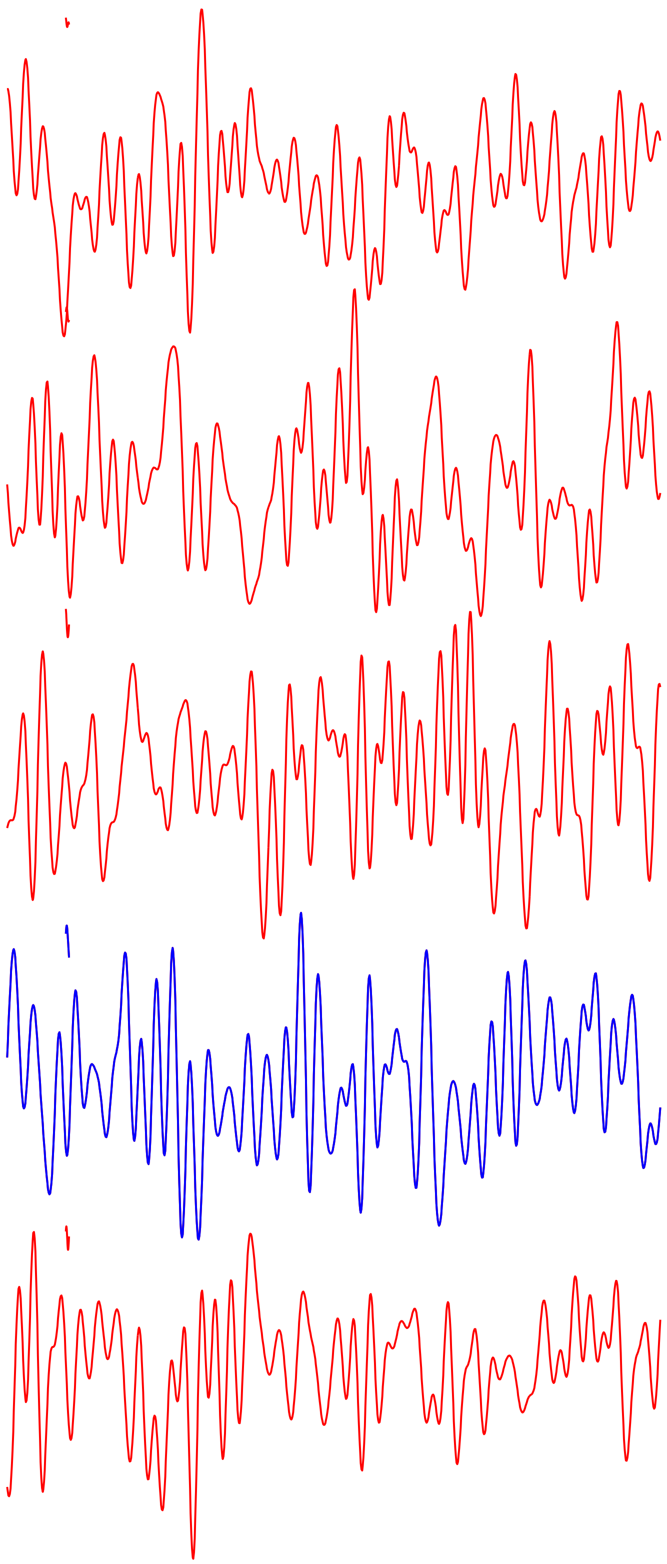


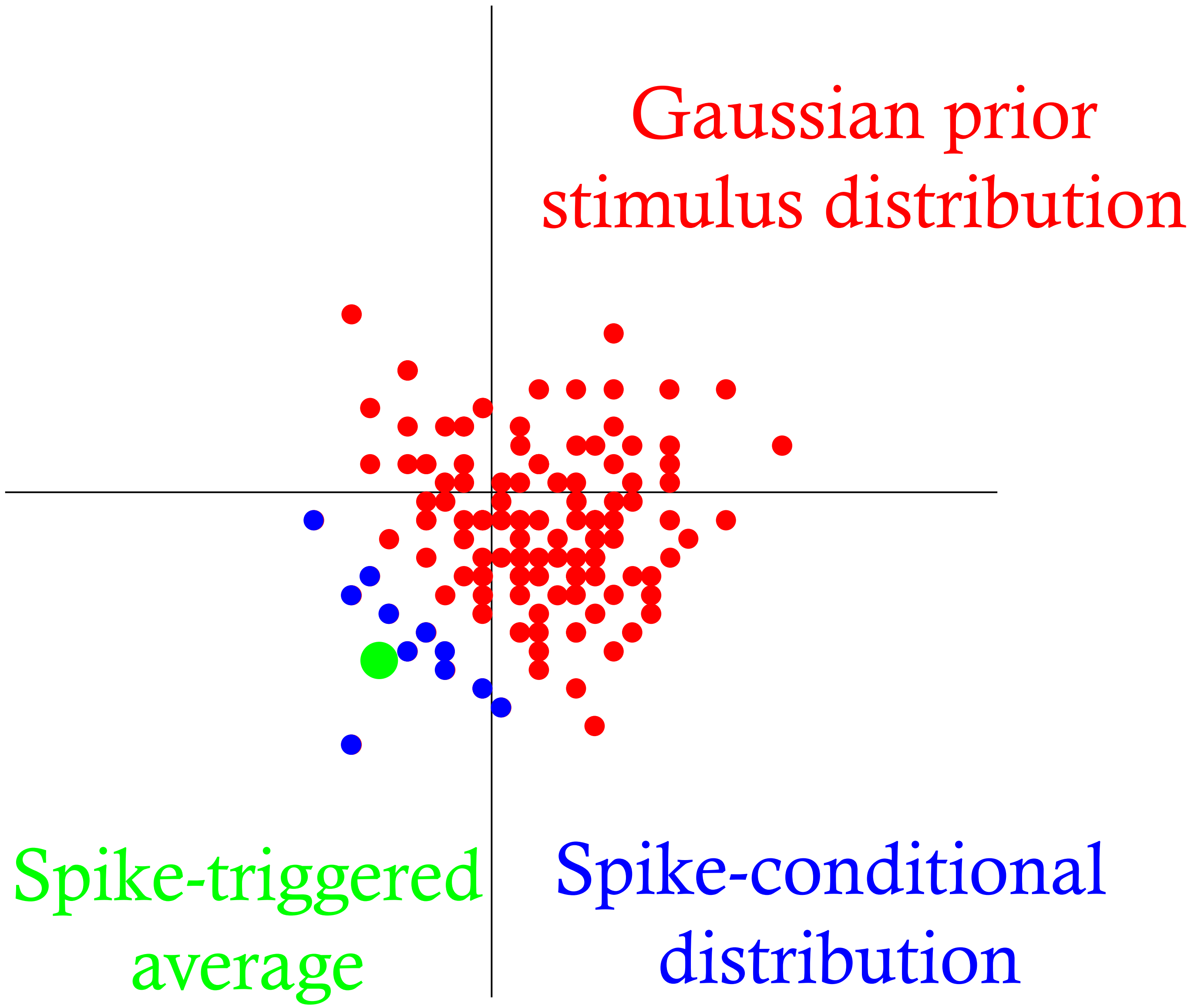
Determining multiple features from white noise



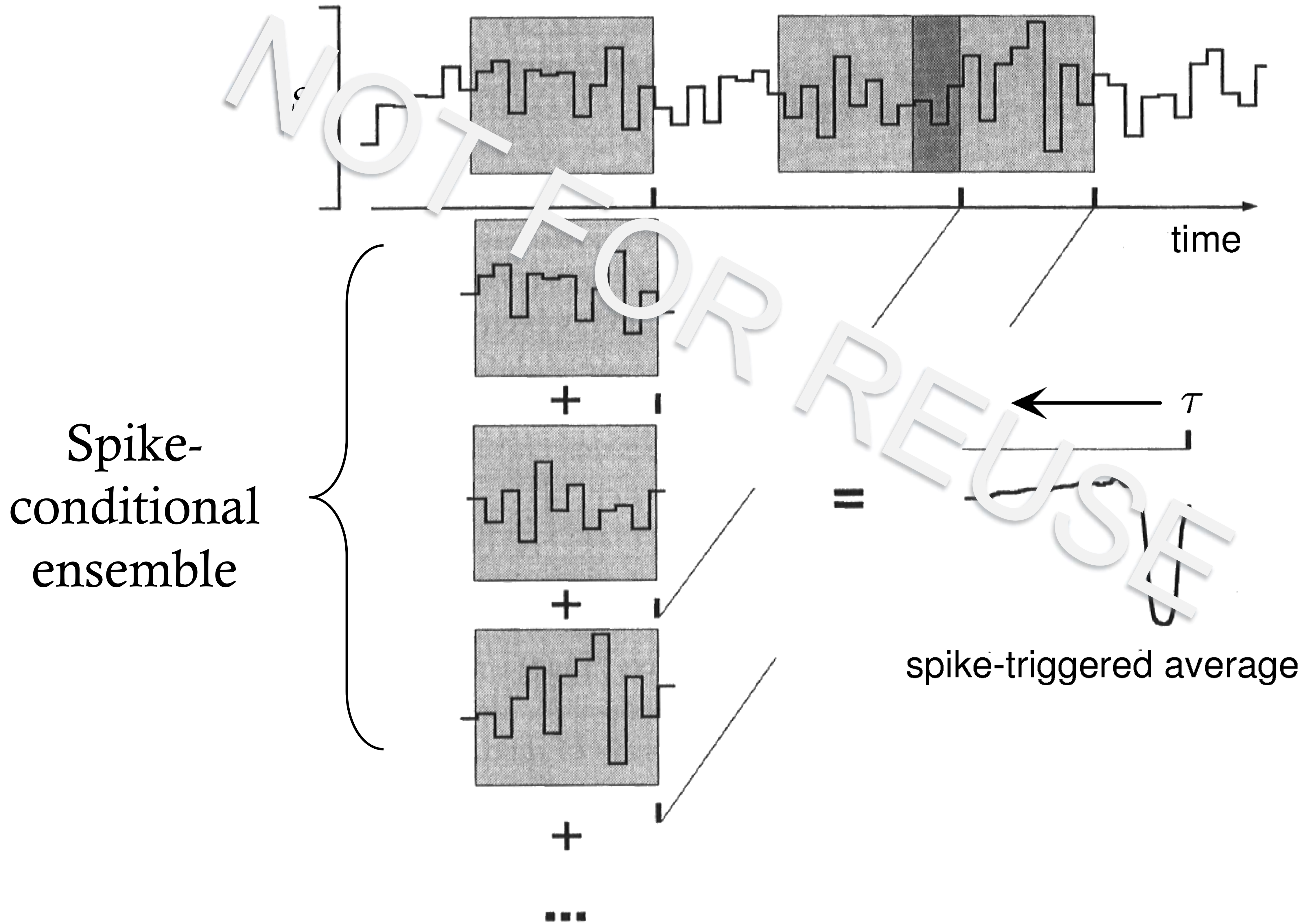


Determining linear features from white noise



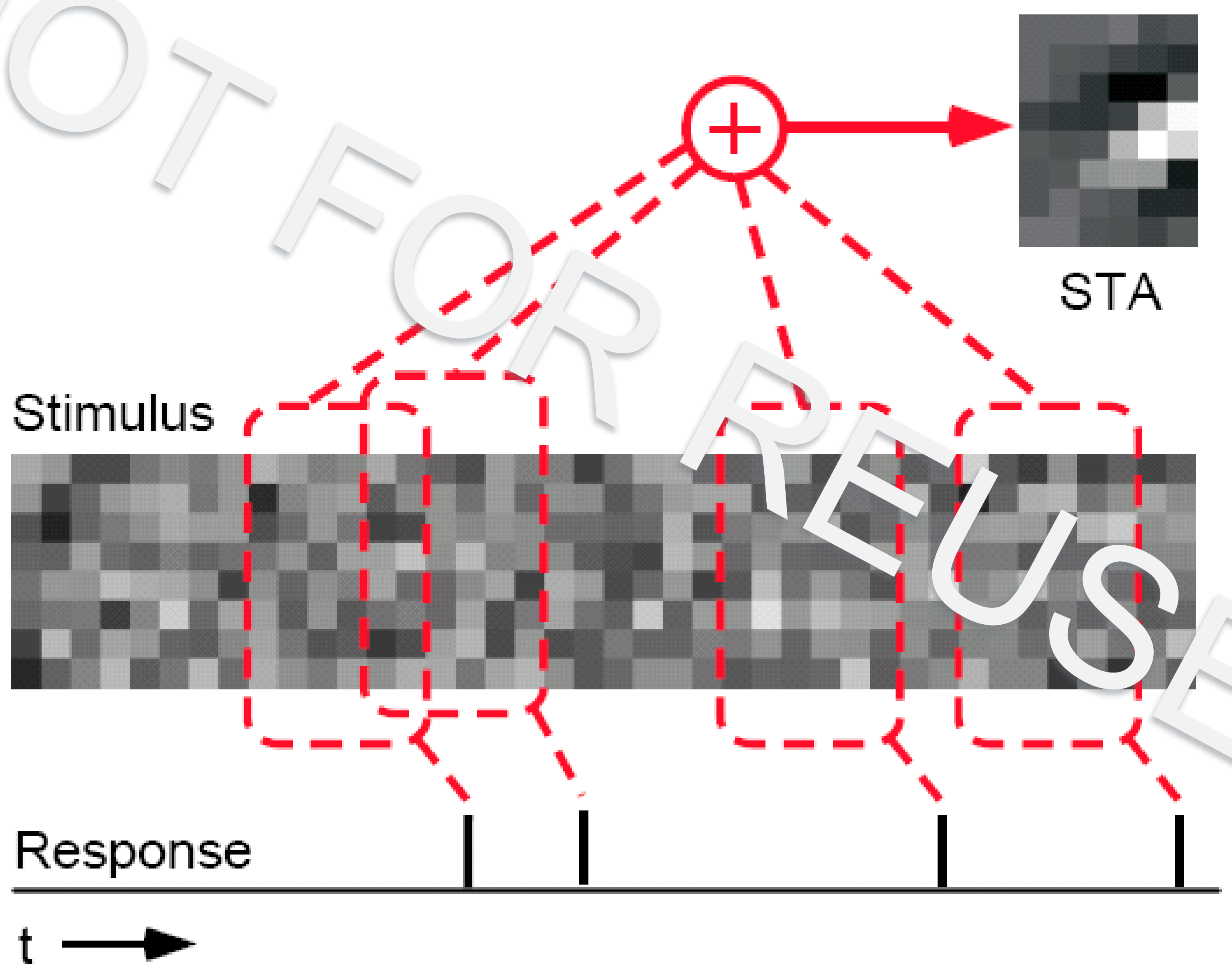


Reverse correlation: the spike-triggered average



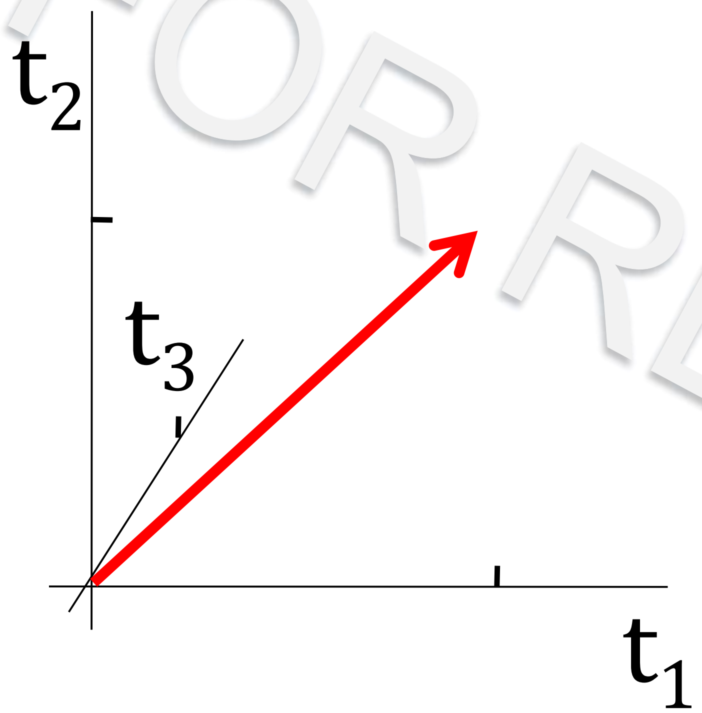
Dayan and Abbott, Theoretical Neuroscience

The spike-triggered average



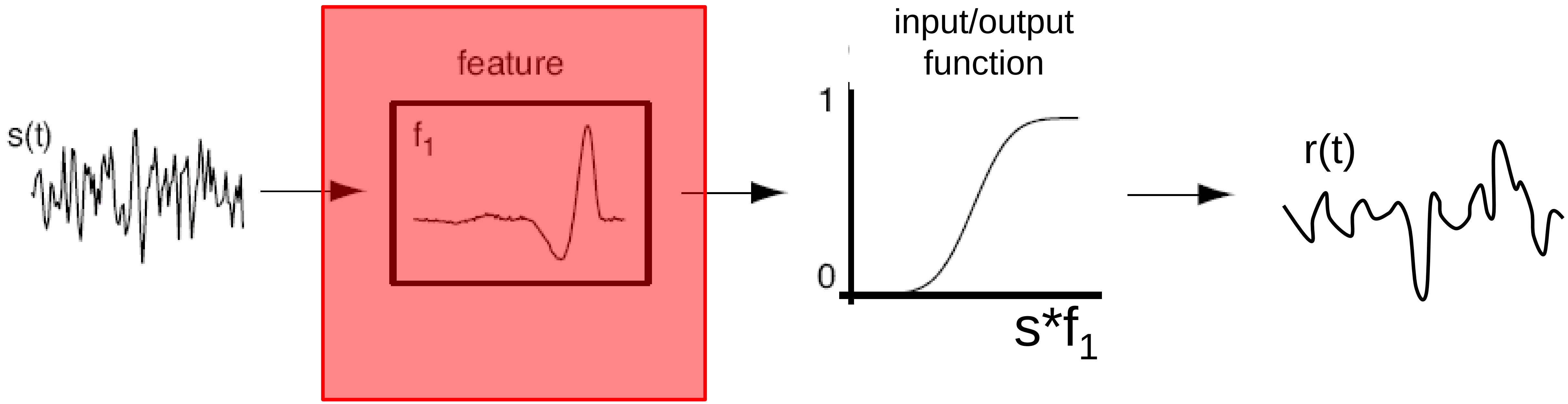
Linear filtering

Stimulus feature *f* is a vector in a high-dimensional stimulus space



Linear filtering = convolution = projection

How to find the components of this model



Determining the nonlinear input/output function

The input/output function is:

$$P(\text{spike}|\text{stimulus}) \Rightarrow P(\text{spike}|s_1)$$

This can be found from data using Bayes' rule:

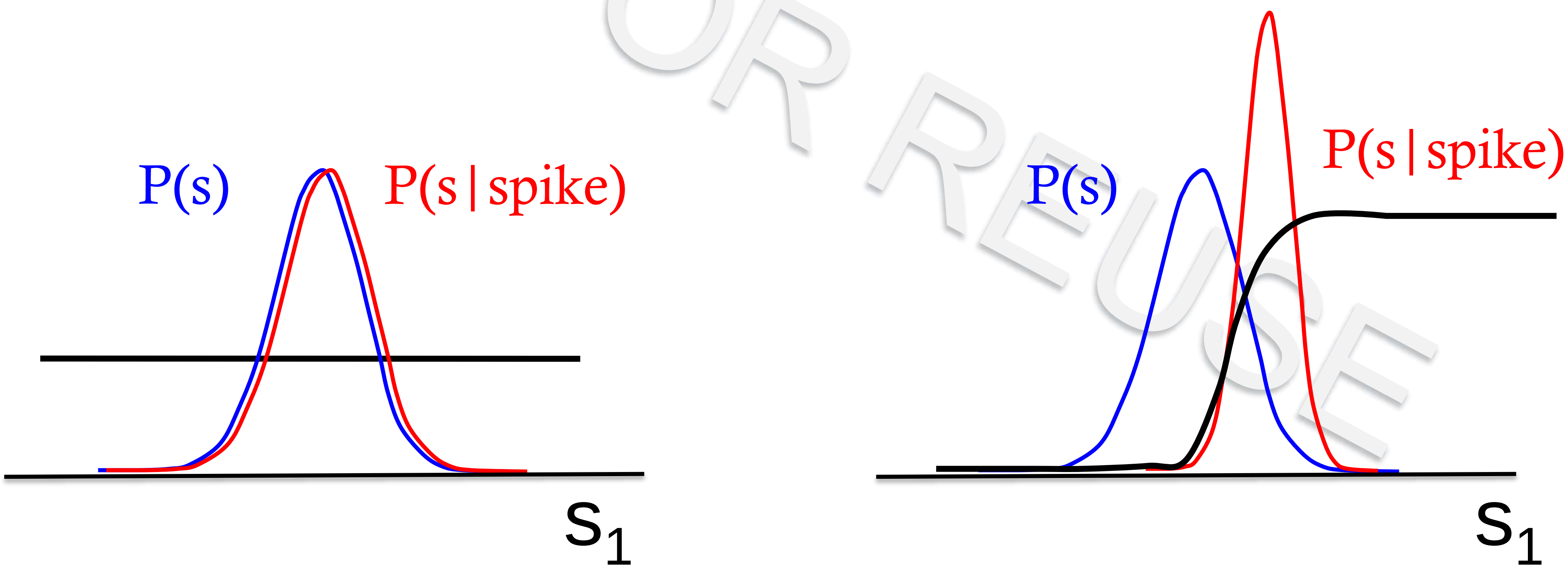
$$P(\text{spike}|s_1) = \frac{P(s_1|\text{spike})P(\text{spike})}{P(s_1)}$$

$P(s_1)$

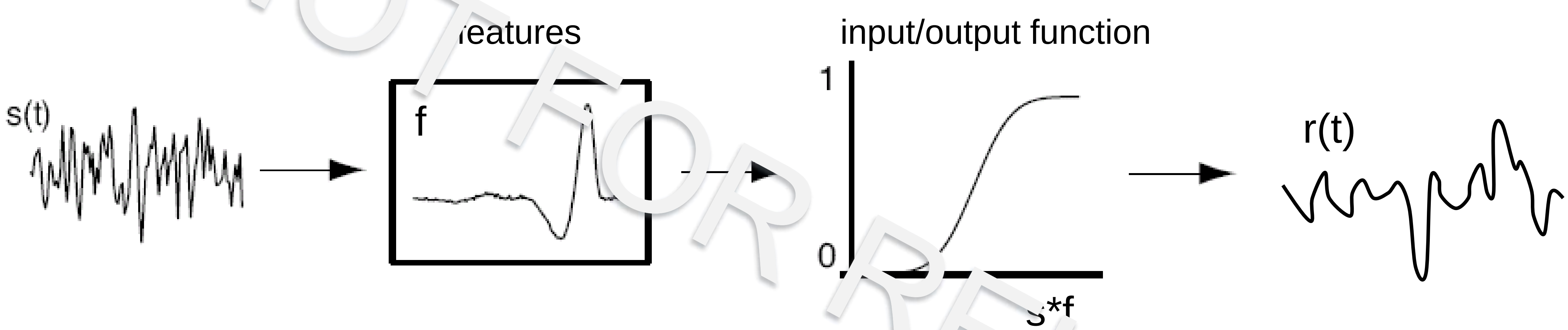
$P(s_1|\text{spike})$

Nonlinear input/output function

$P(\text{spike}\,|\,s_1) = P(s_1\,|\,\text{spike})\, P(\text{spike}) \,/\, P(s_1)$

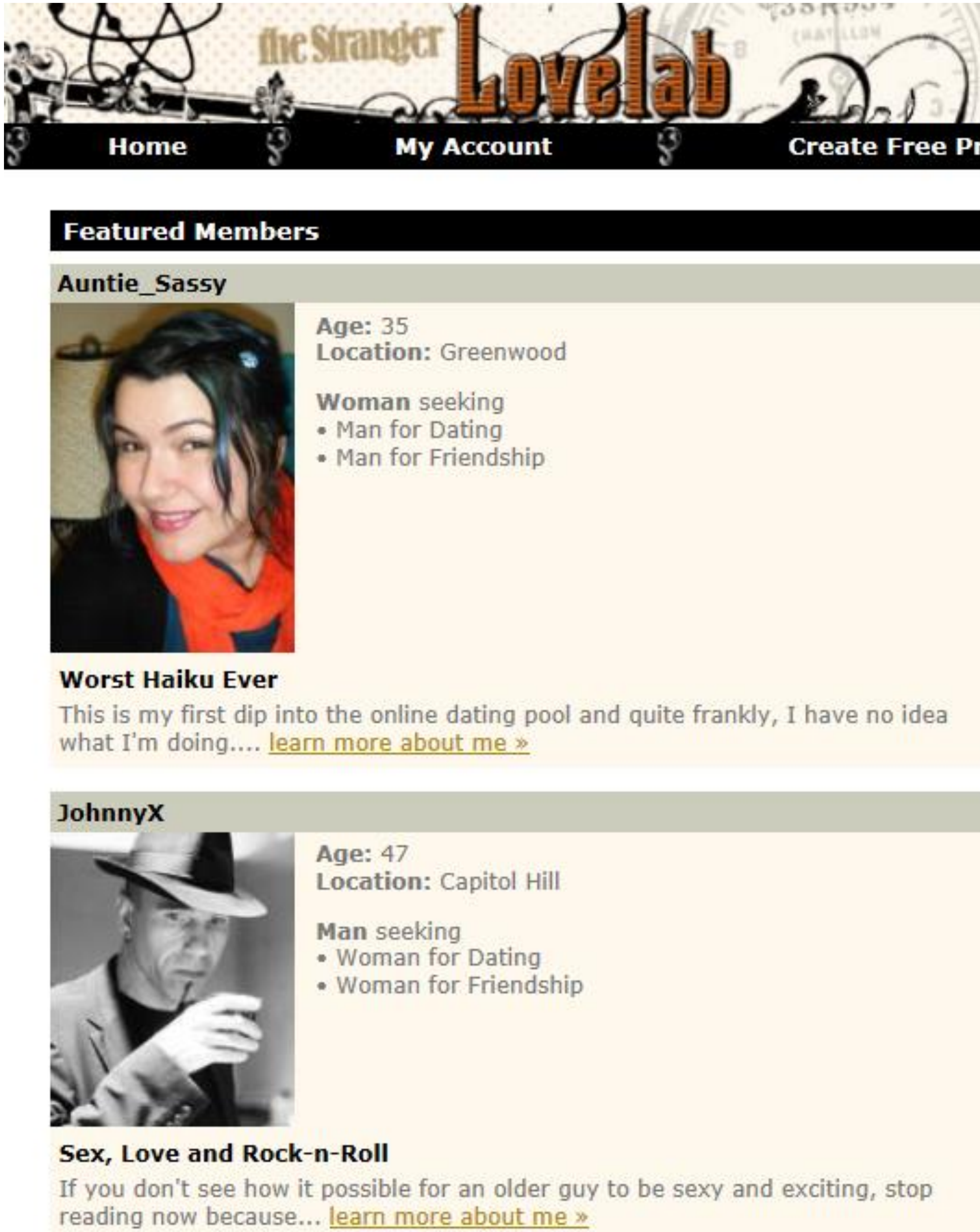


Linear/nonlinear models

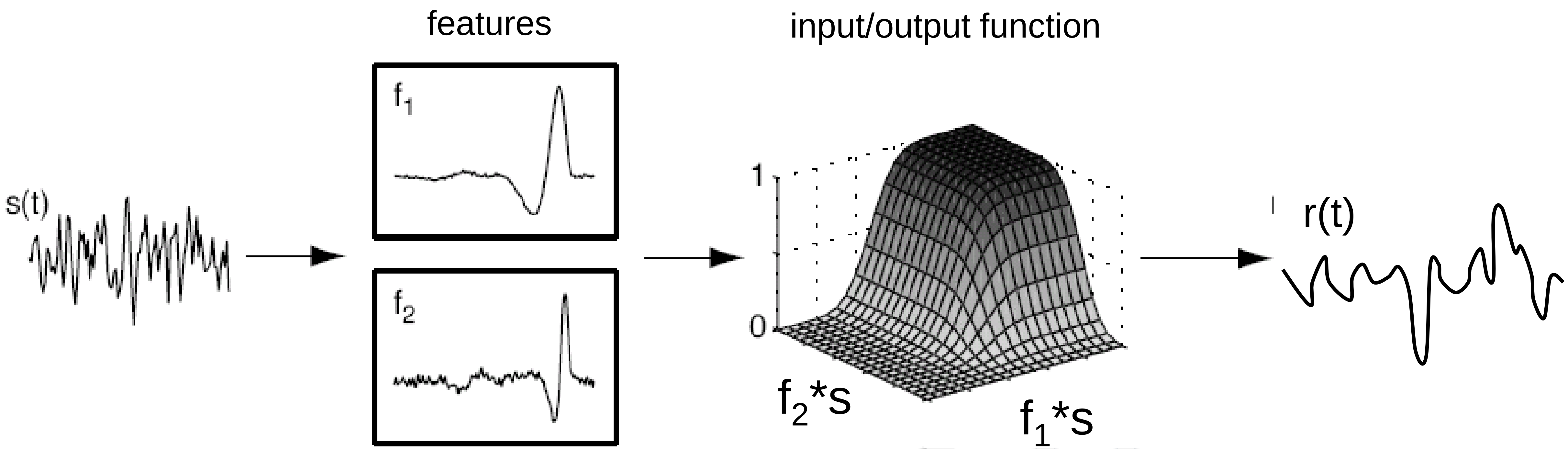


Linear filter & nonlinearity: $r(t) = g(\int f(t-\tau)\, s(\tau)\, dt)$

High-dimensional feature selection

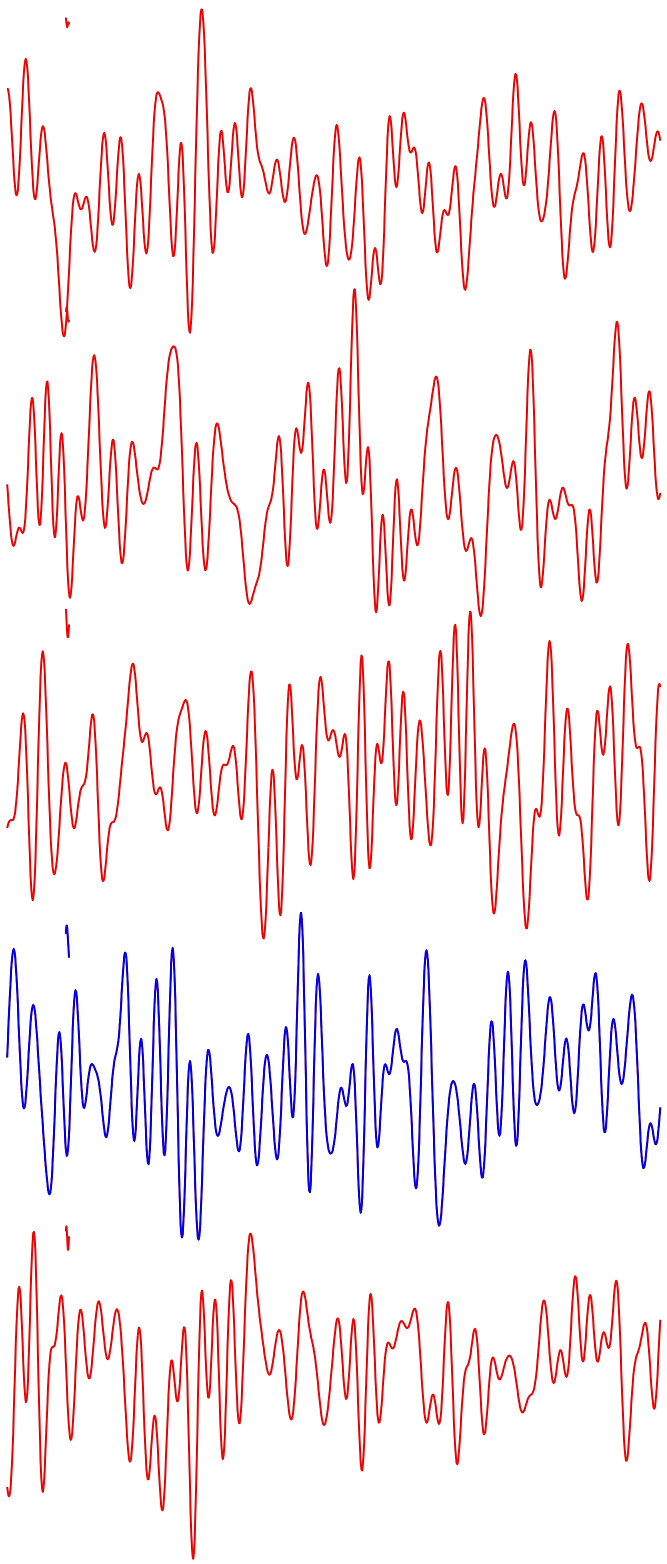


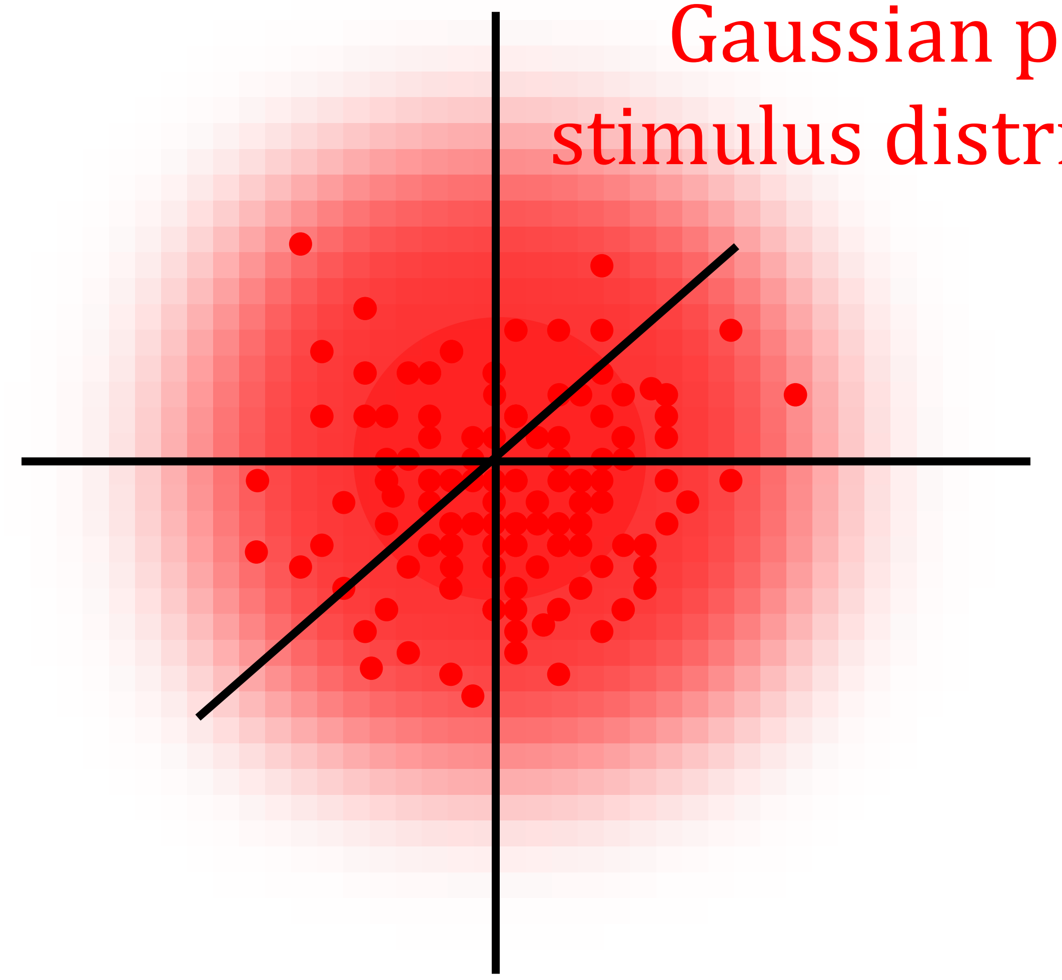
Less basic coding models



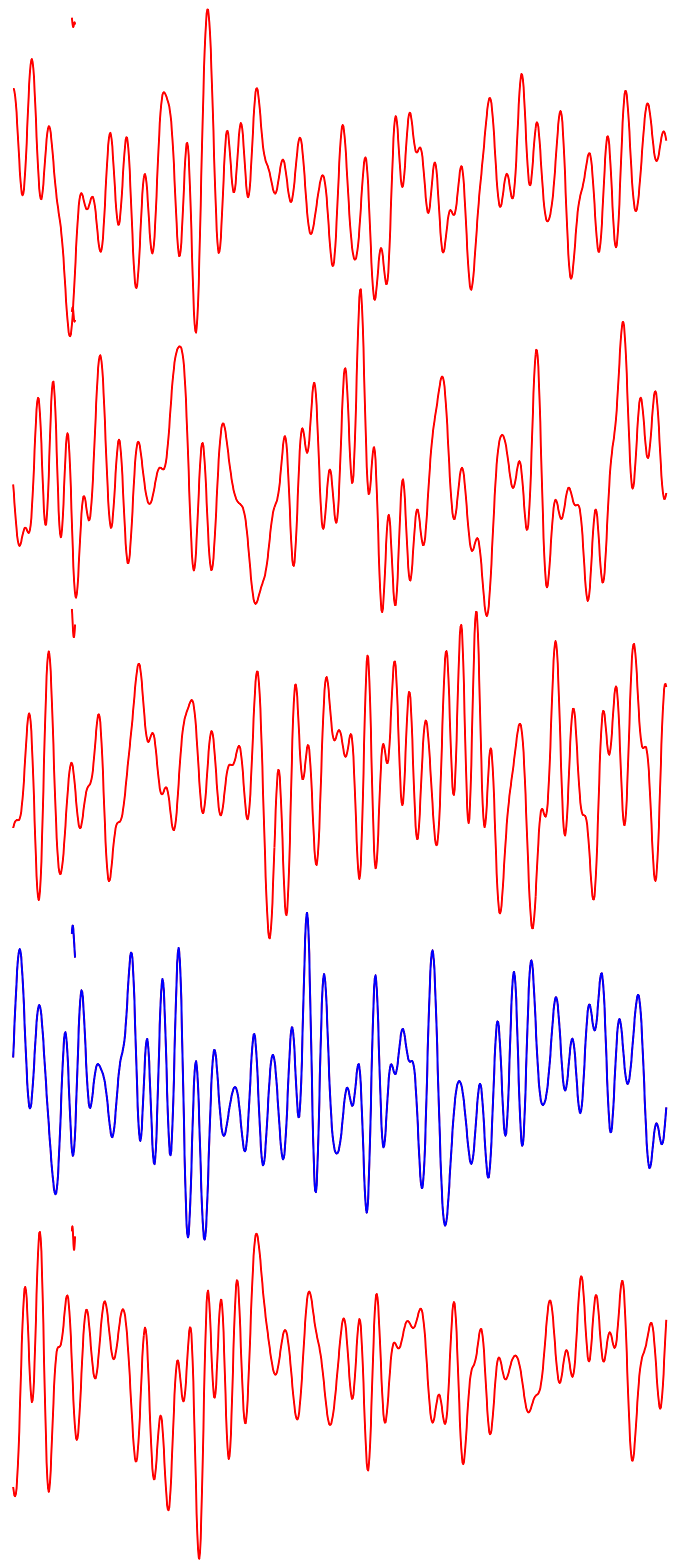
Linear filter*s* & nonlinearity: $r(t) = g(f_1{*}s, f_2{*}s, \ldots, f_n{*}s)$

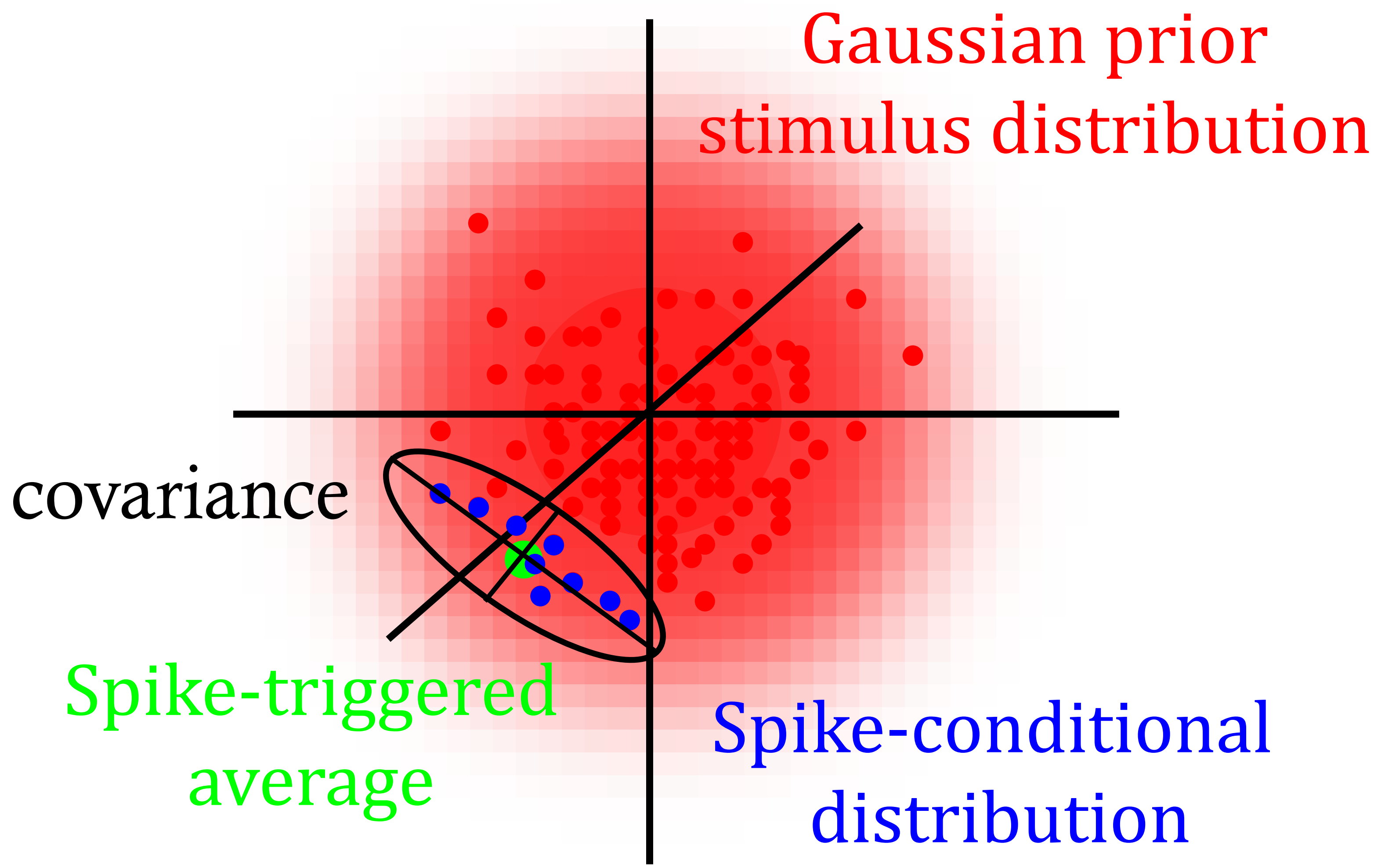
Determining multiple features from white noise



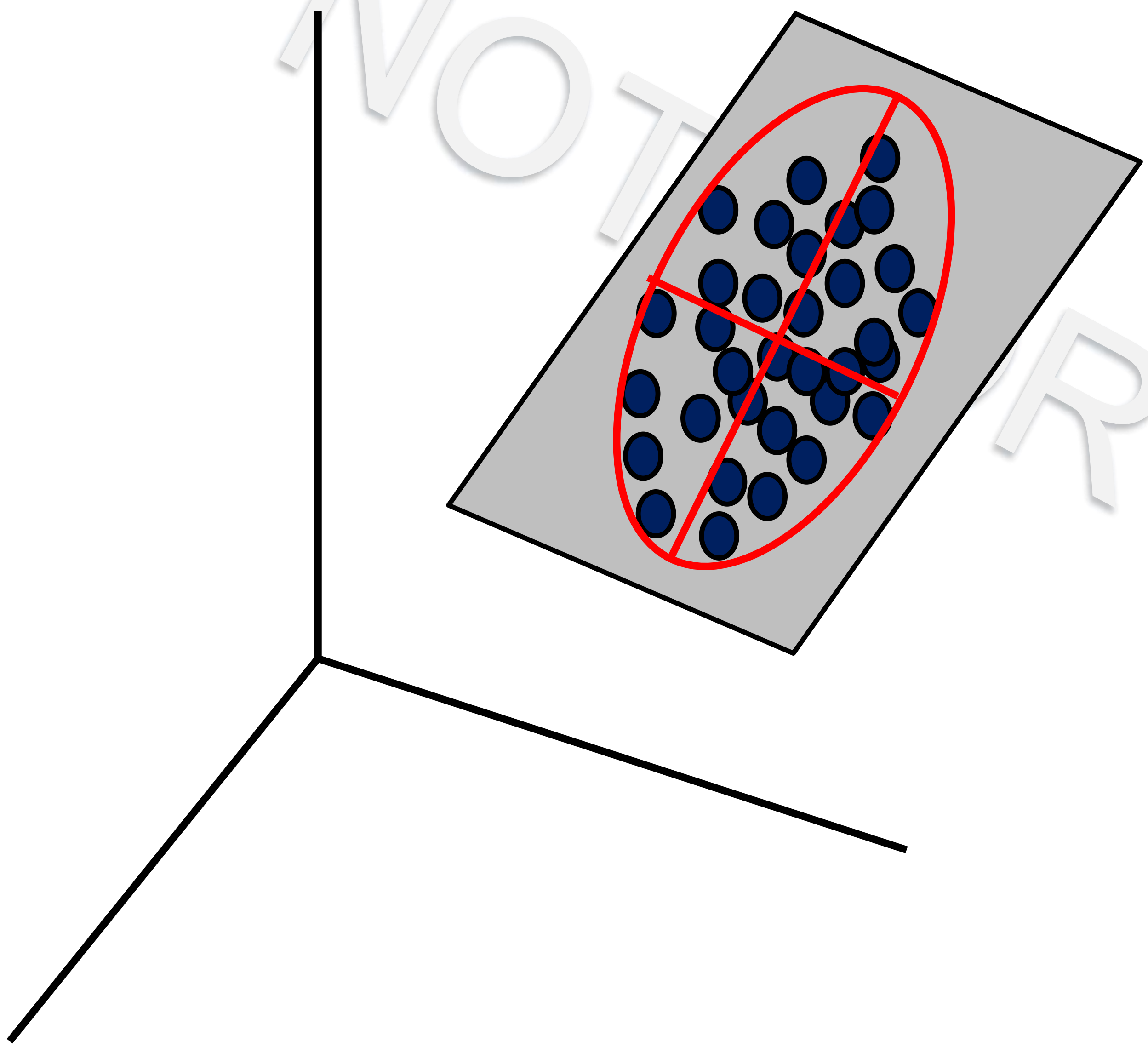


Determining multiple features from white noise

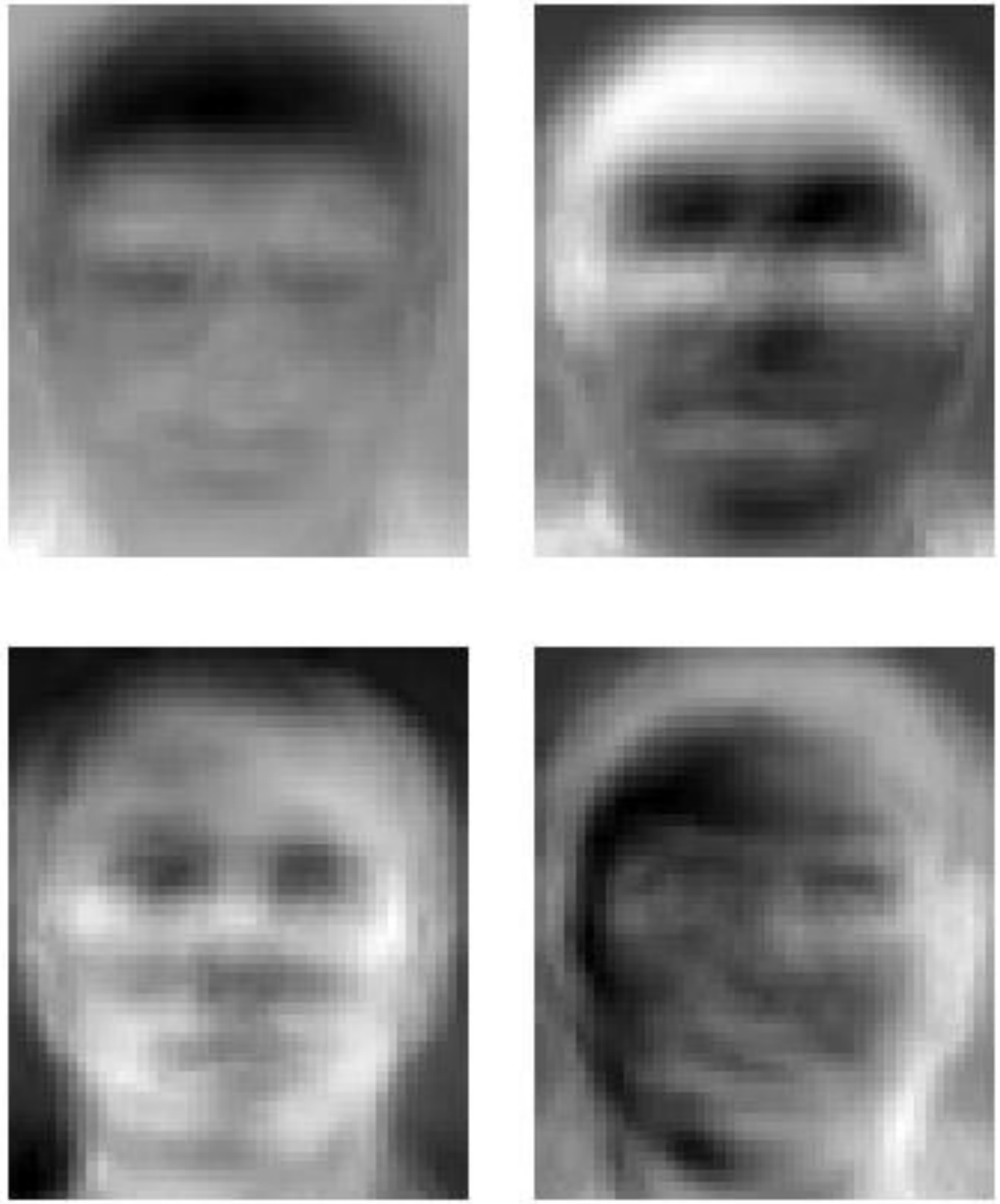




Principal component analysis

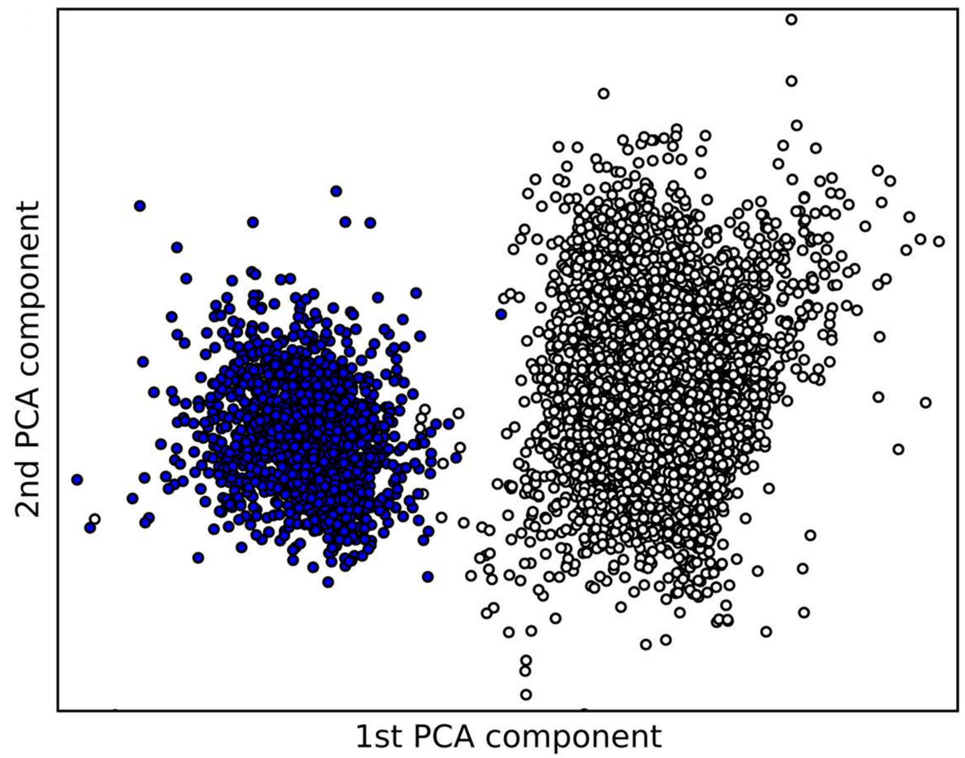


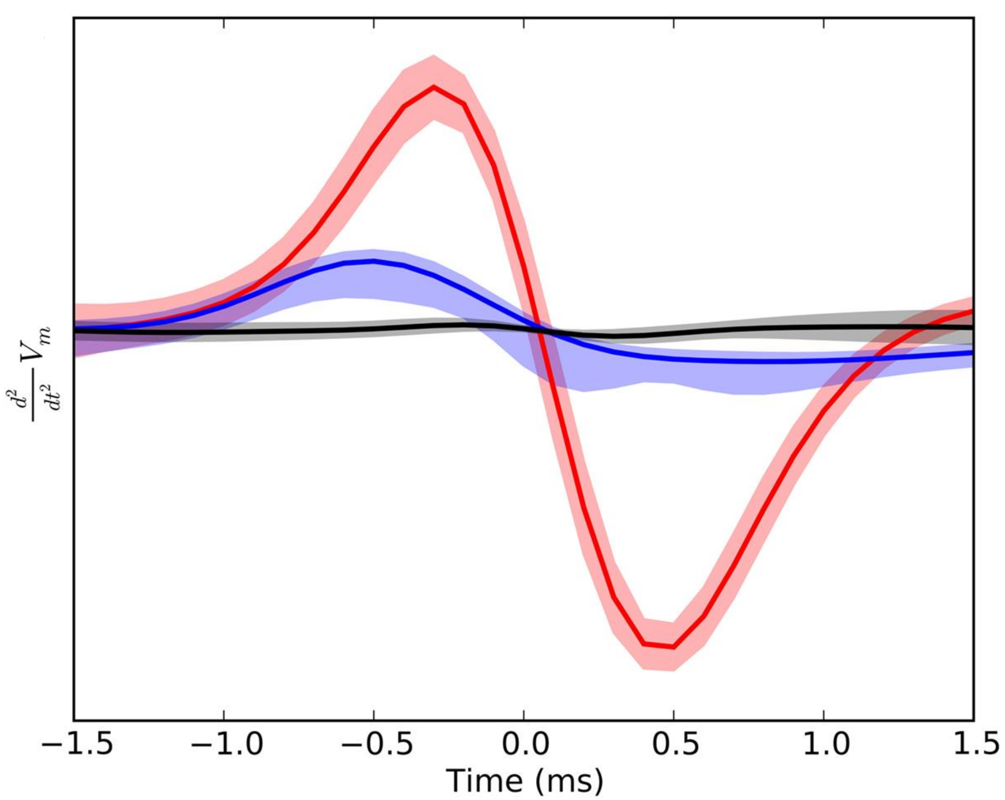
Principal component analysis: eigenfaces



ATT Labs, Cambridge (via Wikipedia)

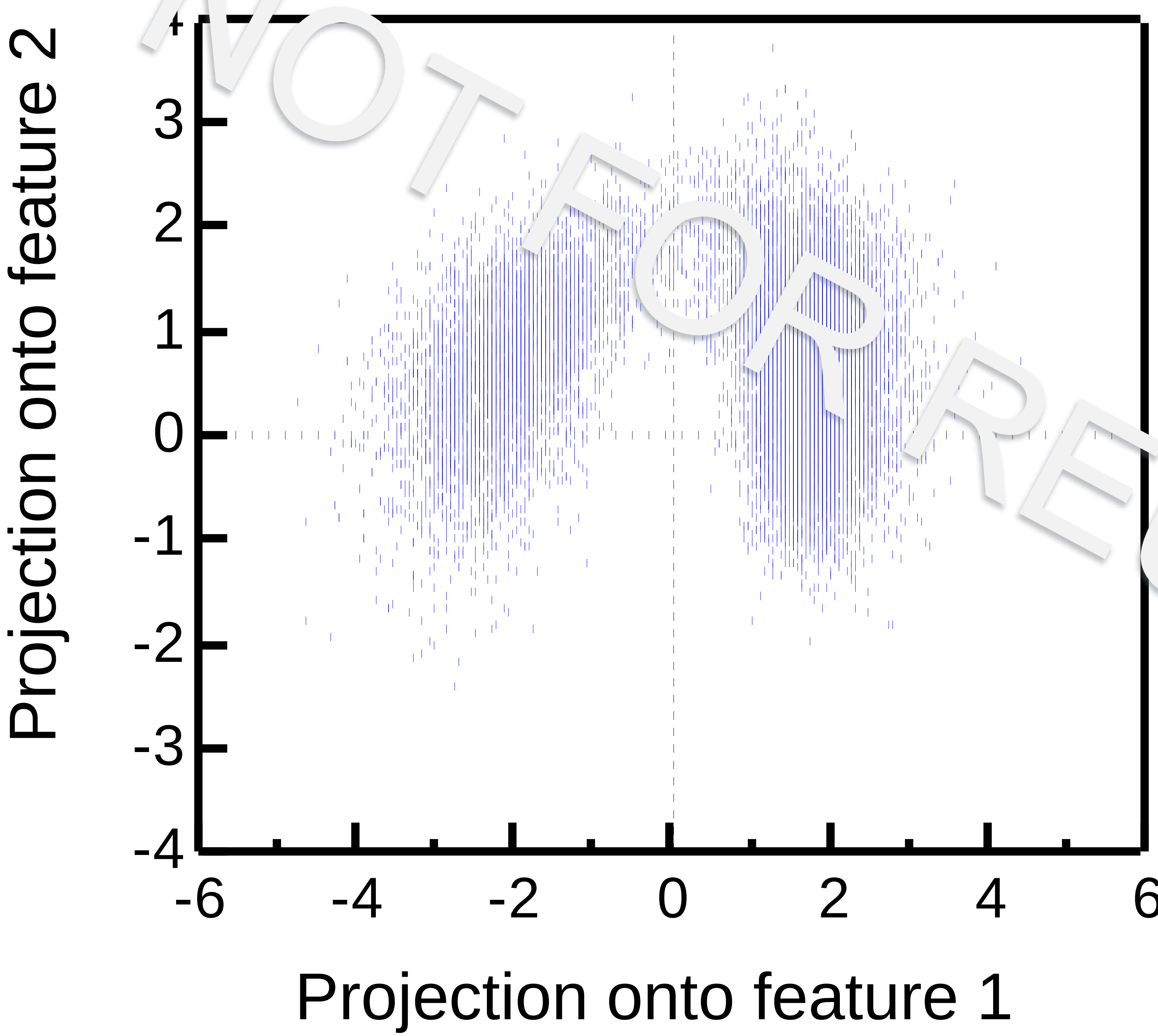
Principal component analysis: spike sorting





Koepsell et al., Front. Syst. Neurosci., 2009

Finding interesting features in the retina



Fairhall et al., J. Neurophysiol., 2007