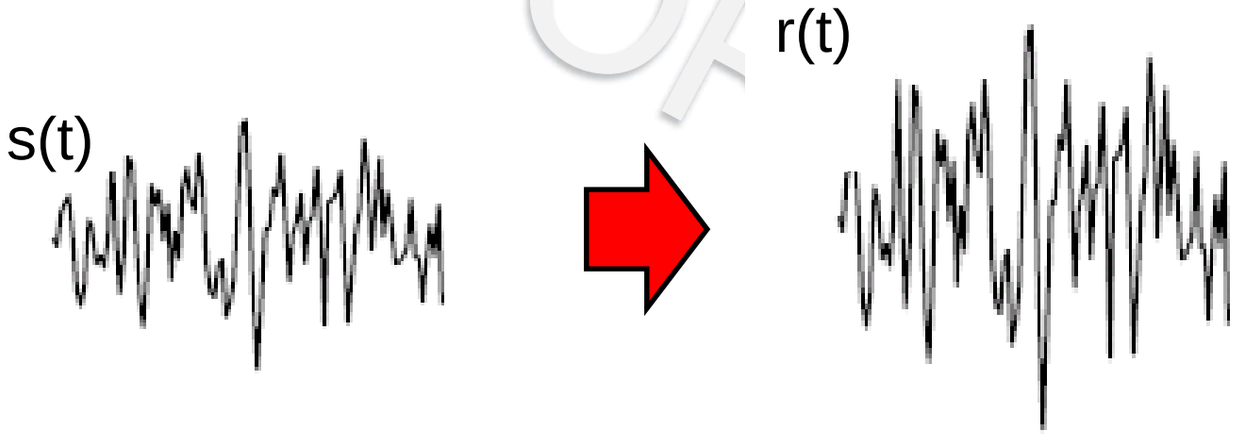
# Constructing response models

P(response | stimulus) → r(t) given a stimulus *s*

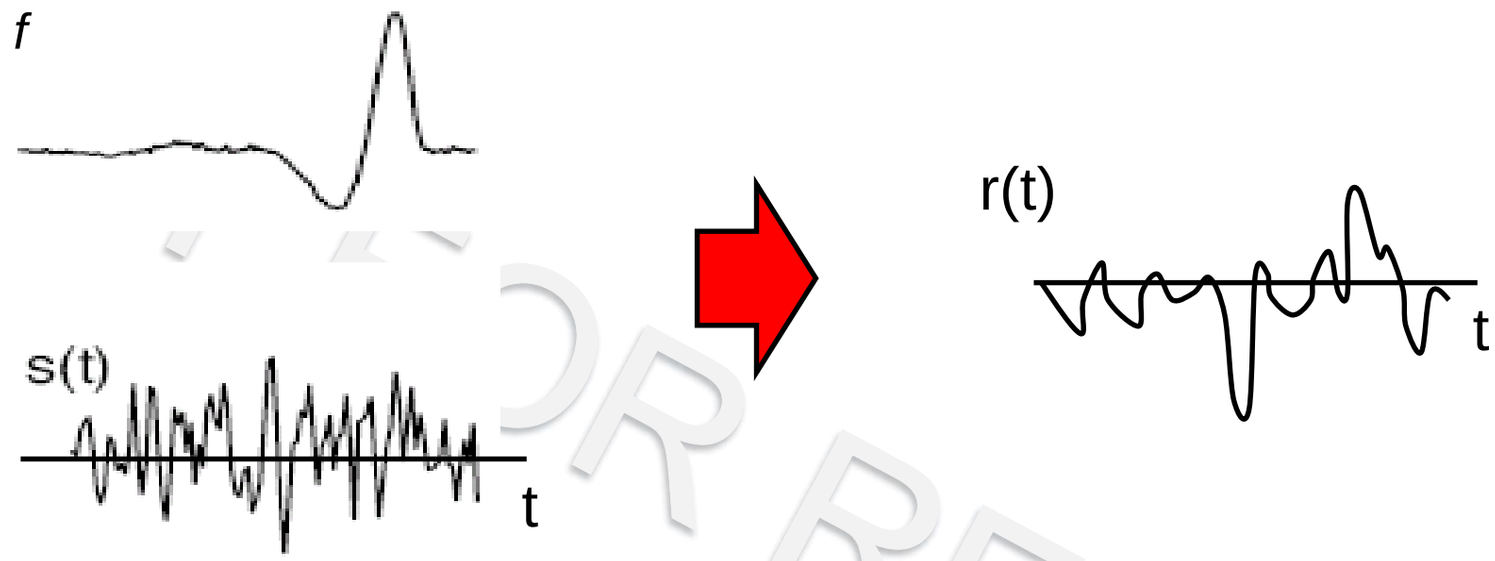
P(response | stimulus)

# 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

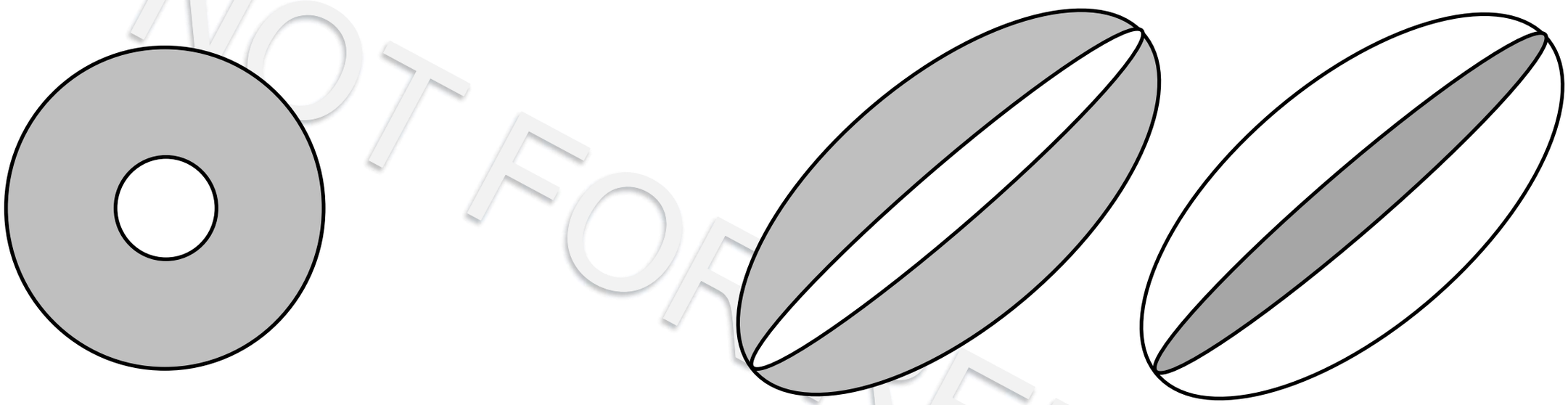

Linear filter: $$r(t) = \sum_{k=0}^{n} s_{t-k} f_k$$

# 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

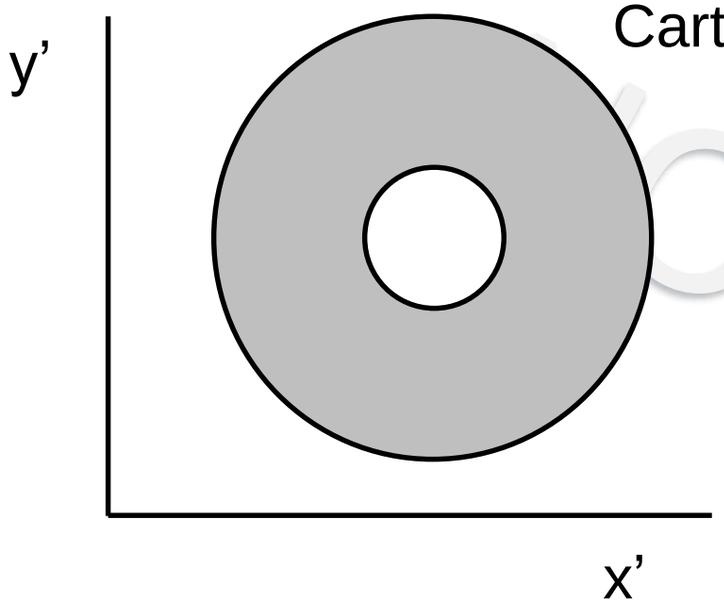
$$r(t) = \sum_{k=0}^{n} s_{t-k} f_k$$

Temporal filter

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

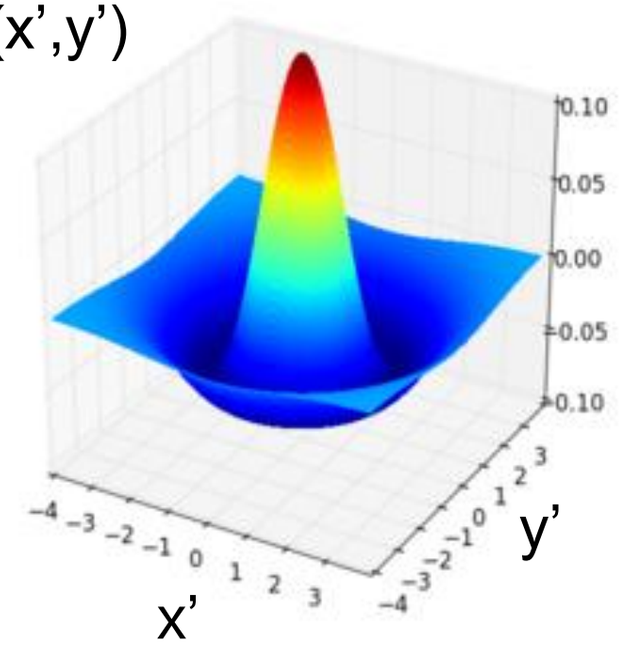
$$= \int_{-\infty}^{\infty} dx'dy'\, s(x-x', y-y') f(x', y')$$

# Spatial filtering and retinal receptive fields




Cartoon f(x',y')

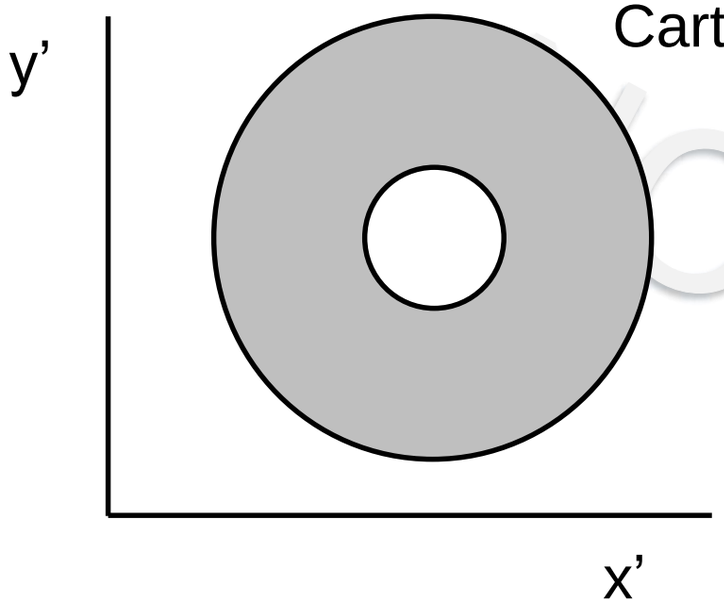
f(x',y')




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

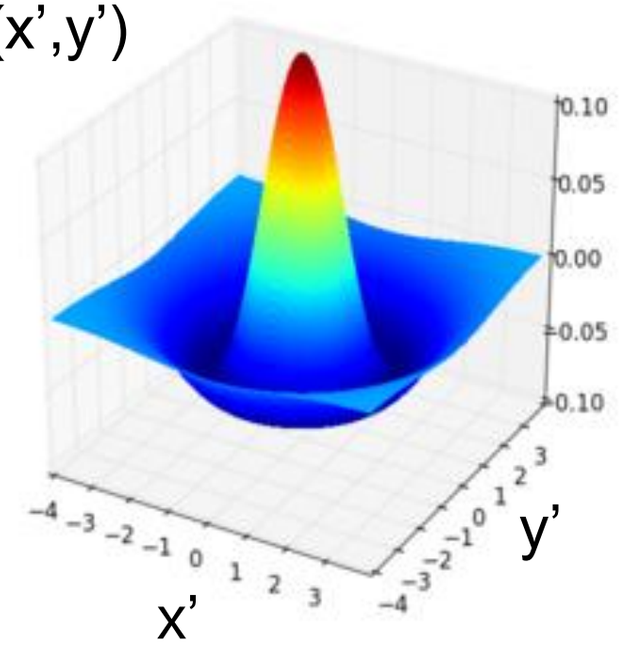
http://en.wikibooks.org/wiki/Sensory_Systems/Visual_System

# Spatial filtering and receptive fields




Cartoon f(x',y')

f(x',y')

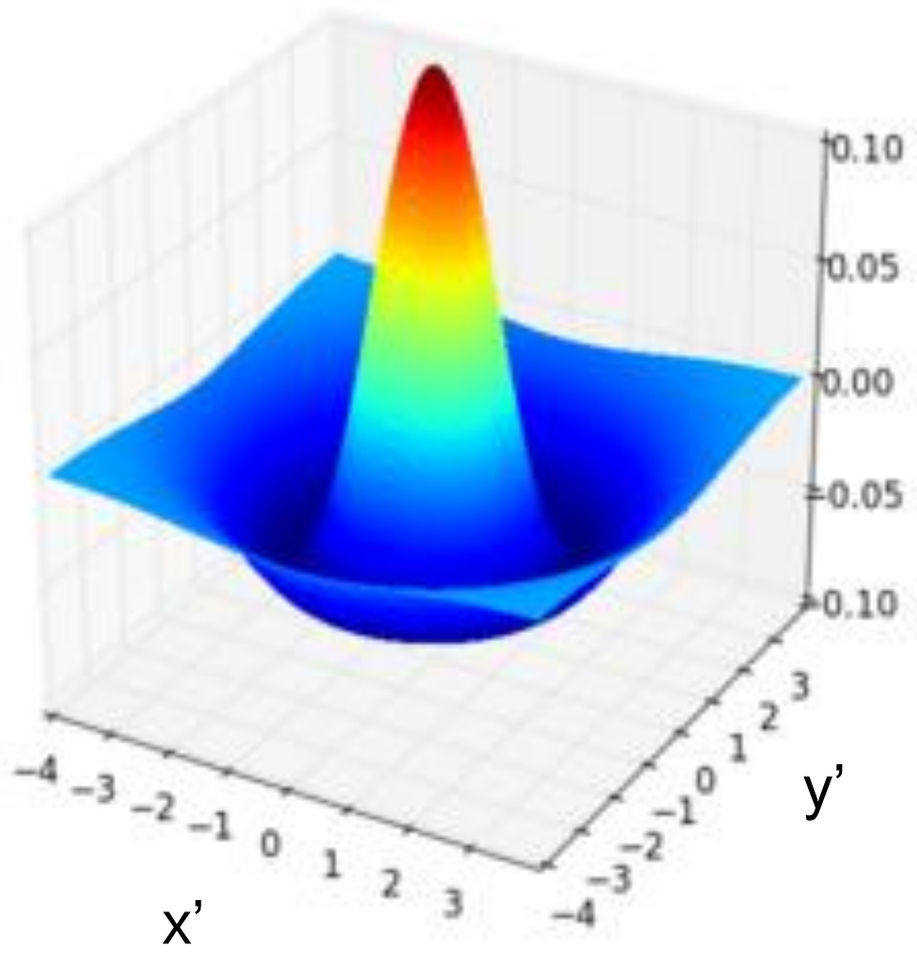



$$r(x,y) = \sum_{x\prime=-n,y\prime=-n}^{n} s_{x-x\prime,y-y\prime} f_{x\prime,y\prime}$$

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

# Spatial filtering and receptive fields

f(x’,y’)






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

# Spatial filtering

7.2.1. Overview



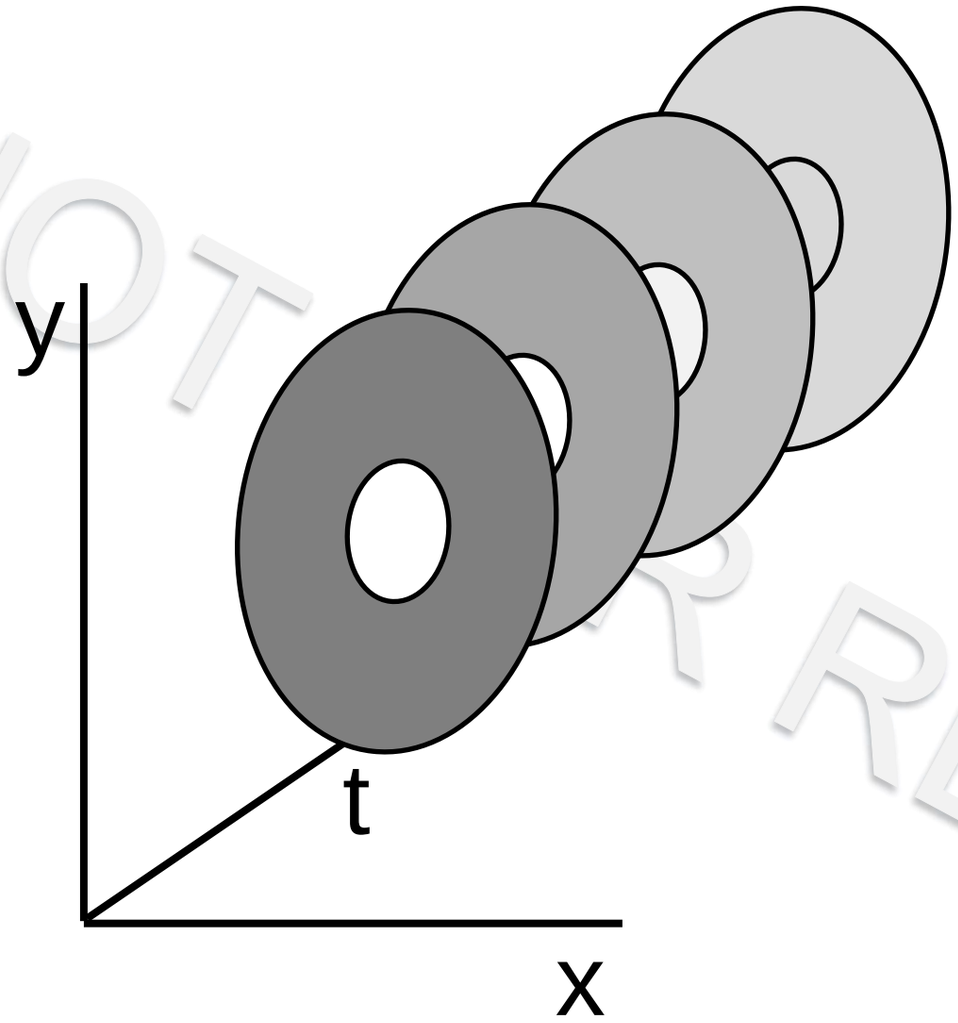
**Figure 16.136. Applying example for the "Difference of Gaussians" filter**

Original image

Filter "Difference of Gaussians" applied

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

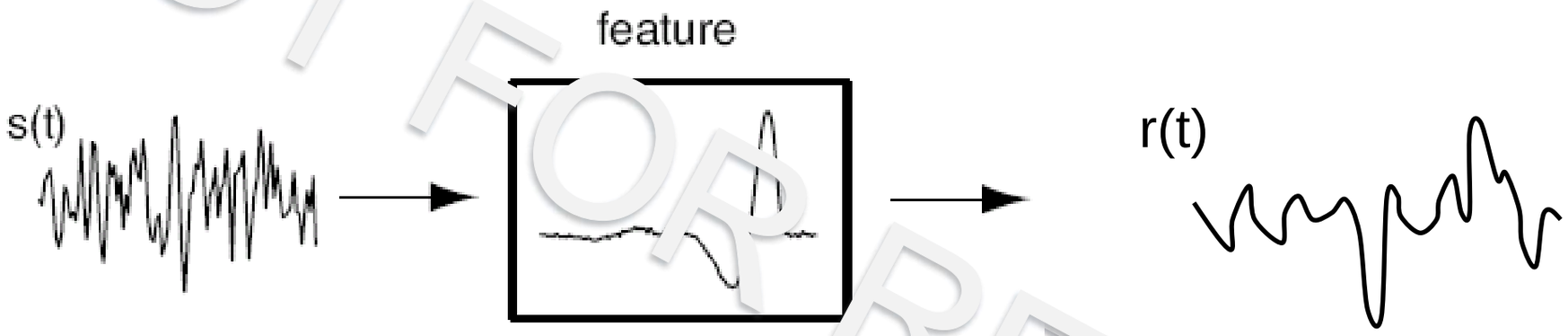
# Basic coding model: *spatiotemporal* filtering






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

# Basic coding model: temporal filtering

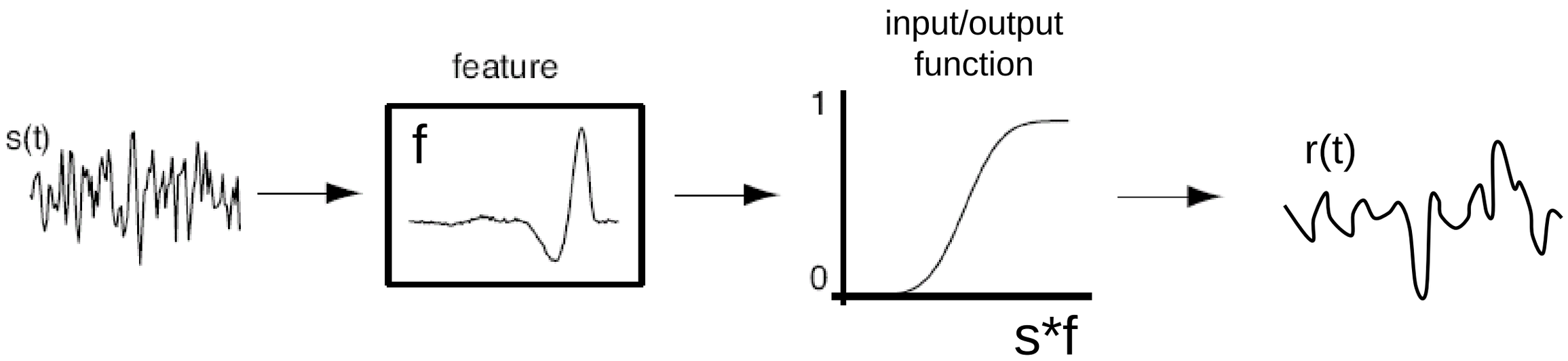





Linear filter: $r(t) = \int s(t-\tau)\, f(\tau)\, d\tau$

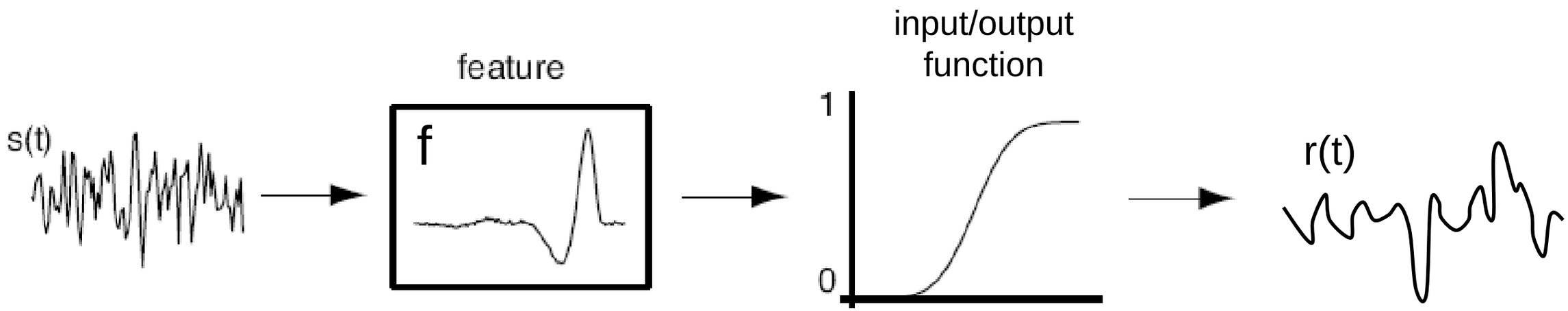
…shortcomings?

# Next most basic coding model



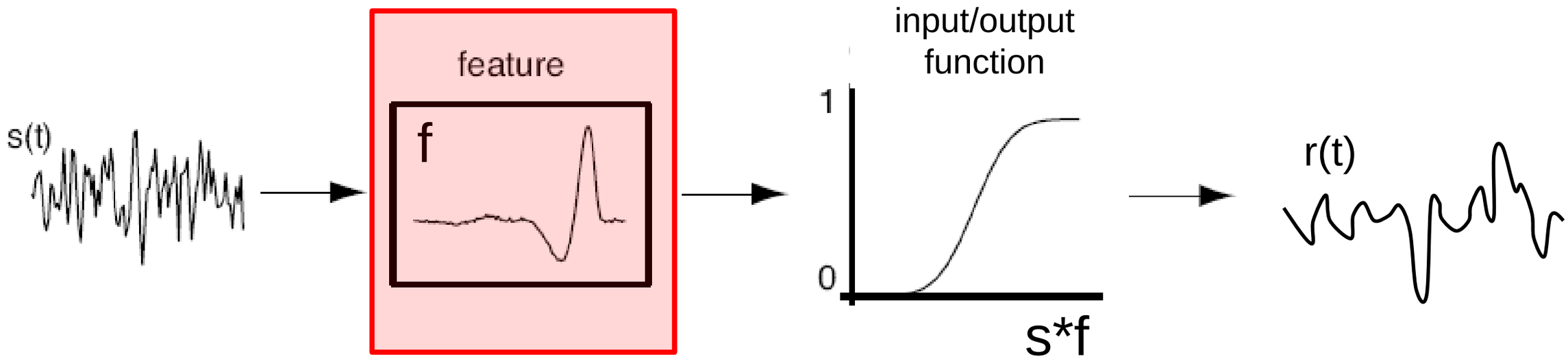

Linear filter & nonlinearity: $r(t) = g(\int s(t-\tau)\, f(\tau)\, d\tau)$

# How to find the components of this model




s*f

# How to find the components of this model



# How to proceed?

P(response | stimulus)

Our problem is one of dimensionality!

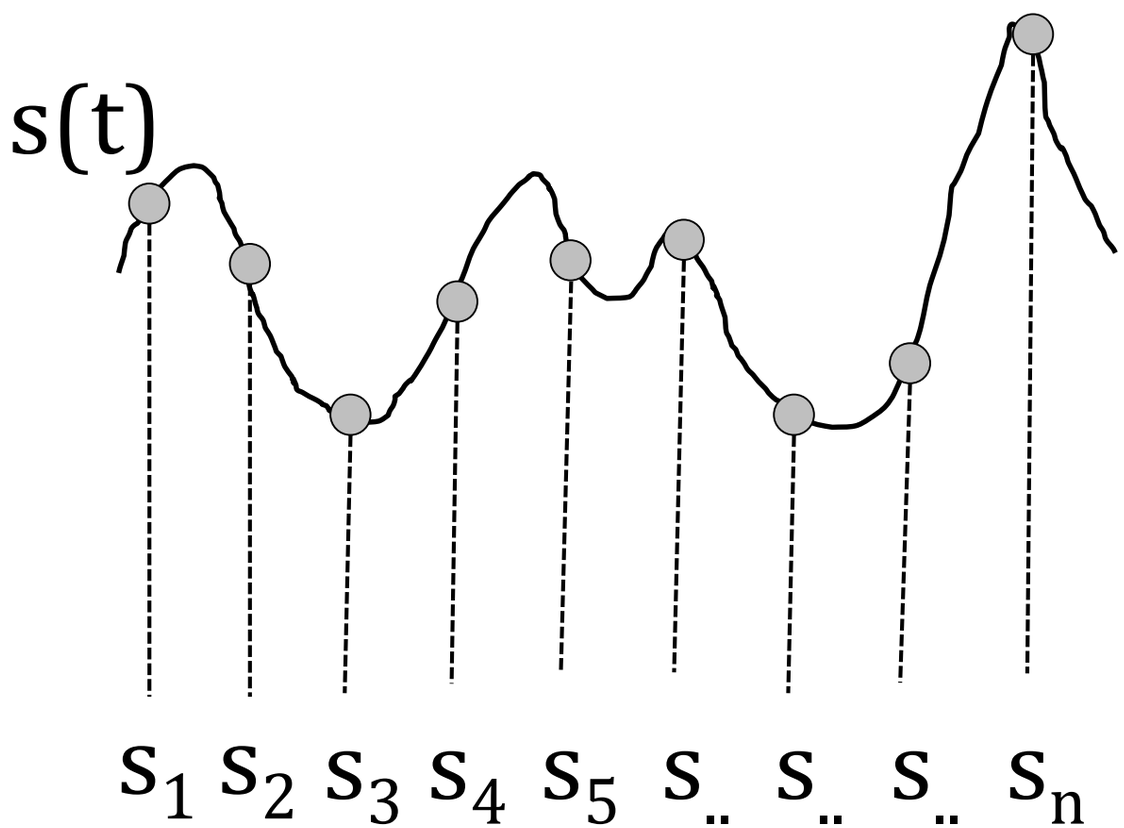
We want to sample the responses of the system to many stimuli so we can characterize what it is about the input that triggers responses.

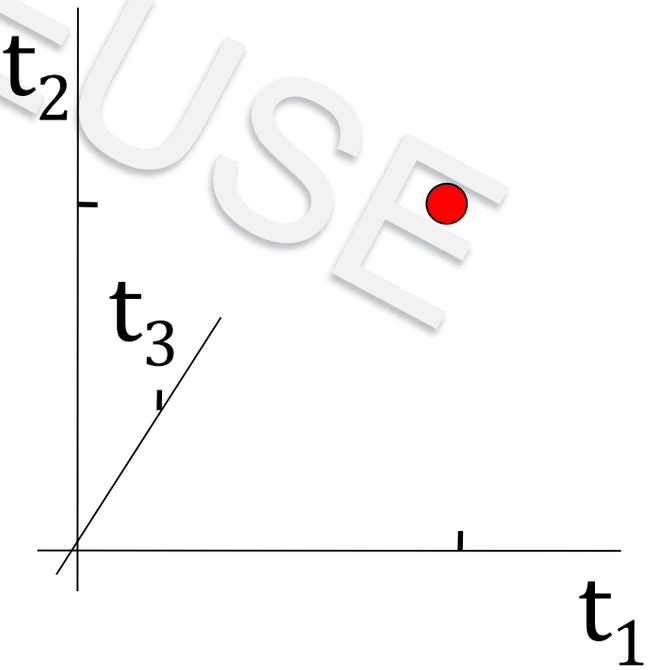
P(response | stimulus) → P(response | $s_1$)

# 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.





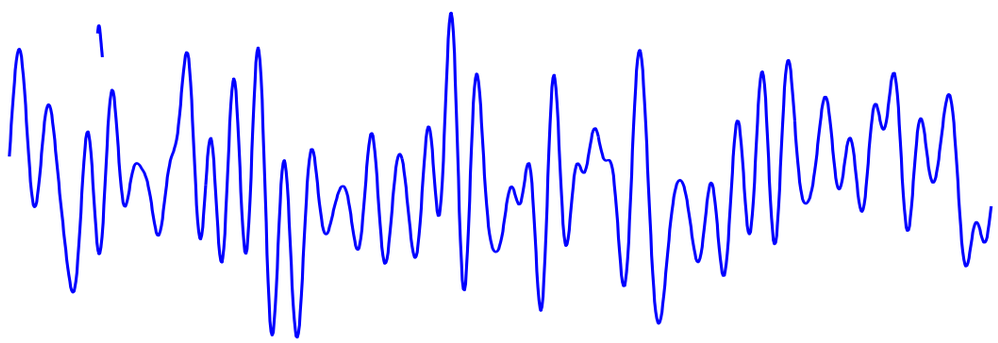

$s(t) = (s_{t1}, s_{t2}, s_{t3}, \ldots, s_{tn})$

# What is the right stimulus to use?

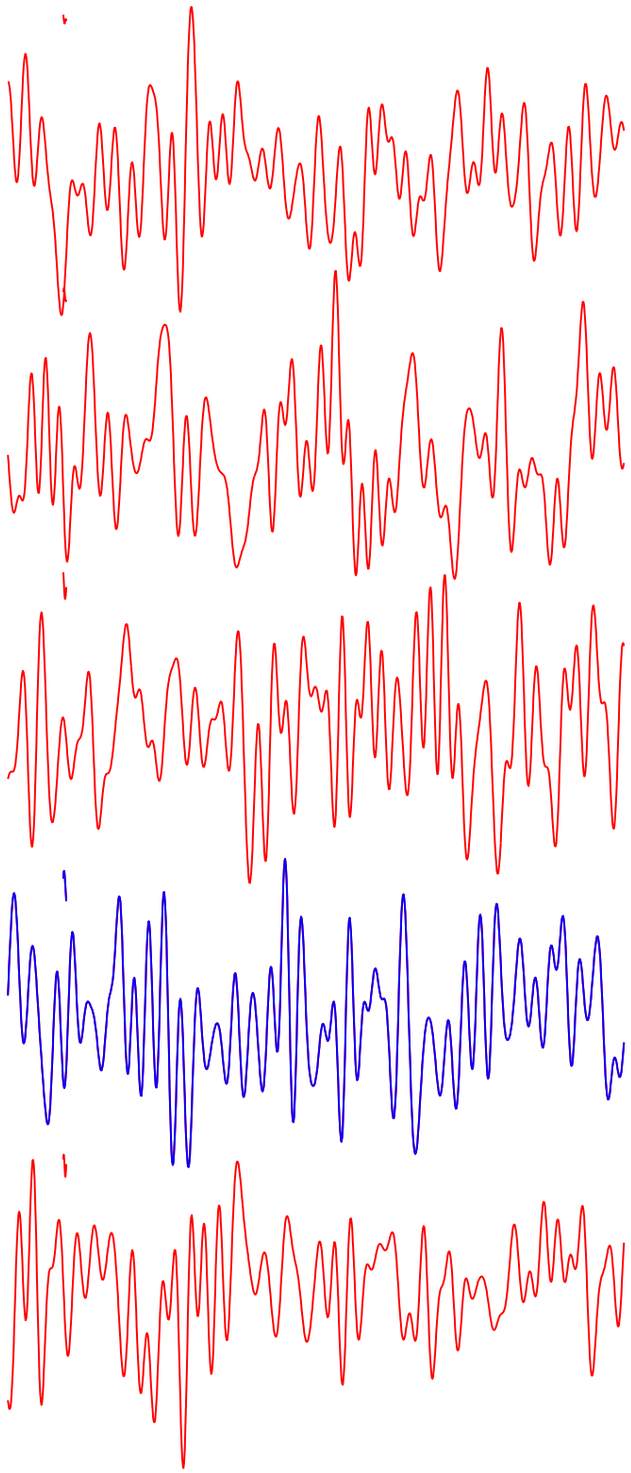
We want to sample the responses of the system to a variety of stimuli so we can characterize what it is about the input that triggers responses.

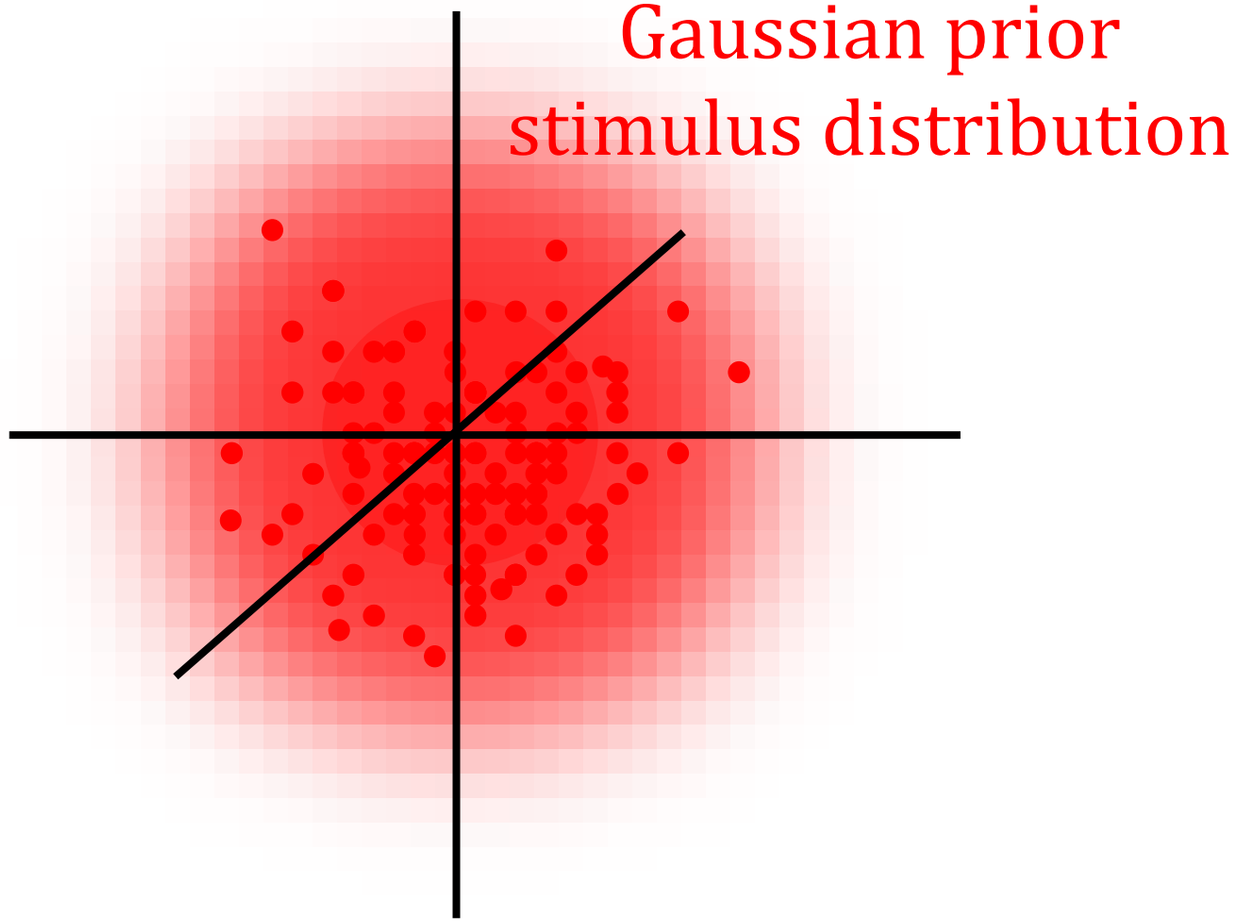
$$P(\text{response} \mid \text{stimulus}) \rightarrow P(\text{response} \mid s_1, s_2, \ldots, s_n)$$

One common and useful method is to use **white noise**

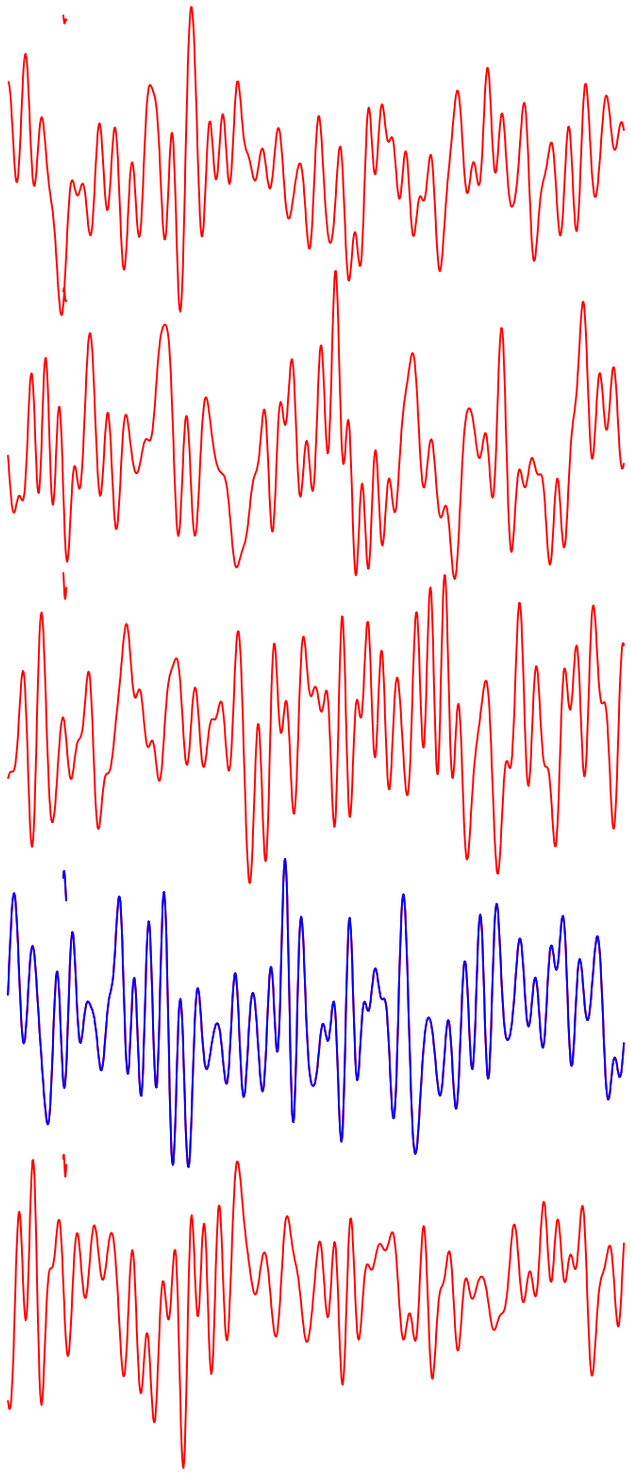


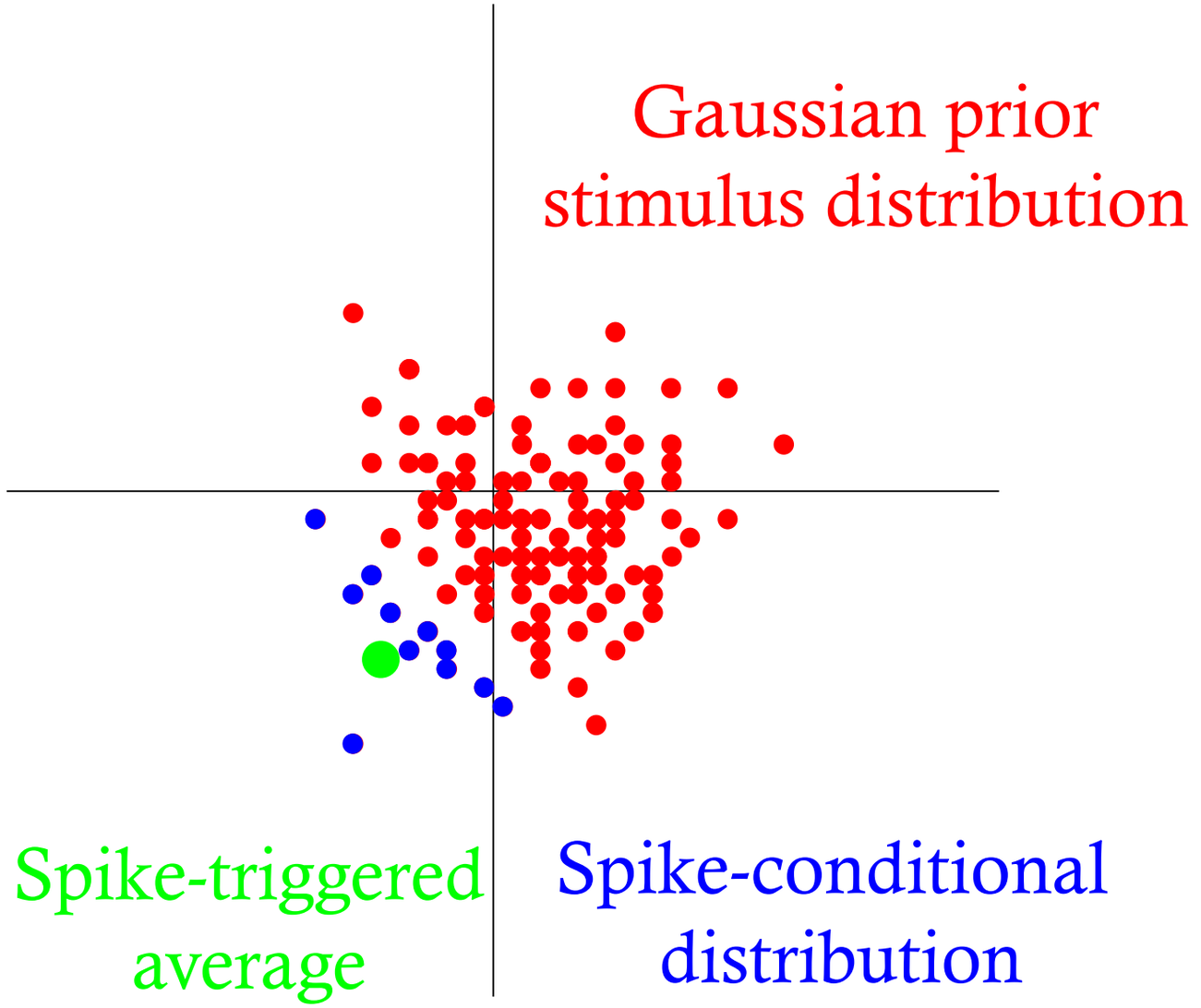
# Determining multiple features from white noise



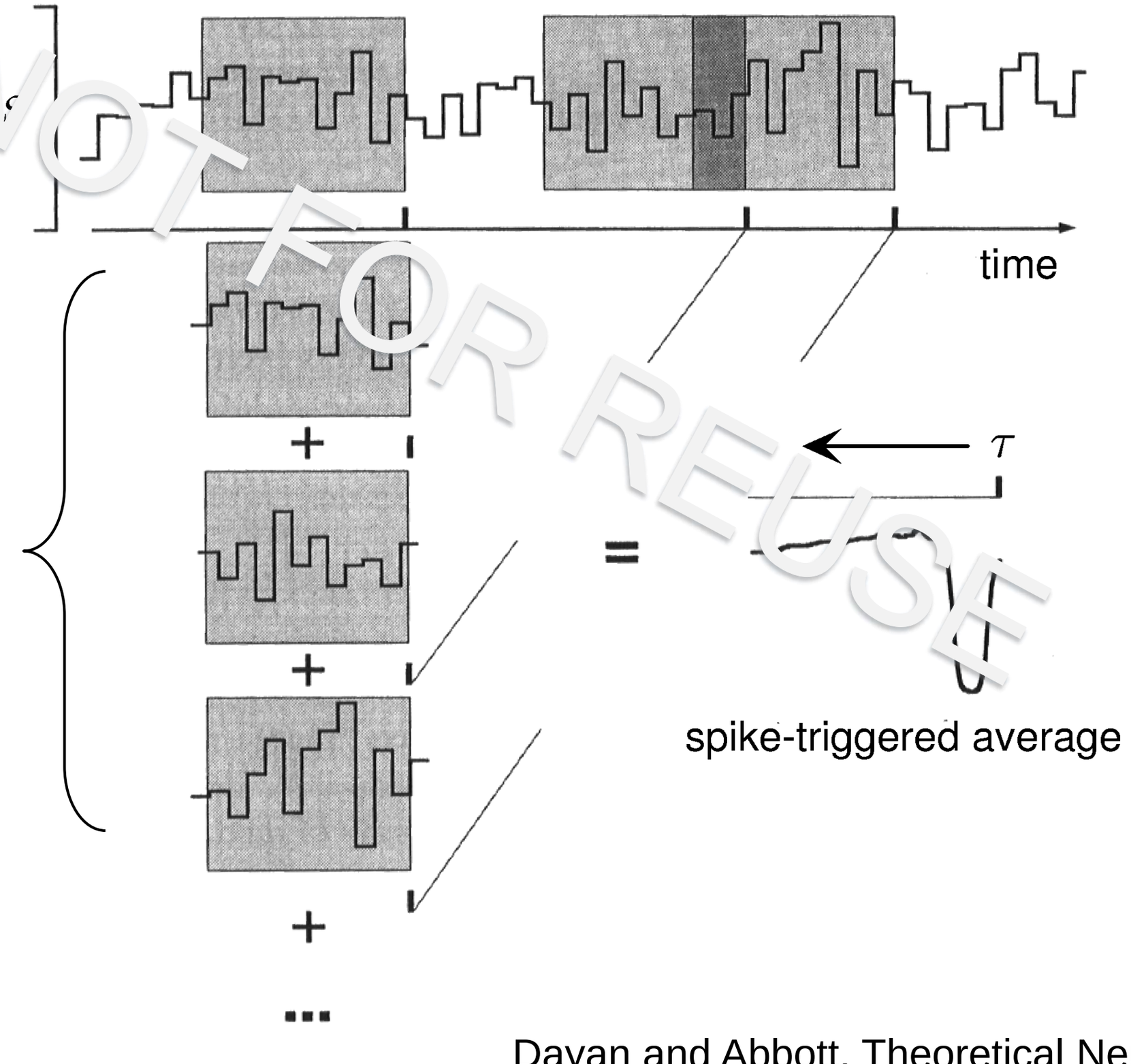


# Determining linear features from white noise





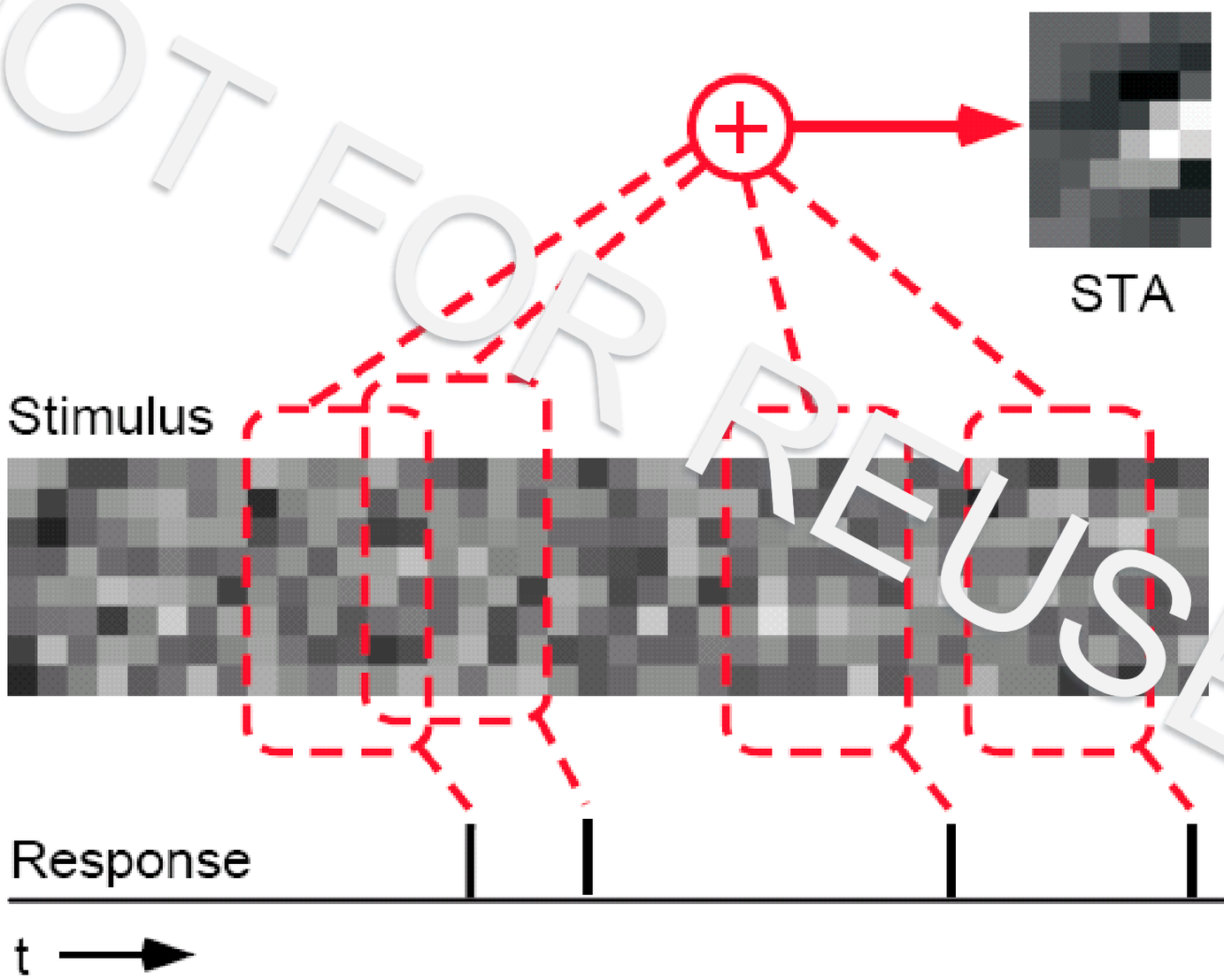
# Reverse correlation: the spike-triggered average





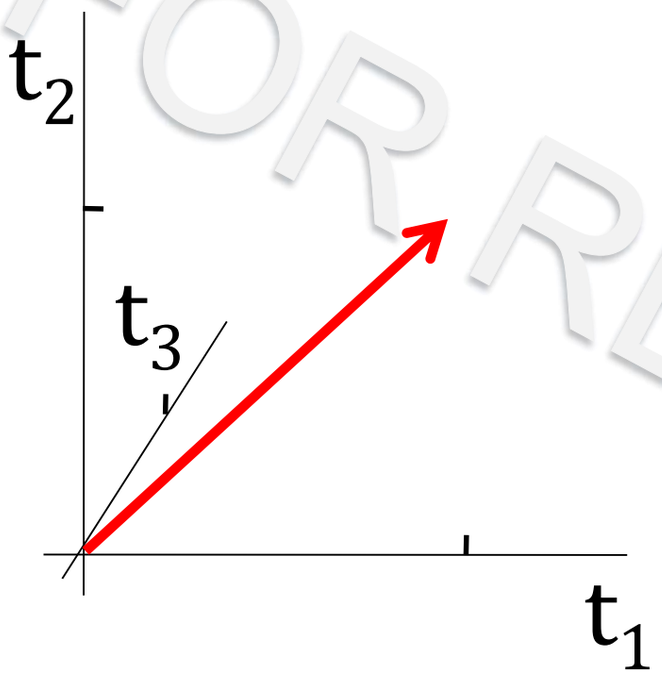

Dayan and Abbott, Theoretical Neuroscience

# The spike-triggered average



# Linear filtering

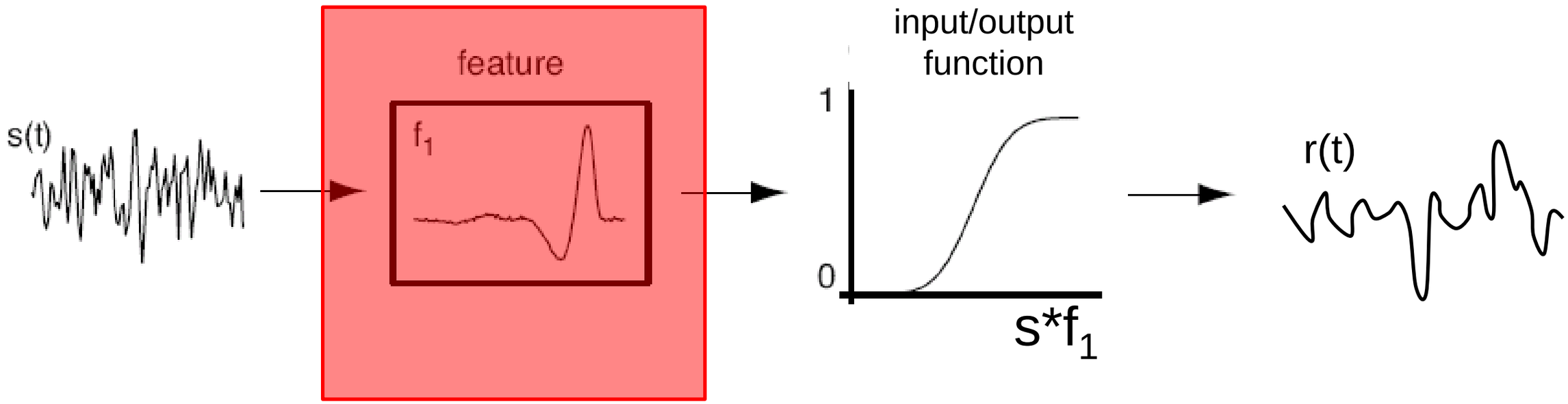
Stimulus feature *f* is a vector in a high-dimensional stimulus space






Linear filtering = convolution = projection

# How to find the components of this model



# Determining the nonlinear input/output function

The input/output function is:

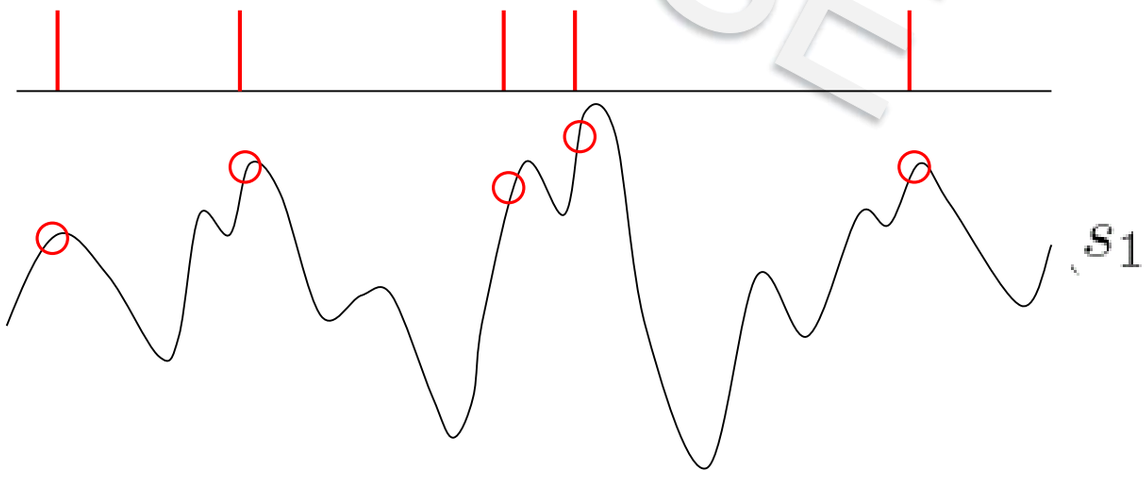
$$P(\text{spike}|\text{stimulus}) \Rightarrow P(\text{spike}|s_1)$$

This can be found from data using Bayes' rule:

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

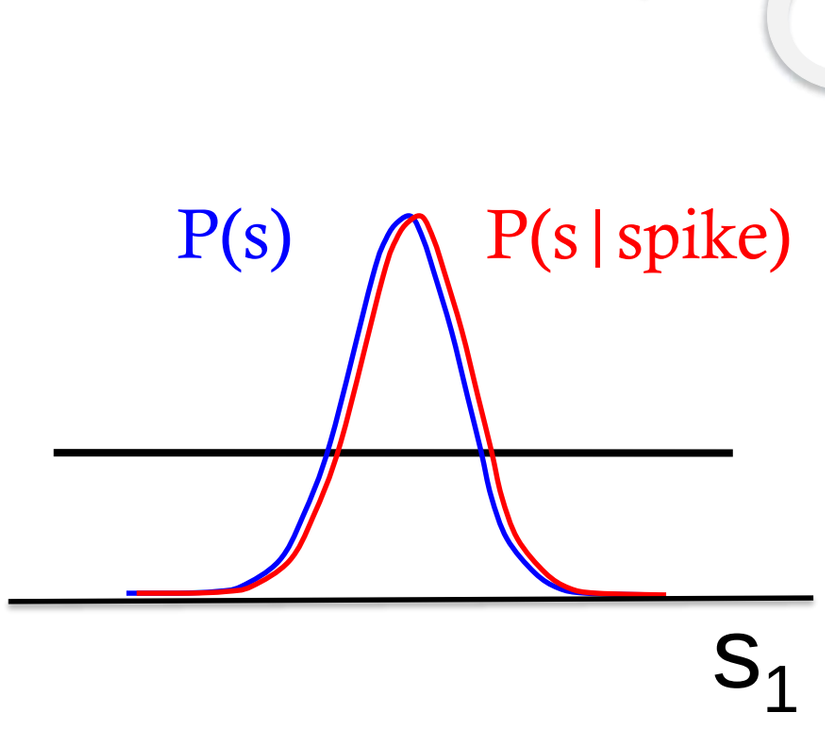
$P(s_1)$

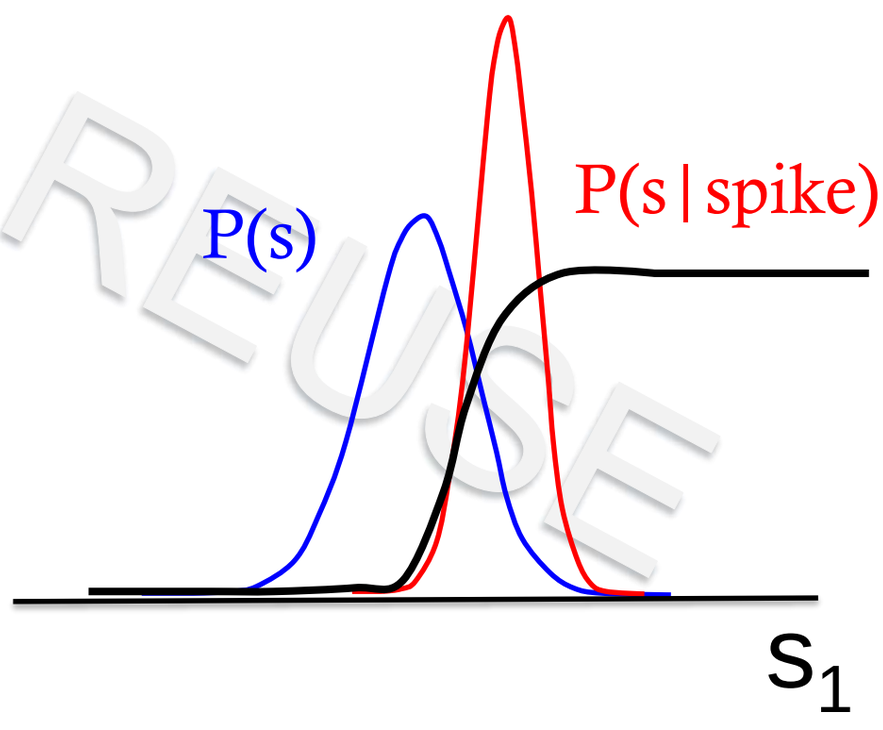
$P(s_1|\text{spike})$



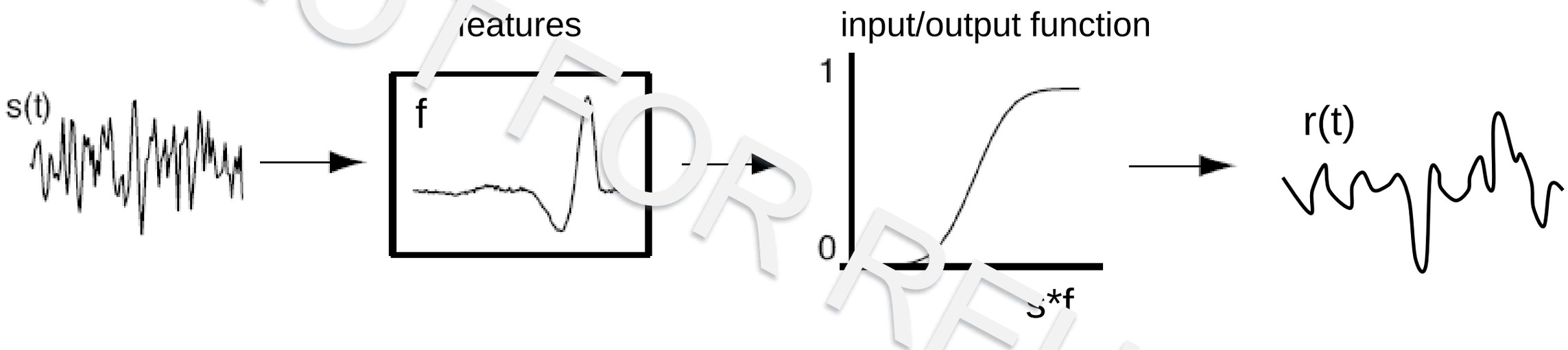
# Nonlinear input/output function

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



# Linear/nonlinear models






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

# High-dimensional feature selection


**Featured Members**

**Auntie_Sassy**

**Age:** 35
**Location:** Greenwood

**Woman** seeking
- Man for Dating
- Man for Friendship

**Worst Haiku Ever**

This is my first dip into the online dating pool and quite frankly, I have no idea what I'm doing.... learn more about me »

**JohnnyX**

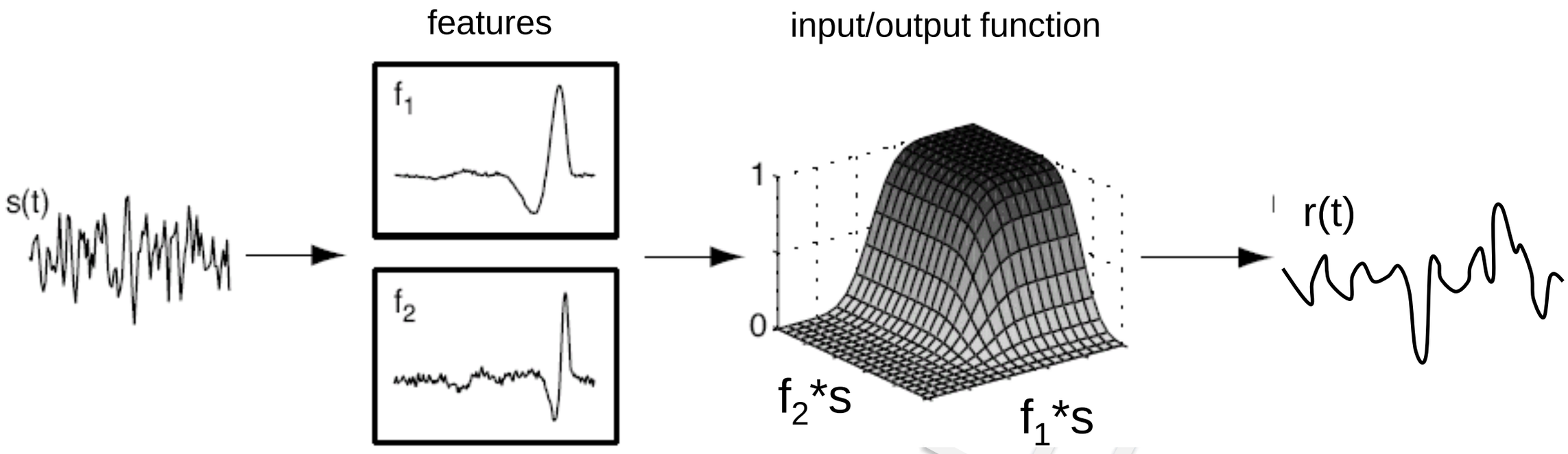
**Age:** 47
**Location:** Capitol Hill

**Man** seeking
- Woman for Dating
- Woman for Friendship

**Sex, Love and Rock-n-Roll**

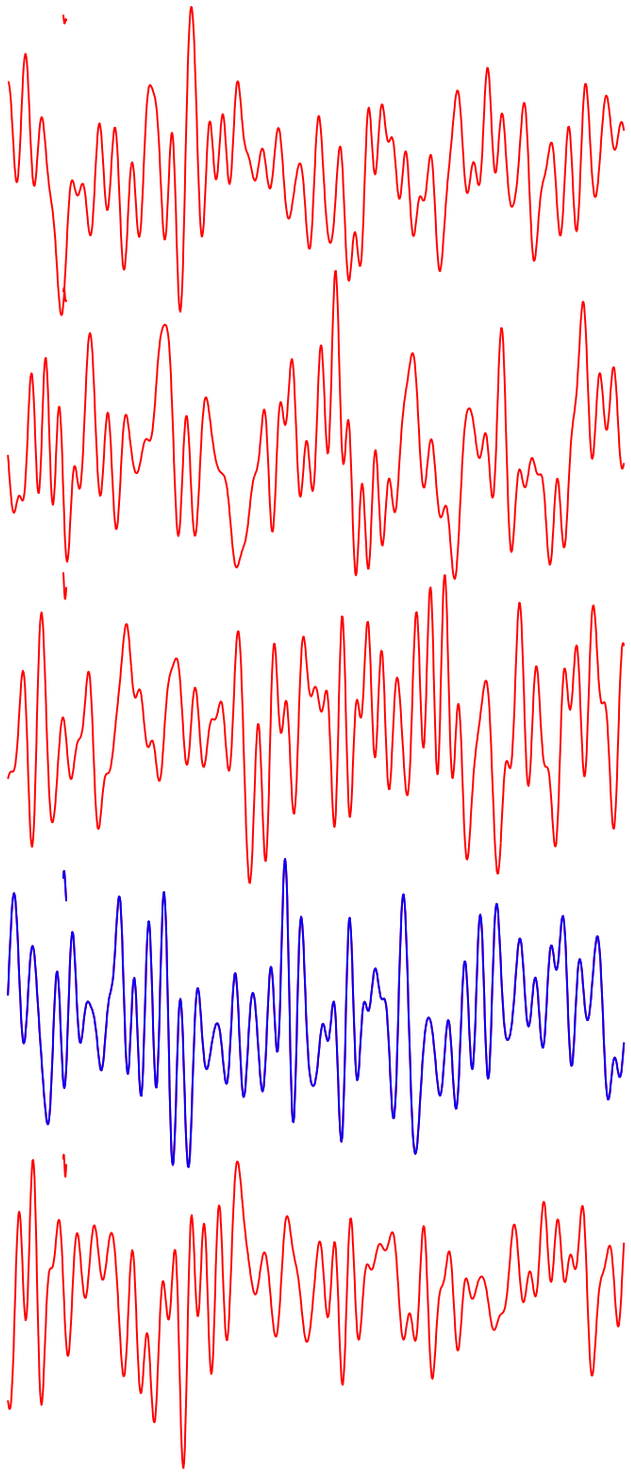
If you don't see how it possible for an older guy to be sexy and exciting, stop reading now because... learn more about me »

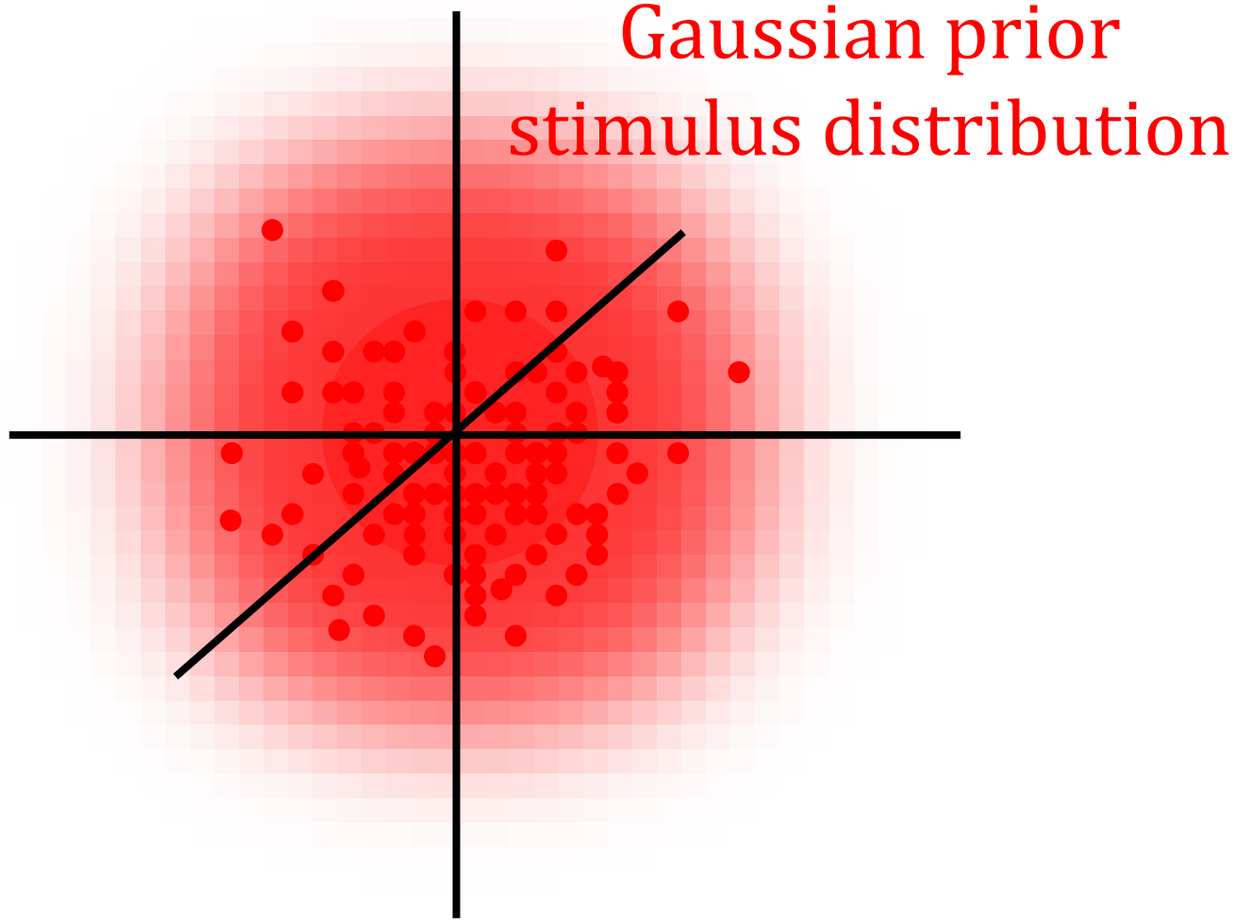
# Less basic coding models



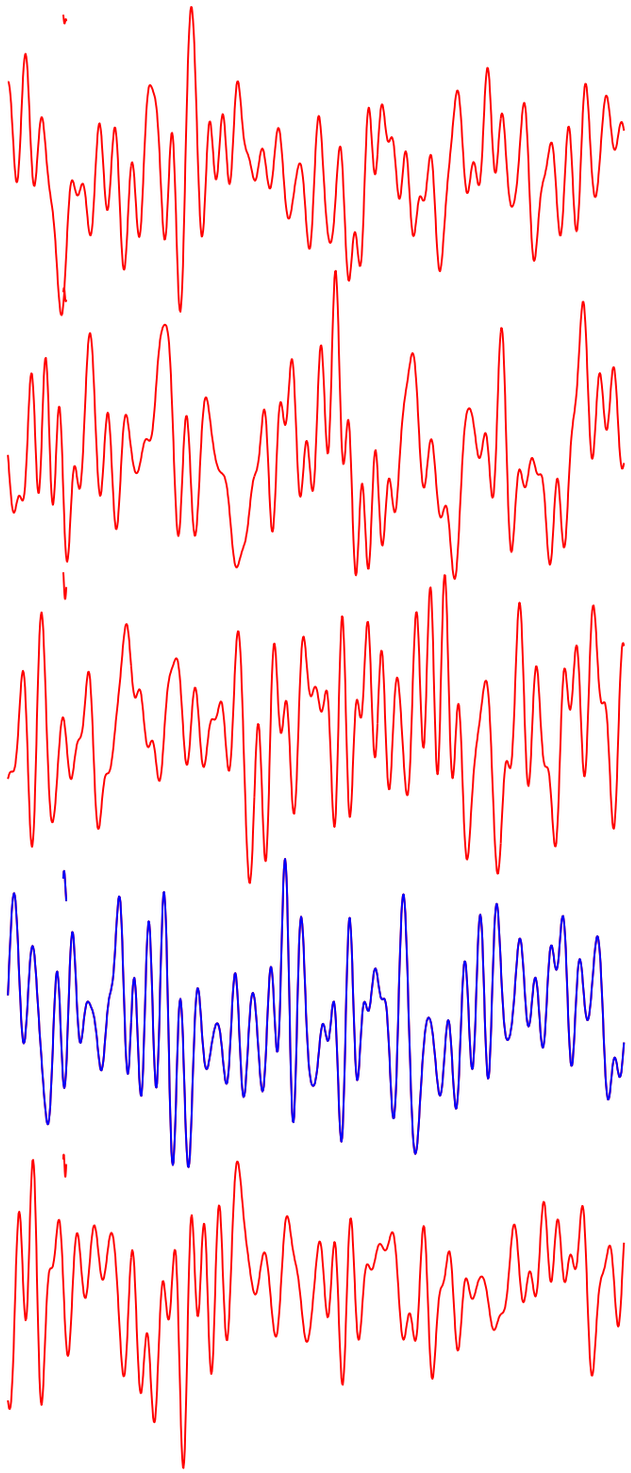

Linear filters & nonlinearity: $r(t) = g(f_1*s, f_2*s, \ldots, f_n*s)$

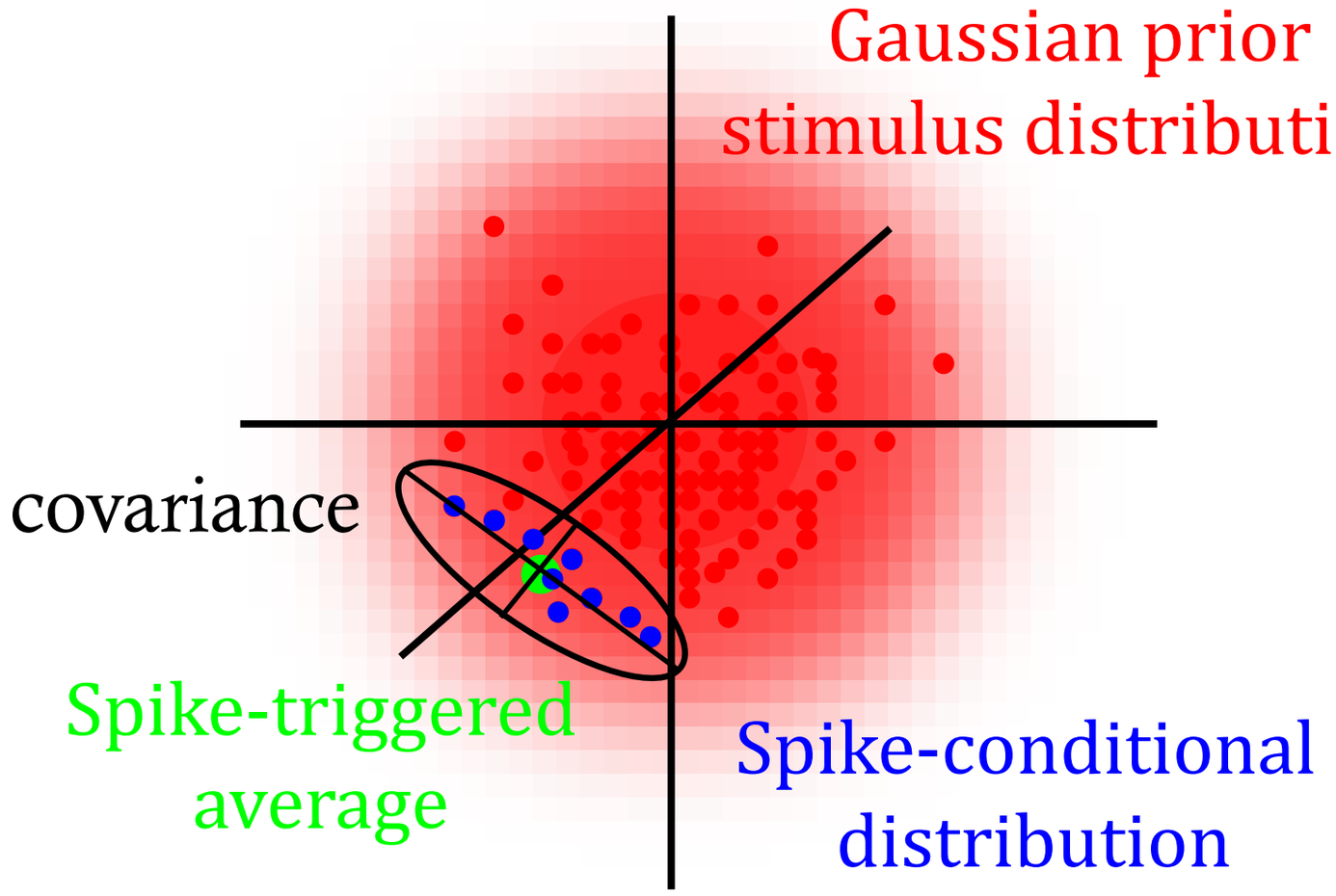
# Determining multiple features from white noise



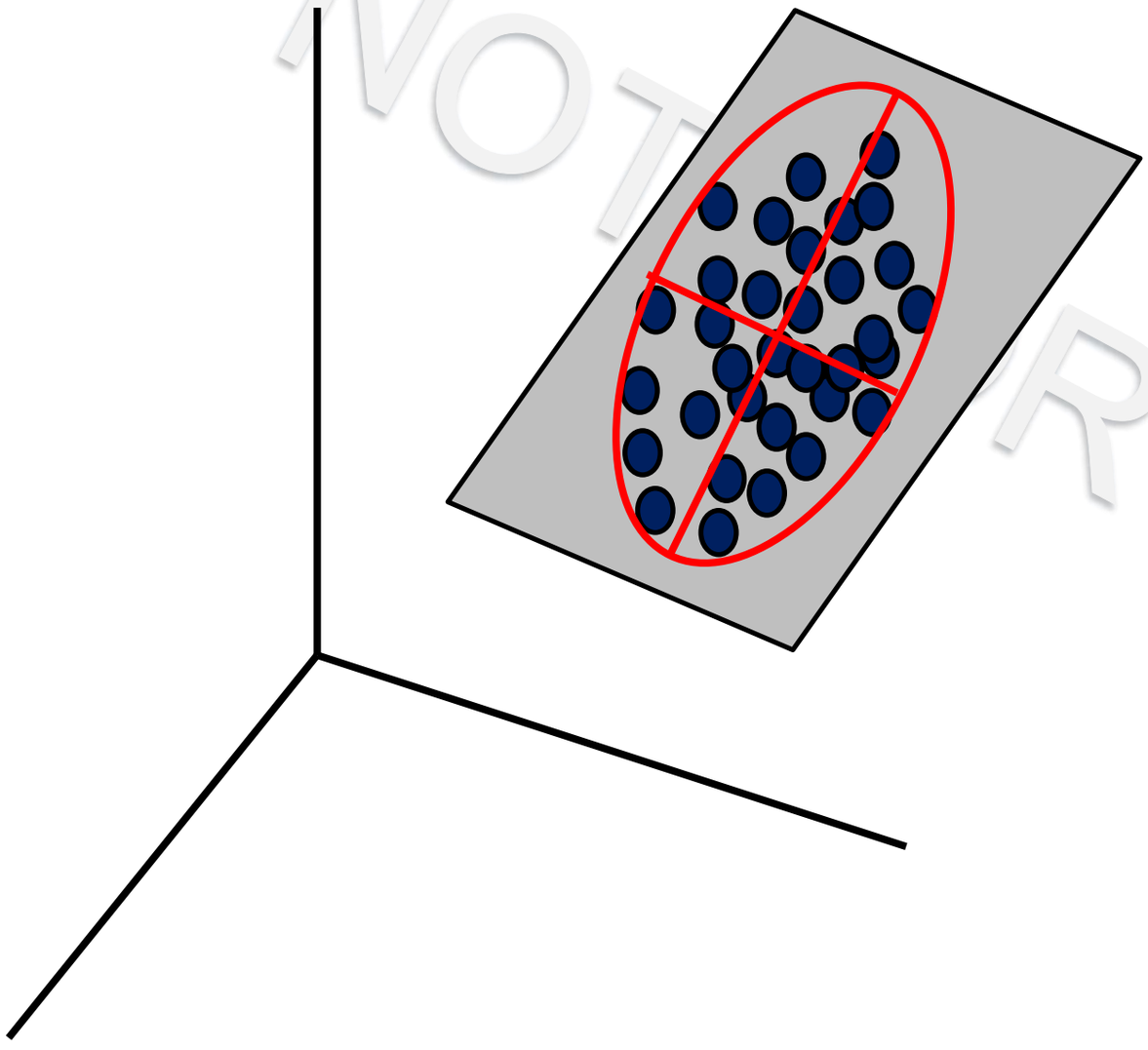


# Determining multiple features from white noise

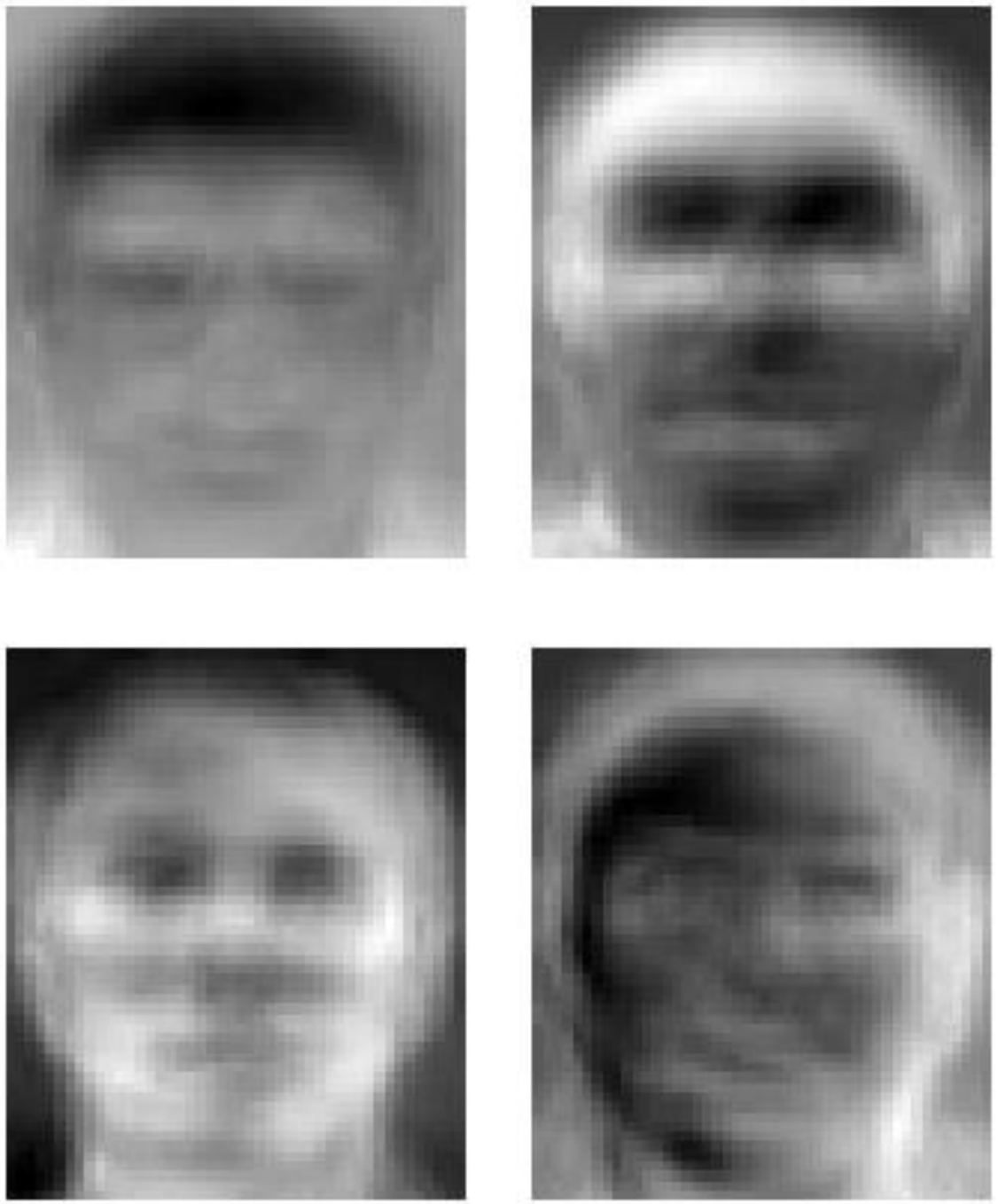




# Principal component analysis

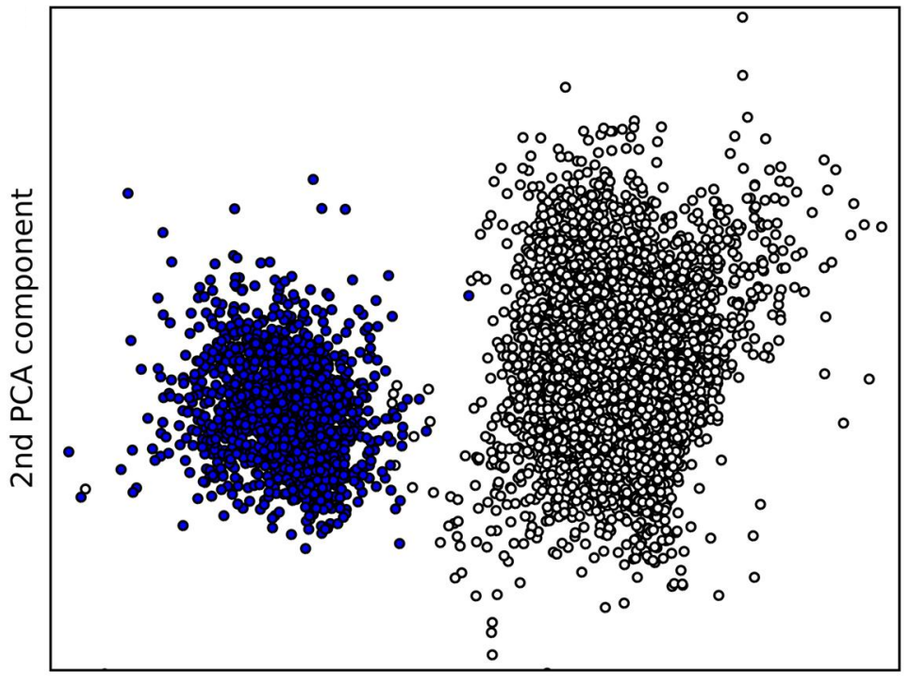


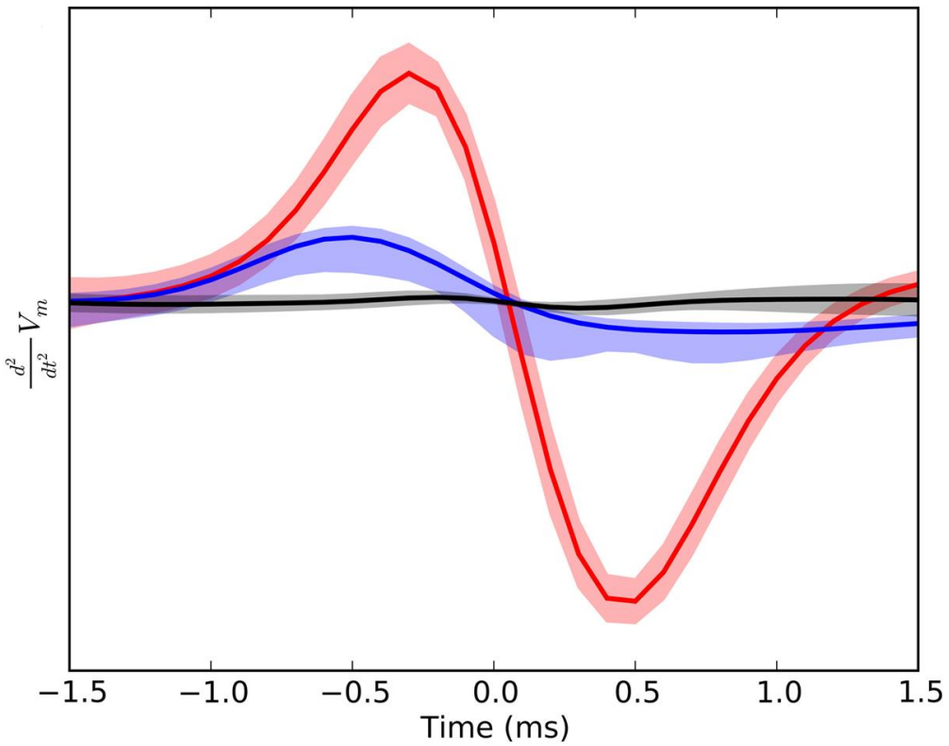
# Principal component analysis: eigenfaces



ATT Labs, Cambridge (via Wikipedia)

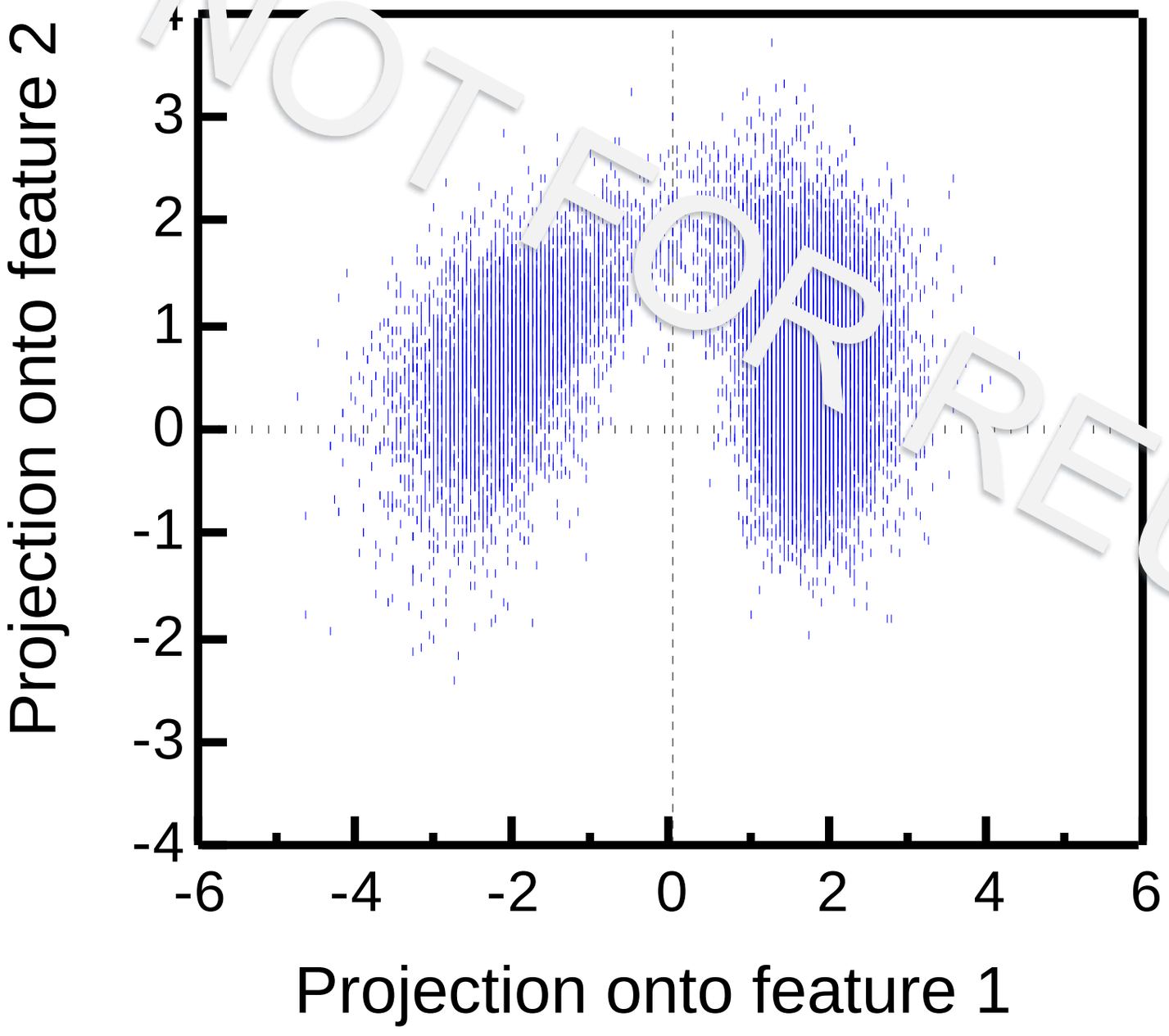
# Principal component analysis: spike sorting




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

# Finding interesting features in the retina






Fairhall et al., J. Neurophysiol., 2007