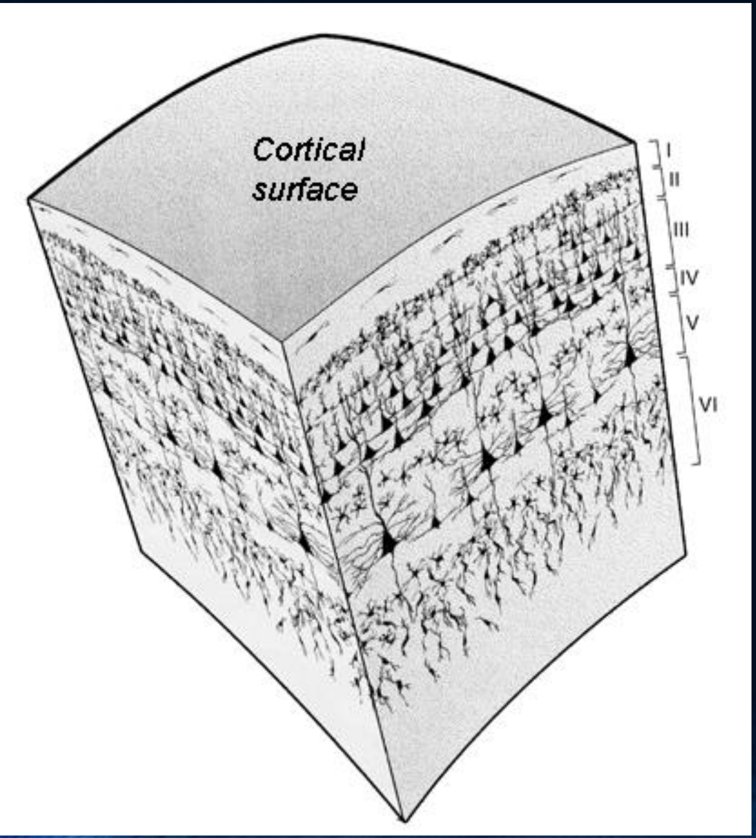
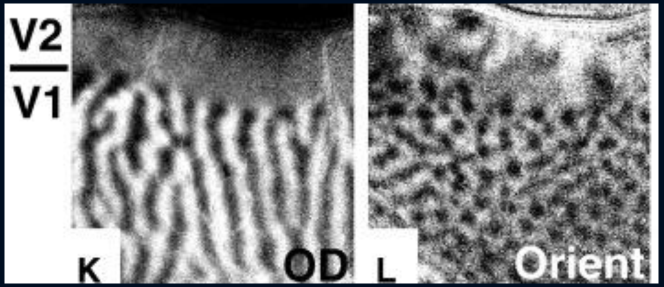
Big Data and the Brain

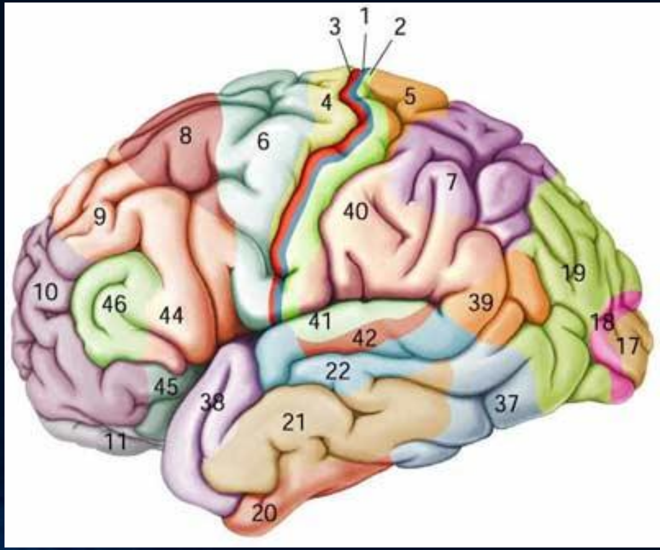
MICHAEL OLIVER
GALLANT LAB
UC BERKELEY

The brain is gigantic

- The human brain has ~100 billion neurons connected by ~100 trillion synapses
- Multiple levels of organization

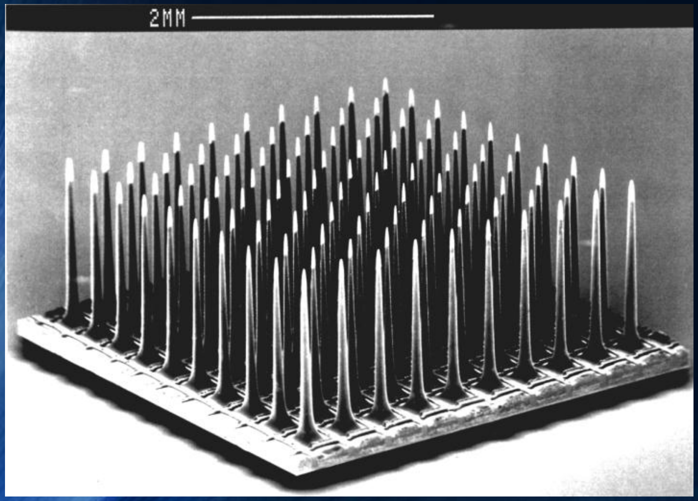




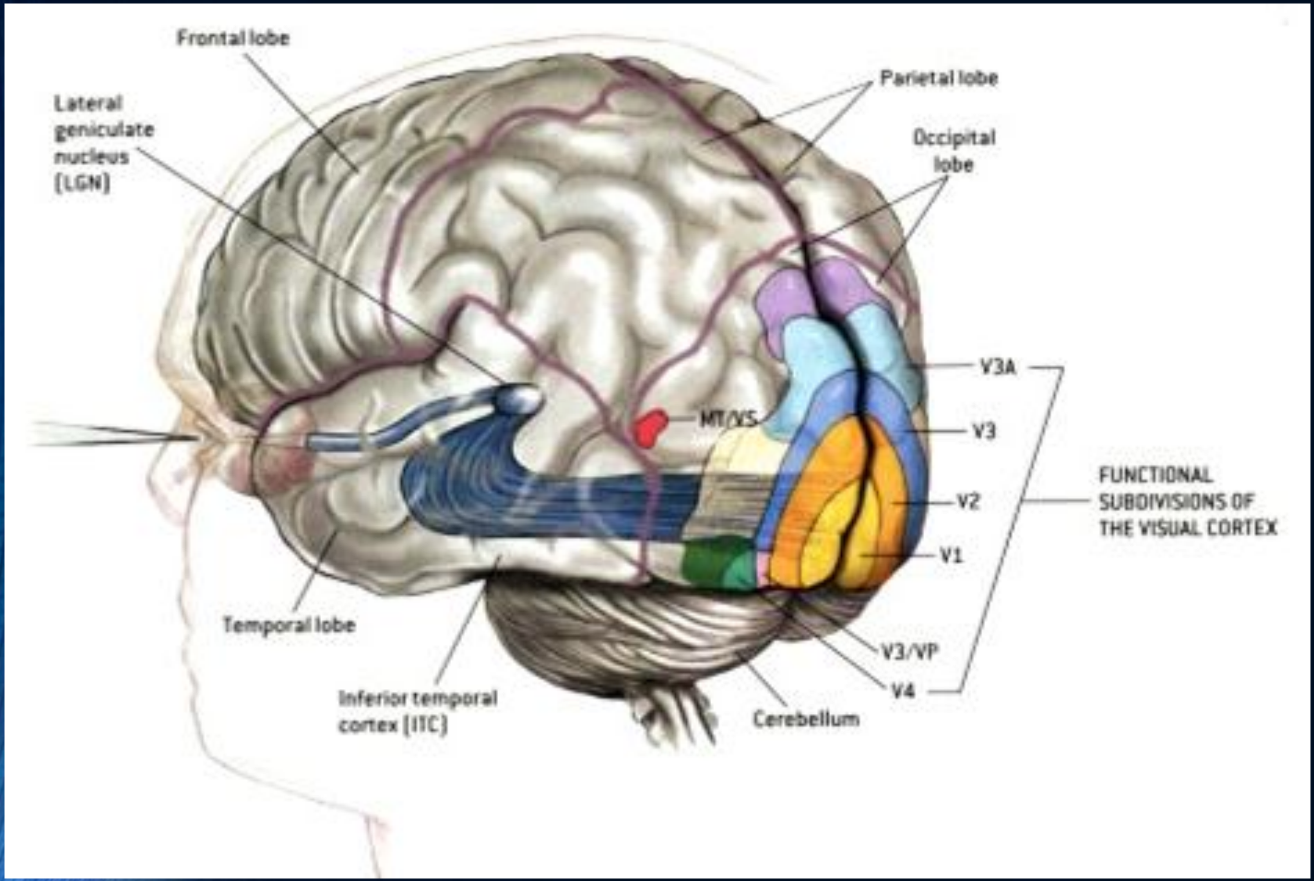


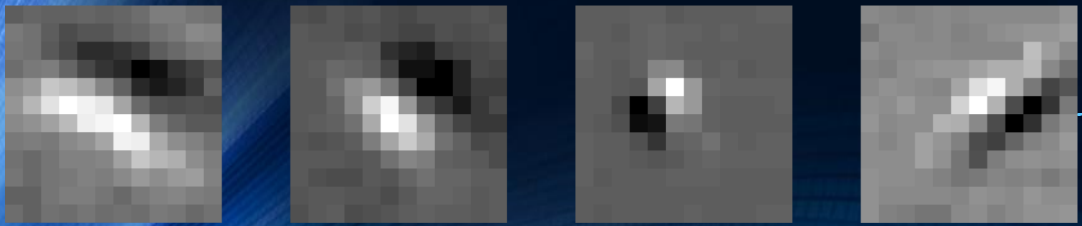
But our data is only “Big”

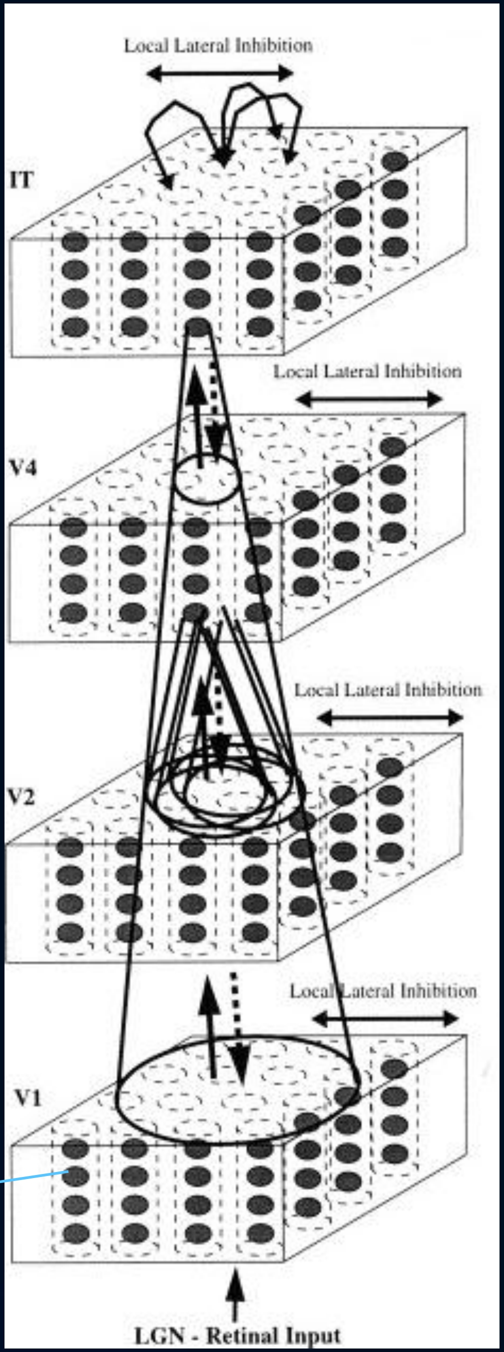
- Electrophysiology experiments can record from ~100 neurons simultaneously
- fMRI experiments we can record from ~90,000 voxels of about 20 mm^3
  - There are over 2,000,000 neurons per voxel



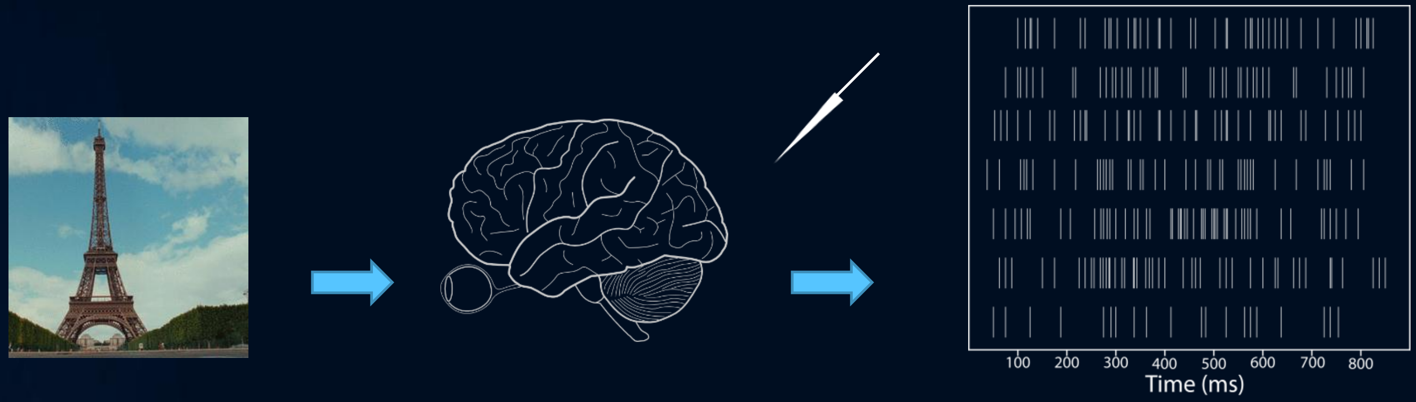
The visual brain





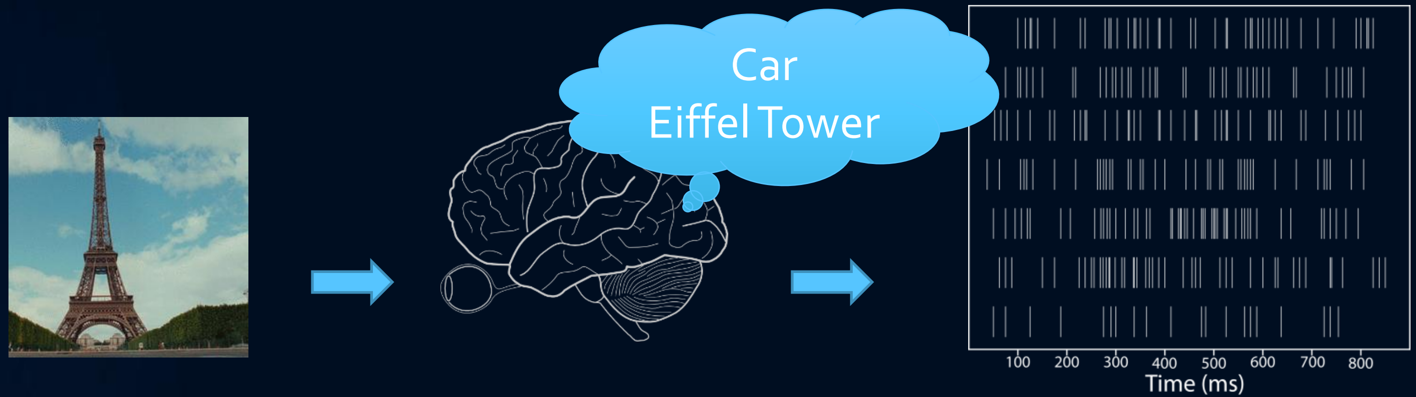


System identification and neuroscience



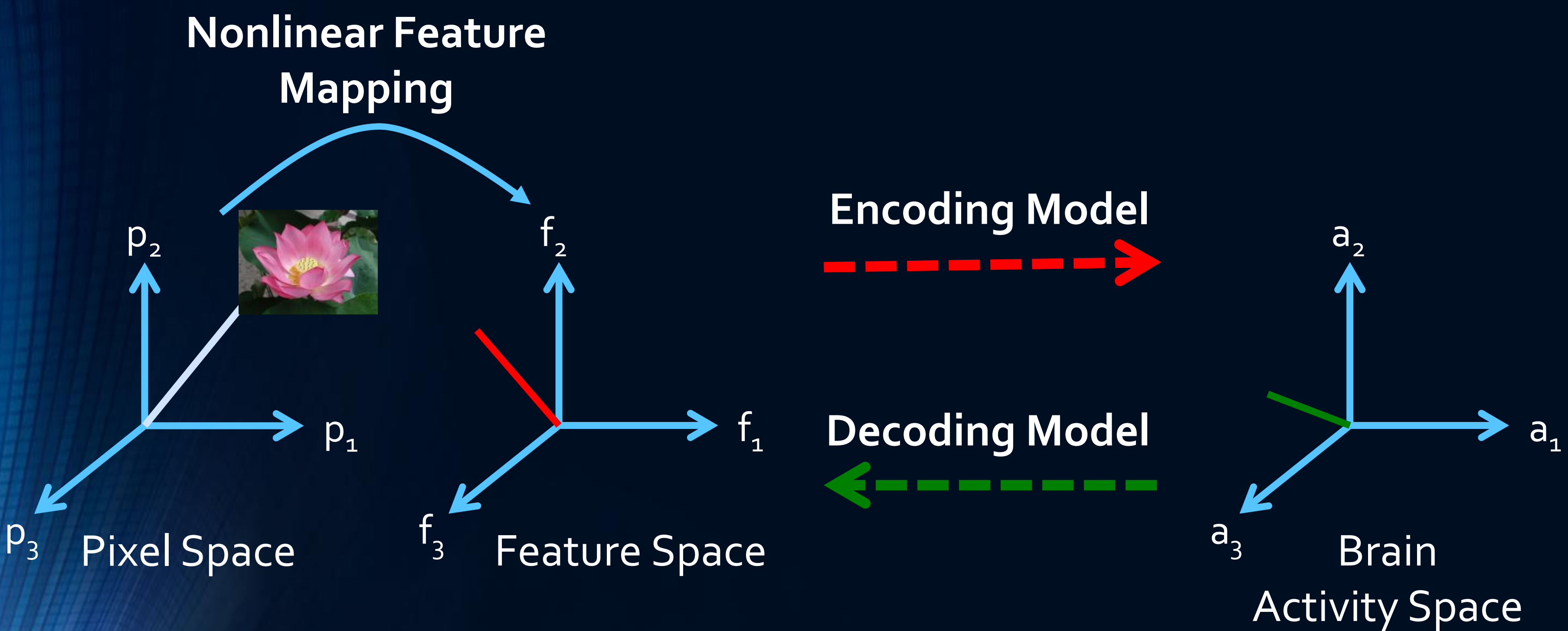
- Model each neuron based on relationship between stimulus and response
- Evaluate models based on their ability to predict responses to novel stimuli

Why system identification in visual cortex is hard

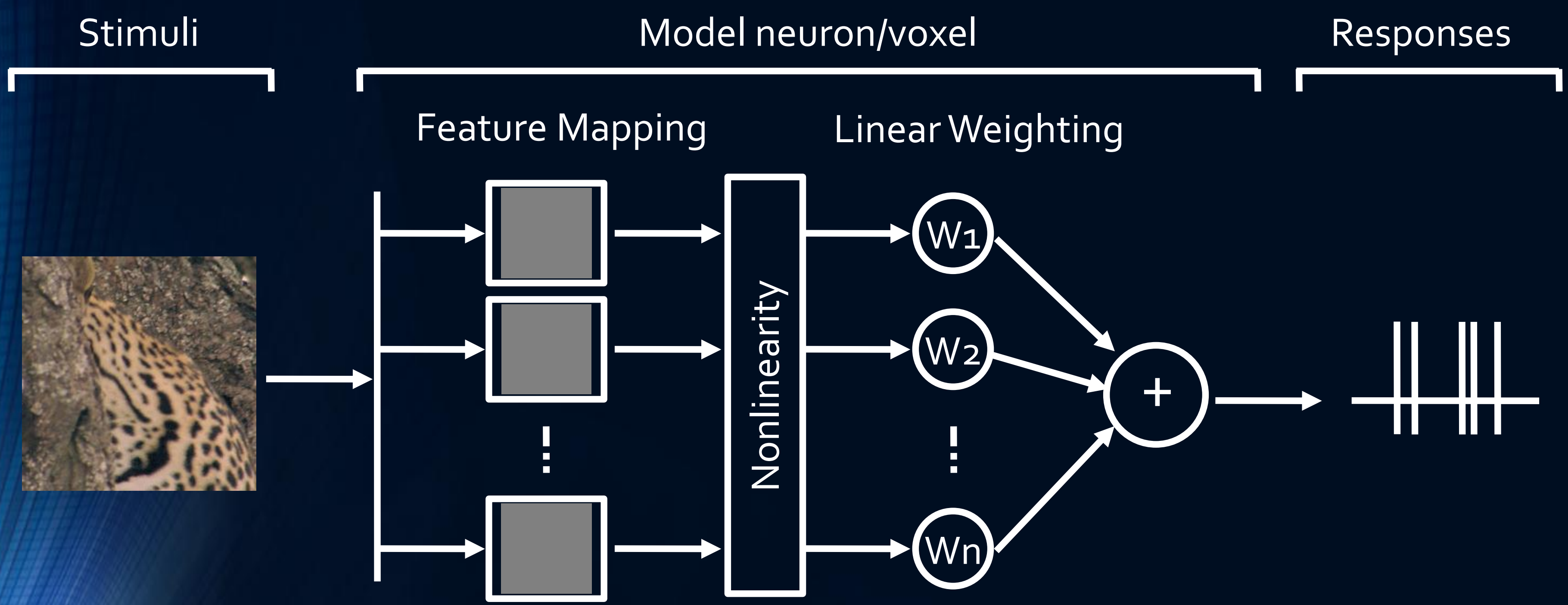


- Non-linear
- High dimensional
- Interpretability is important!

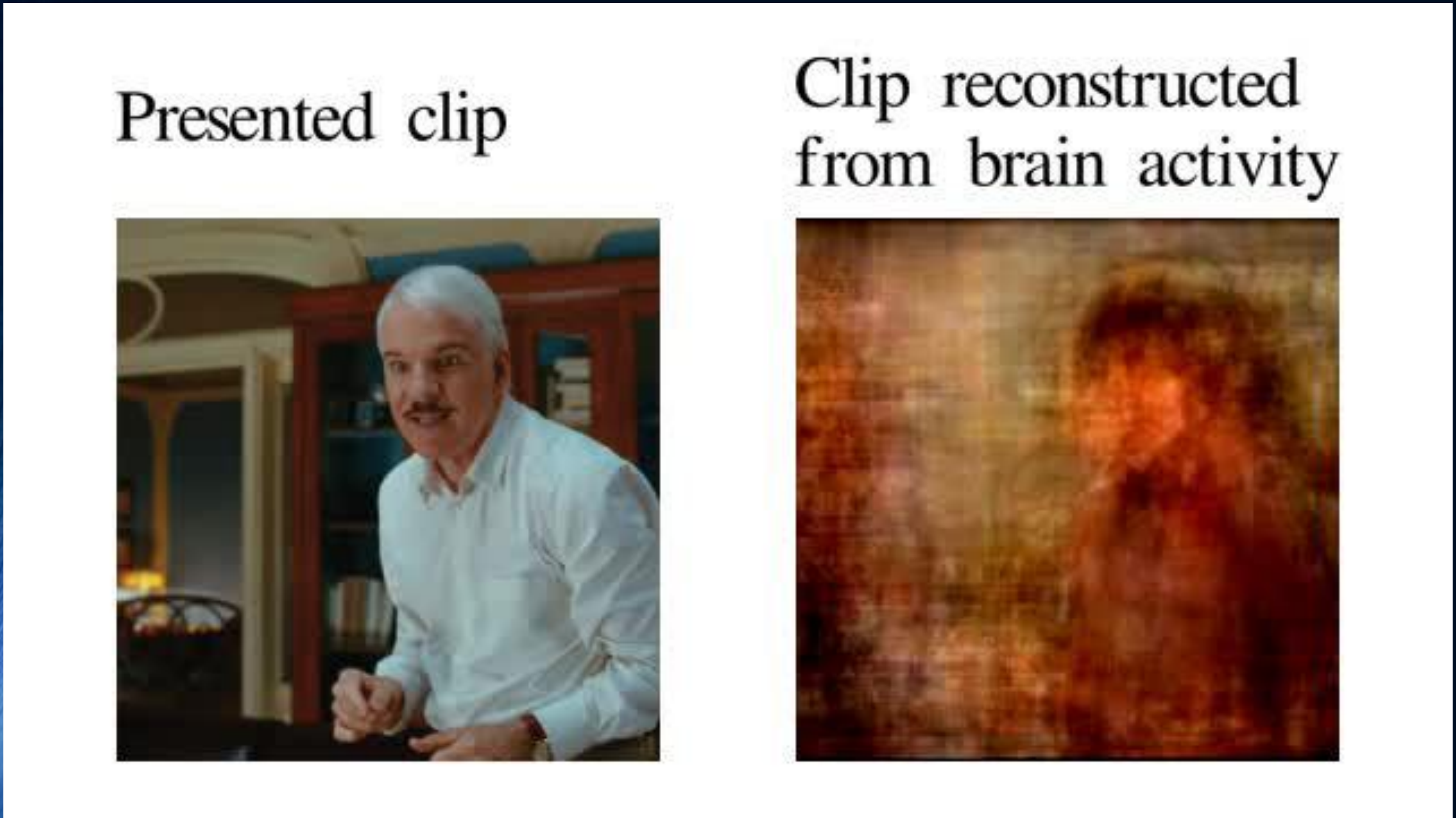
Linearized regression



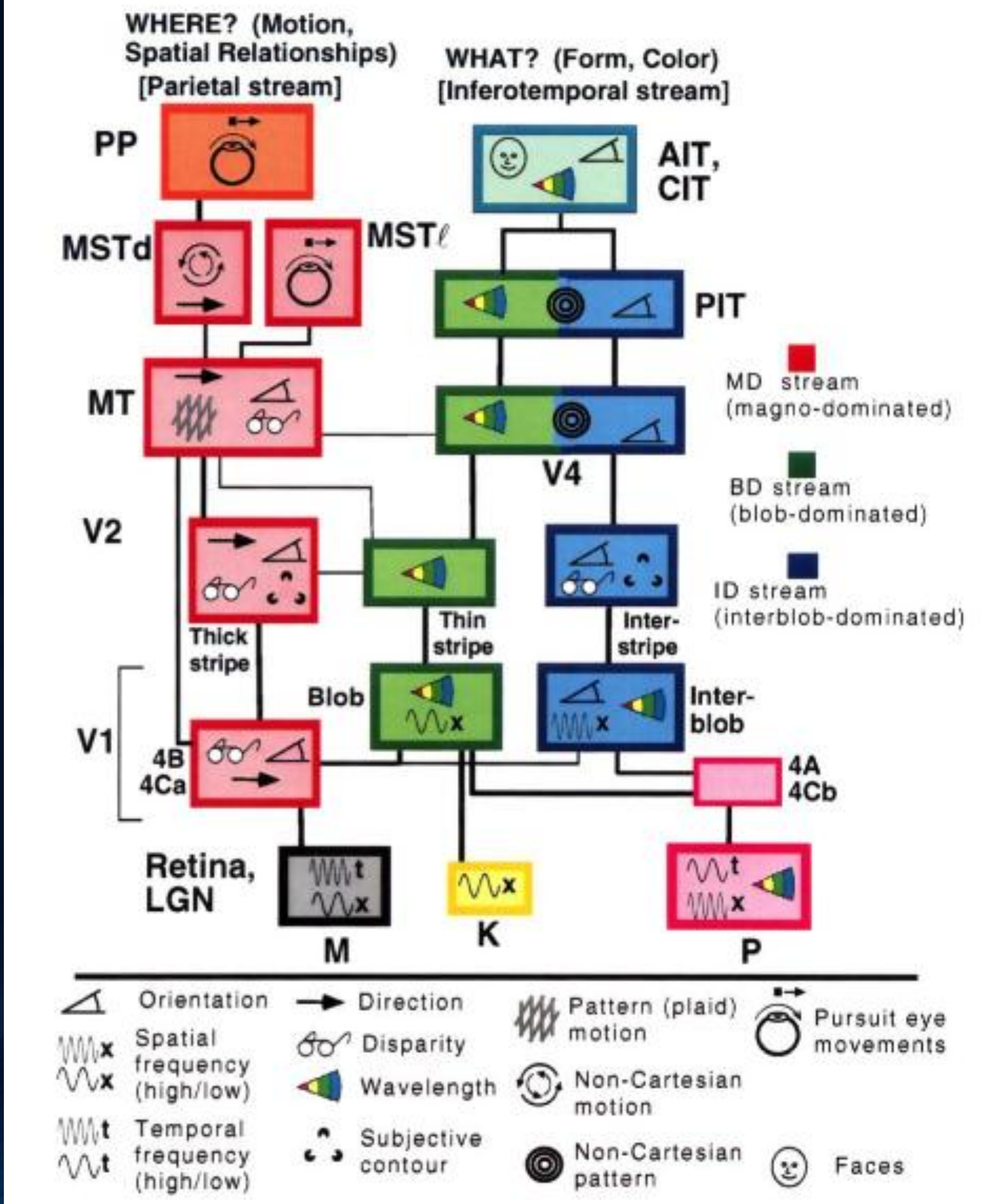
Feature spaces



Movie reconstruction from fMRI data



Nishimoto S, et al. Curr Biol. 2011 Oct 11;21(19):1641-6



Van Essen DC, **Gallant JL**. Neuron. 1994 Jul;13(1):1-10.

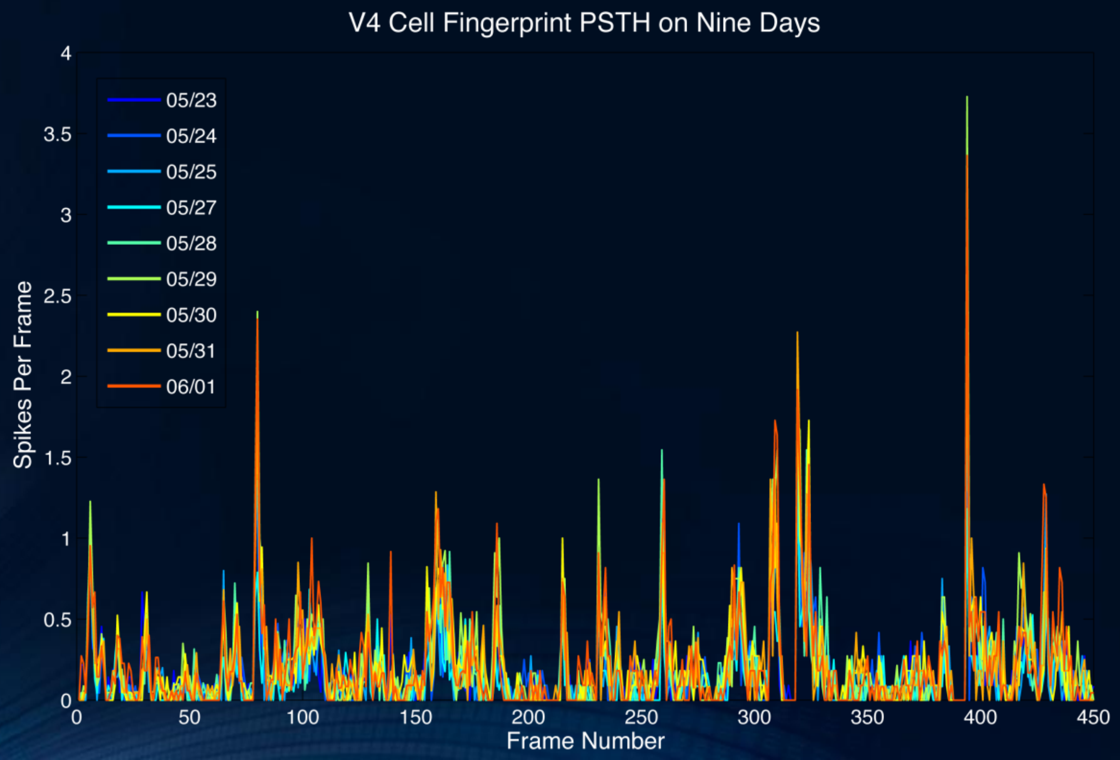
Tuning in V4

We need big data!

- V4 receptive fields are moderately large
- Potential stimulus space is very large
- Natural images span the relevant space
- Response is highly nonlinear

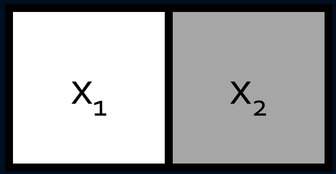
How to get big data

- Implantable electrodes allow us to record from the same cell over many days
- We used over 1 million frames of natural movies, the largest ever stimulus set in V4



General nonlinear modeling

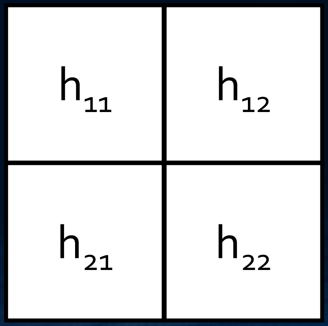
- The Volterra Series:

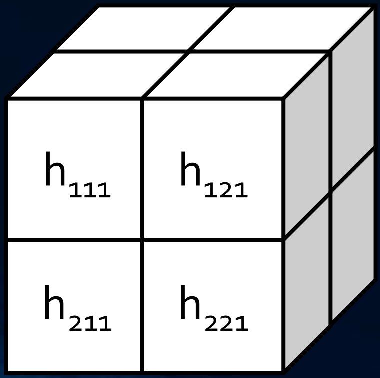


$$F(x(t)) = h_0 + \sum_{l=0}^{\infty} h_1(l)x_1(t-l) + \sum_{l=0}^{\infty} h_2(l)x_2(t-l) + \sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty} h_{11}(l_1,l_2)x_1(t-l_1)x_1(t-l_2) +$$

$$\sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty} h_{12}(l_1,l_2)x_1(t-l_1)x_2(t-l_2) + \sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty} h_{22}(l_1,l_2)x_2(t-l_1)x_2(t-l_2)\ldots$$

- Can control model flexibility by order choice
- Parameter space grows quickly with order





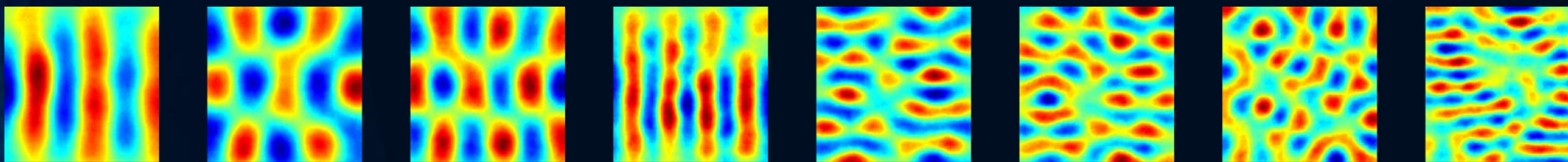
The scale of the problem

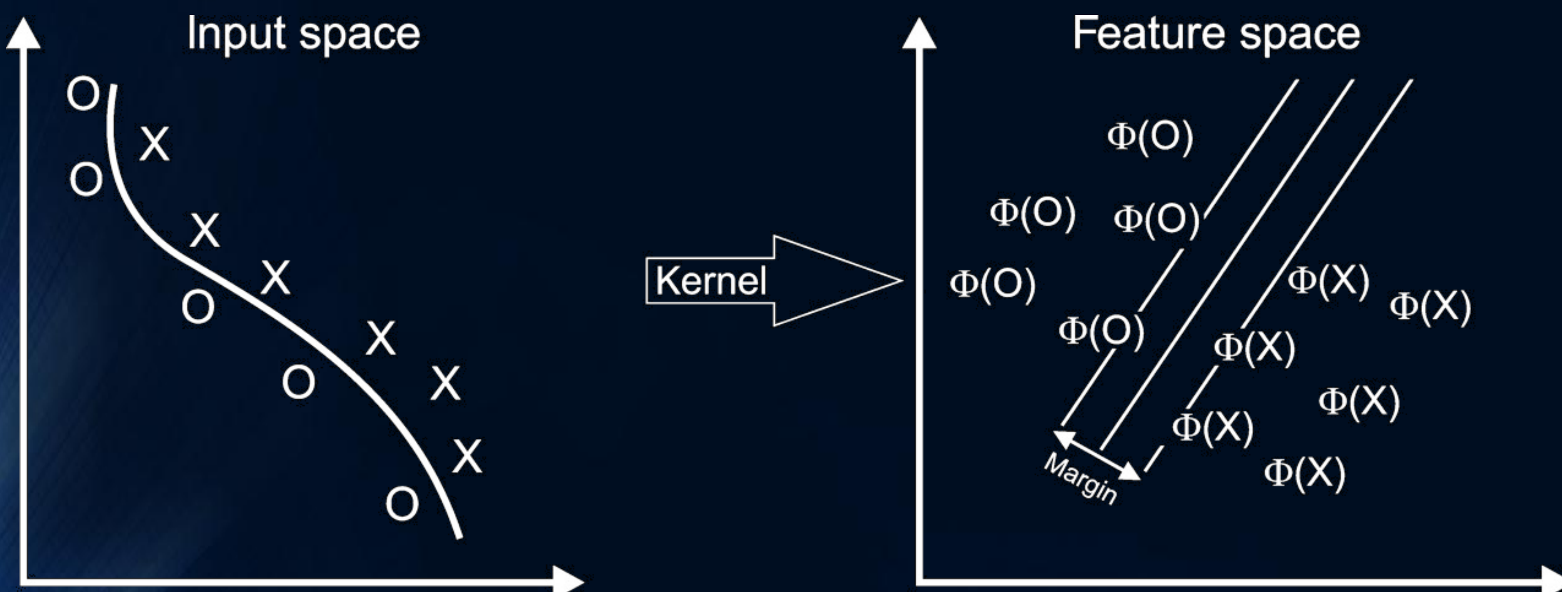
- If we have 1000 pixels
  - 2nd Order: 500,000+ coefficients
  - 3rd Order: 160 million+ coefficients
  - 4th Order: 40 billion+ coefficients

- But we actually have about 196,608 pixels…
  - 2nd Order: 19 billion+ coefficients
  - 3rd Order: 1.2 quadrillion+ coefficients
  - 4th Order: 62 quintillion+ coefficients

Pixels
1.2×10^{15}

Taming dimensionality





Pixels 1.2×10^{15} → PCA 1.6×10^{8} →

Kernel regression

$$K = k(x^i, x^j)$$ For all i,j =1:n in training data

$$\alpha = K^{-1} y$$ Calculate weights for kernel regression model

Use weights to make predictions for new x

$$y(x) = \sum_i \alpha_i k(x, x^i)$$

$$y(x) = \sum_i \alpha_i \left\langle \phi(x), \phi(x^i) \right\rangle$$

Kernel function equivalent to dot product in feature space ϕ

Pixels 1.2×10^{15} → PCA 1.6×10^{8} → Kernel 1×10^{6} →

The Inhomogeneous Polynomial Kernel

$$k(x, x') = (x \cdot x' + 1)^d$$

2nd Order IHP:

$$(x \cdot x' + 1)^2 = (x_1 x_1' + x_2 x_2' + 1)^2$$

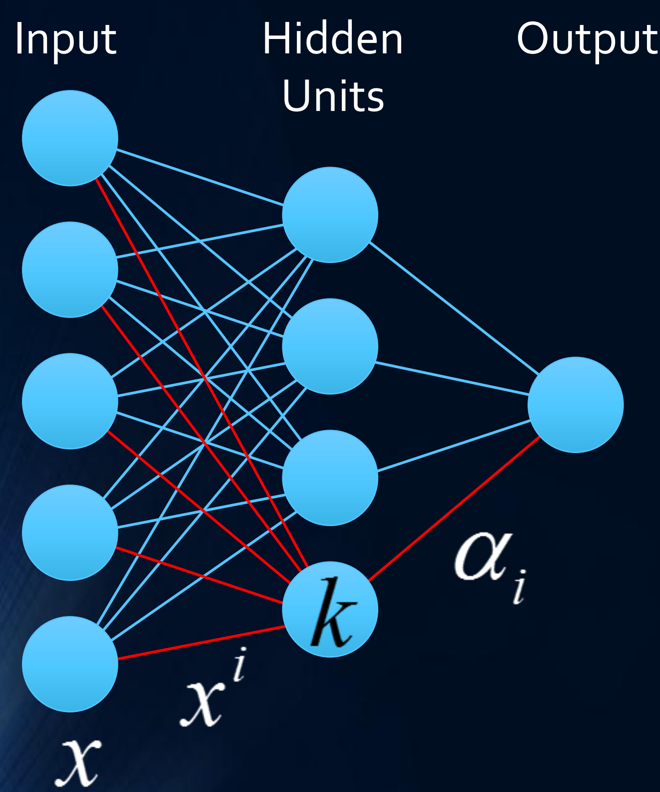
$$(x_1 x_1' + x_2 x_2' + 1)(x_1 x_1' + x_2 x_2' + 1)$$

$$(x_1 x_1')^2 + (x_2 x_2')^2 + 2 x_1 x_1' x_2 x_2 + 2 x_1 x_1 + 2 x_2 x_2 + 1$$

$$\begin{pmatrix} (x_1)^2 \\ (x_2)^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2} x_1 \\ \sqrt{2} x_2 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} (x_1')^2 \\ (x_2')^2 \\ \sqrt{2} x_1' x_2' \\ \sqrt{2} x_1' \\ \sqrt{2} x_2' \\ 1 \end{pmatrix}$$

Implicitly Maps to feature space containing all first and second order terms

Neural networks as kernel machines



In a standard NN:

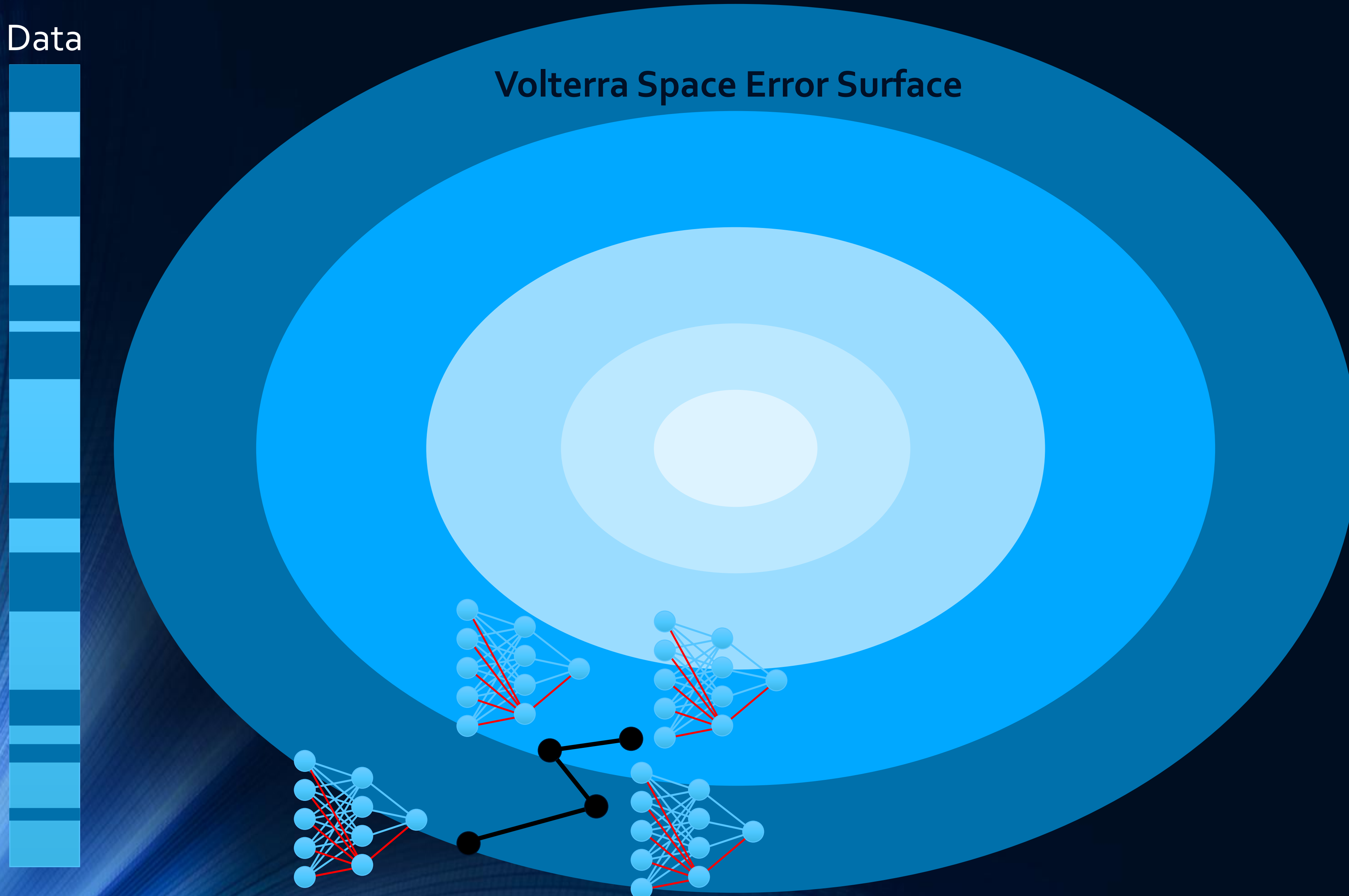
$$k(x, x^i) = \tanh(x \cdot x^i)$$

$$y(x) = \sum_i \alpha_i k(x, x^i)$$

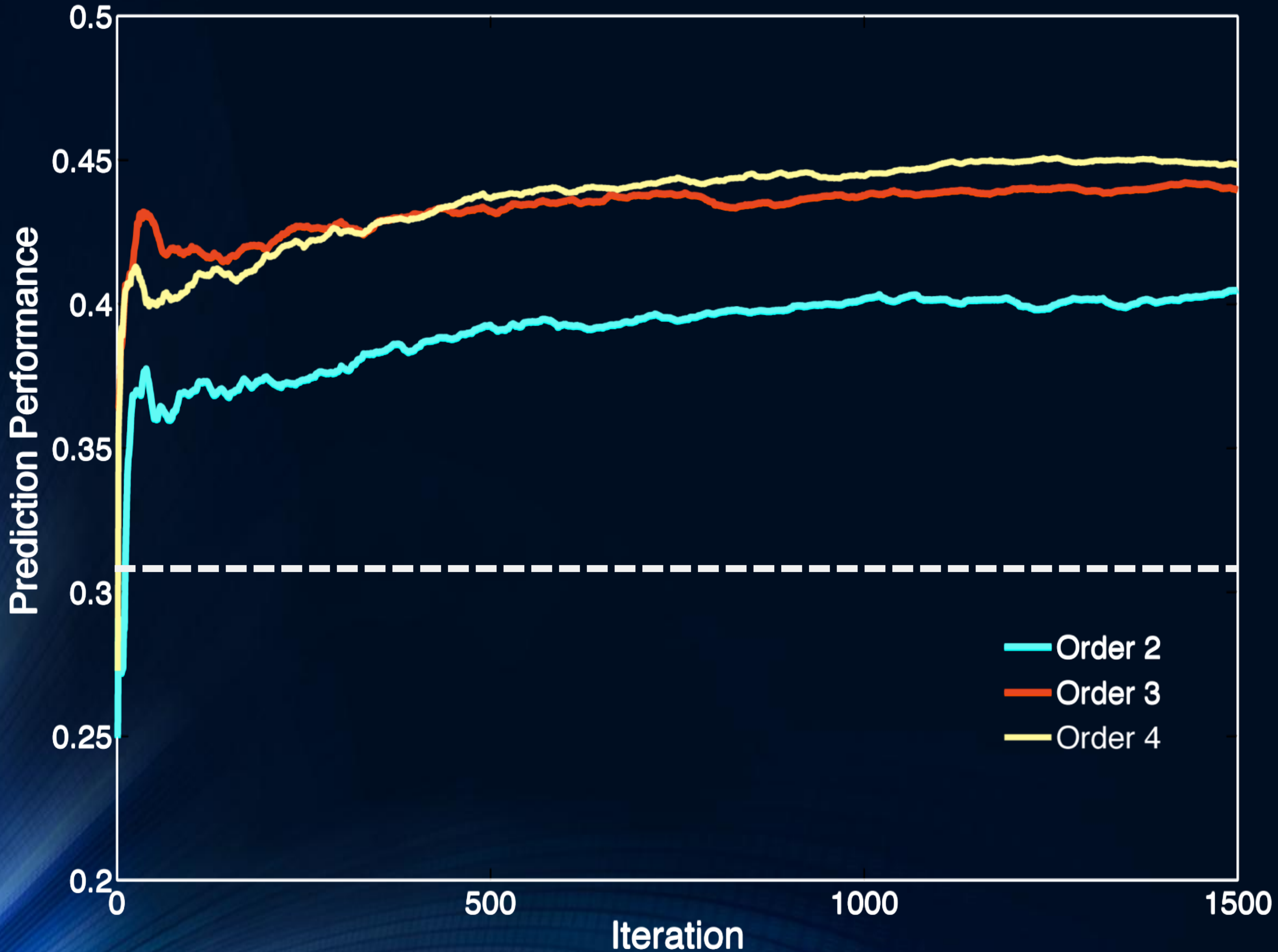
From NN perspective, kernel regression with a tanh kernel function is equivalent to a NN with hidden units = training samples

Pixels 1.2×10^{15} → PCA 1.6×10^{8} → Kernel 1×10^{6} → IPKN 1.5×10^{4}

Stochastic gradient boosting

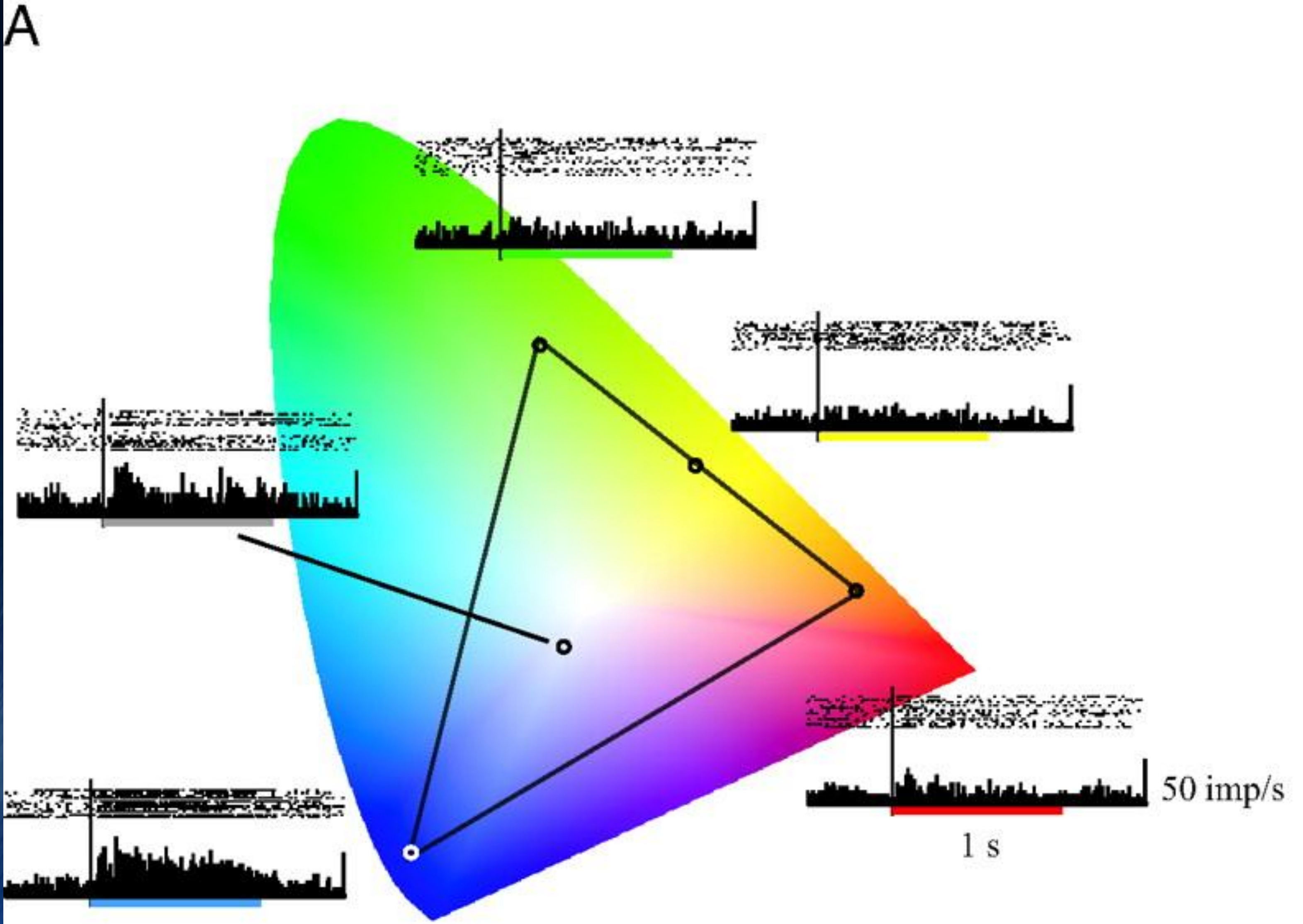


Prediction performance by model order



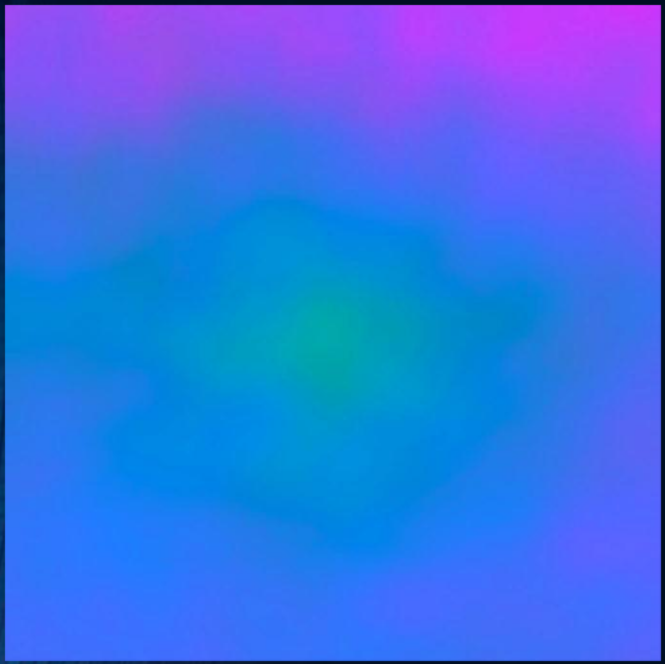
Color constancy in V4

V4 Tuning for Color

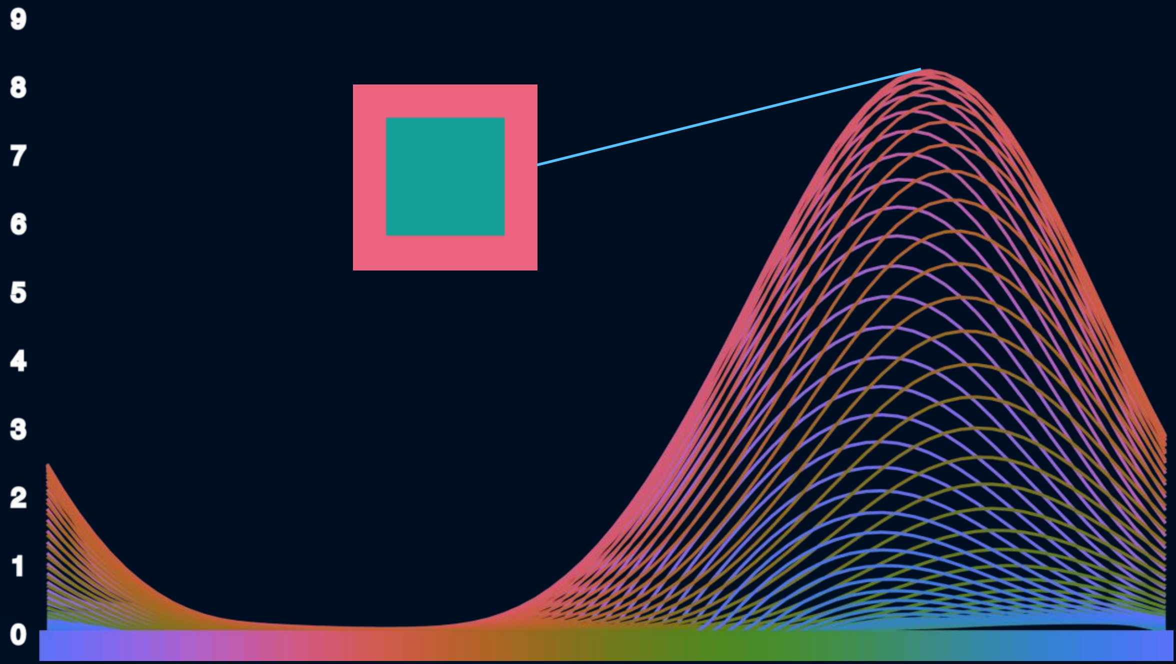


Kusunoki M et al. J Neurophysiol 2006;95:3047-3059

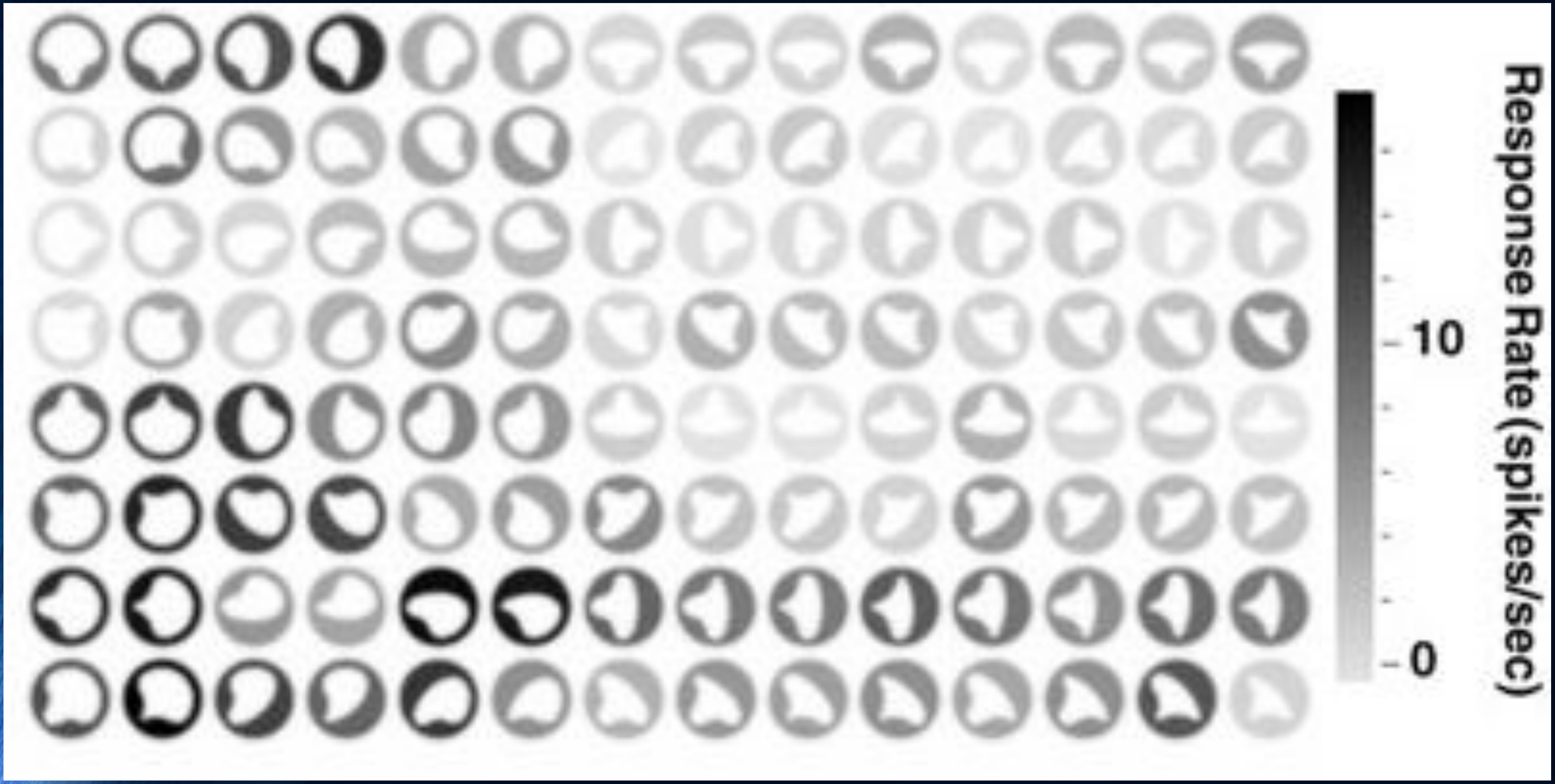
Color Tuning of V4 Cells



First Order Color Coefficients

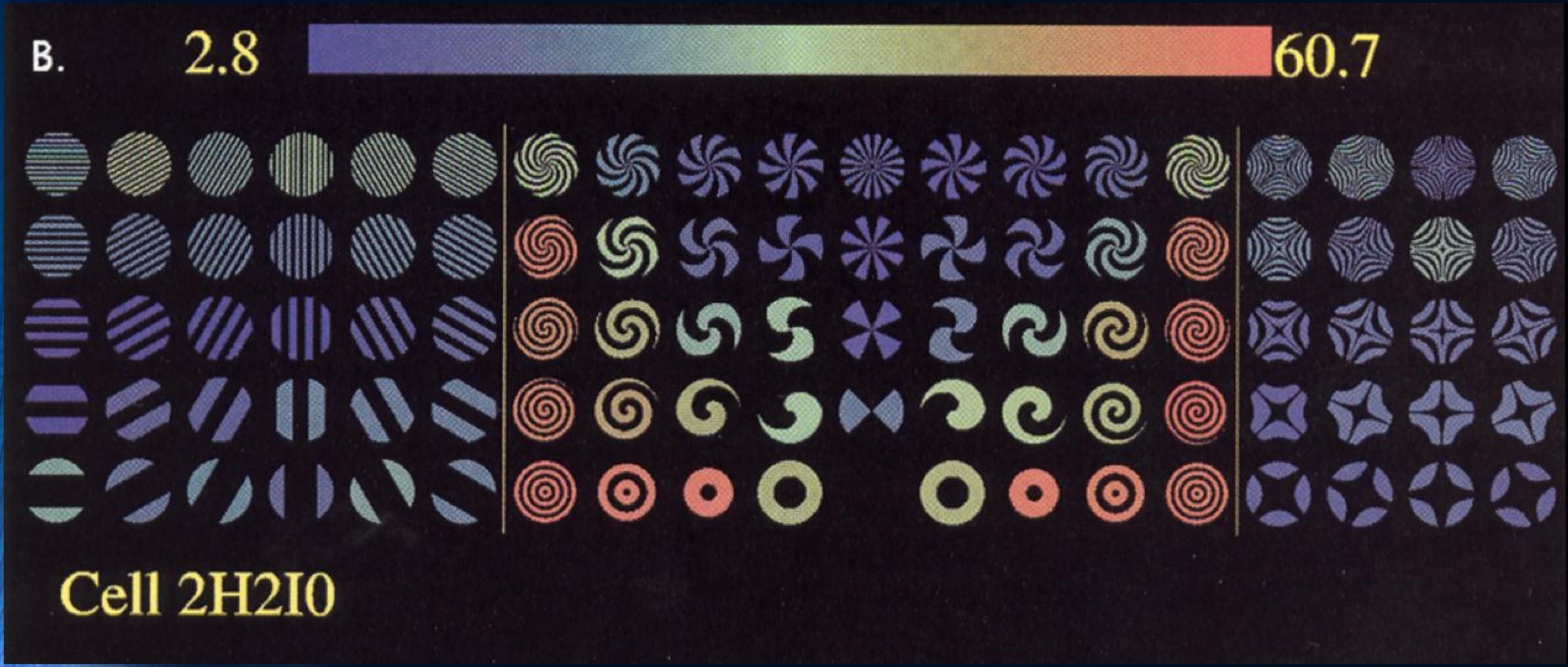


V4 tuning to curvature



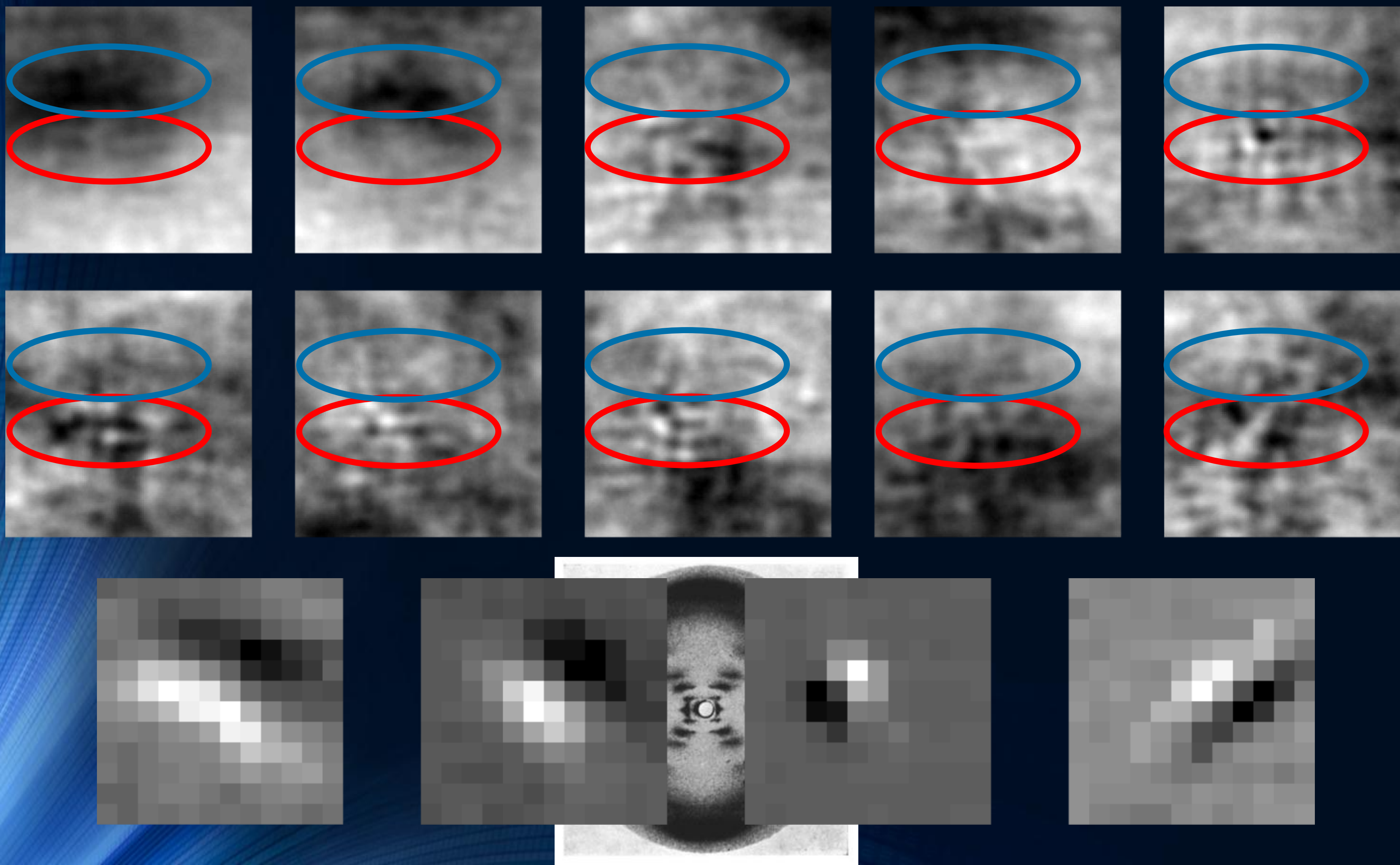
Pasupathy A , and Connor C E J Neurophysiol 2001;86:2505-2519

V4 tuning to Non-Cartesian Gratings

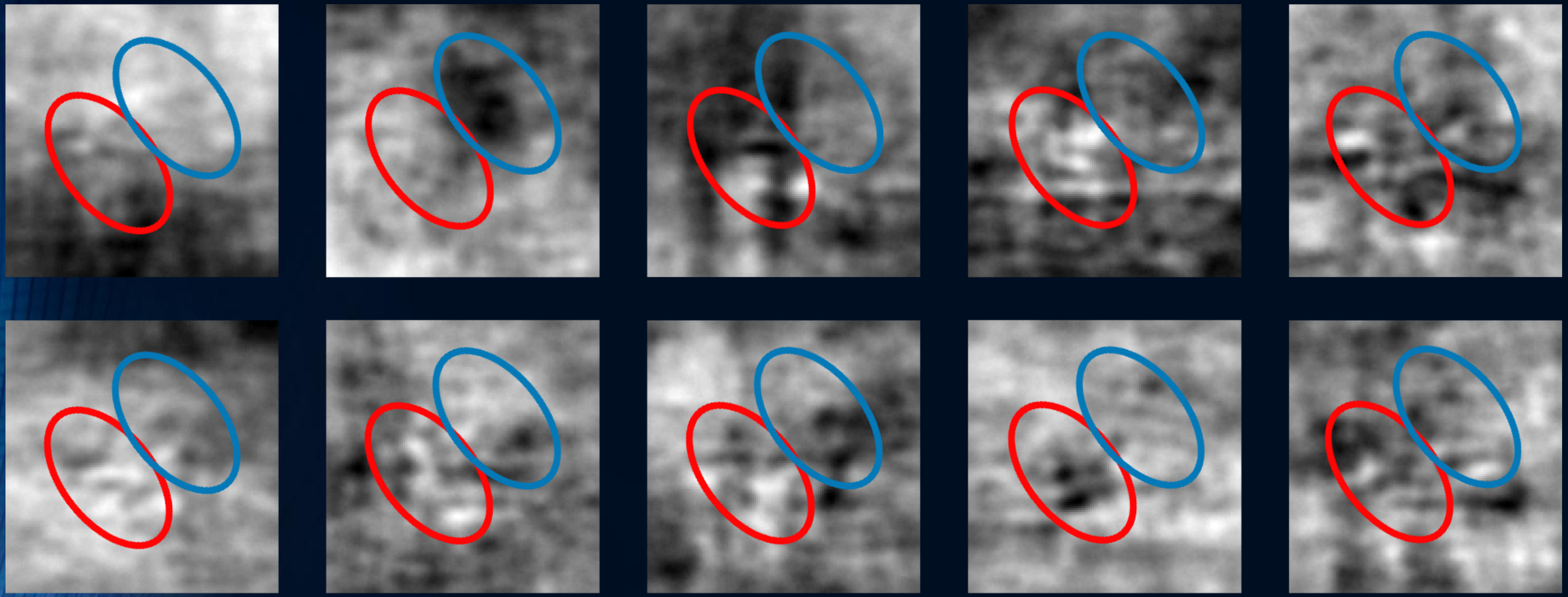


Gallant JL, Connor CE, Rakshit S, Lewis JW, Van Essen DC.
J Neurophysiol. 1996 Oct;76(4):2718-39.

Eigenvectors of second order V4 receptive field model



Eigenvectors of second order V4 receptive field model



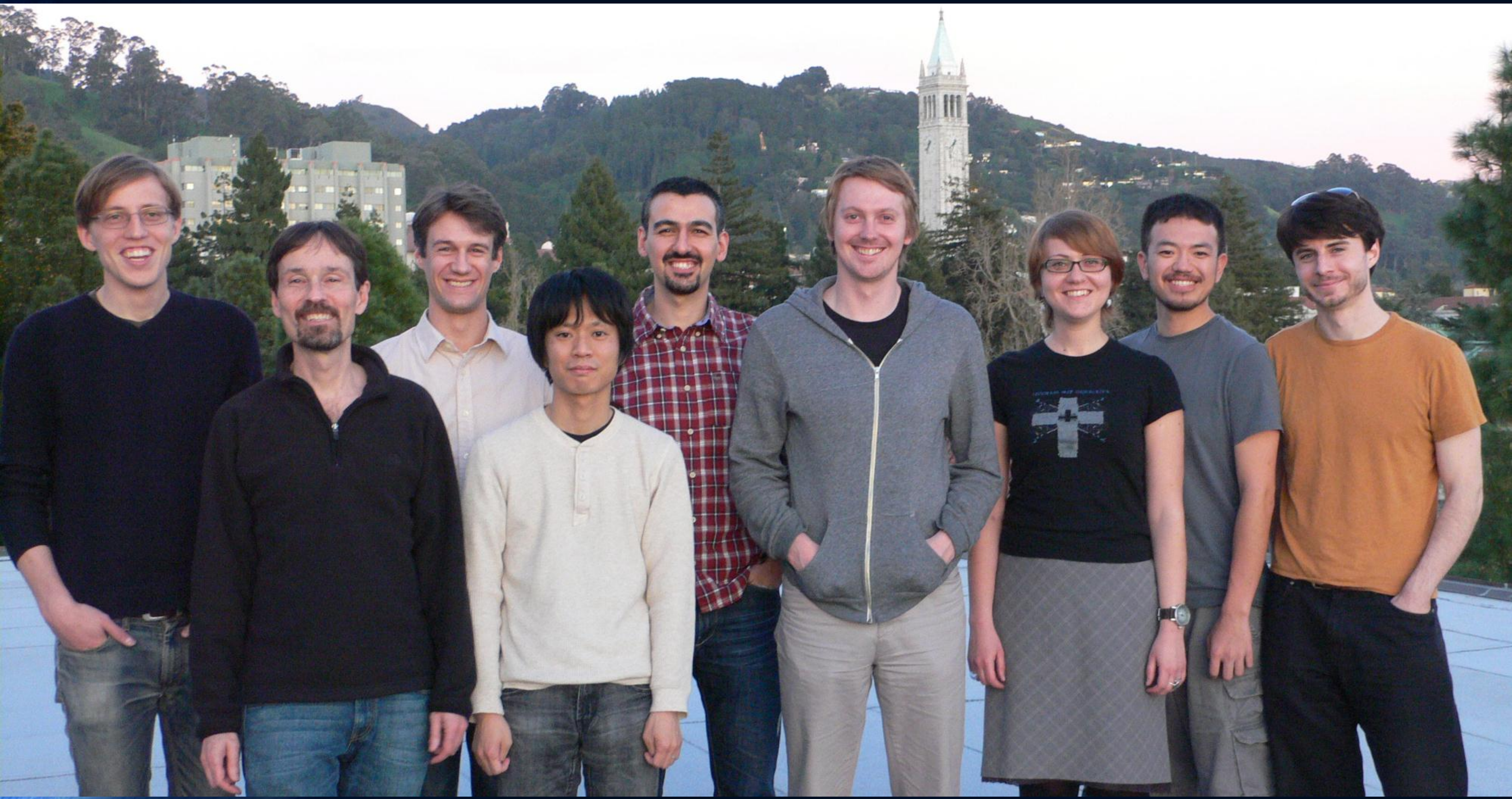
Shape tuning of a V4 cell's Volterra model

Shape tuning of a V4 cell's Volterra model

Embracing the Complexity

- Demonstrated a way to make this big problem tractable
- Shown many reported features of V4 tuning can exist in a single cell
- Interpretation of large models is still a major problem
- Need tensor libraries that exploit symmetry to decompose large models

Thank you!



V4 High Response Movie Frames

Stochastic Gradient Boosted IPKNs

- Use IPKNs as the weak learners
- Fit to sample of data using backprop w/ a stopping set
- Perform line search to determine step size that minimizes error on sample
- Multiply step size by learning rate and update function
- Ensemble is equivalent to a single Volterra model!

Some Important Unanswered Questions

- What information are our models missing in V2, V4 and beyond?
- Do we need nonlinear combinations of basis functions?
- Can we derive new basis functions?

Extracting Coefficients from Model

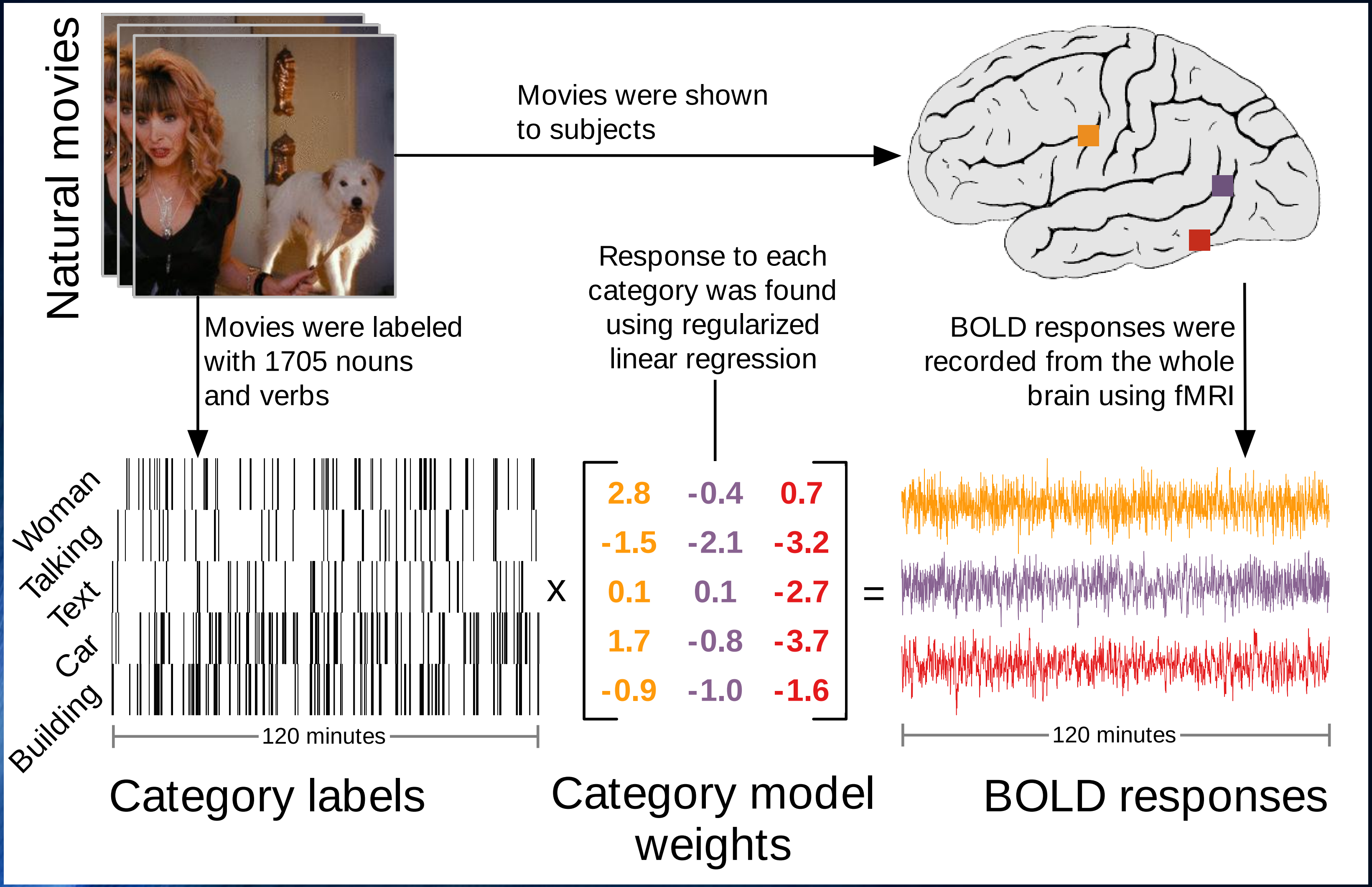
- Create a design matrix or the desired order of interactions from the support vectors/input weights
- Multiply by the output weights and weight by correction factor
- Extract coefficients from each iteration's network, weight by step size and sum to get final set of coefficients

$$\Phi_n = (\phi_n(x^1), \phi_n(x^2), \ldots, \phi_n(x^i))$$

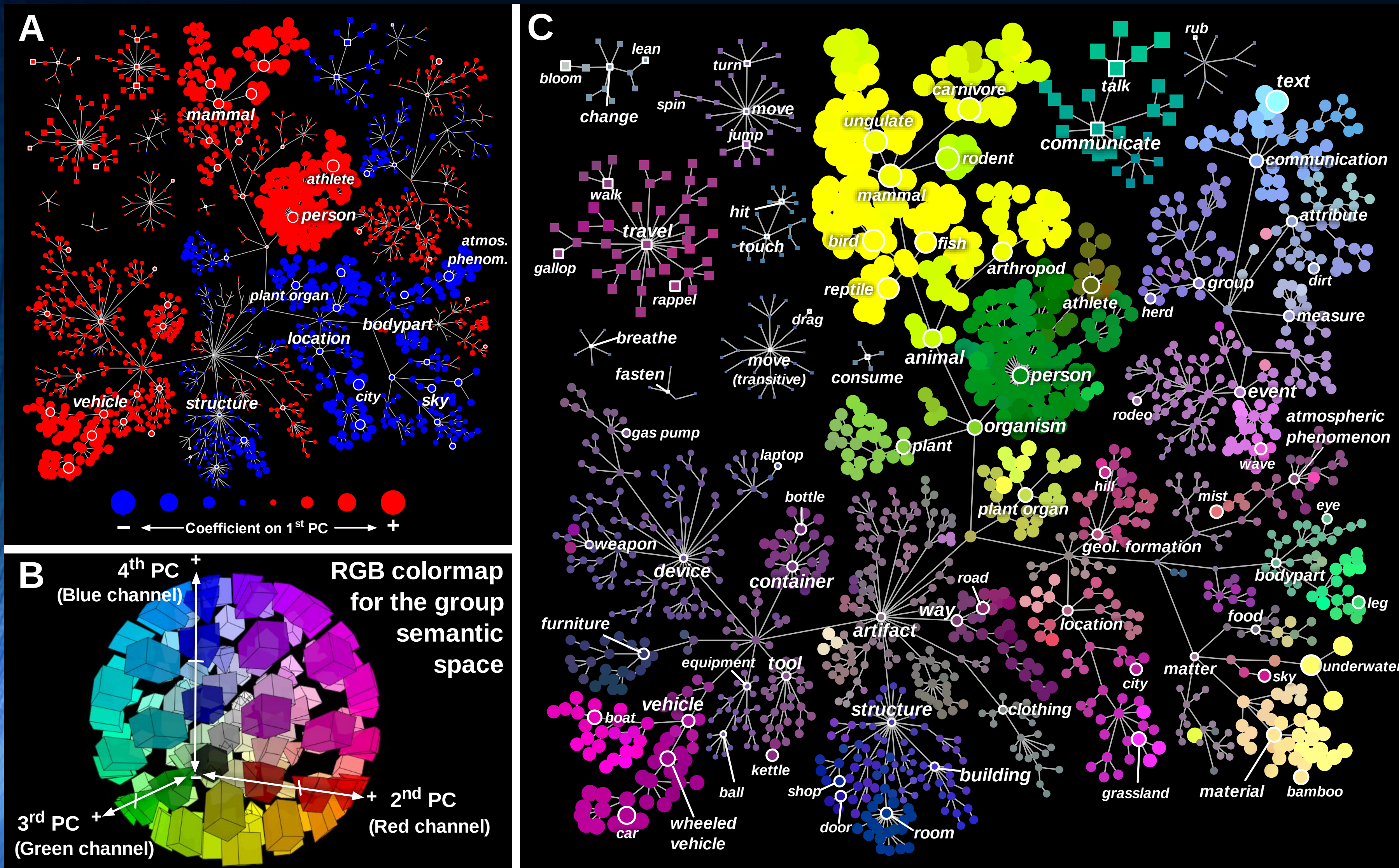
$$\eta_n = a_n \Phi_n^T K^{-1} y$$

$$\eta_n = a_n \Phi_n^T \alpha$$

Feature Spaces

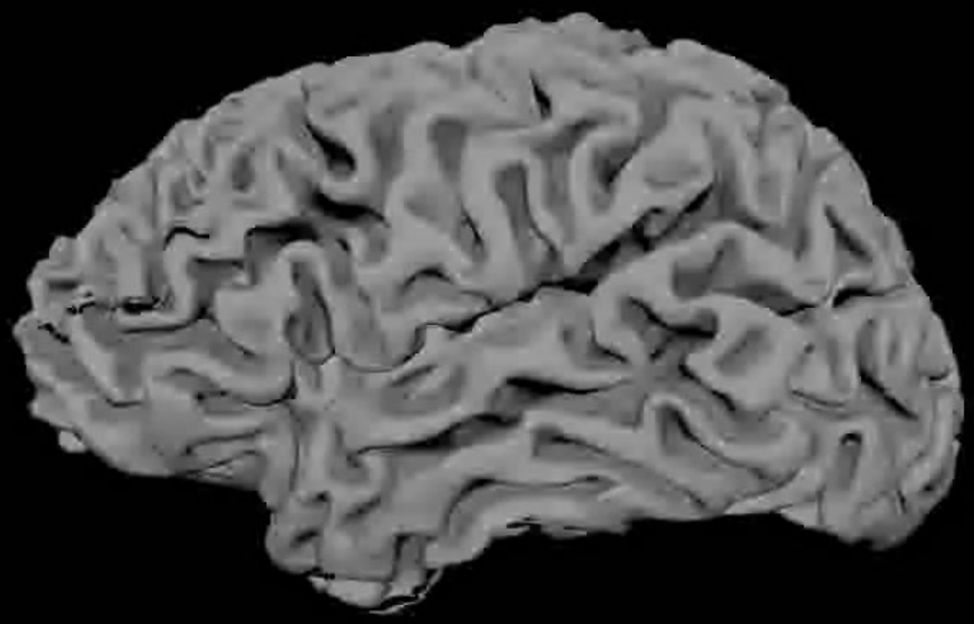


Constructing a Semantic Space



Semantic Decoding

Flattening the Brain



Visualizing Semantic Space