Machine Learning Methods for Neural Data Analysis

Lecture 10: Decoding neural spike trains

Scott Linderman

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

Announcements

- **Lab 5 clarifications:**
  - Include $\log y_{nt}!$ in your Poisson negative log likelihood calculation.
    - Is there an easy way to implement the log likelihood? Recall Lab 2.
- **Office hours:**
  - Jaime's OH moved to 1-3pm Thursday, **this week only.**
  - Mine are 1-2:30pm today.

Final project

- **Friday, February 19**: Initial proposal (~0.5-1pg).
  - Groups (1-3 people)
  - Topic / Type of project
  - Data (suggestions forthcoming)
- **Friday, March 5**: Final proposal (~1-2pg)
  - With some preliminary results (summary plots, etc.)
- **Friday, March 12**: Work on labs in class
- **Monday, March 15—Fri March 19**: In class presentations
- **Friday March 19**: Final report due (~5pg) + Colab notebook

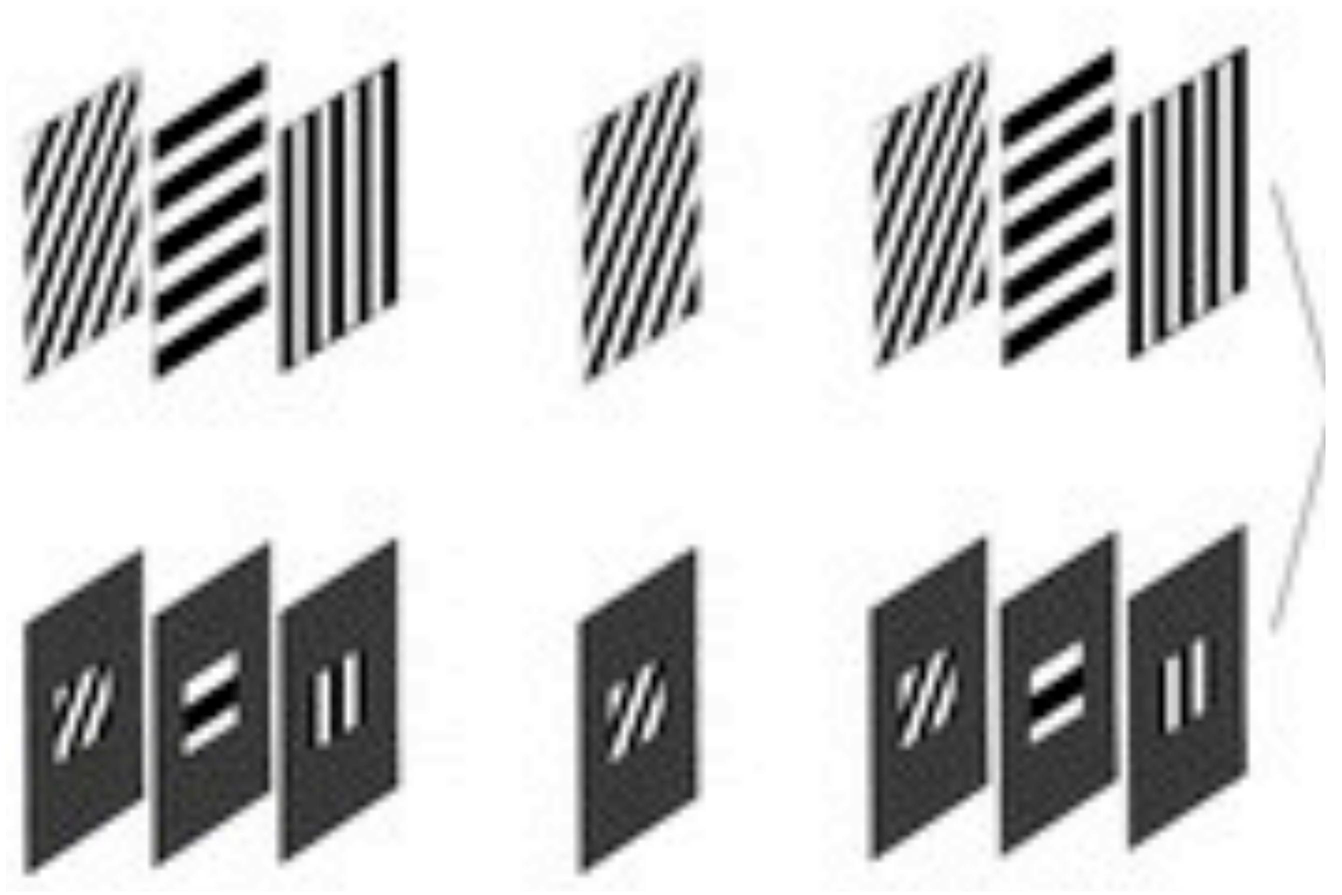
Agenda

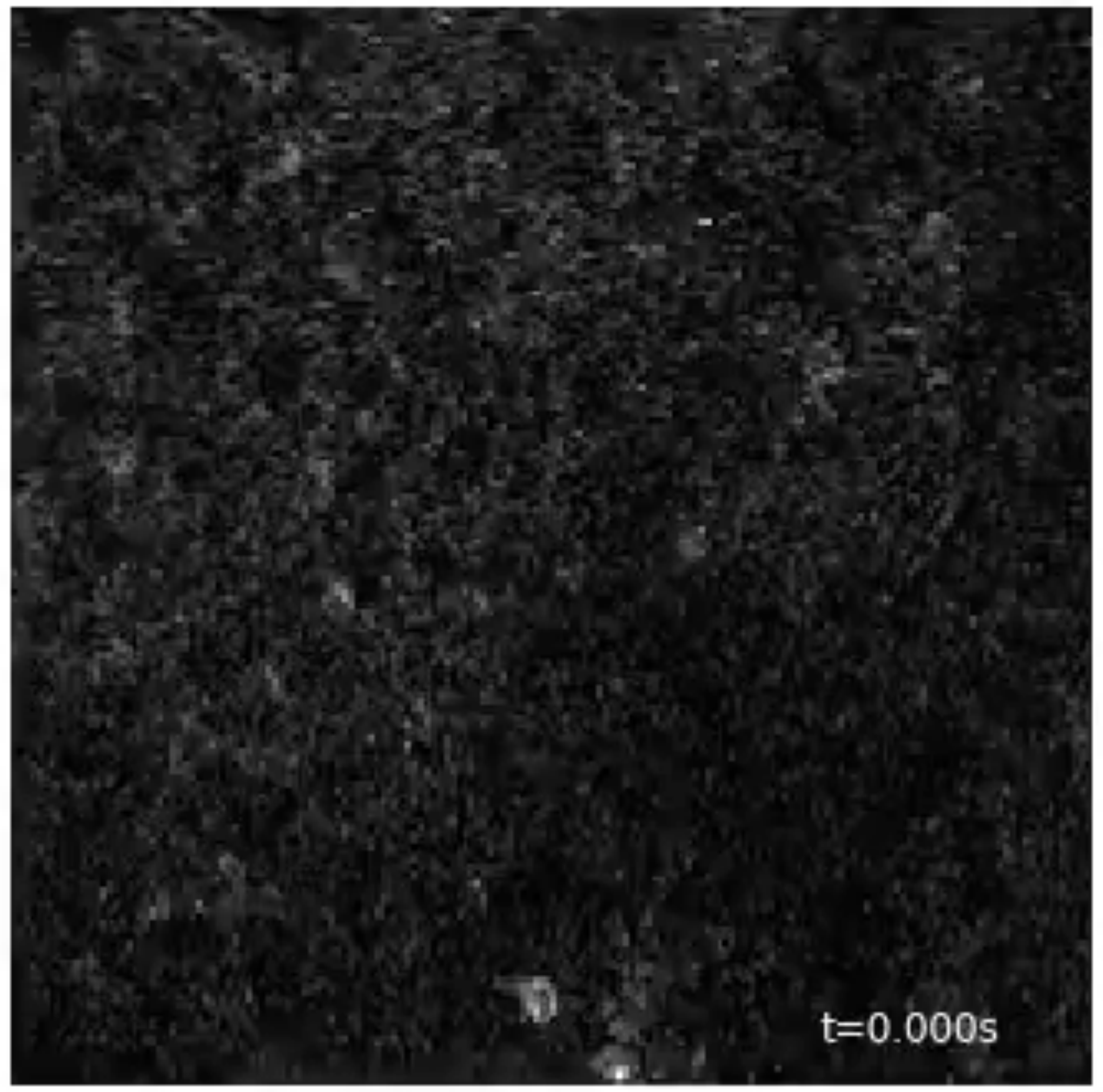
Decoding neural spike trains

- Bayesian decoders
  - A straw man model, just for illustration
  - An aside on the multivariate Gaussian distribution
  - Improving upon the basic model
- “Direct” decoders and structured prediction

Big picture

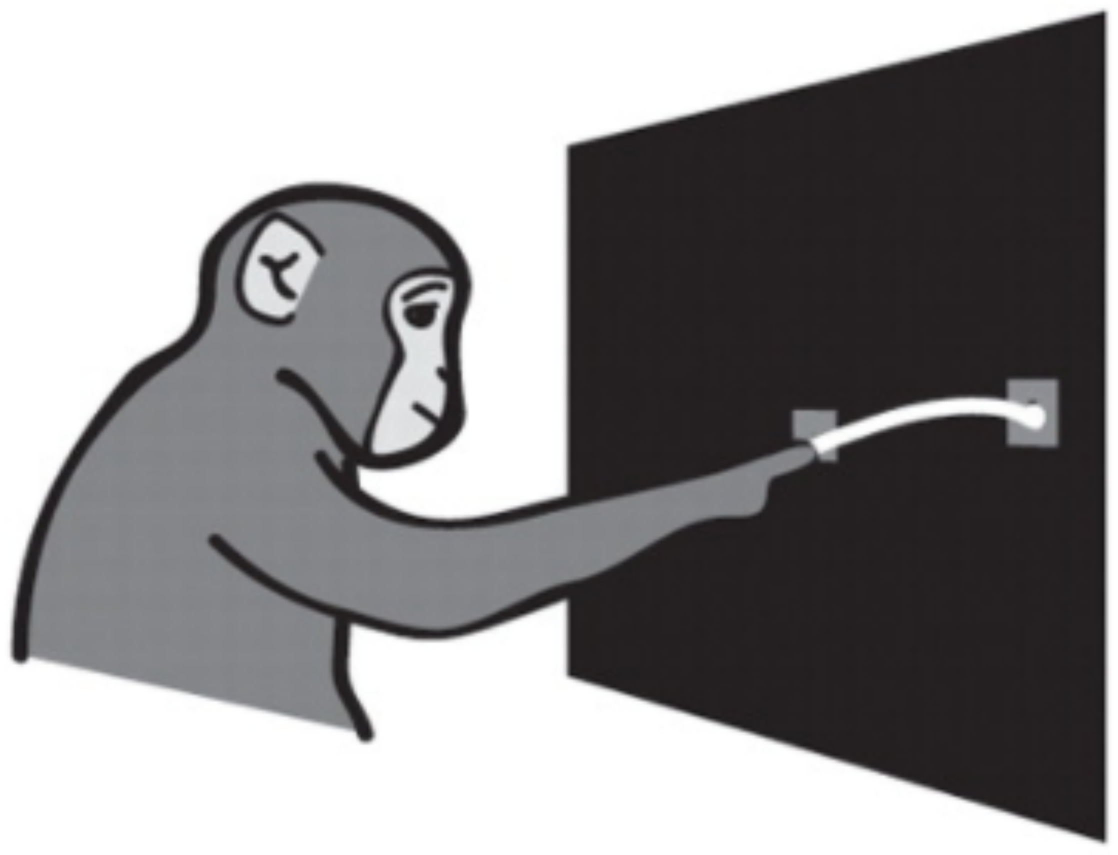
Stimulus —*encoding*→ Neural Recording —*decoding*→ Behavioral Video



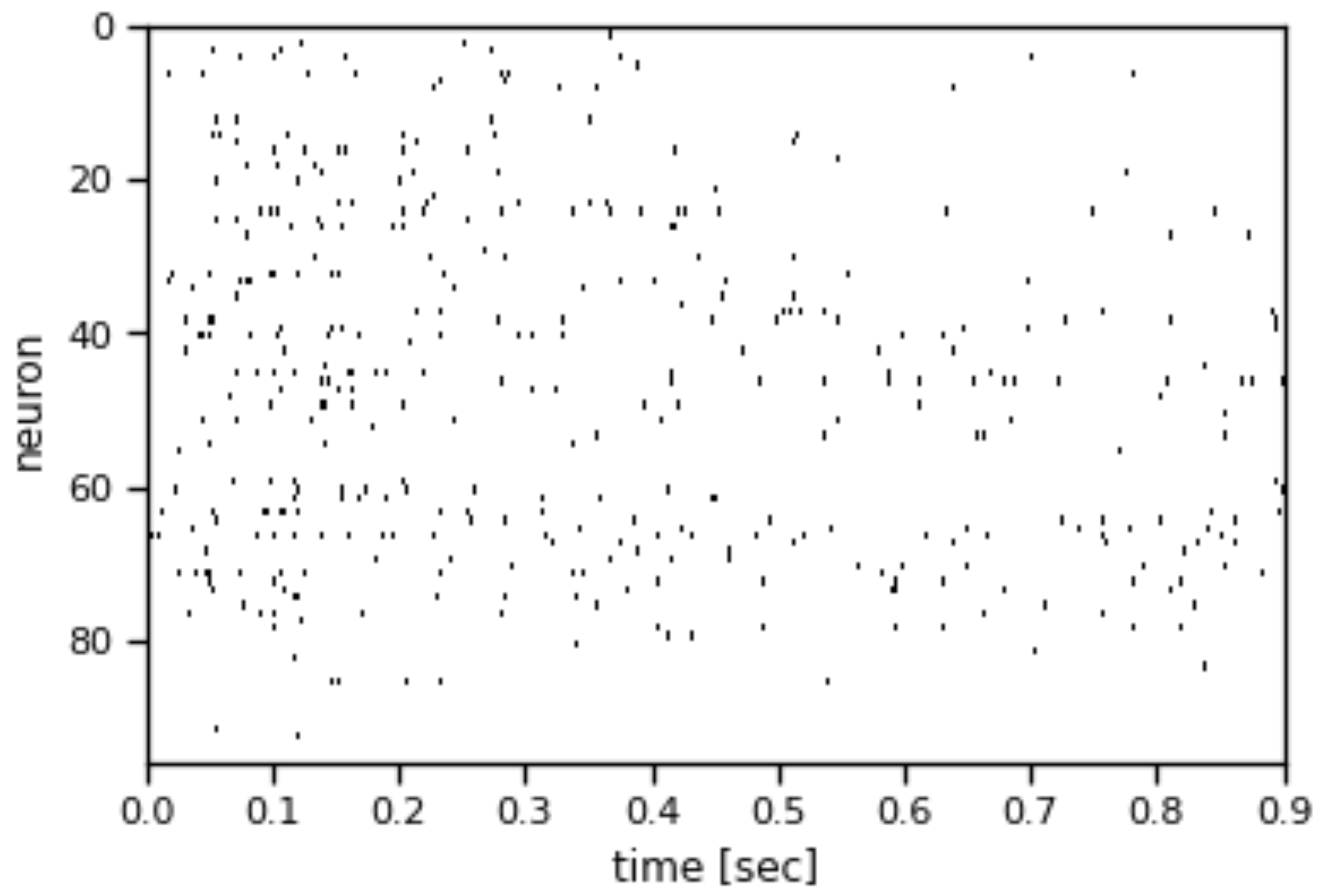


In statistics lingo, it's all regression.

Decoding movement from recordings in motor cortex

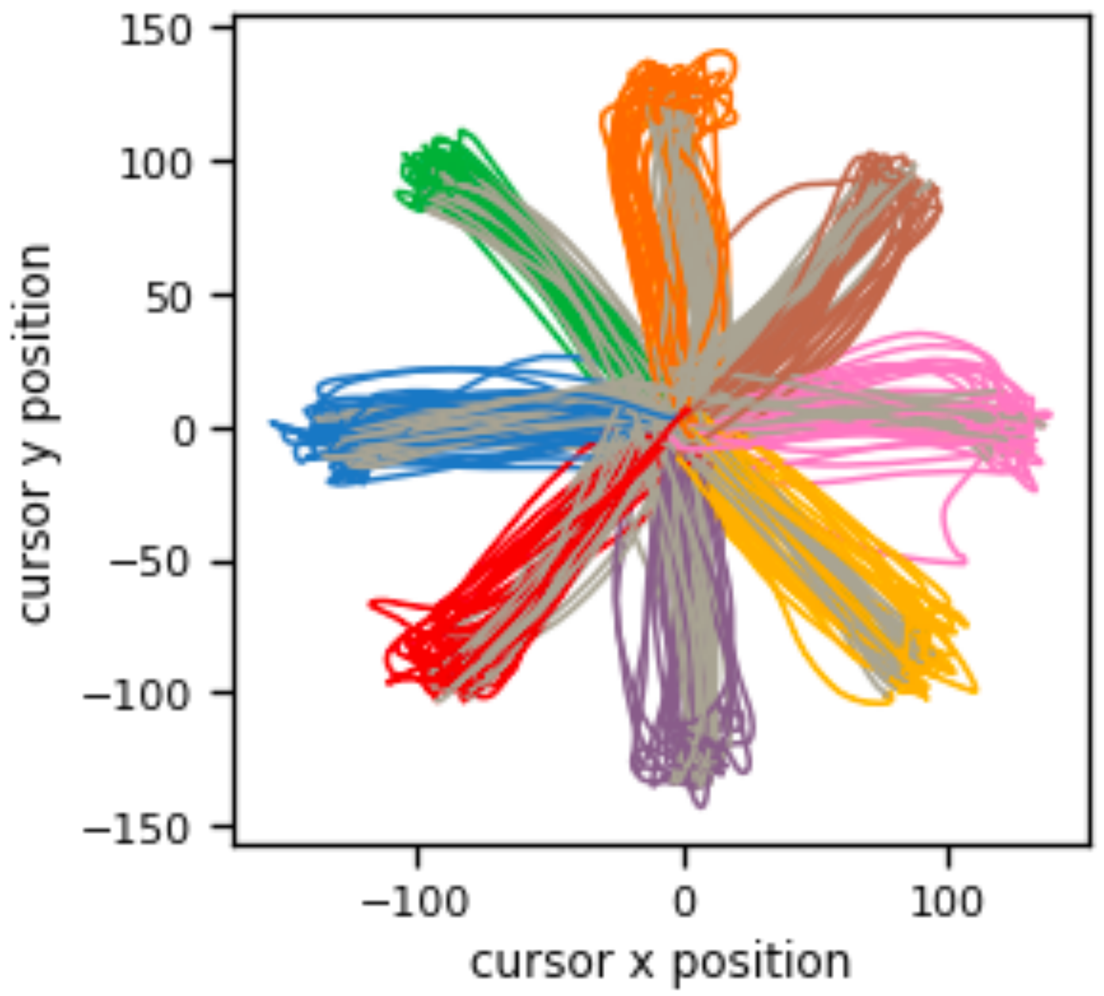


Y



$\longrightarrow$

X

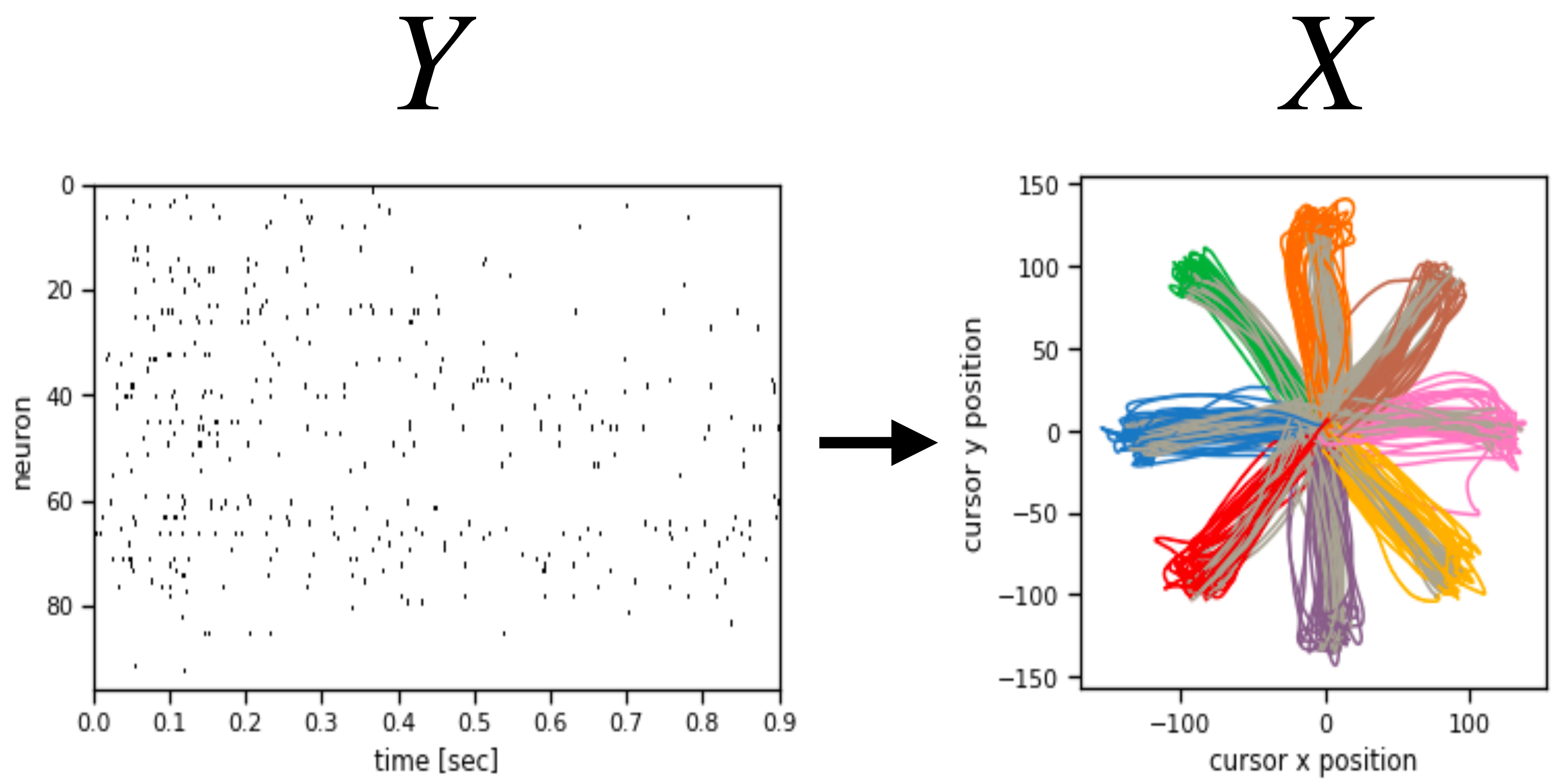


GOAL: estimate $p(X \mid Y)$

Shenoy Lab (Stanford)

Decoding movement from neural spike trains

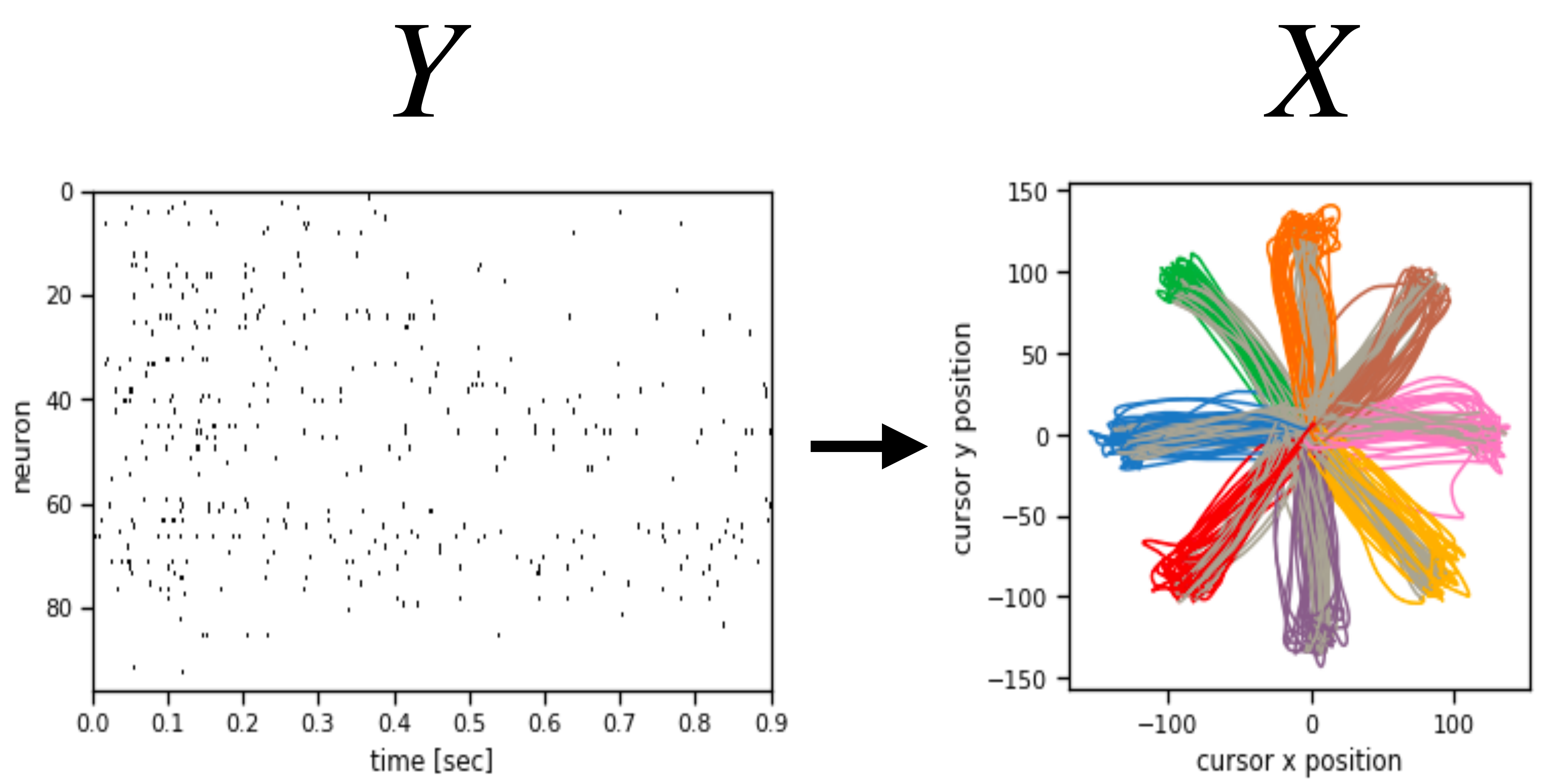
Some ideas



Decoding movement from neural spike trains

Some ideas

- It's just a regression problem… let's use the same techniques (GLMs, CNNs, etc) that we used for encoders.
    - I'll call these "direct" decoders, and we'll return to this idea in the second half of lecture.
- First, suppose we know something about the prior distribution of movement, $p(X)$. E.g. current position and velocity determine next position.
- Moreover, suppose we know something about what the neurons encode. E.g. suppose the neurons encode current velocity.
- Can we use that knowledge to inform our decoder?



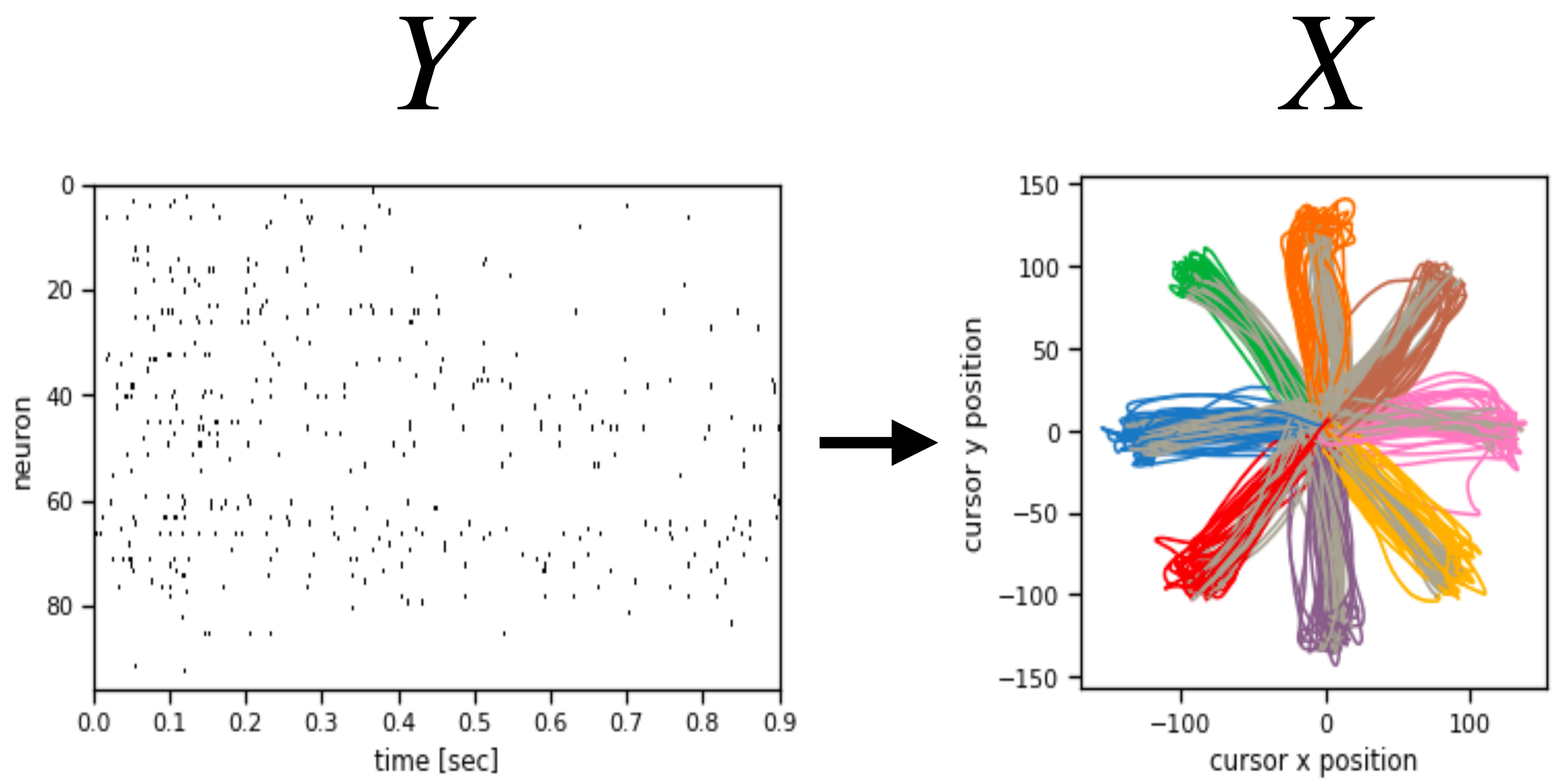
Decoding movement from neural spike trains

Bayesian decoders

- Bayes' Rule tells us how to combine a **prior** $p(X)$and a **likelihood** $p(Y \mid X)$ to obtain a **posterior**,

$$p(X \mid Y) = \frac{p(Y \mid X)p(X)}{p(Y)}$$
$$\propto p(Y \mid X)p(X)$$

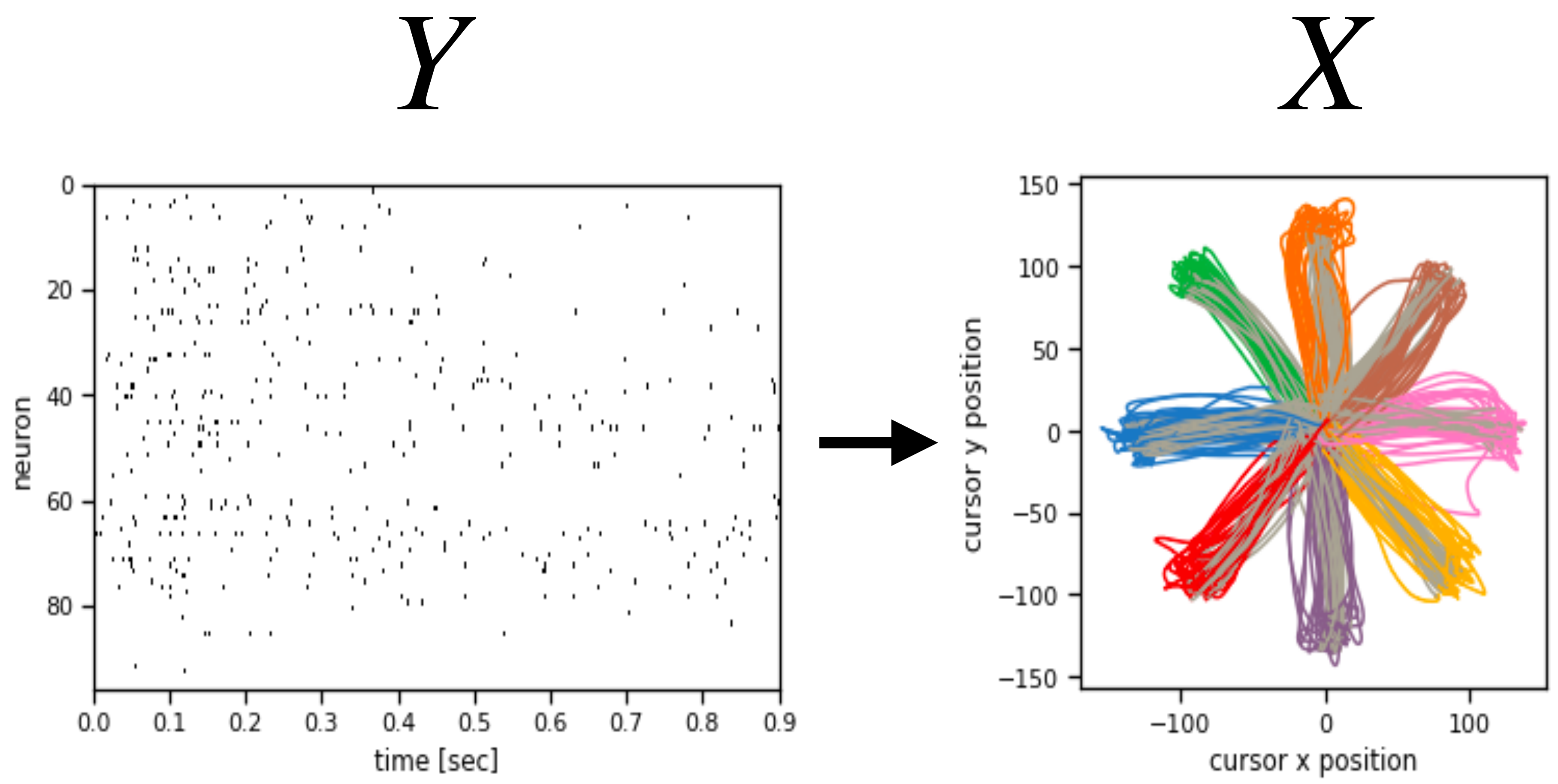
- Here, the likelihood is the **encoder** and the posterior is the **decoder**.



Decoding movement from neural spike trains

A very simple model

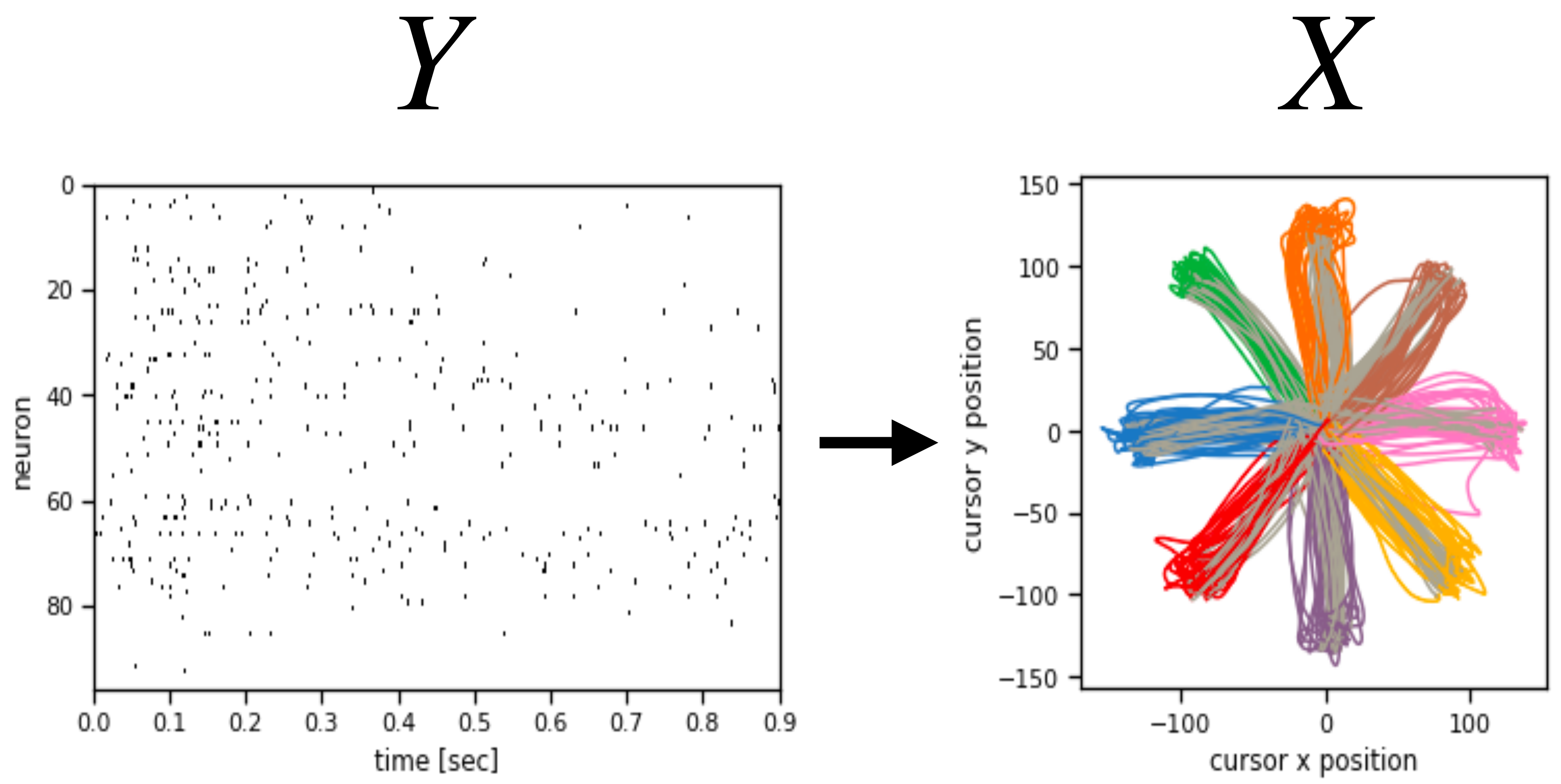
- Let $y_t \in \mathbb{N}^N$ denote the spike counts of N neurons at time t.
- Let $x_t \in \mathbb{R}^2$ denote the position of the cursor at time t.



Decoding movement from neural spike trains

A simple example

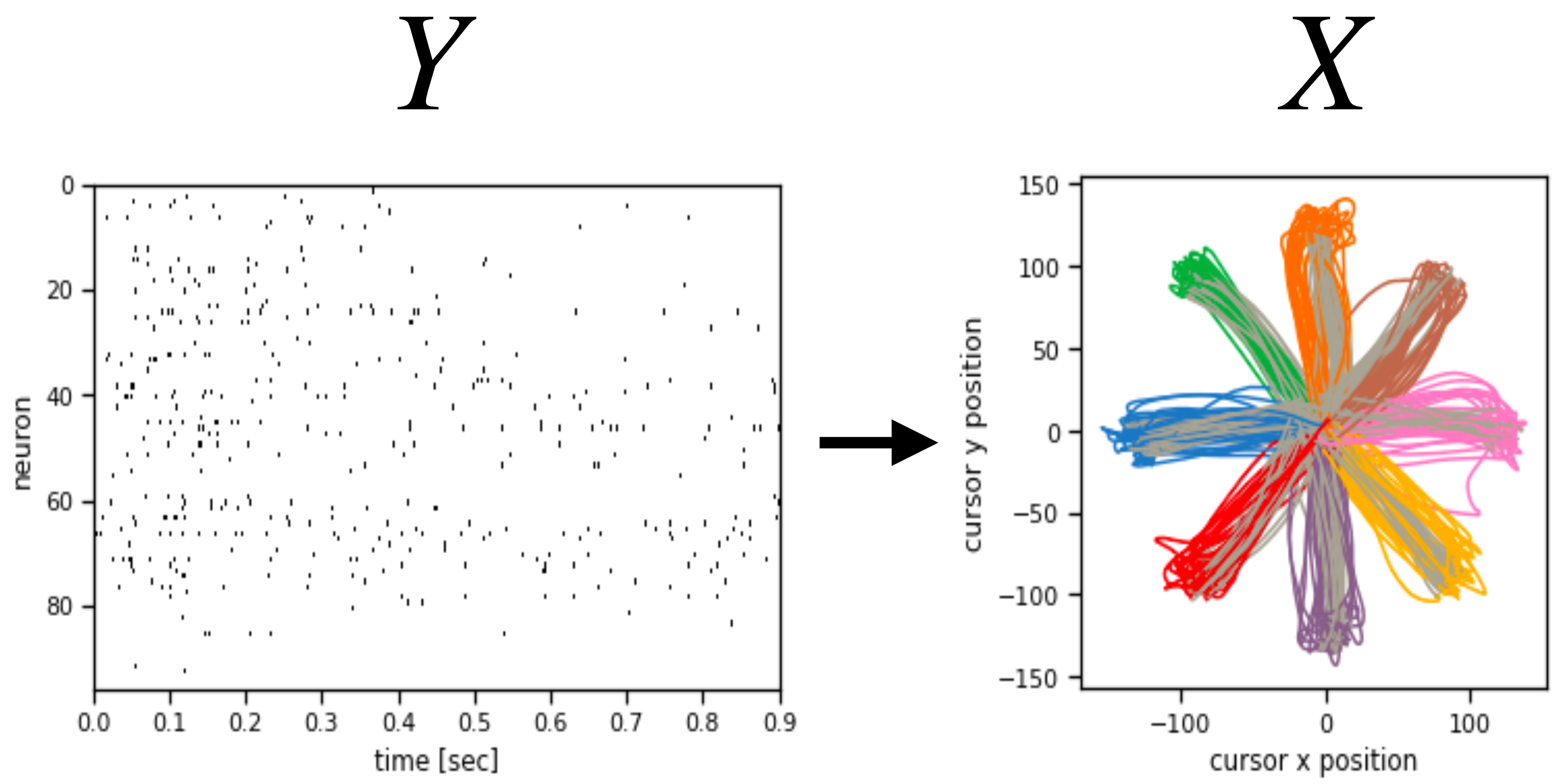
Consider the following likelihood (i.e. encoder)…



Decoding movement from neural spike trains

A simple example

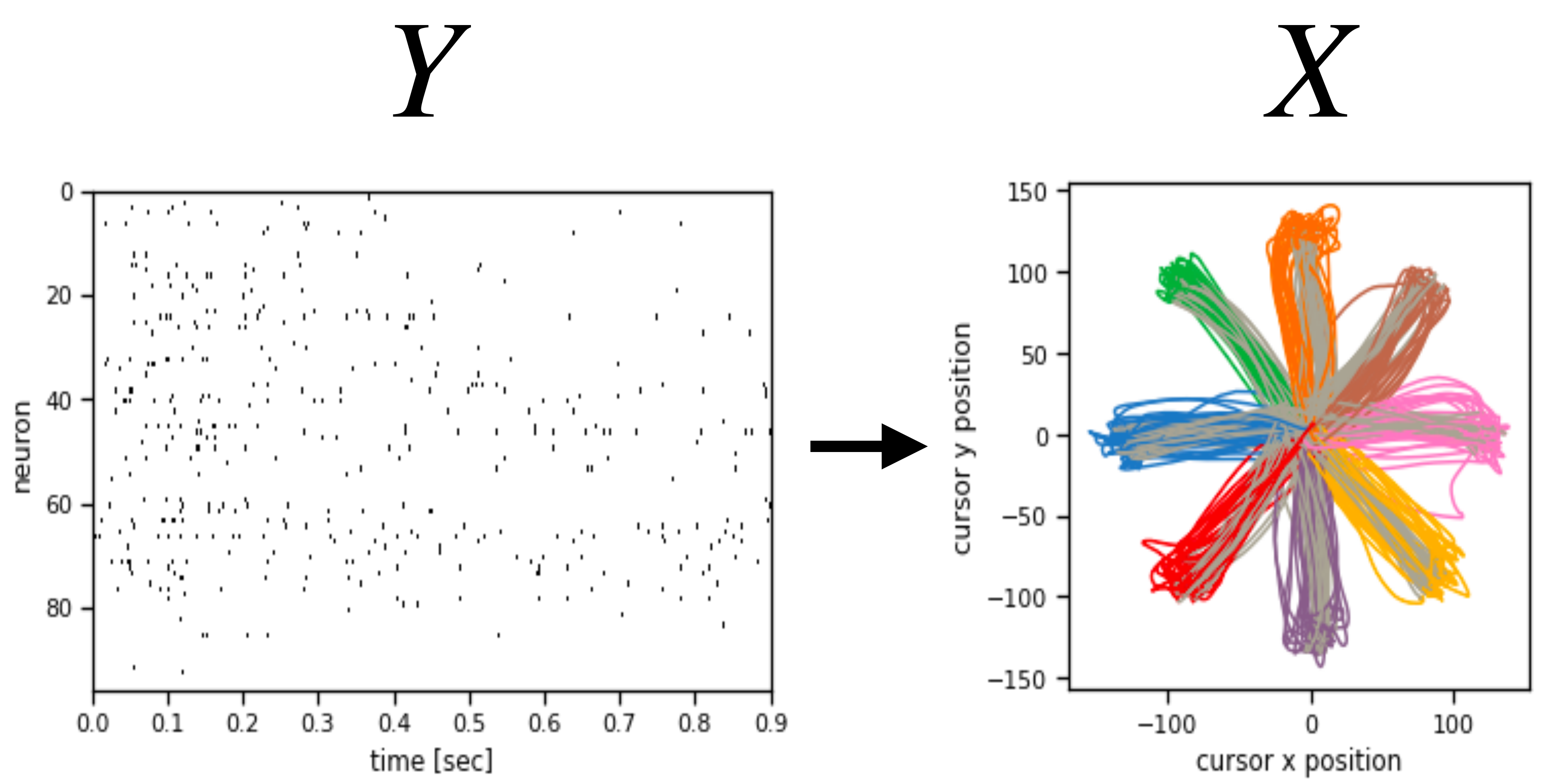
- Consider the following prior…



Decoding movement from neural spike trains

A simple example

Question: What's wrong with this model?

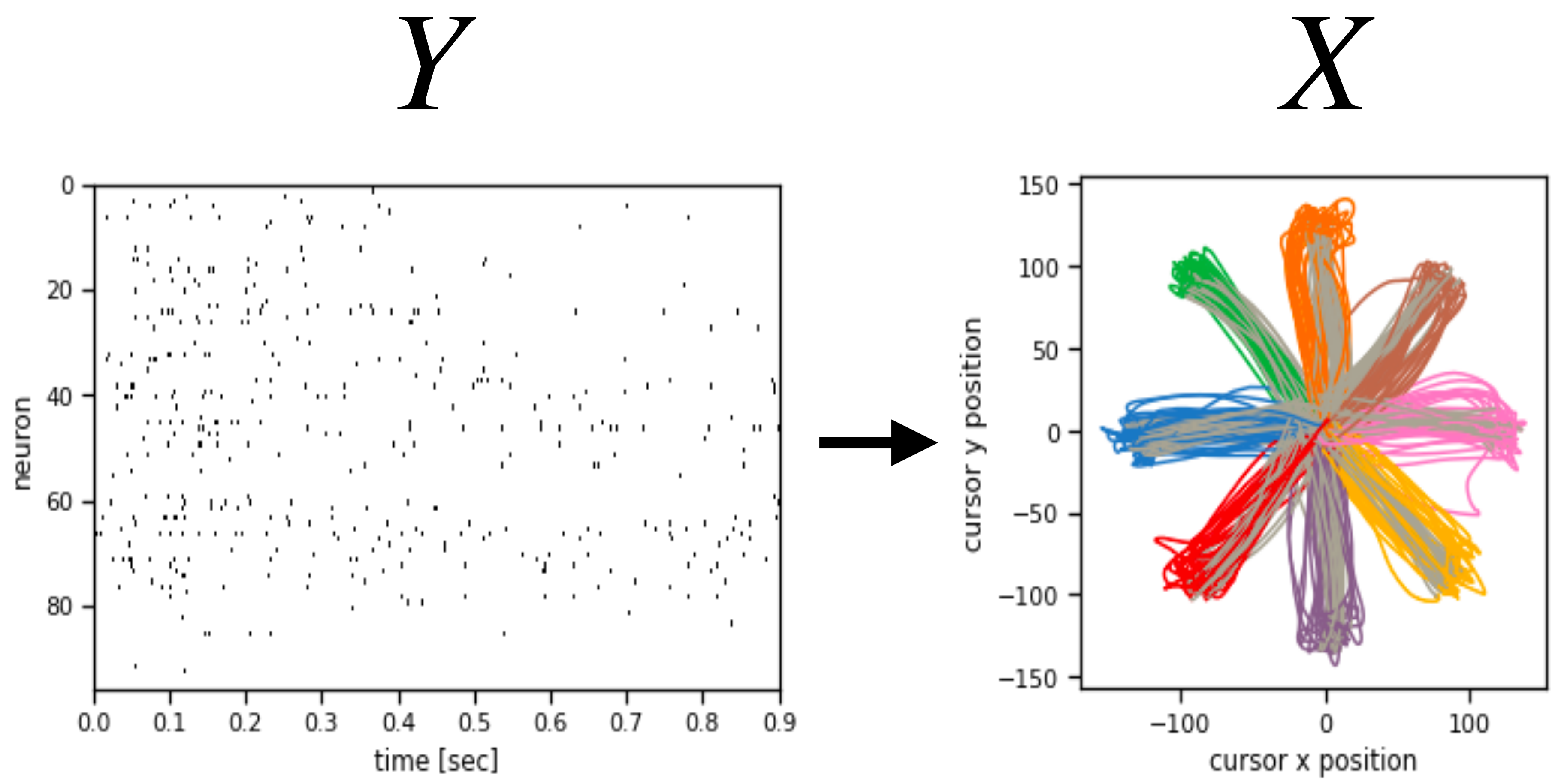


Decoding movement from neural spike trains

Deriving the posterior (decoder)

The one good thing about this model is it’s easy to work with!

Derive the posterior…



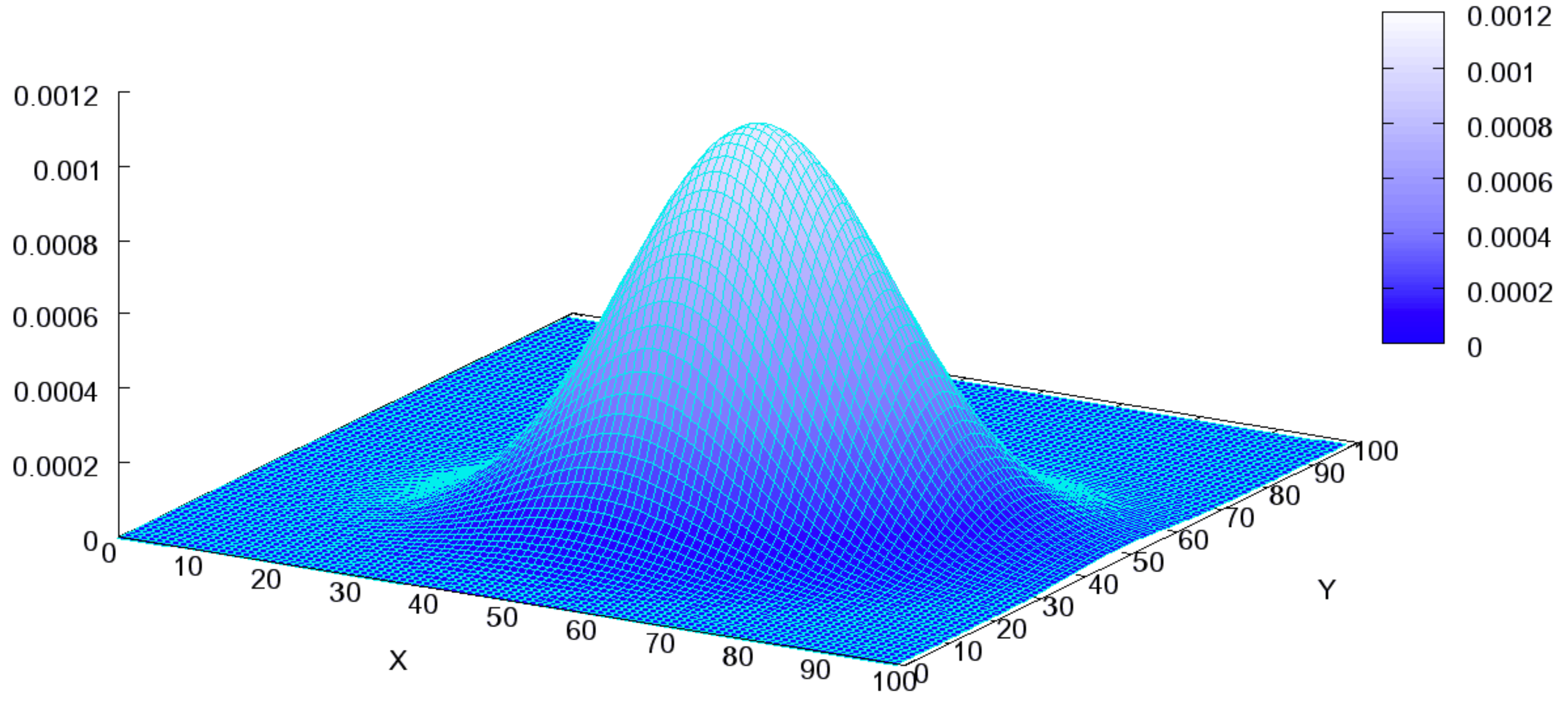
Aside: the multivariate Gaussian distribution

The multivariate Gaussian distribution

- Start with the standard normal distribution,
    - $z_d \sim \mathcal{N}(0,1) \iff p(z_d) = (2\pi)^{-1/2} \exp\left\{-\frac{z_d^2}{2}\right\}$
- Let $z = (z_1, \ldots, z_D)$ denote a vector of iid standard normal r.v.'s. Then,

$$
\begin{aligned}
p(z) &= \prod_{d=1}^{D} p(z_d) \\
&= \prod_{d=1}^{D} (2\pi)^{-1/2} \exp\left\{-\frac{z_d^2}{2}\right\} \\
&= (2\pi)^{-D/2} \exp\left\{-\frac{1}{2} z^\top z\right\}
\end{aligned}
$$

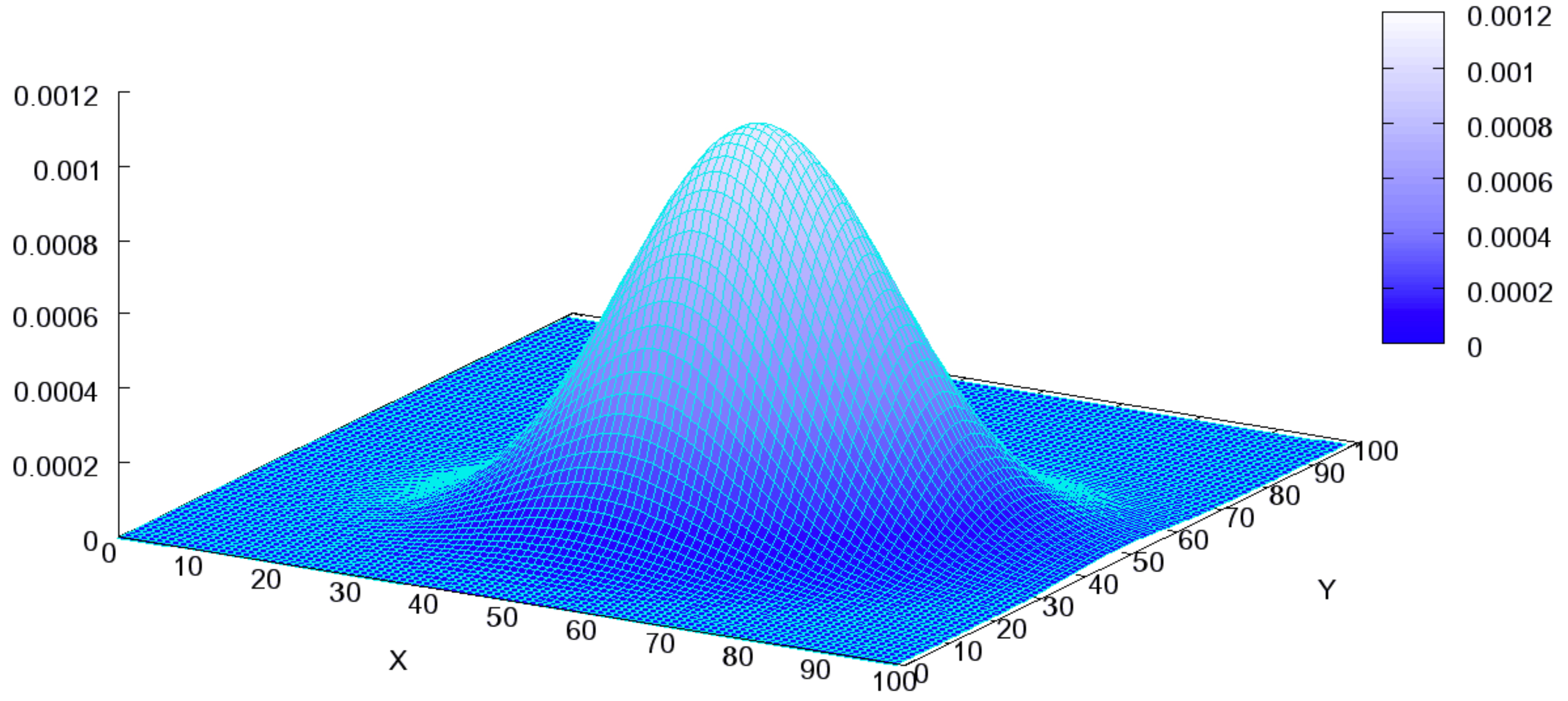
- We say $z \sim \mathcal{N}(0, I)$, a **multivariate normal distribution** with mean 0 and covariance I.



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

Aside: the multivariate Gaussian distribution

- Now let $x = \mu + \Sigma^{1/2} z$ for $\mu \in \mathbb{R}^D$ and (invertible) $\Sigma^{1/2} \in \mathbb{R}^{D \times D}$.
- Derive $p(x)$…

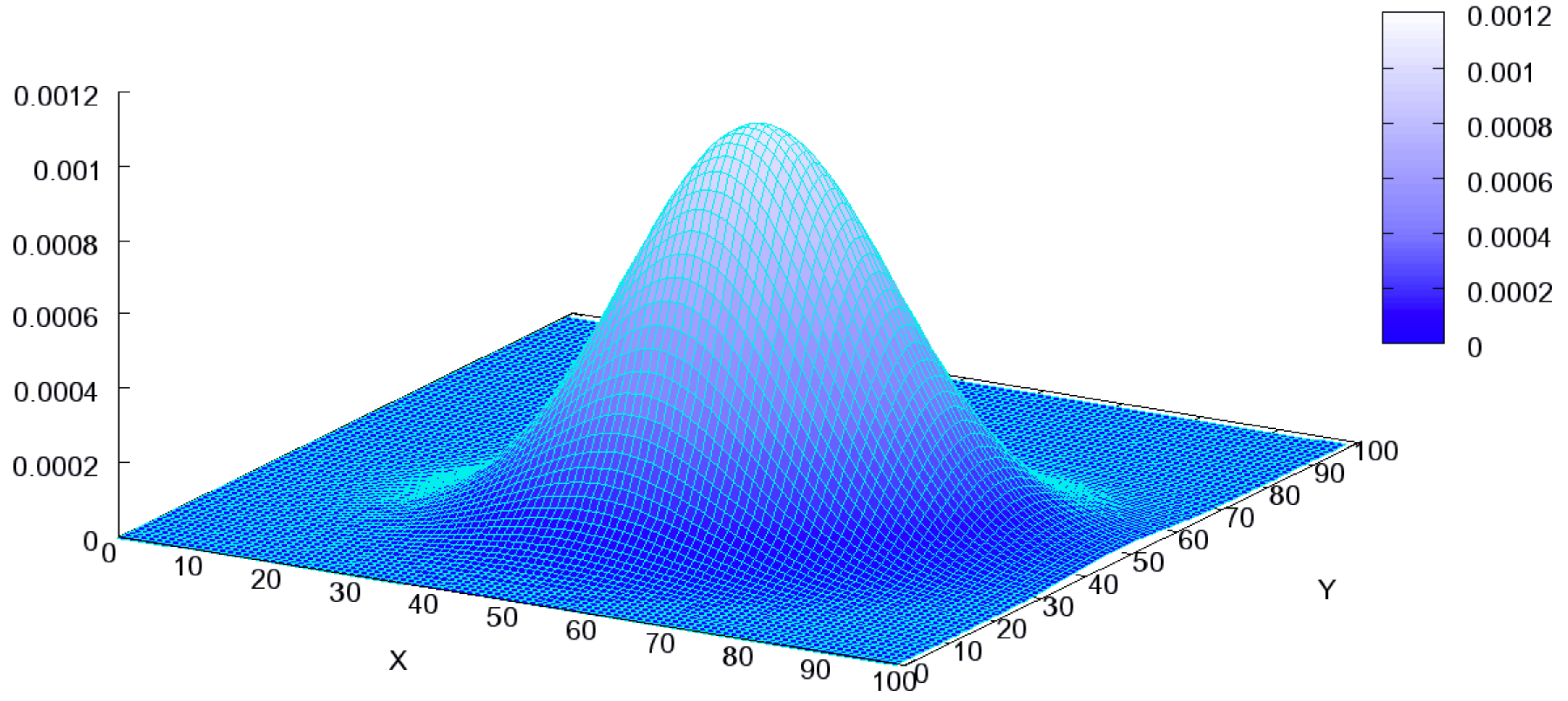


https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Aside: the multivariate Gaussian distribution

“Information” form / natural parameters

$$p(x) = (2\pi)^{-D/2} \exp\left\{ -\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu) \right\}$$



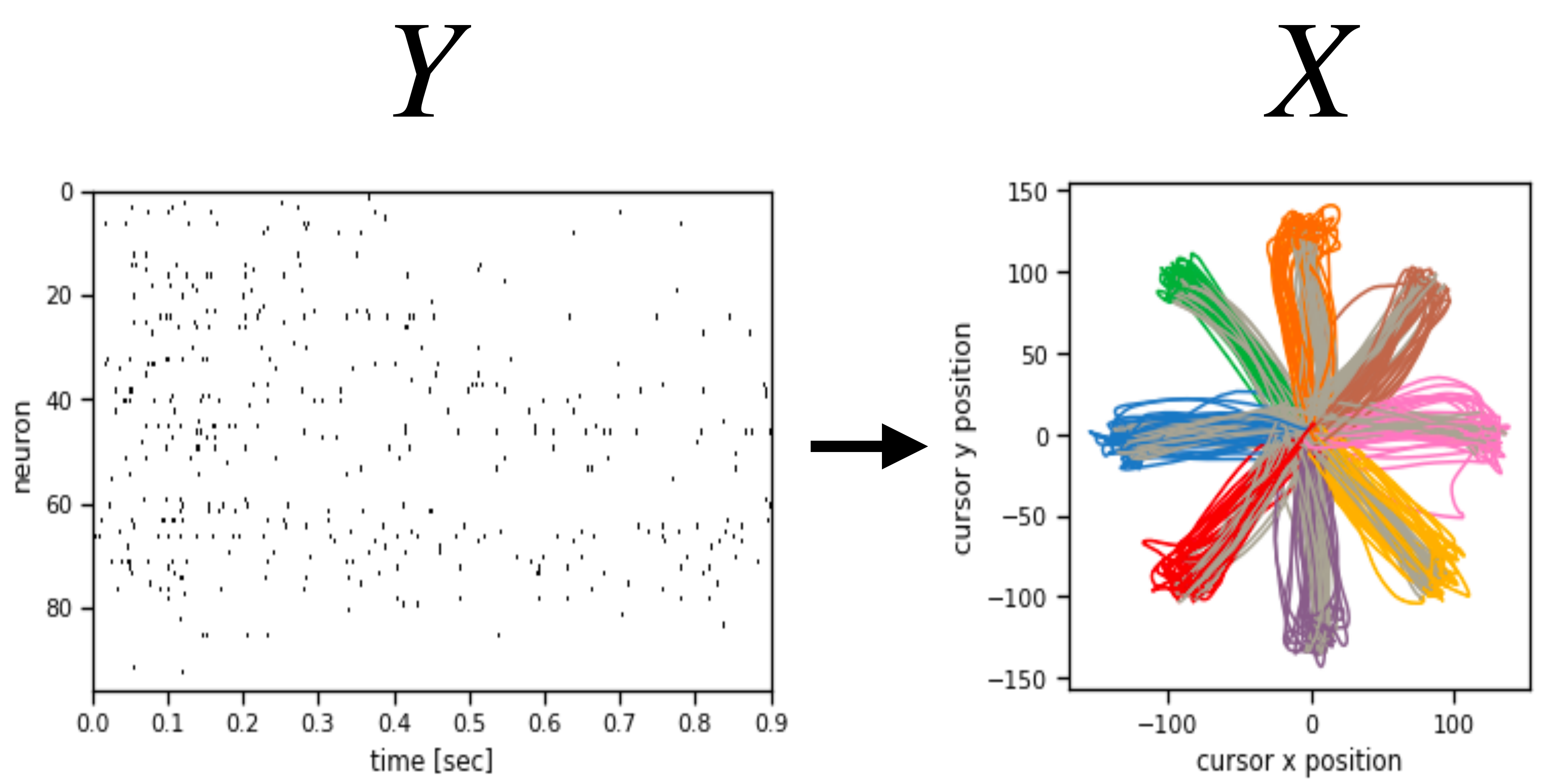
https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Decoding movement from neural spike trains

Deriving the posterior (decoder)

$$p(X \mid Y) \propto \prod_{t=1}^{T} \left[p(y_t \mid x_t)\, p(x_t) \right]$$

$$= \prod_{t=1}^{T} \left[\prod_{n=1}^{N} \mathcal{N}(y_{tn} \mid c_n^\top x_t + d_n, r_n^2)\, \mathcal{N}(x_t \mid 0, Q) \right]$$



Improving upon the basic model

Decoding movement from neural spike trains

A linear dynamical system (LDS) model

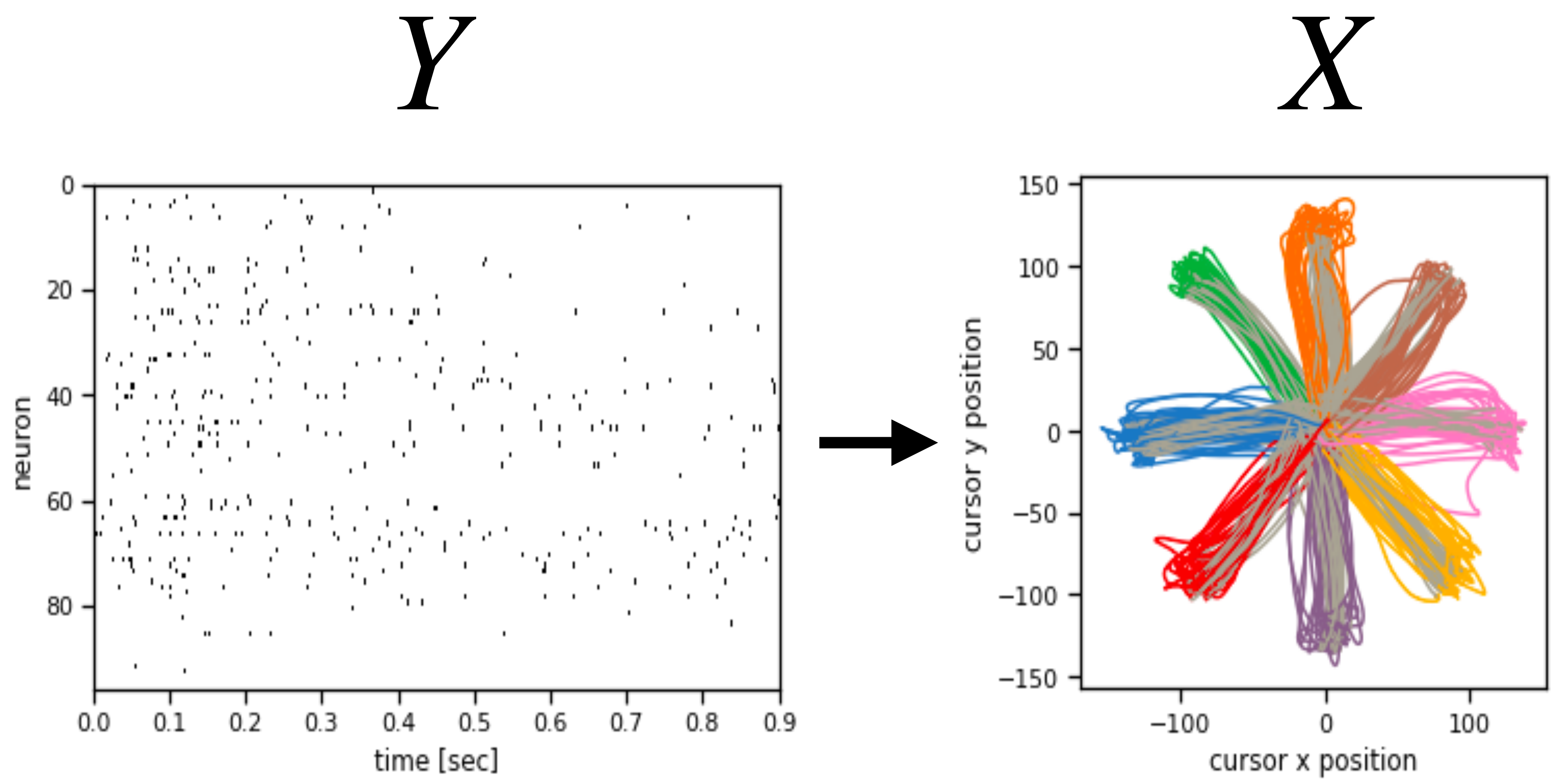
- One of the problems with the basic model is that it treated each time bin as independent.

- Instead, consider the following prior

$$p(X) = p(x_1)\prod_{t=2}^{T} p(x_t \mid x_{t-1})$$

$$= \mathcal{N}(x_1 \mid 0, Q)\prod_{t=2}^{T} \mathcal{N}(x_t \mid Ax_{t-1}, Q)$$

- Parameterized by **dynamics matrix** $A \in \mathbb{R}^{D \times D}$.



Decoding movement from neural spike trains

Derive the posterior under the LDS

$$p(X \mid Y) \propto \left[\mathcal{N}(x_1 \mid 0, Q) \prod_{t=2}^{T} \mathcal{N}(x_t \mid A x_{t-1}, Q) \right] \left[\prod_{t=1}^{T} \mathcal{N}(y_t \mid C x_t + d, R) \right]$$

Decoding movement from neural spike trains

Derive the posterior under the new model (continued)

Decoding movement from neural spike trains

Final results

$$p(X \mid Y) = \mathcal{N}(\mathrm{vec}(X) \mid \mu, \Sigma)$$

$$\Sigma = J^{-1} \qquad \mu = J^{-1}h$$

$$J = \begin{bmatrix} J_{11} & J_{21}^\top & & & \\ J_{21} & J_{22} & J_{32}^\top & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & J_{T,T-1}^\top \\ & & & J_{T,T-1} & J_{TT} \end{bmatrix} \qquad h = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_T \end{bmatrix}$$

Where

- The diagonal blocks are $J_{tt} = Q^{-1} + A^\top Q^{-1} A$ (except for J_{11} and J_{TT}).
- The lower diagonal blocks are $J_{t,t-1} = -Q^{-1}A$
- The linear coefficients are $h_t = C^\top R^{-1}(y_t - d)$.

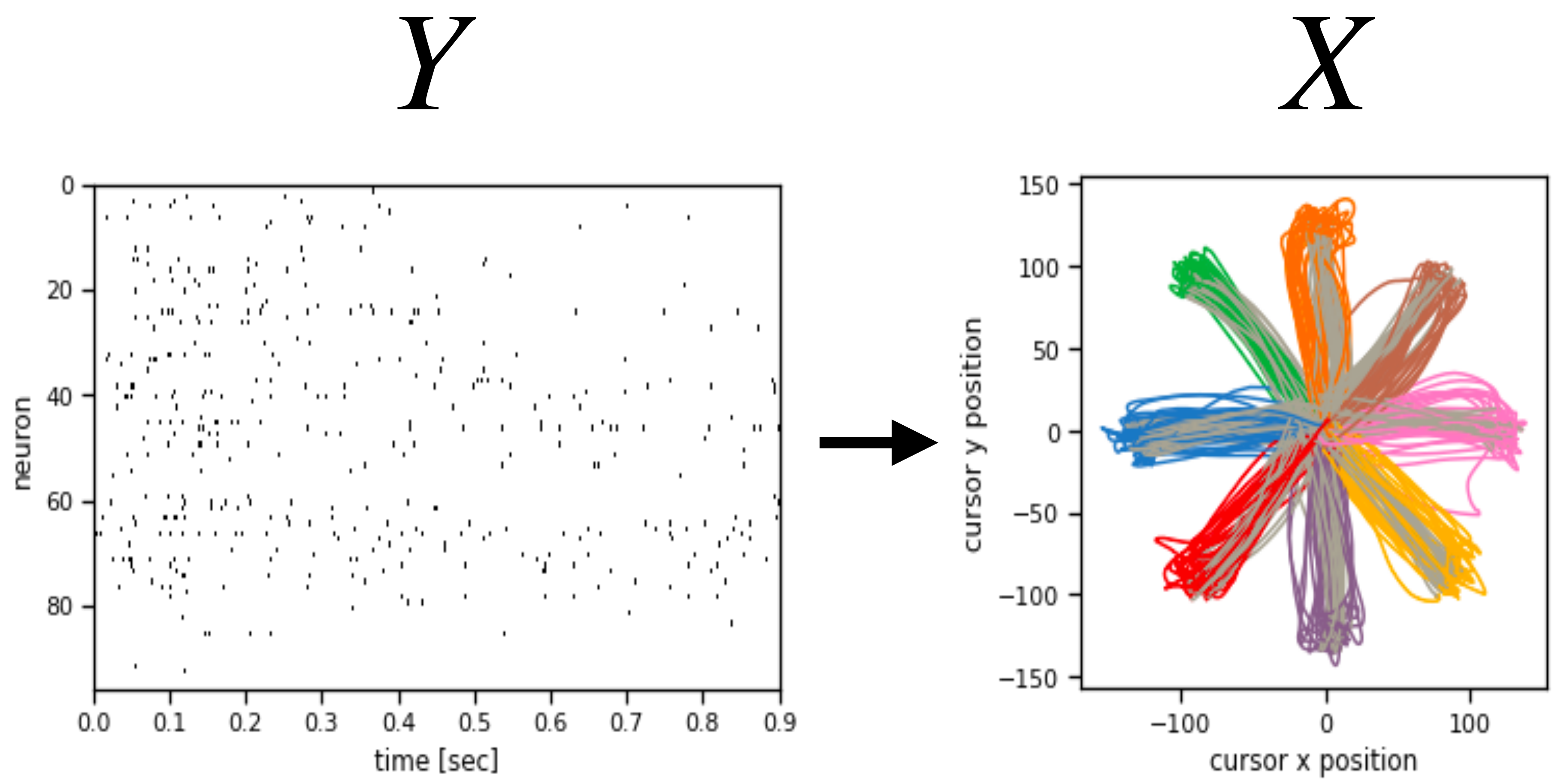
Decoding movement from neural spike trains

Poisson observations

- So far we've used a linear, Gaussian encoder for the spikes, even though they are counts!
- Suppose instead,

$$p(Y \mid X) = \prod_{t=1}^{T} \prod_{n=1}^{N} \mathrm{Po}\left(y_{tn} \mid f(c_n^\top x_t + d_n)\right)$$

- The posterior is no longer Gaussian, but it's common to approximate it as one.



Decoding movement from neural spike trains

Laplace approximation

Approximate the posterior as

$$p(X \mid Y) \approx \mathcal{N}(\mu, \Sigma)$$

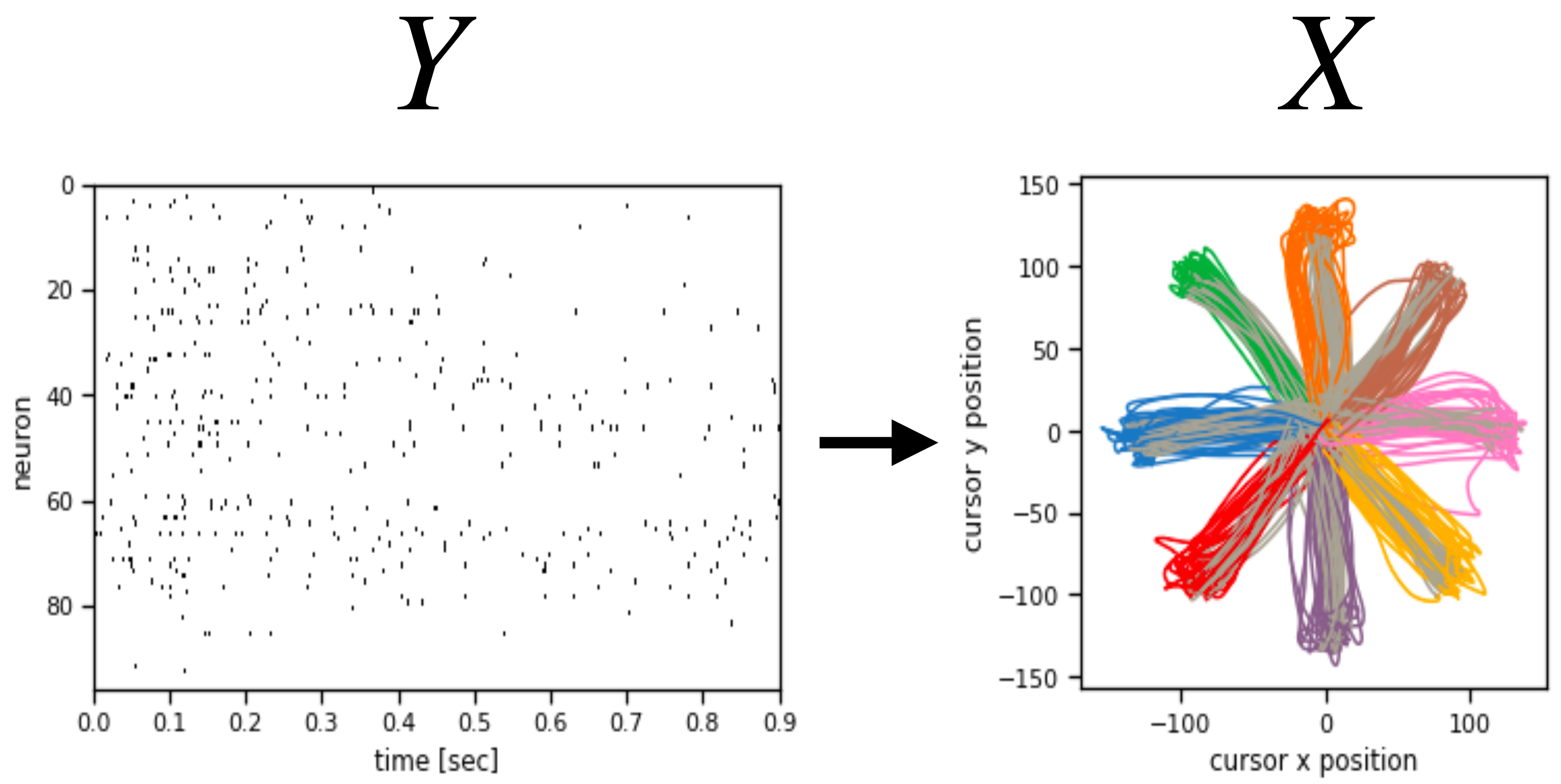
where

$$\mathcal{L}(X) = -\log p(X, Y)$$

$$\mu = \operatorname{argmin}_X \mathcal{L}(X)$$

$$\Sigma = \left[\nabla^2 \mathcal{L}(X) \Big|_{X=\mu} \right]^{-1}$$

For GLM encoders, the log joint is concave and μ and Σ can be found efficiently.



Decoding movement from neural spike trains

Laplace approximation under a Poisson GLM encoder

Derive the Hessian under the Poisson GLM encoder

$$-\log p(Y \mid X) = -\sum_{t=1}^{T} \sum_{n=1}^{N} \log \mathrm{Po}\left(y_{tn} \mid f(c_n^\top x_t + d_n)\right)$$

“Direct” decoders and structured prediction

Decoding movement from neural spike trains

Structured decoders

- If we're going to make a Gaussian approximation anyway, why not learn more flexible means and covariances?

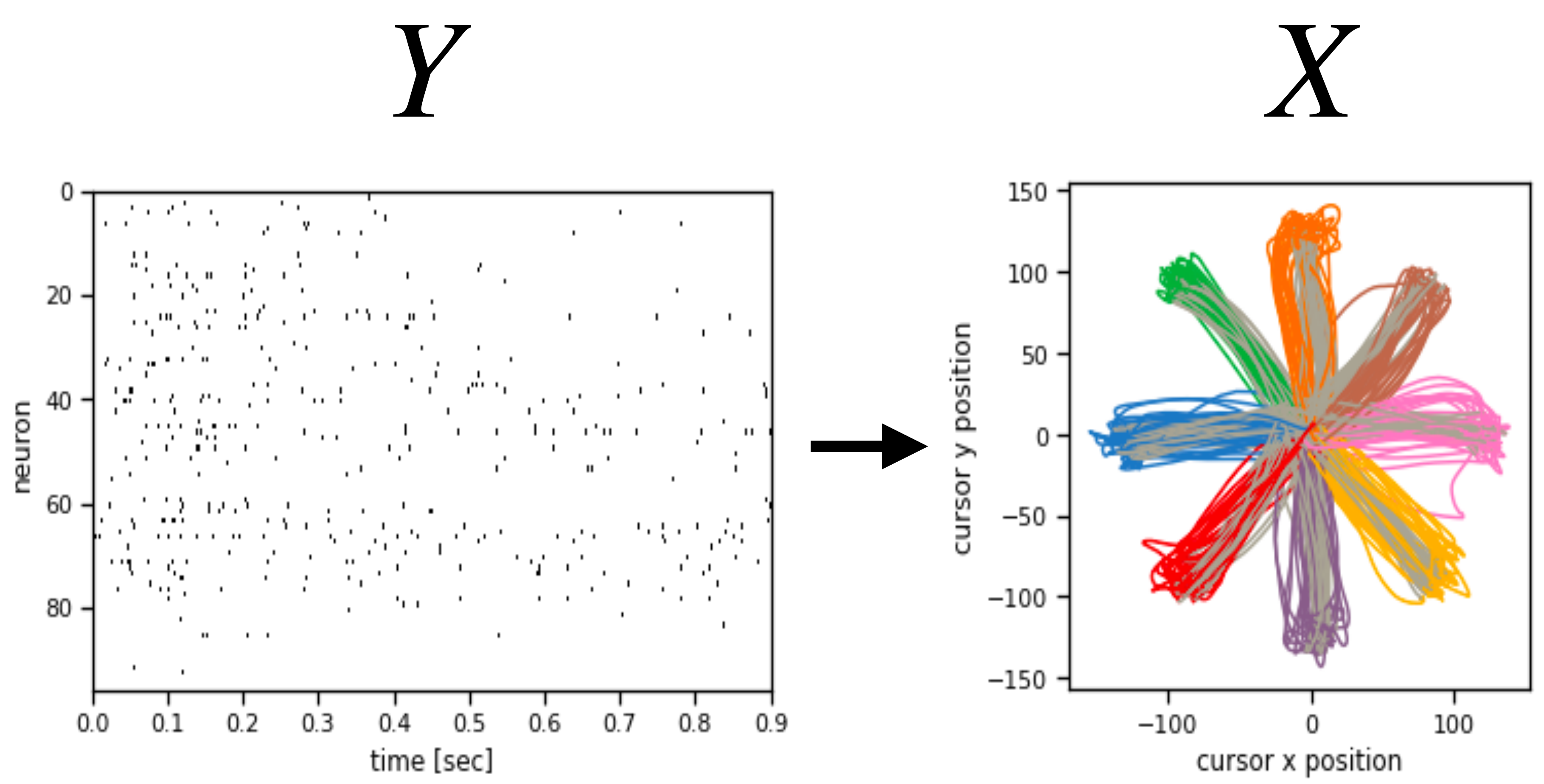
- Recall the form of the LDS posterior,

$$J_{tt} = Q^{-1} + A^\top Q^{-1} A$$

$$J_{t,t-1} = -\,Q^{-1} A$$

$$h_t = C^\top R^{-1} (y_t - d)$$

- **Idea**: replace these with learned functions of $y_{1:T}$.



Decoding movement from neural spike trains

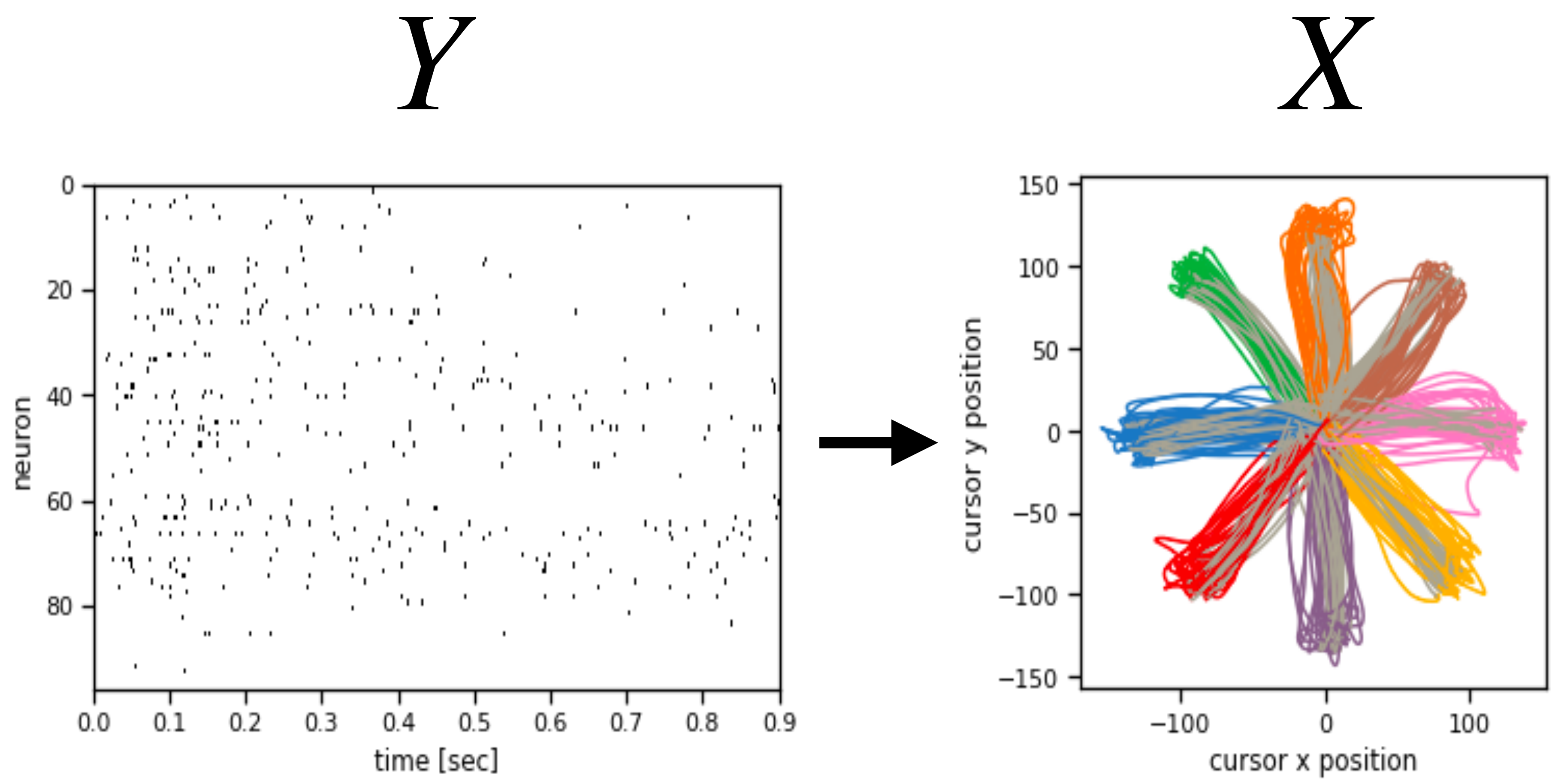
Structured decoders

- For example,

$$p(X \mid Y) = \mathcal{N}(\mathrm{vec}(X) \mid \mu, \Sigma)$$

- $$\mu = J(Y)^{-1} h(Y)$$
  $$\Sigma = J(Y)^{-1}$$

- Where $J(Y)$ is composed of blocks $J_{tt}(y_{t-\Delta:t+\Delta})$, $J_{t,t-1}(y_{t-\Delta:t+\Delta})$ and $h(Y)$ is composed of blocks $h_t(y_{t-\Delta:t+\Delta})$.



Conclusion

- Decoding and encoding are two sides of the same coin.
- We can treat decoding as a simple regression problem, but sometimes we have prior information about X or the encoder $p(Y \mid X)$ that we can leverage.
- Bayesian rule tells us how to combine prior and likelihood to derive a posterior distribution.
- However, the posterior rarely has a simple, closed form, so we need some approximations.
- Structured decoders give us a way to capture general dependency structure while allowing more flexible features of the data to be learned and incorporated.