Machine Learning Methods for Neural Data Analysis

Lecture 9: Encoding RGCs Part II

Scott Linderman

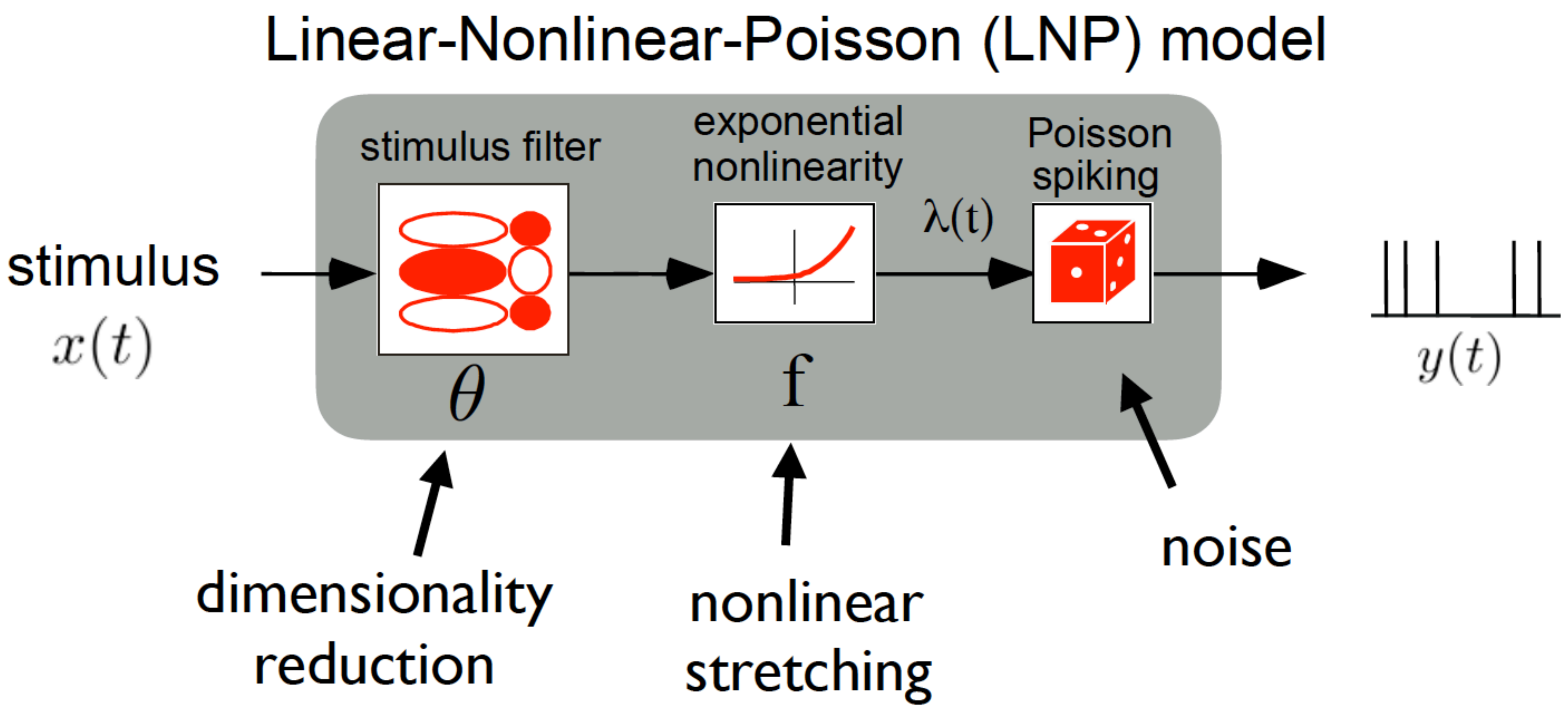
STATS 220/320 (NBIO220, CS339N). Winter 2021.

Agenda

1. More generalized linear models
2. Incorporating interactions between neurons
3. Point processes: the continuous time limit

Encoding models of RGC responses

Basic linear-nonlinear-Poisson (LNP) model



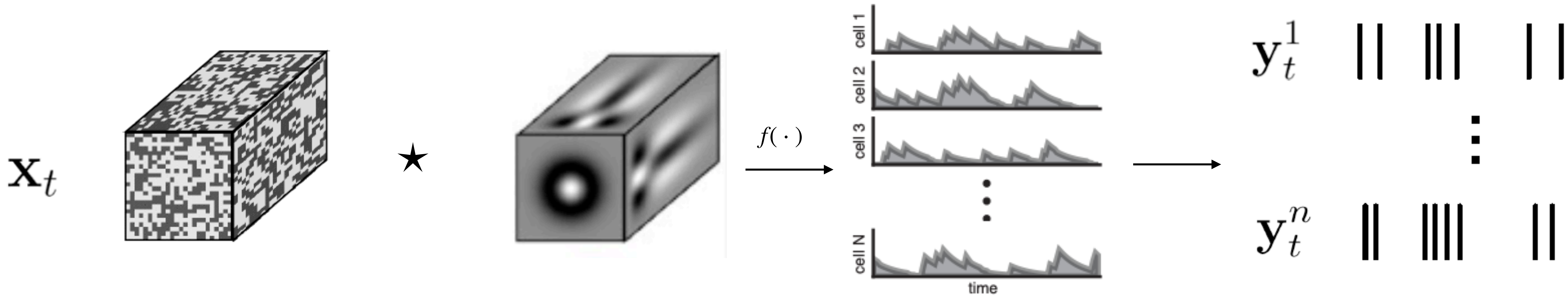
In statistics, we call this a generalized linear model (GLM).

Slides from Jonathan Pillow

Encoding models

Generalized linear models

Stimulus **Filter Weights** **Firing Rates** **Neural Recording**



W_n

$$\lambda_{nt} = f([X \star W_n]_t)$$

Slides from Jonathan Pillow

Encoding models of RGC responses

Generalized linear models

- Let $X \in \mathbb{R}^{P_H \times P_W \times T/\Delta t}$ denote a stimulus movie and $Y \in \mathbb{N}^{N \times T/\Delta t}$ denote the resulting spike train, where Δt is the bin size and T is the duration of recording.
- Let $W \in \mathbb{R}^{N \times P_H \times P_W \times D}$ denote the weights of the GLM.
- **Poisson GLM**

$$\lambda_{n,t} = f\left(\sum_{i=1}^{P_H} \sum_{j=1}^{P_W} \sum_{d=1}^{D} x_{i,j,t-d} w_{nijd} \right) = f([X \star W_n]_t) = f(\langle \tilde{X}_t, W_n \rangle)$$

$$y_{n,t} \sim \mathrm{Po}(\lambda_{n,t}\, \Delta t)$$

Encoding models of RGC responses

Generalized linear models

- Estimate the weights W by maximizing the likelihood,

$$
\begin{aligned}
\mathscr{L}(W) &= \sum_{n=1}^{N} \sum_{t=1}^{T/\Delta t} \log p(y_{n,t} \mid \lambda_{n,t}) \\
&= \sum_{n=1}^{N} \sum_{t=1}^{T/\Delta t} - \lambda_{n,t}\,\Delta t + y_{n,t} \log \lambda_{n,t}\,\Delta t + \mathrm{c} \\
&= \sum_{n=1}^{N} \sum_{t=1}^{T/\Delta t} - f(\langle \tilde{X}_t, W_n \rangle)\Delta t + y_{n,t} \log f(\langle \tilde{X}_t, W_n \rangle) + \mathrm{c} \\
&= \sum_{n=1}^{N} \sum_{t=1}^{T/\Delta t} - \exp\left\{\langle \tilde{X}_t, W_n \rangle\right\}\Delta t + y_{n,t}\langle \tilde{X}_t, W_n \rangle + \mathrm{c} \qquad \text{when } f(a) = e^a
\end{aligned}
$$

Encoding models of RGC responses

Generalized linear models

- Estimate the weights W by maximizing the likelihood,

$$
\begin{aligned}
\nabla_{W_n} \mathscr{L}(W) &= \sum_{t=1}^{T/\Delta t} \left(y_{n,t} - \exp\left\{ \langle \tilde{X}_t, W_n \rangle \right\} \Delta t \right) \tilde{X}_t \\
&= \sum_{t=1}^{T/\Delta t} \left(y_{n,t} - f(\langle \tilde{X}_t, W_n \rangle) \Delta t \right) \tilde{X}_t \\
&= \sum_{t=1}^{T/\Delta t} \left(y_{n,t} - \mathbb{E}[y_{n,t}] \right) \tilde{X}_t
\end{aligned}
$$

- Recall **logistic regression**; we had the same form!
- This is a property of GLMs with “canonical” link functions.

Encoding models of RGC responses

Generalized linear models

- Does gradient ascent converge to a global optimum?

$$\nabla^2_{W_n} \mathscr{L}(W) = \sum_{t=1}^{T/\Delta t} \nabla_{W_n} \left(y_{n,t} - \exp\left\{ \langle \tilde{X}_t, W_n \rangle \right\} \Delta t \right) \tilde{X}_t$$

$$= - \sum_{t=1}^{T/\Delta t} \exp\left\{ \langle \tilde{X}_t, W_n \rangle \right\} \Delta t \cdot \tilde{X}_t \otimes \tilde{X}_t$$

- Here, $\otimes$ denotes an outer product between two tensors.

- The Hessian is negative definite $\Longrightarrow \mathscr{L}(W)$ is concave $\Longrightarrow$ gradient ascent will reach a global optimum.

Encoding models of RGC responses

Factoring the weights

- How big are the weights in the previous model?
- Instead, factor the weights into a spatial and a temporal component,

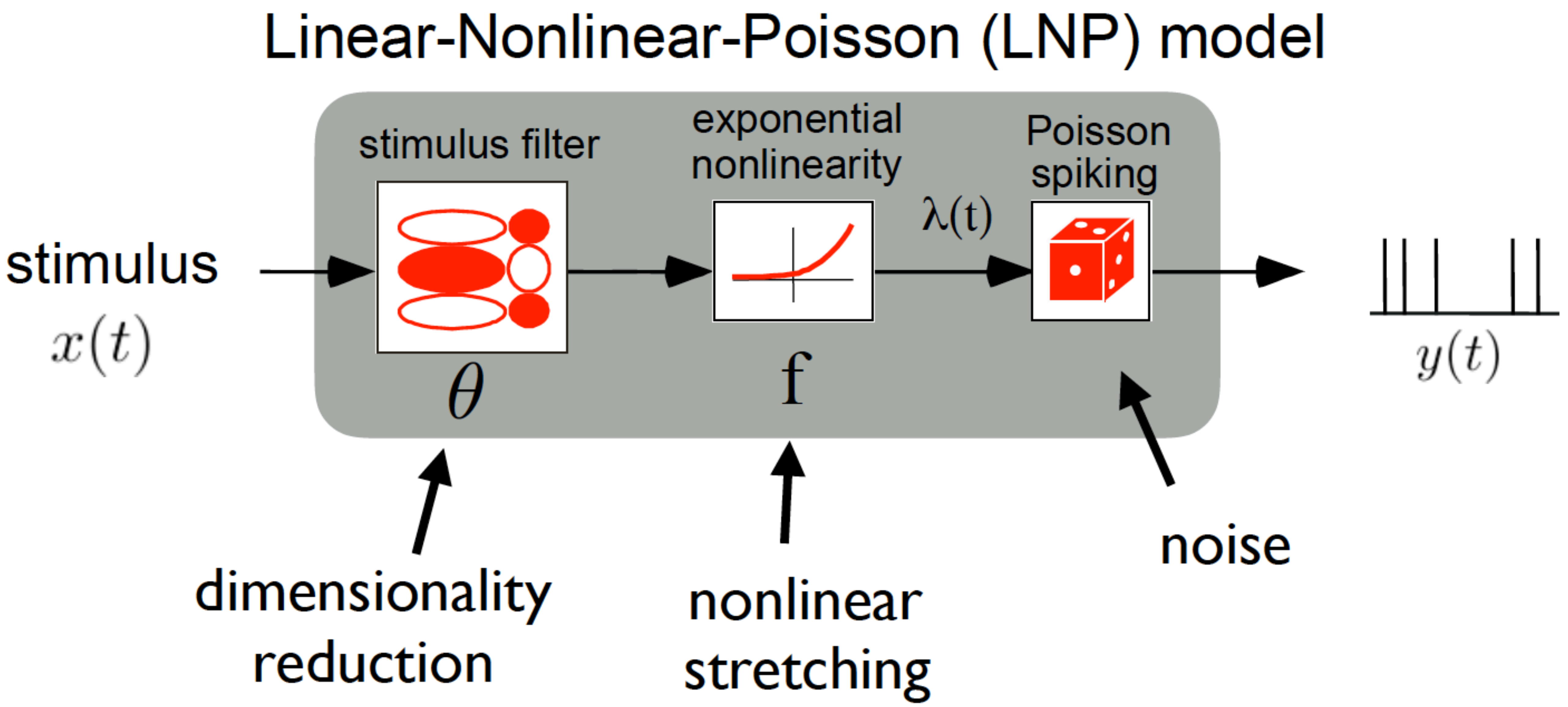
$$W_n = U_n \otimes V_n$$

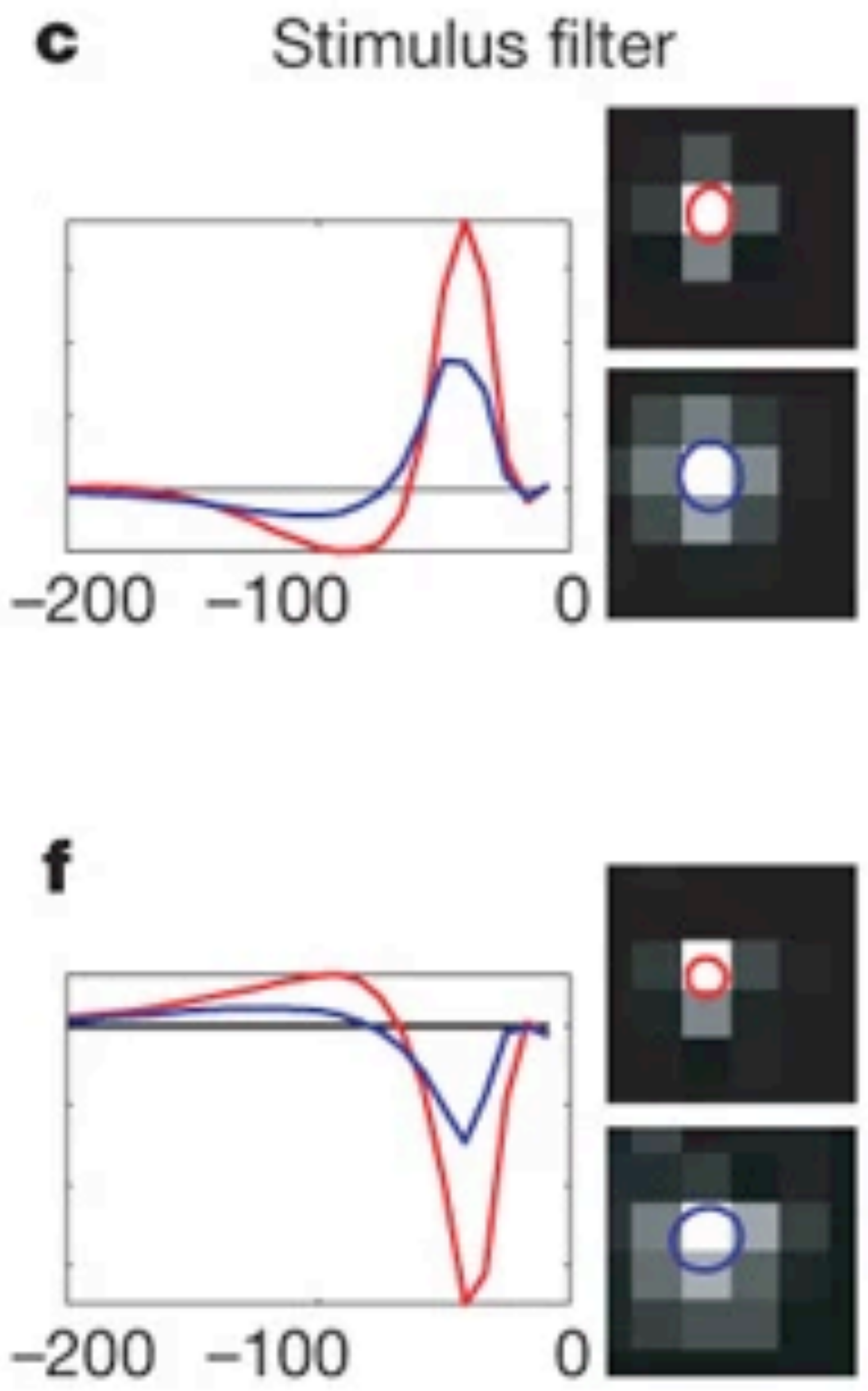
- where
  - $U_n \in \mathbb{R}^{P_H \times P_W}$ is the spatial footprint (receptive field)
  - $V_n \in \mathbb{R}^D$ is the temporal response

As with the convolutional spike sorting models from Lab 2, $X \star W_n = \underbrace{\langle U_n, X \rangle}_{\in \mathbb{R}^T} \star \underbrace{V_n}_{\in \mathbb{R}^D}$

Encoding models of RGC responses

Adding interactions between neurons





Pillow et al (*Nature*, 2008)

Encoding models of RGC responses

Going deeper

- The GLM is elegant in its simplicity, but the **linearity assumption** is still restrictive.
- We can use **hand-crafted nonlinear features** (i.e. replace X with $\phi(X)$), but can we do better?
- The deep learning mantra is that given enough data, we can **learn features**.
- Here, the learned features may **mimic intermediate layer cells.**
    - The first few layers of the retina look a lot like a convolutional network.

Encoding models of RGC responses

Going deeper

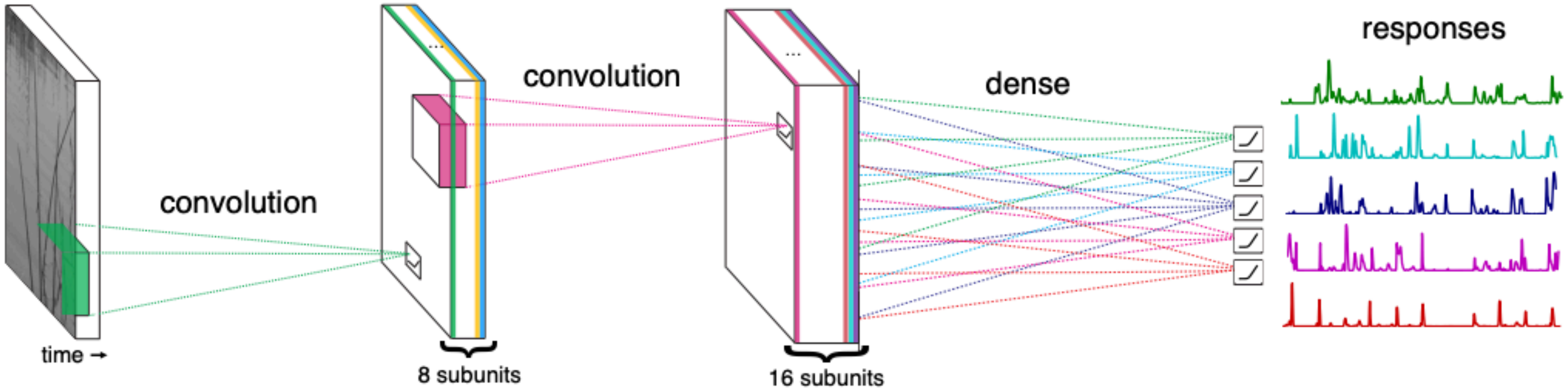


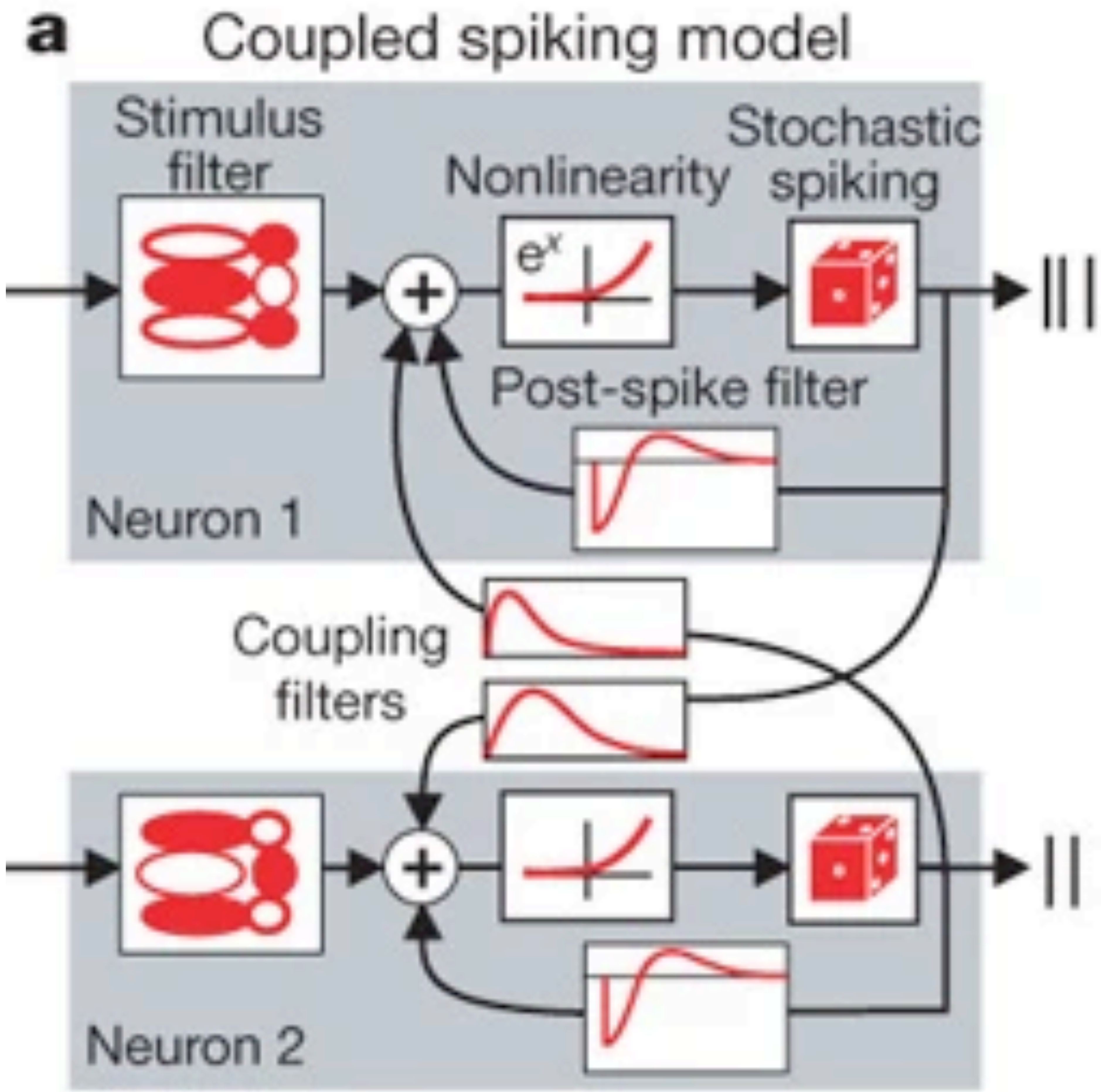
Figure 1: A schematic of the model architecture. The stimulus was convolved with 8 learned spatiotemporal filters whose activations were rectified. The second convolutional layer then projected the activity of these subunits through spatial filters onto 16 subunit types, whose activity was linearly combined and passed through a final soft rectifying nonlinearity to yield the predicted response.

McIntosh et al (*NeurIPS*, 2016)

Incorporating interactions between neurons

Incorporating interactions between neurons

Coupled GLMs



Pillow et al (*Nature*, 2008)

Encoding models of RGC responses

Yet another convolutional model…

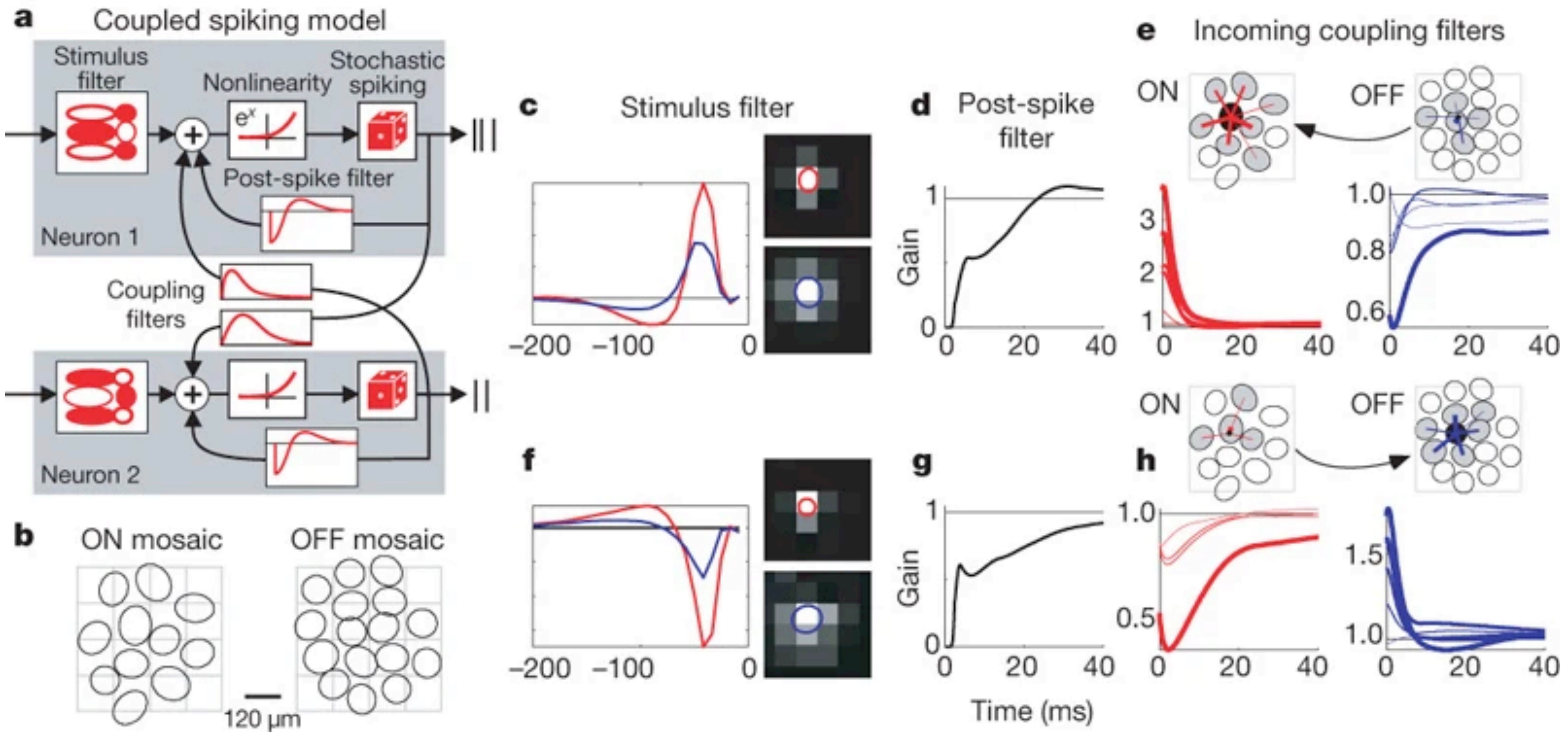
$$\lambda_{n,t} = f\left(\sum_{i=1}^{P_H}\sum_{j=1}^{P_W}\sum_{d=1}^{D} x_{i,j,t-d} w_{n,i,j,d} + \sum_{m=1}^{N}\sum_{d=1}^{D} y_{m,t-d} g_{n,m,d}\right)$$

$$= f\left([X \star W_n]_t + [Y \star G_n]_t\right)$$

- $g_{n,n} \in \mathbb{R}^D$ are the self-coupling weights. They can implement refractory periods, e.g.
- $g_{n,m} \in \mathbb{R}^D$ capture the directed influence of neuron m on neuron n.
- (Didn't I say it's all convolutional models in the end?)

Incorporating interactions between neurons

Results from an application to 27 ON/OFF RGCs



Pillow et al (*Nature*, 2008)

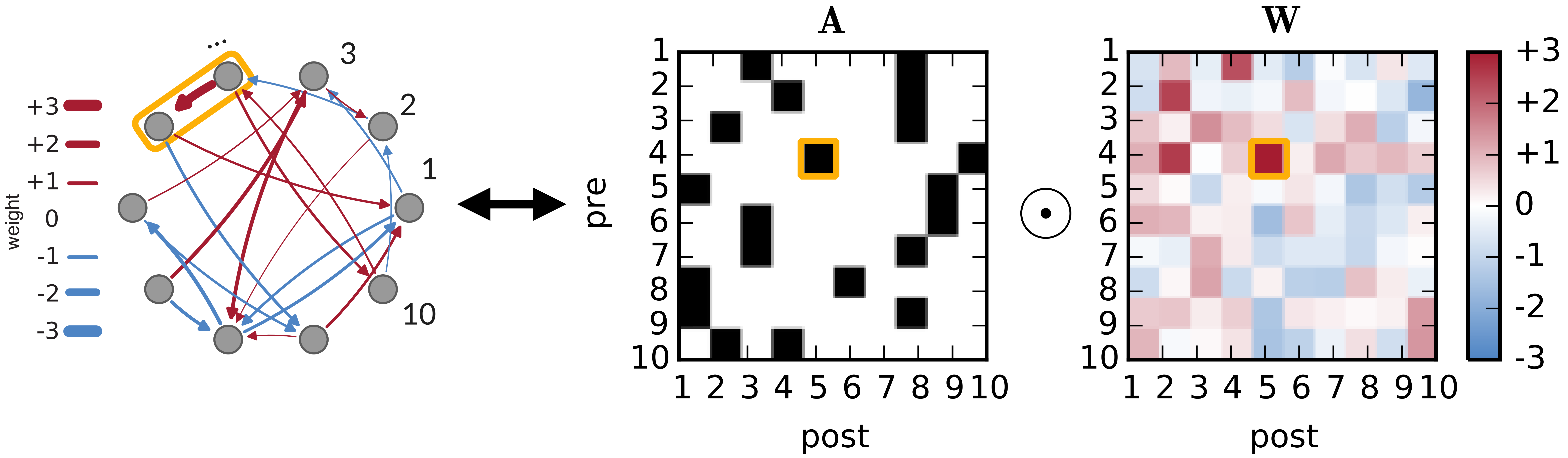
Incorporating interactions between neurons

Parameter explosion

- **Question:** How does the number of parameters grow with the number of neurons?
- In 2008, twenty seven RGCs was a fair number.
- Nowadays, we can record orders of magnitude more neurons simultaneously.
    - E.g. Stringer et al (*Nature,* 2019): 10,000 cells in visual cortex.
- We need either more data, a different model, or constraints on model parameters to reduce complexity.

Incorporating interactions between neurons

Separately modeling the interaction sparsity and weights



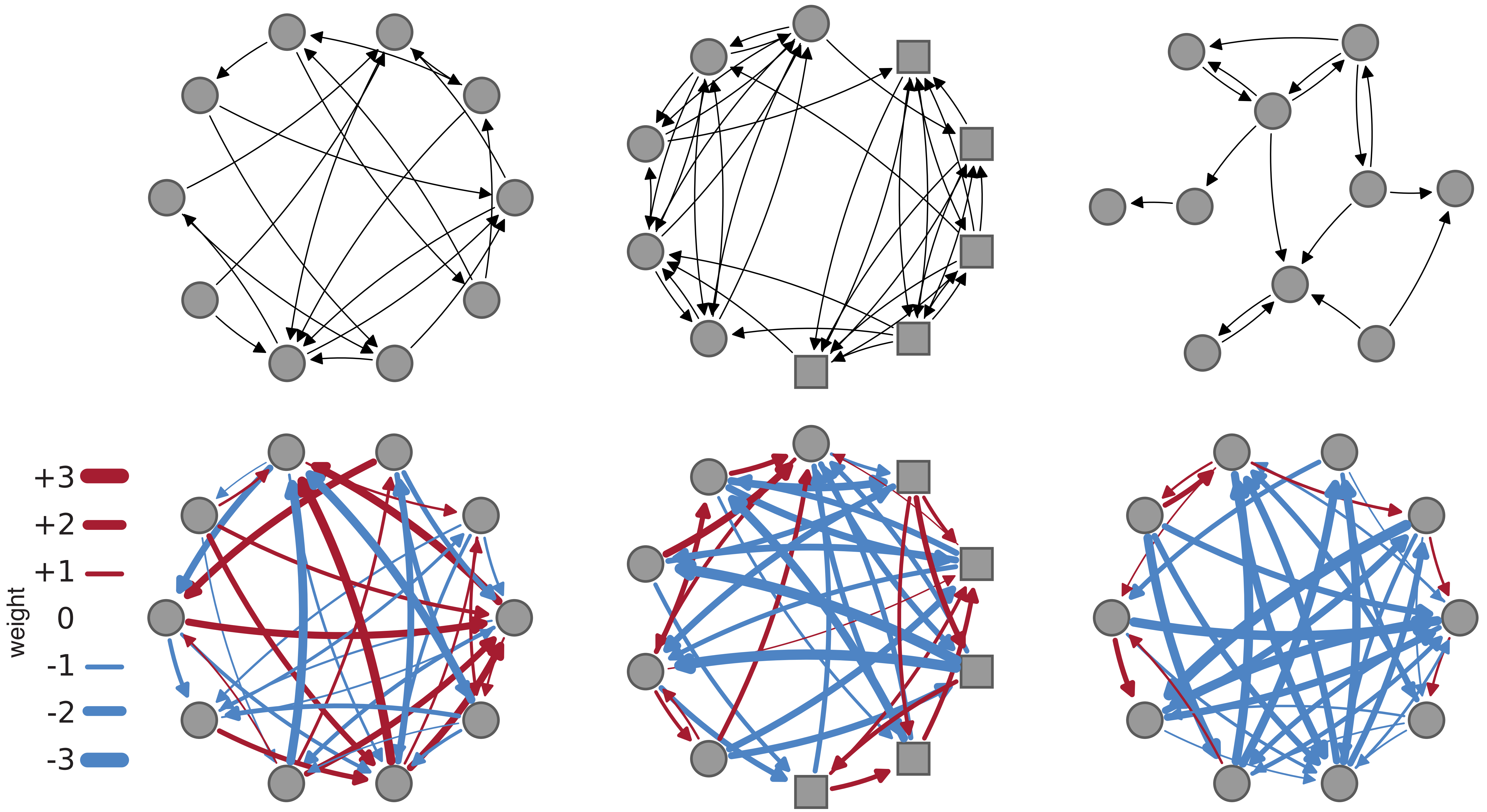
Binary Adjacency Matrix

Real-Valued Weight Matrix

Encoding models of RGC responses

Latent variable models for networks

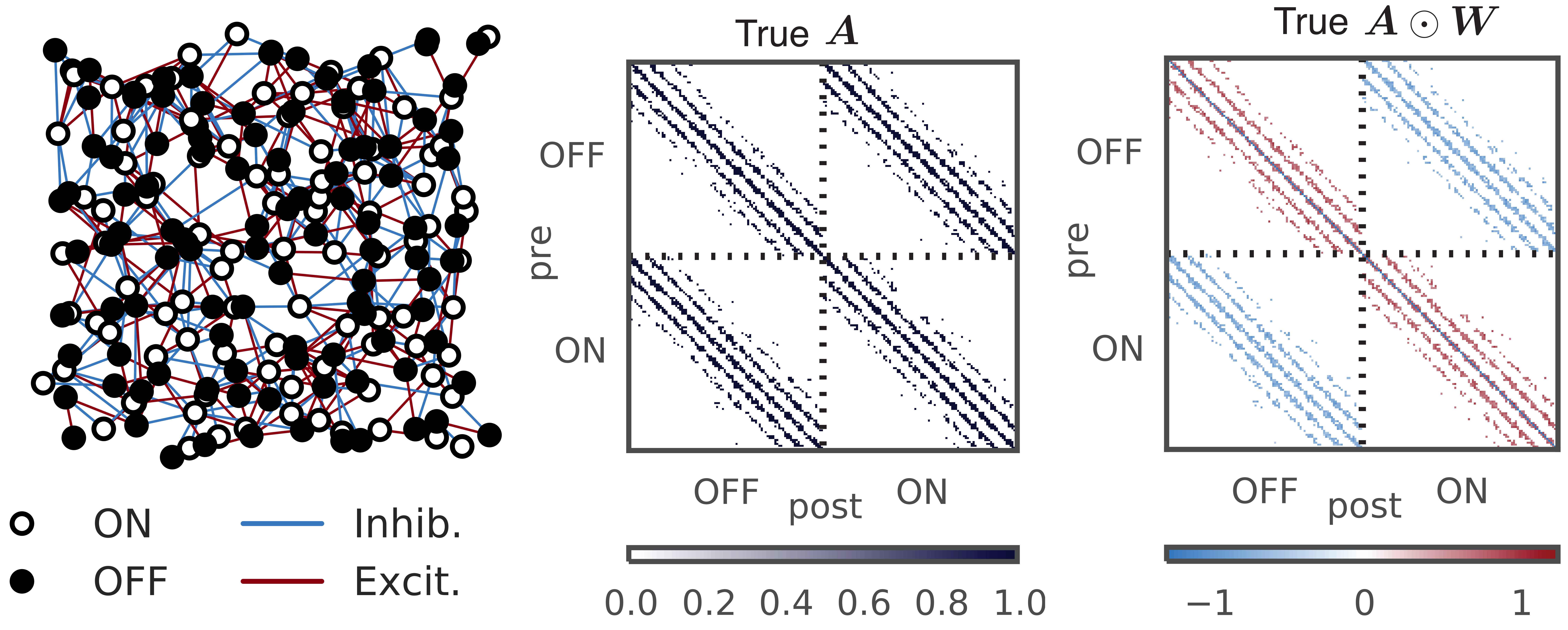
	Independent Edge Model	Stochastic Block Model (SBM)	Latent Distance Model
Types or Features Z	None	Type { ■, ● }	Location $\mathbb{R}^D$
Adjacency Models A			
Weight Models W			



Hoff (*NeurIPS*, 2007)

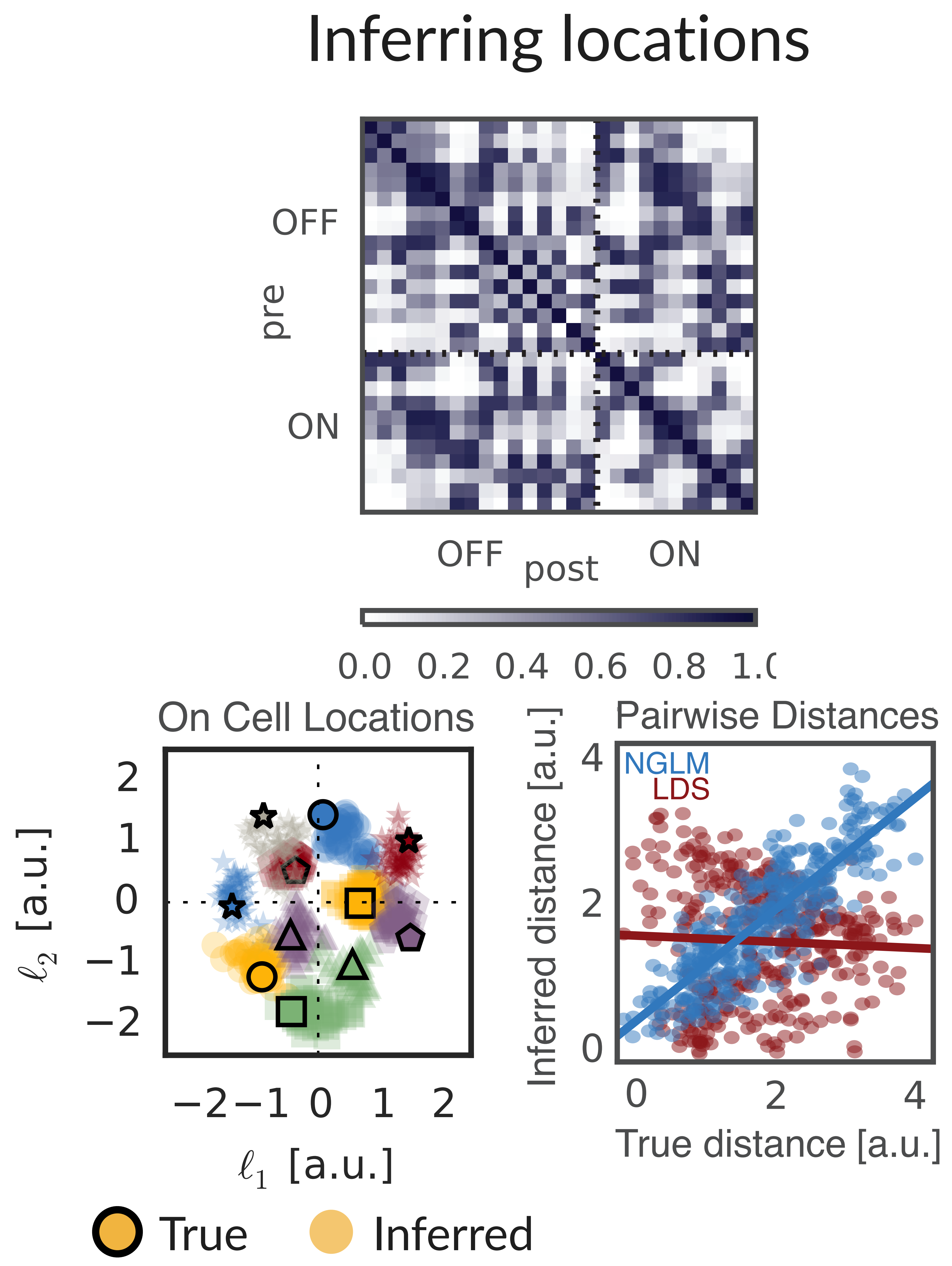
Example: a synthetic retina

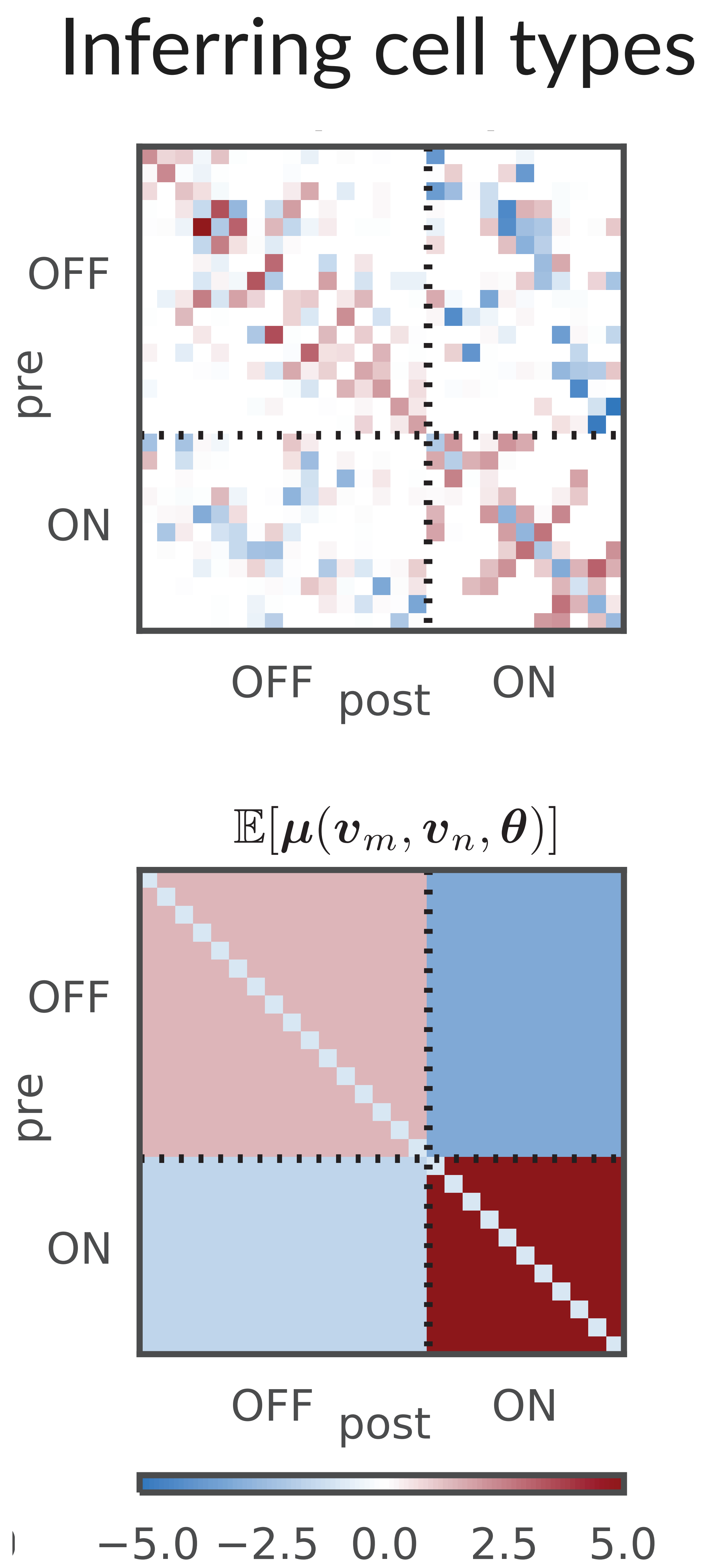
Latent variables: types & locations. **Adjacency:** distance-dependent. **Weights:** type-dependent.

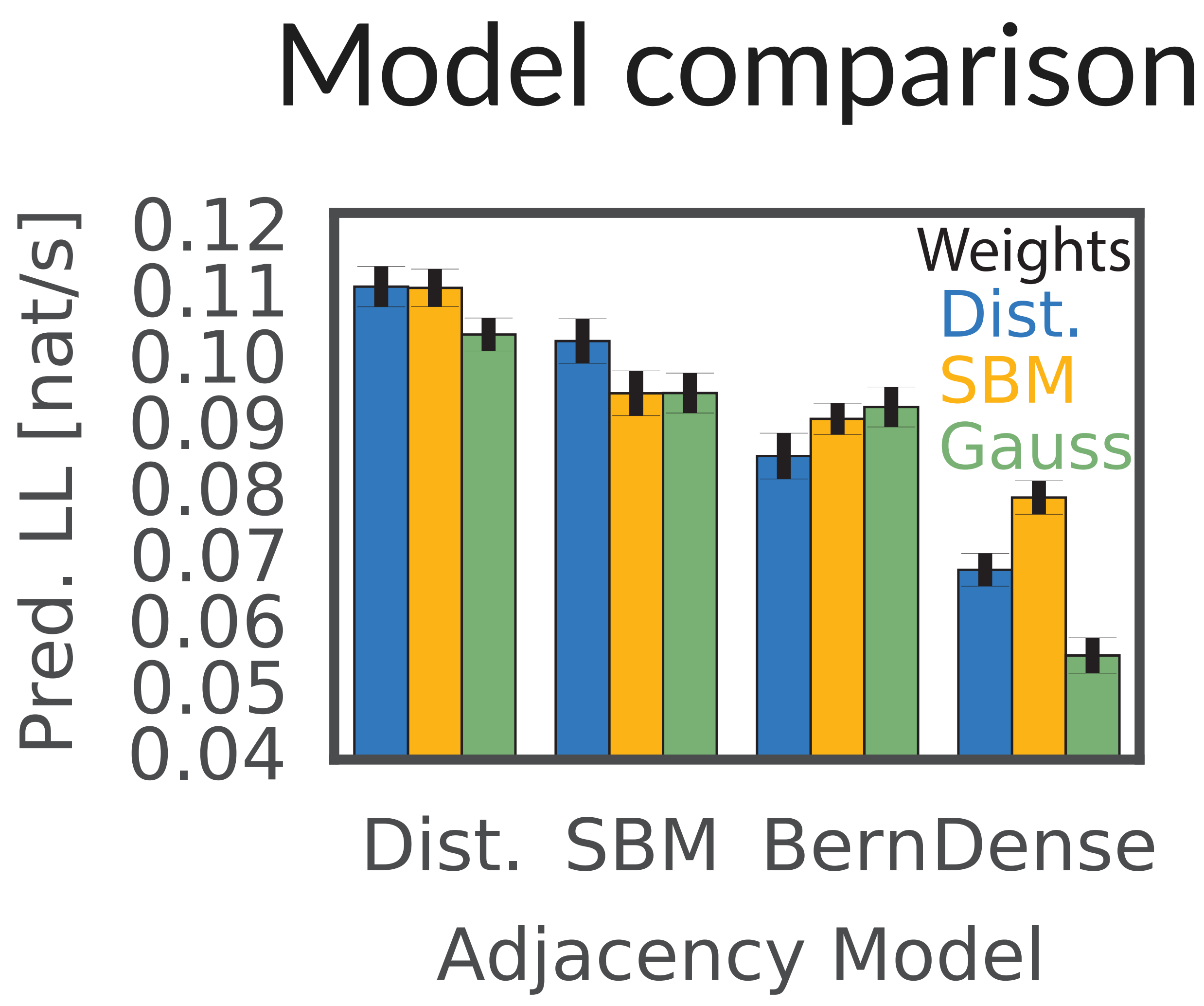


Linderman, Adams, Pillow (*NeurIPS*, 2016)

Application to real primate retina data







Linderman, Adams, Pillow (*NeurIPS*, 2016)

Poisson processes: the continuous time limit

Point processes

From spike *counts* to spike *times*

- To simplify notation, consider a single neuron for now and drop the n subscript.
- So far we've worked with **spike counts** $(y_1, \ldots, y_{T/\Delta t}) \in \mathbb{N}^{T/\Delta t}$ in discrete bins.
- With small bins, most have zero spikes!
- Instead, let's model the set of **spike times** $\{t_1, \ldots, t_S\} \subset [0,T)$ directly.

Poisson processes

- Stochastic processes that generate sets of points $\{t_s\}_{s=1}^{S}$; i.e. "spikes."

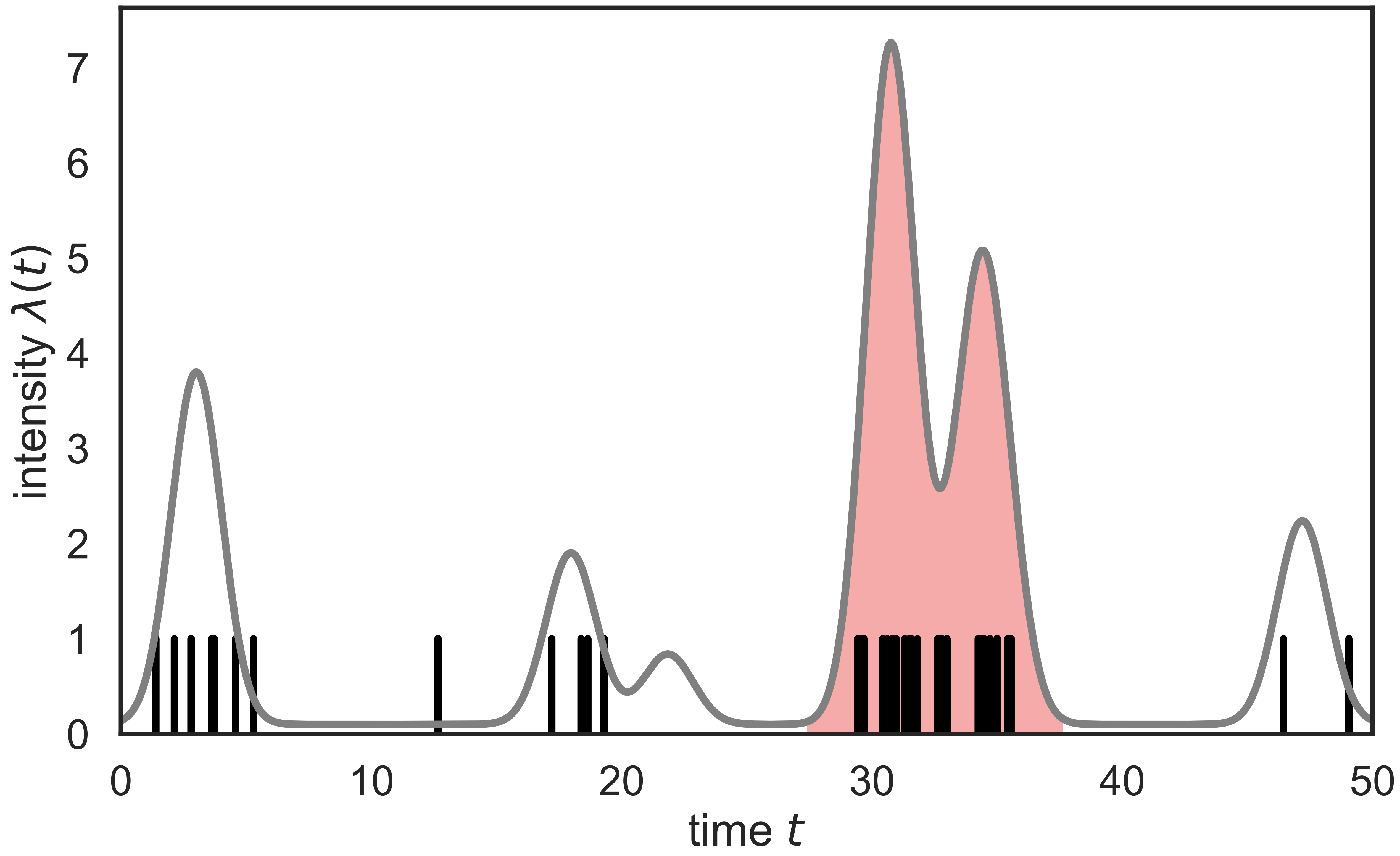
- Governed by an intensity function $\lambda(t)$.

- Poisson-distributed number of events in any interval:

$$N(\mathscr{A}) \sim \mathrm{Po}\left(\int_{\mathscr{A}} \lambda(t)\,\mathrm{d}t\right).$$

- Non-overlapping intervals are independent:

$$N(\mathscr{A}_1) \perp\!\!\!\perp N(\mathscr{A}_2) \quad \text{if} \quad \mathscr{A}_1 \cap \mathscr{A}_2 = \varnothing.$$



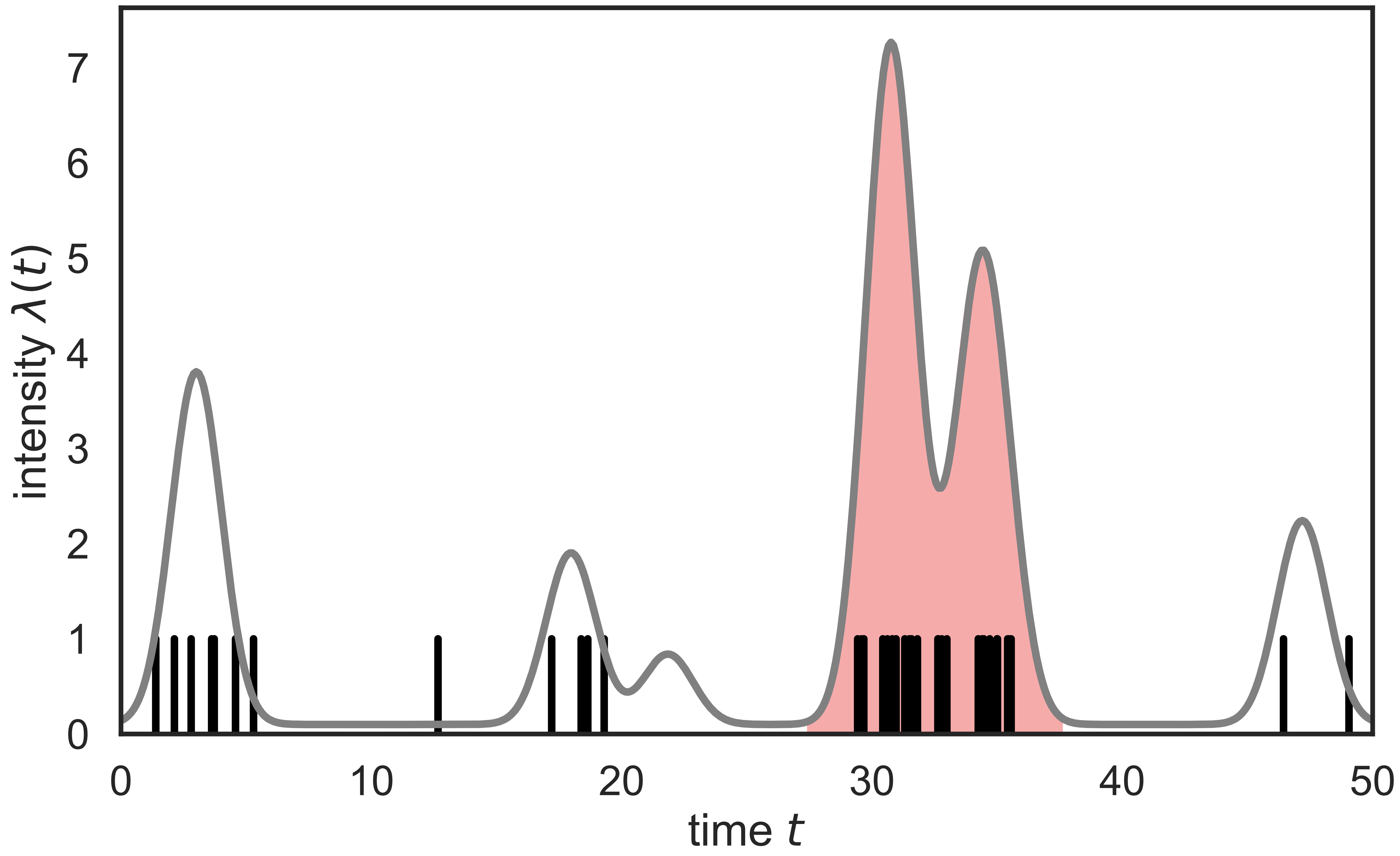
Simulating a Poisson Process

- Sample the total number of spikes:

$$S \sim \mathrm{Po}\left(\int_0^T \lambda(t)\,\mathrm{d}t\right).$$

- **Independently sample** the spike times:

$$t_s \sim \frac{\lambda(\,\cdot\,)}{\int_0^T \lambda(t)\,\mathrm{d}t} \quad \text{for } s = 1,\ldots,S.$$



Poisson processes

Deriving the Poisson process likelihood

Poisson processes

From GLMs to Poisson processes

- We can generalize our spike count model by assuming spike times are uniformly distributed within each bin.
- Let $\mathcal{S}_t \subset [t, t + \Delta t)$ denote the spike times in a single bin.
- Model:

$$y_t \sim \mathrm{Po}\left(\lambda_t \Delta t\right) \text{ for } t = 1,\ldots,T/\Delta t$$

$$t_s \sim \mathrm{Unif}\left([t, t + \Delta t)\right) \text{ for } s = 1,\ldots,y_t$$

$$\mathcal{S}_t = \{t_s\}_{s=1}^{y_t}$$

$$\mathcal{S} = \cup_{t=1}^{T/\Delta t} \mathcal{S}_t$$

Poisson processes

Deriving the likelihood again

- Derive the likelihood of this model and take the limit as $\Delta t \to 0$.

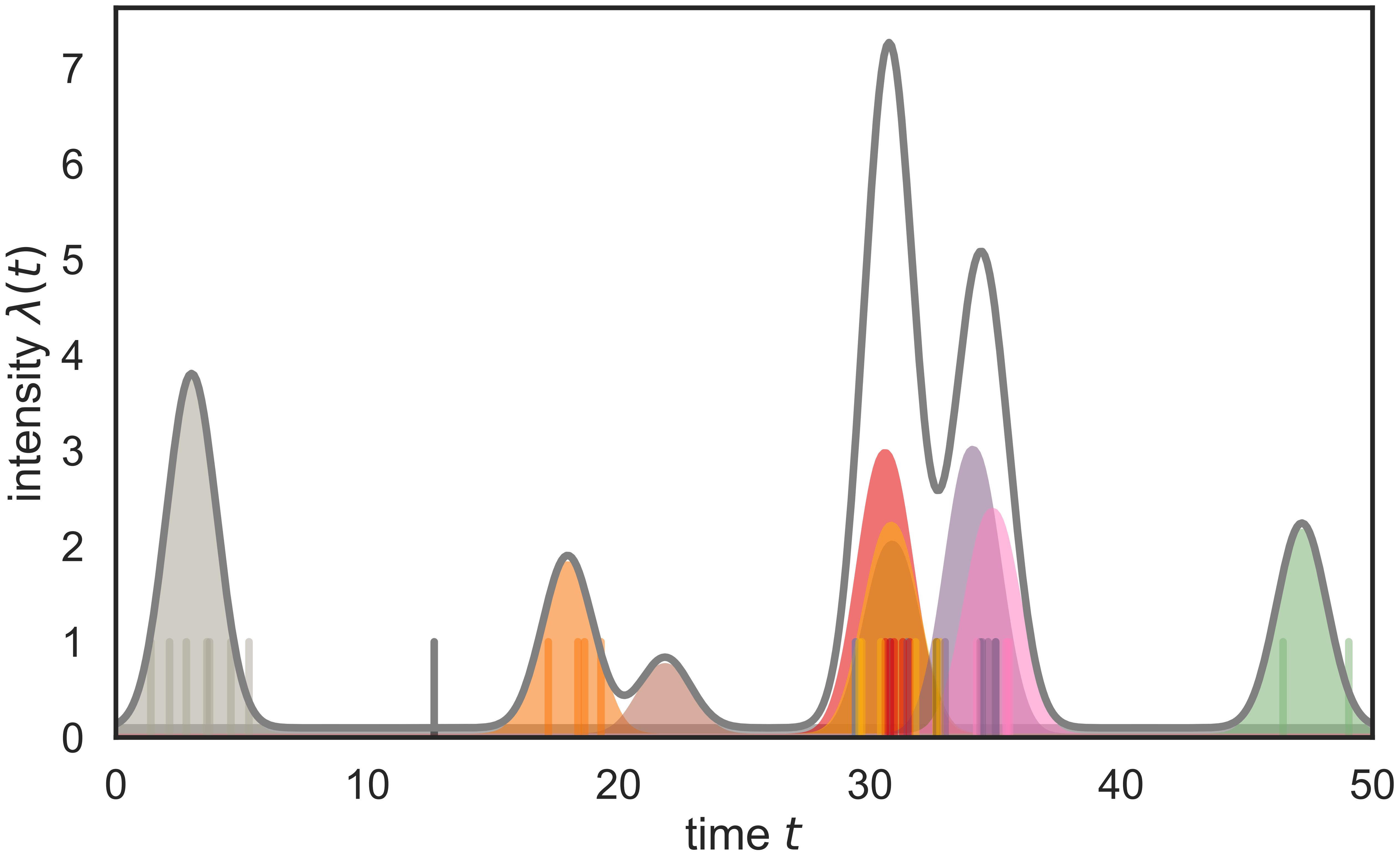
Poisson Superposition

- Suppose we have spikes from K independent Poisson processes.

$$\{t_s^{(k)}\}_{s=1}^{S_k} \sim \mathrm{PP}(\lambda_k(t)) \text{ for } k = 1, \ldots, K.$$

- The union is a Poisson process with the sum of the intensities,

$$\bigcup_{k=1}^{K} \{t_s^{(k)}\}_{s=1}^{S_k} \sim \mathrm{PP}\left(\sum_{k=1}^{K} \lambda_k(t) \right).$$



Hawkes Processes

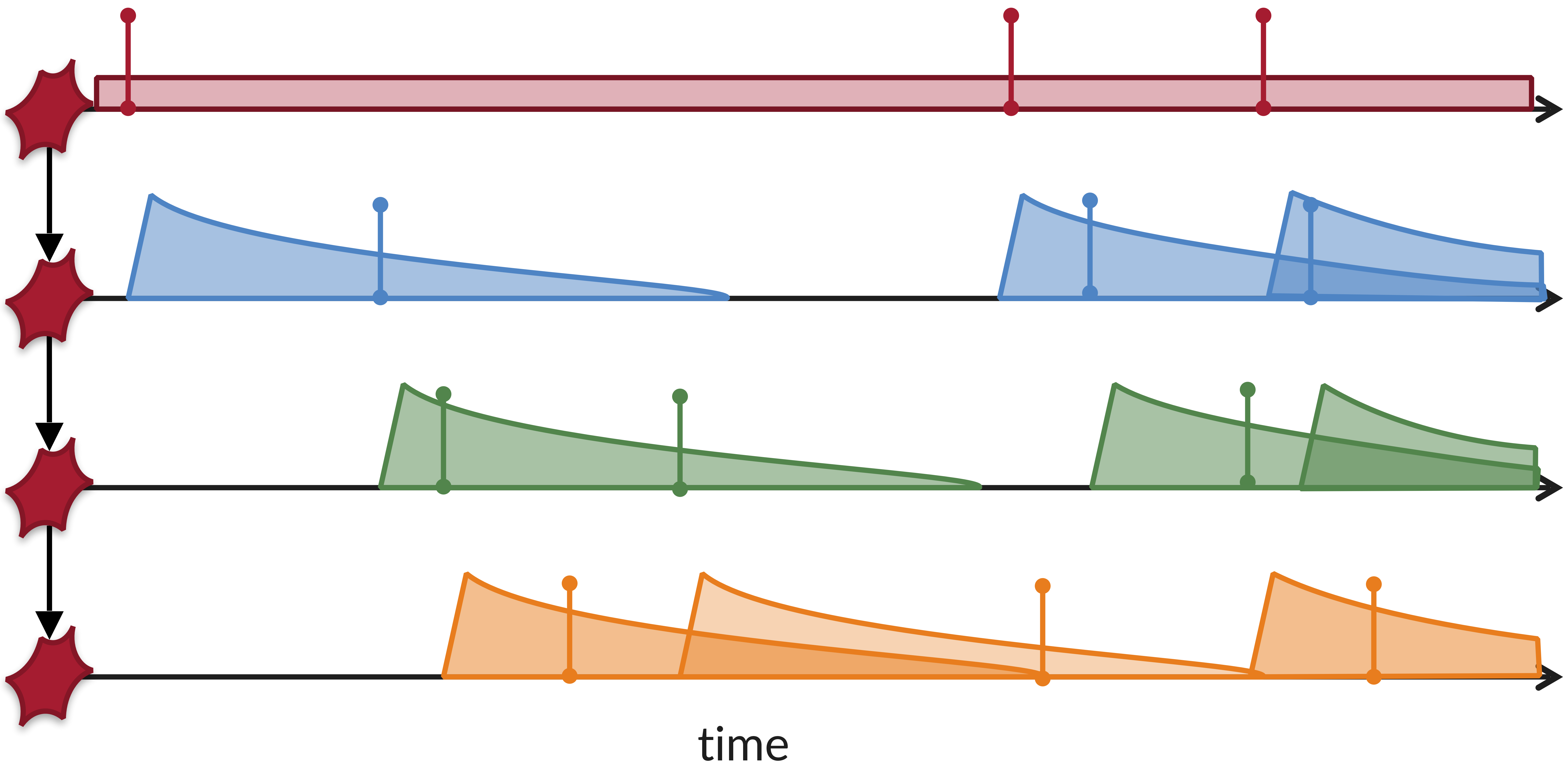
- Poisson processes are limited by the fact that non-overlapping intervals are independent.

- Hawkes processes (Hawkes, 1971) are point processes with **history dependence.**

$$\lambda(t) = \lambda_0 + \sum_{s:t_s<t} h(t - t_s)$$ **[past spikes affect current rate]**

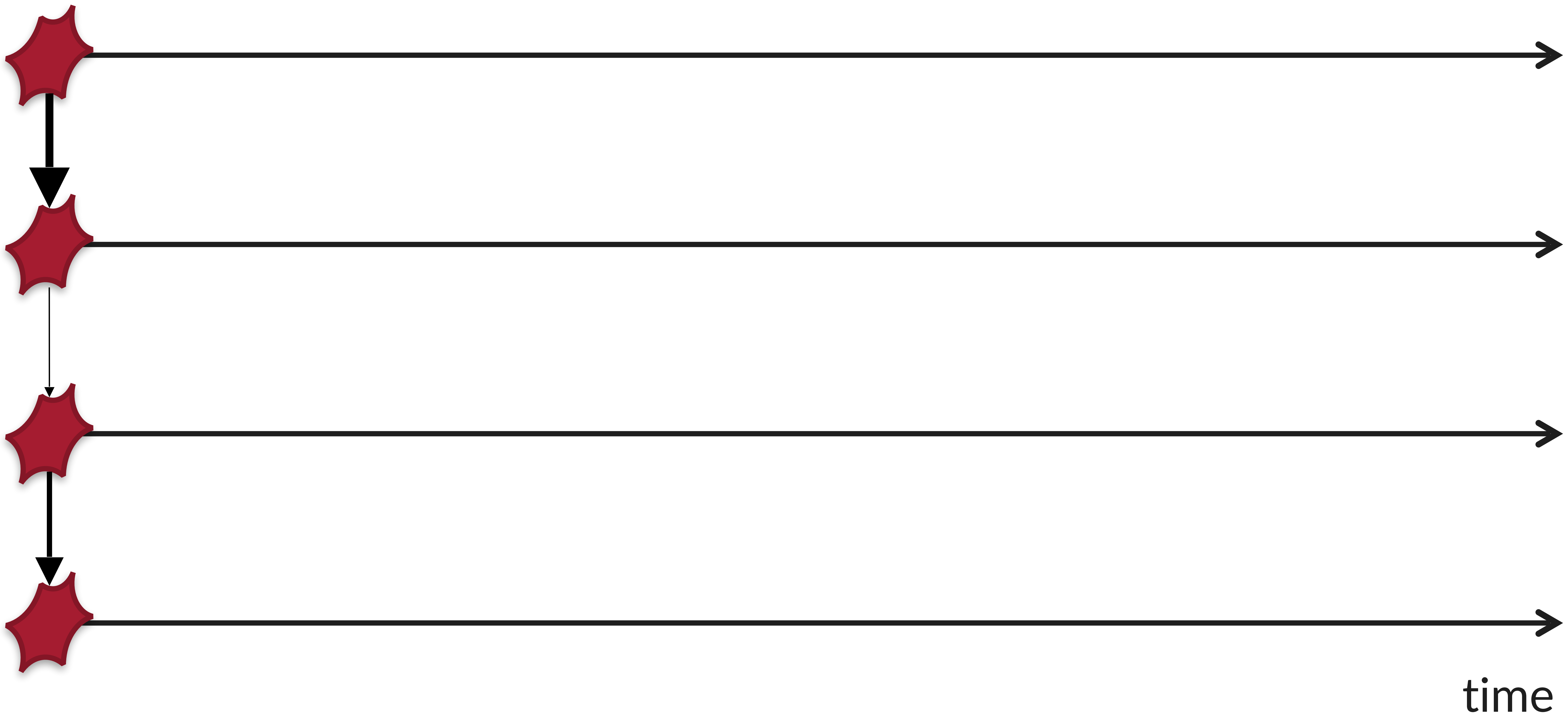
- We can easily generalize this model to have multiple **interacting processes.**

Simulating a Hawkes Process

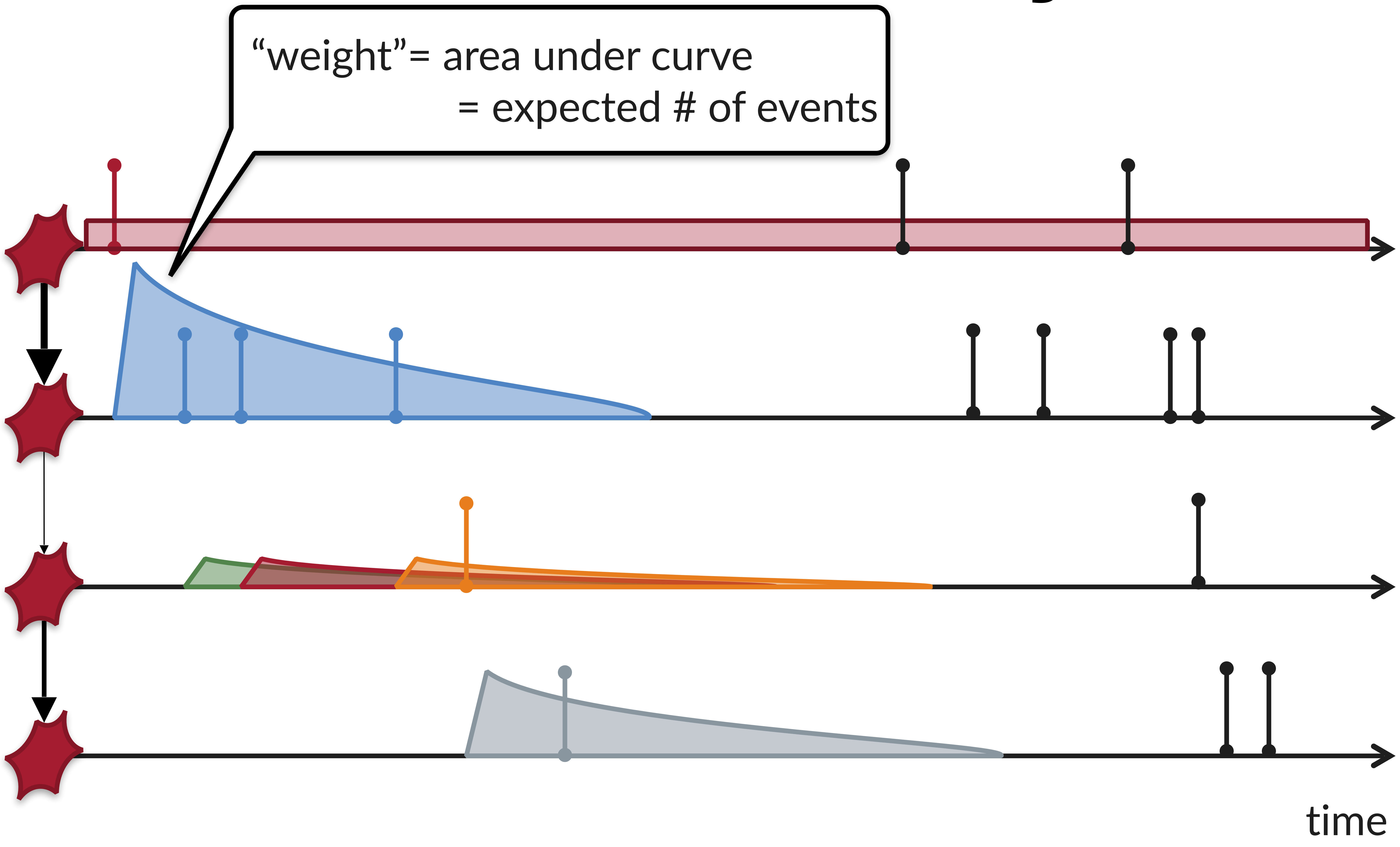


Weighted Hawkes Process Dynamics

What if some interactions are stronger than others?



Weighted Hawkes Process Dynamics



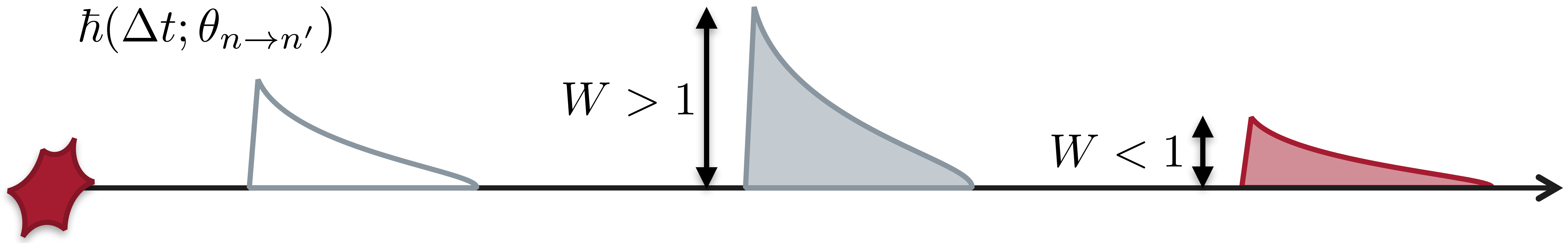
Incorporating network models

$$h_{n\to n'}(\Delta t) = A_{n\to n'} \cdot W_{n\to n'} \cdot \hbar(\Delta t; \theta_{n\to n'})$$

Binary adjacency matrix

Weight matrix

Normalized impulse response



Conclusion

- **Generalized linear models** are commonly used for modeling neural responses to external stimuli.
  - They are intuitive, easy to fit, and easy to generalize.
  - We can also go deeper, adding additional layers to learn features.
- We can capture **correlations between neurons** via coupling weights.
  - Number of parameters grows as $O(N^2)$, but network priors offer interpretable regularization.
- In the continuous time limit, these models become Poisson processes and Hawkes processes, respectively.

Further reading

- Pillow, Jonathan W., Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M. Litke, E. J. Chichilnisky, and Eero P. Simoncelli. 2008. “Spatio-Temporal Correlations and Visual Signalling in a Complete Neuronal Population.” Nature 454 (7207): 995–99.

- McIntosh, L. T., Maheswaranathan, N., Nayebi, A., Ganguli, S., & Baccus, S. A. (2016). Deep learning models of the retinal response to natural scenes. Advances in neural information processing systems, 29, 1369.

- Linderman, Scott W., Ryan P. Adams, and Jonathan W. Pillow. "Bayesian latent structure discovery from multi-neuron recordings." Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016.

- Linderman, Scott W., and Ryan P. Adams. 2014. “Discovering Latent Network Structure in Point Process Data.” In Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML).