

Machine Learning Methods for Neural Data Analysis

Lecture 2: Basic Spike Sorting

Scott Linderman

STATS 220/320 (*NBIO220, CS339N*). Winter 2021.

Survey Results

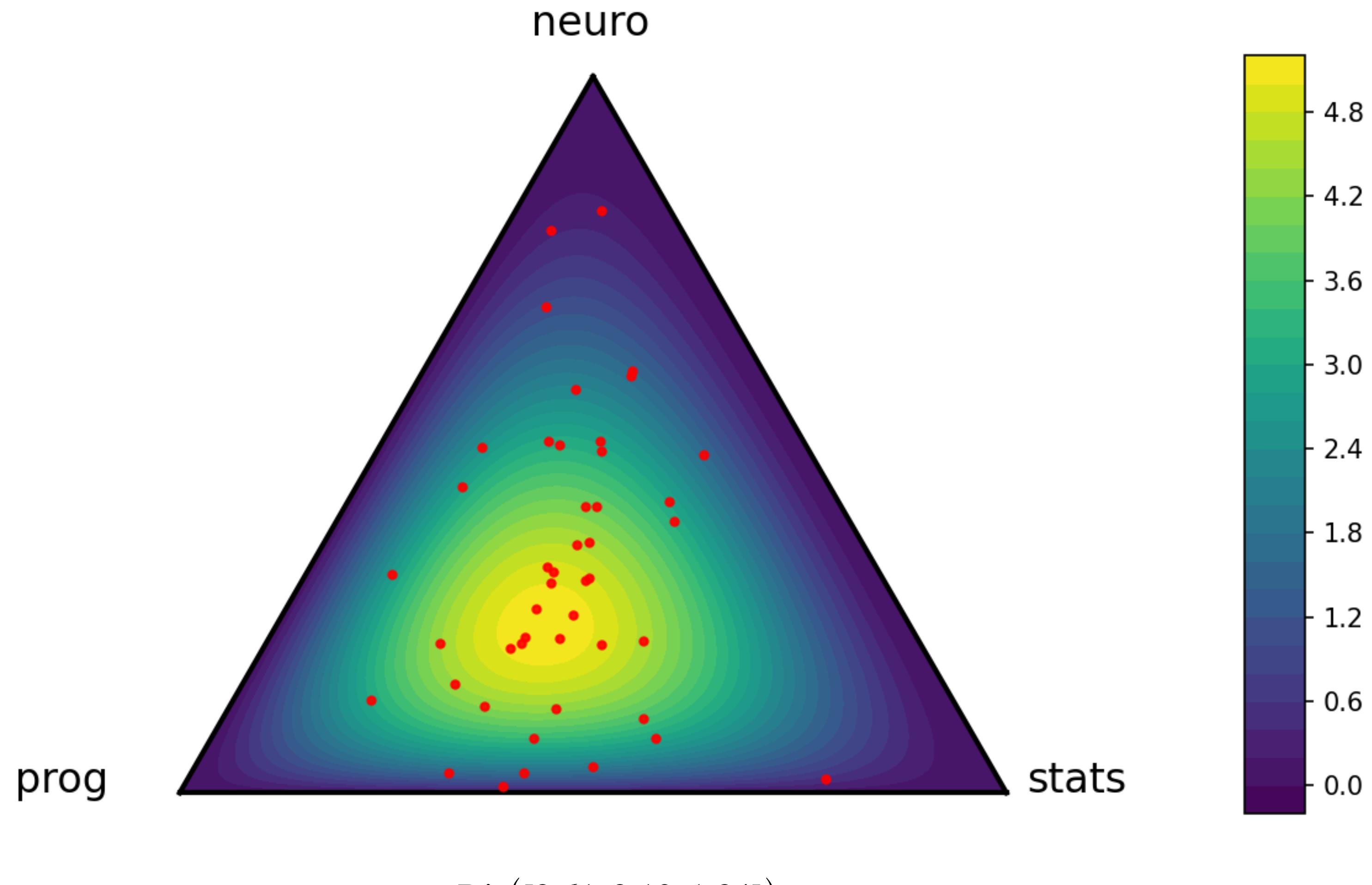
Where are you zooming in from?

- Stanford / Palo Alto / Campus / Menlo Paro
- LA
- NYC
- Atlanta
- Hawaii!
- Virginia
- Seattle
- Singapore
- Belgium
- Texas
- Milwaukee
- New Jersey
- SoCal
- Truckee
- San Jose

Have you picked up any hobbies?

- Cooking
- Cleaning
- Urban Survival
- Coffee
- Cycling
- Miso
- Pandemic puppy!
- Guitar
- Golf
- Card shuffling
- Needlepointing!
- Wearing masks
- Sewing
- No
- Yes
- Writing
- Birding
- Losing track of what day it is
- Jumping rope
- Learning Romanian
- Journaling
- Yoga
- Walking in a boot ☹
- Legos
- Sourdough
- Yu-gi-oh!
- Crocheting
- Reading
- Surfing
- Chess
- History podcasts
- Neuroscience
- Ukulele
- Drones
- Photography
- Music production

Survey Results

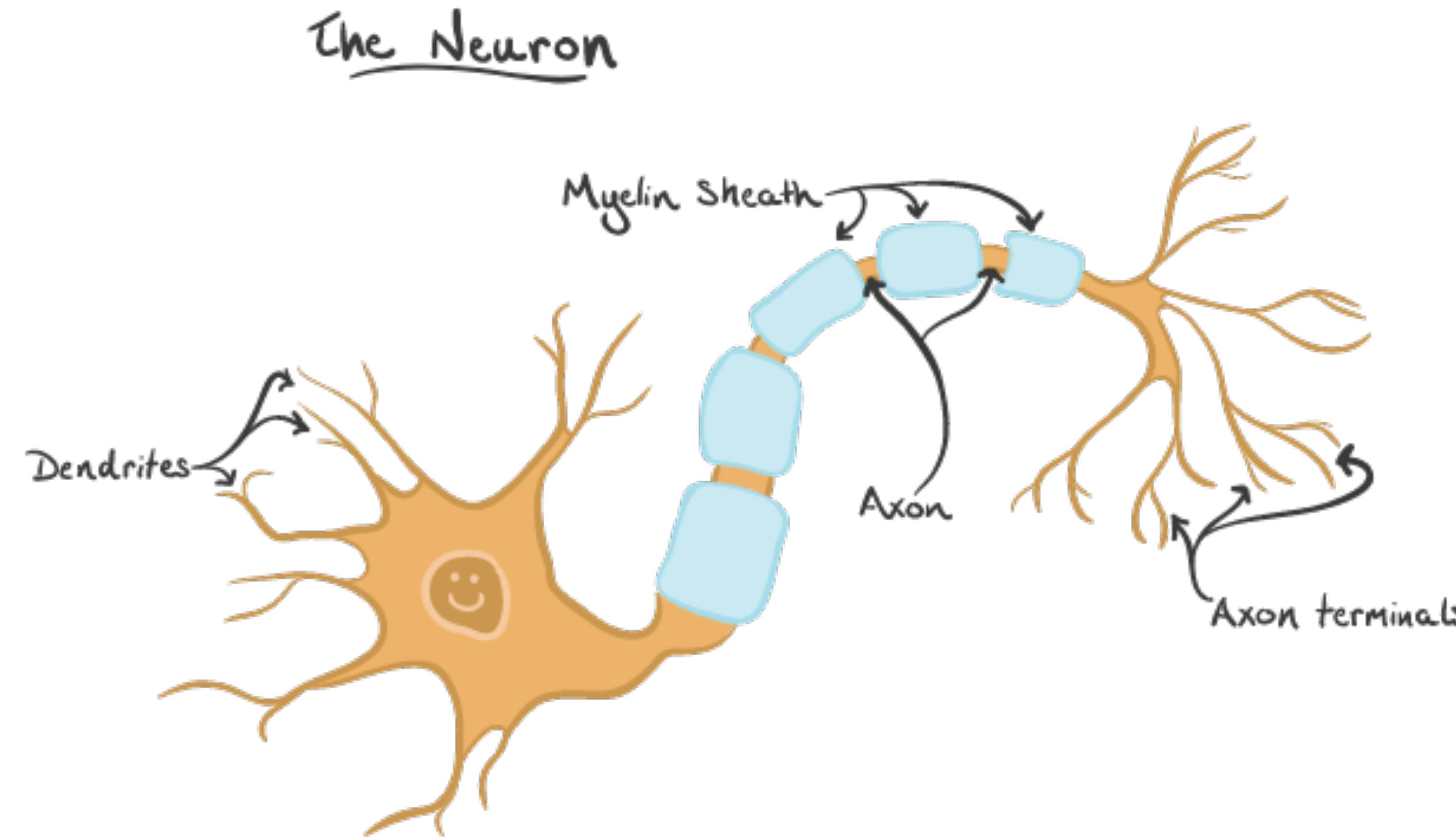


Agenda

1. Basic neurobiology
2. A spike sorting model
3. Maximum a posteriori inference

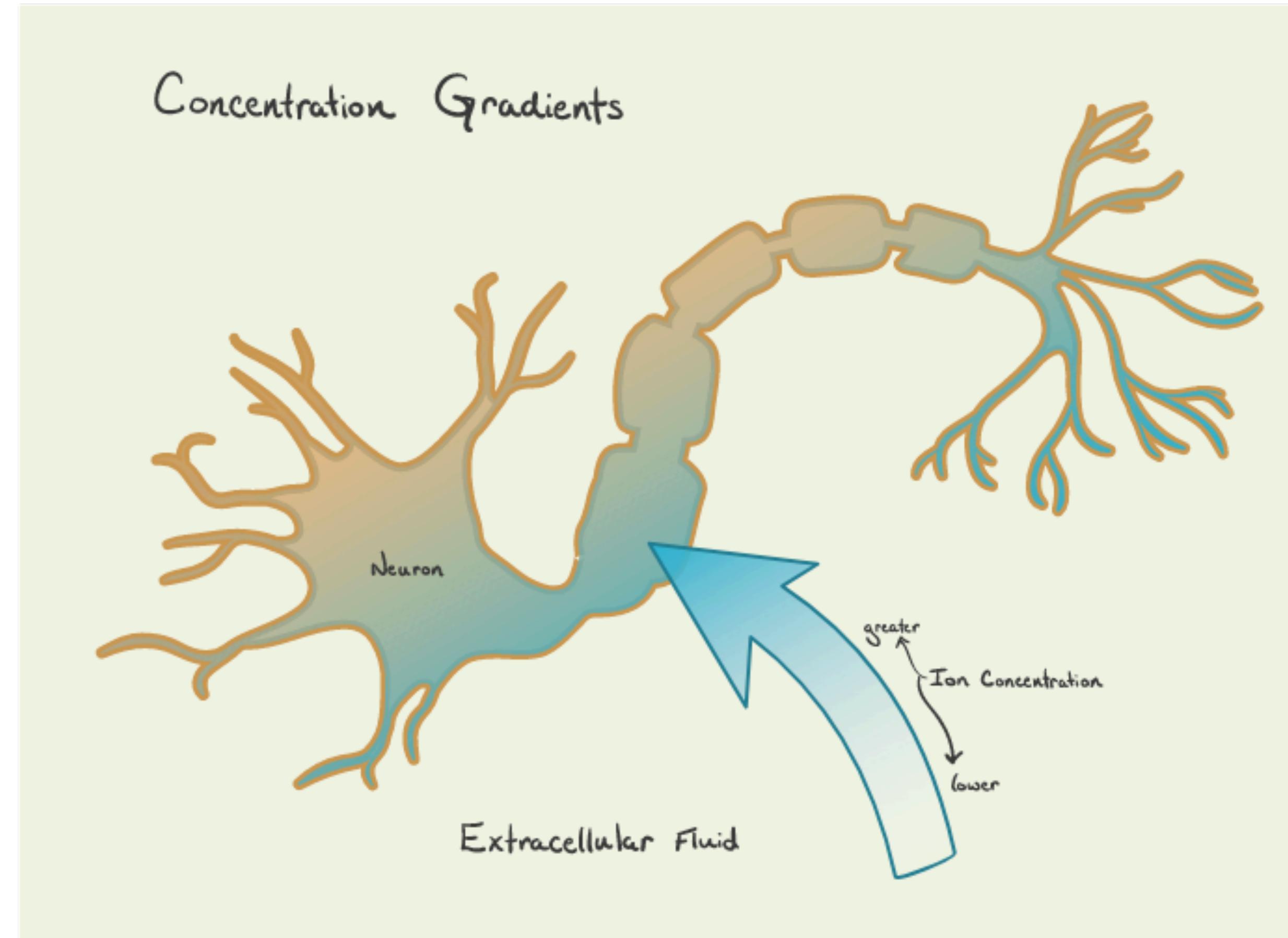
Neurobiology 101

Anatomy of a neuron



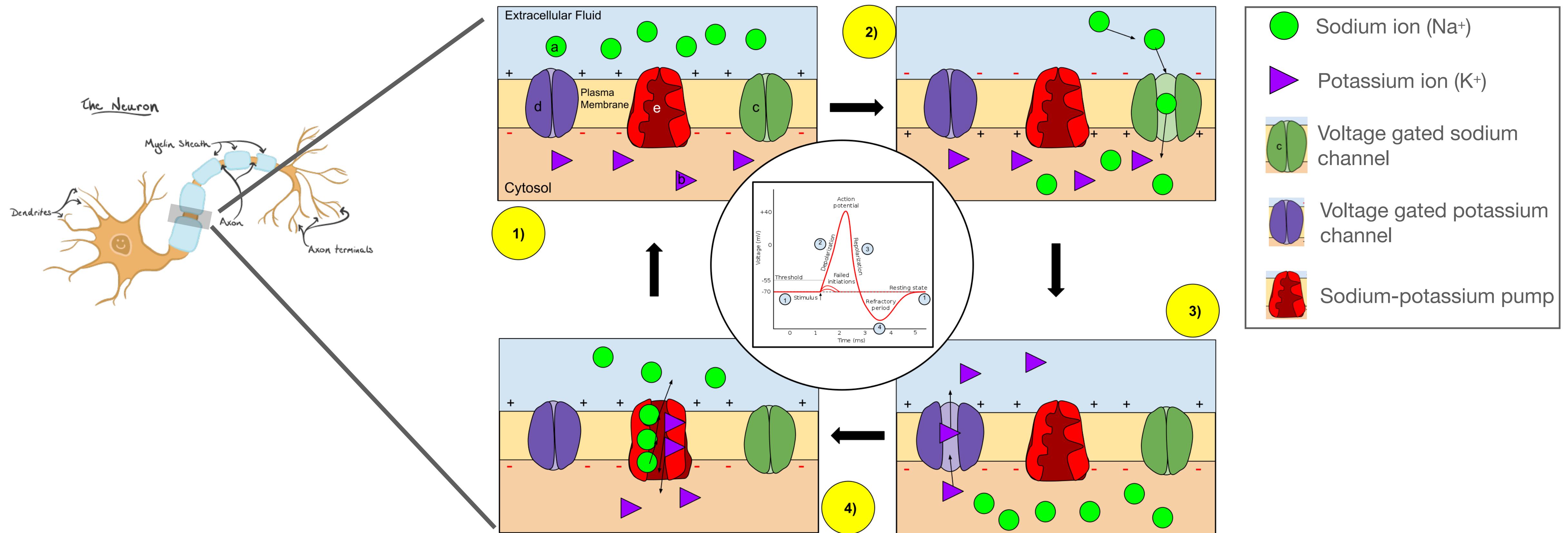
Neurobiology 101

Anatomy of a neuron



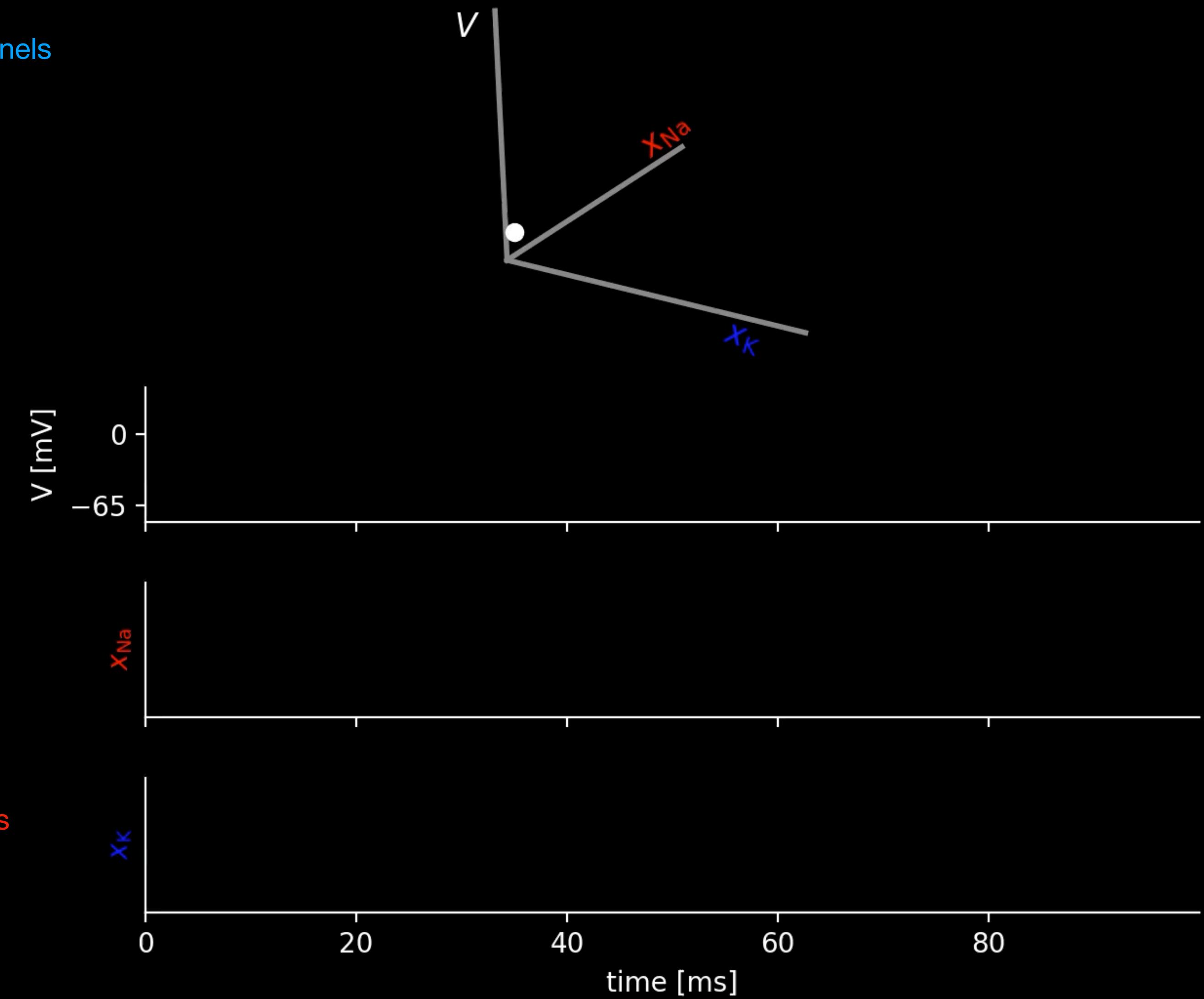
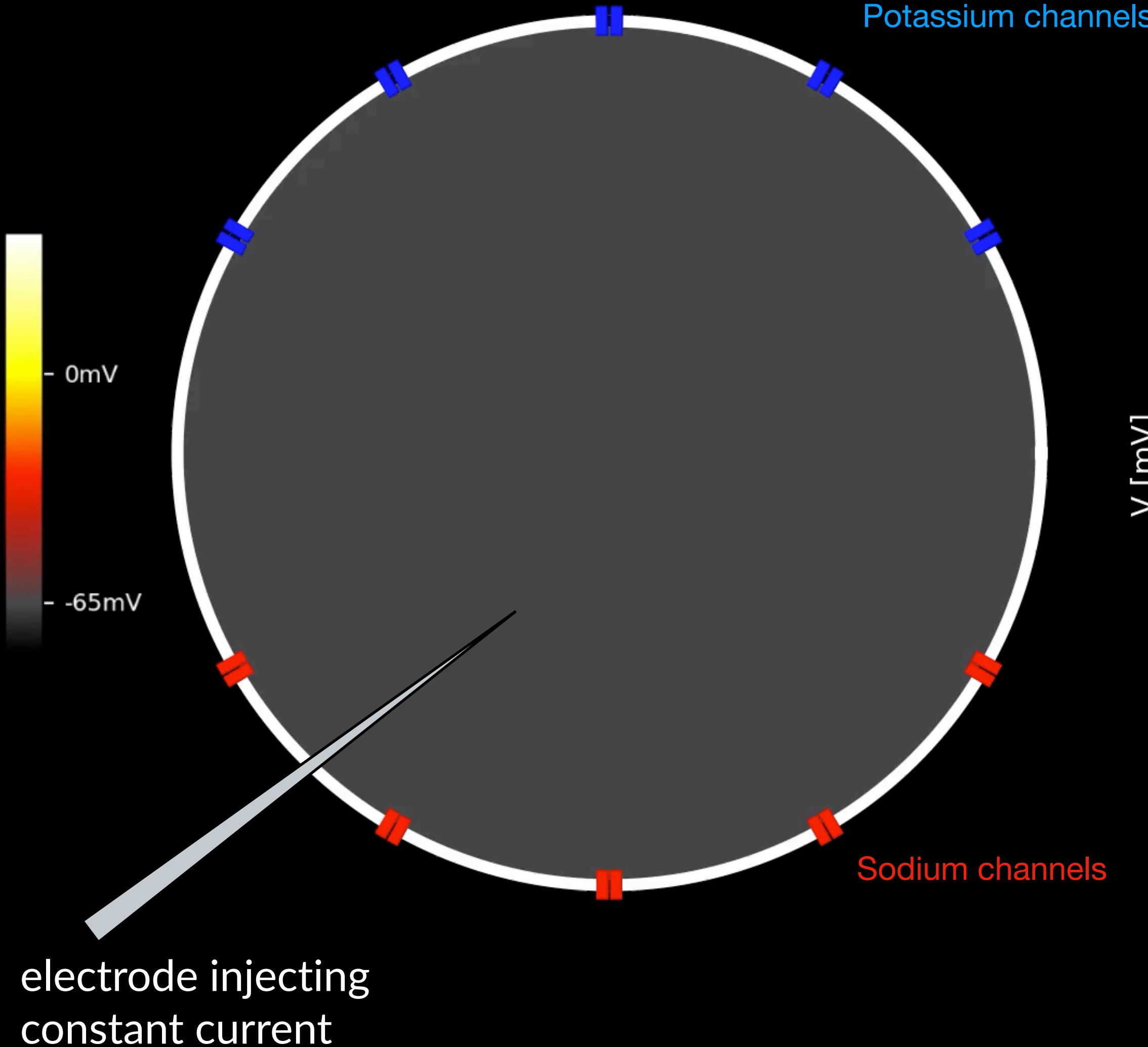
Neurobiology 101

Voltage-gated ion channels



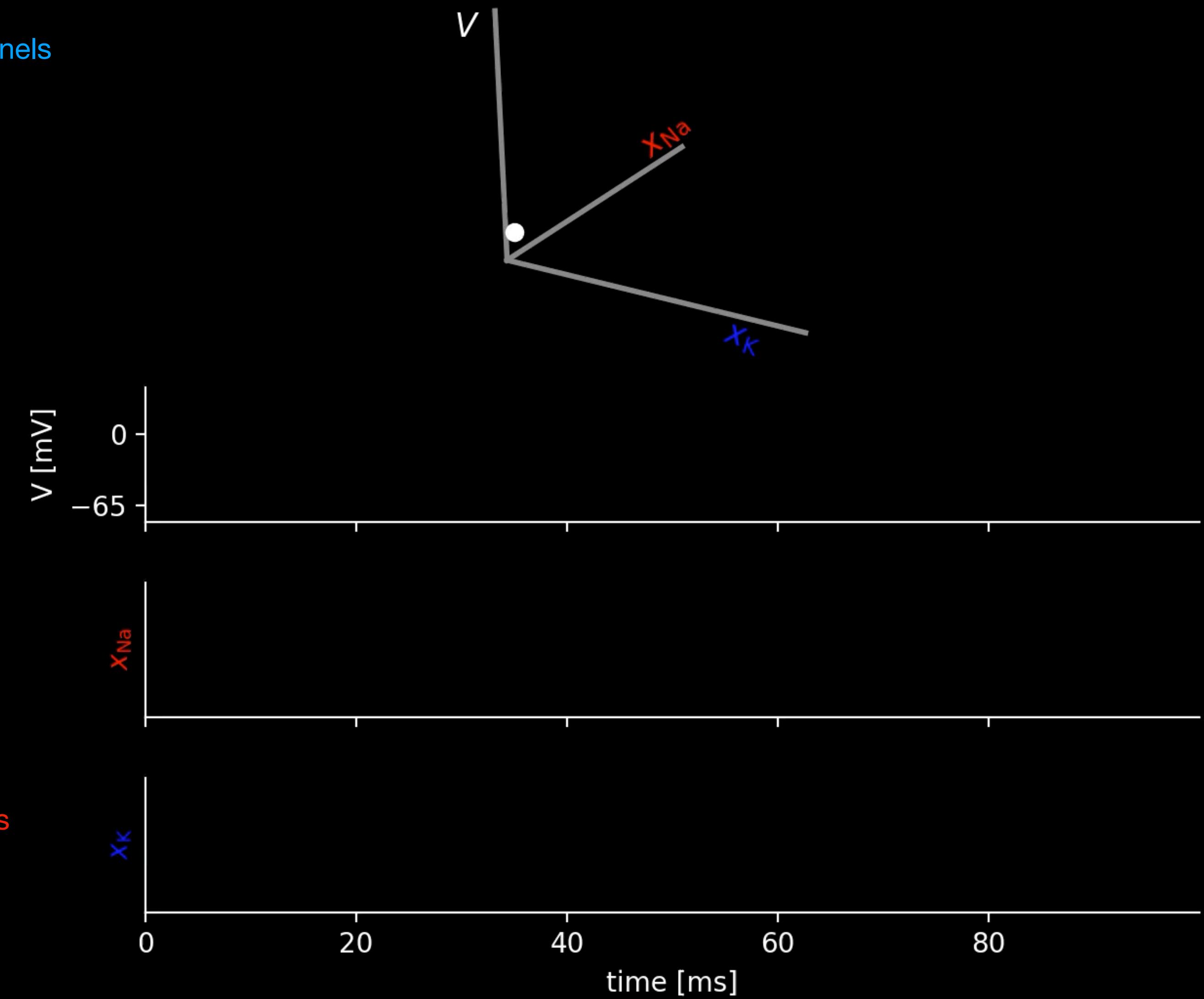
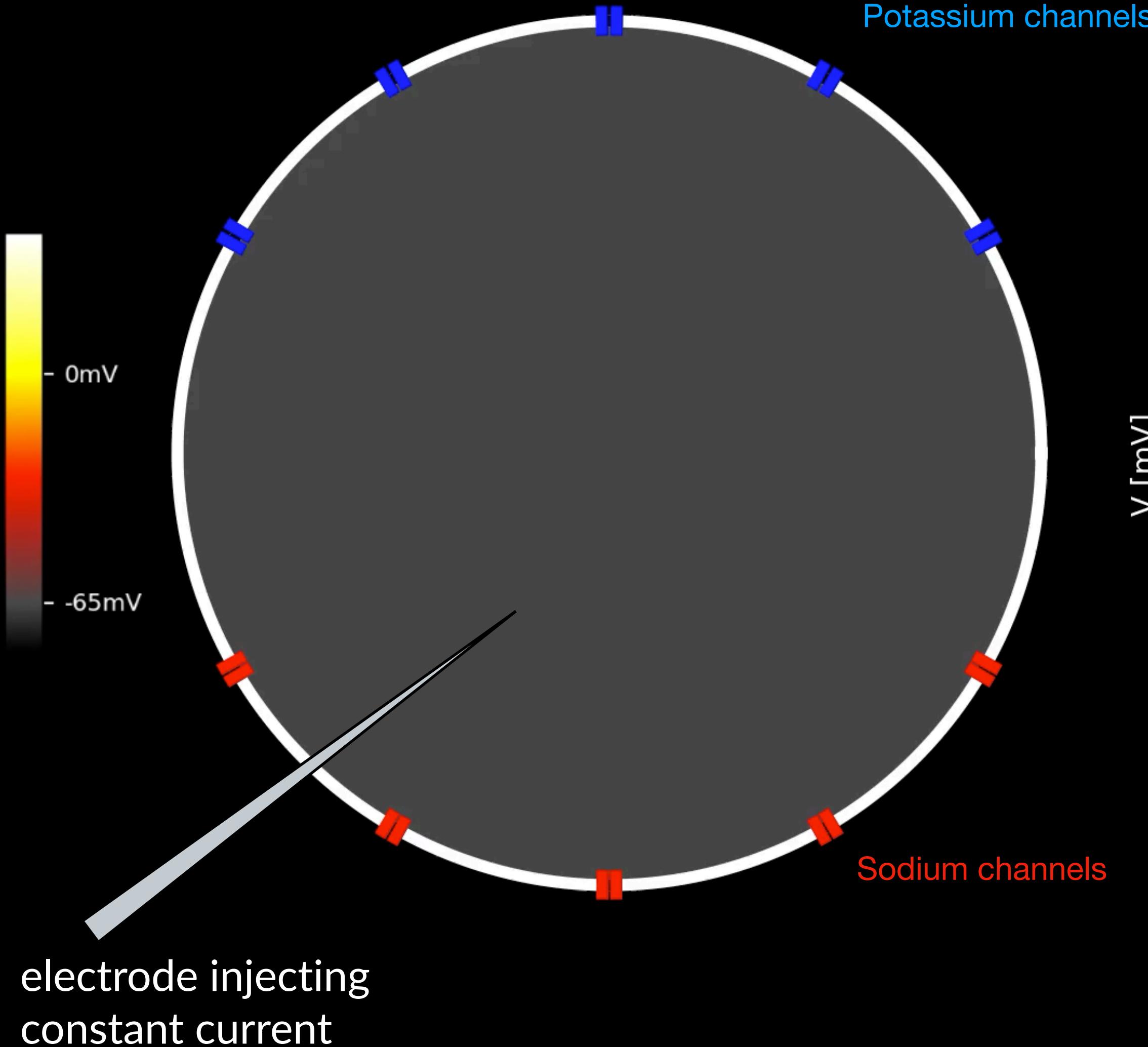
Neurobiology 101

Action potentials



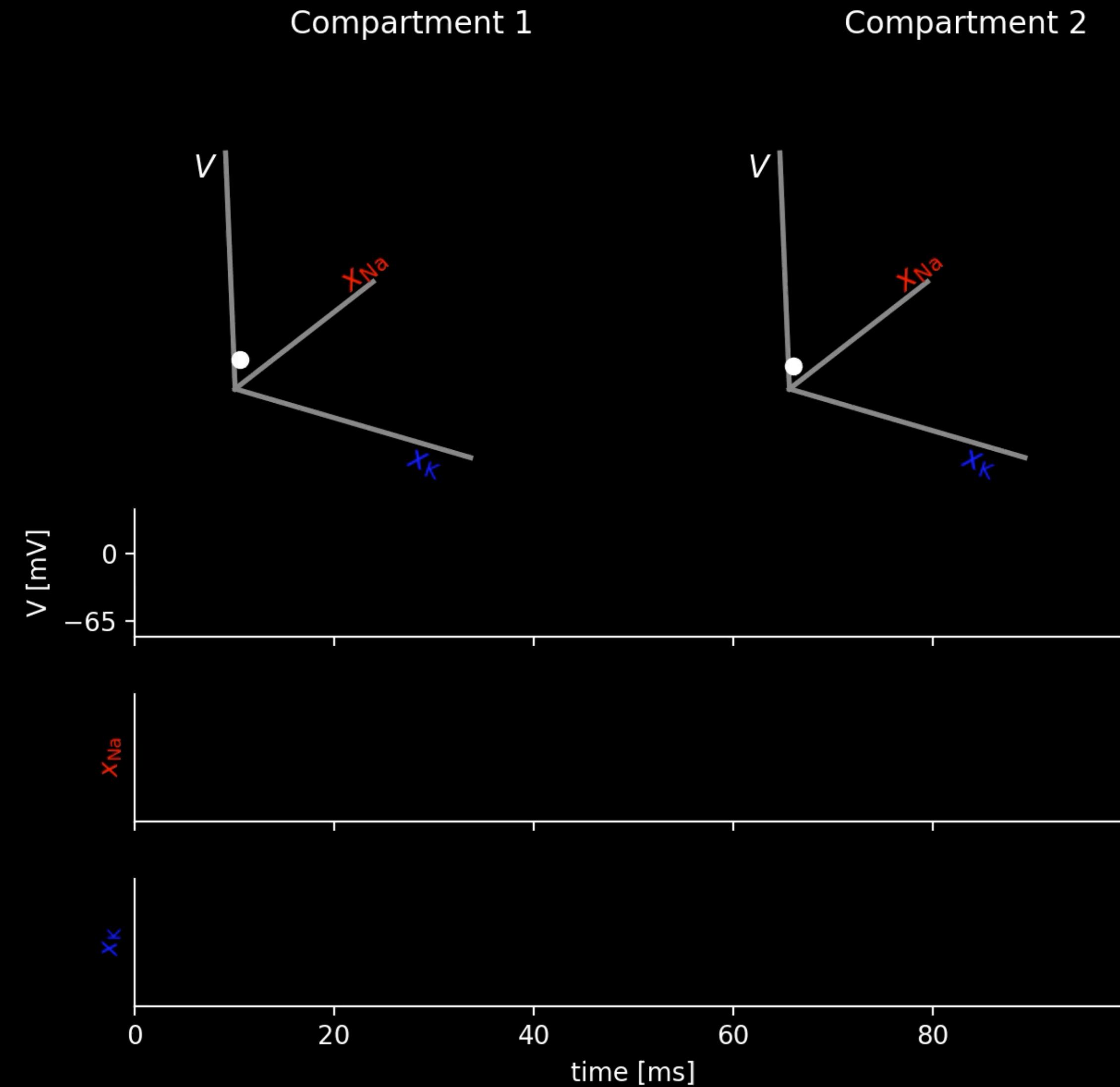
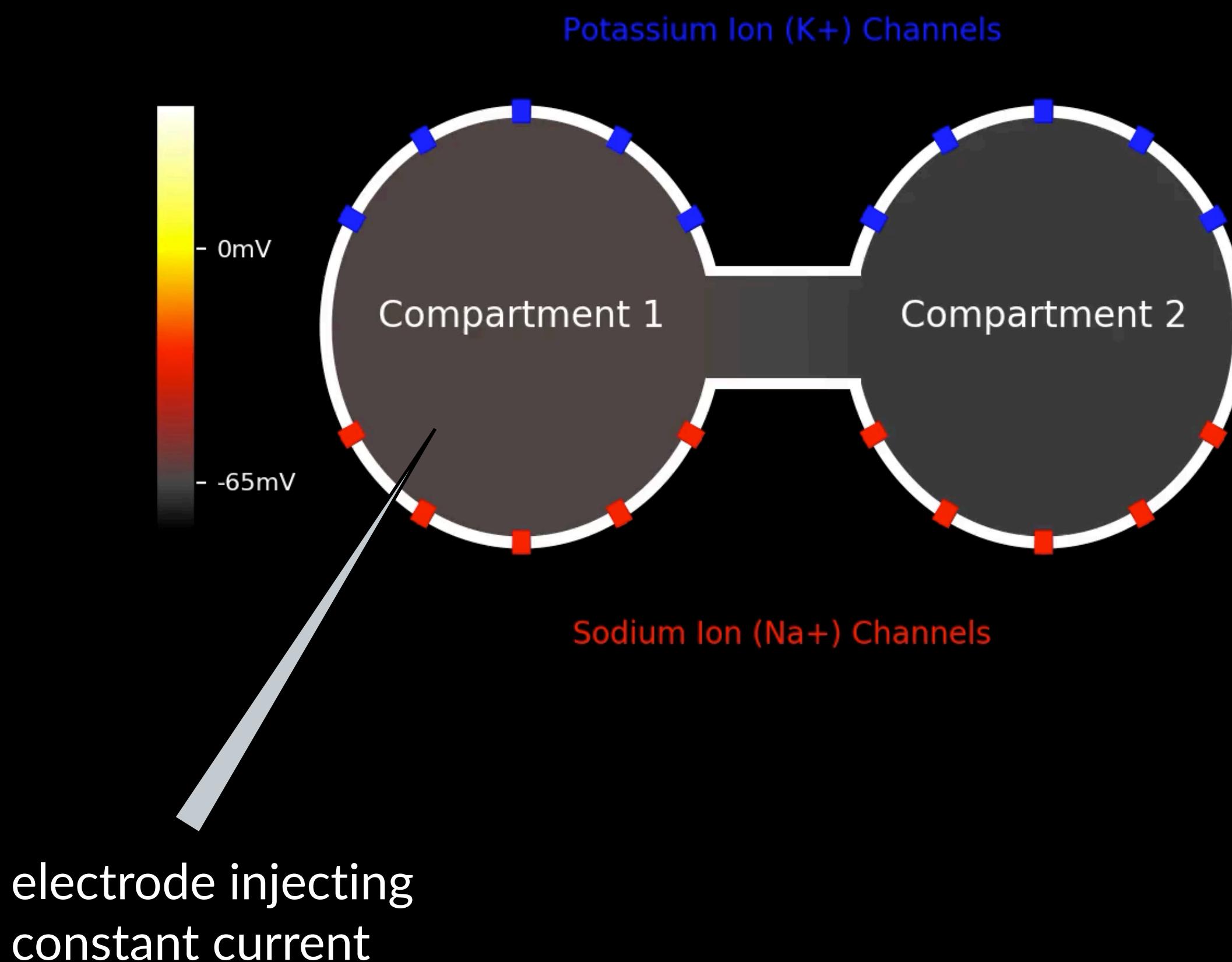
Neurobiology 101

Action potentials



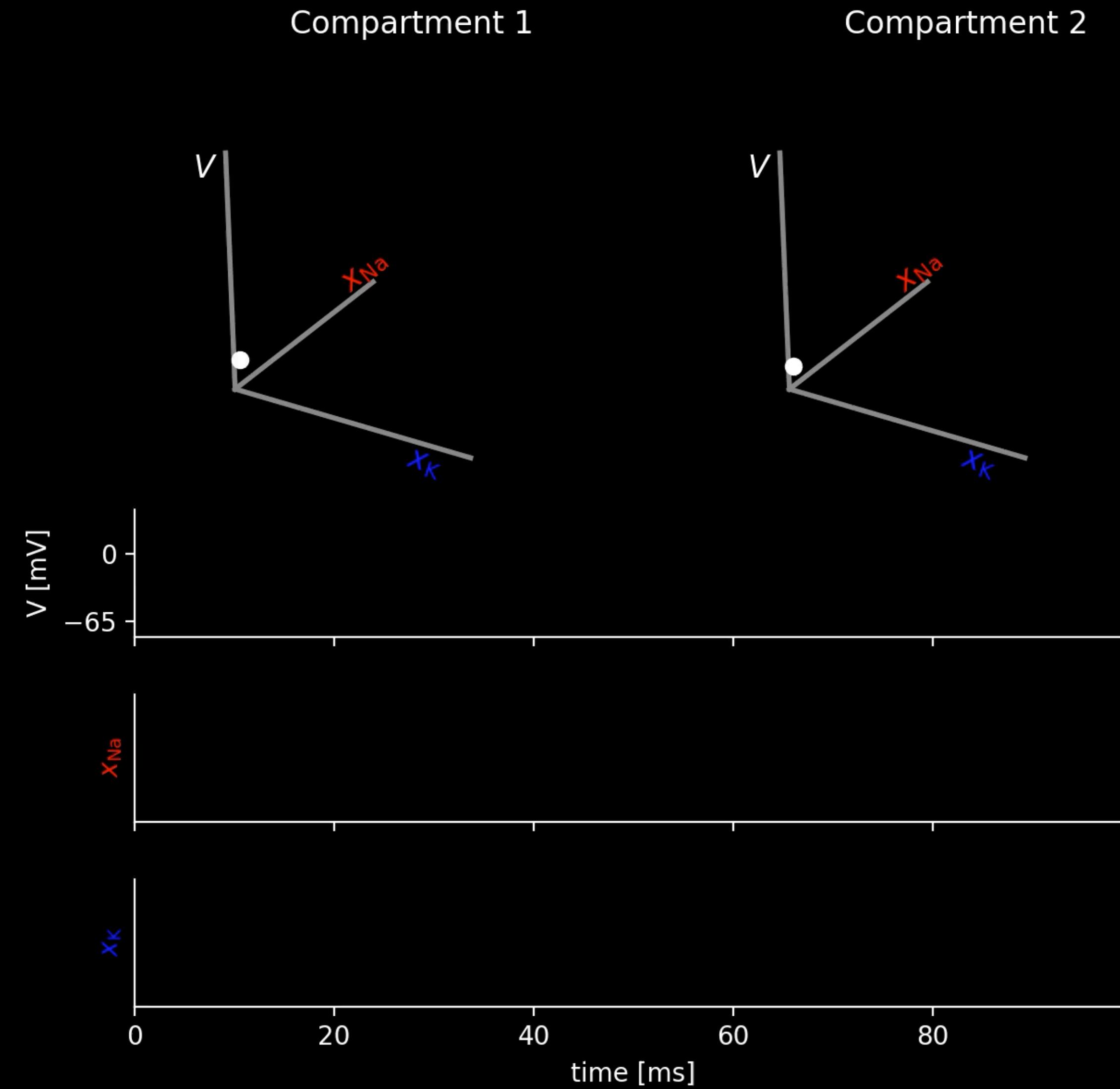
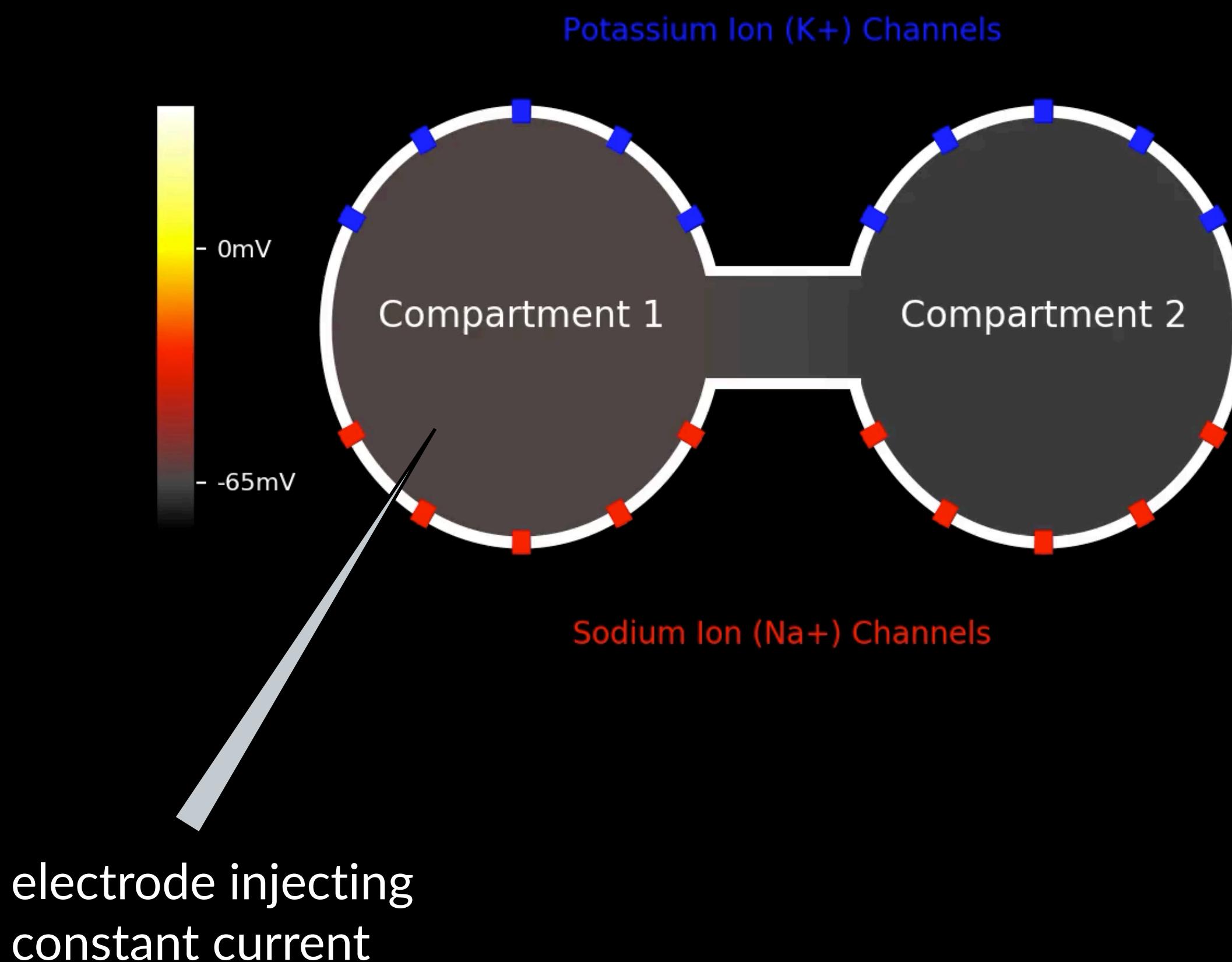
Neurobiology 101

Action potential propagation



Neurobiology 101

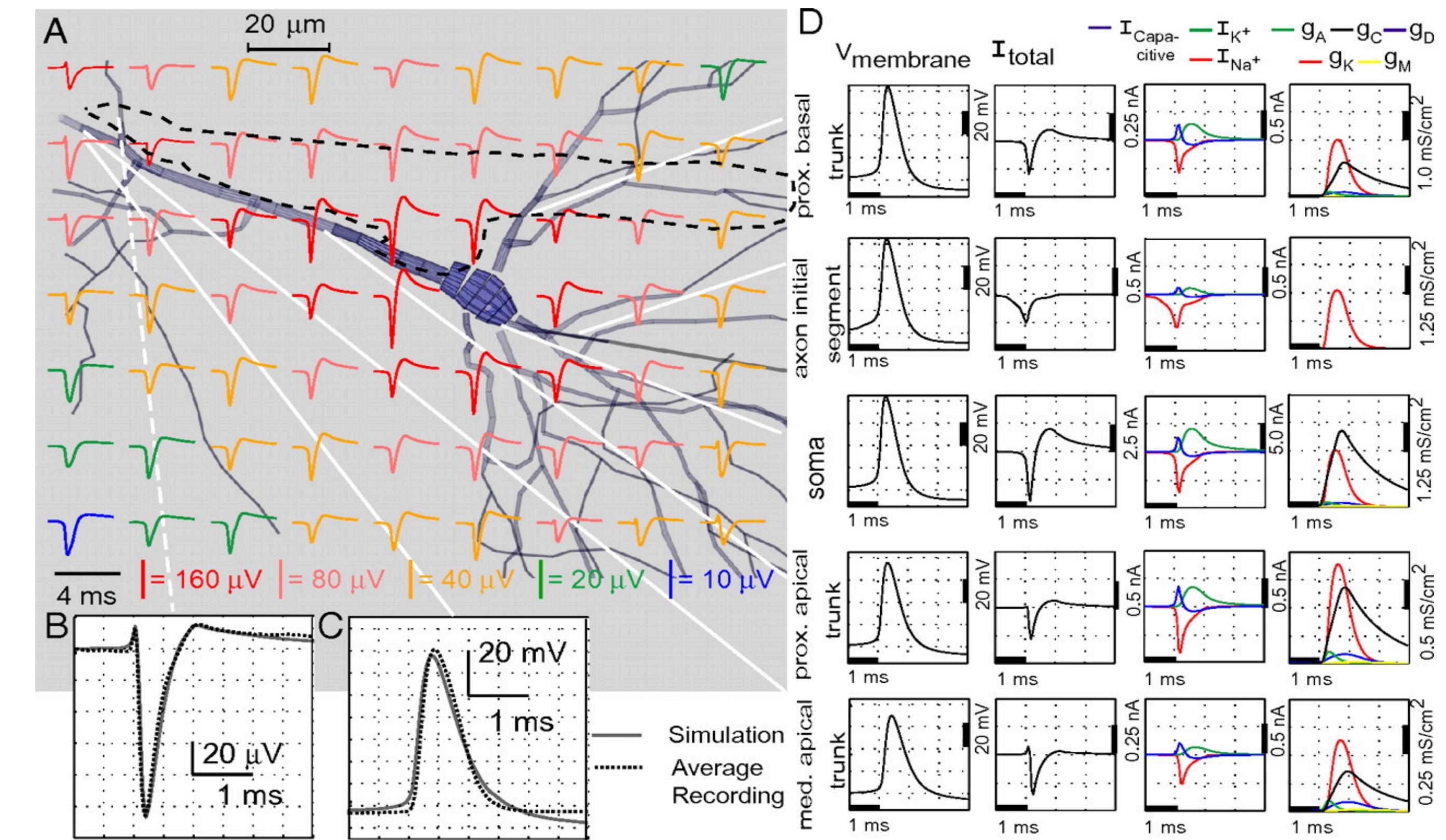
Action potential propagation



Neurobiology 101

Extracellular voltage recordings

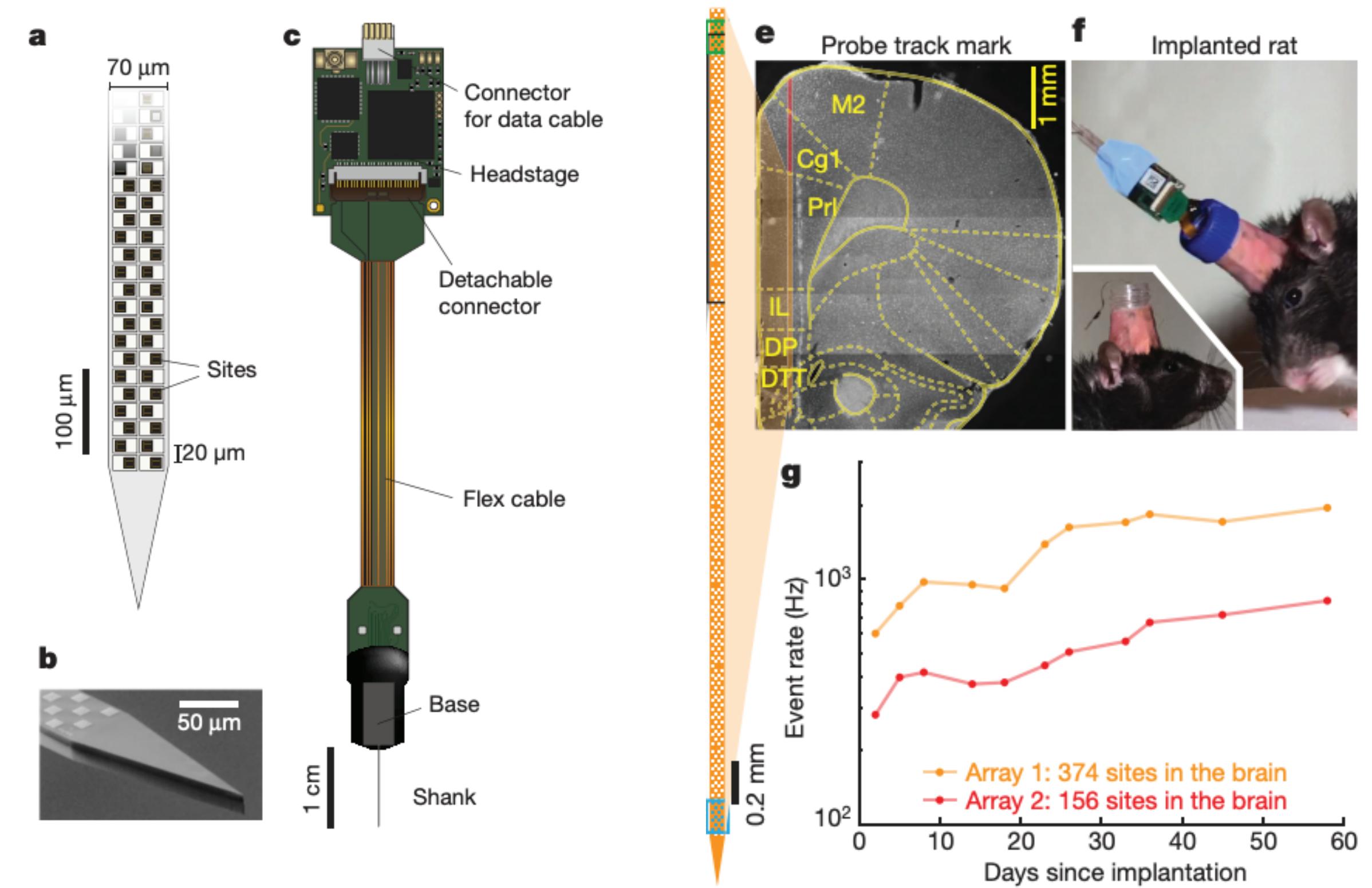
- The *membrane potential* spikes by 50-100mV during an action potential.
- The *extracellular action potential* (EAP) is roughly proportional to the total current (I_{total}) in nearby compartments of the cell.
- The EAP shows a triphasic response with a sharp negative deflection of 50-100 μ V.
- Amplitudes fall off with distance from the cell.



Neuropixels

High-density silicon probes

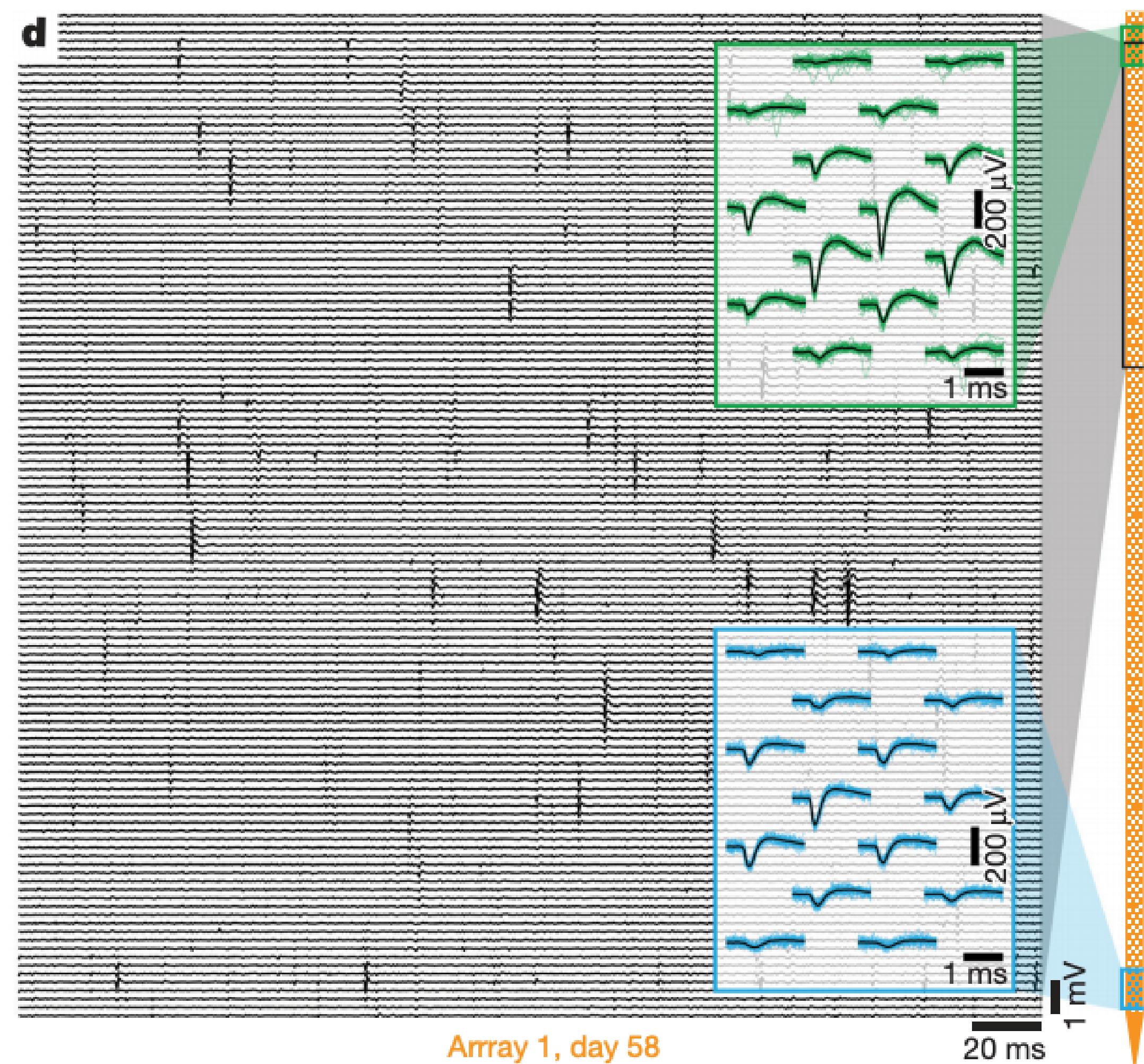
- Modern recording probes like **Neuropixels** measure the electrical activity of **hundreds of cells** across **multiple brain regions** simultaneously.
- First gen. Neuropixels had 960 recording sites spaced 20 μm apart, of which 384 could be used simultaneously.
- Finely spaced sites means that single neurons can activate 5-50 sites.
- Compare spacing to scale bar on previous slide.



Neuropixels

High-density silicon probes

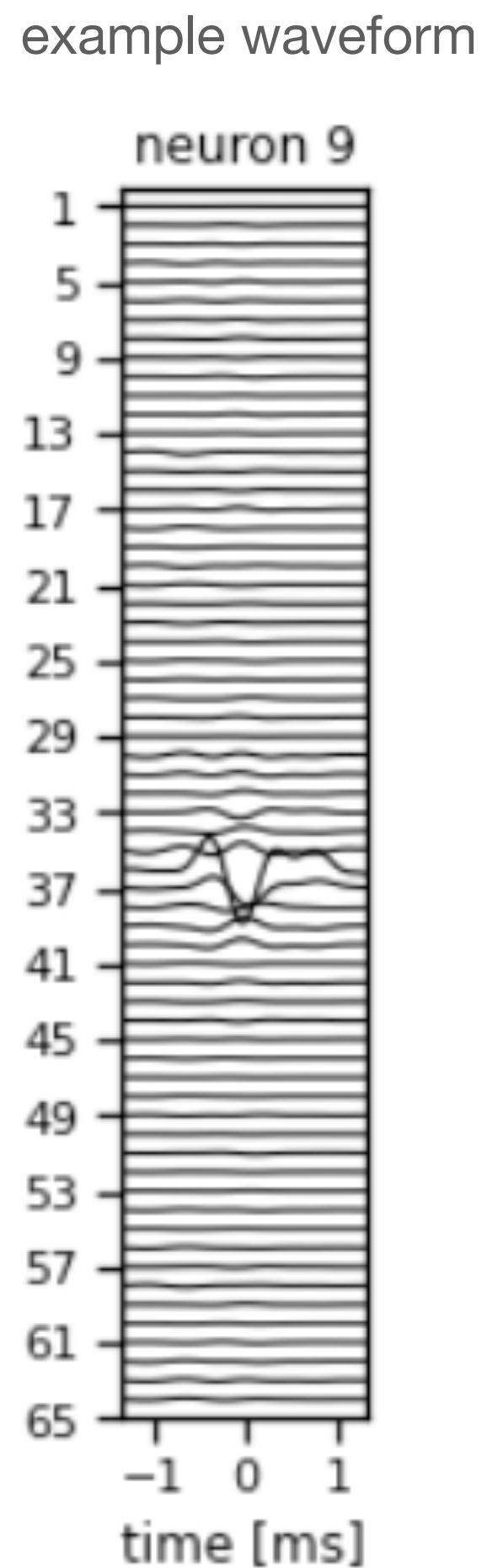
- The raw data is a **multidimensional time series of voltage measurements**, one for each recording site on the probe.
- When neurons near the probe fire an **action potential**, it registers a **spike in the voltage** on nearby channels.
- Our goal is to **find the spikes** in this time series and **assign neuron labels** based on their waveforms.



A Spike Sorting Model

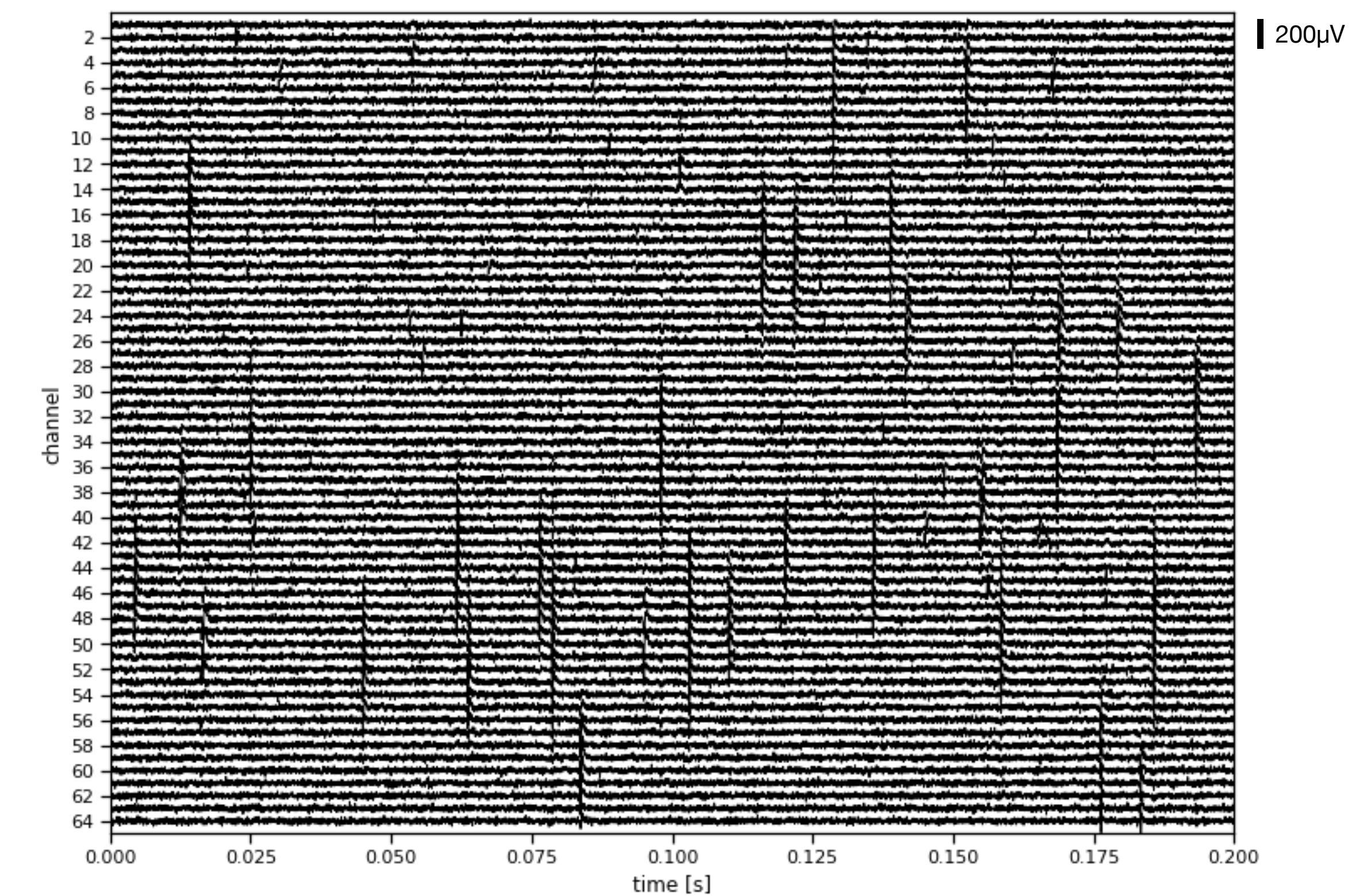
Terminology

- *Channel*: a single recording site
- *Waveform*: the voltage transient induced by a spike.
- *Sample*: a single time step's measurement of voltage across all channels.
- *Sampling frequency*: the number of samples taken per second (here, 30kHz = 30,000 samples/sec)



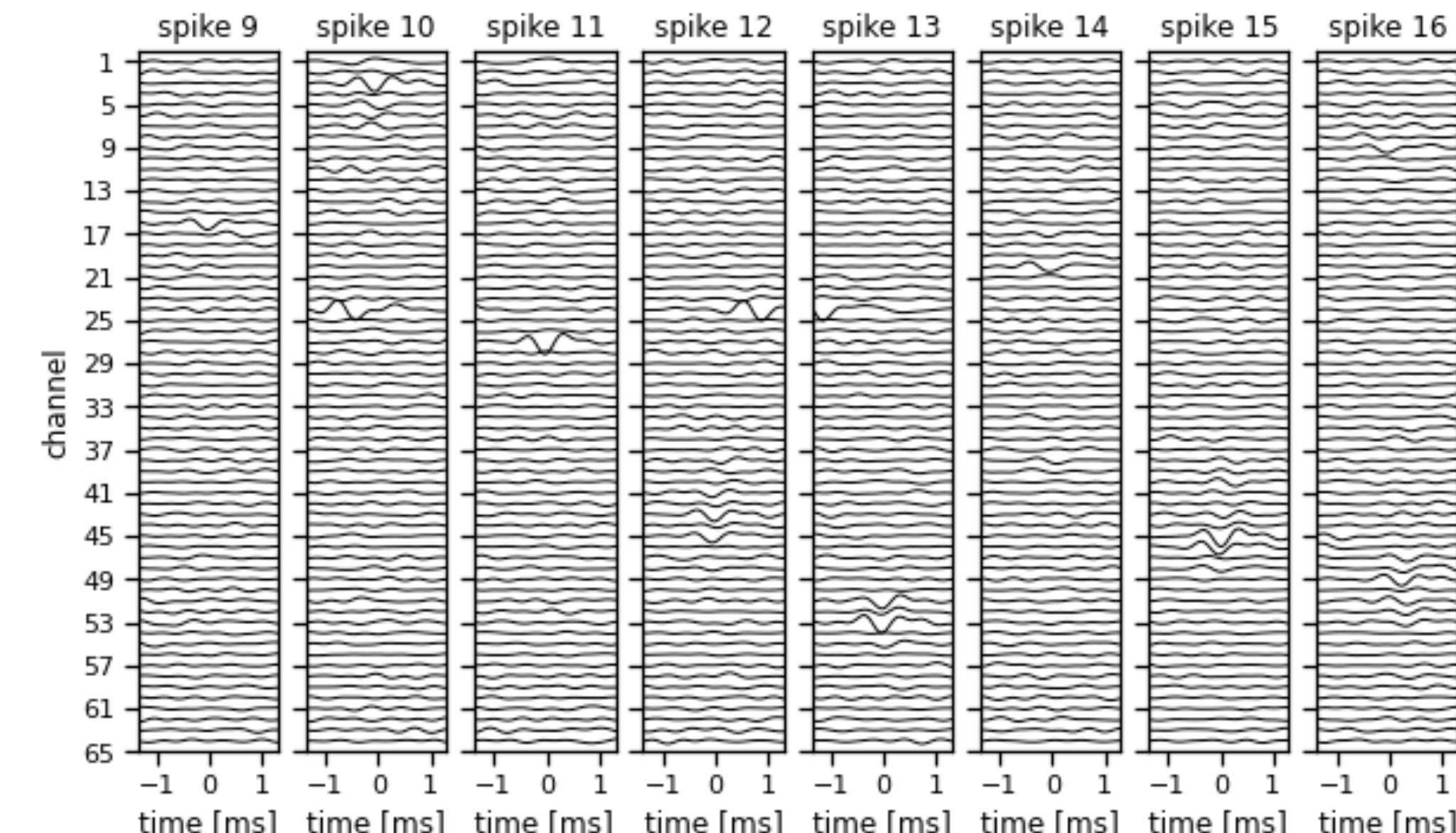
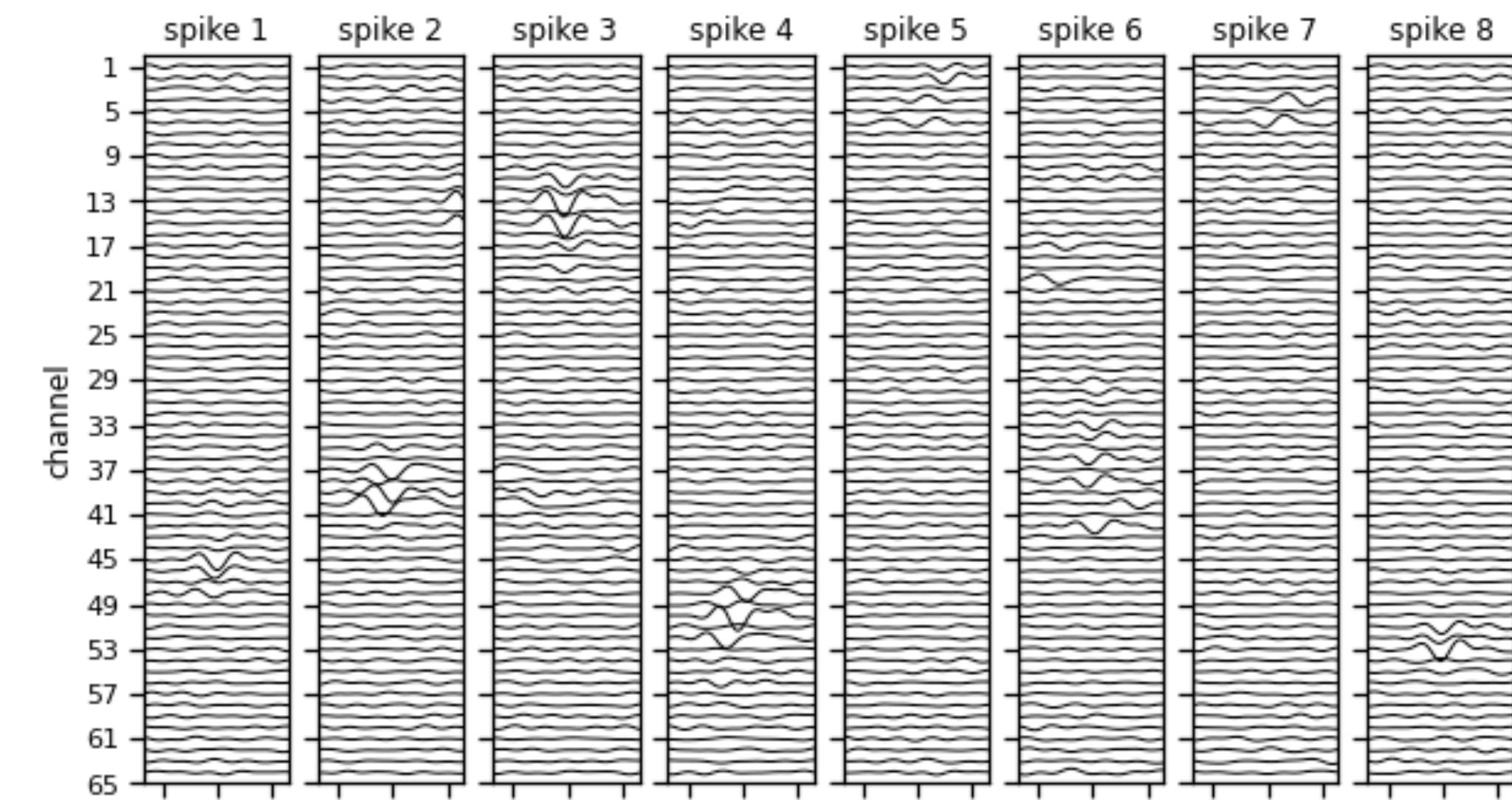
Semi-synthetic data

- We'll work with a “semi-synthetic” recording with only 64 channels (i.e. sites), but the models we develop will be agnostic to the number of channels.
- It's *semi-synthetic* because the spike times are randomly distributed and the noise is Gaussian, but the waveforms are taken from real recordings.



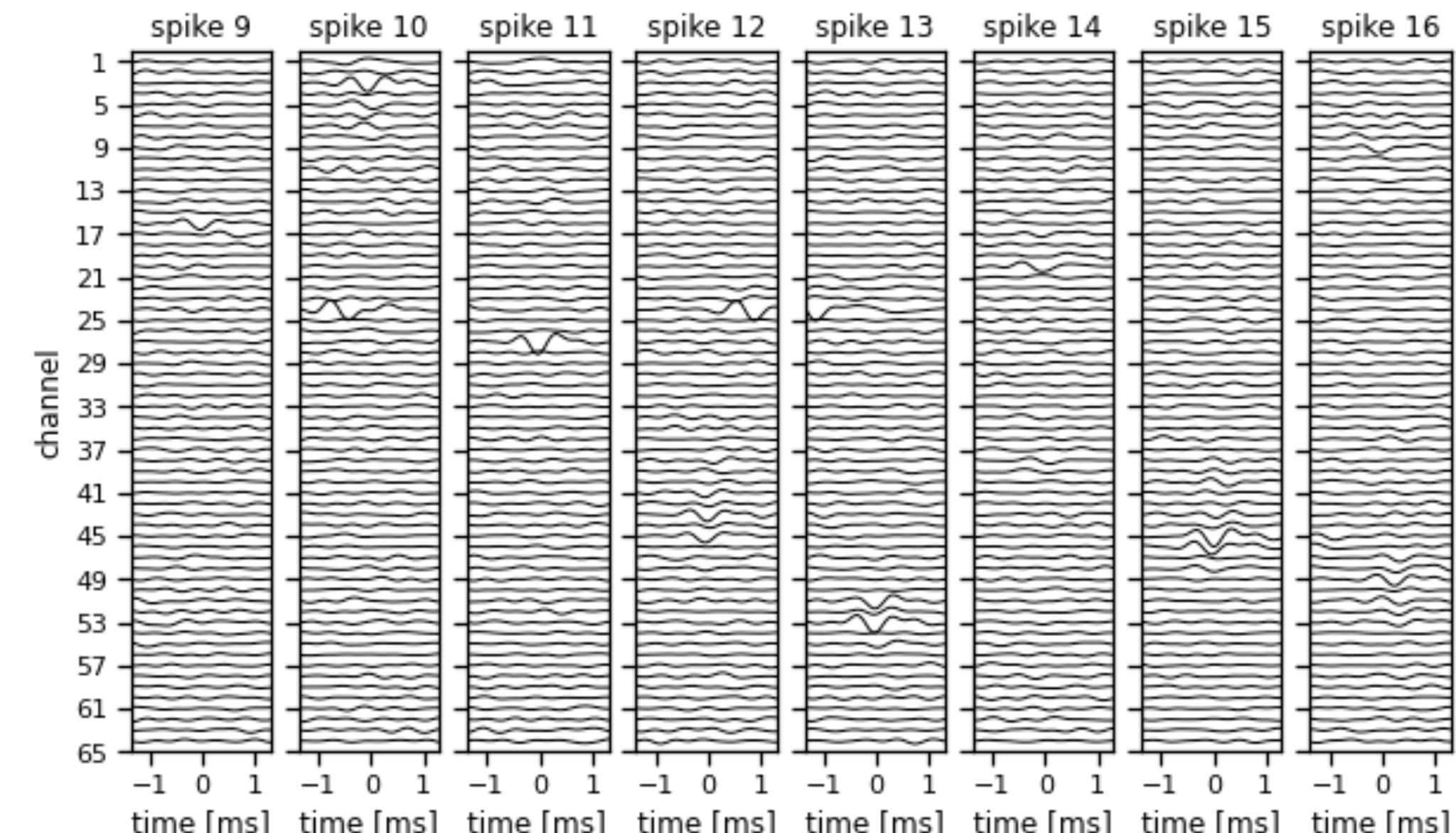
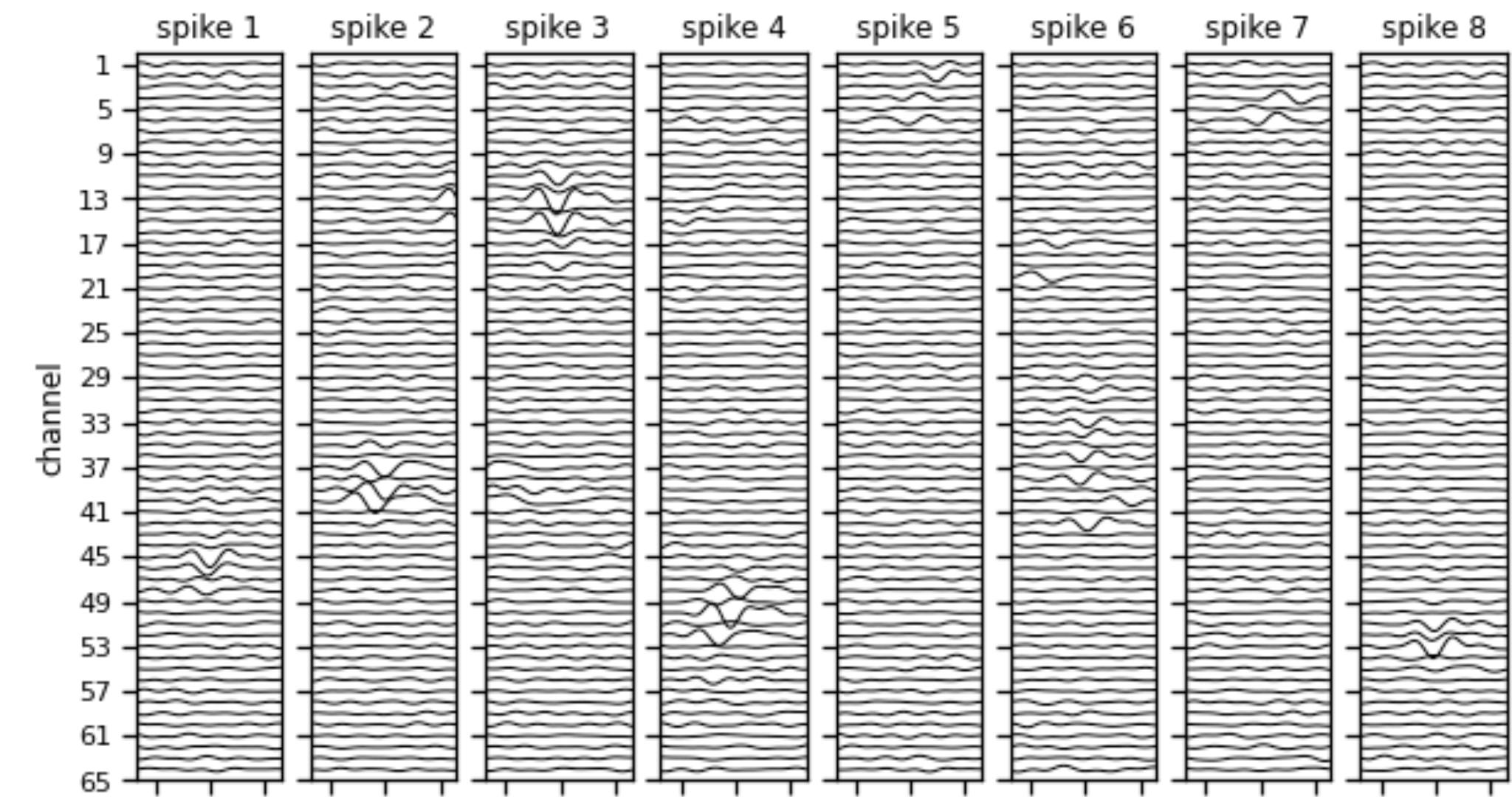
Preprocessing

- Assume the data has been **bandpass filtered** to remove high- and low-frequency artifacts/noise.
- Assume we have **standardized** the data to remove across-**channel correlations** and give each channel **unit variance**.
- Assume we have found **putative spikes** in the resulting data and extracted a **window** of $\pm 1.35\text{ms}$ around each one.



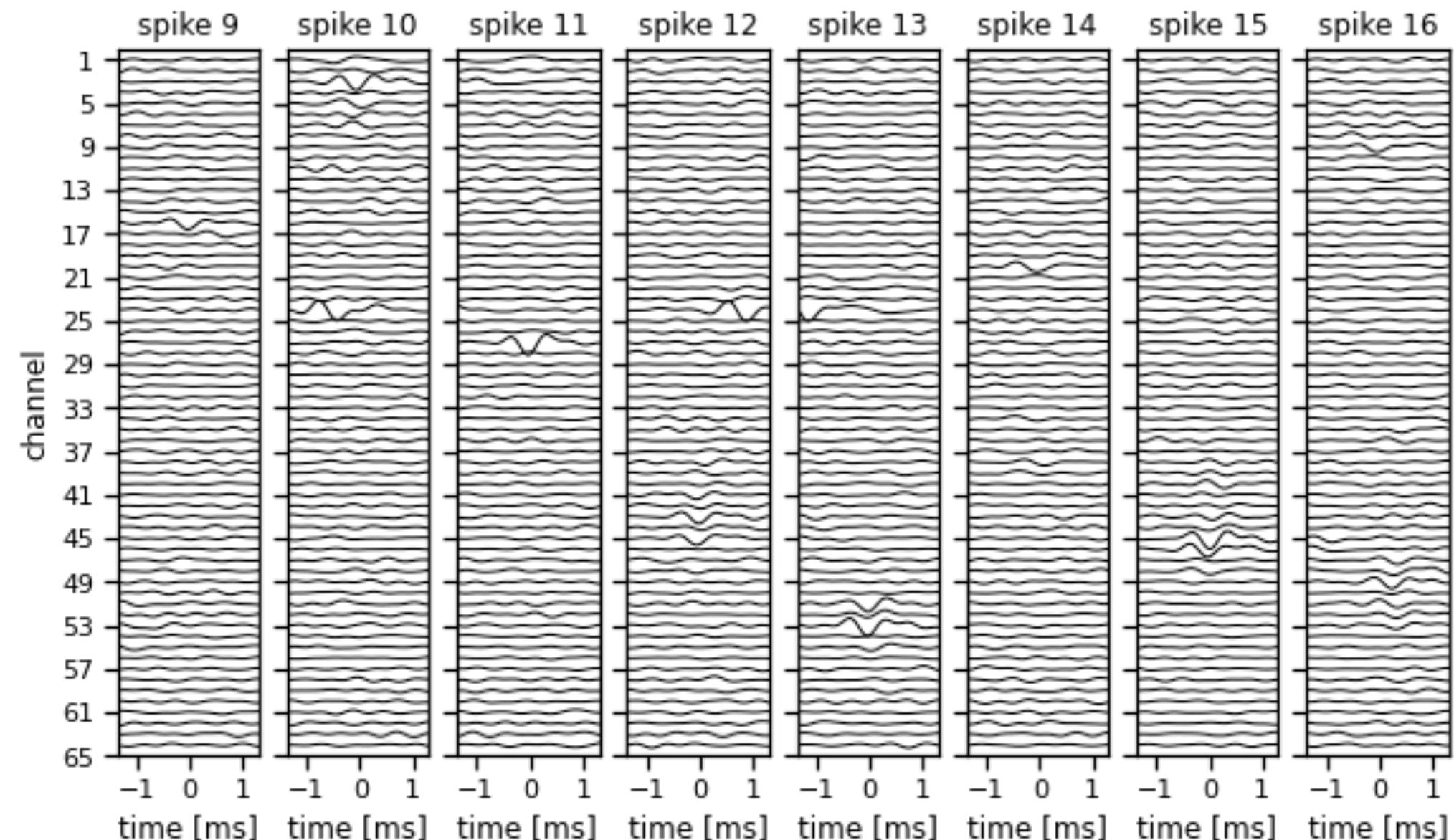
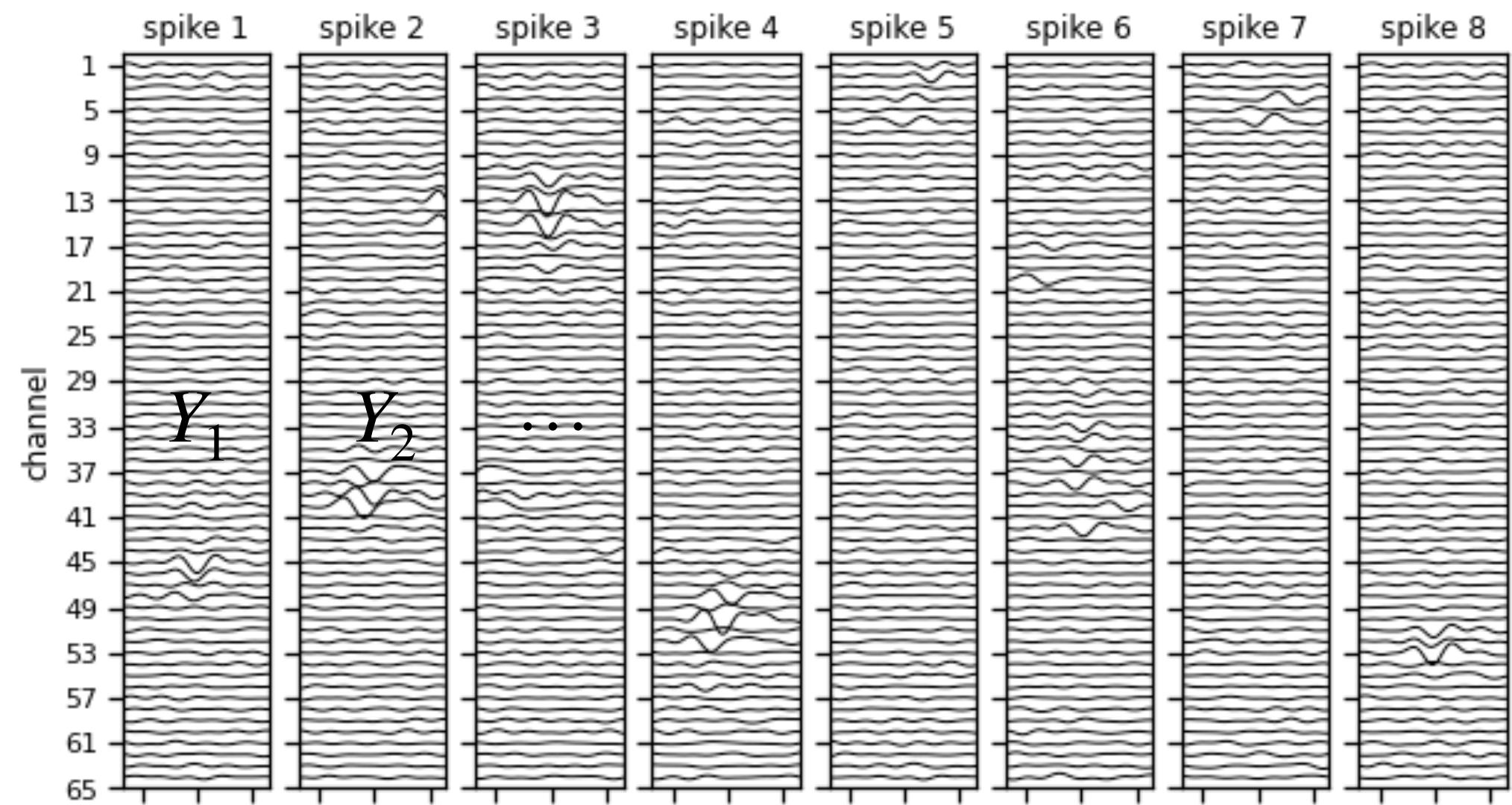
Constants

- Let S denote the number of putative **spikes**.
- C denote the number of **channels**.
- D denote the **duration** (in samples) of a spike waveform. E.g. 81 samples $\approx 2.7\text{ms}$ at 30kHz.
- N denote the (unknown) number of **neurons** that generated the spikes.
- K denote the **rank** of the templates.



Data and Latent Variables

- **Data:**
 - Let $Y \in \mathbb{R}^{S \times C \times D}$ denote the array of extracted **spike waveforms**.
- **Latent Variables:**
 - Let $n_s \in \{1, \dots, N\}$ denote the neuron to which spike s is assigned. We call it the **label**.
 - Let $a_s \in \mathbb{R}_+$ denote the **amplitude** of spike s .



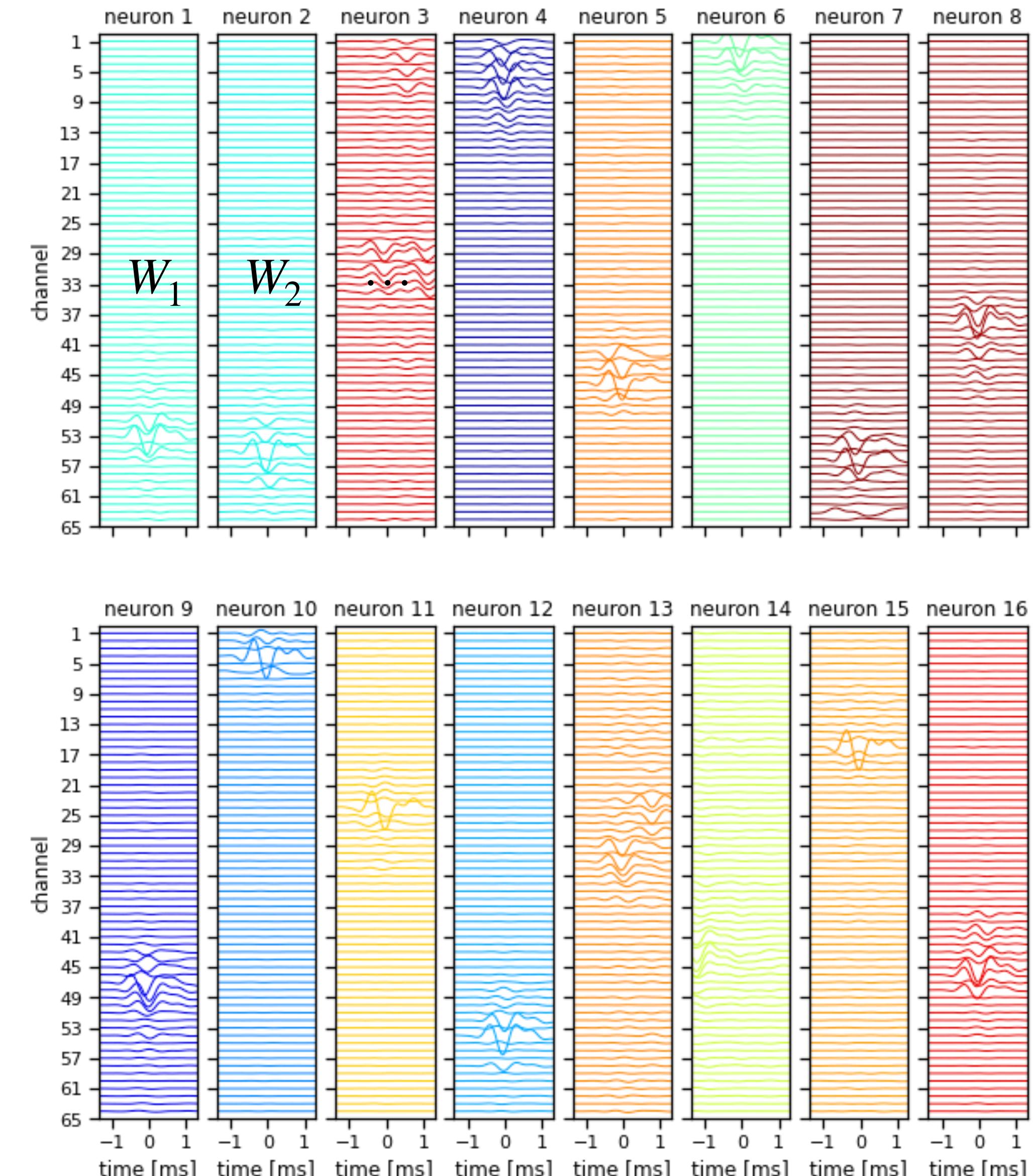
Parameters and Hyperparameters

- **Parameters:**

- Let $W \in \mathbb{R}^{N \times C \times D}$ denote the array of **waveform templates** for each neuron.
- Let $\lambda_n \in \mathbb{R}_+$ denote the inverse scale (i.e. rate) of the prior distribution on amplitudes.
- Let $\pi \in \Delta_N$ denote the prior probability of neuron labels.

- **Hyperparameters:**

- Let σ^2 denote the noise variance (shared by all channels).



Probabilistic Model

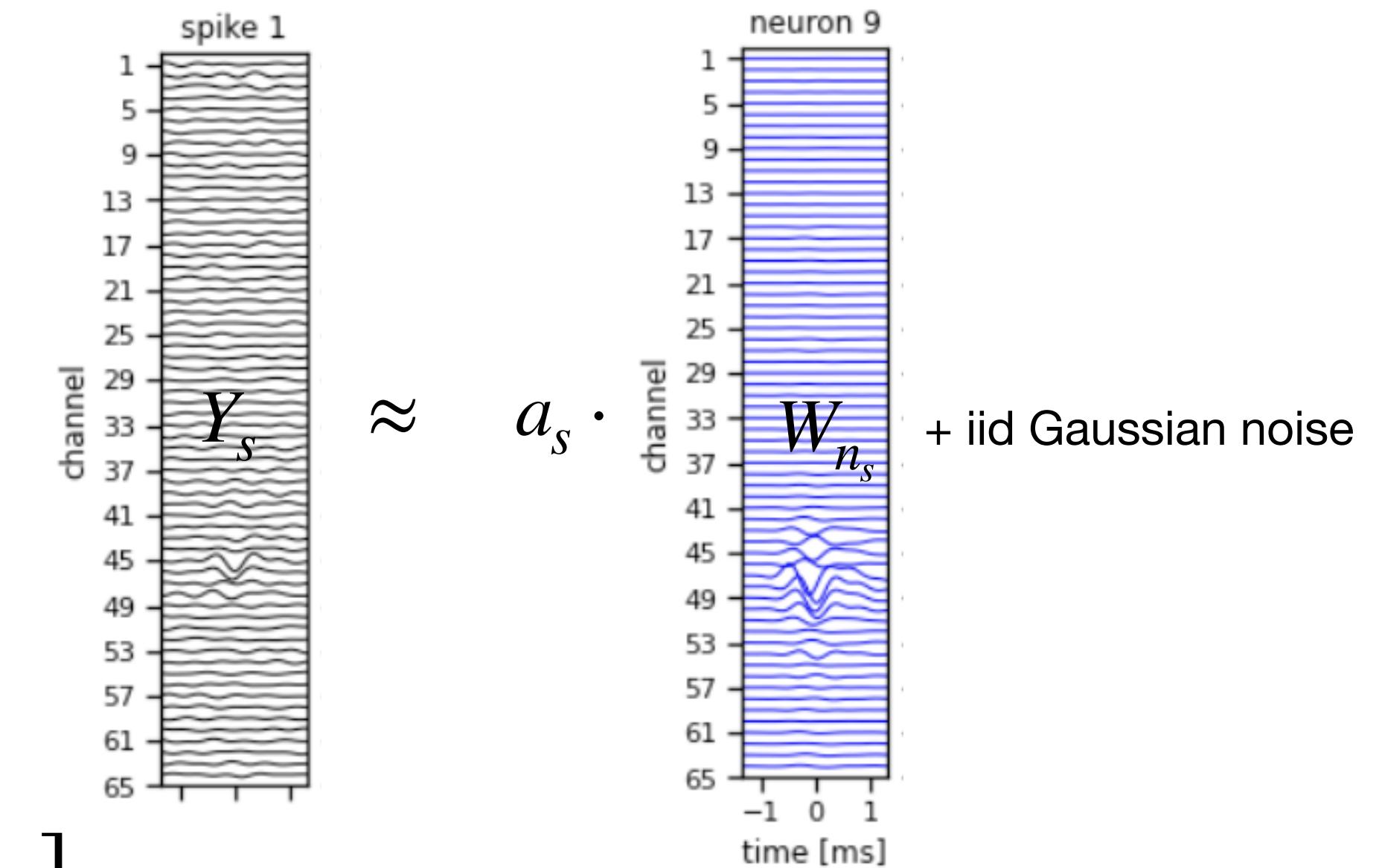
Likelihood

- Assume each spike is a noisy, scaled version of the template of the neuron that generated it.

$$p(Y_s | n_s, a_s, W, \sigma^2) = \prod_{c=1}^C \prod_{d=1}^D \mathcal{N}(y_{scd} | a_s \cdot w_{n_s cd}, \sigma^2)$$

- Taking the log and expanding the Gaussian density

$$\begin{aligned} \log p(Y_s | n_s, a_s, W, \sigma^2) &= \sum_{c=1}^C \sum_{d=1}^D \left[-\frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} (y_{scd} - a_s \cdot w_{n_s cd})^2 \right] \\ &= -\frac{CD}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \|Y_s - a_s W_{n_s}\|_F^2. \end{aligned}$$



Aside: Frobenius norm of a matrix

- The squared L² norm of a **vector** $x \in \mathbb{R}^M$ is

$$\|x\|_2^2 = \sum_{m=1}^M x_m^2 = x^\top x.$$

- The (squared) Frobenius norm of a **matrix** $X \in \mathbb{R}^{C \times D}$ is the sum of squared entries, or the (squared) L² norm of the **vectorized** (flattened) matrix:

$$\|X\|_F^2 = \sum_{c=1}^C \sum_{d=1}^D x_{cd}^2 = \text{vec}(X)^\top \text{vec}(X) = \|\text{vec}(X)\|_2^2$$

- We can write it equivalently as a **trace**:

$$\|X\|_F^2 = \text{Tr}(X^\top X) = \text{Tr}(XX^\top)$$

Probabilistic Model

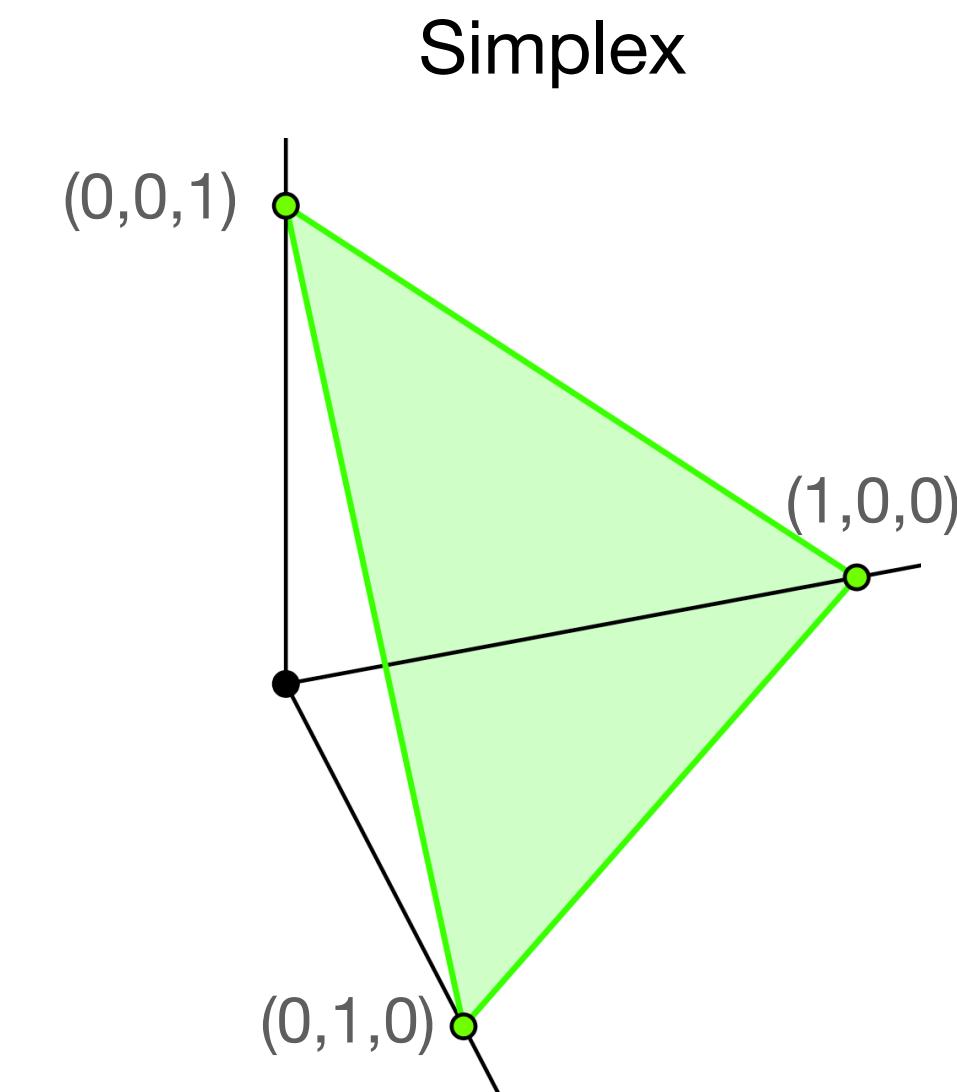
Prior on spike labels

- Assume the neuron labels are drawn according to discrete distribution π

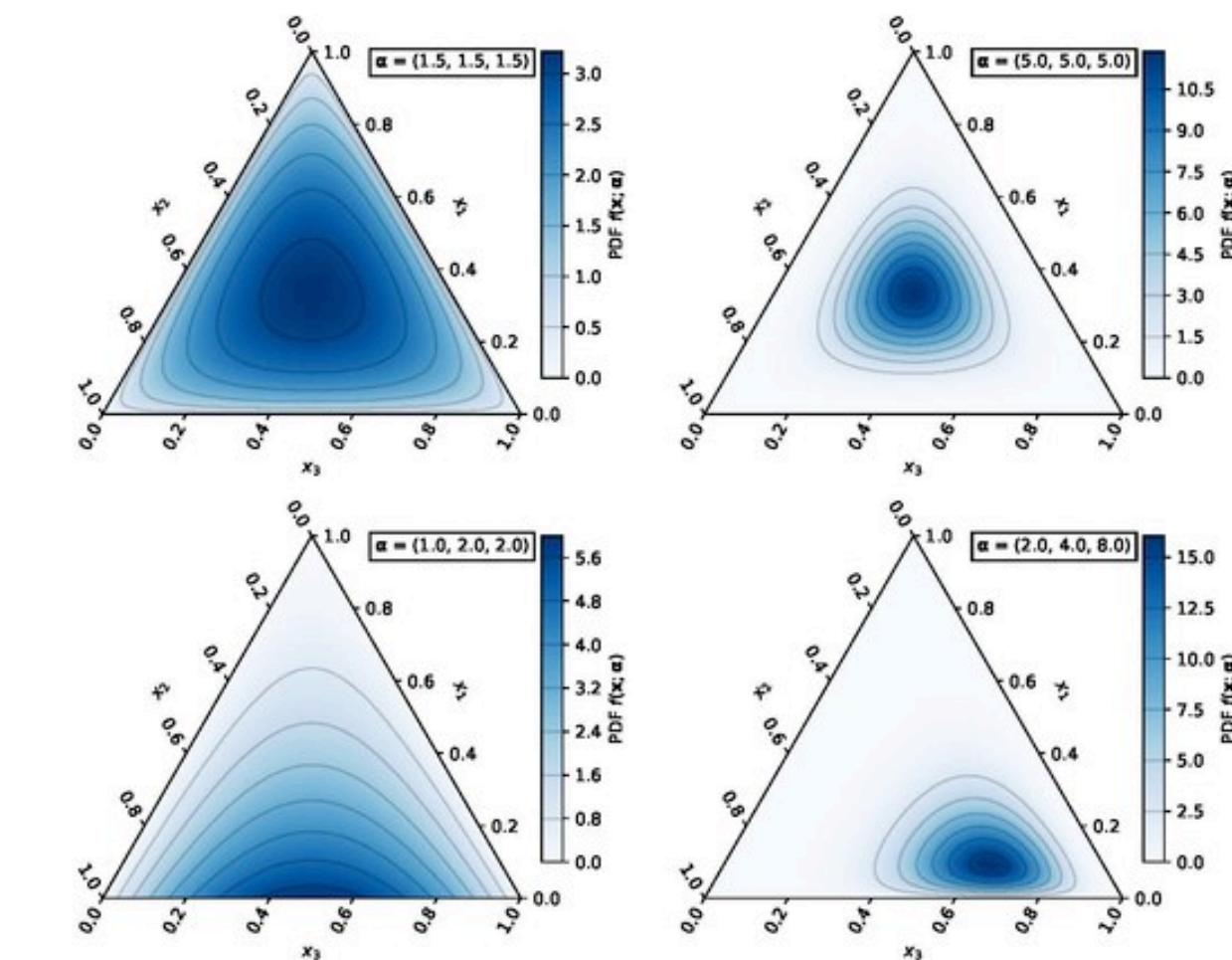
$$n_s \sim \pi \iff \Pr(n_s = n \mid \pi) = \pi_n$$

- Give π a uniform prior distribution, or equivalently, a Dirichlet with concentration 1:

$$\pi \sim \text{Dir}(1_N) \iff p(\pi) \propto 1$$



Simplex



<https://en.wikipedia.org/wiki/Simplex>

https://en.wikipedia.org/wiki/Dirichlet_distribution

Probabilistic Model

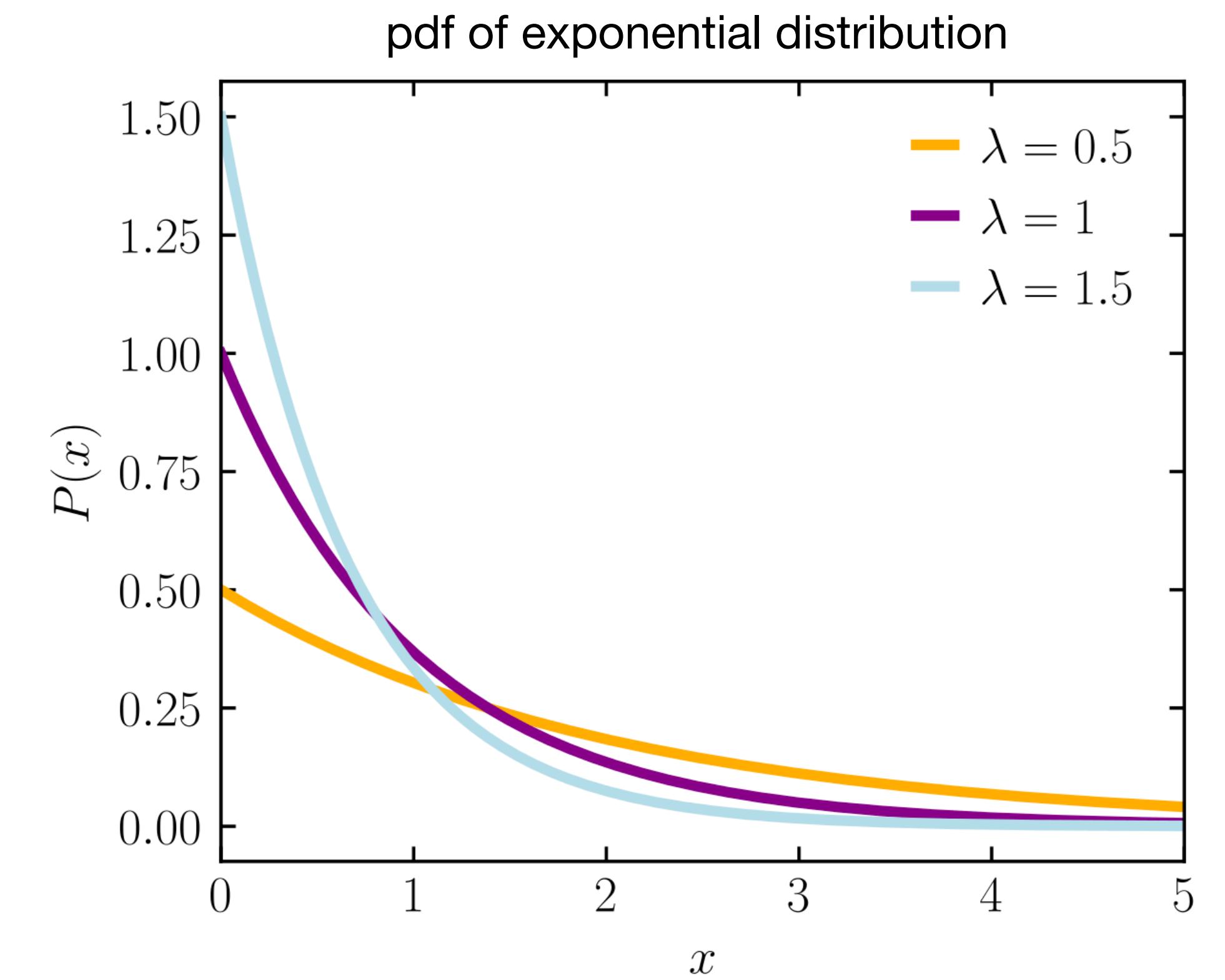
Prior on spike amplitudes

- Assume the spike amplitudes are drawn from an exponential distribution.

$$a_s \sim \text{Exp}(\lambda_{n_s}) \iff p(a_s | n_s, \lambda) = \text{Exp}(a_s | \lambda_{n_s}) \\ = \lambda_{n_s} e^{-\lambda_{n_s} a_s}$$

- The inverse-scale (i.e. rate) depends on the spike label n_s .
- Assume the inverse-scale has an (improper) uniform prior on \mathbb{R}_+

$$p(\lambda_n) \propto 1$$



Probabilistic Model

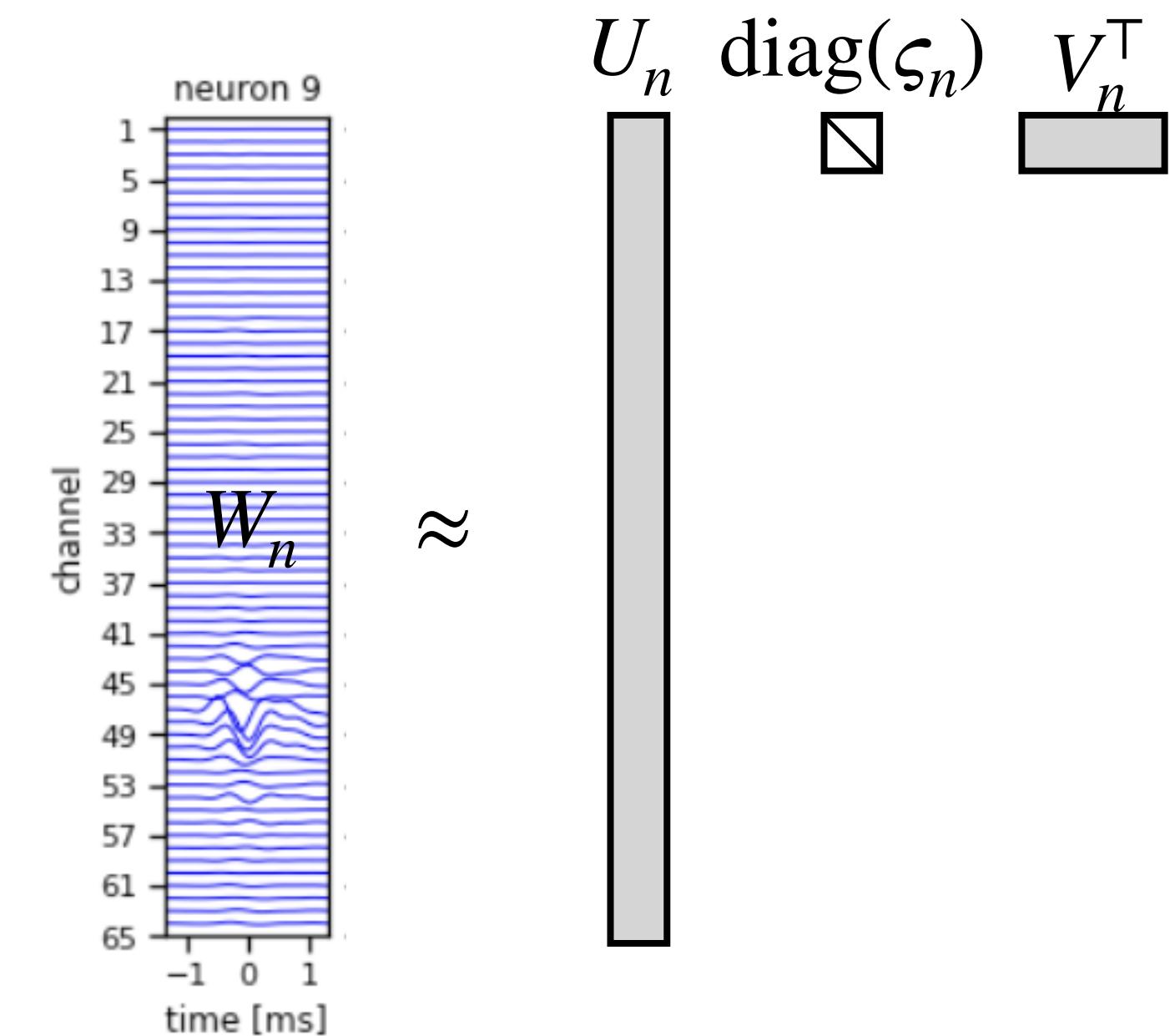
Prior on templates

- Finally, in order for the amplitudes to be meaningful, we need to **constrain the magnitude** of the templates W_n .
- Specifically, we assume $\|W_n\|_F = 1$ for each neuron.
- Moreover, we assume the templates are low rank:

$$W_n = U_n \text{diag}(\varsigma_n) V_n^\top$$

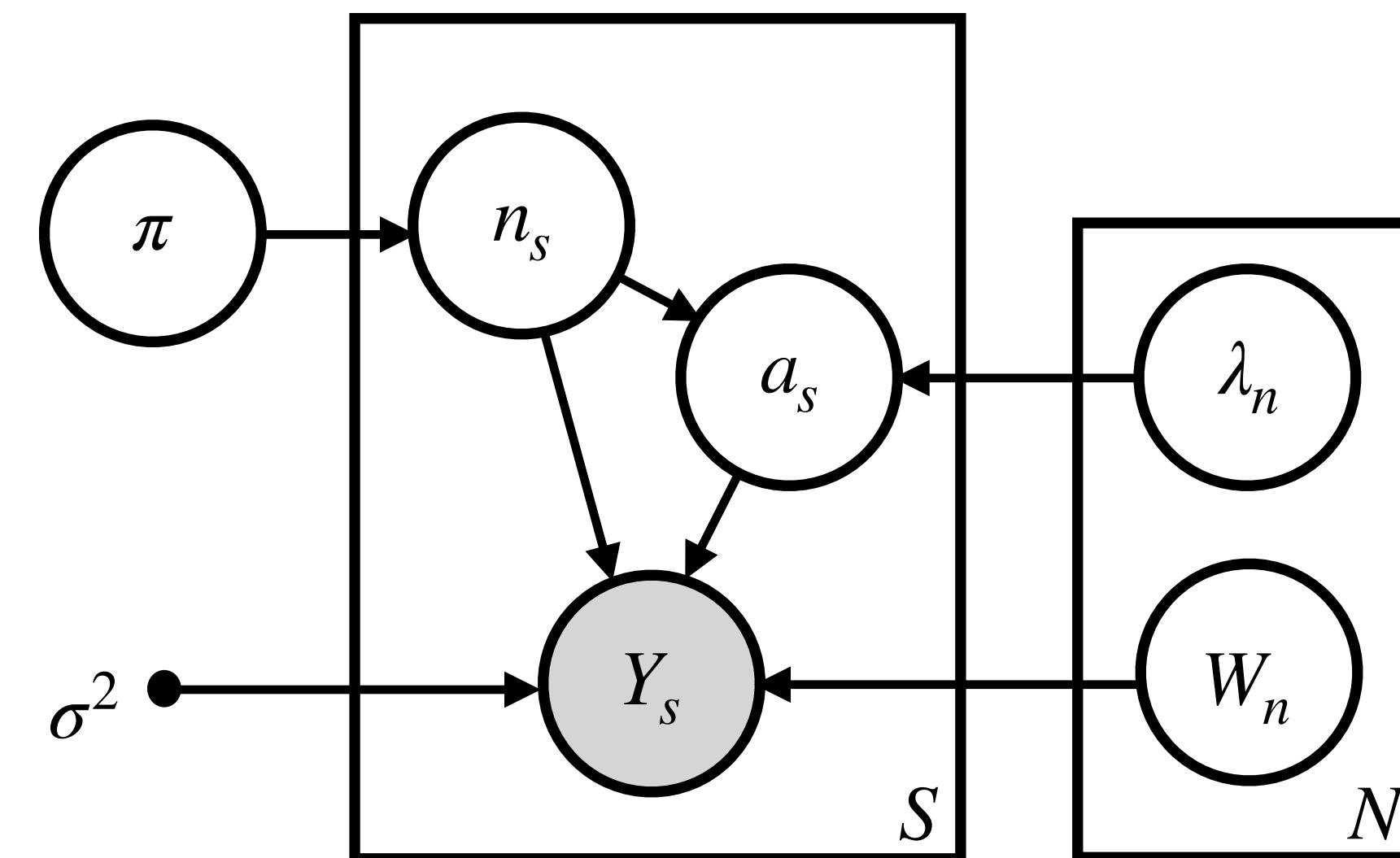
where $U_n \in \mathbb{R}^{C \times K}$ and $V_n \in \mathbb{R}^{D \times K}$ are orthogonal matrices, and $\varsigma_n \in \mathbb{R}_+^K$ are the singular values.

- Exercise:** show that $\|W_n\|_F = 1 \iff \|\varsigma_n\|_2 = 1$ using the “trace” definition of the Frobenius norm: $\|X\|_F^2 = \text{Tr}(X^\top X)$.
- We assume W_n is uniformly distributed on the set of rank- K , unit-norm matrices, which we call \mathcal{S}_K .



Probabilistic Model

Graphical Model

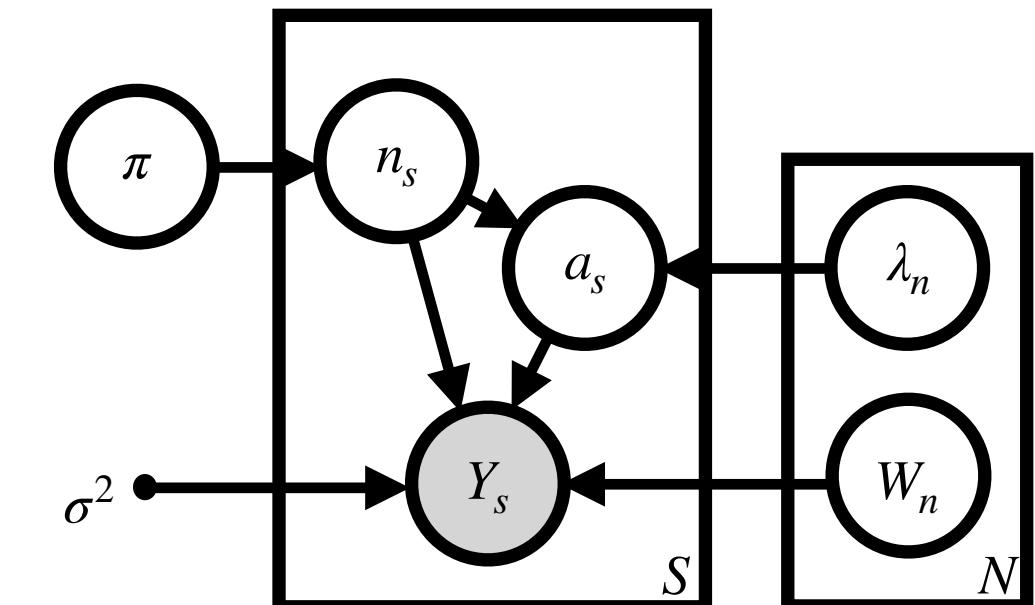


Inference

Maximum a posteriori estimation

Coordinate ascent

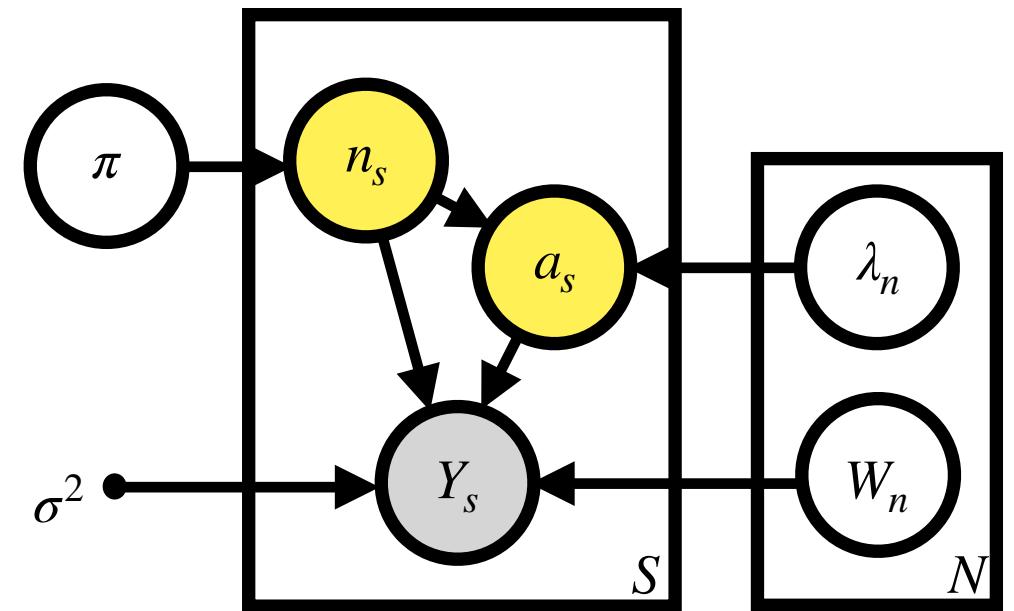
- Initialize parameters W, λ, π
- Iterate until convergence:
 1. Optimize **latent variables** n_s, a_s for each spike s .
 2. Optimize **templates** W_n for each neuron n .
 3. Optimize **amplitude rates** λ_n for each neuron n .
 4. Optimize **neuron probabilities** π .



[In each case, maximize log joint probability wrt one variable, holding others fixed.]

Maximum a posteriori estimation

Optimizing the latent variables



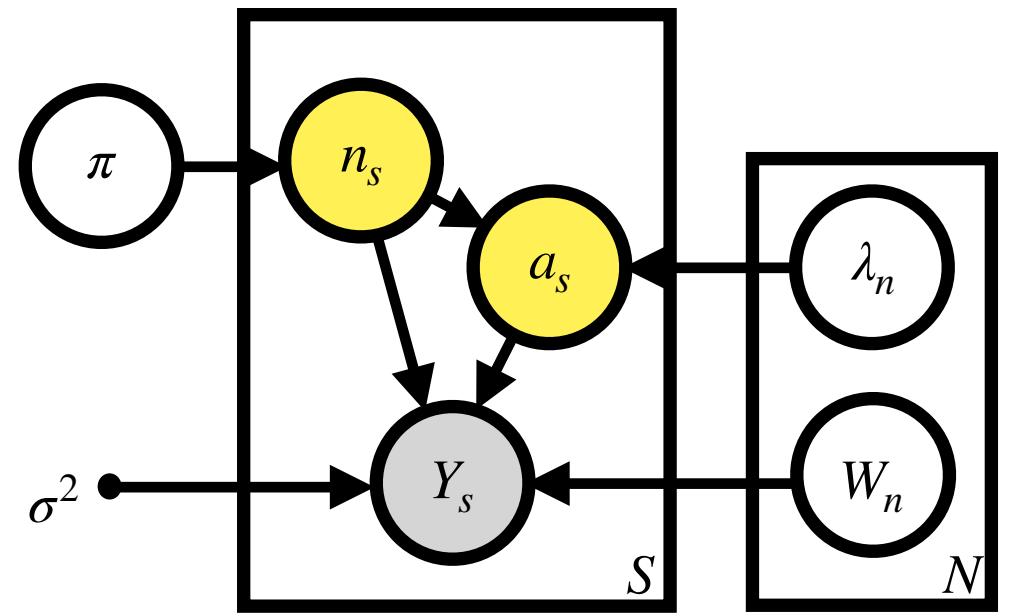
Maximum a posteriori estimation

Optimizing the latent variables

- Let

$$\mathcal{L}(n_s, a_s) \triangleq \log p(Y_s | n_s, a_s, W, \sigma^2) + \log p(n_s | \pi) + \log p(a_s | n_s, \lambda)$$

$$= -\frac{1}{2\sigma^2} \|Y_s - a_s W_{n_s}\|_F^2 + \log \pi_{n_s} - a_s \lambda_{n_s} + c$$



Maximum a posteriori estimation

Optimizing the latent variables

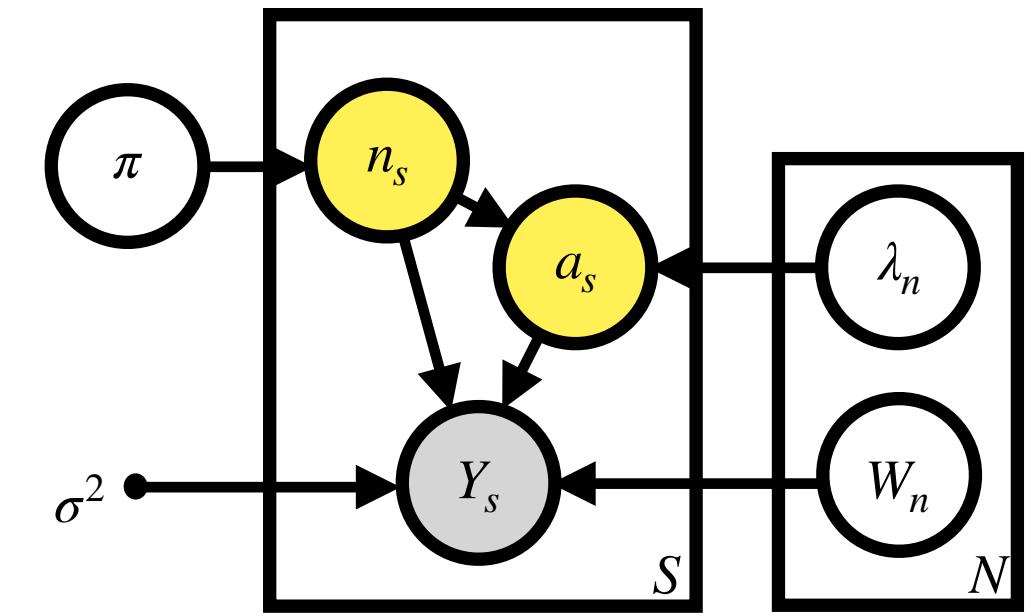
- Let

$$\mathcal{L}(n_s, a_s) \triangleq \log p(Y_s | n_s, a_s, W, \sigma^2) + \log p(n_s | \pi) + \log p(a_s | n_s, \lambda)$$

$$= -\frac{1}{2\sigma^2} \|Y_s - a_s W_{n_s}\|_F^2 + \log \pi_{n_s} - a_s \lambda_{n_s} + c$$

- Focus on the amplitude first. As a function of a_s ,

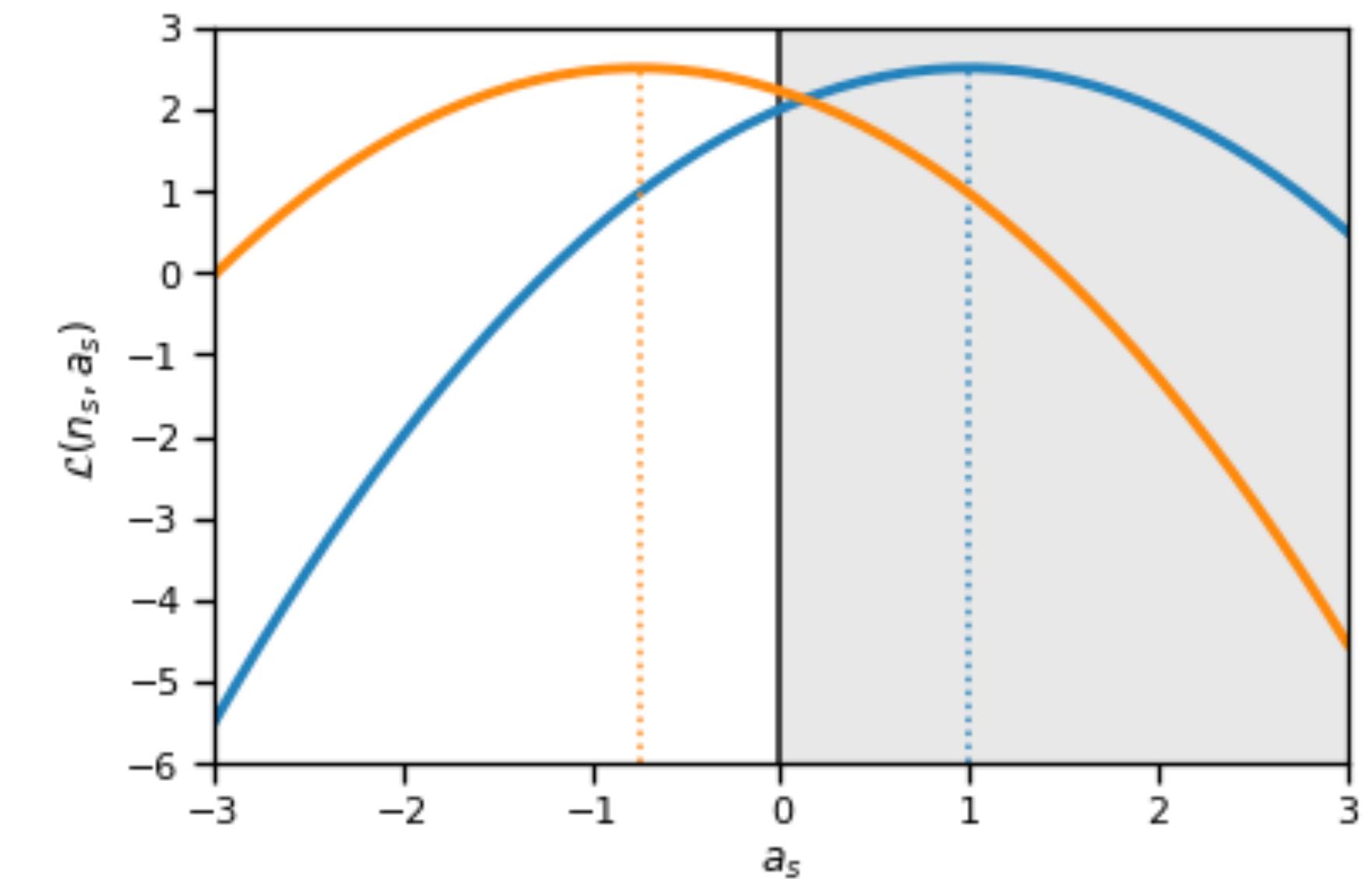
$$\begin{aligned} \mathcal{L}(n_s, a_s) &= -\frac{1}{2\sigma^2} \text{Tr} \left((Y_s - a_s W_{n_s})^\top (Y_s - a_s W_{n_s}) \right) - a_s \lambda_{n_s} + c \\ &\propto -\frac{1}{2} \text{Tr} \left((Y_s - a_s W_{n_s})^\top (Y_s - a_s W_{n_s}) \right) - a_s \sigma^2 \lambda_{n_s} + c \\ &= \left(\text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s} \right) a_s - \frac{1}{2} \text{Tr}(W_{n_s}^\top W_{n_s}) a_s^2 + c \\ &= \left(\text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s} \right) a_s - \frac{1}{2} a_s^2 + c \end{aligned}$$



Maximum a posteriori estimation

Completing the square and solving for the optimal amplitudes

- The objective is a quadratic function of the amplitude a_s .

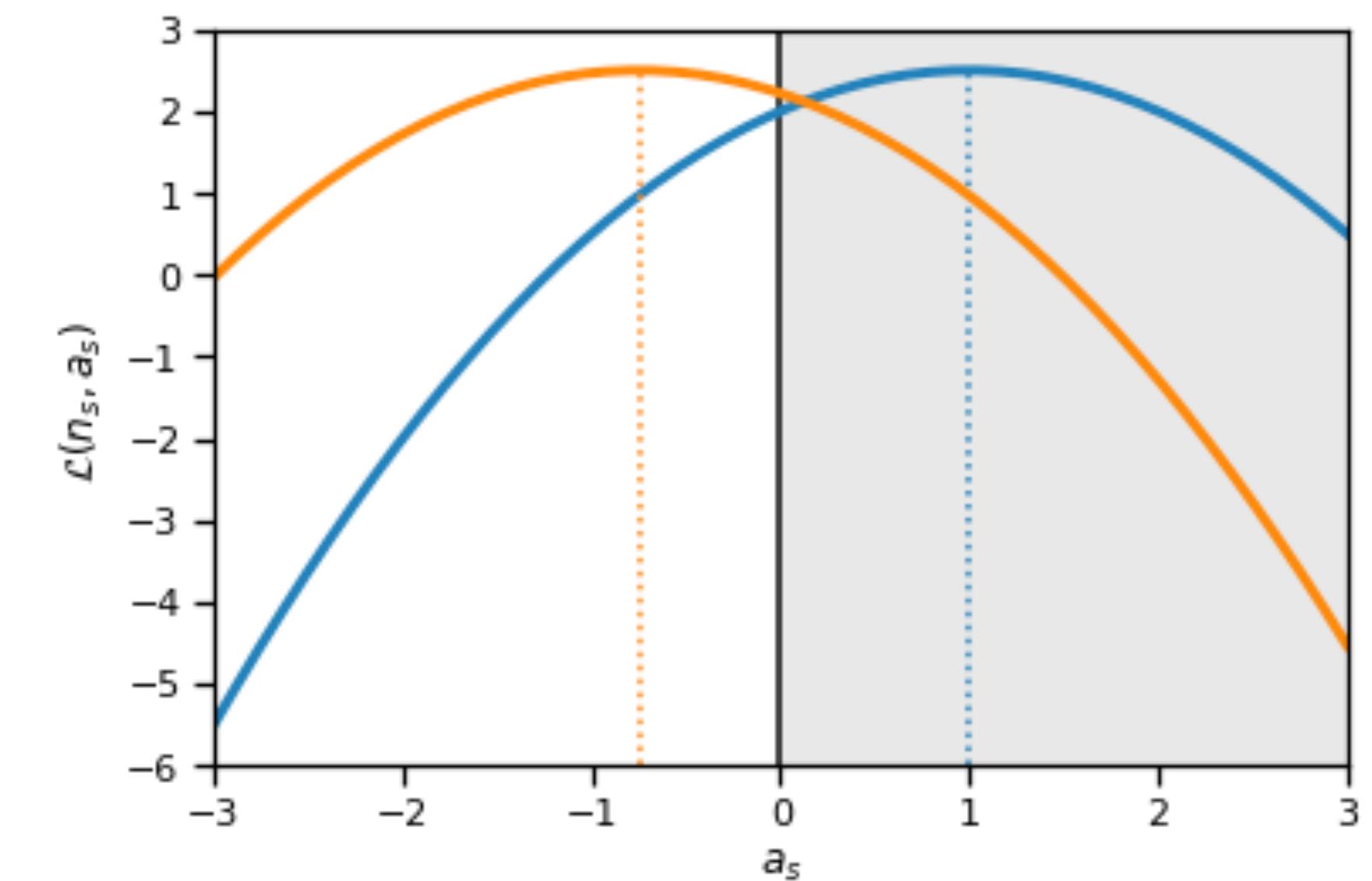


Maximum a posteriori estimation

Completing the square and solving for the optimal amplitudes

- The objective is a quadratic function of the amplitude a_s .
- **Complete the square to show**

$$\begin{aligned}\mathcal{L}(n_s, a_s) &= \left(\text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s} \right) a_s - \frac{1}{2} a_s^2 + c \\ &= -\frac{1}{2} \left(a_s - (\text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s}) \right)^2 + c\end{aligned}$$



Maximum a posteriori estimation

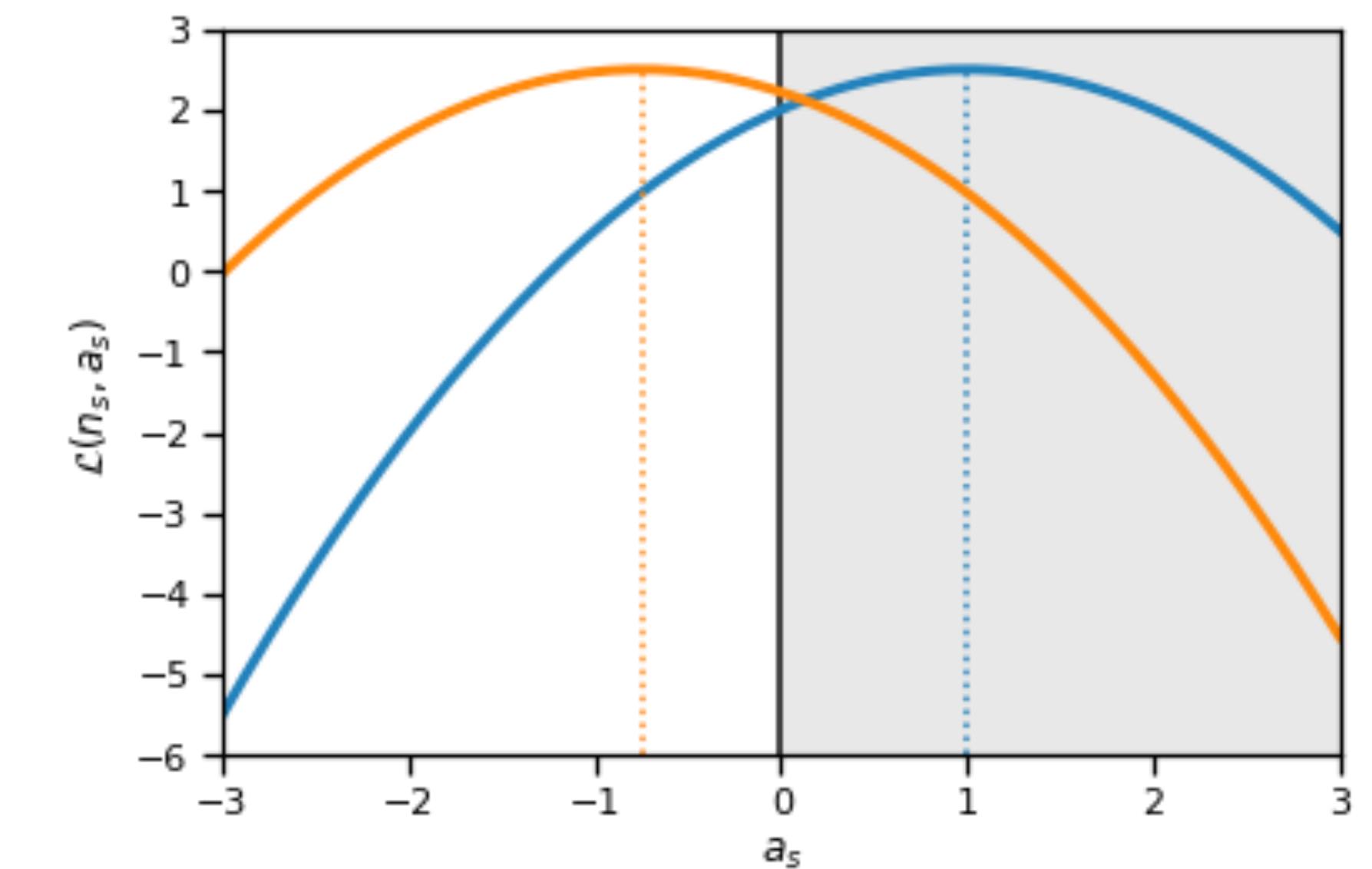
Completing the square and solving for the optimal amplitudes

- The objective is a quadratic function of the amplitude a_s .
- **Complete the square to show**

$$\begin{aligned}\mathcal{L}(n_s, a_s) &= \left(\text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s} \right) a_s - \frac{1}{2} a_s^2 + c \\ &= -\frac{1}{2} \left(a_s - (\text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s}) \right)^2 + c\end{aligned}$$

- The maximum is obtained at

$$a_s^\star(n_s) = \max \left\{ 0, \text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s} \right\}$$



Maximum a posteriori estimation

Solving for the optimal labels

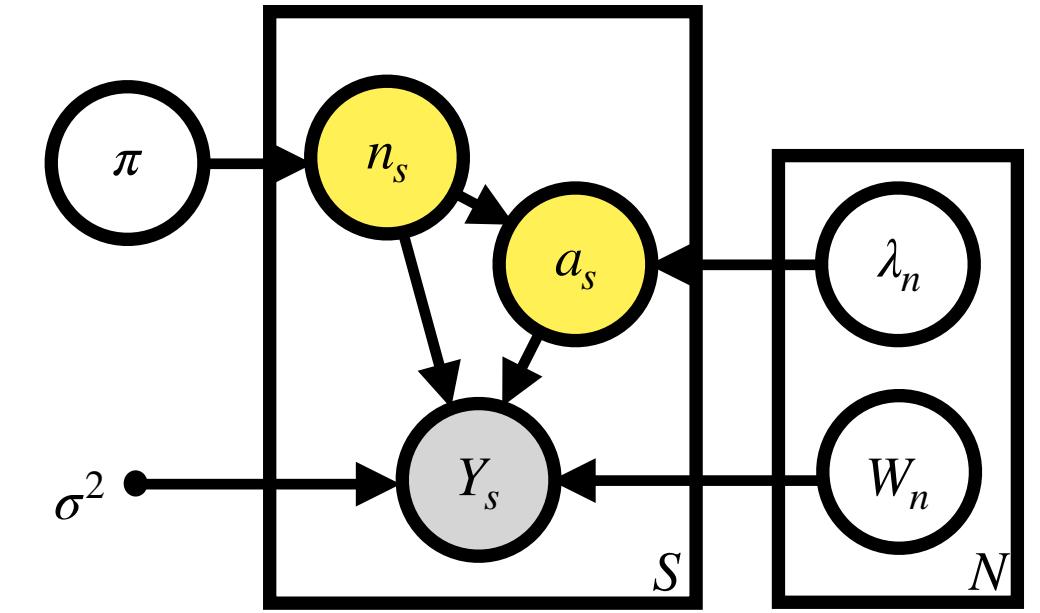
Plug in $a_s^*(n_s) = \max \left\{ 0, \text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s} \right\}$ to get

$$\mathcal{L}(n_s) \triangleq \mathcal{L}(n_s, a_s^*(n_s))$$

$$= \left(\text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s} \right) a_s^*(n_s) - \frac{1}{2} (a_s^*(n_s))^2 + \log \pi_{n_s} + c$$

$$= \frac{1}{2} (a_s^*(n_s))^2 + \log \pi_{n_s} + c$$

Then set $n_s = \operatorname{argmax} \mathcal{L}(n_s)$ and $a_s = a_s^*(n_s)$ for each spike.



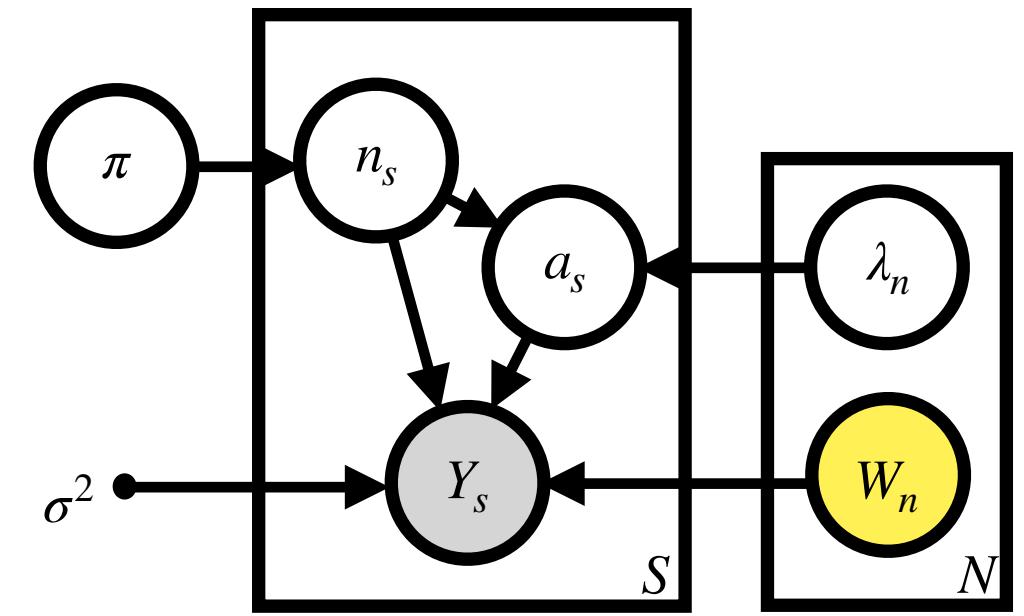
Maximum a posteriori estimation

Optimizing the templates

- As a function of the template W_n , the log joint probability is

$$\mathcal{L}(W_n) \triangleq \log p(Y, \{a_s, n_s\}_{s=1}^S, W, \lambda, \pi \mid \sigma^2)$$

$$= -\frac{1}{2\sigma^2} \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot \|Y_s - a_s W_n\|_F^2 + c$$

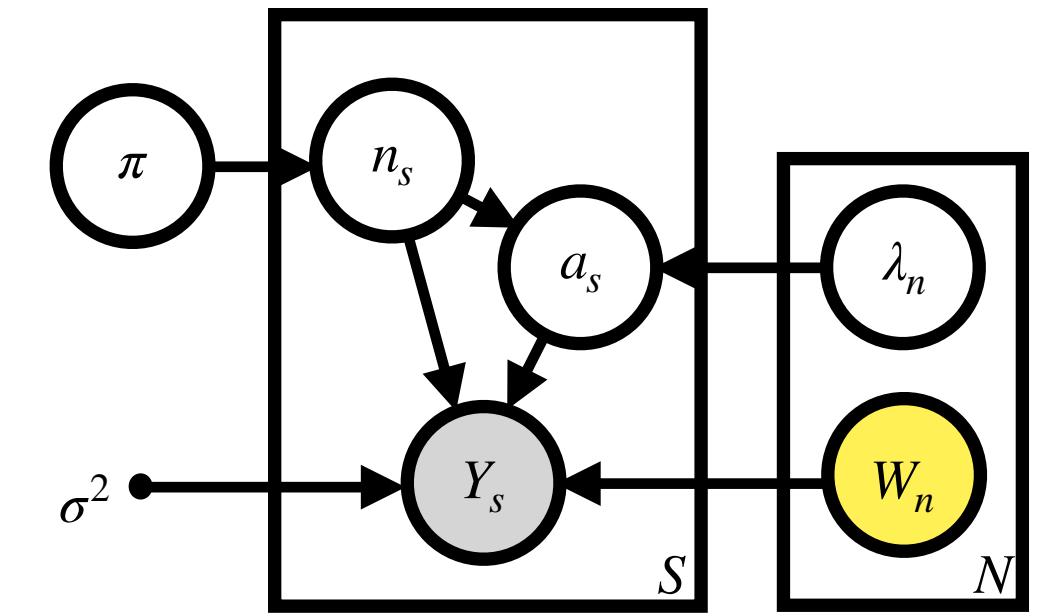


Maximum a posteriori estimation

Optimizing the templates

- As a function of the template W_n , the log joint probability is

$$\begin{aligned}\mathcal{L}(W_n) &\triangleq \log p(Y, \{a_s, n_s\}_{s=1}^S, W, \lambda, \pi \mid \sigma^2) \\ &= -\frac{1}{2\sigma^2} \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot \|Y_s - a_s W_n\|_F^2 + c \\ &= \frac{1}{\sigma^2} \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot a_s \text{Tr}(Y_s^\top W_n) + c\end{aligned}$$



Maximum a posteriori estimation

Optimizing the templates

- As a function of the template W_n , the log joint probability is

$$\mathcal{L}(W_n) \triangleq \log p(Y, \{a_s, n_s\}_{s=1}^S, W, \lambda, \pi | \sigma^2)$$

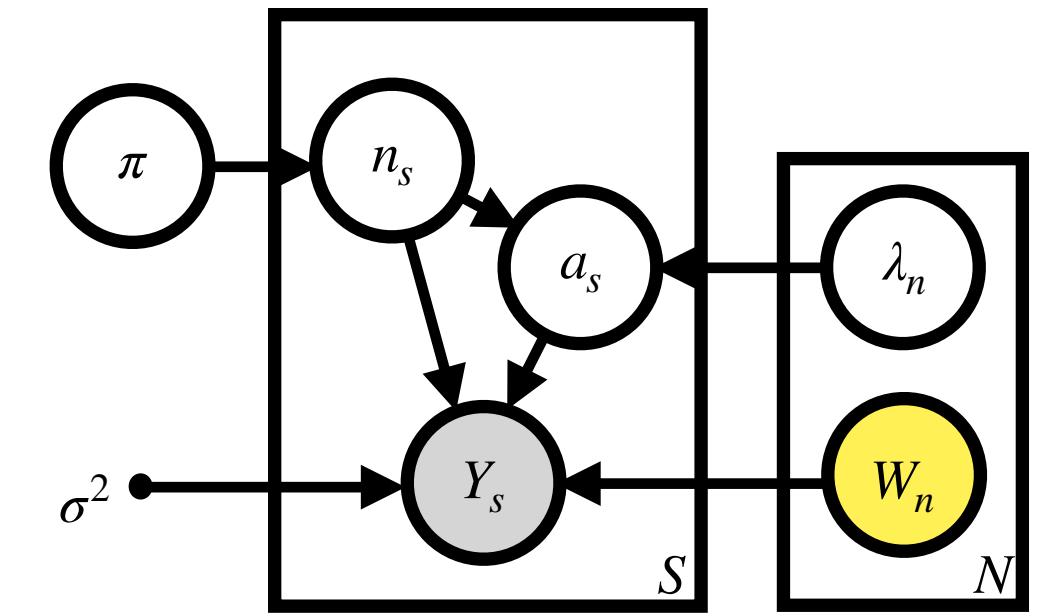
$$= -\frac{1}{2\sigma^2} \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot \|Y_s - a_s W_n\|_F^2 + c$$

$$= \frac{1}{\sigma^2} \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot a_s \text{Tr}(Y_s^\top W_n) + c$$

$$\propto \text{Tr}(\bar{Y}_n^\top W_n) + c,$$

where

$$\bar{Y}_n = \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot a_s Y_s.$$



Maximum a posteriori estimation

Optimizing the templates

- As a function of the template W_n , the log joint probability is

$$\mathcal{L}(W_n) \triangleq \log p(Y, \{a_s, n_s\}_{s=1}^S, W, \lambda, \pi | \sigma^2)$$

$$= -\frac{1}{2\sigma^2} \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot \|Y_s - a_s W_n\|_F^2 + c$$

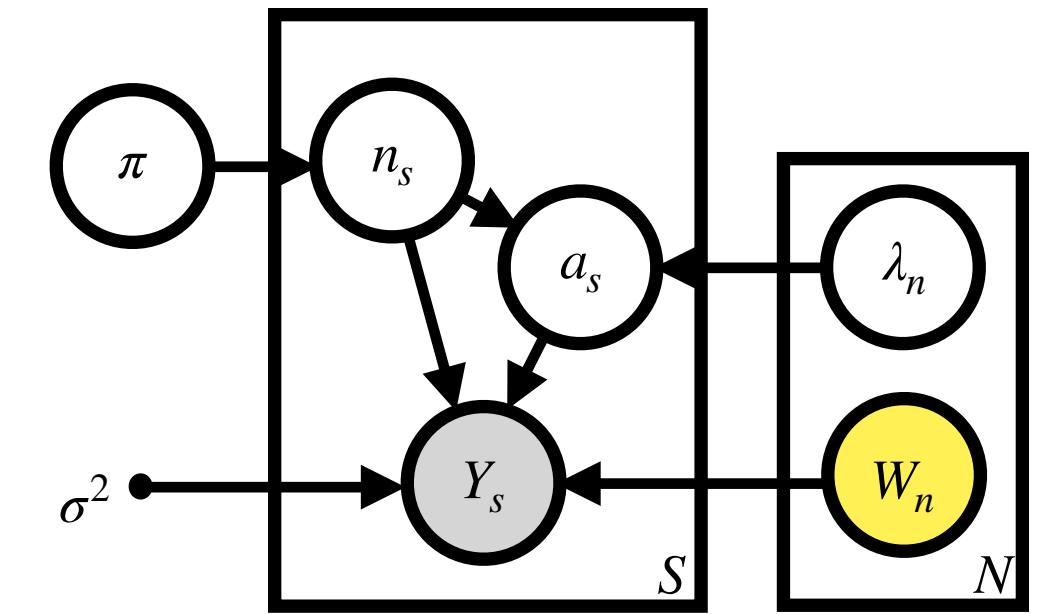
$$= \frac{1}{\sigma^2} \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot a_s \text{Tr}(Y_s^\top W_n) + c$$

$$\propto \text{Tr}(\bar{Y}_n^\top W_n) + c,$$

where

$$\bar{Y}_n = \sum_{s=1}^S \mathbb{I}[n_s = n] \cdot a_s Y_s.$$

- Recall that the templates are constrained to be unit norm and low rank; i.e. $W_n \in \mathcal{S}_K$.



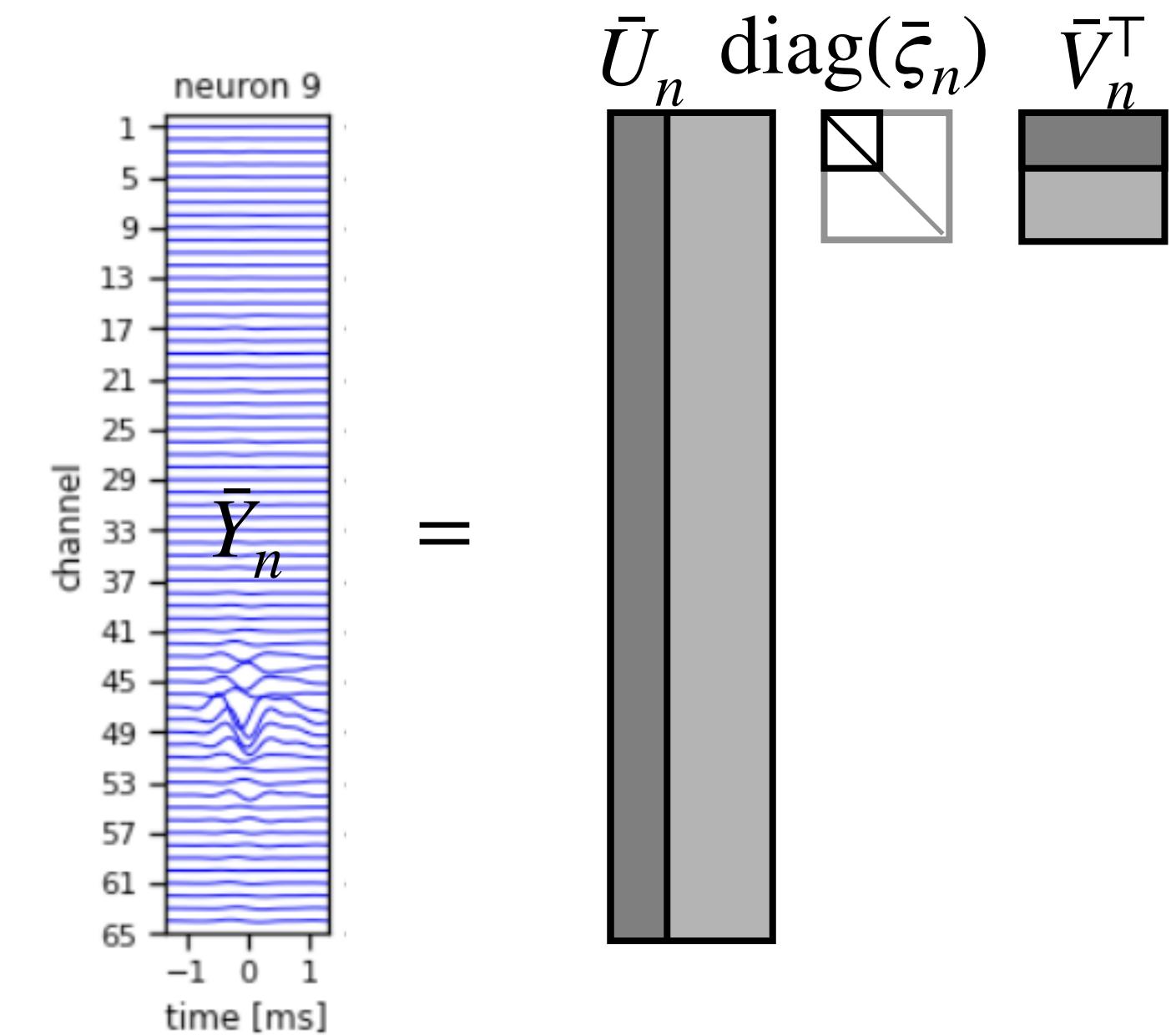
Maximum a posteriori estimation

Optimizing the templates

- The best unit-norm low rank template is found by the SVD of \bar{Y}_n . Let,

$$\bar{Y}_n = \bar{U}_n \text{diag}(\bar{\zeta}_n) \bar{V}_n^\top$$

denote the SVD.



Maximum a posteriori estimation

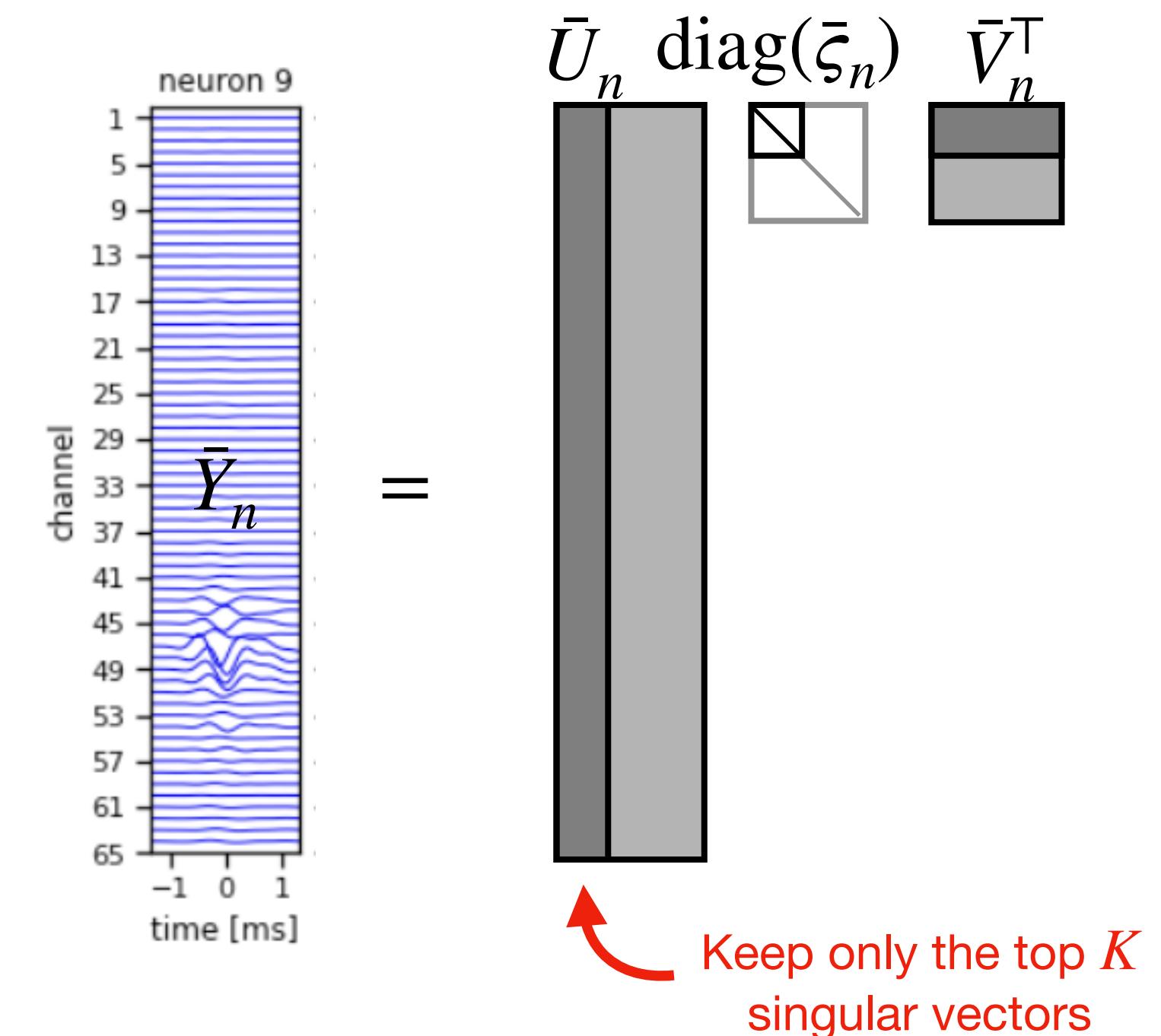
Optimizing the templates

- The best unit-norm low rank template is found by the SVD of \bar{Y}_n . Let,

$$\bar{Y}_n = \bar{U}_n \text{diag}(\bar{\zeta}_n) \bar{V}_n^\top$$

denote the SVD.

- Set $U_n = [\bar{U}_n]_{1:K}$ and $V_n = [\bar{V}_n]_{1:K}$. That is, keep only the K columns with the largest singular values.



Maximum a posteriori estimation

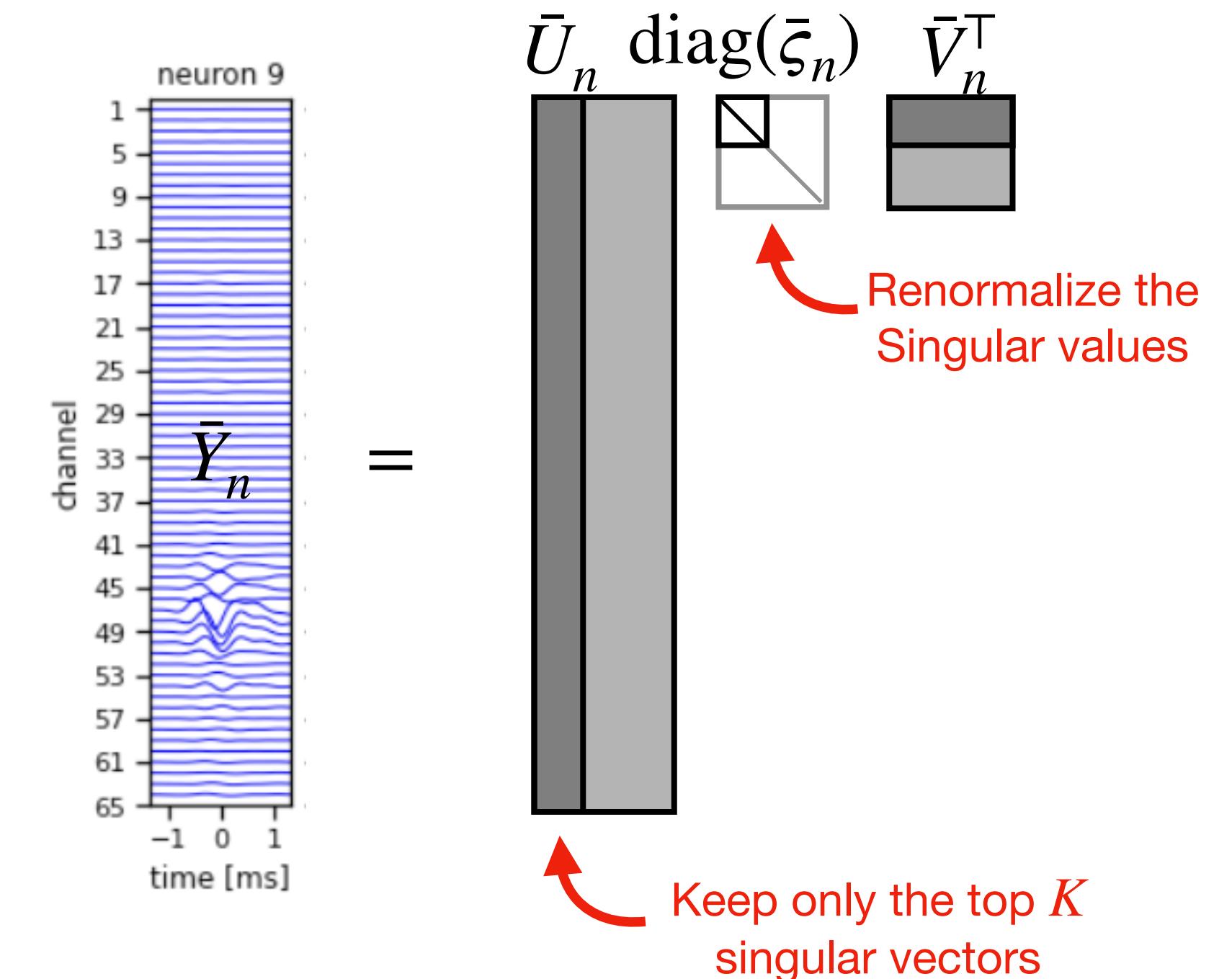
Optimizing the templates

- The best unit-norm low rank template is found by the SVD of \bar{Y}_n . Let,

$$\bar{Y}_n = \bar{U}_n \text{diag}(\bar{\zeta}_n) \bar{V}_n^\top$$

denote the SVD.

- Set $U_n = [\bar{U}_n]_{1:K}$ and $V_n = [\bar{V}_n]_{1:K}$. That is, keep only the K columns with the largest singular values.
- Set $\zeta_n = \frac{\bar{\zeta}_{n,1:K}}{\|\bar{\zeta}_{n,1:K}\|_2}$. That is, keep the top K singular values and normalize them by their L² norm.



Maximum a posteriori estimation

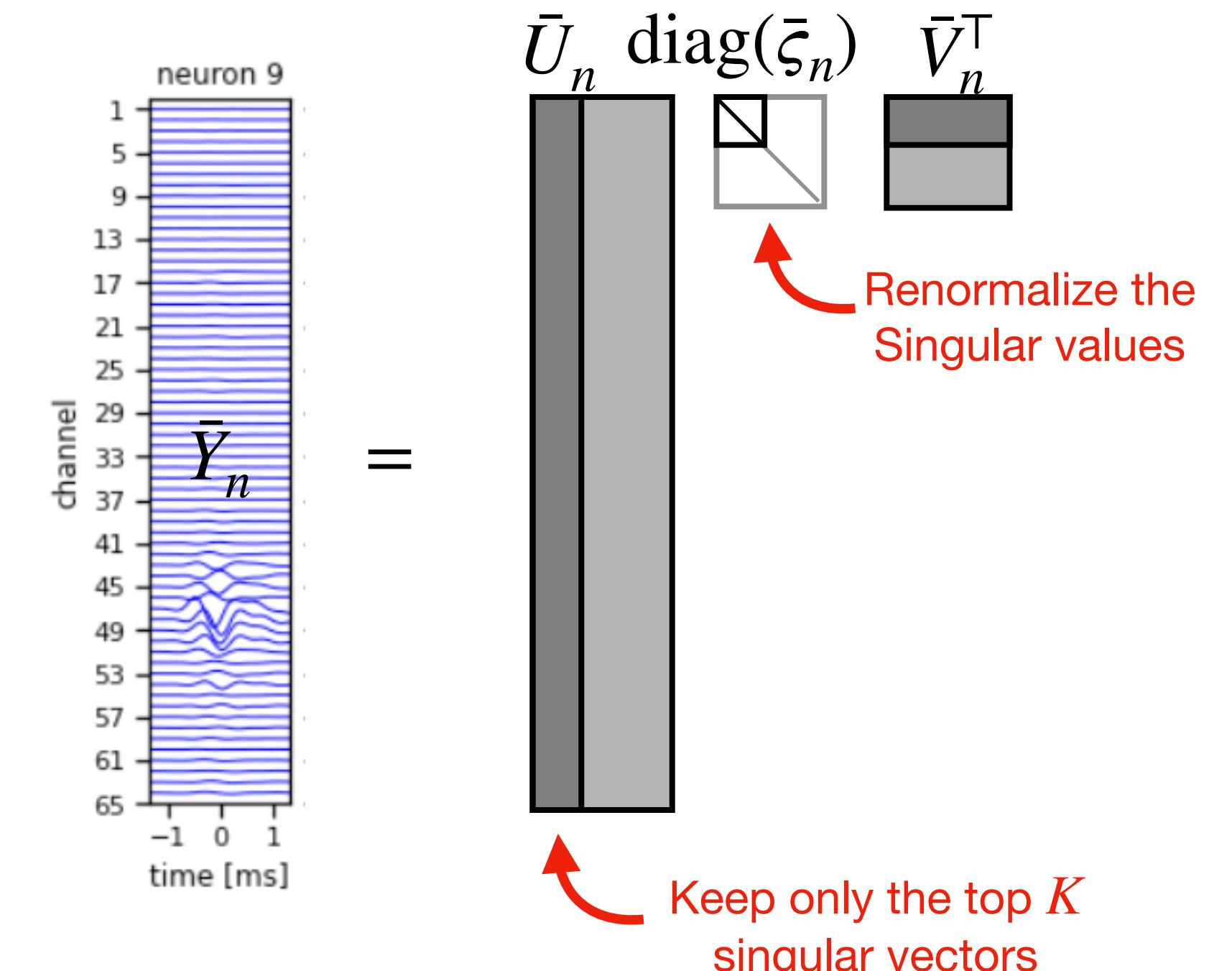
Optimizing the templates

- The best unit-norm low rank template is found by the SVD of \bar{Y}_n . Let,

$$\bar{Y}_n = \bar{U}_n \text{diag}(\bar{\zeta}_n) \bar{V}_n^\top$$

denote the SVD.

- Set $U_n = [\bar{U}_n]_{1:K}$ and $V_n = [\bar{V}_n]_{1:K}$. That is, keep only the K columns with the largest singular values.
- Set $\zeta_n = \frac{\bar{\zeta}_{n,1:K}}{\|\bar{\zeta}_{n,1:K}\|_2}$. That is, keep the top K singular values and normalize them by their L² norm.
- Set $W_n = U_n \text{diag}(\zeta_n) V_n^\top$.



Maximum a posteriori estimation

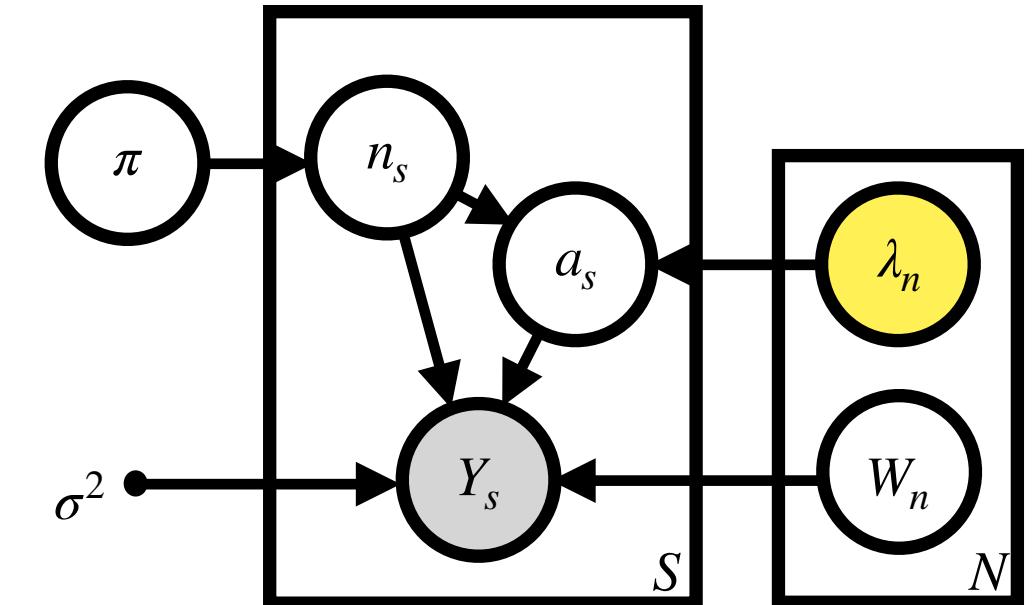
Optimizing the amplitude rates

- **Exercise:** Show that the optimal amplitude rates are

$$\lambda_n^{-1} = \frac{1}{S_n} \sum_{s=1}^S \mathbb{I}[n_s = n] a_s,$$

Where $S_n = \sum_{s=1}^S \mathbb{I}[n_s = n]$.

- Take the derivative of the log joint wrt λ_n , set to zero, and solve.



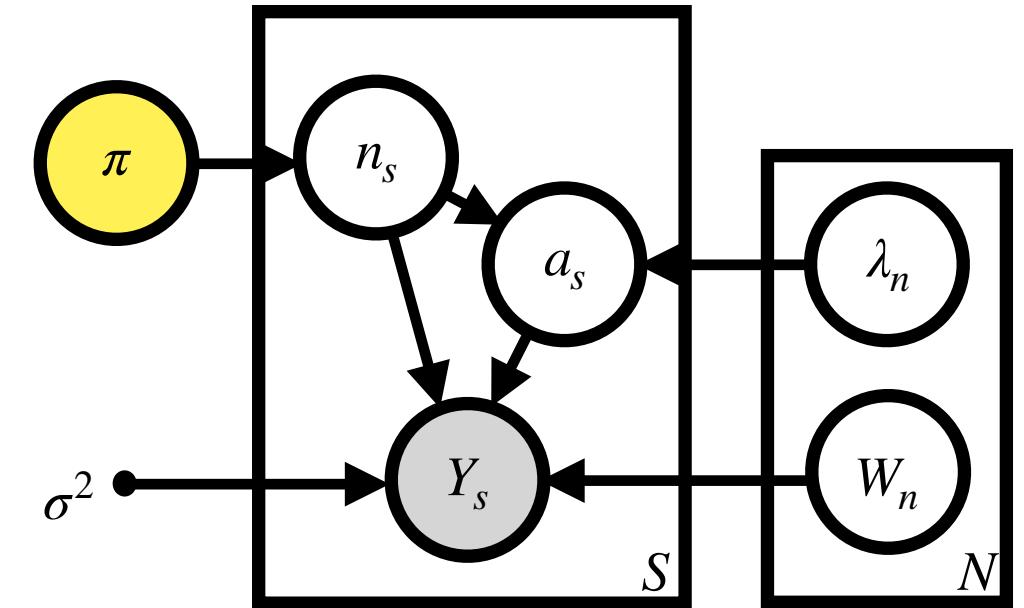
Maximum a posteriori estimation

Optimizing the amplitude rates

- **Exercise:** Show that the optimal neuron probabilities are

$$\pi_n = \frac{S_n}{\sum_{n'=1}^N S_{n'}}.$$

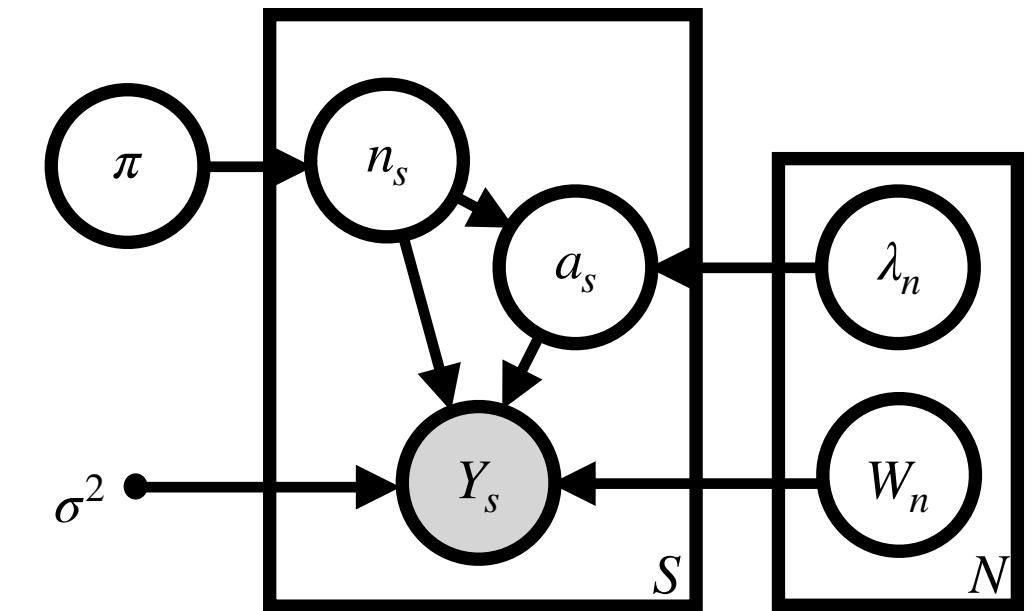
- Form the Lagrangian $\mathcal{L}(\pi, \eta) = \log p(\{n_s\}_{s=1}^S \mid \pi) - \eta \left(\sum_{n=1}^N \pi_n - 1 \right)$
- Take the derivative of the Lagrangian wrt π_n , set to zero, and solve to get an expression in terms of S_n and η .
- Take the derivative wrt η , substitute for π_n , set to zero, and solve.



Maximum a posteriori estimation

Putting it all together

- Initialize parameters W, λ, π
- Iterate until convergence:
 1. Optimize **latent variables** n_s, a_s for each spike s .
 - Solve for $a_s^\star(n_s) = \text{Tr}(Y_s^\top W_{n_s}) - \sigma^2 \lambda_{n_s}$, set $n_s = \text{argmax } \frac{1}{2}(a_s^\star(n_s))^2 + \log \pi_{n_s}$.
 2. Optimize **templates** W_n for each neuron n .
 - Form \bar{Y}_n , take SVD, keep the top singular vectors, renormalize the singular values.
 3. Optimize **amplitude rates** λ_n for each neuron n .
 - Set to weighted average of amplitudes of spikes assigned to neuron n .
 4. Optimize **neuron probabilities** π .
 - Set to empirical distribution of spike labels.



Conclusion

- High density silicon probes (e.g. **Neuropixels**) and multi-electrode arrays record **extracellular action potentials** of nearby neurons. The spike sorting problem is to **cluster** those EAPs based on their **waveforms**.
- We developed a simple **mixture model** with labels, amplitudes, and template waveforms.
- We derived a **coordinate ascent algorithm** for *maximum a posteriori* (MAP) inference.
- **Next time:** you'll implement the algorithm in lab! You'll have to do some preprocessing to filter the data and extract spikes (see extra slides), and you'll write code for the coordinate updates.

Further reading

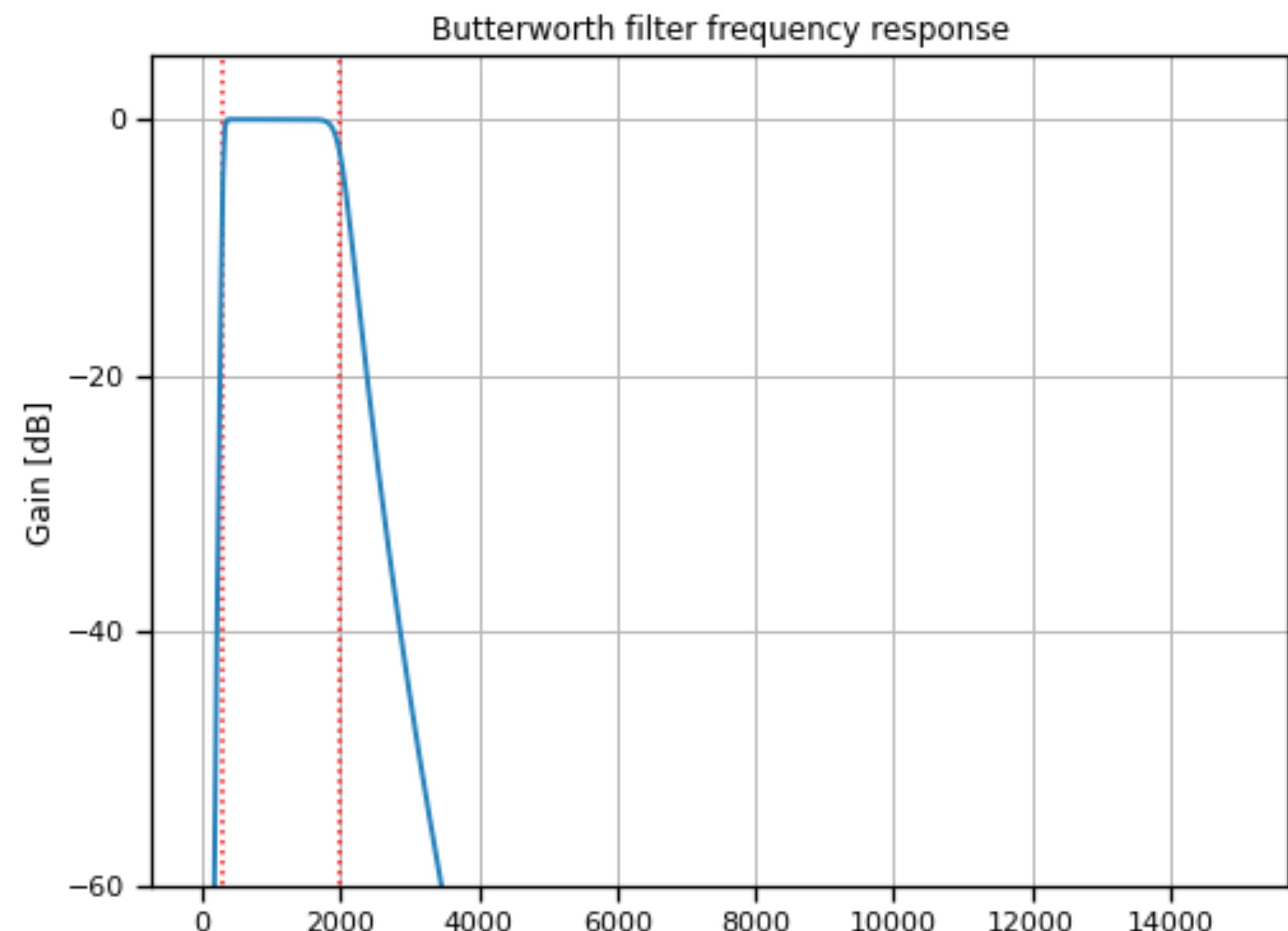
- Ch 2.1 of the course notes.
- Neuropixels:
 - Jun, James J., Nicholas A. Steinmetz, Joshua H. Siegle, Daniel J. Denman, Marius Bauza, Brian Barbarits, Albert K. Lee, et al. 2017. “Fully Integrated Silicon Probes for High-Density Recording of Neural Activity.” *Nature* 551 (7679): 232–36.
 - *NeuroPixels v1*
 - Steinmetz, Nicholas A., Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, et al. 2020. “Neuropixels 2.0: A Miniaturized High-Density Probe for Stable, Long-Term Brain Recordings.” Cold Spring Harbor Laboratory. <https://doi.org/10.1101/2020.10.27.358291>.
 - *NeuroPixels v2*
- Spike sorting:
 - Pachitariu, Marius, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Harris Kenneth D. 2016. “Kilosort: Realtime Spike-Sorting for Extracellular Electrophysiology with Hundreds of Channels.” Cold Spring Harbor Laboratory. <https://doi.org/10.1101/061481>.
 - *Kilosort is the inspiration for the model we built and will continue next week*
 - Quijan Quiroga, Rodrigo, and Stefano Panzeri. 2009. “Extracting Information from Neuronal Populations: Information Theory and Decoding Approaches.” *Nature Reviews Neuroscience* 10 (3): 173–85.
 - *Review of “classical” spike sorting methods like the one we studied today.*
- Neurobiology 101:
 - See Khan Academy and Wikipedia links in slides.
 - Gold, Carl, Darrell A. Henze, Christof Koch, and György Buzsáki. 2006. “On the Origin of the Extracellular Action Potential Waveform: A Modeling Study.” *Journal of Neurophysiology* 95 (5): 3113–28.
 - Koch, Christof. 2004. *Biophysics of Computation: Information Processing in Single Neurons* (Computational Neuroscience Series). Oxford University Press.

Preprocessing

Preprocessing

Bandpass filtering

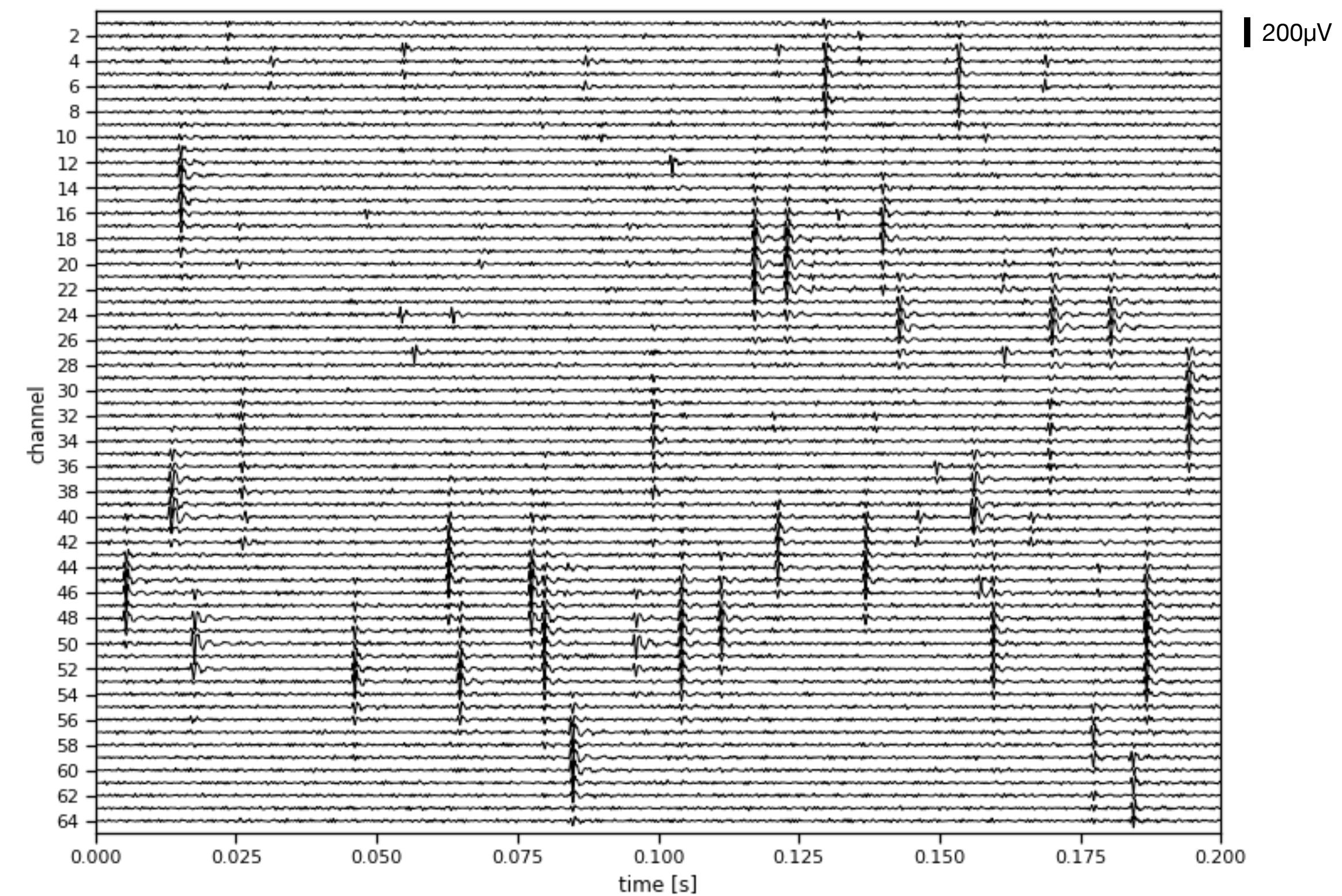
- Spikes and the resulting EAPs are only a few milliseconds long.
- The rise and fall happen take about 0.5ms.
- Real data also has slower signals like local field potentials (LFPs), which have time scales of ~3ms to 500ms.
- Since we are interested in spikes, we use a bandpass filter to focus on the [300Hz, 2000Hz] frequency content; i.e. signals varying over 0.5ms to 3ms.



Preprocessing

Bandpass filtering

- Spikes and the resulting EAPs are only a few milliseconds long.
- The rise and fall happen take about 0.5ms.
- Real data also has slower signals like local field potentials (LFPs), which have time scales of ~3ms to 500ms.
- Since we are interested in spikes, we use a bandpass filter to focus on the [300Hz, 2000Hz] frequency content; i.e. signals varying over 0.5ms to 3ms.



Preprocessing

Removing correlations between channels

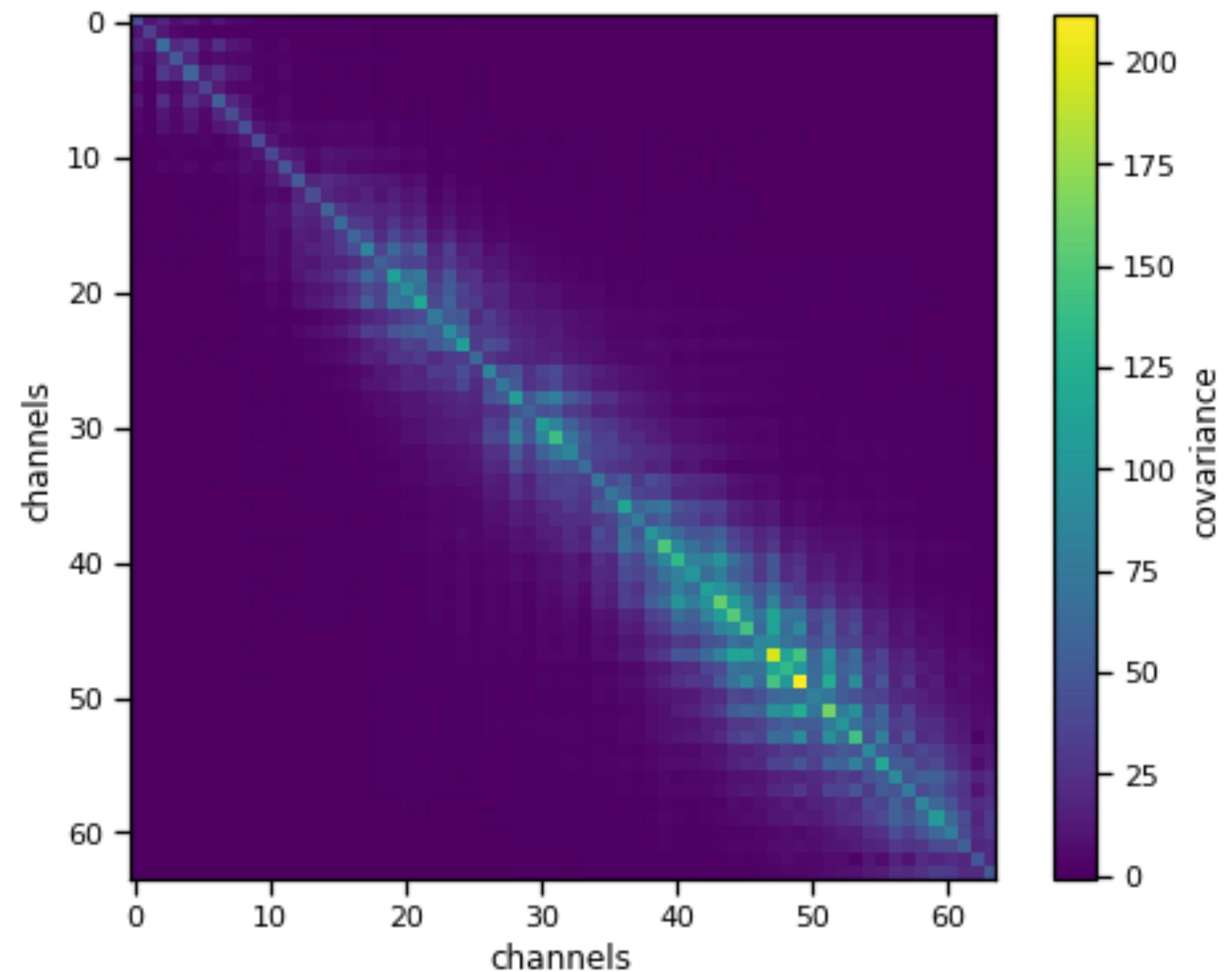
- Since the channels are so closely spaced, noise on nearby channels tends to be correlated.
- We could allow for correlated noise in the model, but it's often easier to correct for it in preprocessing.

- Let

$$\hat{\Sigma} = \frac{1}{T} \sum_{t=1}^T x_t x_t^\top$$

denote the empirical covariance matrix, where x_t is a single sample of the voltage.

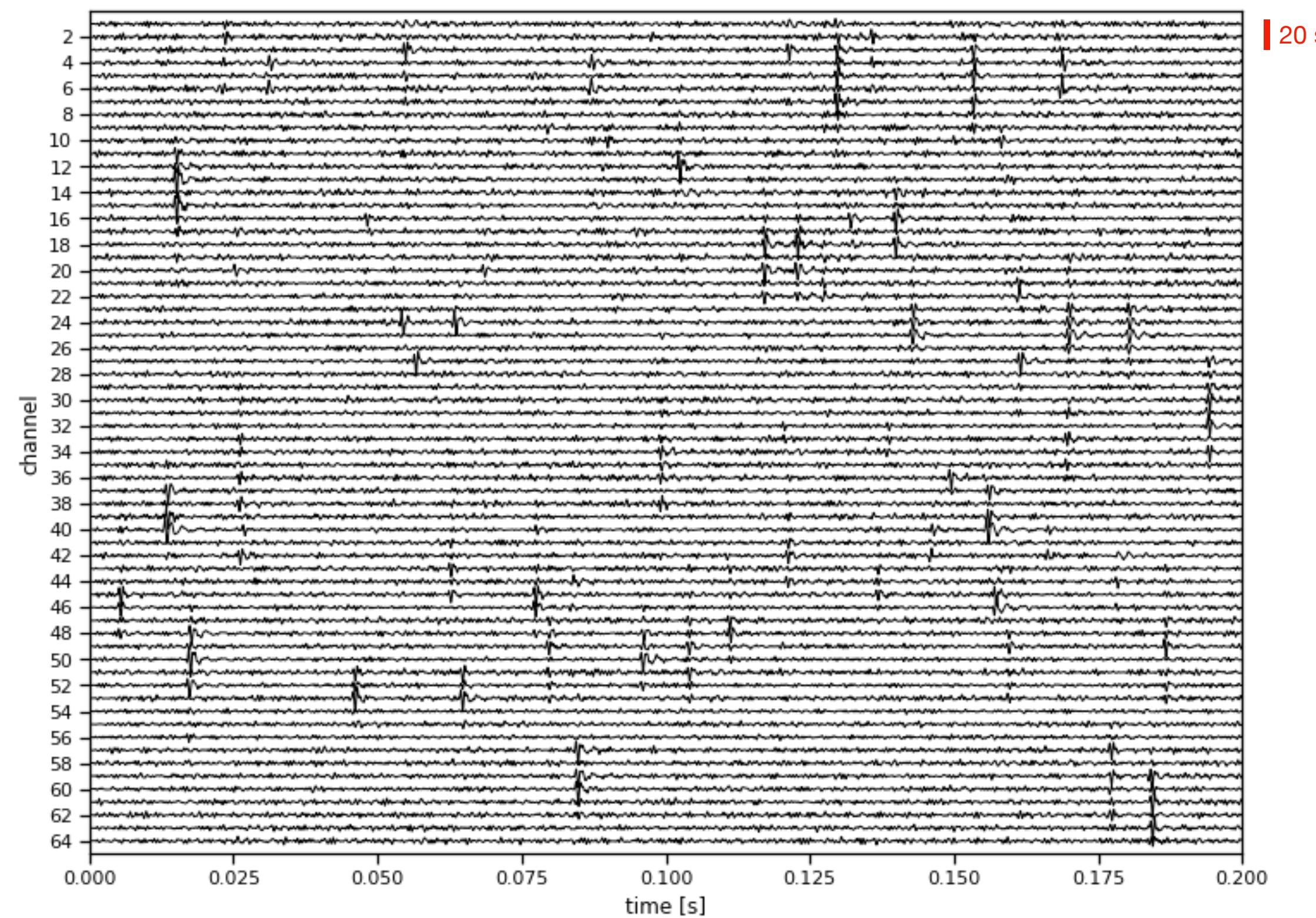
- Let $\tilde{x}_t = \hat{\Sigma}^{-1/2} x_t$ denote a transformed sample obtained by multiplying by the inverse square root of the covariance.
- The transformed samples have identity covariance:
$$\text{Cov}(\tilde{x}) = \Sigma^{-1/2} \Sigma (\Sigma^{-1/2})^\top = I.$$
- The inverse square root is easily computed via a singular value decomposition. (More on this in the lab.)



Preprocessing

Removing correlations between channels

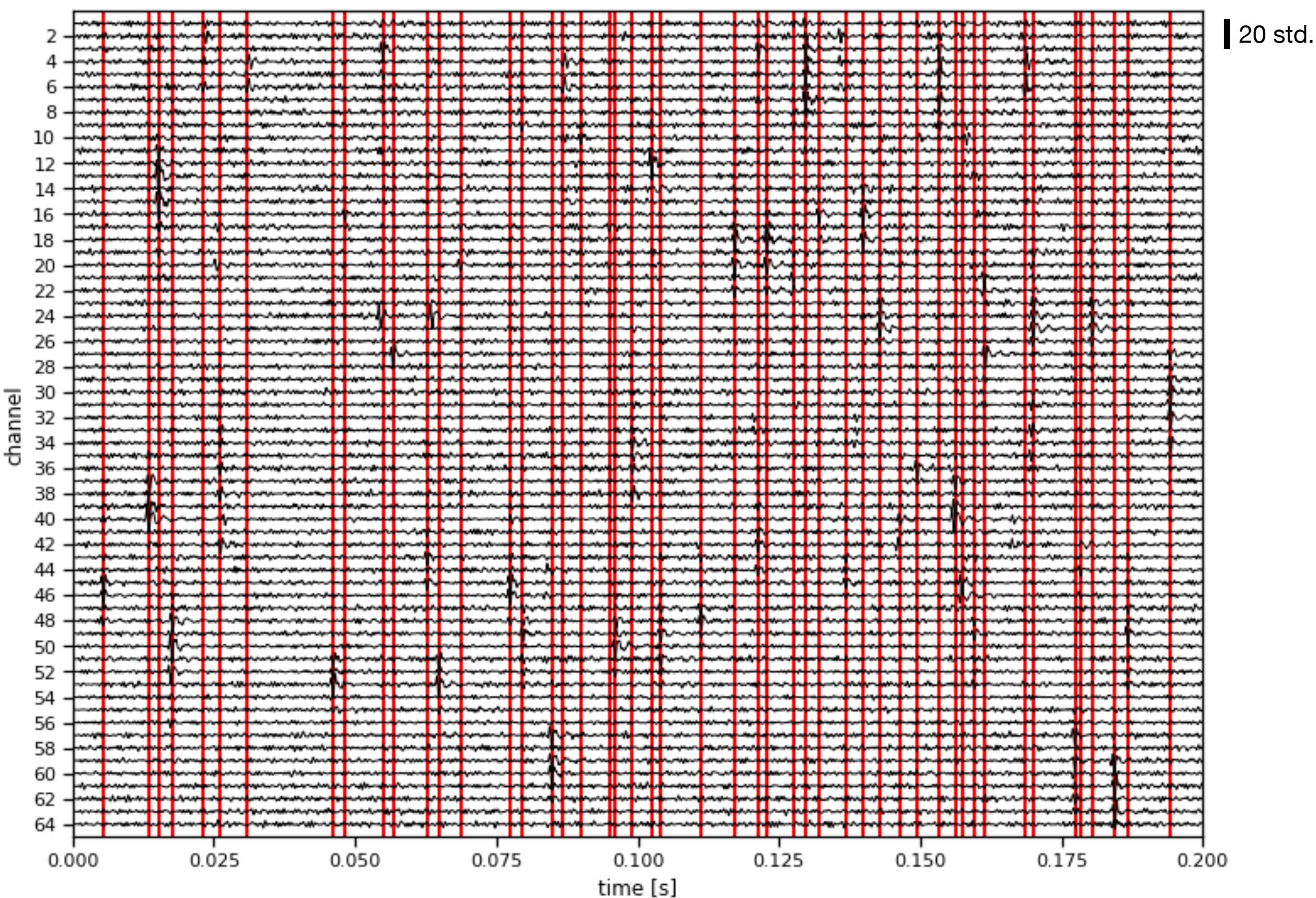
- The transformed data is now in normalized units.



Preprocessing

Finding putative spikes

- We use a simple “peak finding” algorithm to find candidate spikes.
- For example, we look for negative peaks at least 4 standard deviations in amplitude on each channel.
- Then we combine across channels to get estimates of spike times.



Preprocessing

Extract spike waveforms

- Finally, extract a short window (2-3ms) around each putative spike.
- Stack them into an array of shape (#spikes x #channels x #samples per waveform).

