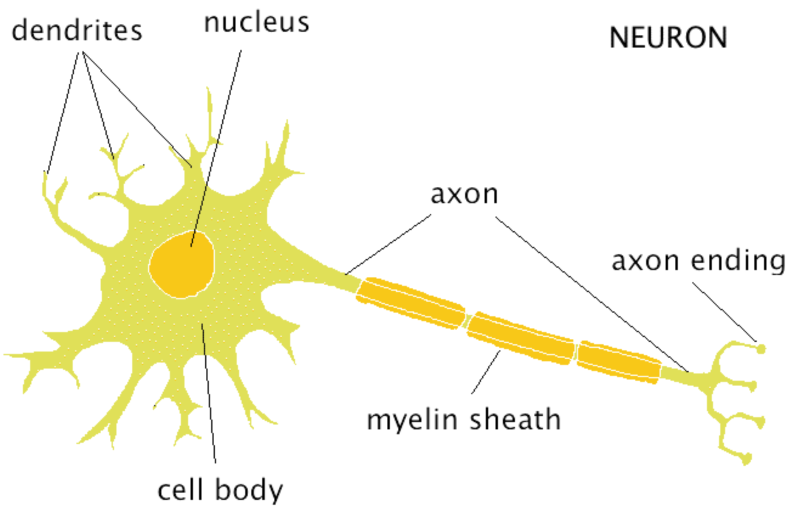
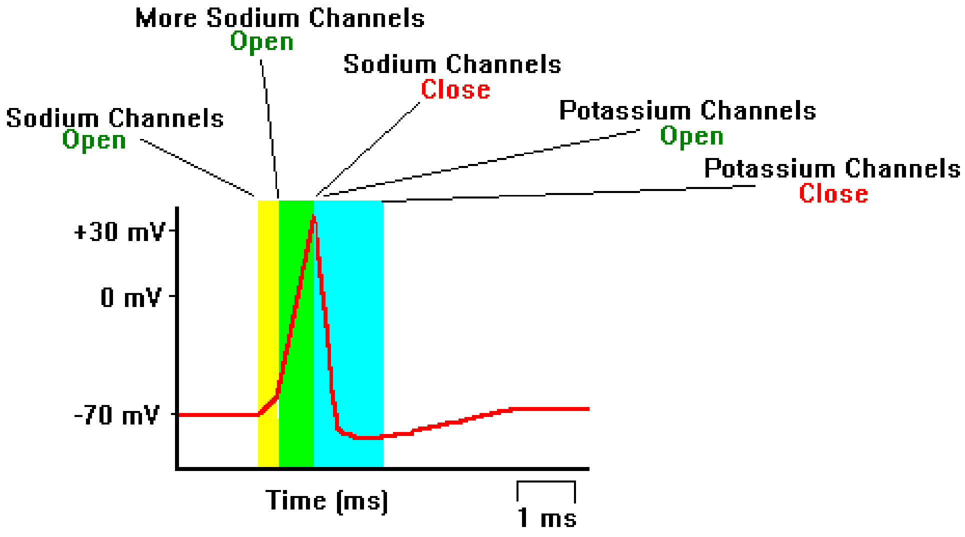
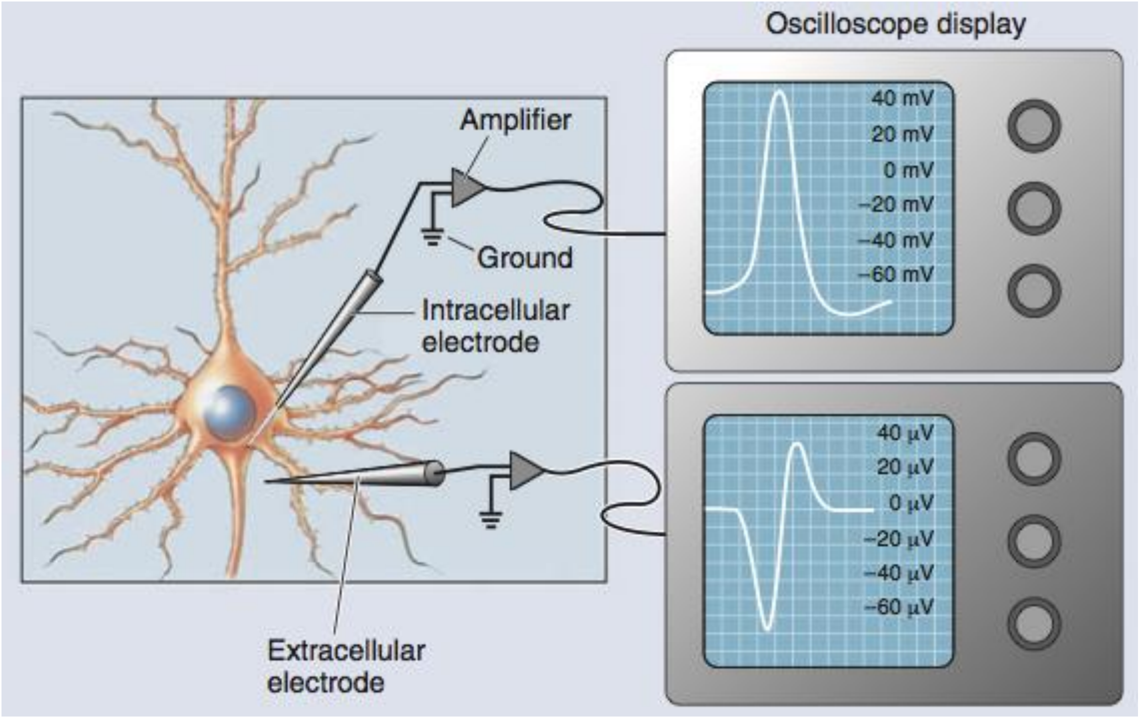
What this course will not cover

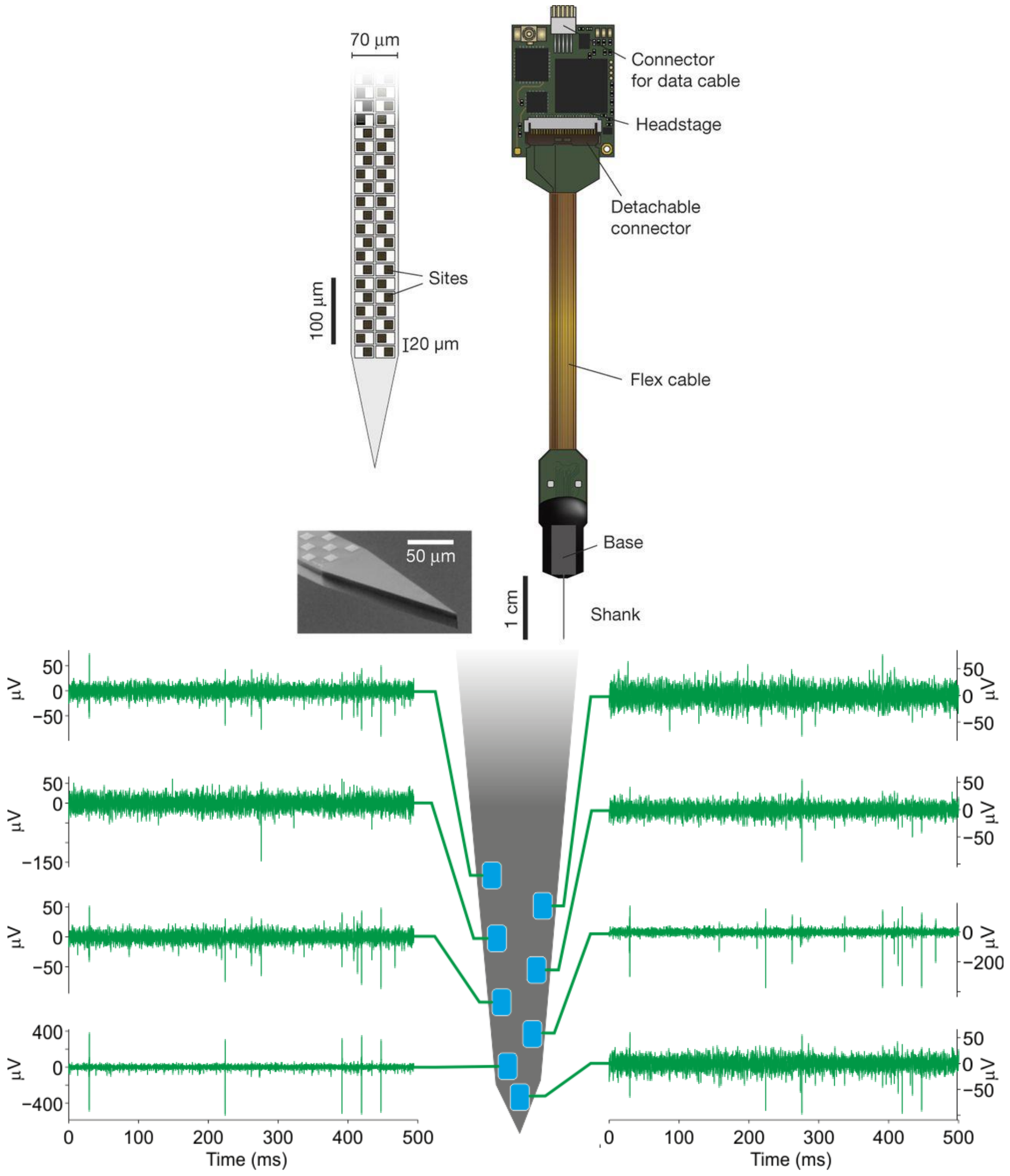
Action potentials (spikes) create electrical signals in the brain

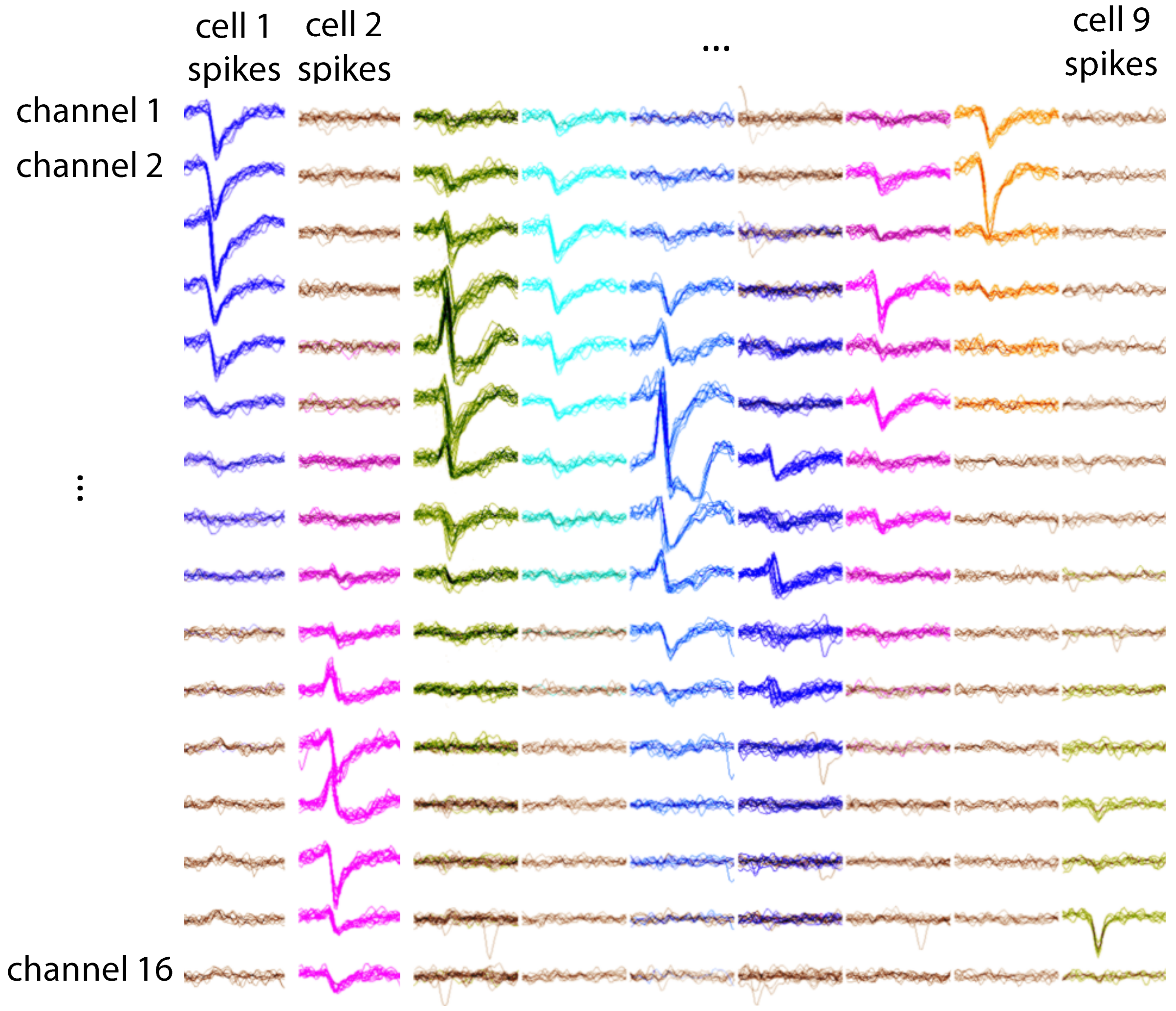






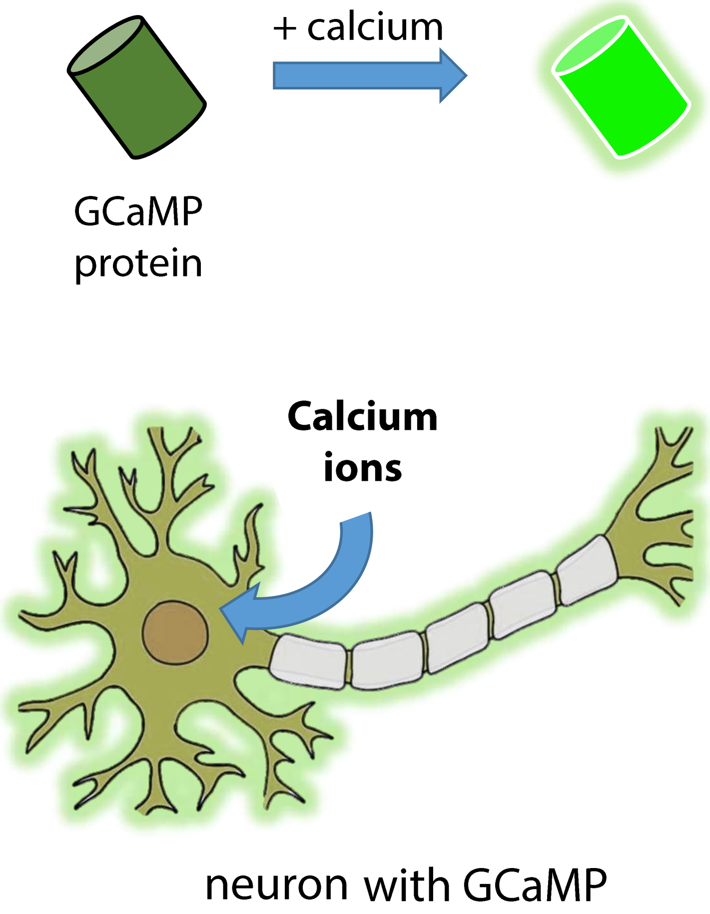
Electrophysiology: recording electrical signals from nearby neurons

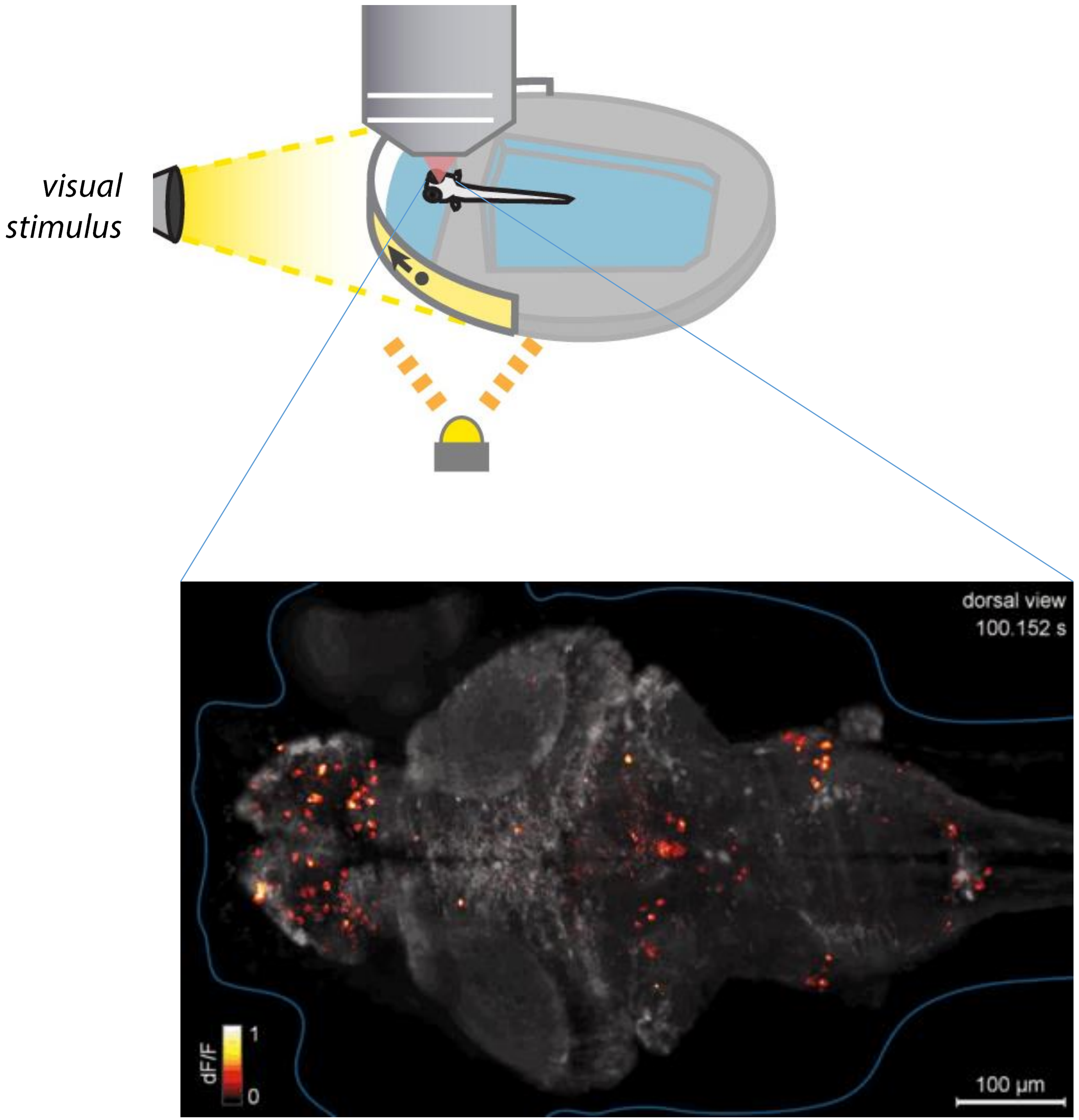




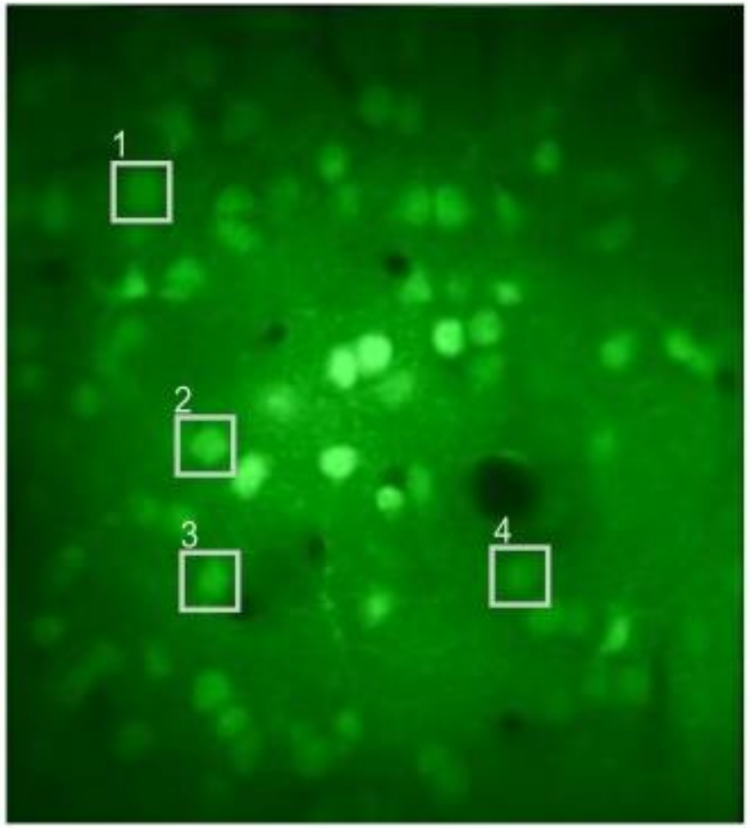
spike sorting

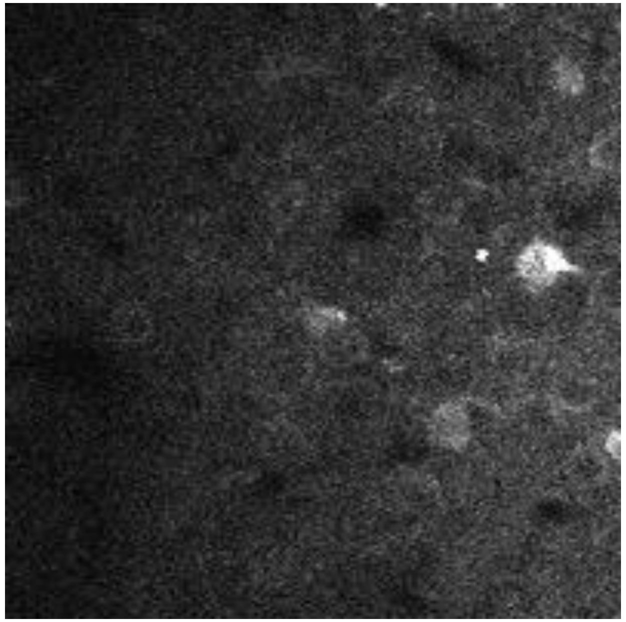
Calcium imaging: recording the electrical activity of neurons using engineered fluorescent proteins





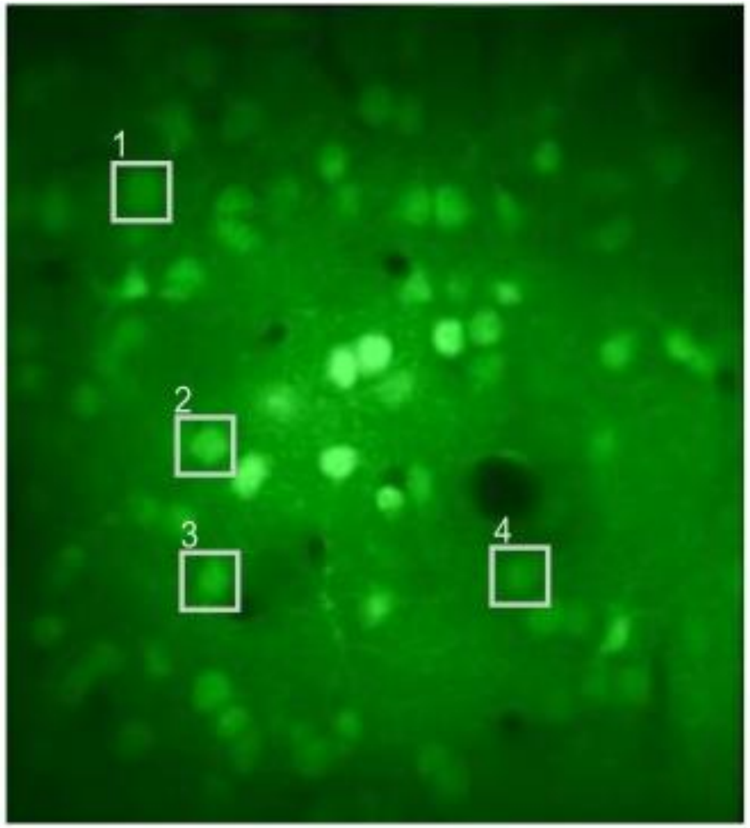
Getting spikes from calcium movies





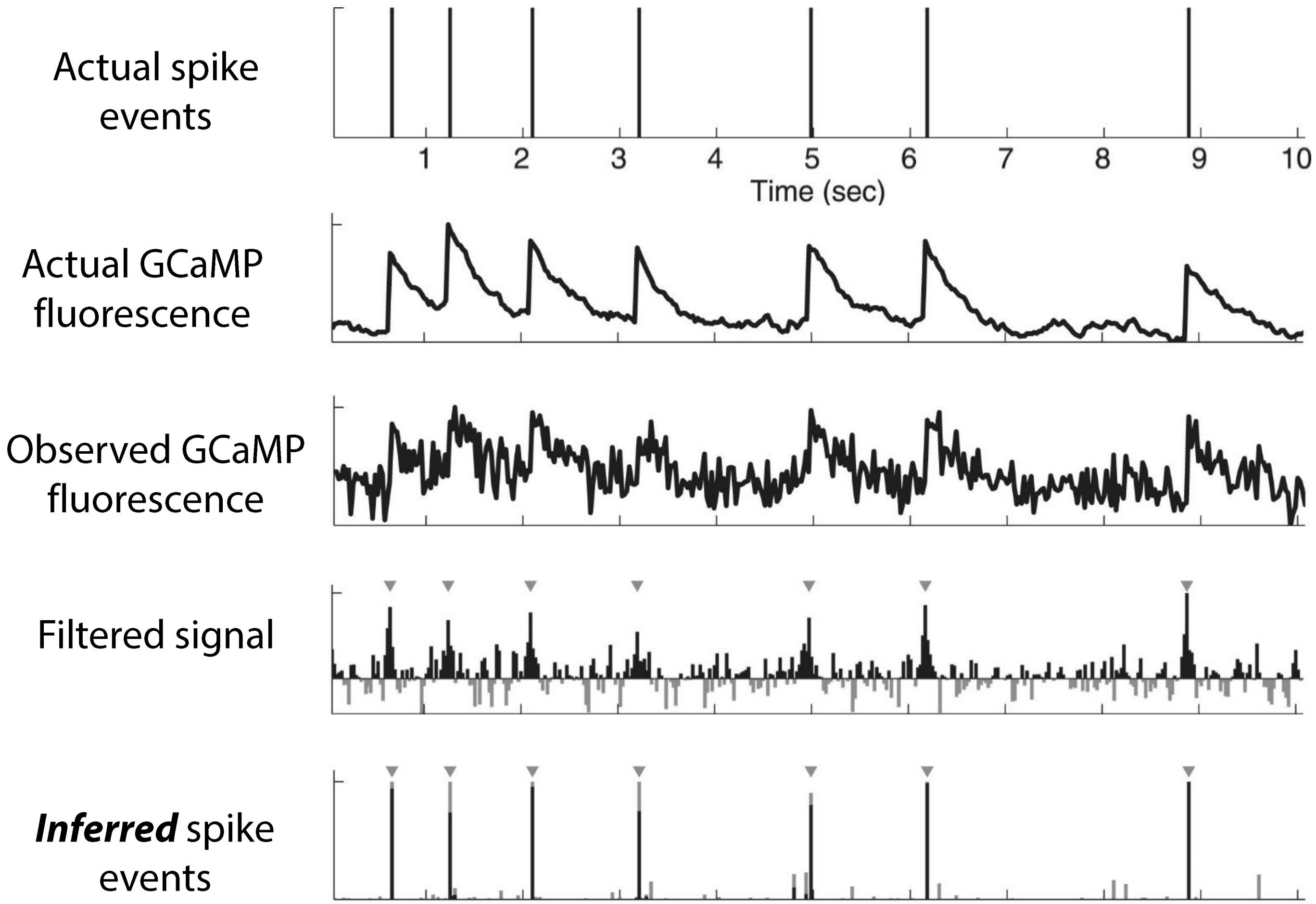
- Filter/motion correction
- Detect neurons

Getting spikes from calcium movies



- Filter/motion correction
- Detect neurons

Spike inference



Tools for preprocessing are becoming increasingly standardized

For electrophysiology/spike sorting

Fast and accurate spike sorting of high-channel count probes with KiloSort

Part of: Advances in Neural Information Processing Systems 29 (NIPS 2016)

[PDF] [BibTeX] [Reviews]

Authors

- Marius Pachitariu
- Nicholas A. Steinmetz
- Shabnam N. Kadir
- Matteo Carandini
- Kenneth D. Harris

https://github.com/cortex-lab/KiloSort

229 lines (147 sloc) | 10.8 KB   Raw | Blame | History

MountainSort

MountainSort is spike sorting software. It is part of MountainLab, a larger framework for conducting reproducible and shareable data analysis.

Installation and basic usage

Overview

There are many ways to use MountainSort and MountainLab, and there is not one set of installation instructions that will fit

https://github.com/flatironinstitute/mountainsort

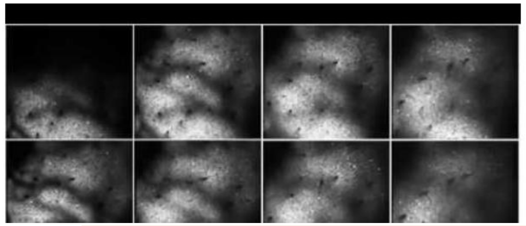
For calcium imaging/spike inference

README.md

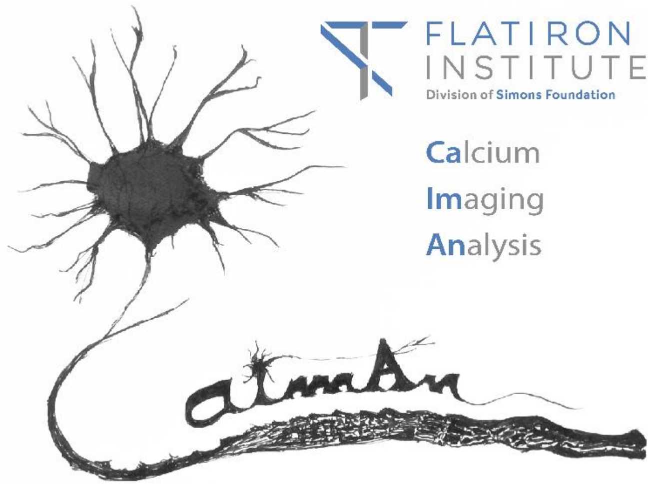
Suite2p: fast, accurate and complete two-photon pipeline

Registration, cell detection, spike extraction and visualization GUI.

Algorithmic details in http://biorxiv.org/content/early/2016/06/30/061507.



https://github.com/cortex-lab/Suite2P



https://github.com/flatironinstitute/CaImAn

What this course will actually cover

Course Outline

Module 1:
Loading and manipulating neural data

Module 2:
Statistical models of single neuron responses

Module 3:
Information representation in neural populations

Module 4:
Models of neural population activity

Course Outline

Module 1:
Loading and manipulating neural data

Module 2:
Statistical models of single neuron responses

Module 3:
Information representation in neural populations

Module 4:
Models of neural population activity

CRCNS

https://crcns.org/data-sets

Data sets used in this course will be posted to:
storage1-andersonlab.caltech.edu

(only accessible from on Caltech campus or via the Caltech VPN)

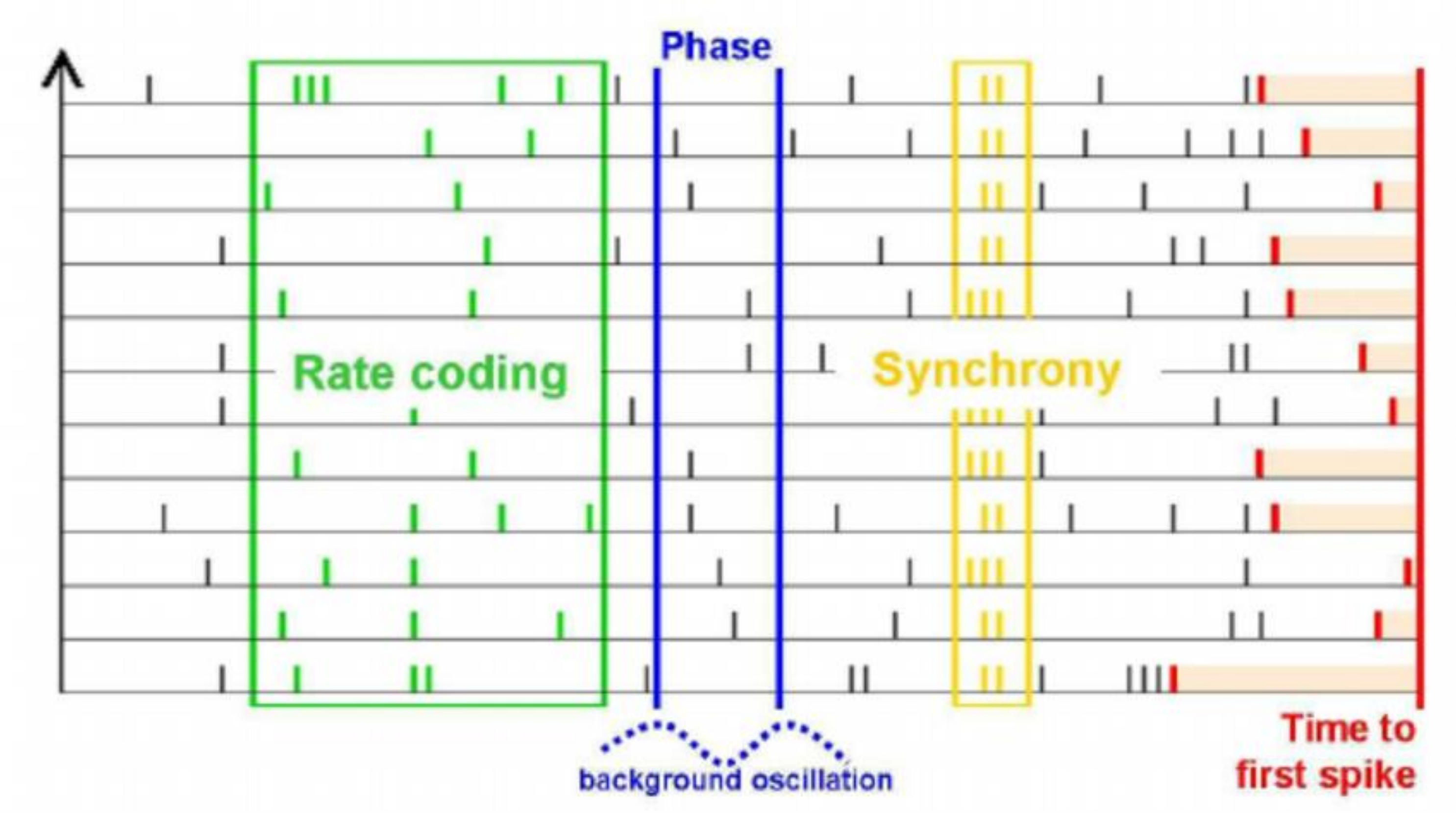
Module 1: big question

What information is represented in the firing of a neuron?

How does the firing of a neuron *depend on* or *encode* **X**, where X could be:

- a) features of a stimulus
- b) the animal's behavior
- c) other information (context, memory, etc)

Features of the neural code



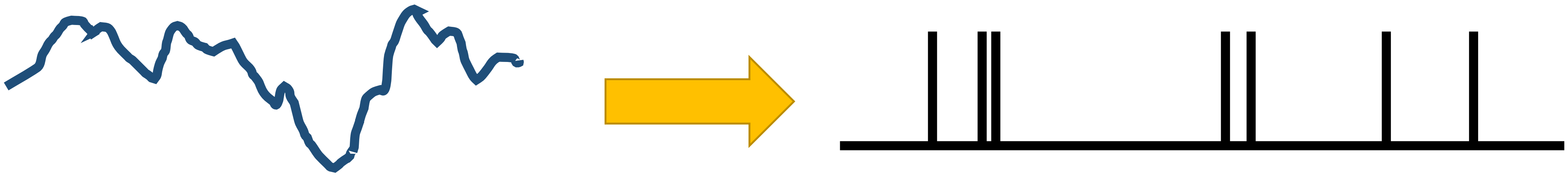
Single cells:

- Spike rate
- Spike timing
- Inter-spike intervals

Populations of neurons:

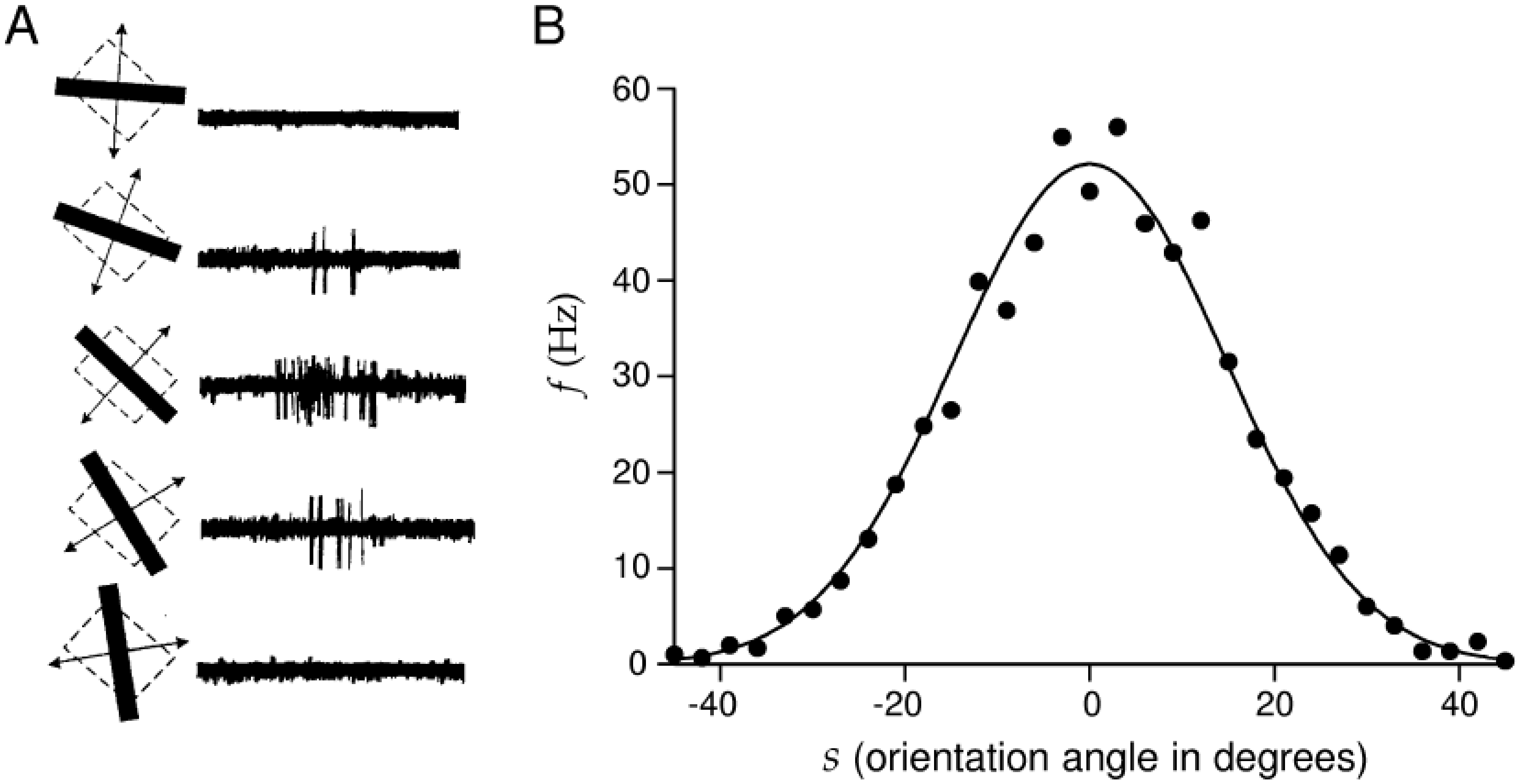
- Population-level dynamics
- Correlations between neurons
- Synchrony

Neural encoding



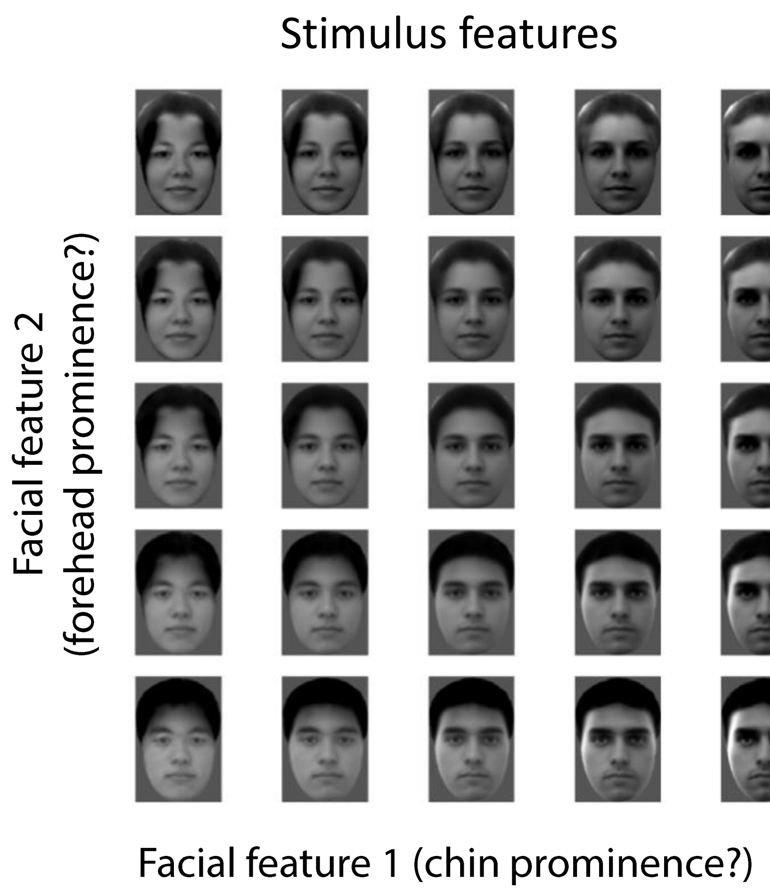
- Brightness of a screen
- Concentration of an odor
- Proximity of a predator
- Movement of a limb

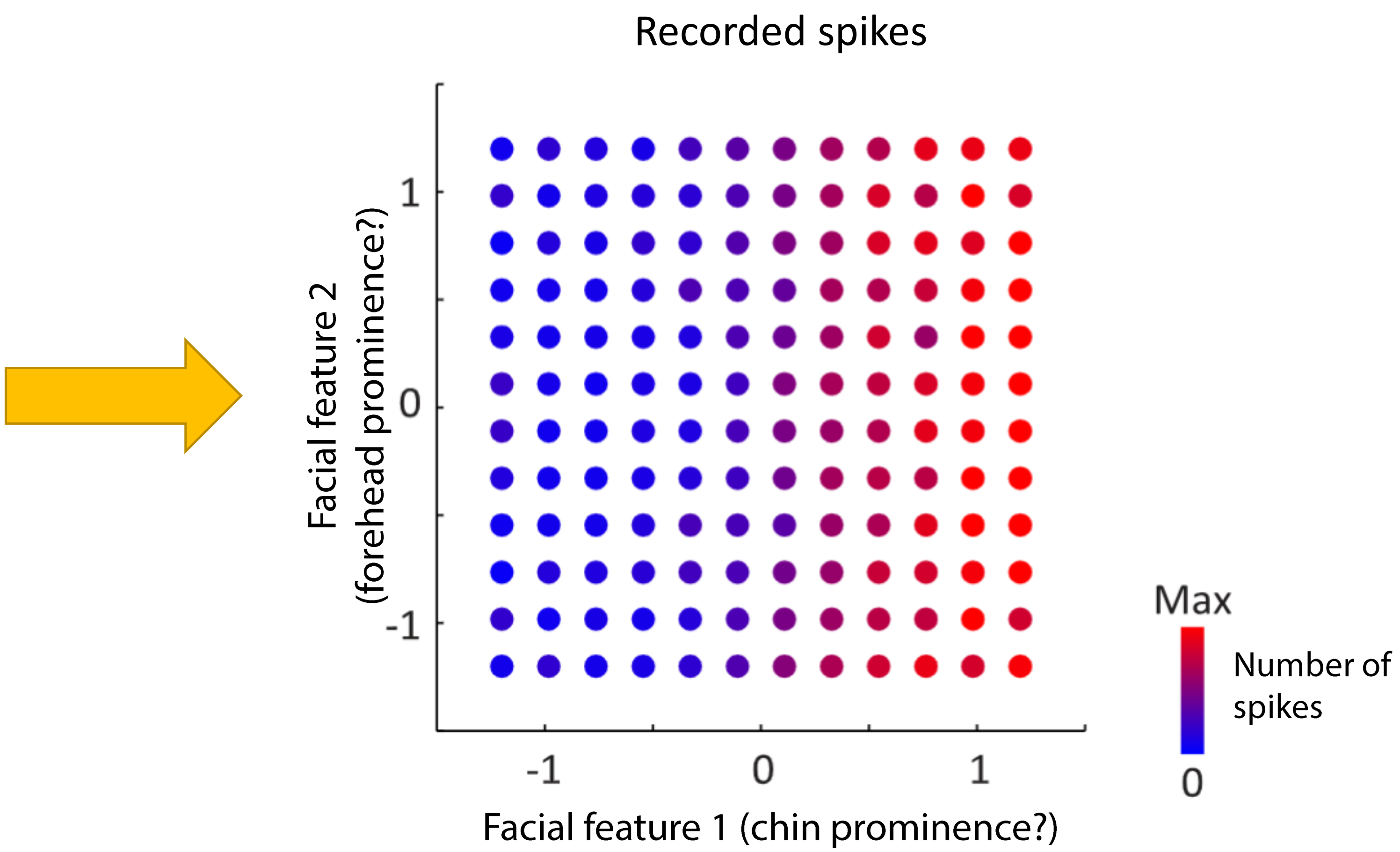
Tuning curves: vary one stimulus parameter, ask how firing rate changes



Gaussian tuning curve of a cortical (V1) neuron

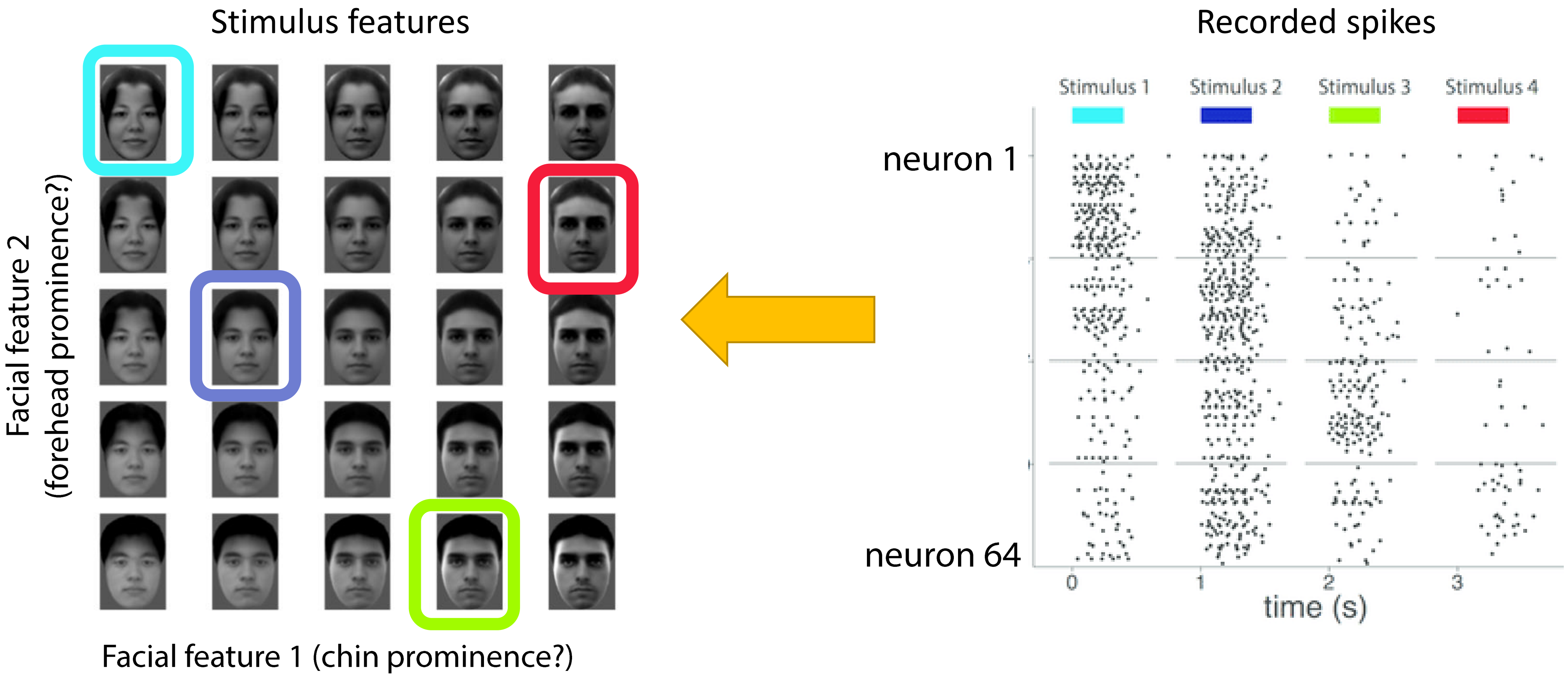
Neural encoding





Chang and Tsao 2017

Neural **decoding**



Another way of framing it:

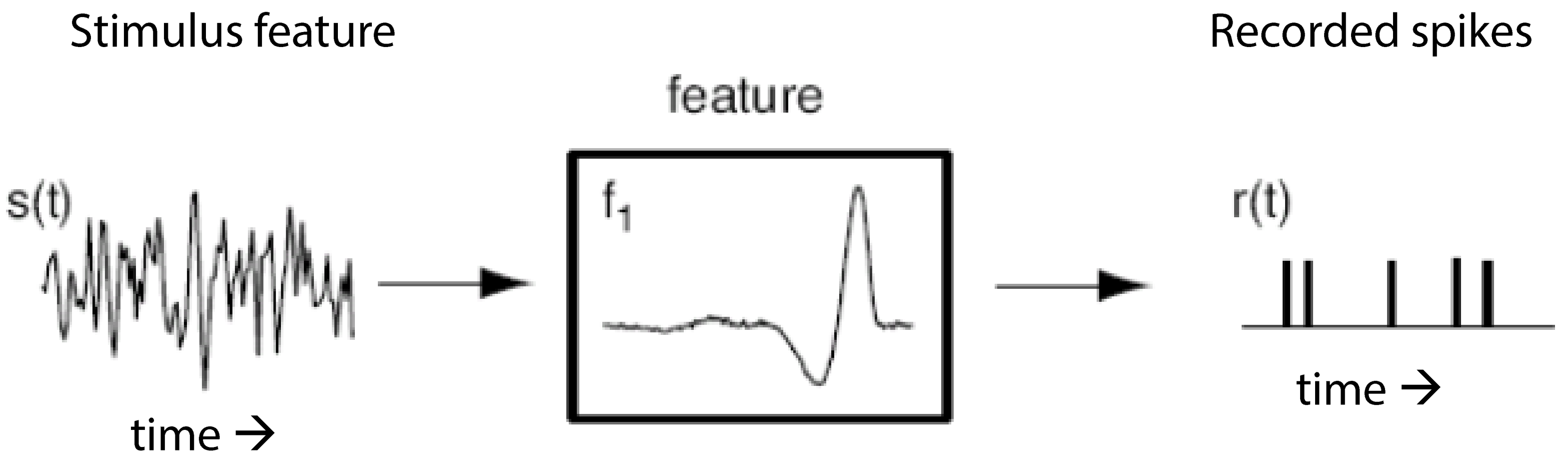
Neural encoding: $P(\text{response} \mid \text{stimulus})$

Neural decoding: $P(\text{stimulus} \mid \text{response})$

Then, with Bayes theorem:

$$P(\text{response} \mid \text{stimulus}) = \frac{P(\text{stimulus} \mid \text{response}) * P(\text{response})}{P(\text{stimulus})}$$

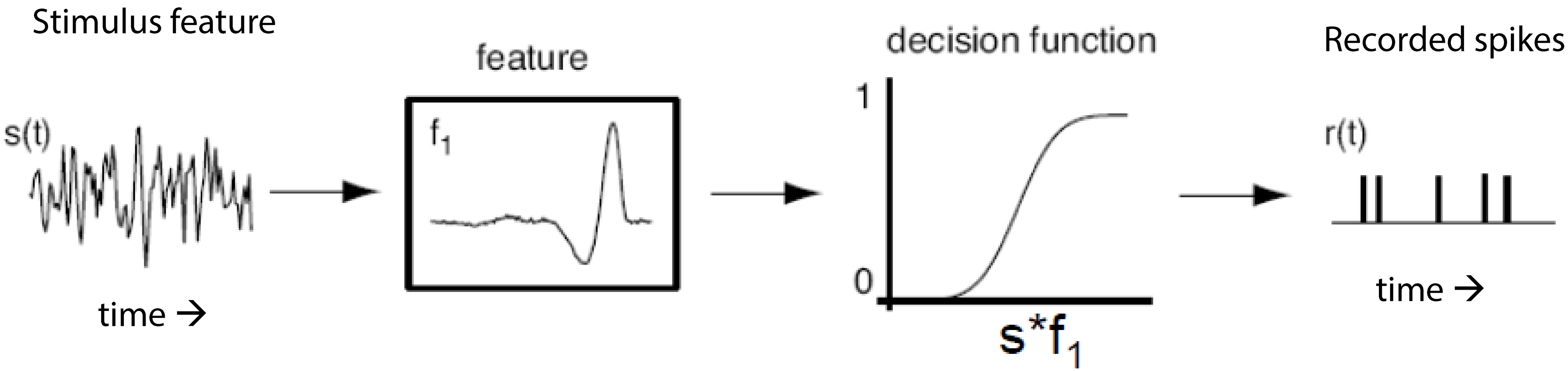
Basic linear encoding model



$$r(t) = \text{stimulus} * \text{filter} = \int f(t-\tau)s(\tau)d\tau$$

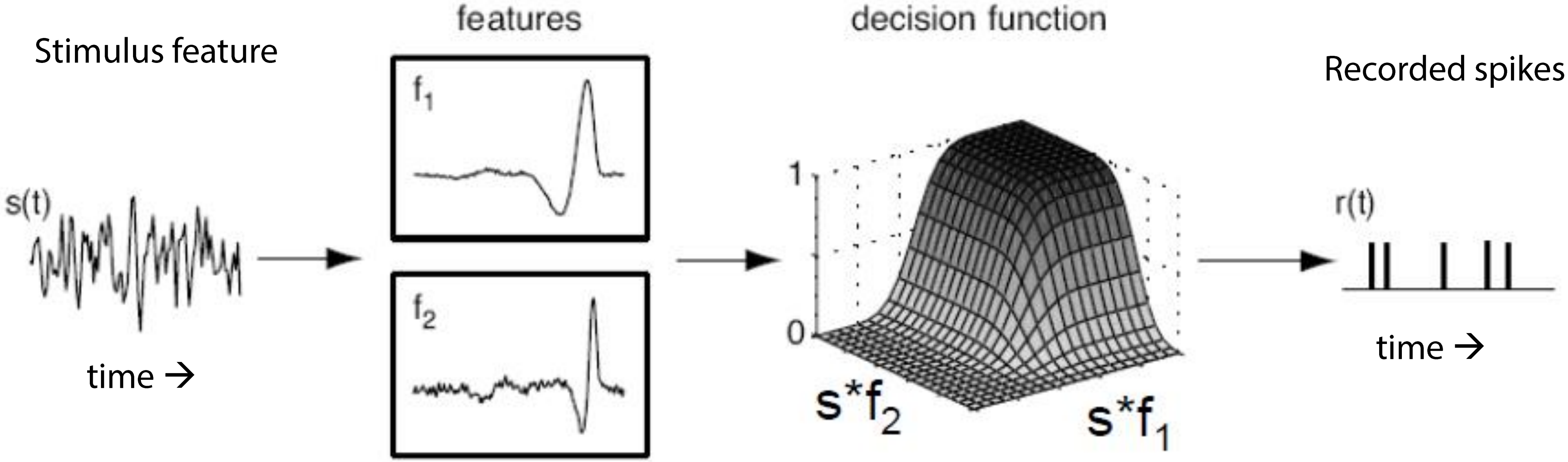
(convolution)

Linear-nonlinear encoding model



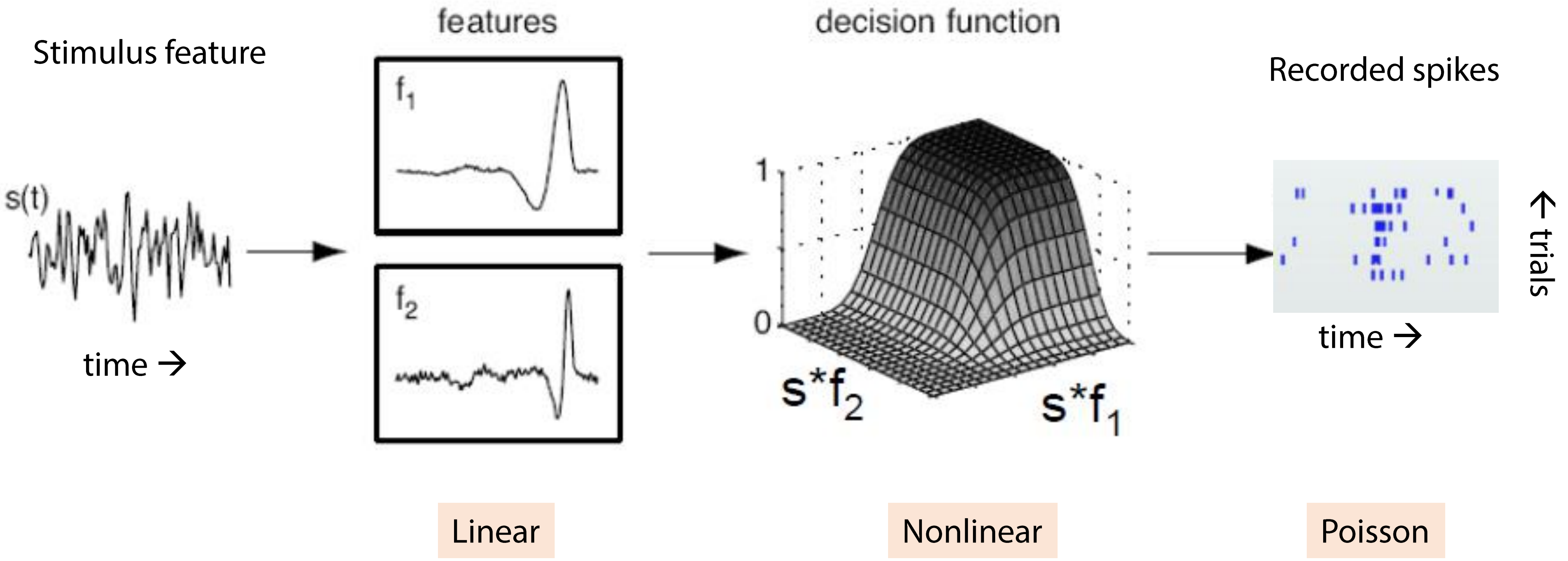
$$r(t) = g(\text{stimulus} * \text{filter}) = g(\int f(t-\tau)s(\tau)d\tau)$$

Linear-nonlinear encoding model

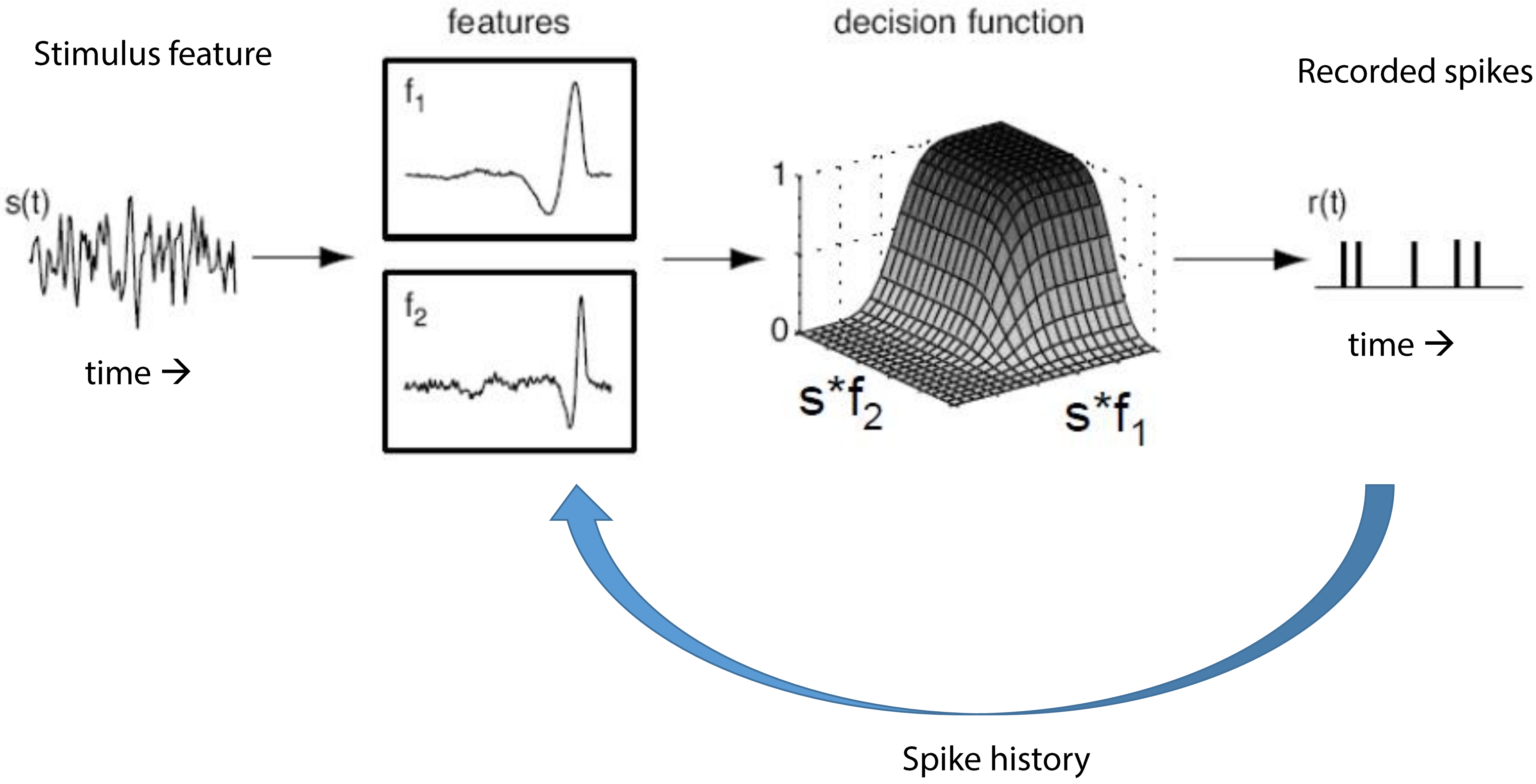


$$r(t) = g(\text{stimulus} * \text{filter } 1, \text{stimulus } * \text{filter } 2, \dots)$$

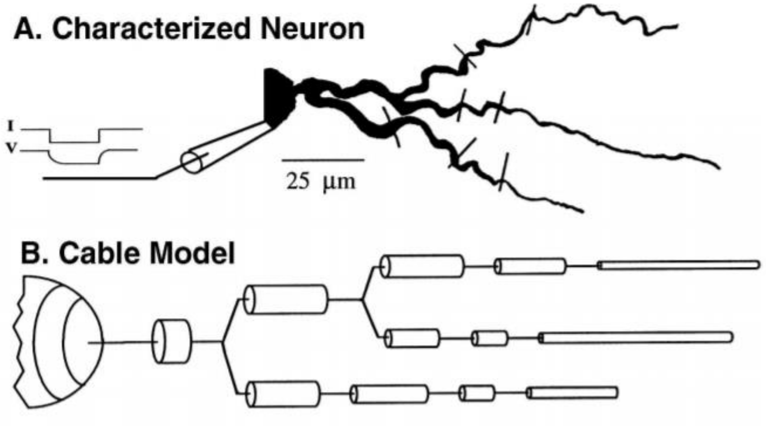
Linear-nonlinear encoding model

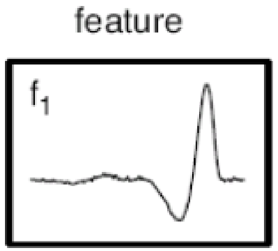


Linear-nonlinear encoding model



What do we mean by "model"?





More complex ⟷ *Less complex*

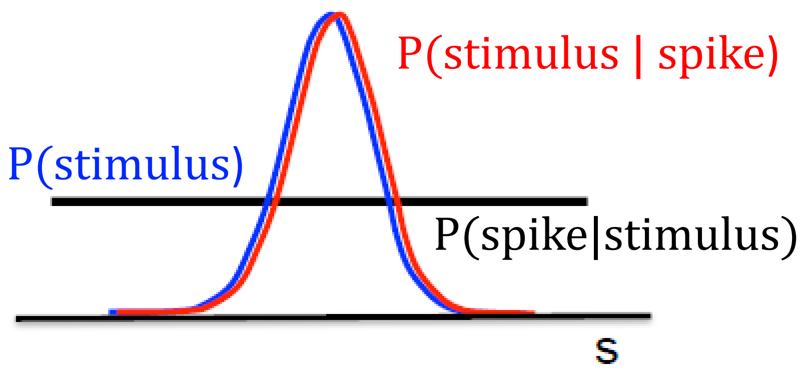
Biophysical, multicompartmental model

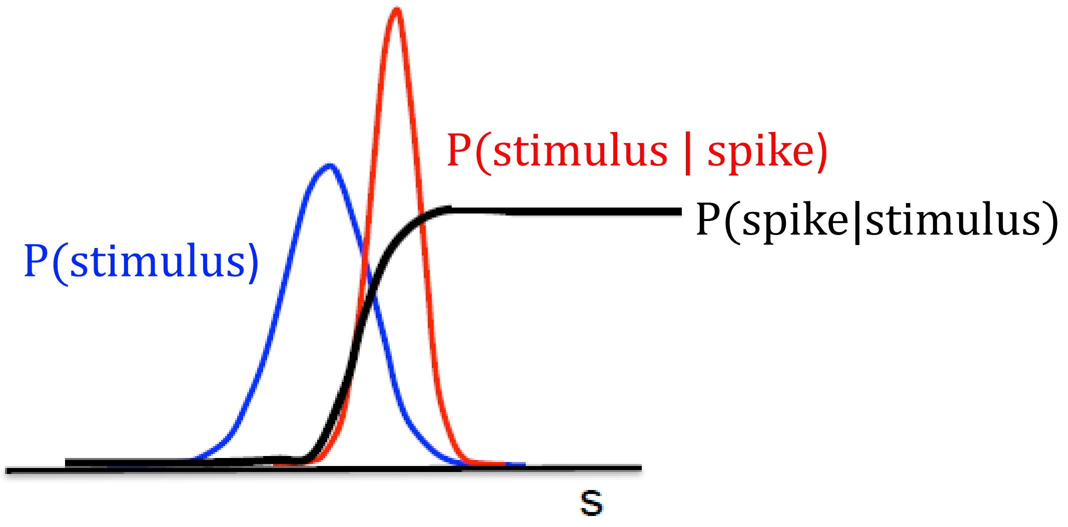
Linear neural filter

"All models are wrong, but some are useful"

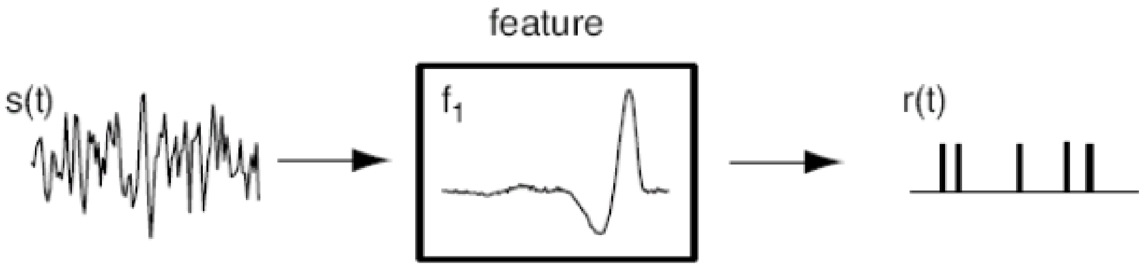
How do we know if we've learned something?

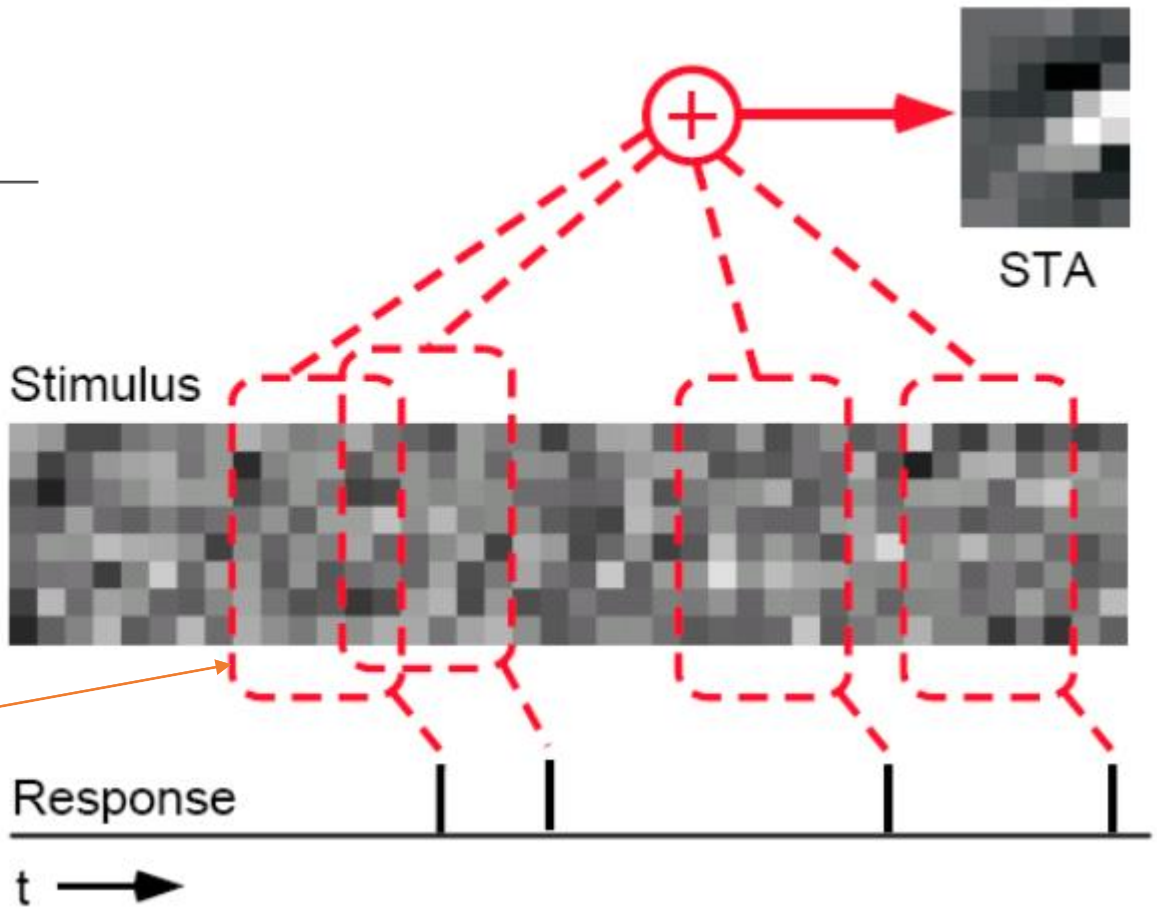
$$P(spike \mid stimulus) = \frac{P(stimulus \mid spike) * spike)}{P(stimulus)}$$





First steps: let’s estimate the neuron’s filter using spike-triggered averaging!





How big should this window be?