FLATIRON INSTITUTE

Generalized Linear Models (GLM)

A conceptual introduction to GLM

Edoardo Balzani, June 24th 2024

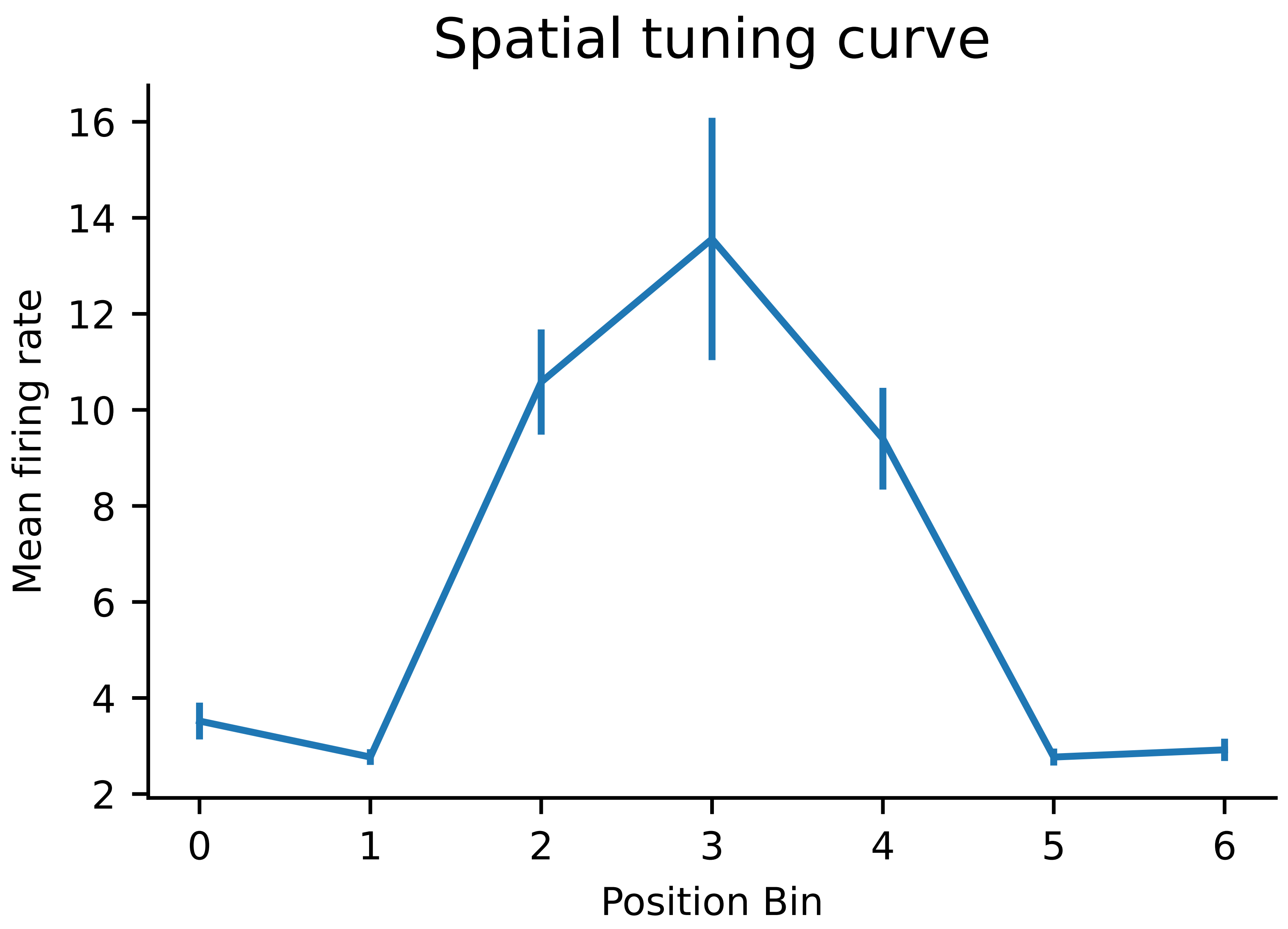
Roadmap

- Why models?
- What are GLMs?
- Why GLMs?
- What can I do with a GLM?
- What features can/should I use?
- Feature construction with Basis
- Overfitting
- Summary
- Today's roadmap

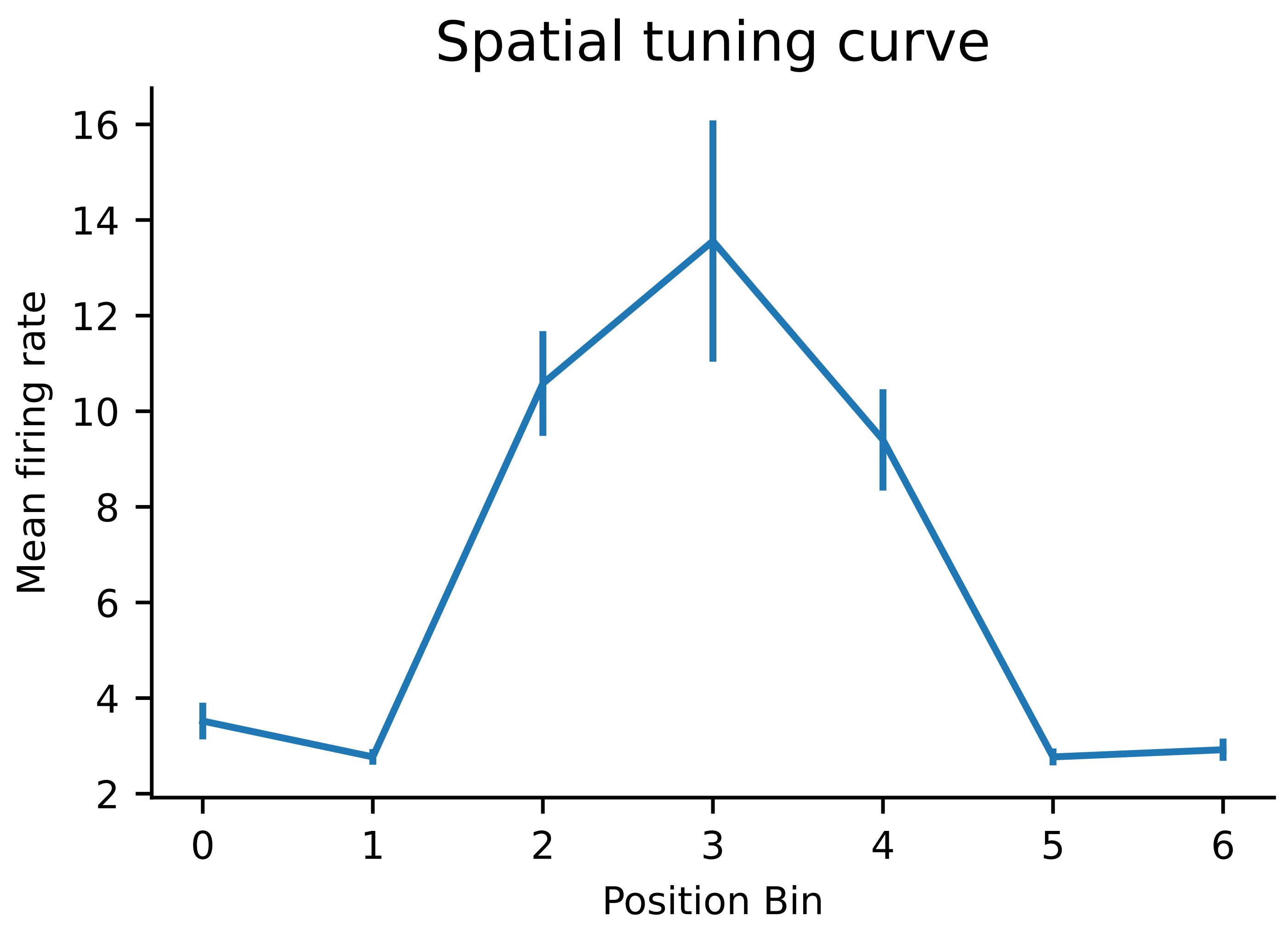
Why models? A hook

linear maze

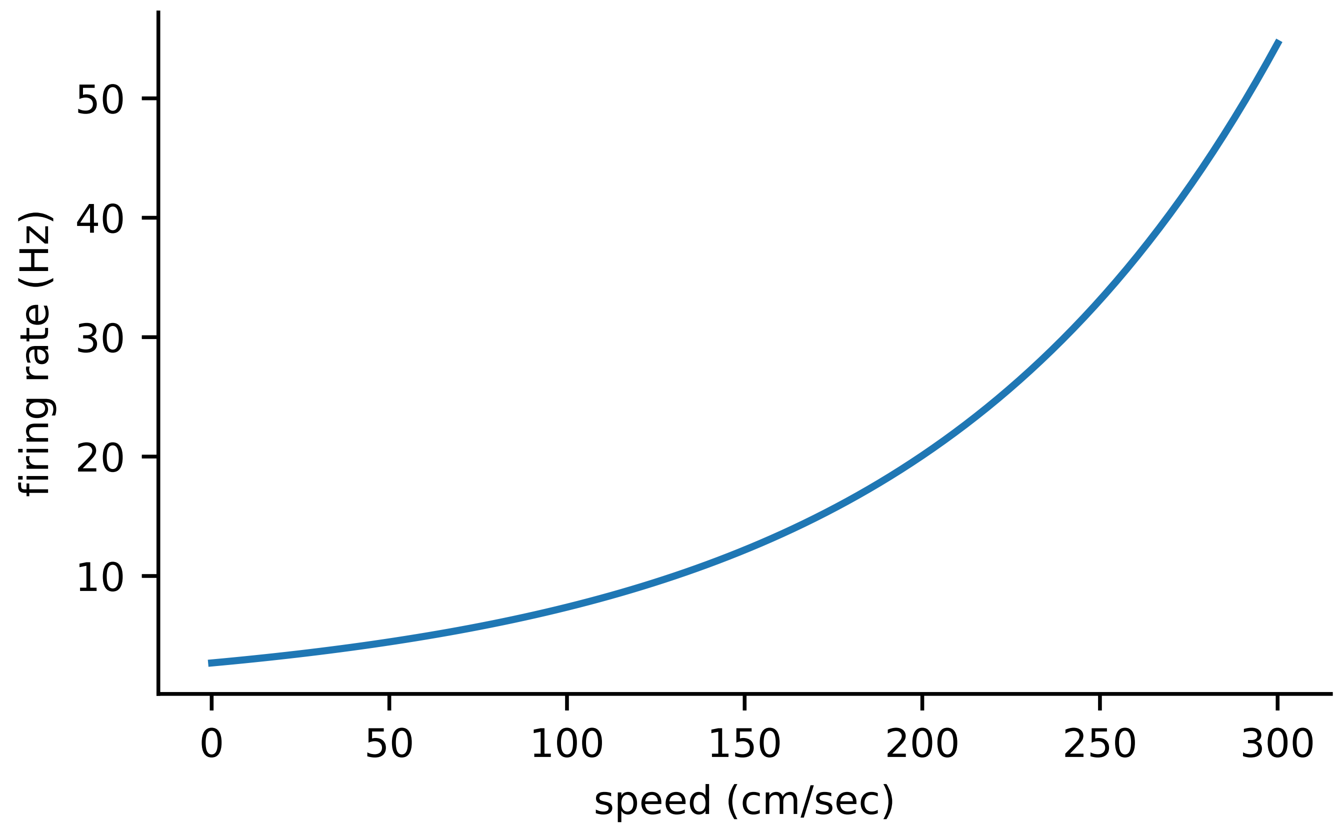
is this neuron encoding the mouse position?



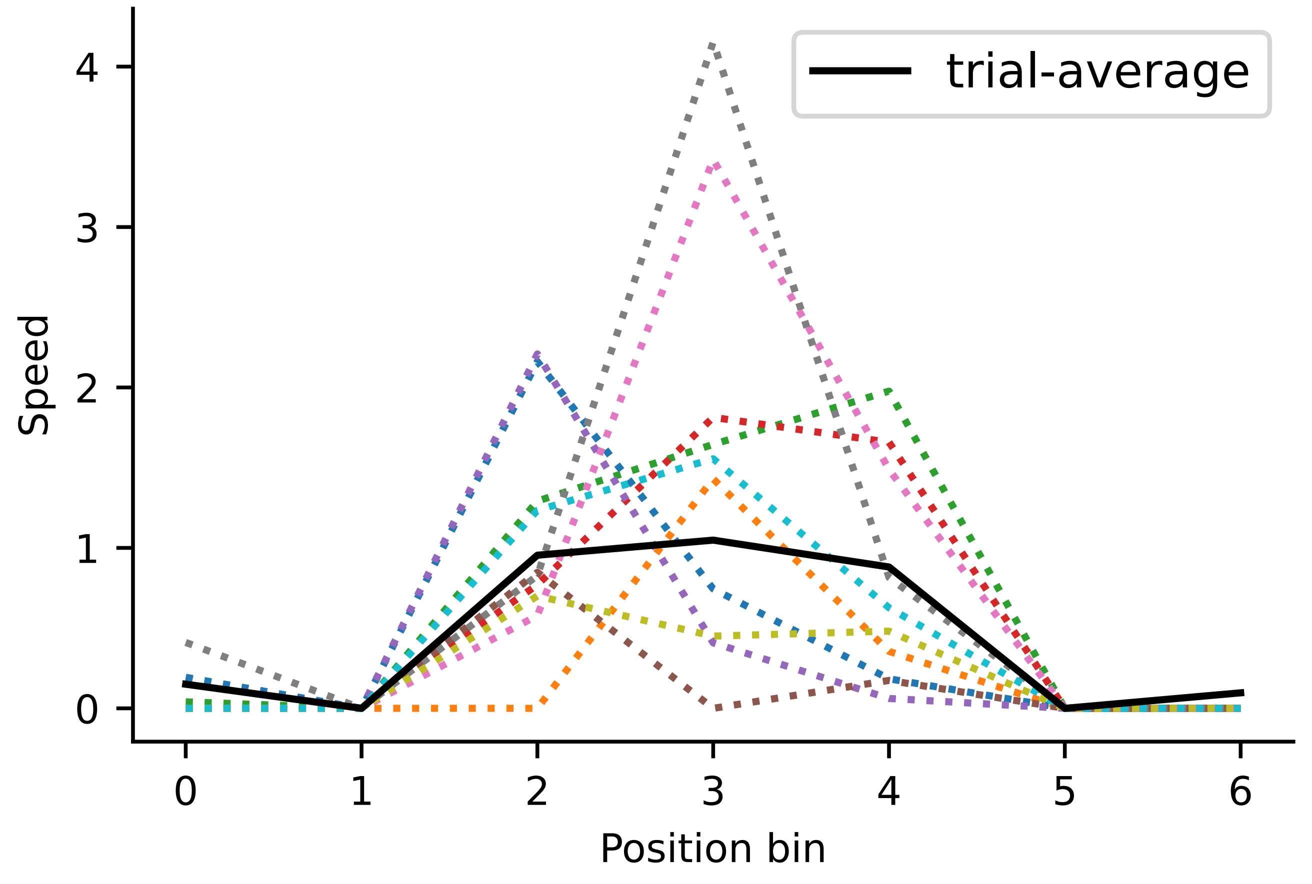
Why models? A hook



..actually, not!



position and speed
are correlated



tuning functions don't tell you the whole story
need better models!

What are GLMs?

Pre-process

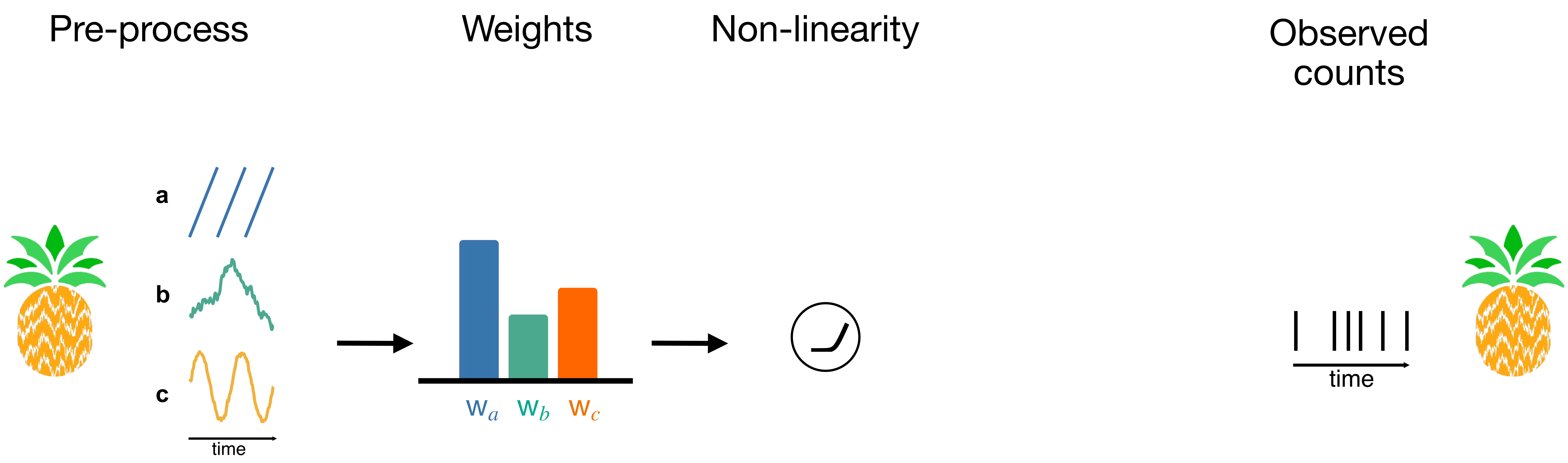
Weights

Observed counts

scale the inputs by some weights

$$\mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c$$

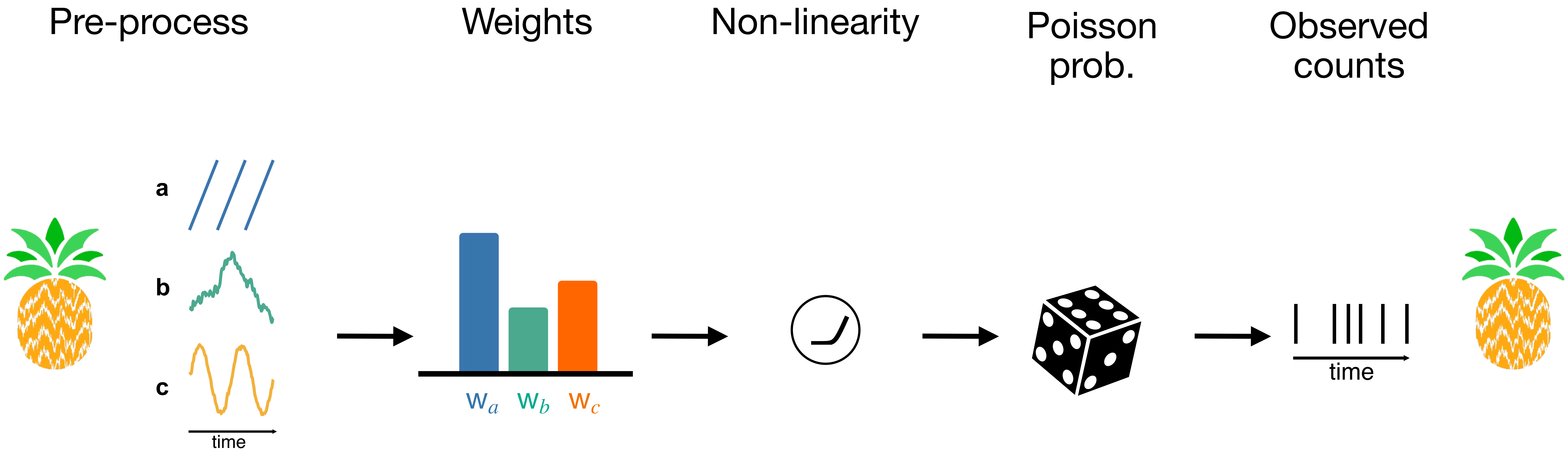
What are GLMs?



non-linearity to make the result positive

$$\text{firing rate} = \exp(\, \mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c \,)$$

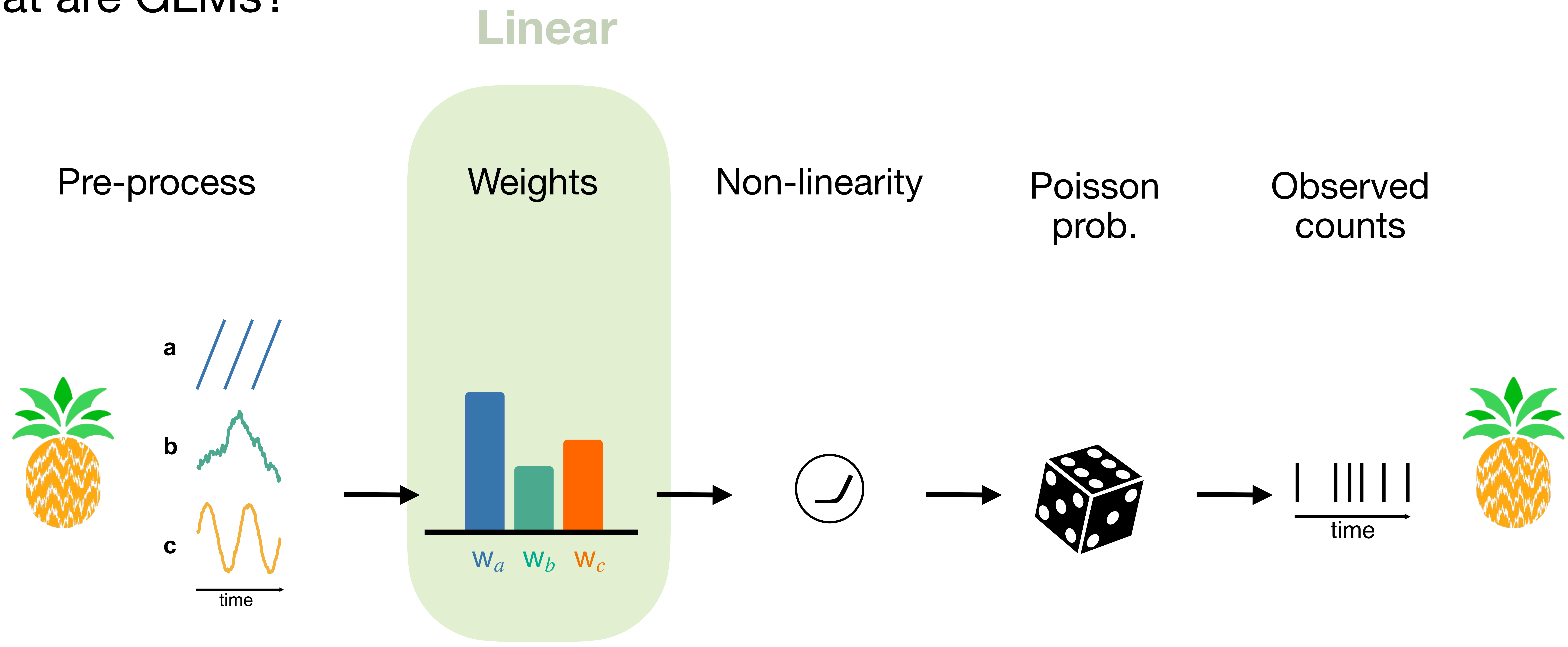
What are GLMs?



$$\text{firing rate} = \exp(\,\mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c\,)$$

$$\text{probability(spike count} = k) = \text{Poisson}(k \mid \text{firing rate})$$

What are GLMs?

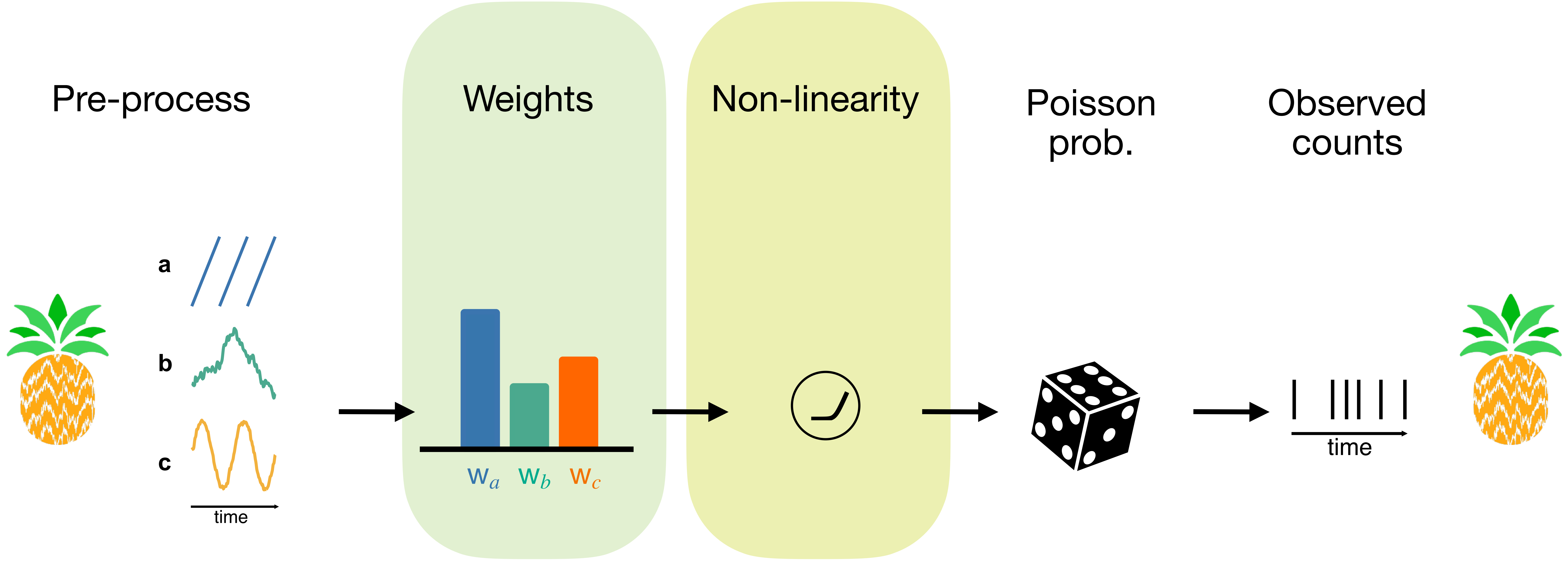


firing rate = exp(**a** $\cdot\, w_a$ + **b** $\cdot\, w_b$ + **c** $\cdot\, w_c$)

probability(spike count $= k$) $=$ Poisson($k \,|\,$ firing rate)

What are GLMs?

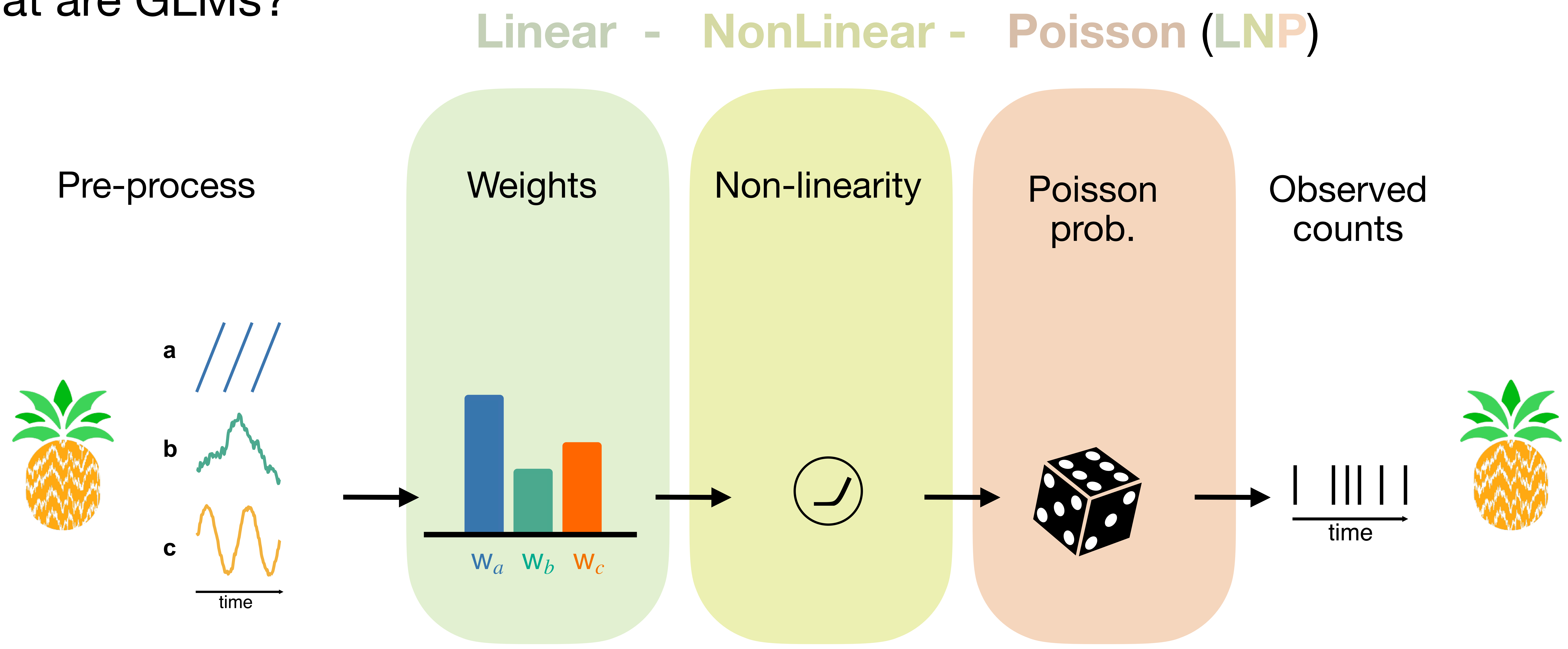
Linear - NonLinear



$$\text{firing rate} = \exp(\,\mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c\,)$$

$$\text{probability(spike count} = k) = \text{Poisson}(k \mid \text{firing rate})$$

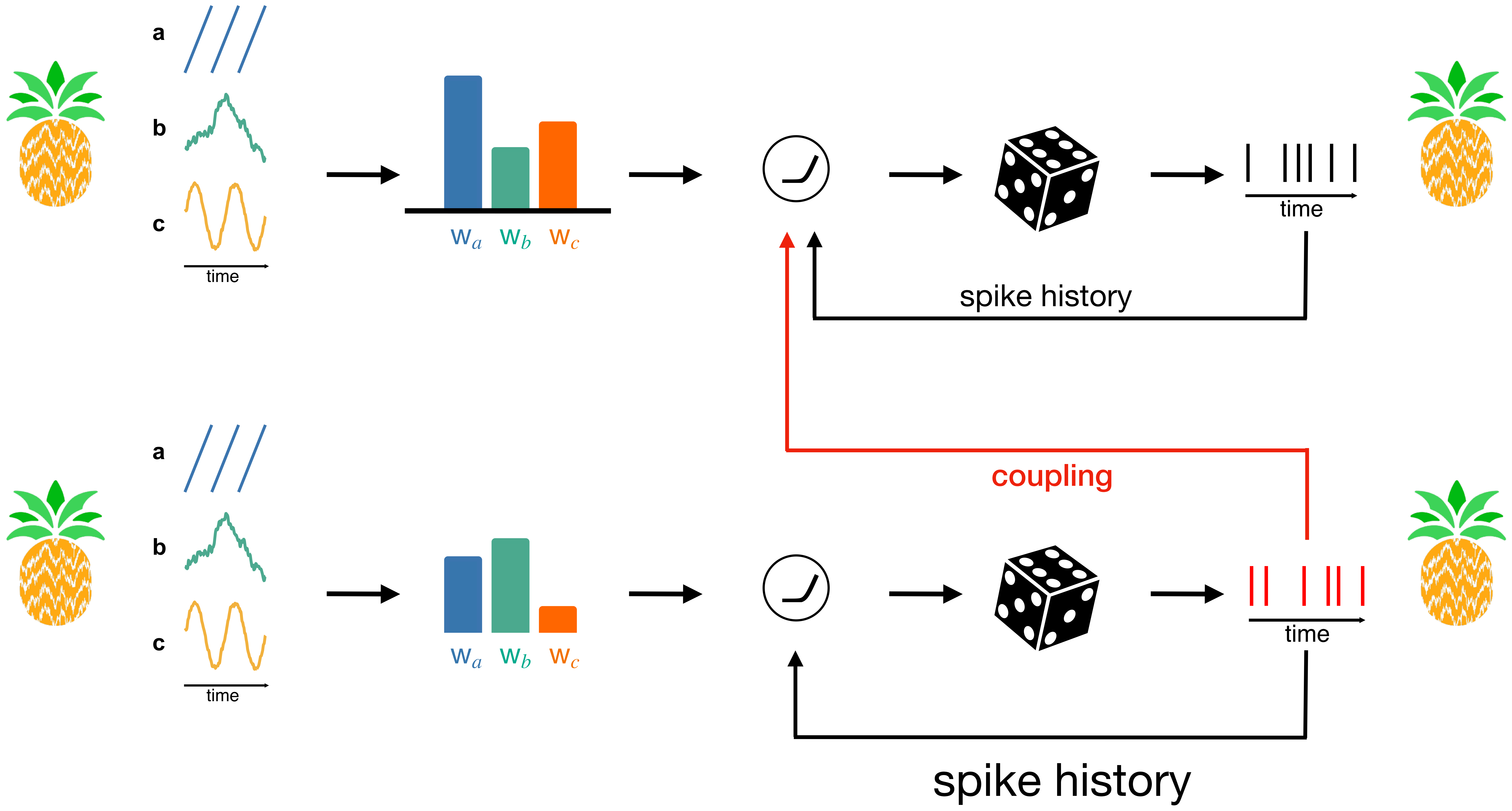
What are GLMs?



$$\text{firing rate} = \exp(\, \mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c \,)$$

$$\text{probability(spike count} = k) = \text{Poisson}(k \,|\, \text{firing rate})$$

What are GLMs?



Terminology

$$\text{firing rate} = \exp(\,\mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c\,)$$

- **a**, **b**, **c** are called **features** or **predictors**

Terminology

$$\text{firing rate} = \exp(\, \mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c \,)$$

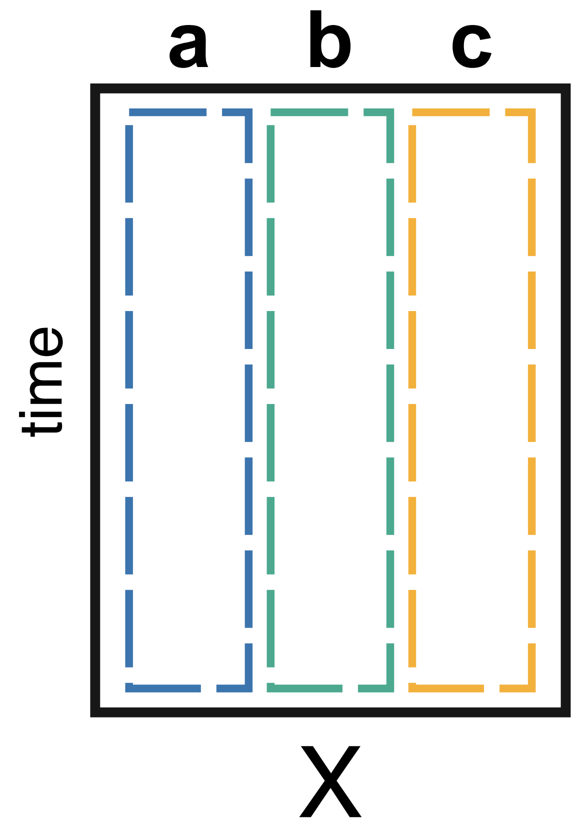
- **a**, **b**, **c** are called **features** or **predictors**
- $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$ are called **weights** or **coefficients**

Terminology

$$\text{firing rate} = \exp(\,\mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c\,)$$

- **a**, **b**, **c** are called **features** or **predictors**
- $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$ are called **weights** or **coefficients**
- Features are concatenated to form the **design** or **feature** matrix $\mathbf{X} = [\mathbf{a}, \mathbf{b}, \mathbf{c}]$

Design matrix

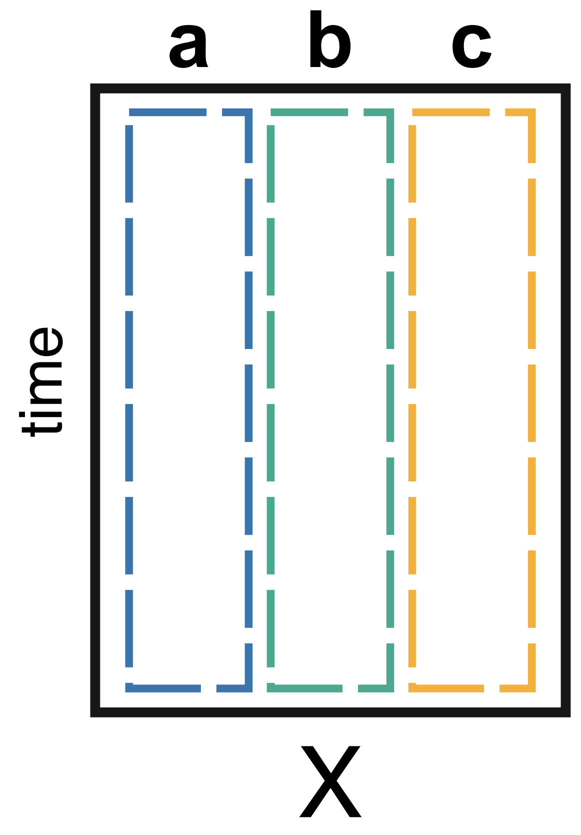


Terminology

$$\text{firing rate} = \exp(\mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c)$$

- **a**, **b**, **c** are called **features** or **predictors**
- $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$ are called **weights** or **coefficients**
- Features are concatenated to form the **design** or **feature** matrix $\mathbf{X} = [\mathbf{a}, \mathbf{b}, \mathbf{c}]$
- The **likelihood** is the probability of observing spike counts given some features and weights.

Design matrix



Likelihood

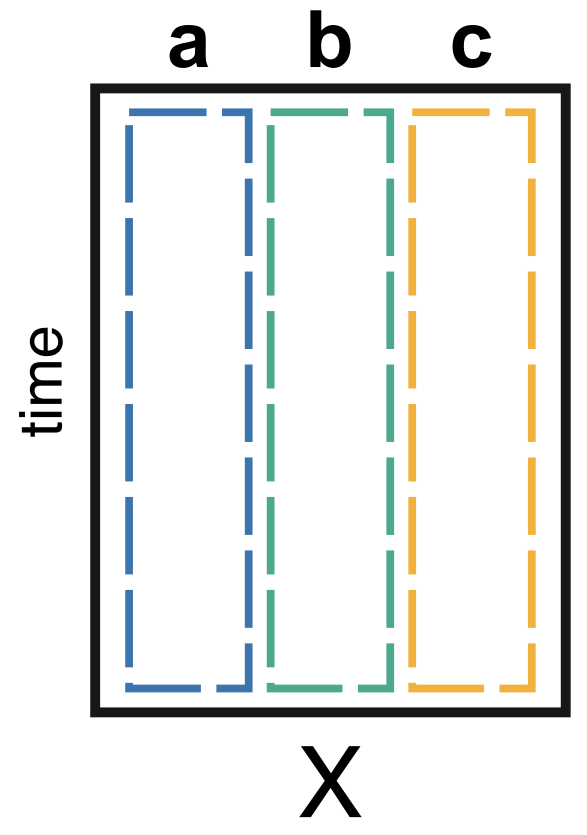
$$\text{probability}(\text{spike count} = k \,|\, \mathbf{X}, \mathbf{w})$$

Terminology

$$\text{firing rate} = \exp(\, \mathbf{a} \cdot w_a + \mathbf{b} \cdot w_b + \mathbf{c} \cdot w_c \,)$$

- **a**, **b**, **c** are called **features** or **predictors**
- $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$ are called **weights** or **coefficients**
- Features are concatenated to form the **design** or **feature** matrix $\mathbf{X} = [\mathbf{a}, \mathbf{b}, \mathbf{c}]$
- The **likelihood** is the probability of observing spike counts given some features and weights.
- The **likelihood is a function of the weights** because counts and features are fixed.

Design matrix



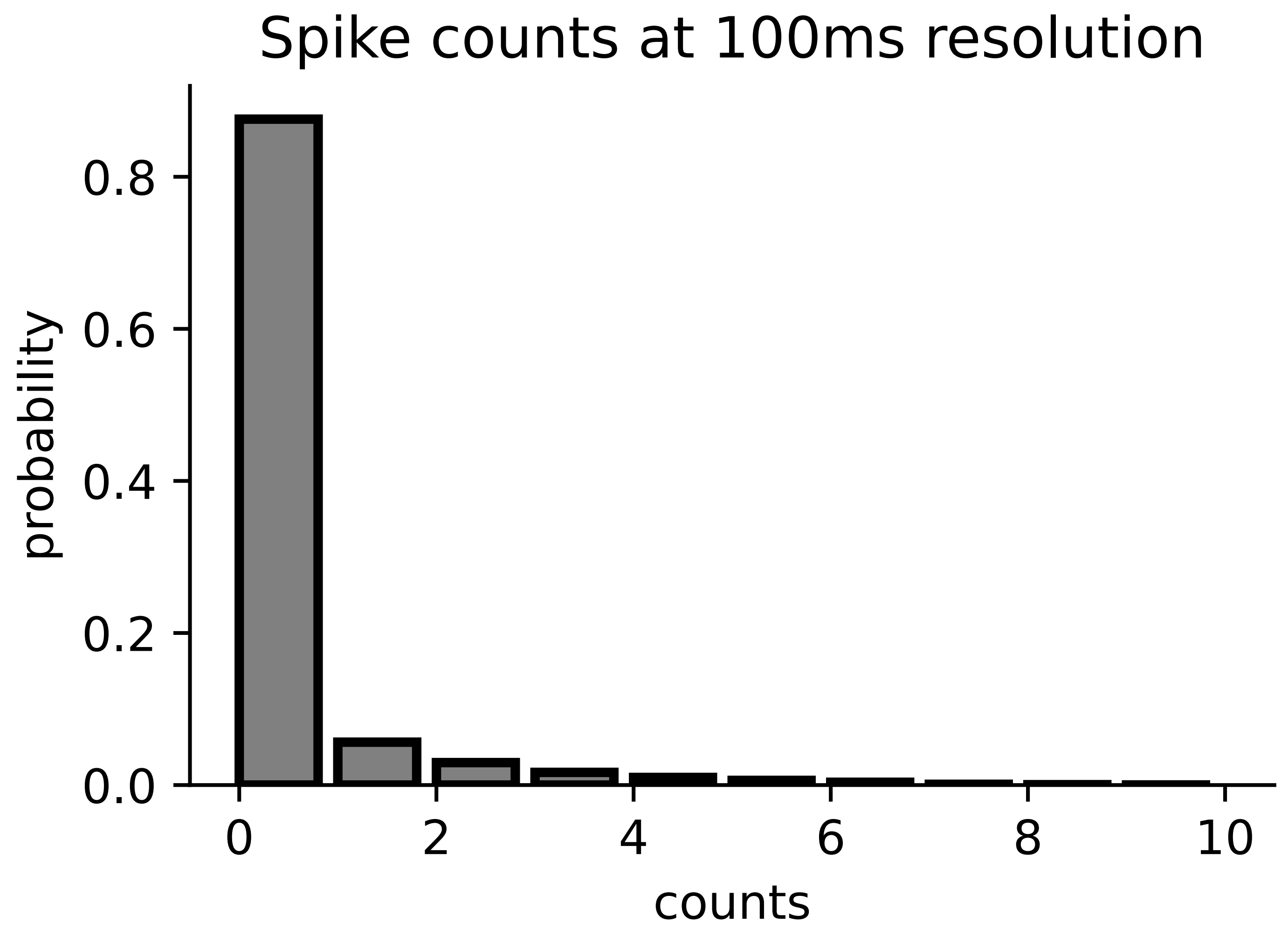
Likelihood

$$\text{probability}(\text{spike count} = k \,|\, \mathbf{X}, \mathbf{w})$$

Why GLMs?

1. Why not linear regression?
   which assumes normality

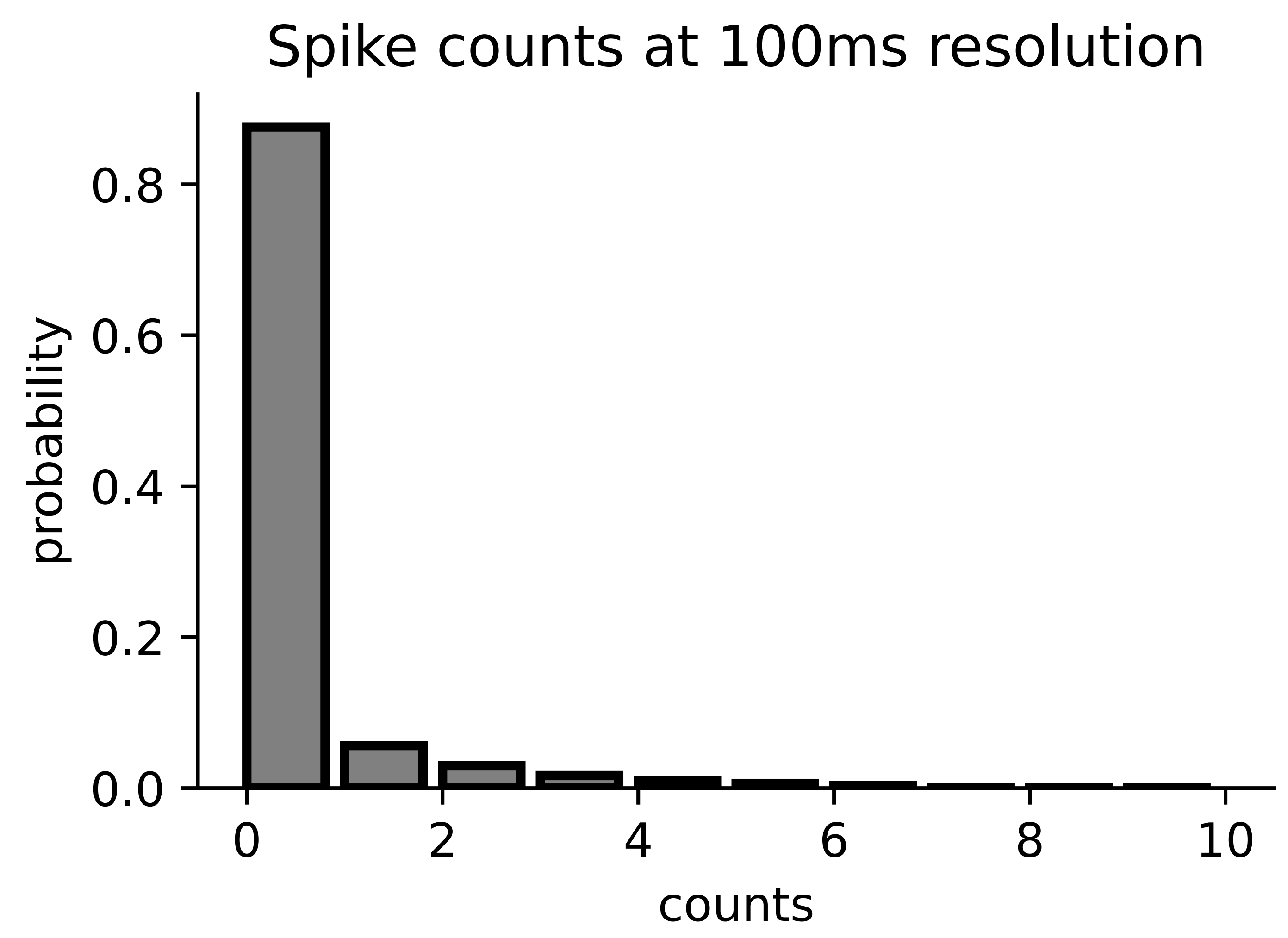
   A. Spike counts are non-Gaussian

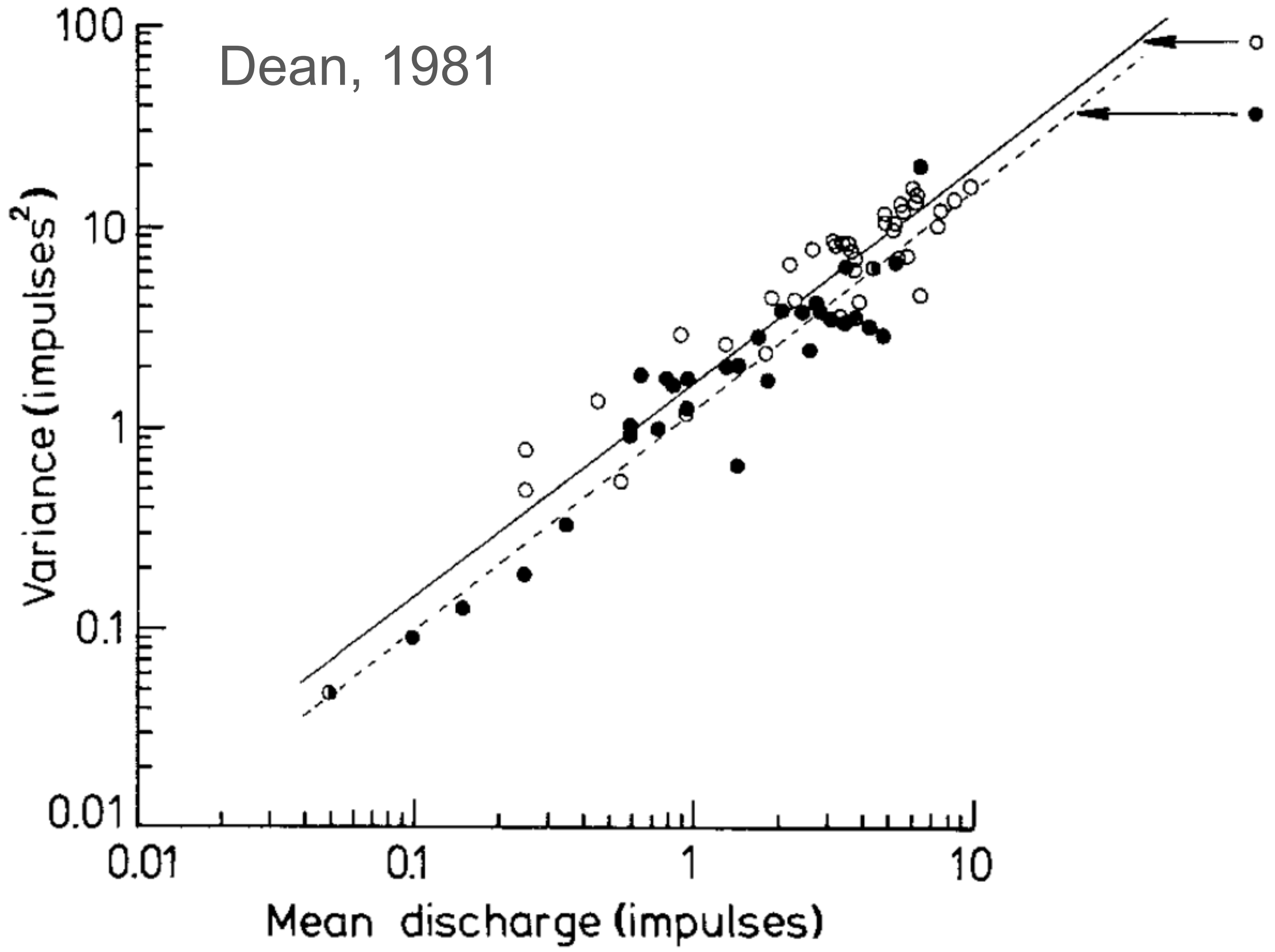


Why GLMs?

1. Why not linear regression?
   which assumes normality

   A. Spike counts are non-Gaussian
   B. Neural activity variance is non-constant





Why GLMs?

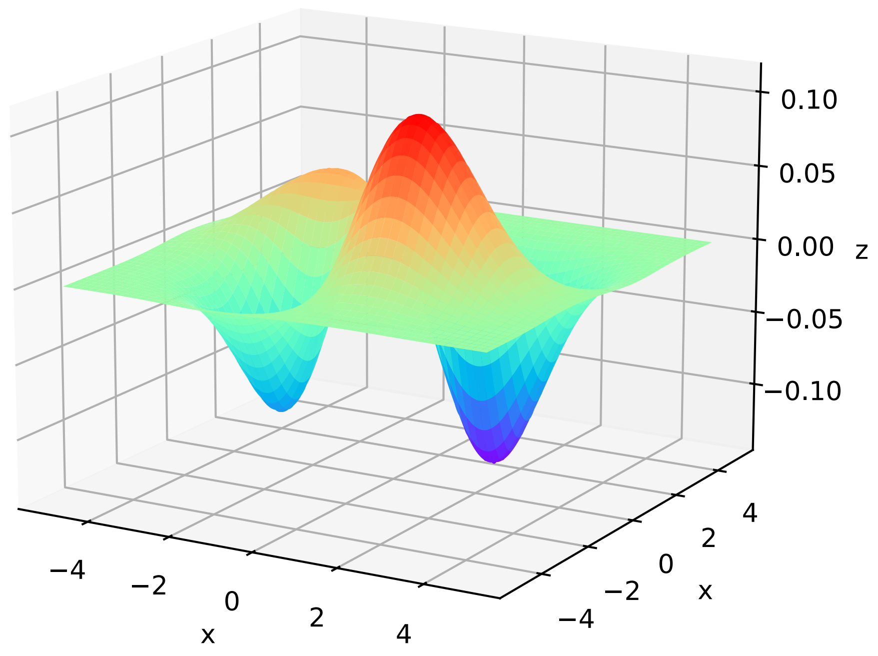
1. Why not linear regression?
   which assumes normality

   A. Spike counts are non-Gaussian

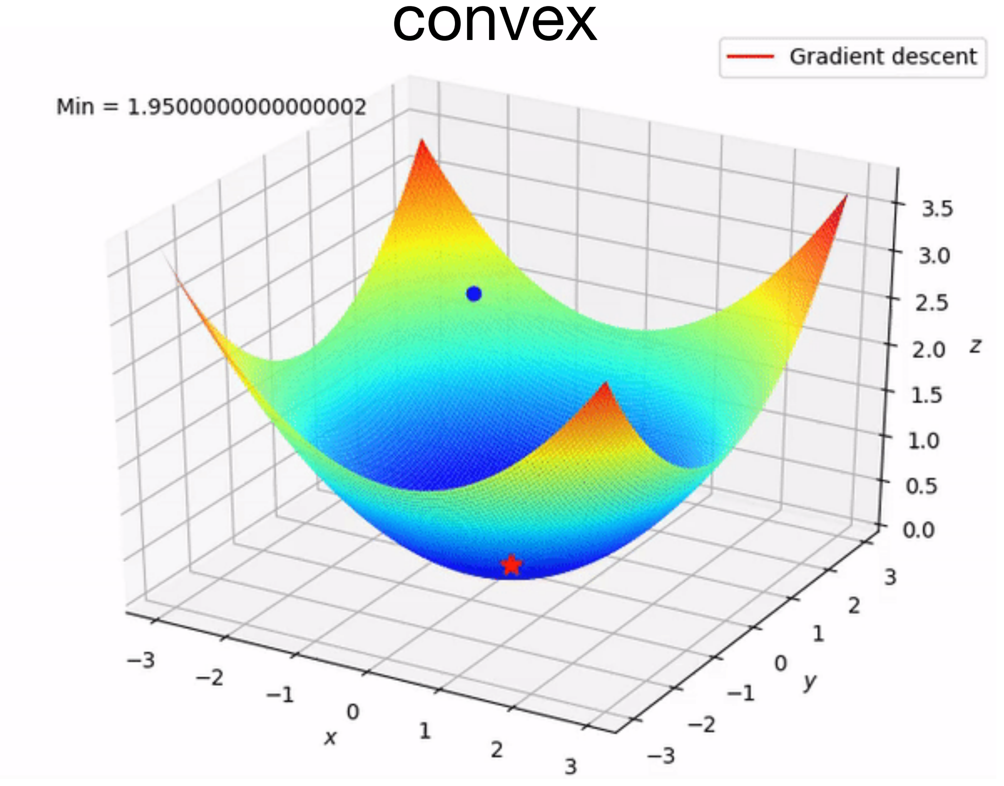
   B. Neural activity variance is non-constant

2. GLM are as **easy to fit** as linear regression
   convex, unique optimal solution

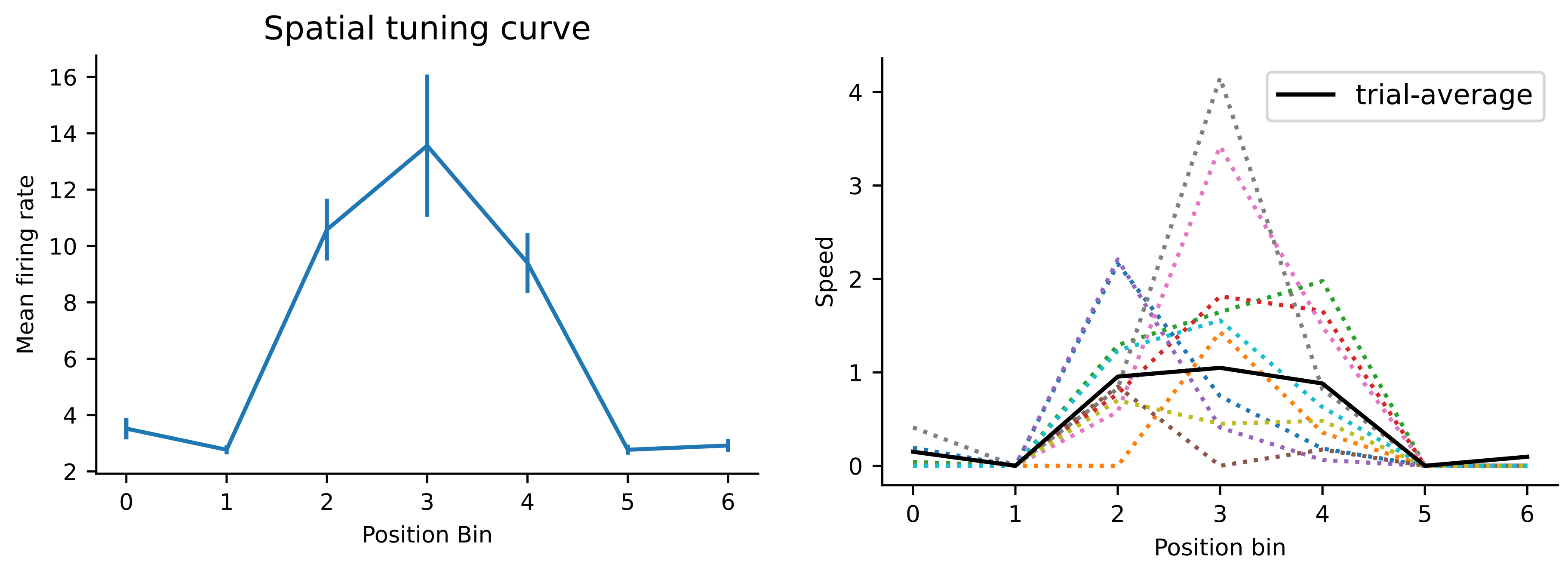
non-convex



convex



Why GLMs?



1. Why not linear regression?
   which assumes normality

   A. Spike counts are non-Gaussian

   B. Neural activity variance is non-constant

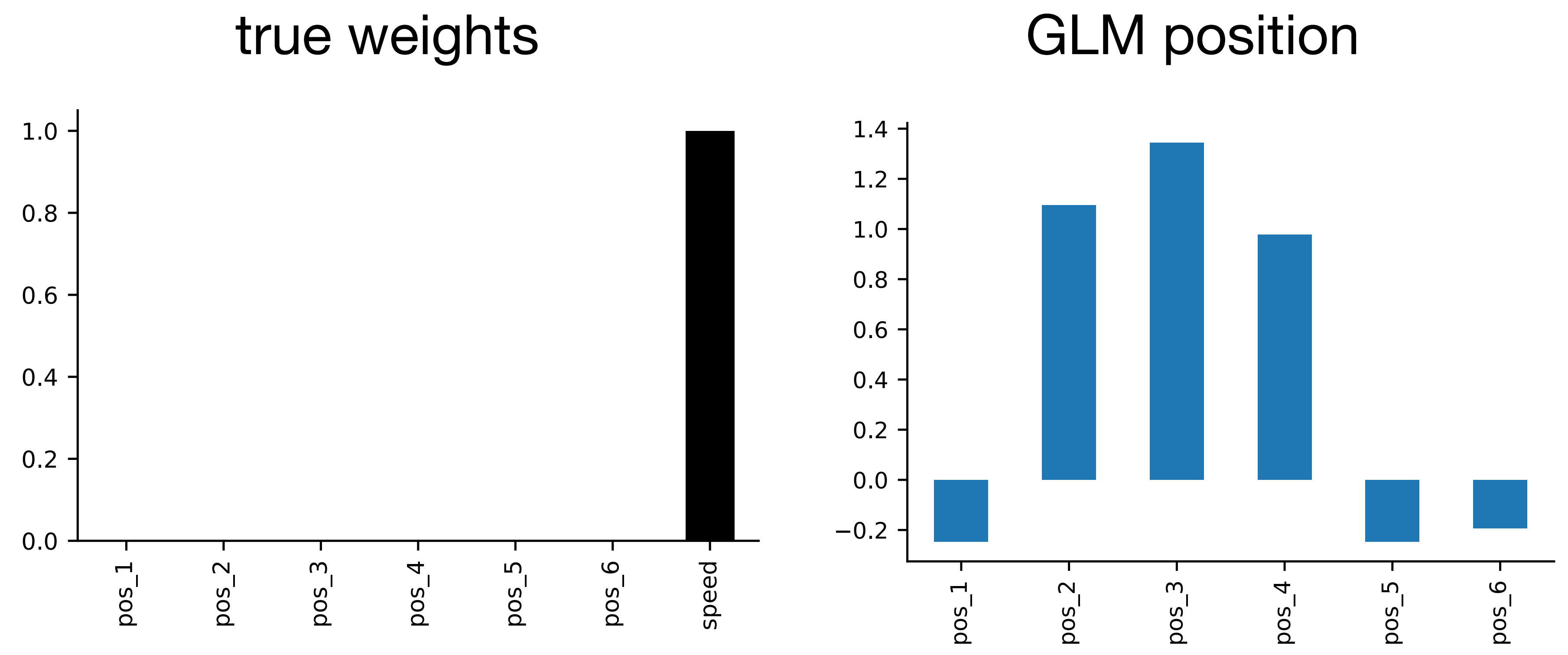
2. GLM are as **easy to fit** as linear regression
   convex, unique optimal solution

3. GLM are **flexible**
   model multiple inputs jointly

Firing rate model:

$$\text{firing rate} = \exp(w_0 \cdot \text{pos}_0(t) + \ldots + w_6 \cdot \text{pos}_6(t))$$

$$\text{pos}_i(t) = \begin{cases} 1 & \text{if mouse is in position } i \text{ at time} t \\ 0 & \text{otherwise} \end{cases}$$



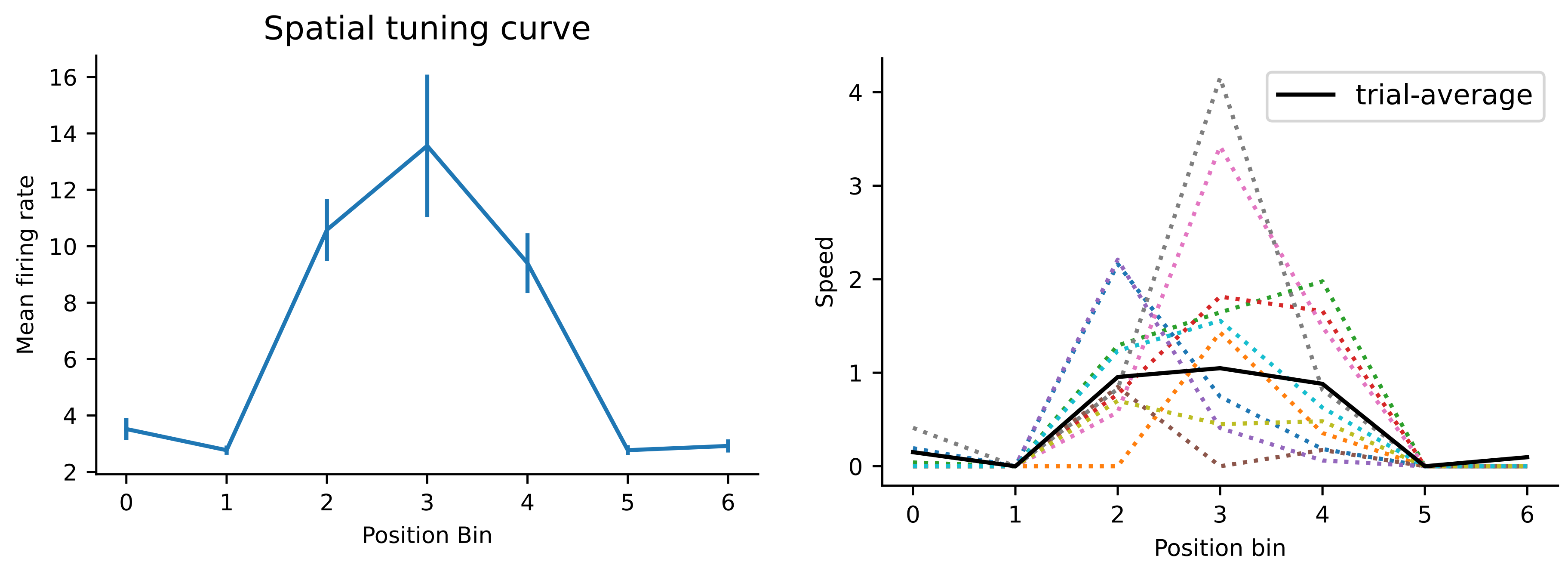
Why GLMs?

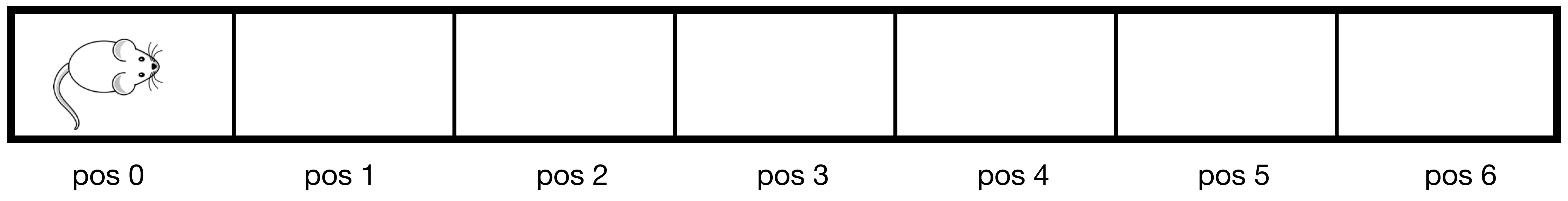
1. Why not linear regression?
   which assumes normality

   A. Spike counts are non-Gaussian
   B. Neural activity variance is non-constant

2. GLM are as **easy to fit** as linear regression
   convex, unique optimal solution

3. GLM are **flexible**
   model multiple inputs jointly

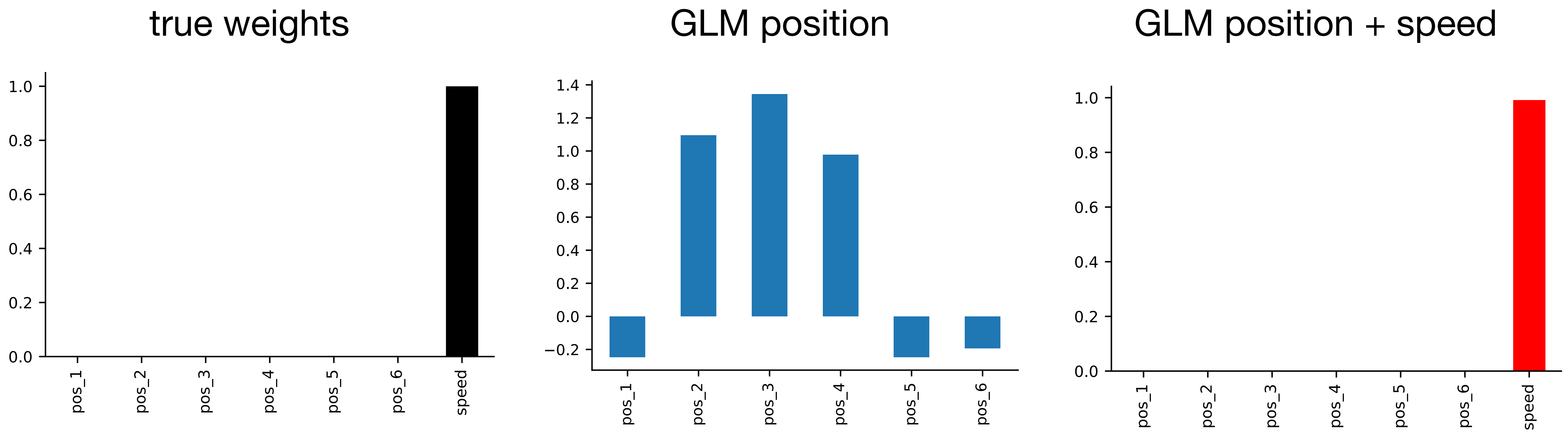




Firing rate model:

$$\text{firing rate} = \exp(w_0 \cdot \text{pos}_0(t) + \ldots + w_6 \cdot \text{pos}_6(t) + w_s \cdot \text{speed}(t))$$

$$\text{pos}_i(t) = \begin{cases} 1 & \text{if mouse is in position } i \text{ at time} t \\ 0 & \text{otherwise} \end{cases}$$



What can I do with a GLM?

1. Model responses to high dimensional inputs

   images, videos, 2D/3D positions…

Pillow at al., 2008
Retina Macaques

Hardcastle et al., 2018
MEC mice

Gardner et al. 2019
MEC rats

Park et al. 2019
LIP Macaques

Weber & Pillow 2017
simulations

Peyrache et al., 2018
ADN mice

What can I do with a GLM?

1. Model responses to high dimensional inputs

   images, videos, 2D/3D positions…

2. Non-linear responses

   place cells, head-direction, grid cells

<u>Pillow at al., 2008</u>
Retina Macaques

<u>Hardcastle et al., 2018</u>
MEC mice

<u>Gardner et al. 2019</u>
MEC rats

<u>Park et al. 2019</u>
LIP Macaques

<u>Weber & Pillow 2017</u>
simulations

<u>Peyrache et al., 2018</u>
ADN mice

What can I do with a GLM?

1. Model responses to high dimensional inputs

   images, videos, 2D/3D positions…

2. Non-linear responses

   place cells, head-direction, grid cells

3. Functional connectivity

   and other time-dependent effects

Pillow at al., 2008
Retina Macaques

Hardcastle et al., 2018
MEC mice

Gardner et al. 2019
MEC rats

Park et al. 2019
LIP Macaques

Weber & Pillow 2017
simulations

Peyrache et al., 2018
ADN mice

What can I do with a GLM?

1. Model responses to high dimensional inputs

   images, videos, 2D/3D positions…

2. Non-linear responses

   place cells, head-direction, grid cells

3. Functional connectivity

   and other time-dependent effects

4. Generate surrogate dataset

Pillow at al., 2008
Retina Macaques

Hardcastle et al., 2018
MEC mice

Gardner et al. 2019
MEC rats

Park et al. 2019
LIP Macaques

Weber & Pillow 2017
simulations

Peyrache et al., 2018
ADN mice

What features can/should I use?

- It’s up to the scientist!
- Choosing features is a way to formulate hypothesis about the neural encoding.
- Any fixed (not learned) transformation of your data is valid* (counting, binning, projecting into Principal Components, filtering, squaring …)

**as long as the resulting time axis matches that of the spike counts*

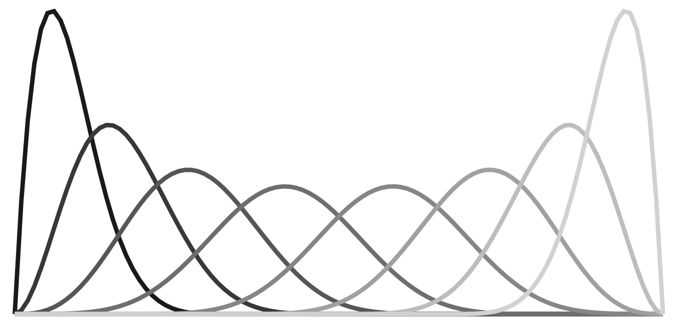
Constructing Features in NeMoS

- NeMoS provides the **basis** module for feature construction

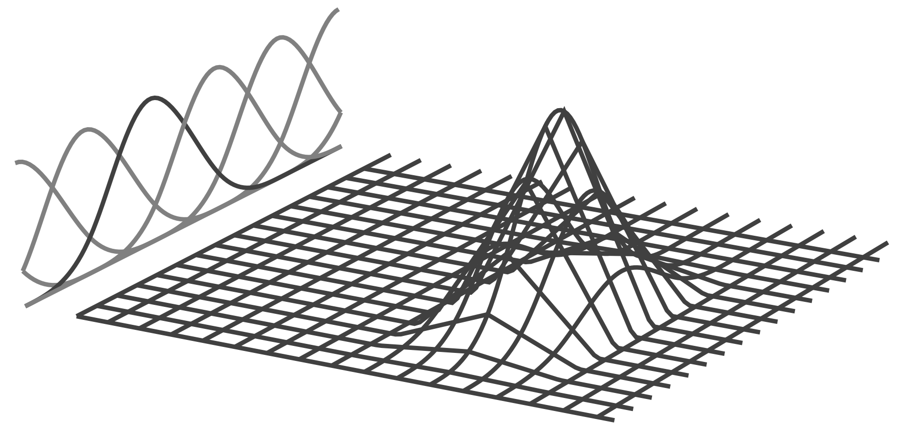
Constructing Features in NeMoS

- NeMoS provides the **basis** module for feature construction
- Basis are **fixed non-linearities**

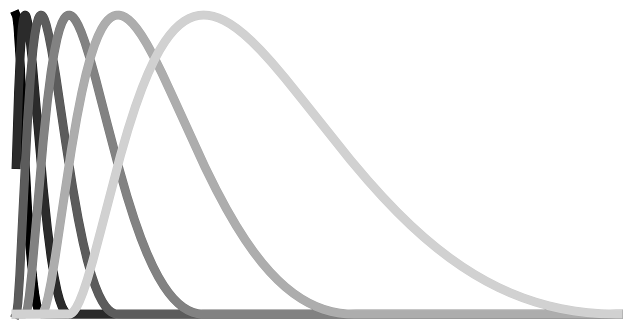
1D



2D



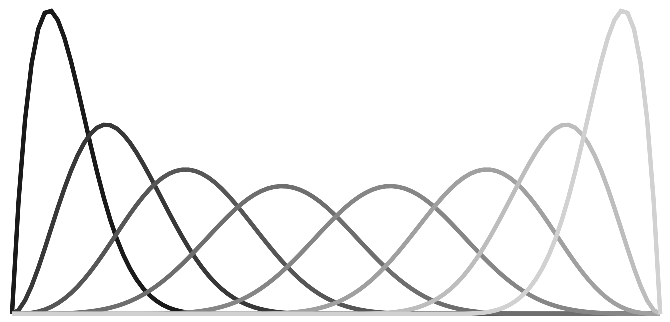
log-stretched



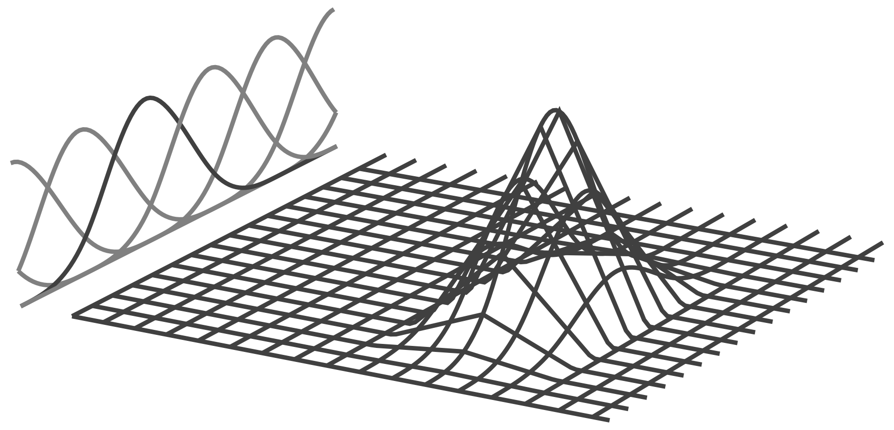
Constructing Features in NeMoS

- NeMoS provides the **basis** module for feature construction
- Basis are **fixed non-linearities**
- Assume that **firing rate varies smoothly/ gradually**

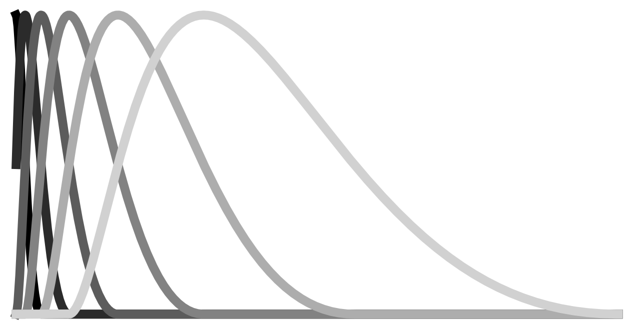
1D



2D



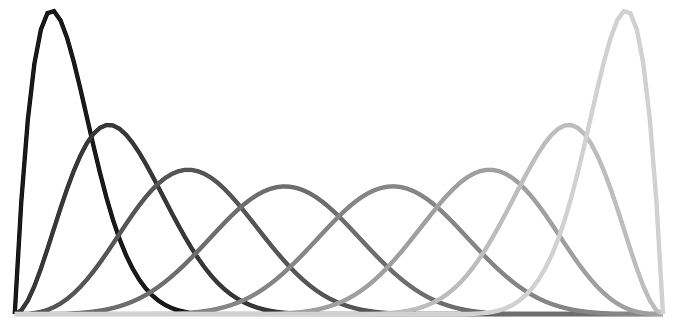
log-stretched



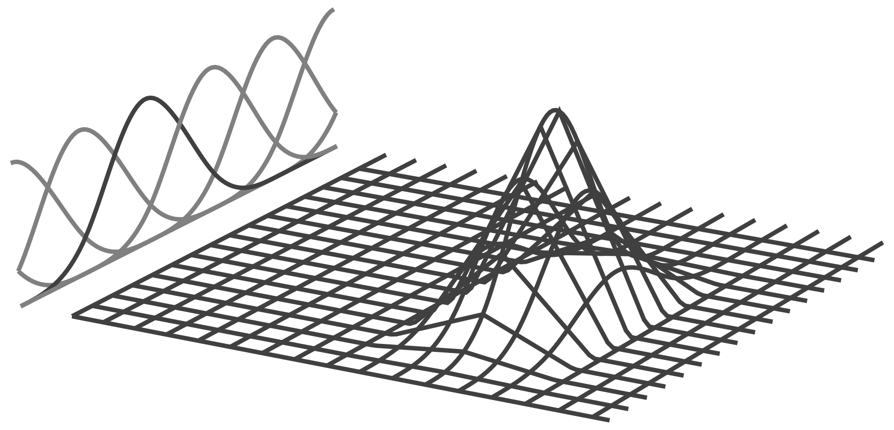
Constructing Features in NeMoS

- NeMoS provides the **basis** module for feature construction
- Basis are **fixed non-linearities**
- Assume that **firing rate varies smoothly/ gradually**
- Used for:
    1. Reducing dimensionality
    2. Non-linear firing rate modulation
    3. Time dependent effects

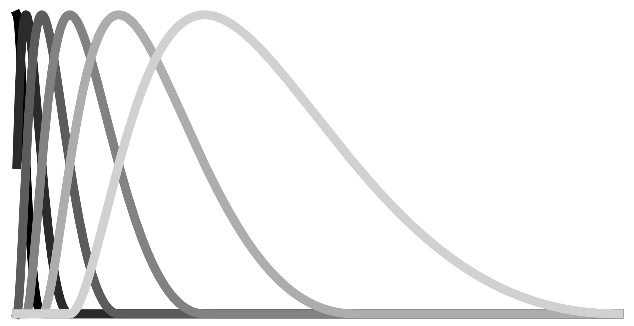
1D



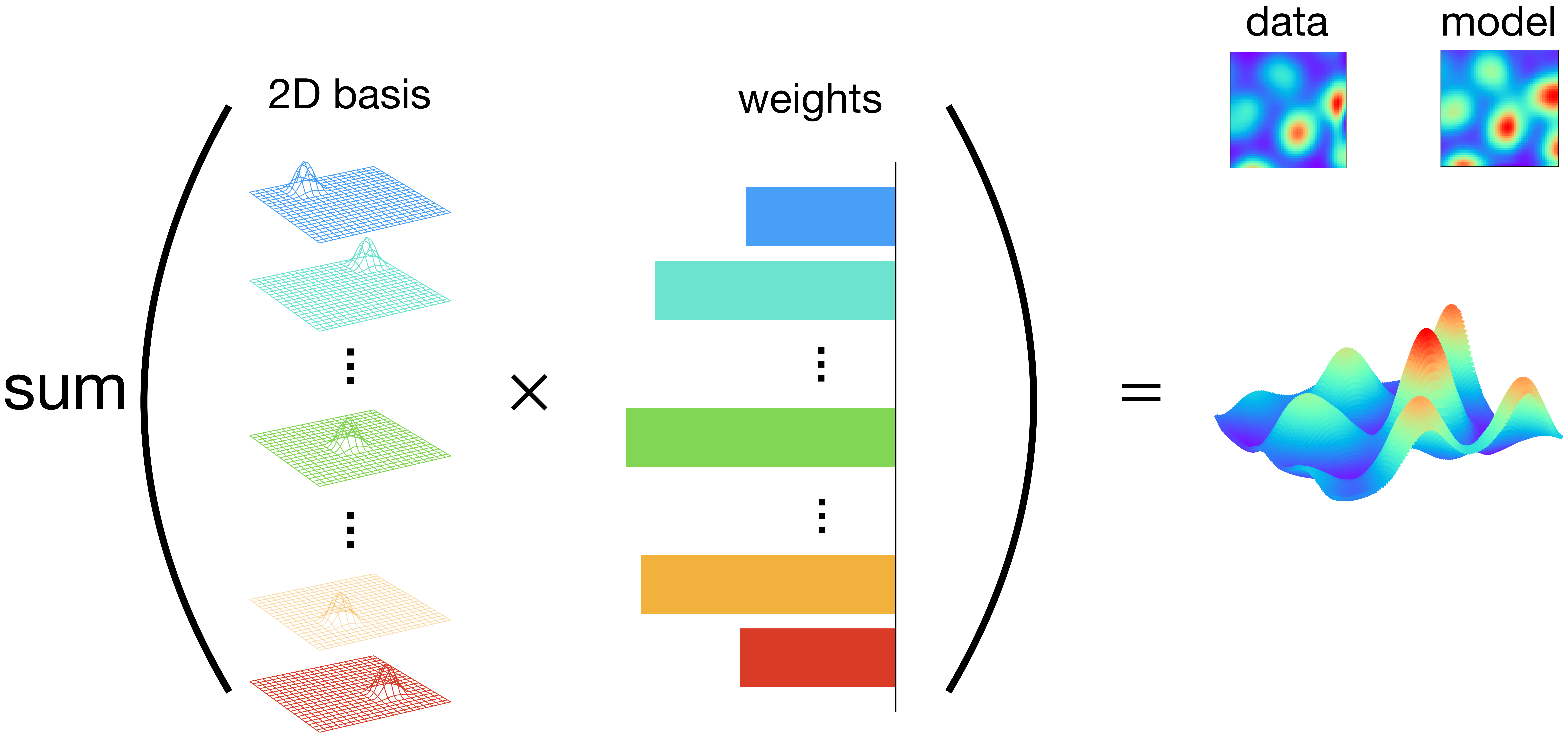
2D



log-stretched

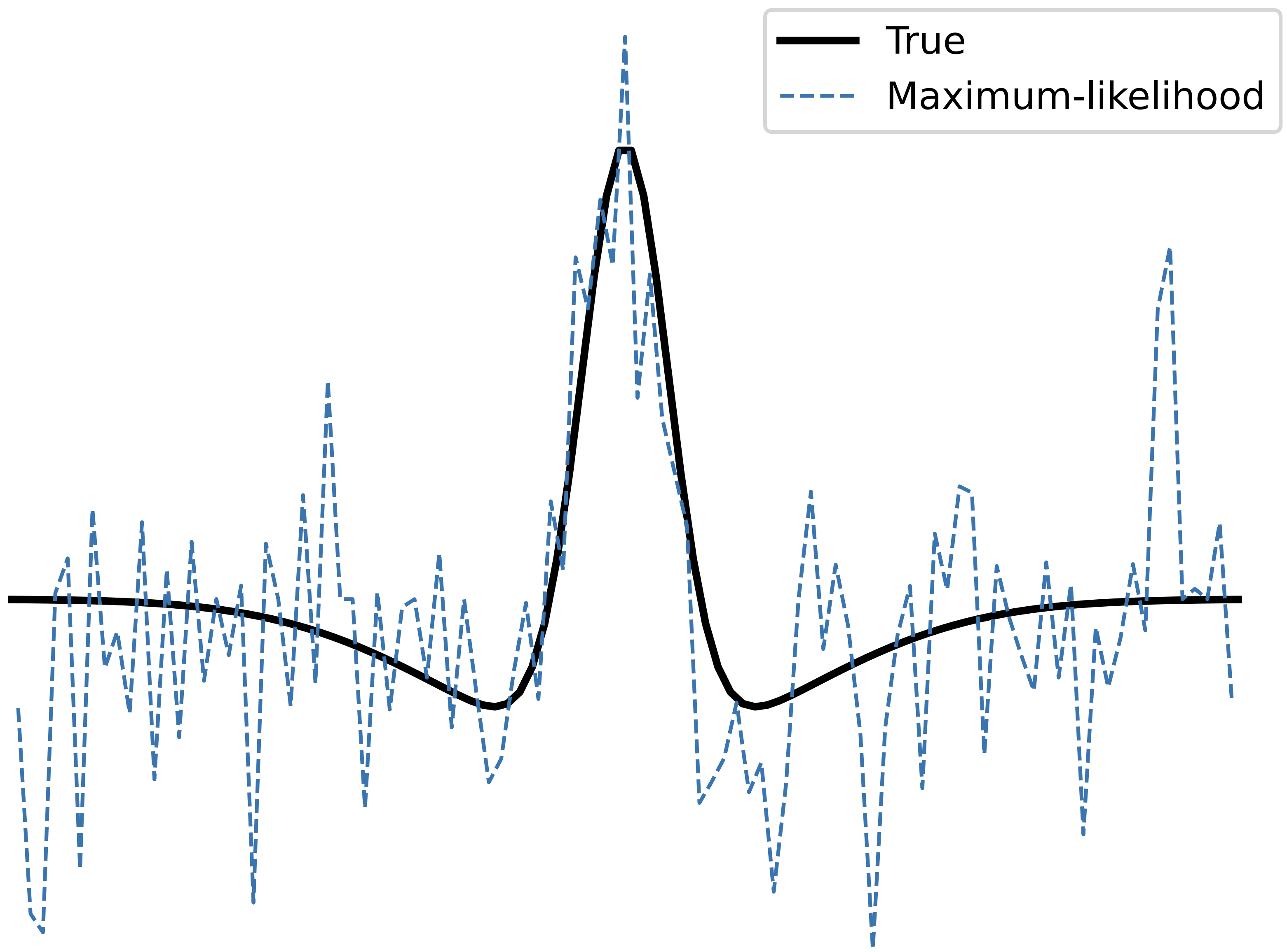


An Example: Grid Cell Modeling



Overfitting

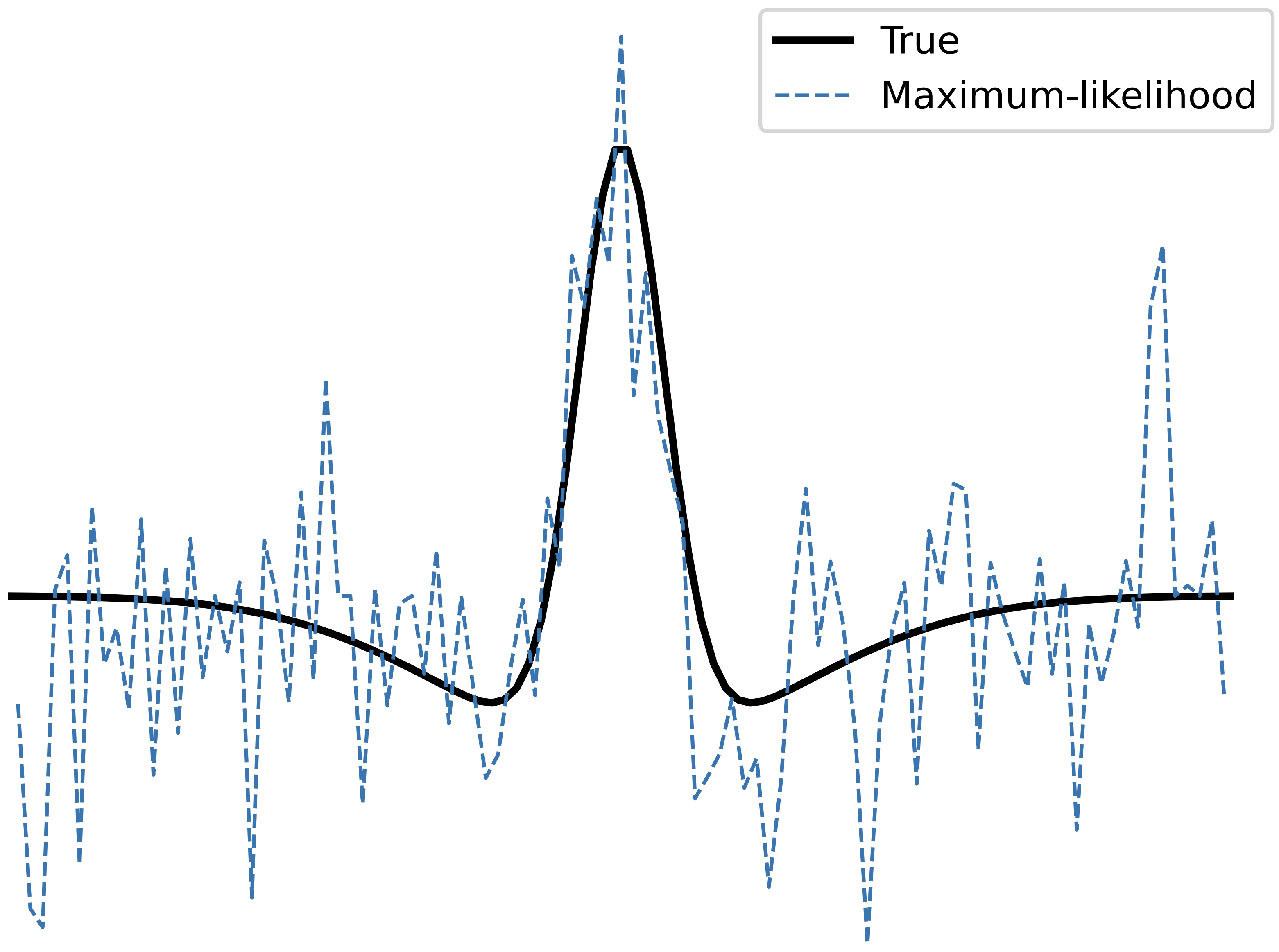
Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w})$



Overfitting

Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w})$

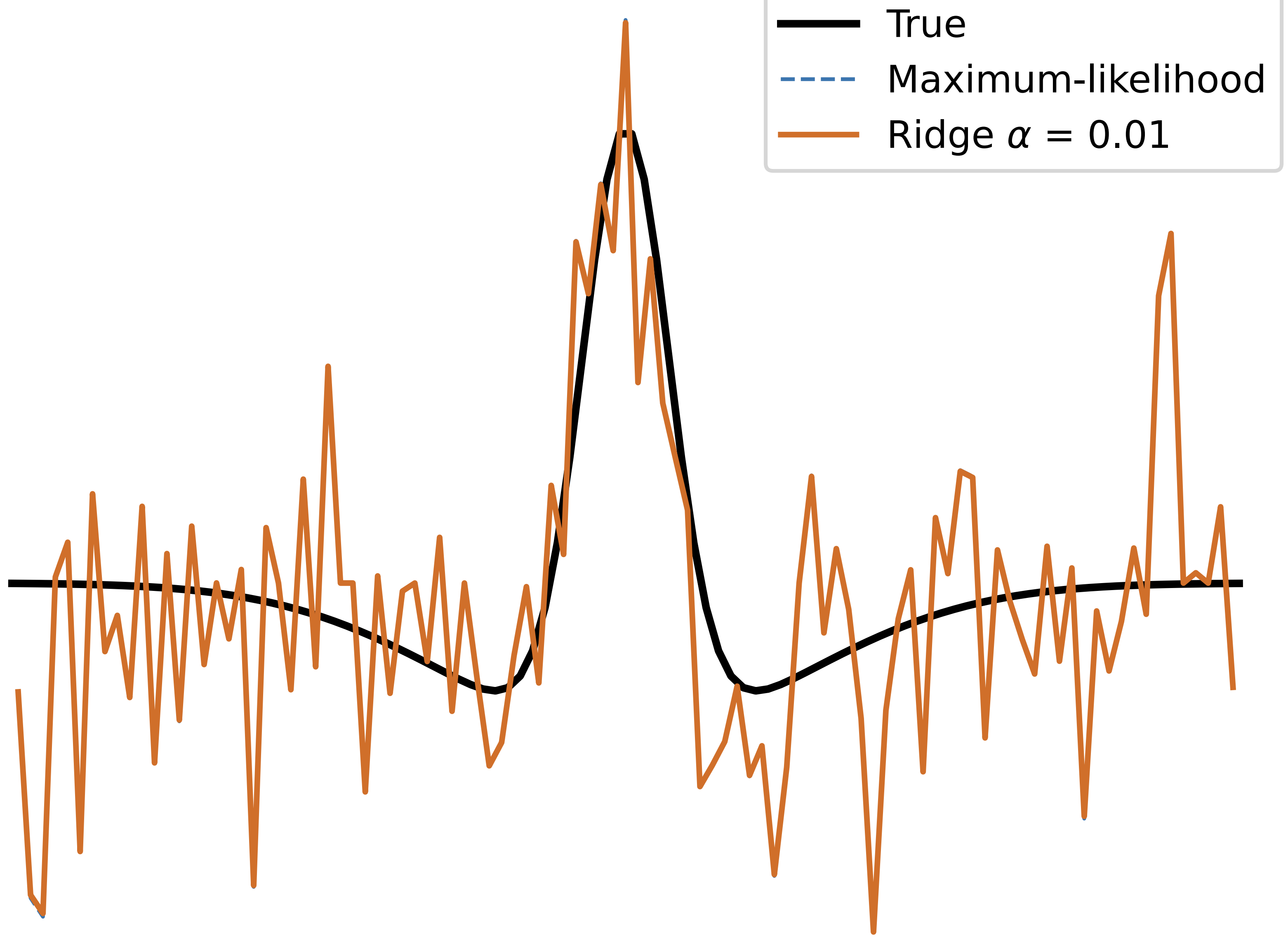
Ridge (L2): $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w}) - \alpha\,(w_1^2 + \ldots. + w_n^2)$



Overfitting

Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w})$

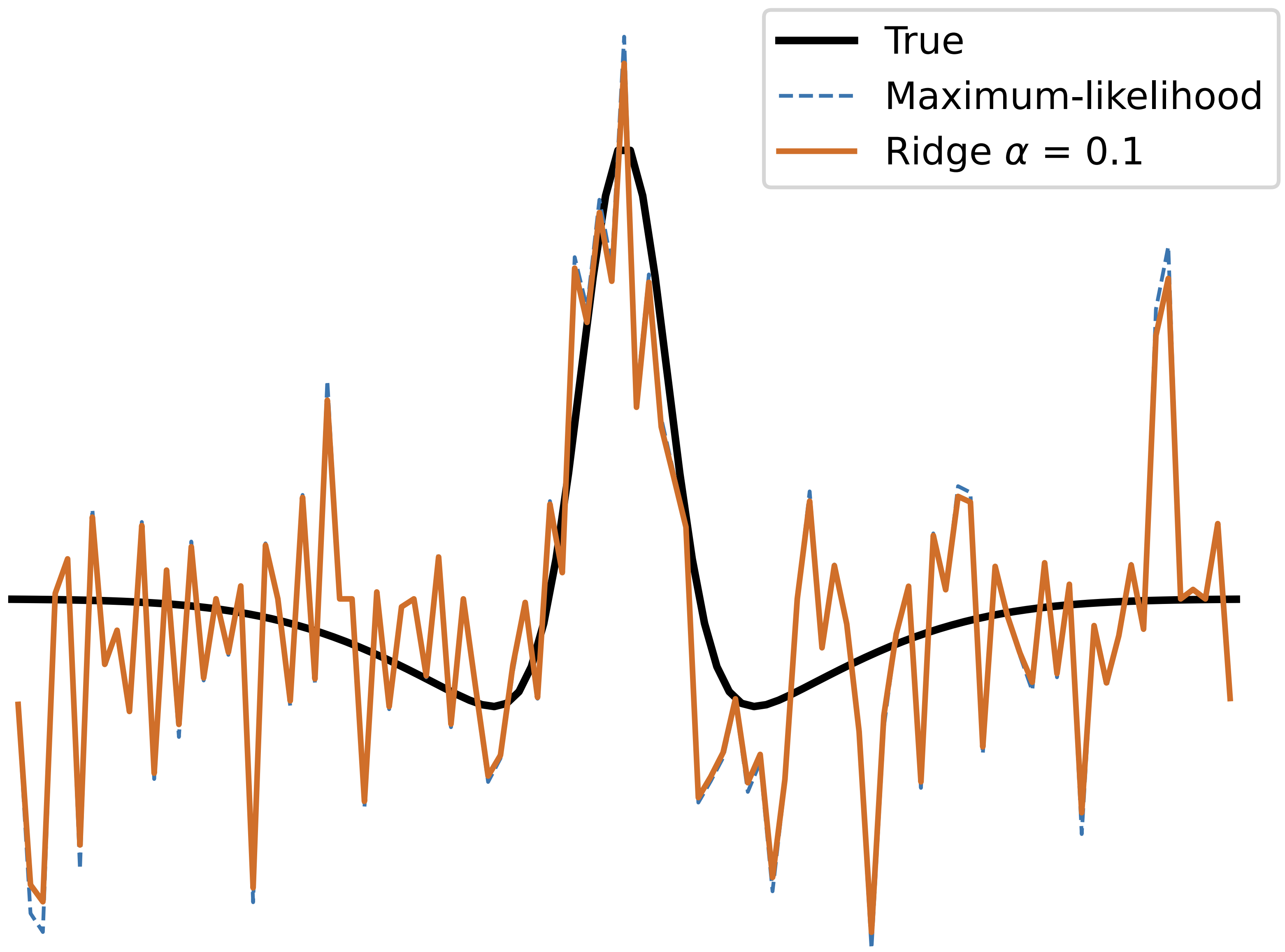
Ridge (L2): $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w}) - \alpha\,(w_1^2 + \ldots. + w_n^2)$



Overfitting

Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w})$

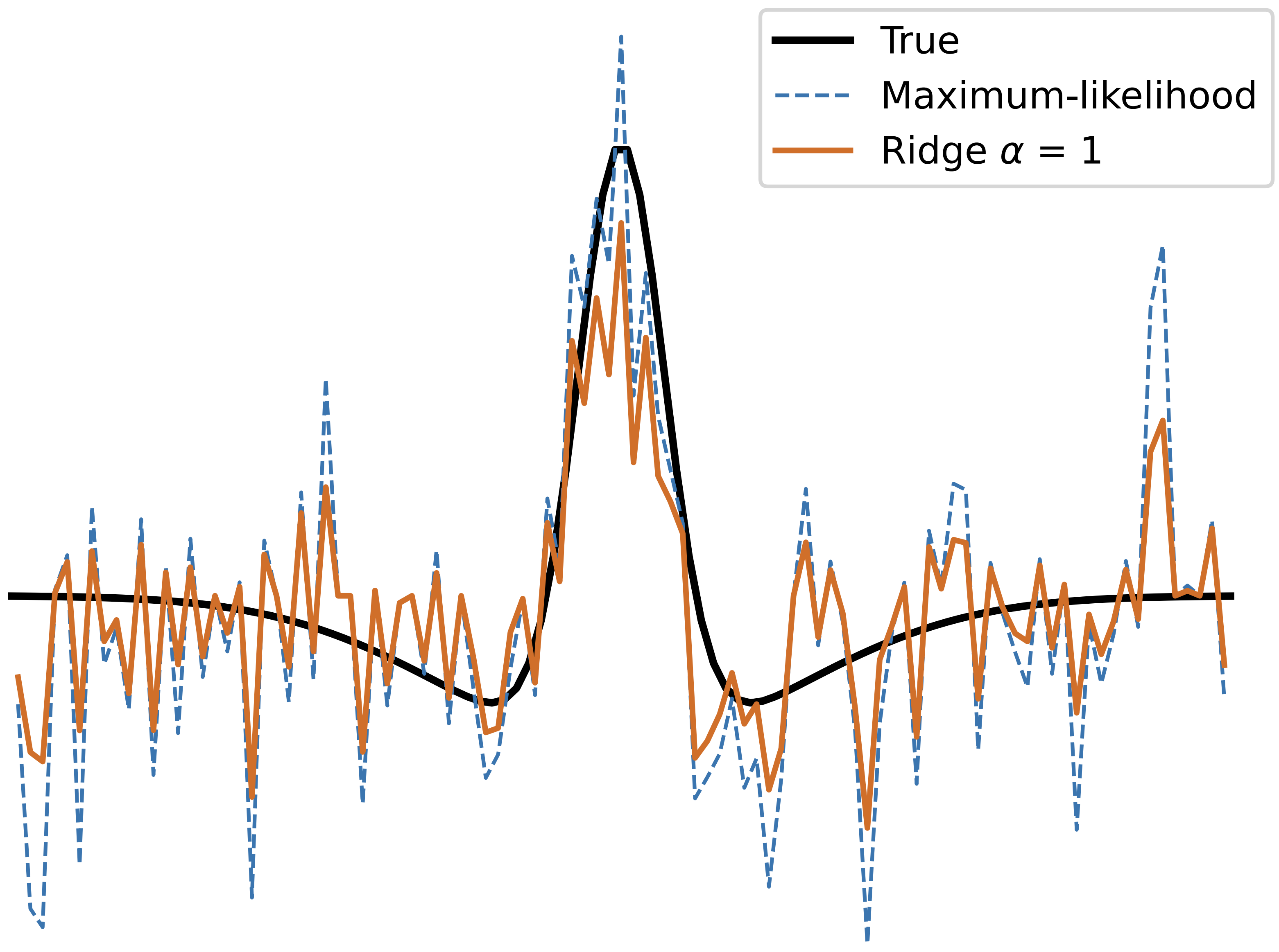
Ridge (L2): $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w}) - \alpha\,(w_1^2 + \ldots. + w_n^2)$



Overfitting

Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \,|\, \mathbf{X}, \mathbf{w})$

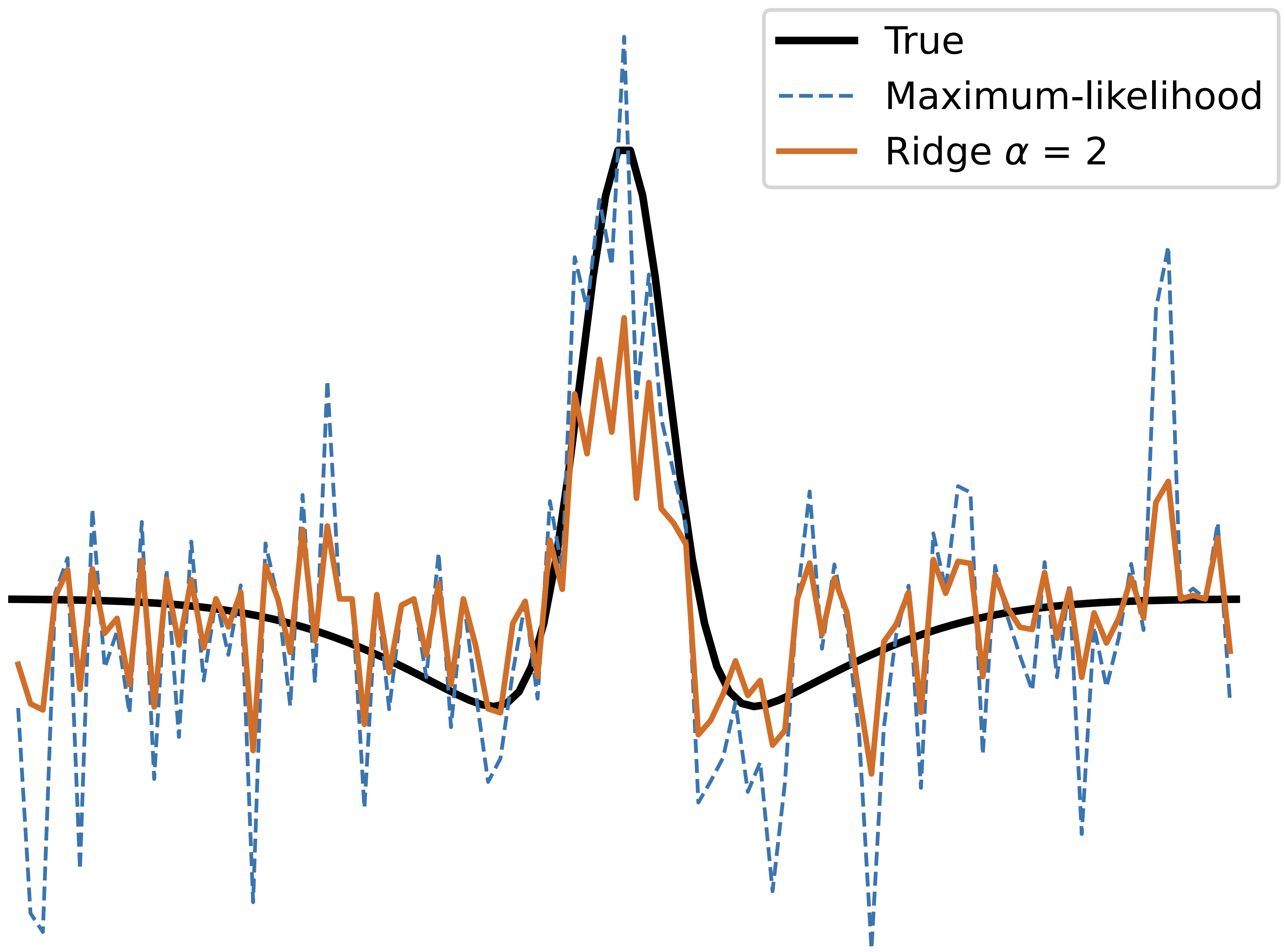
Ridge (L2): $\max_{\mathbf{w}} \log p(\text{counts} \,|\, \mathbf{X}, \mathbf{w}) - \alpha\,(w_1^2 + \ldots. + w_n^2)$



Overfitting

Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w})$

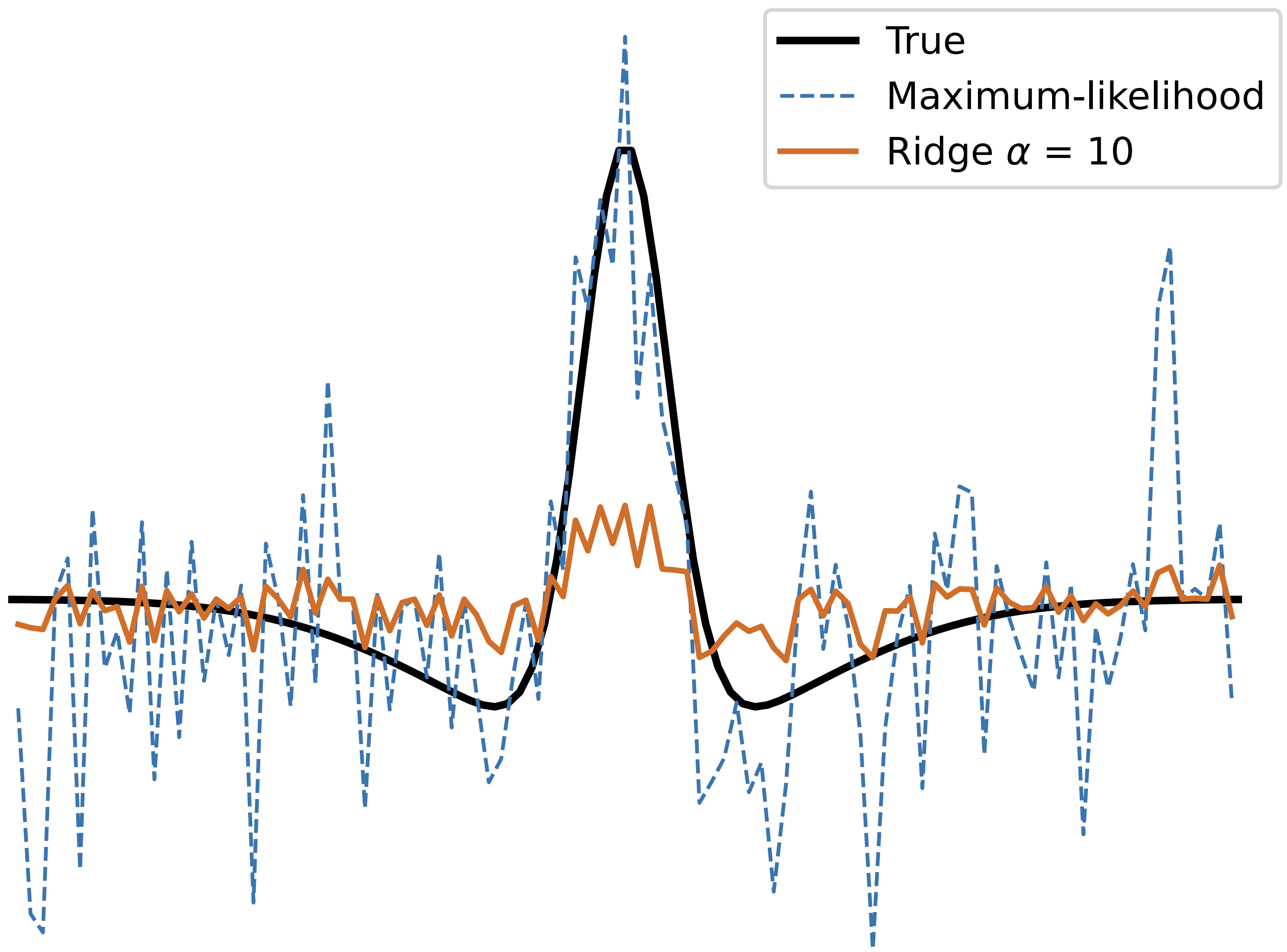
Ridge (L2): $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w}) - \alpha \, (w_1^2 + \ldots . + w_n^2)$



Overfitting

Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w})$

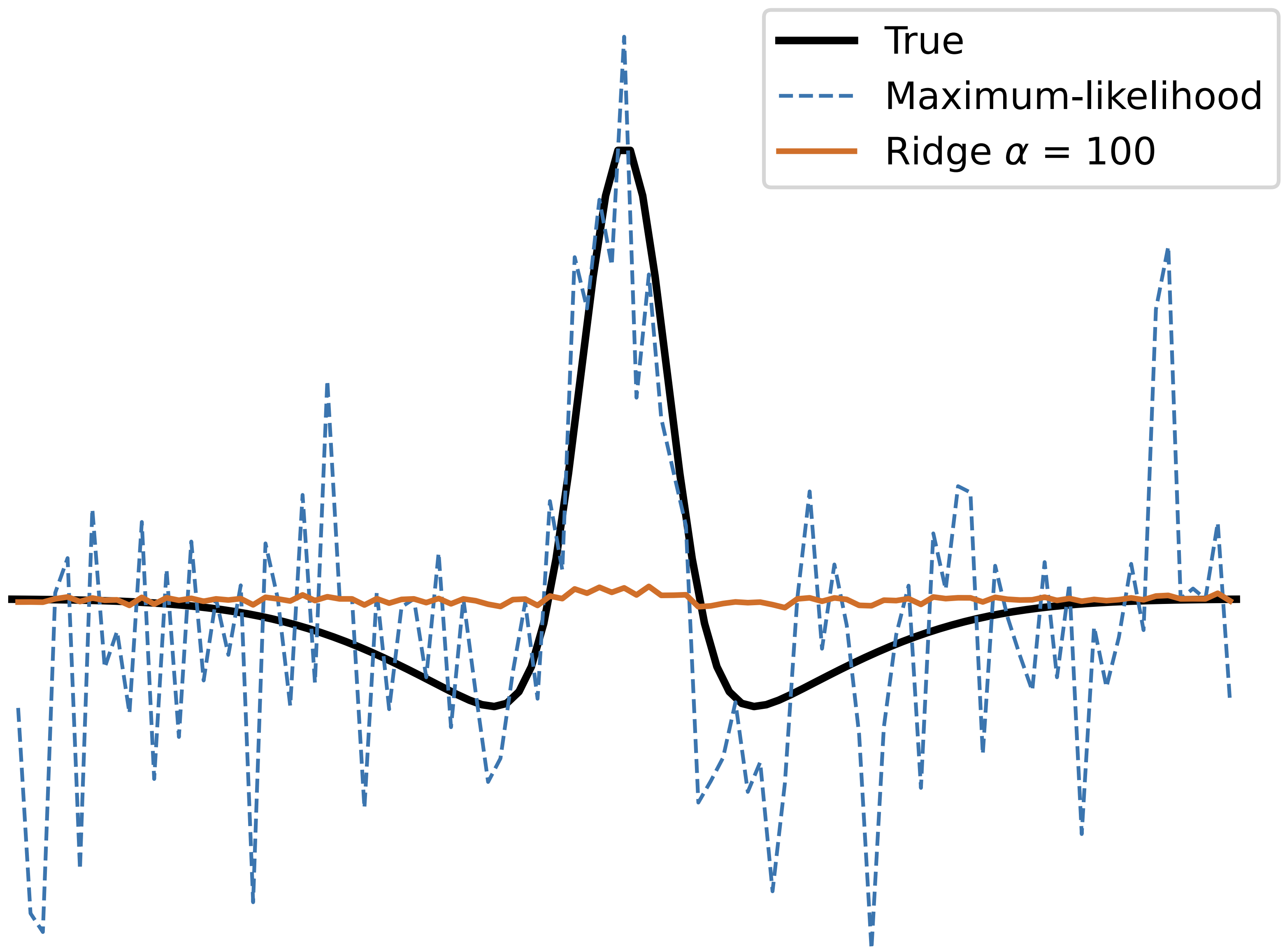
Ridge (L2): $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w}) - \alpha\,(w_1^2 + \ldots. + w_n^2)$



Overfitting

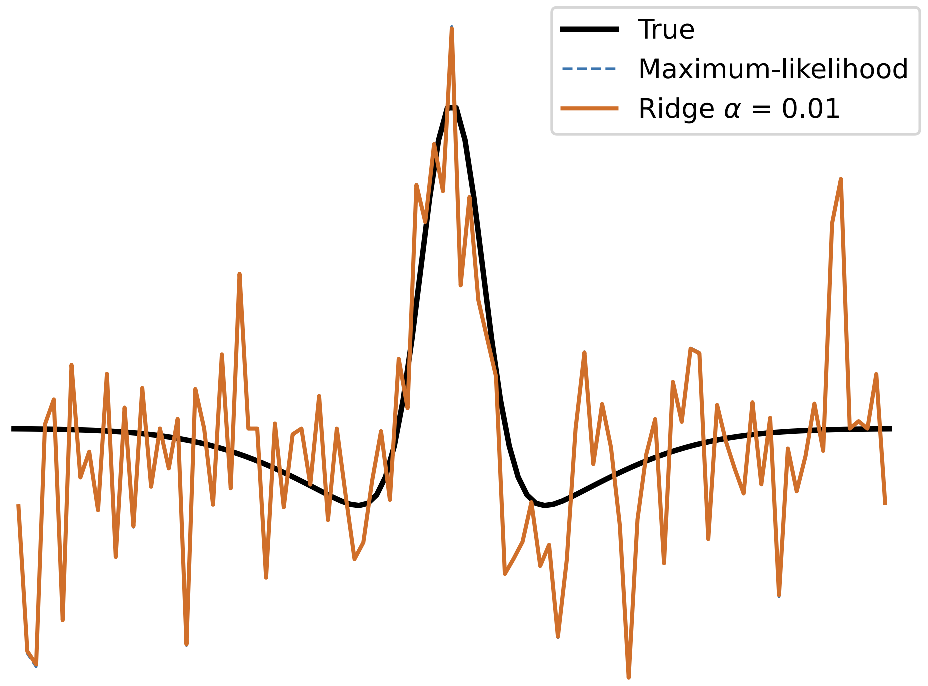
Maximum-Likelihood: $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w})$

Ridge (L2): $\max_{\mathbf{w}} \log p(\text{counts} \mid \mathbf{X}, \mathbf{w}) - \alpha \left(w_1^2 + \ldots . + w_n^2\right)$

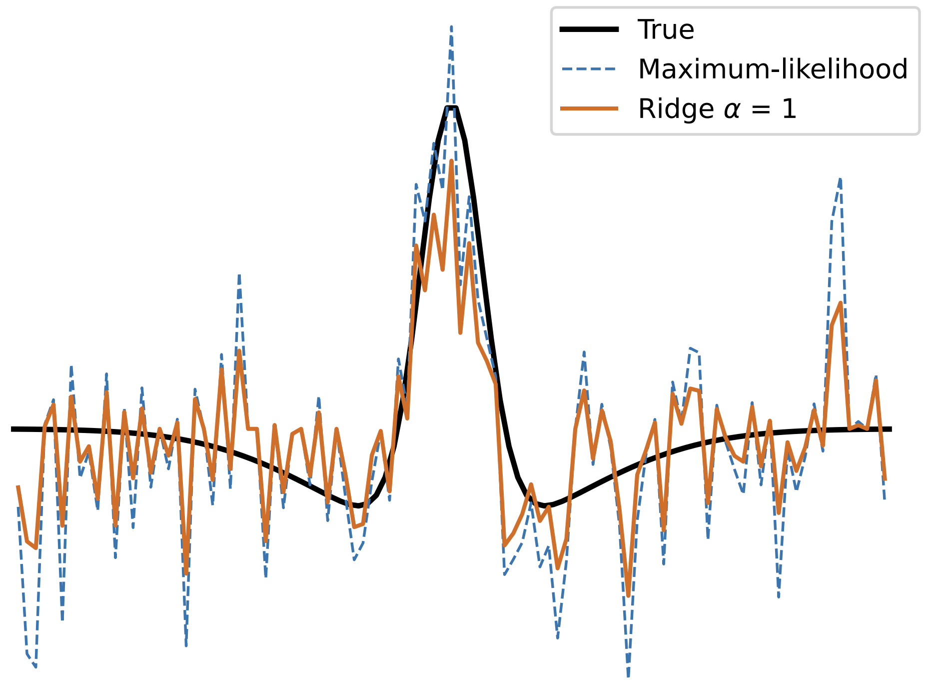


Overfitting

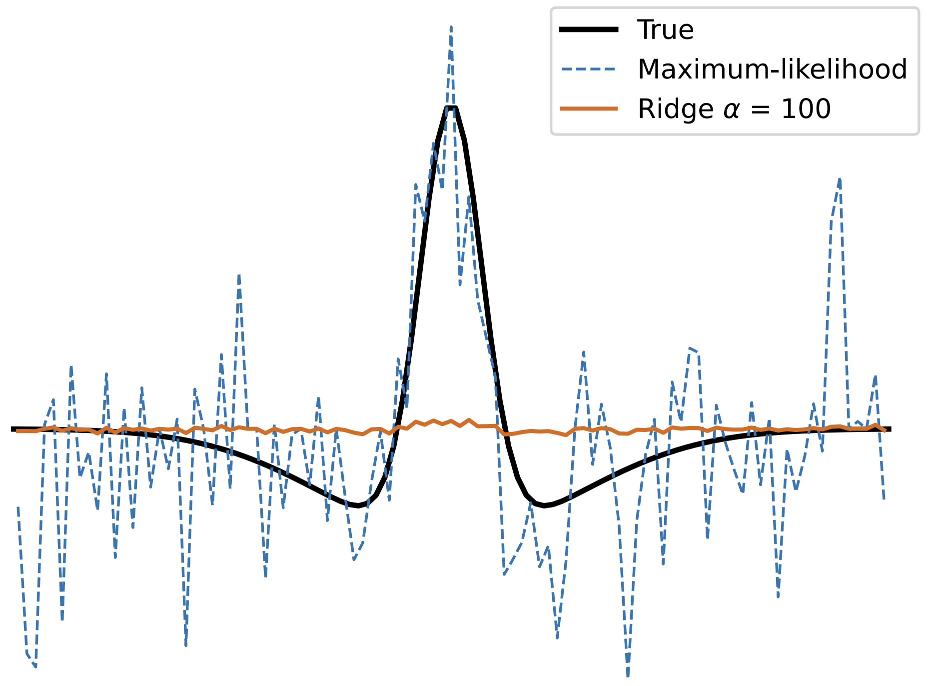
$\alpha = 0.01$



$\alpha = 1$



$\alpha = 100$



How do you select α?

- **Cross-validation**: learn the weights on a subset of your data, test the model on another subset.

Other regularization approaches in NeMoS:

- **Lasso (L1):** Shrink coefficients to zero, enforcing sparsity.
  Feature selection.

- **Group Lasso:** Shrink groups of coefficients to zero.
  Group feature selection.

Summary

- Tuning functions do not fully characterize neural encoding.

Summary

- Tuning functions do not fully characterize neural encoding.
- GLMs retain many of the advantageous properties of linear regression *(easy to fit, unique solution)*

Summary

- Tuning functions do not fully characterize neural encoding.
- GLMs retain many of the advantageous properties of linear regression *(easy to fit, unique solution)*
- Better suited for non-normally distributed data.

Summary

- Tuning functions do not fully characterize neural encoding.
- GLMs retain many of the advantageous properties of linear regression *(easy to fit, unique solution)*
- Better suited for non-normally distributed data.
- Rich framework: model jointly many features, flexible design, regularization…

Today’s roadmap

1. Current injection notebook:
   - Fit an LNP model to intracellular recordings from the Allen Brain Map.
   - Capture temporal effects using NeMoS’ basis.

Today's roadmap

1. Current injection notebook:

    - Fit an LNP model to intracellular recordings from the Allen Brain Map.
    - Capture temporal effects using NeMoS' basis.

2. Head direction notebook:

    - Functional connectivity analysis of neurons in the anterodorsal thalamic nucleus (ADn).

Today’s roadmap

1. Current injection notebook:

    - Fit an LNP model to intracellular recordings from the Allen Brain Map.

    - Capture temporal effects using NeMoS’ basis.

2. Head direction notebook:

    - Functional connectivity analysis of neurons in the anterodorsal thalamic nucleus (ADn).

3. Advanced topics: regularization and model selection.

Today's roadmap

1. Current injection notebook:

    - Fit an LNP model to intracellular recordings from the Allen Brain Map.
    - Capture temporal effects using NeMoS' basis.

2. Head direction notebook:

    - Functional connectivity analysis of neurons in the anterodorsal thalamic nucleus (ADn).

3. Advanced topics: regularization and model selection.

4. Work on your own data (or keep going with your favorite notebook).