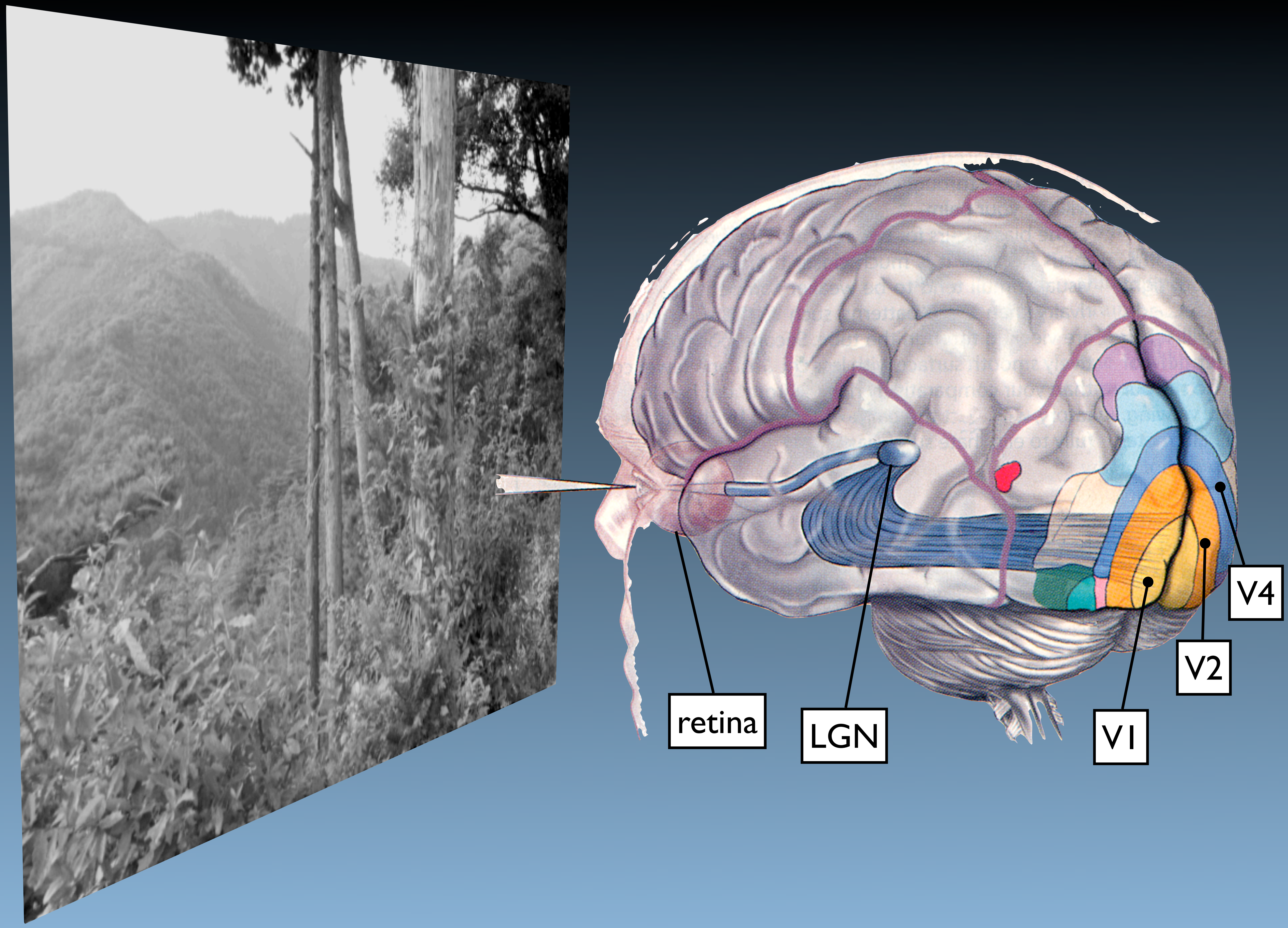
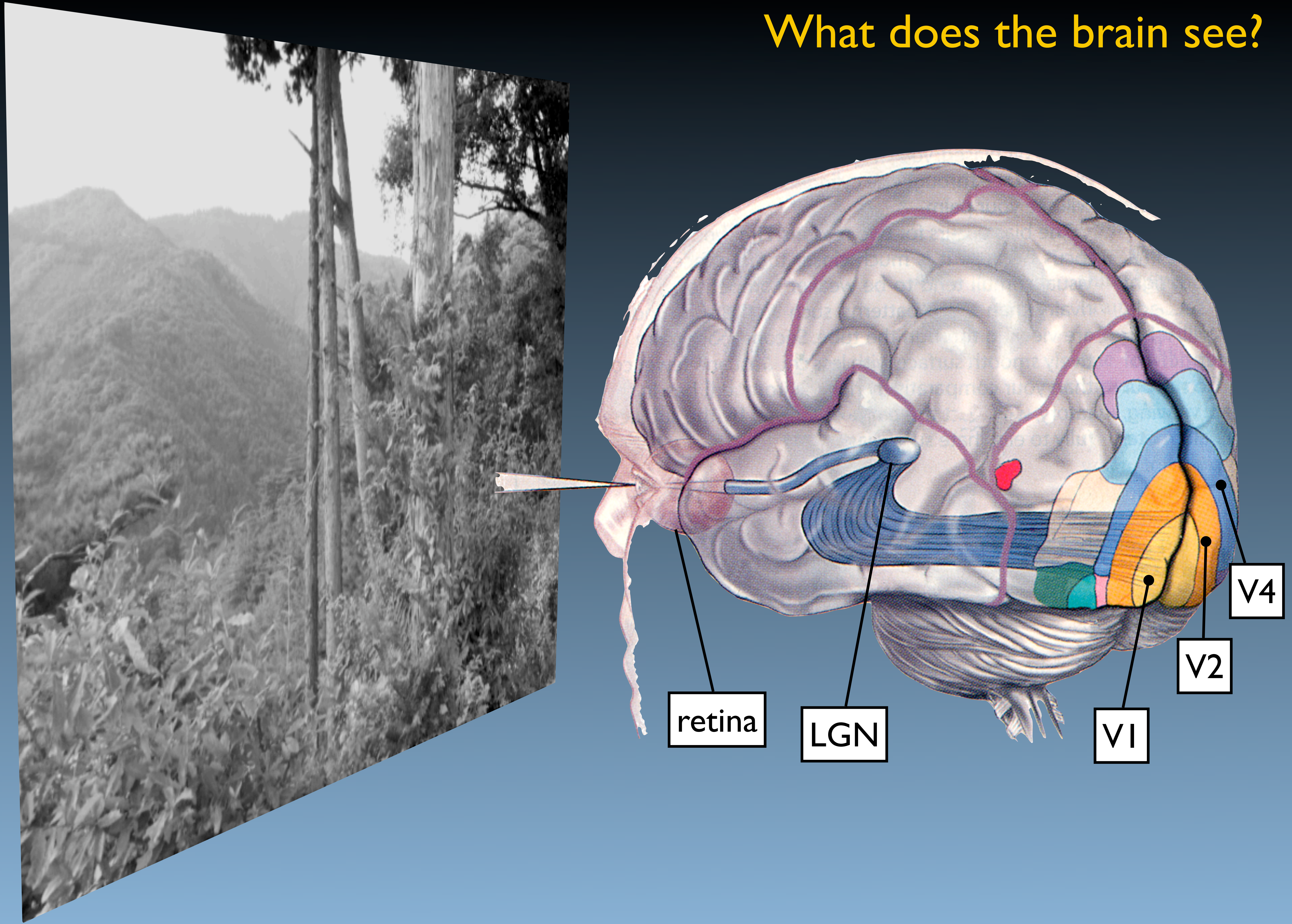
Learning hierarchical representations of natural images

Mike Lewicki

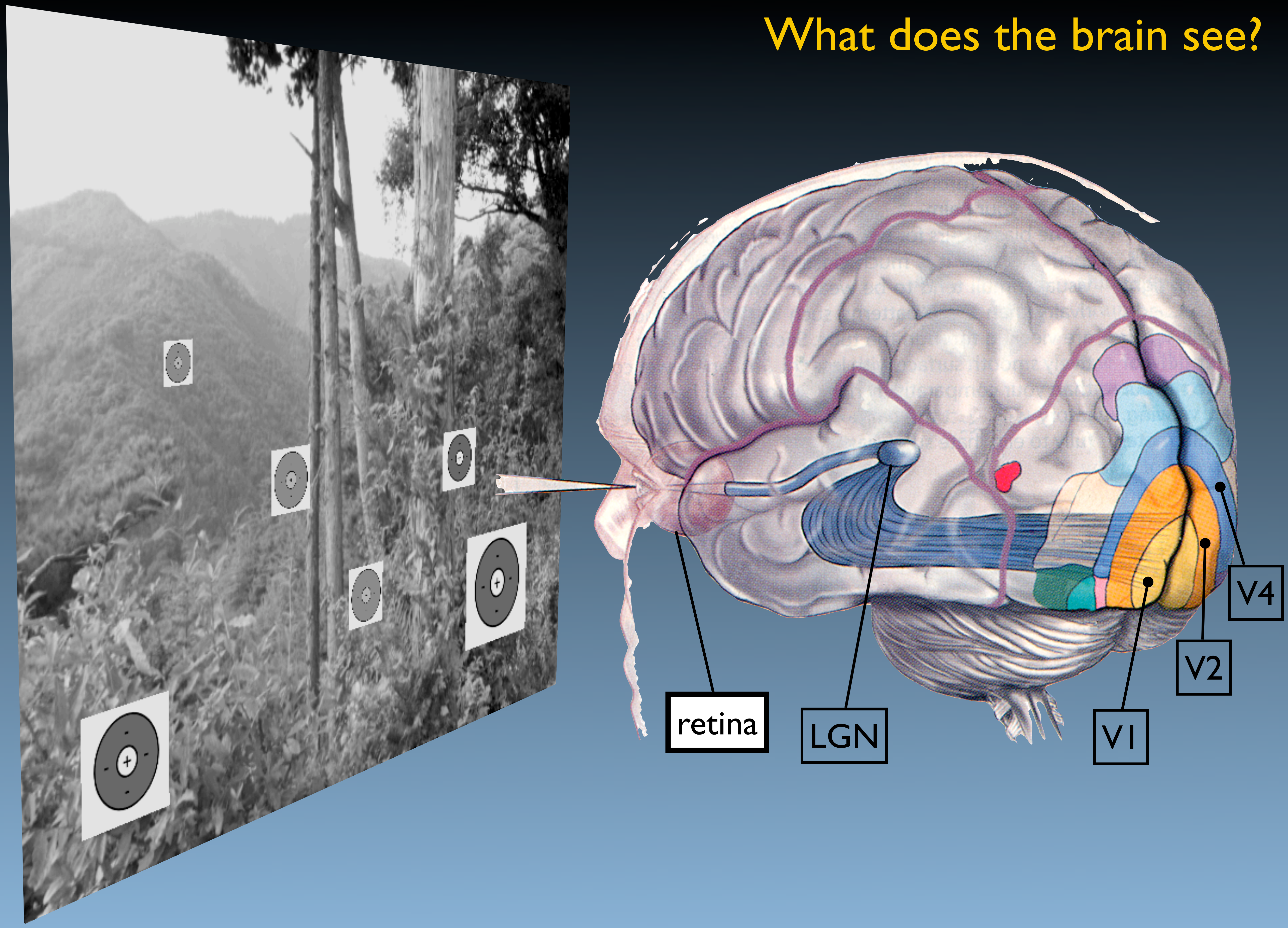
Computer Science Department &
Center for the Neural Basis of Cognition
Carnegie Mellon University


retina
LGN
V1
V2
V4

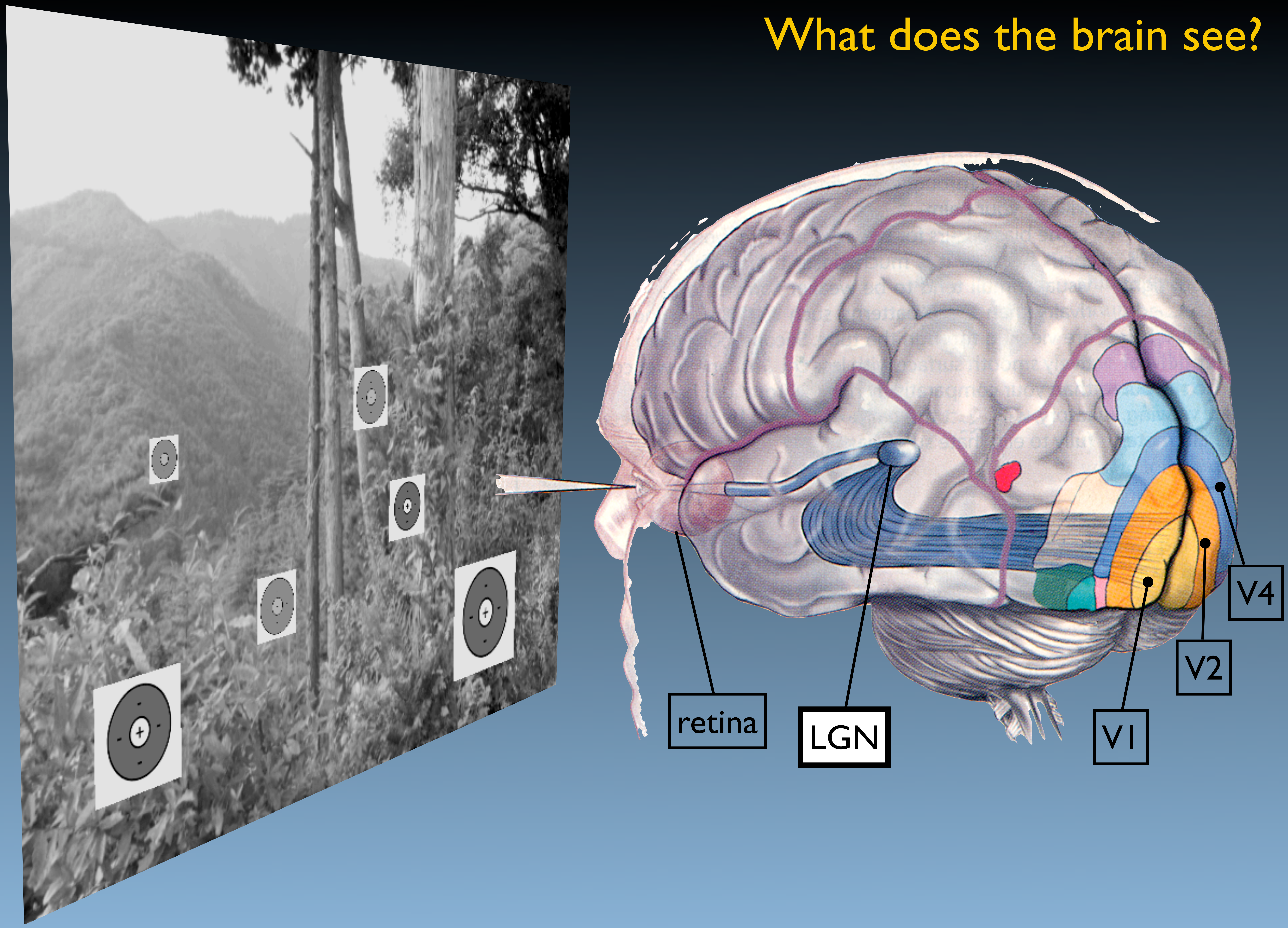
What does the brain see?



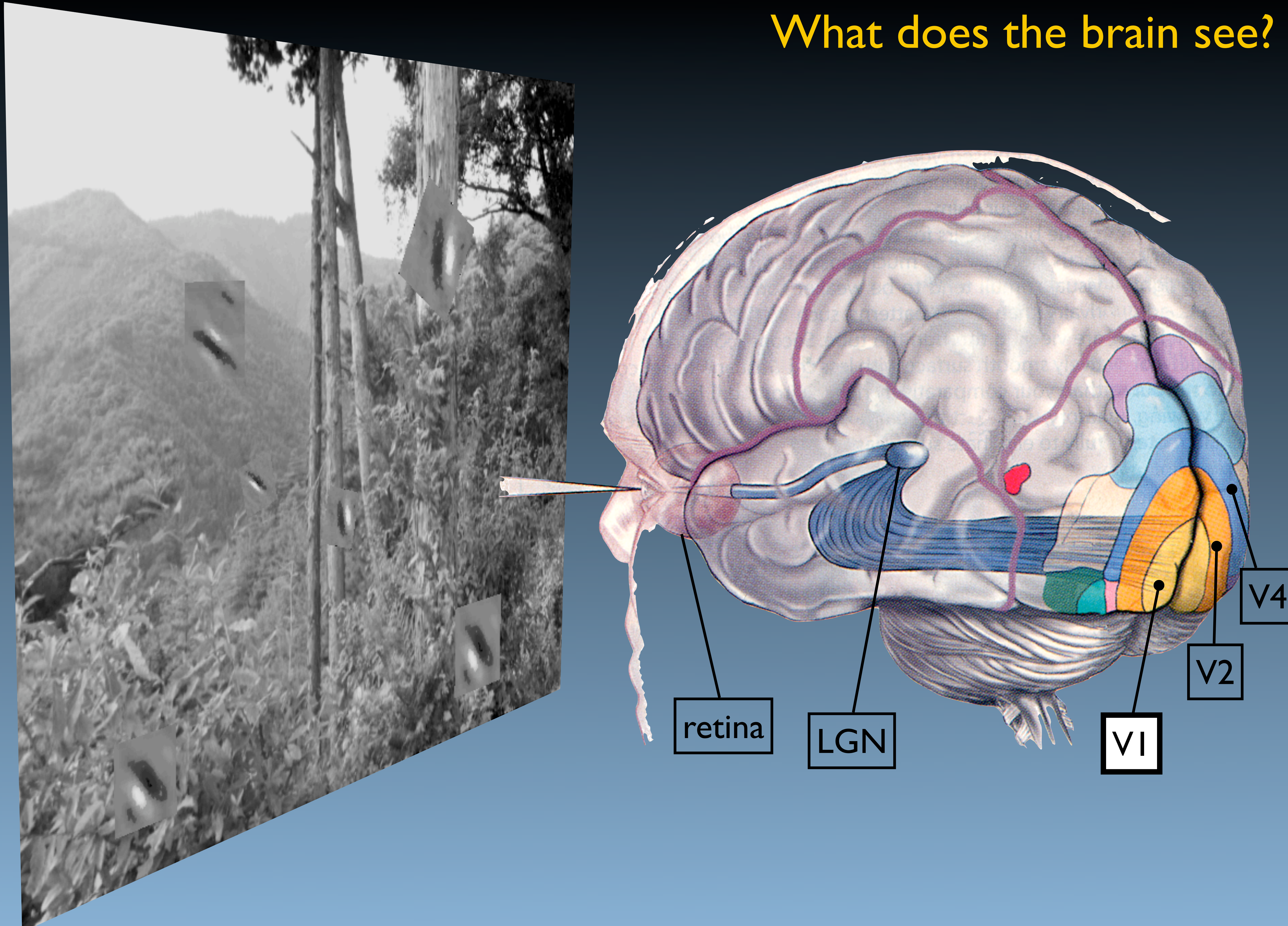
What does the brain see?



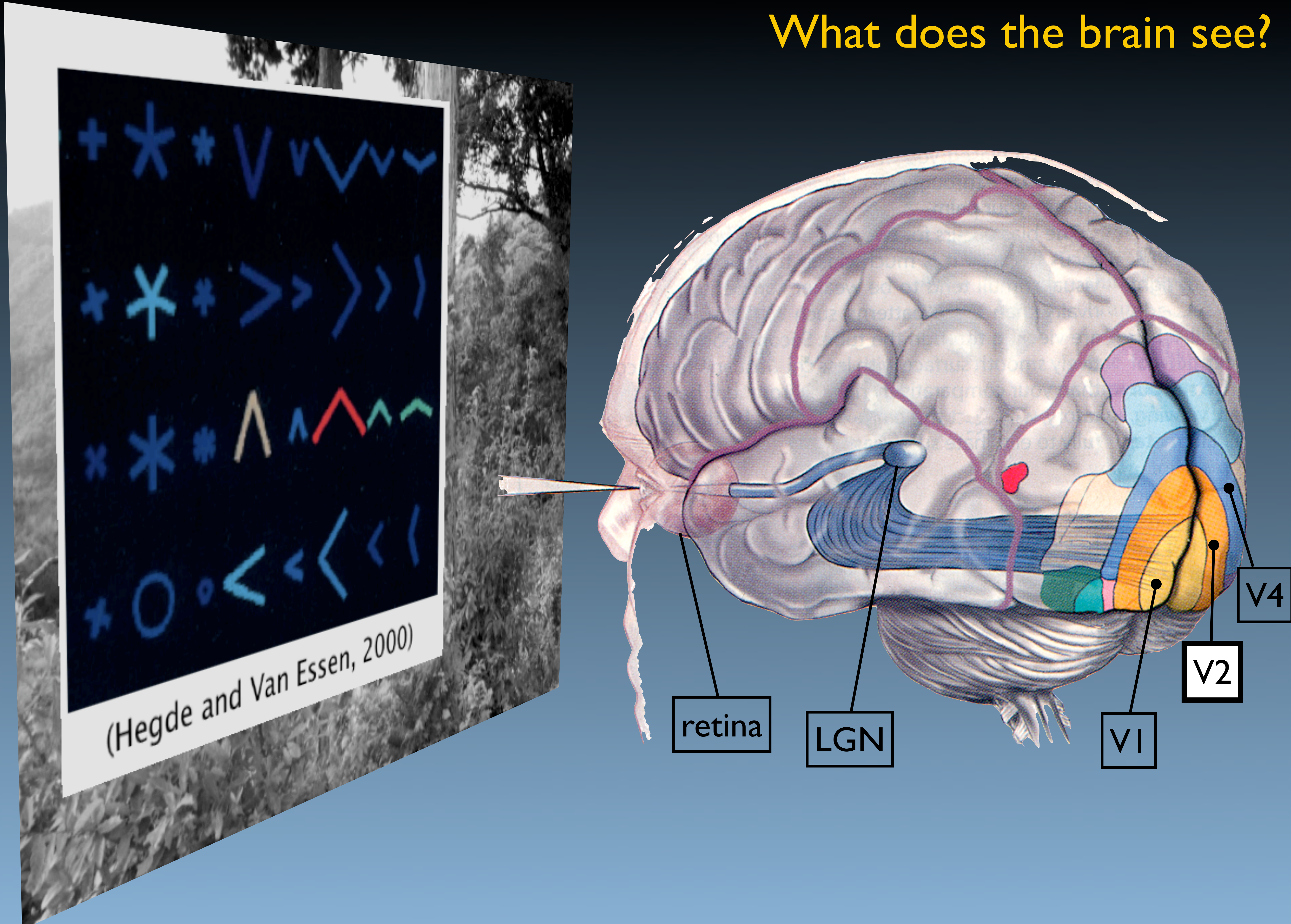
What does the brain see?



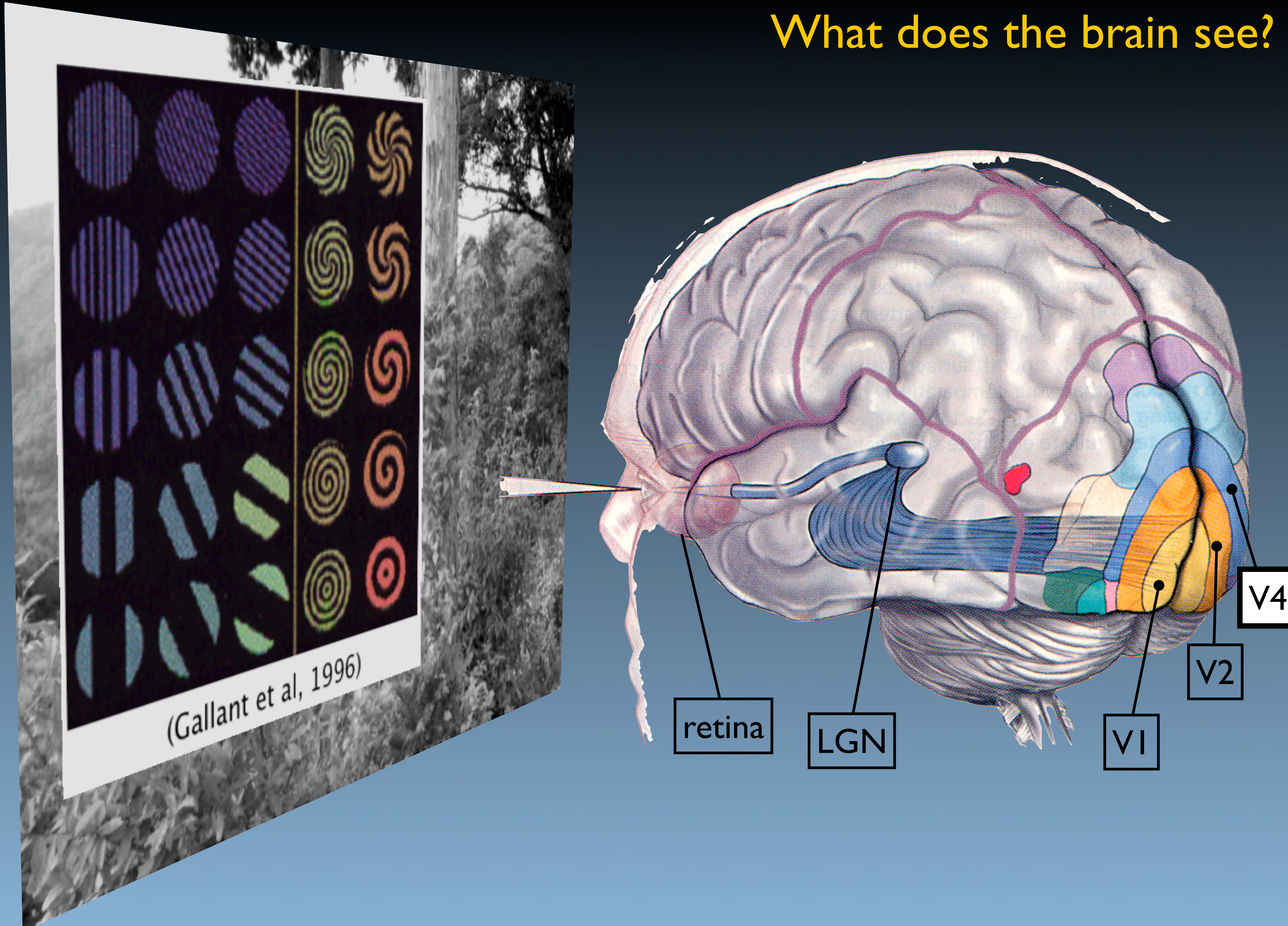
What does the brain see?

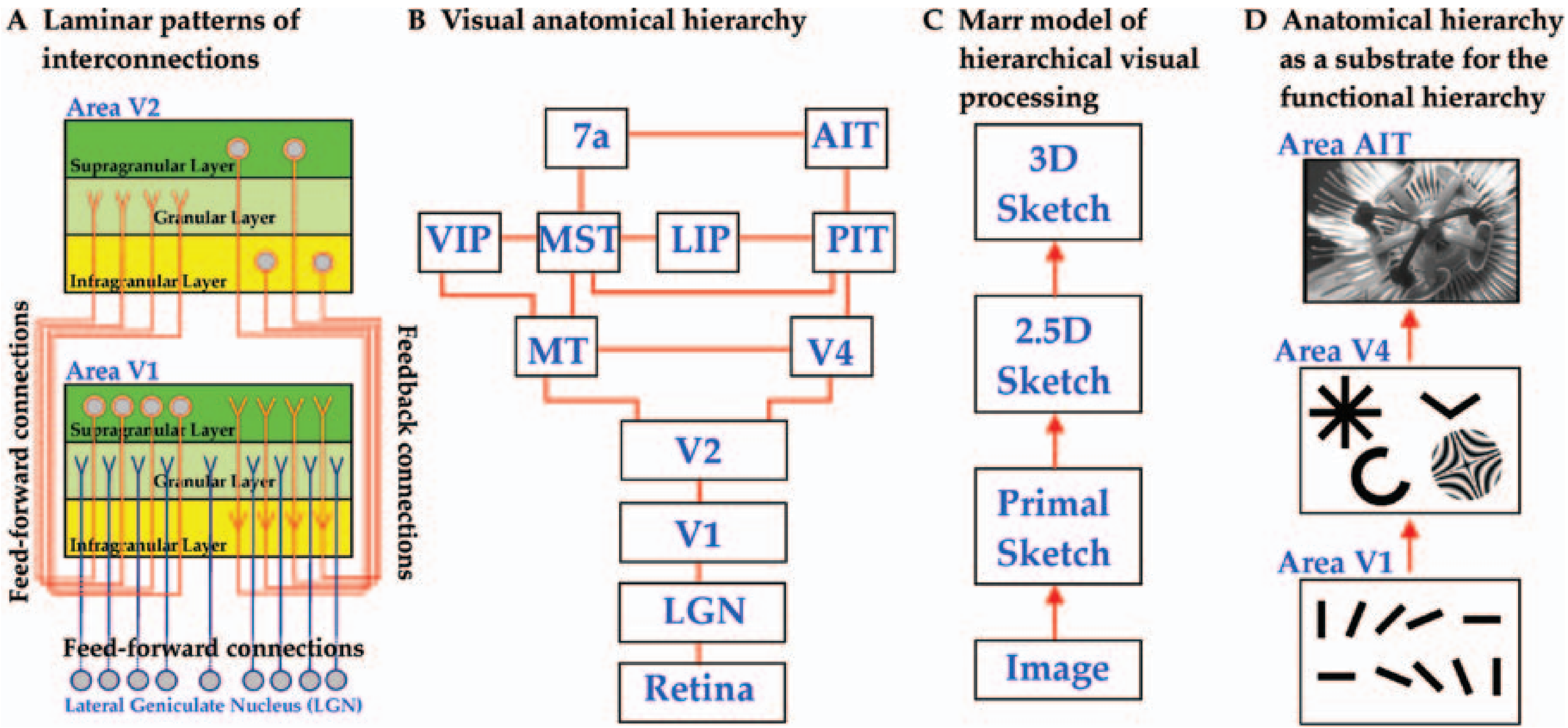


What does the brain see?

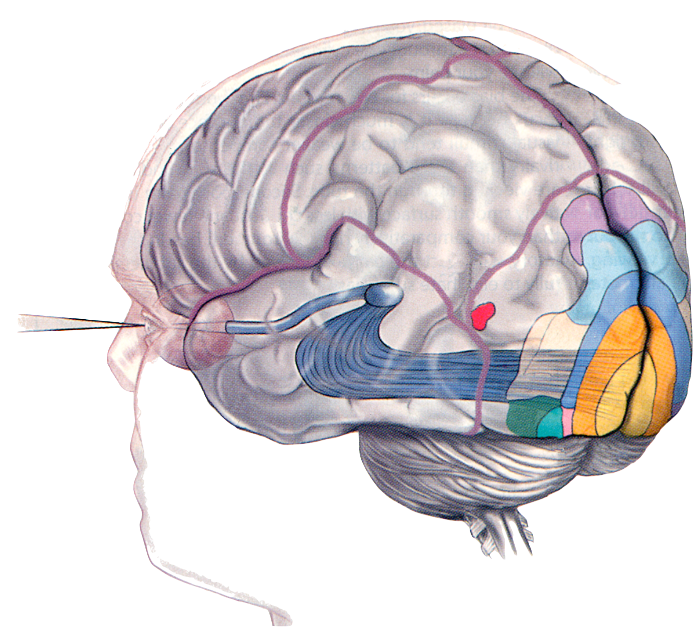


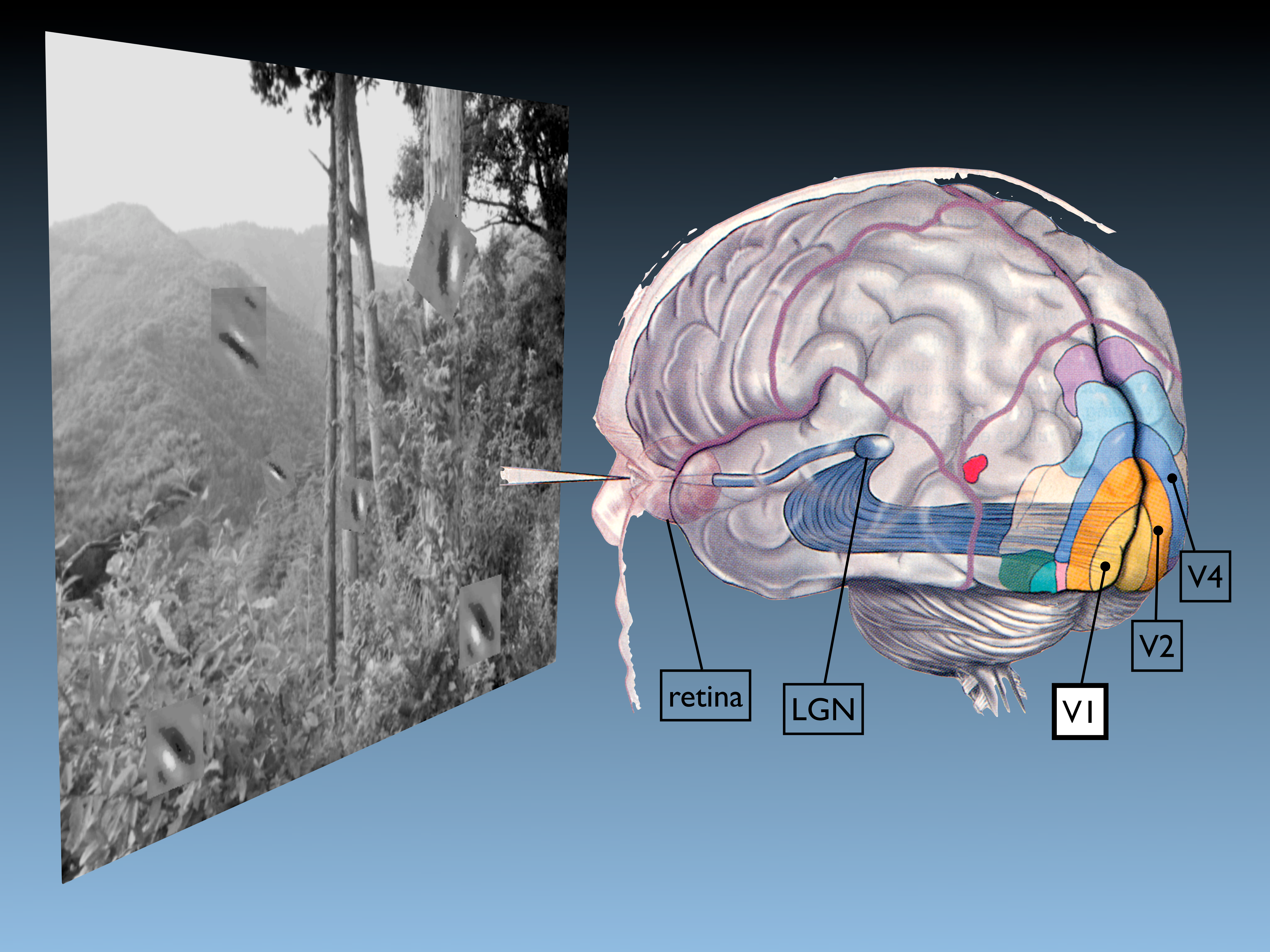
What does the brain see?





from Hedgé and Felleman, 2007




retina
LGN
V1
V2
V4

Oriented Gabor models of individual simple cells

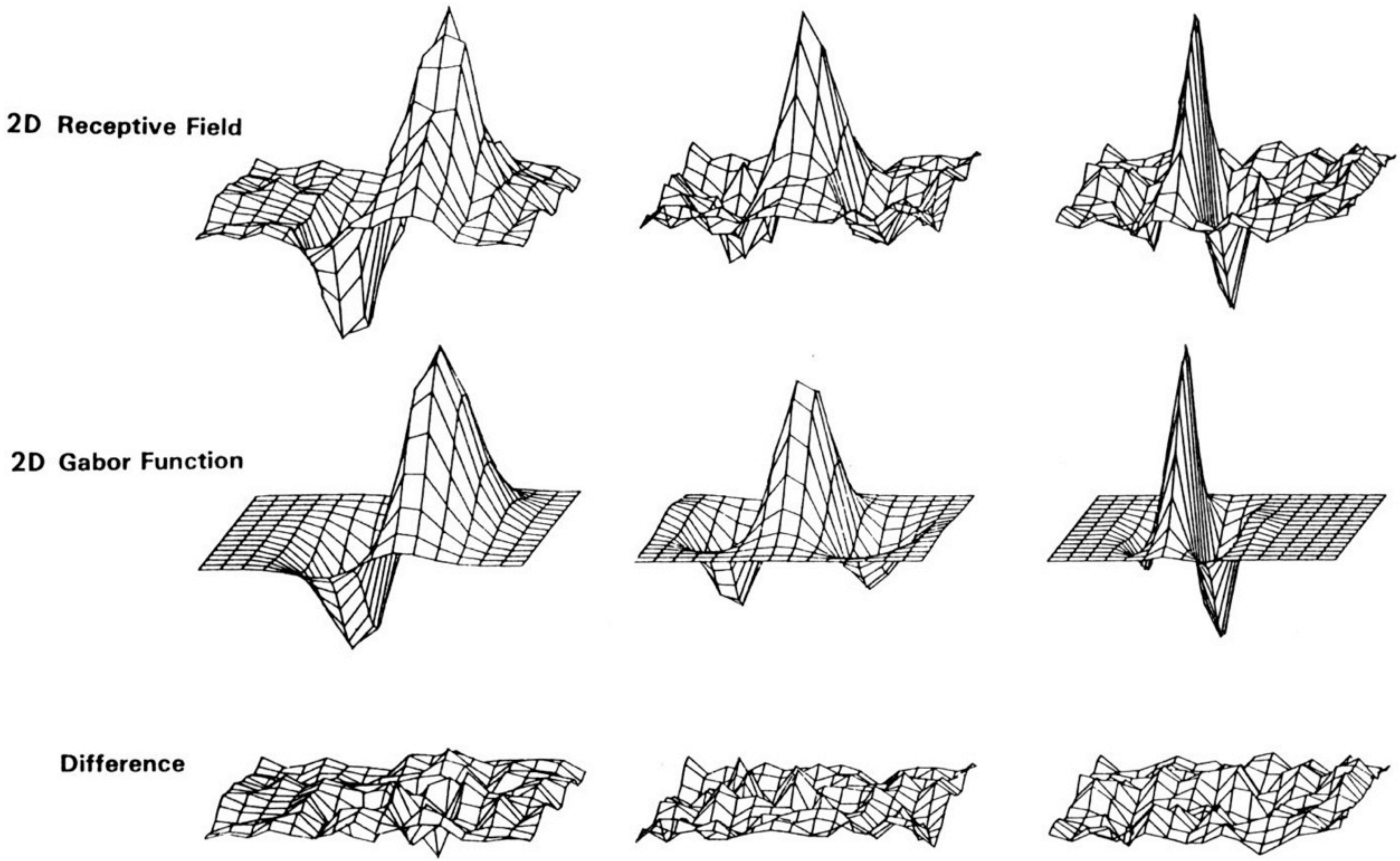


figure from Daugman, 1990; data from Jones and Palmer, 1987

A theoretical approach

- Describe computational function:

  What problems does it need to solve?

- Abstract from the details:

  Incorporate important constraints.

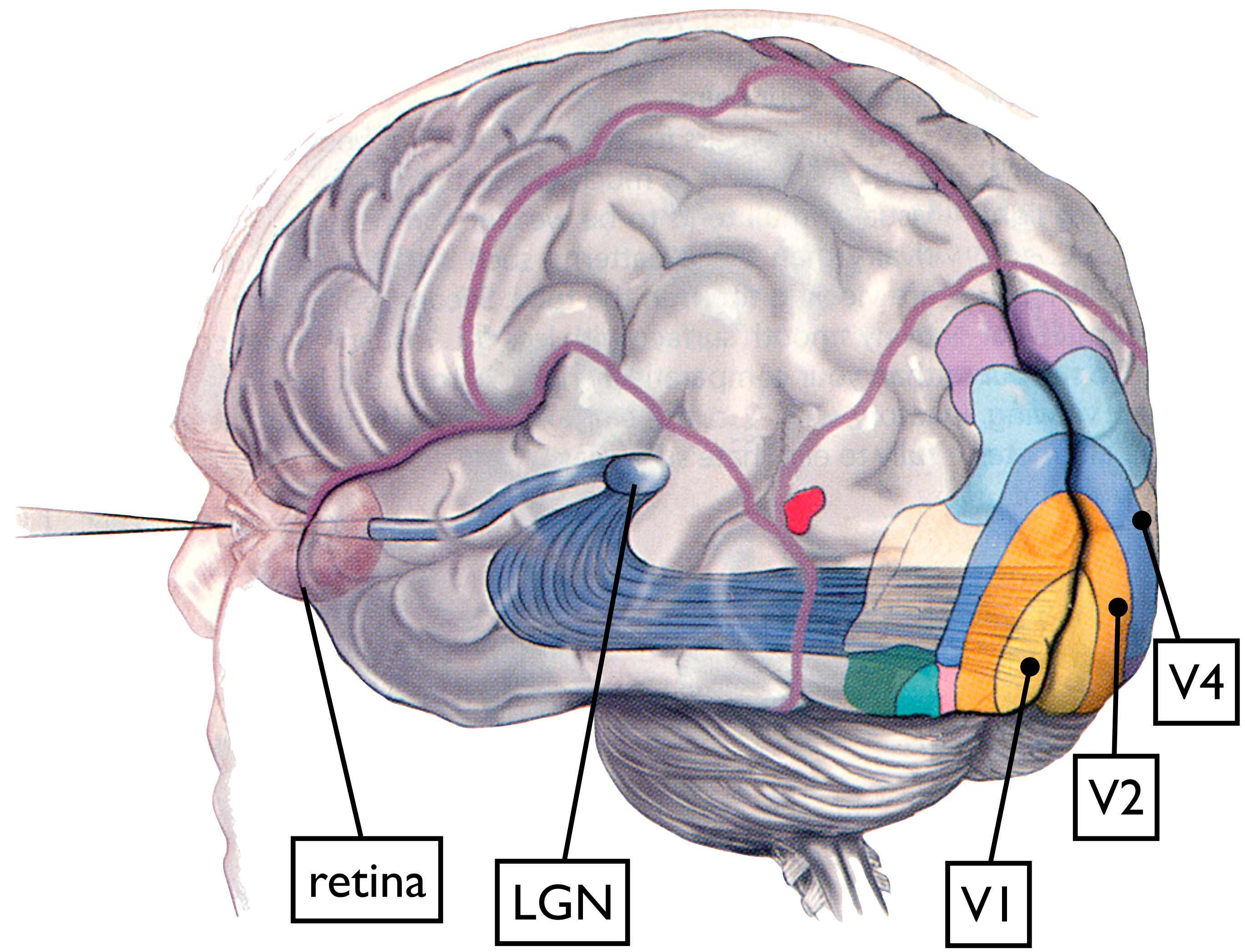
- Demonstrate performance:

  Should be optimal for general images.

- Explain neural data:

  Predict from theoretical principles.

- Models are bottom-up;
  theories are top-down.

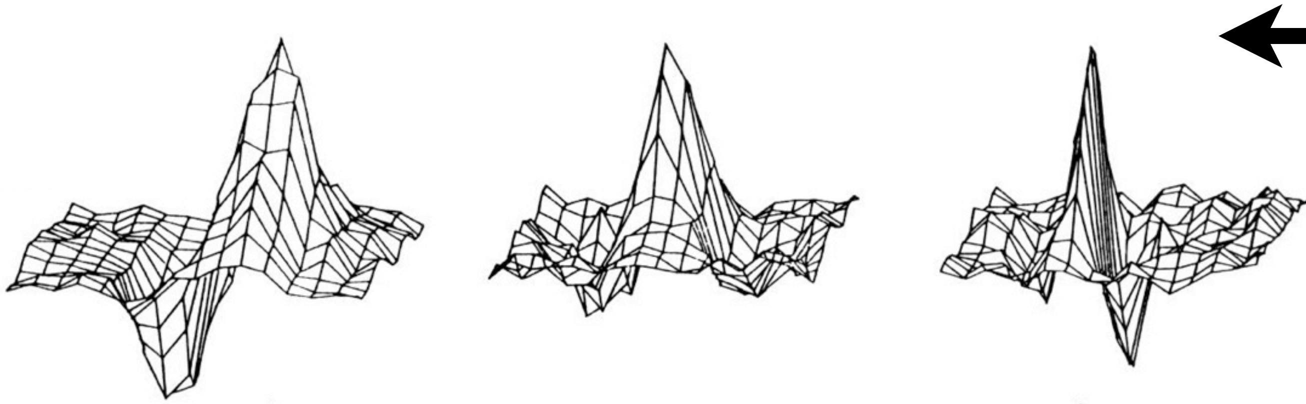


Make theoretical predictions from the natural environment

natural environment

evolution
(or learning)

physiological data:



theory

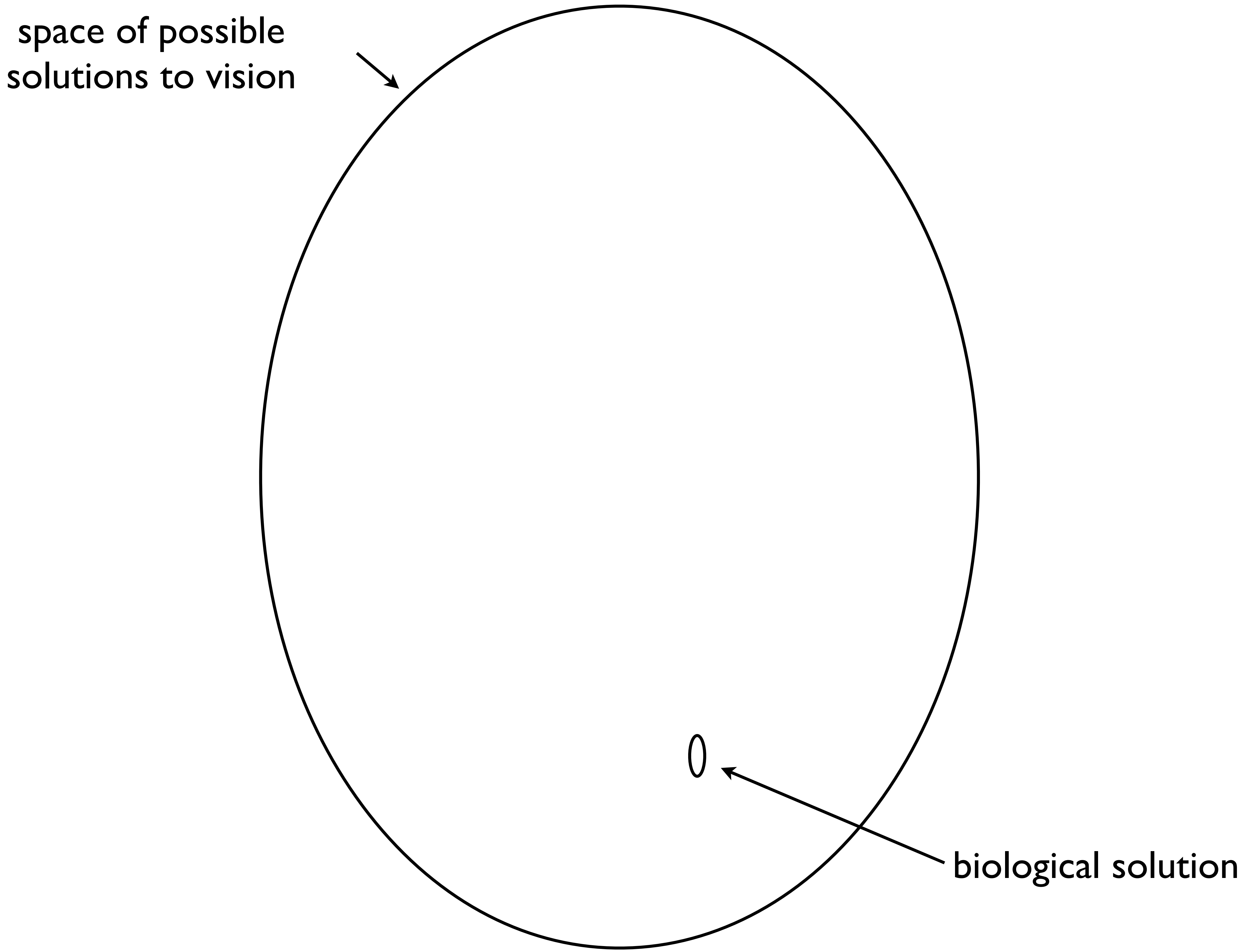
optimal model:
- properties
- performance

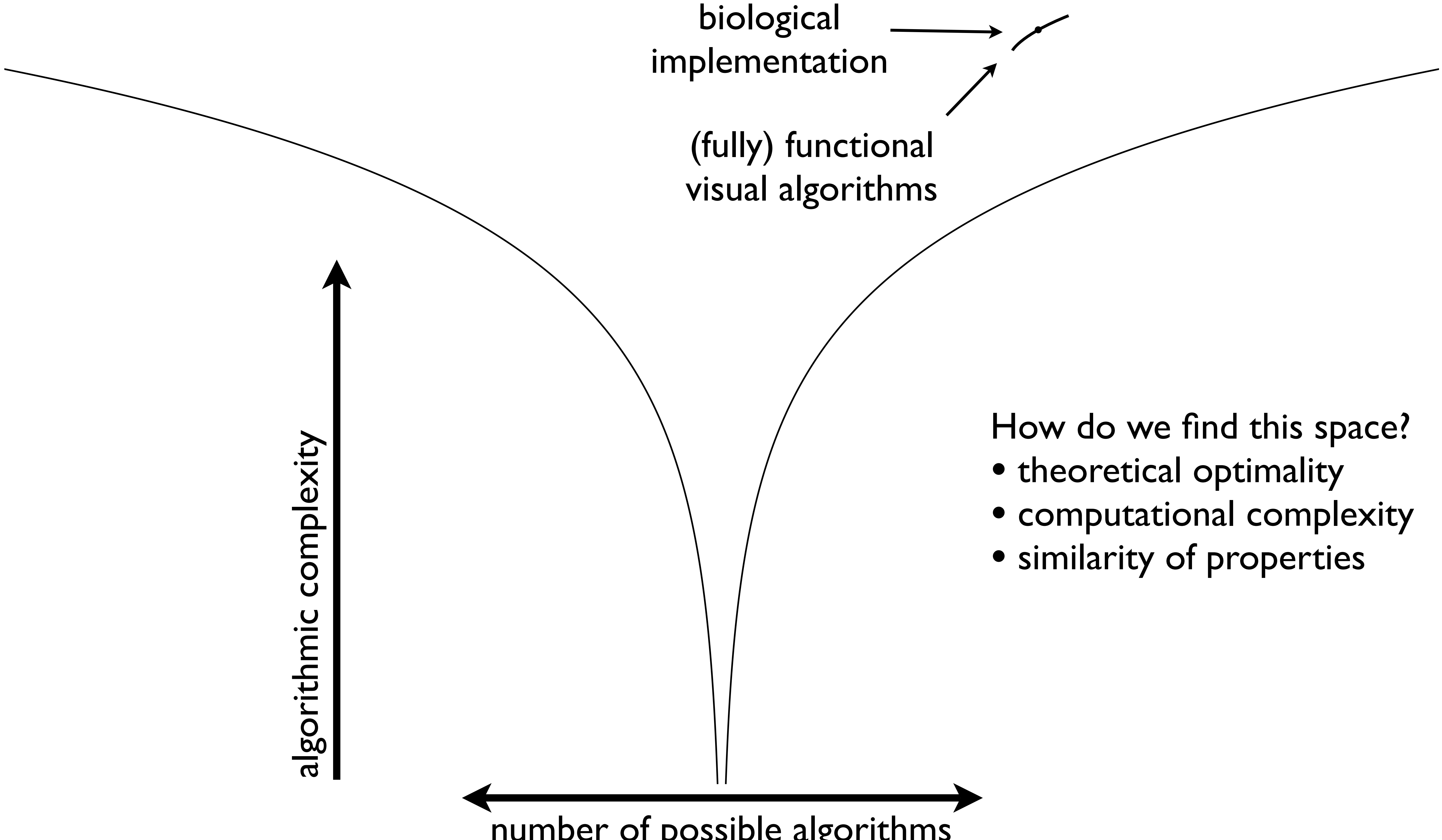
?

only compare to the data *after* optimizing
do not fit the data

Prediction depends on data,
computational goal, and constraints

Is this possible?


space of possible
solutions to vision
biological solution



How do we find this space?

- theoretical optimality
- computational complexity
- similarity of properties

A wing would be a most mystifying structure if one did not know that birds flew.

Horace Barlow, 1961

An algorithm is likely to be understood more readily by understanding the nature of the problem being solved than by examining the mechanism in which it is embodied.

David Marr, 1982

Representing structure in natural images

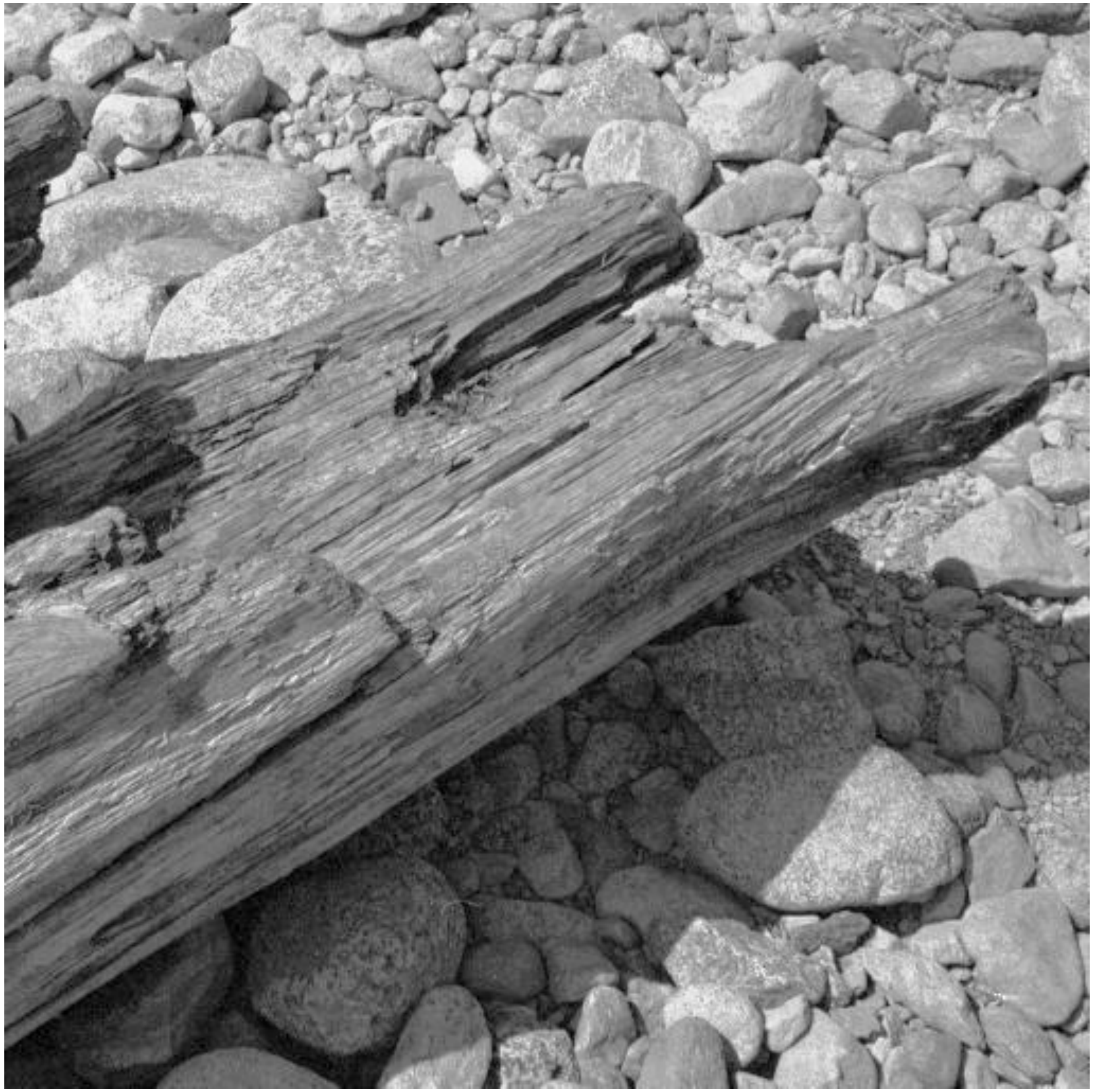


image from Field (1994)

Representing structure in natural images

image from Field (1994)

Representing structure in natural images

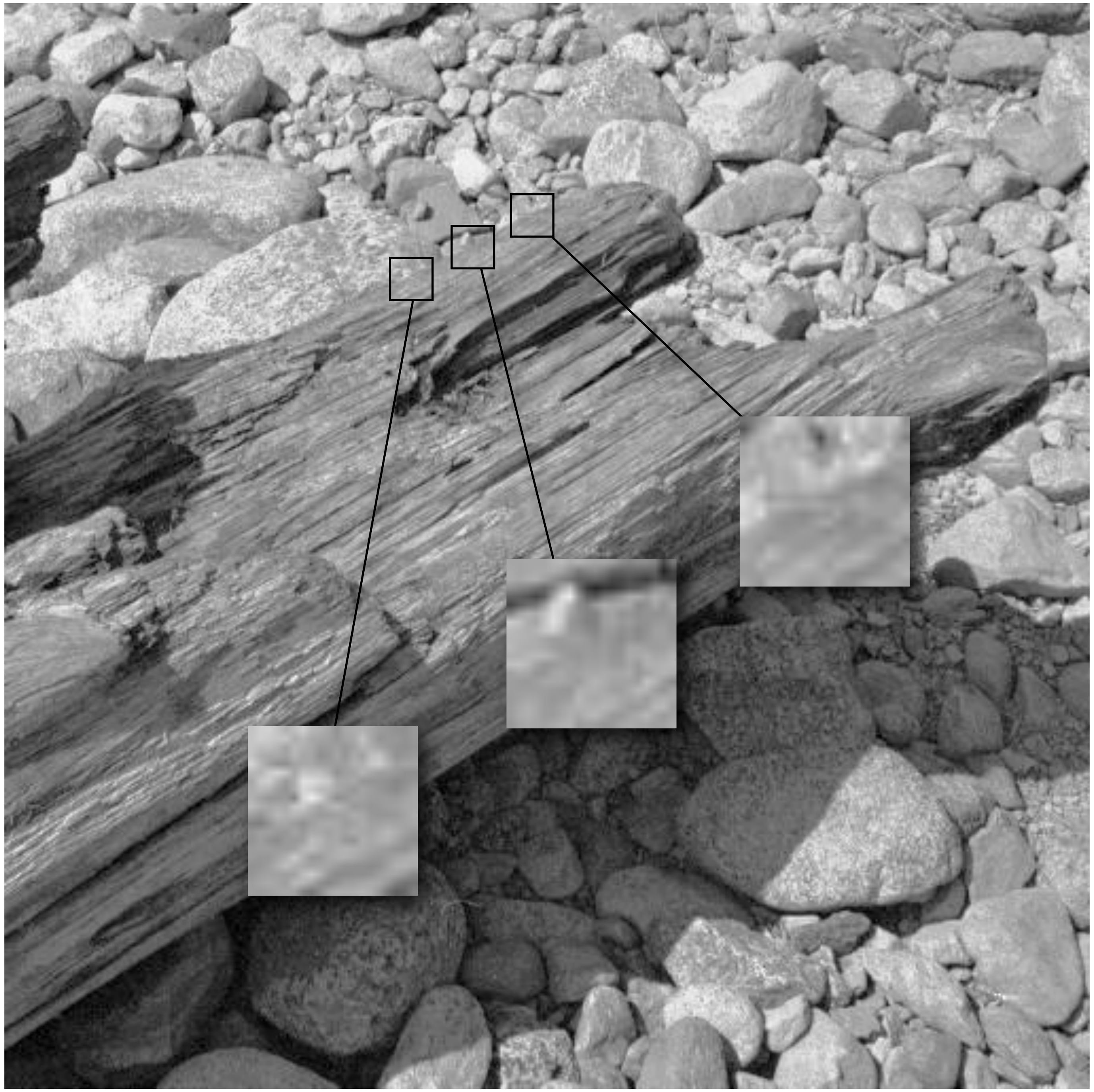


image from Field (1994)

Theory: *Efficient coding of natural images*

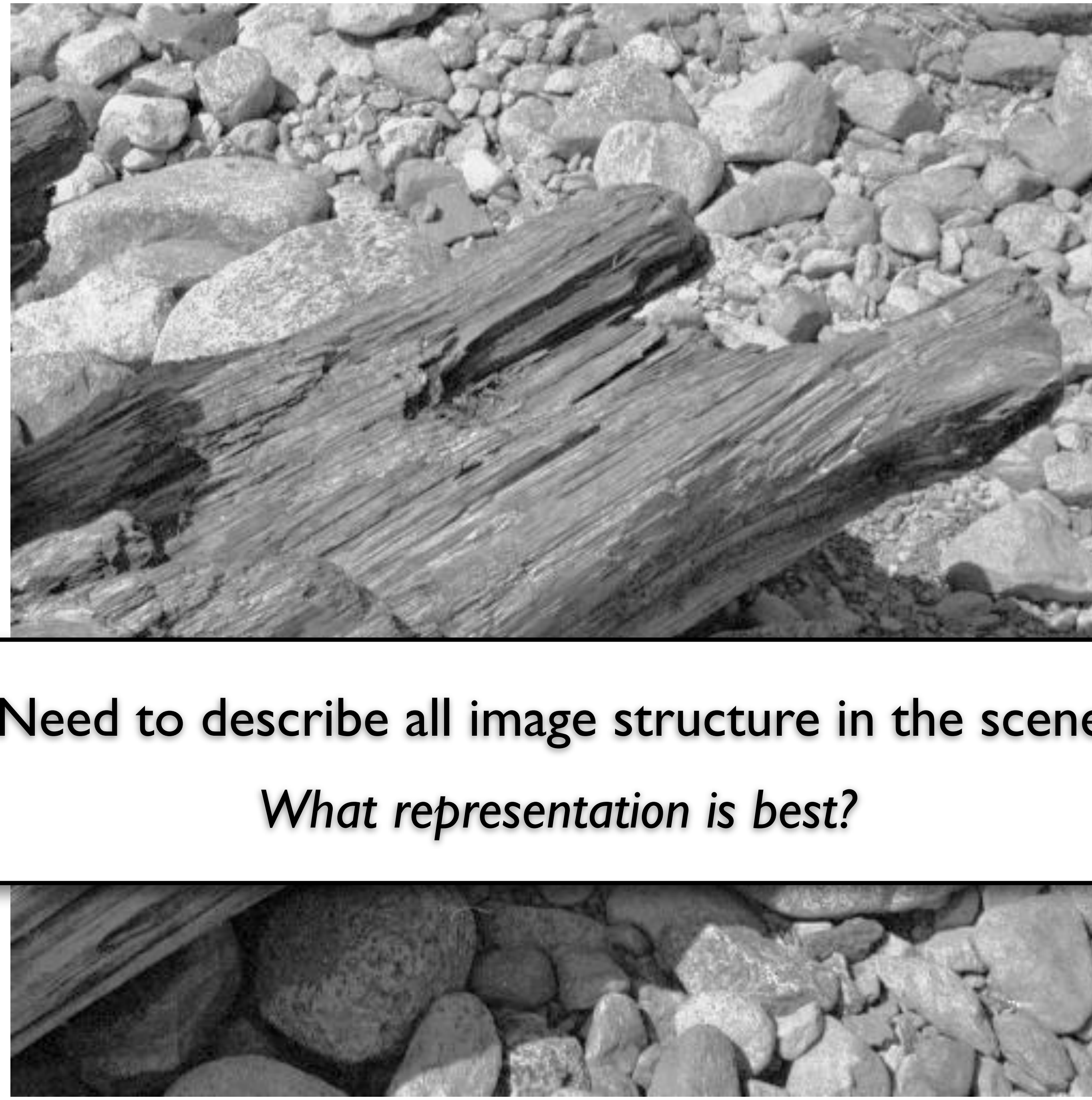


image from Field (1994)

Need to describe all image structure in the scene.

What representation is best?

Describing signals with a simple statistical model

Principle

Good codes capture the statistical distribution of sensory patterns.

How do we describe the distribution?

- Goal is to encode the data to desired precision

$$\begin{aligned}\mathbf{x} &= \vec{a}_1 s_1 + \vec{a}_2 s_2 + \cdots + \vec{a}_L s_L + \vec{\epsilon} \\ &= \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}\end{aligned}$$

- A filter bank description:

$$\hat{\mathbf{s}} = \mathbf{A}^{-1}\mathbf{x} = \mathbf{W}\mathbf{x}$$

Sparse coding

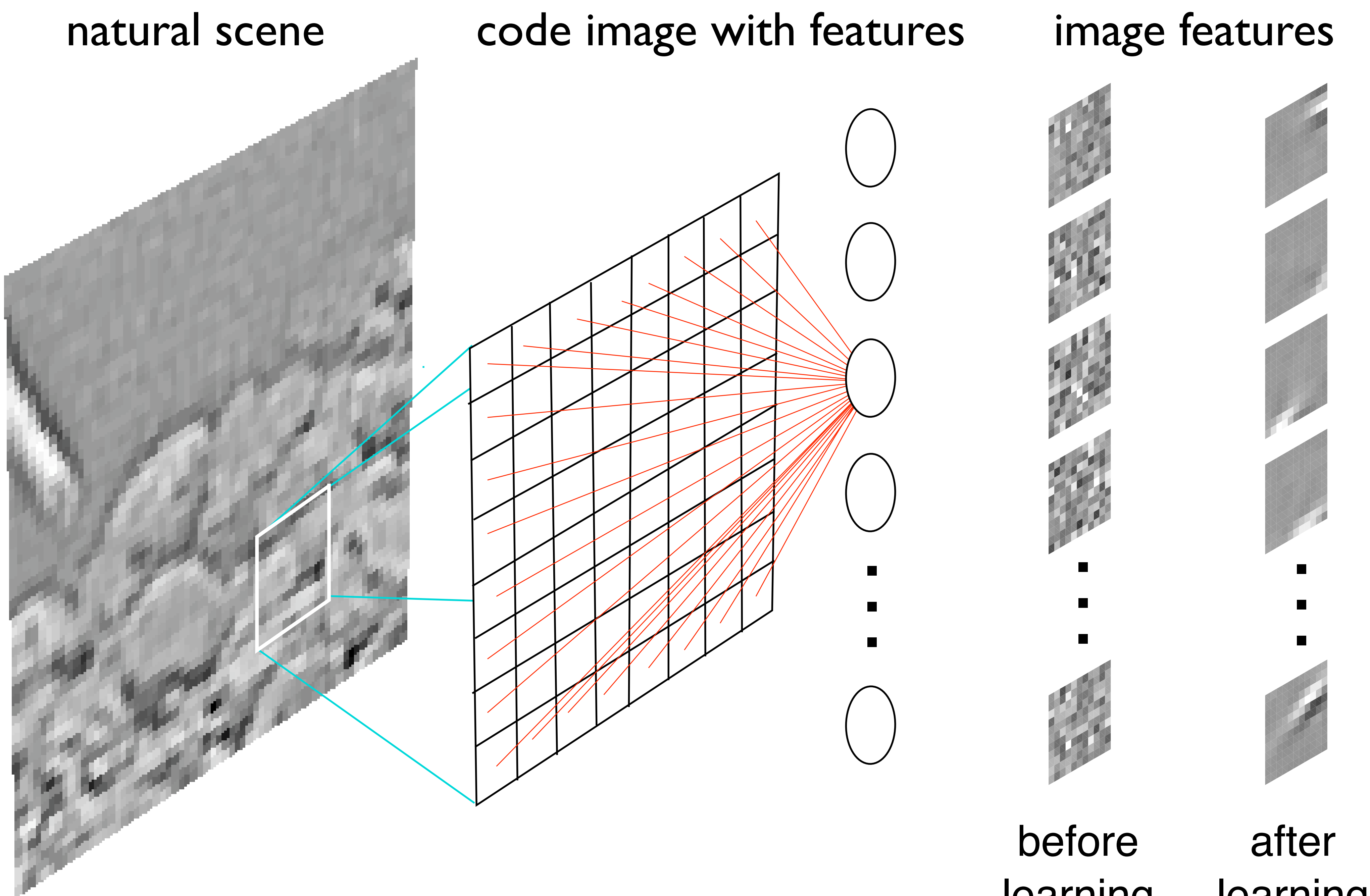
- To learn the optimal codes we optimize two terms:

$$E = -[\text{preserve information}] - \lambda[\text{sparseness of } s_i]$$

- In terms of equations:

$$E = -\sum_n \left[\mathbf{x}_n - \mathbf{A}\mathbf{s}_n\right]^2 - \sum_i S\left(\frac{s_i}{\sigma}\right)$$

- Minimizing this expression finds adapts the image features in **A** to natural images

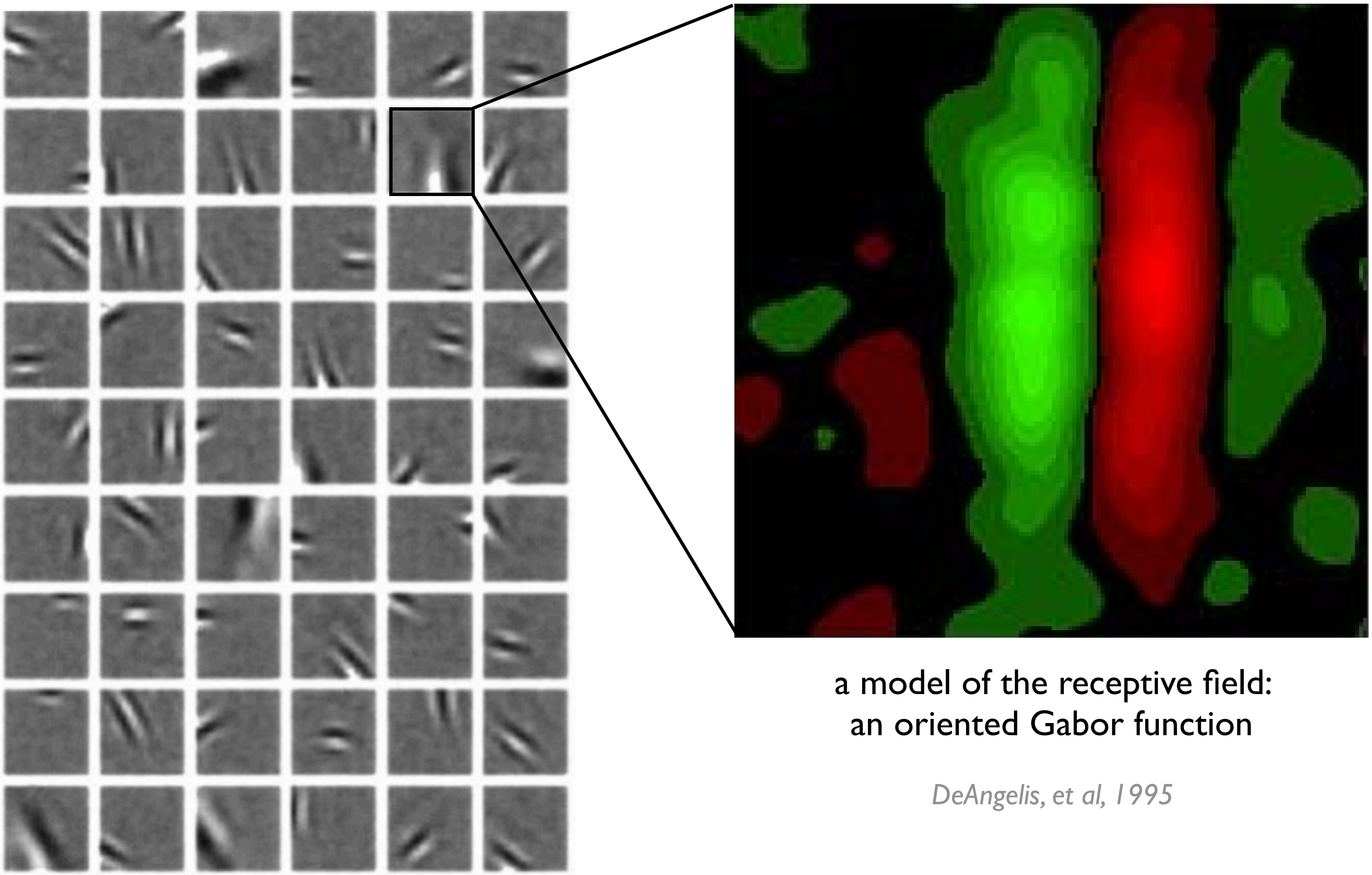


Features optimize coding efficiency:

- minimizes redundancy
- maximizes independence

Olshausen and Field, 1996

Efficient coding theory predicts V1 receptive fields



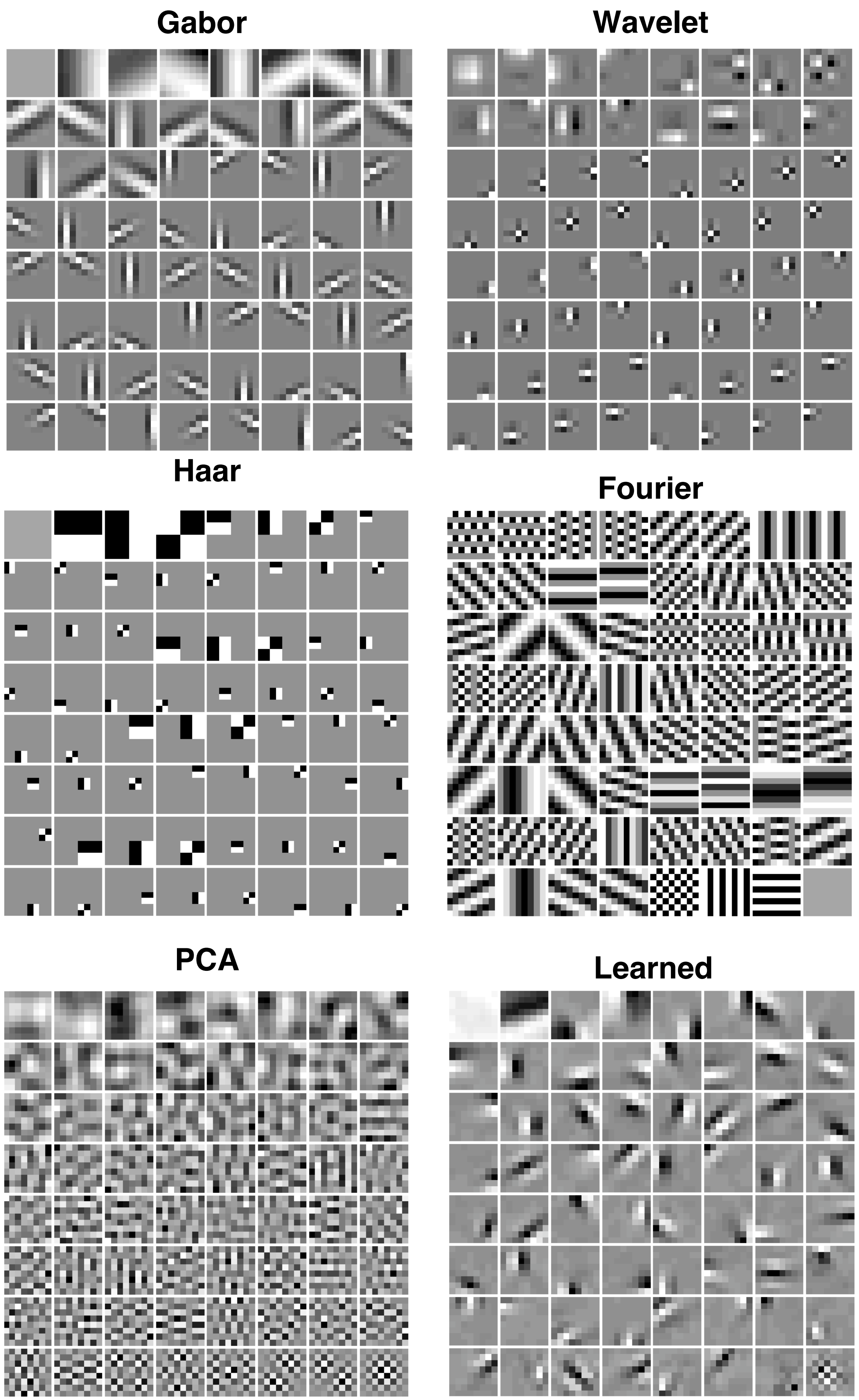
Olshausen and Field, 1996

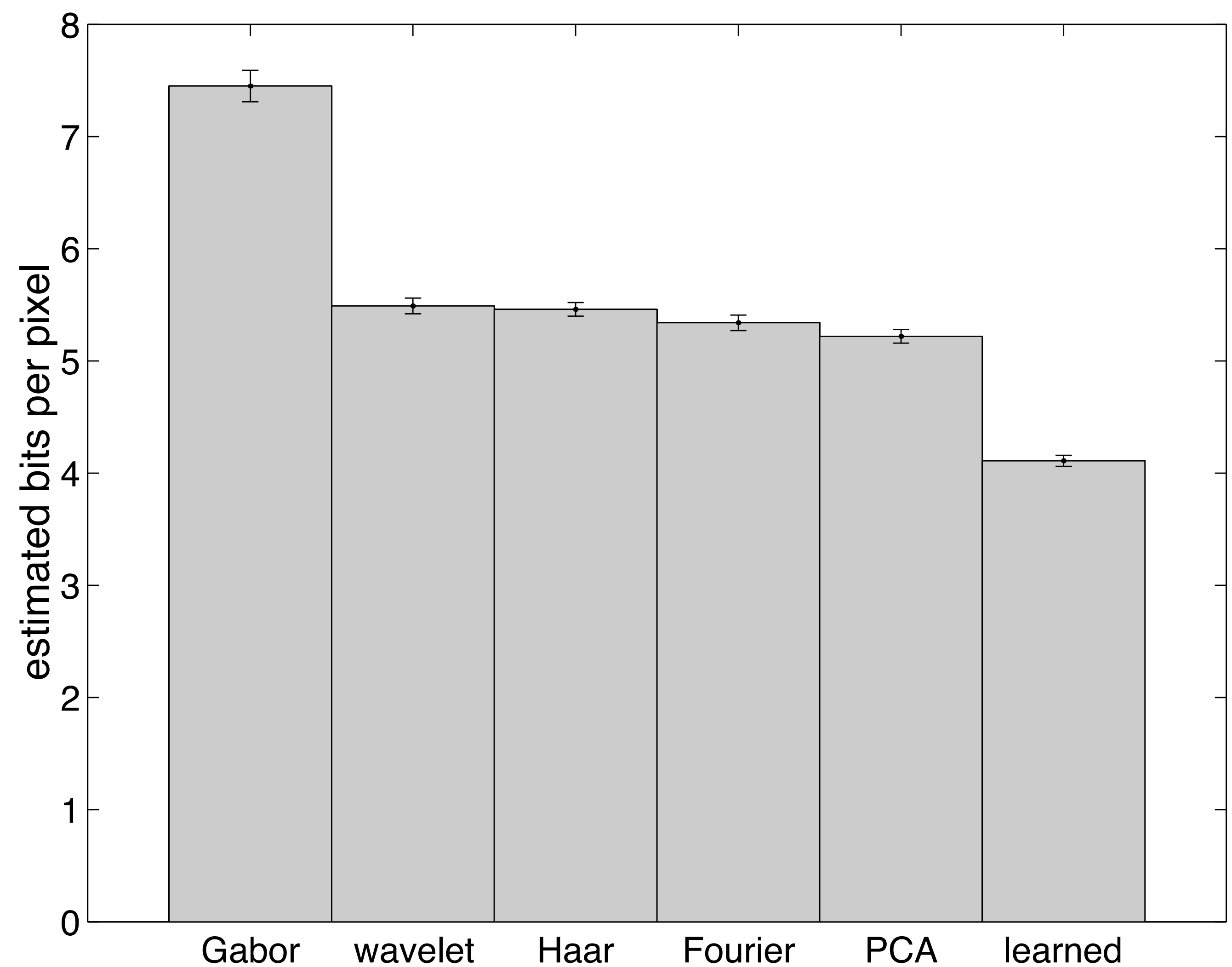
a model of the receptive field:
an oriented Gabor function

DeAngelis, et al, 1995

V1 receptive fields are not just edge "detectors":
an optimal code for all natural image structure

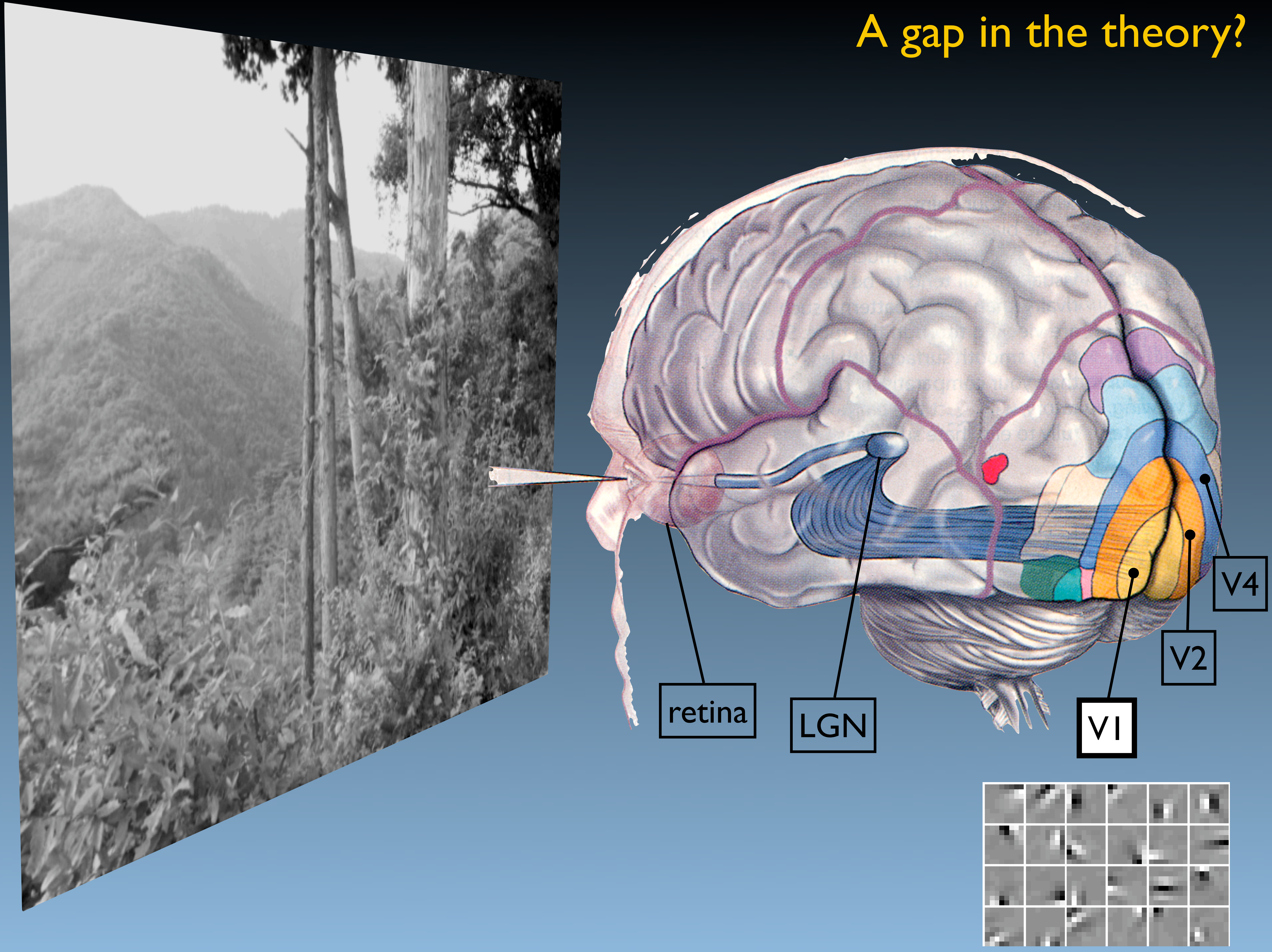
Efficient coding selects best of many possible sensory codes



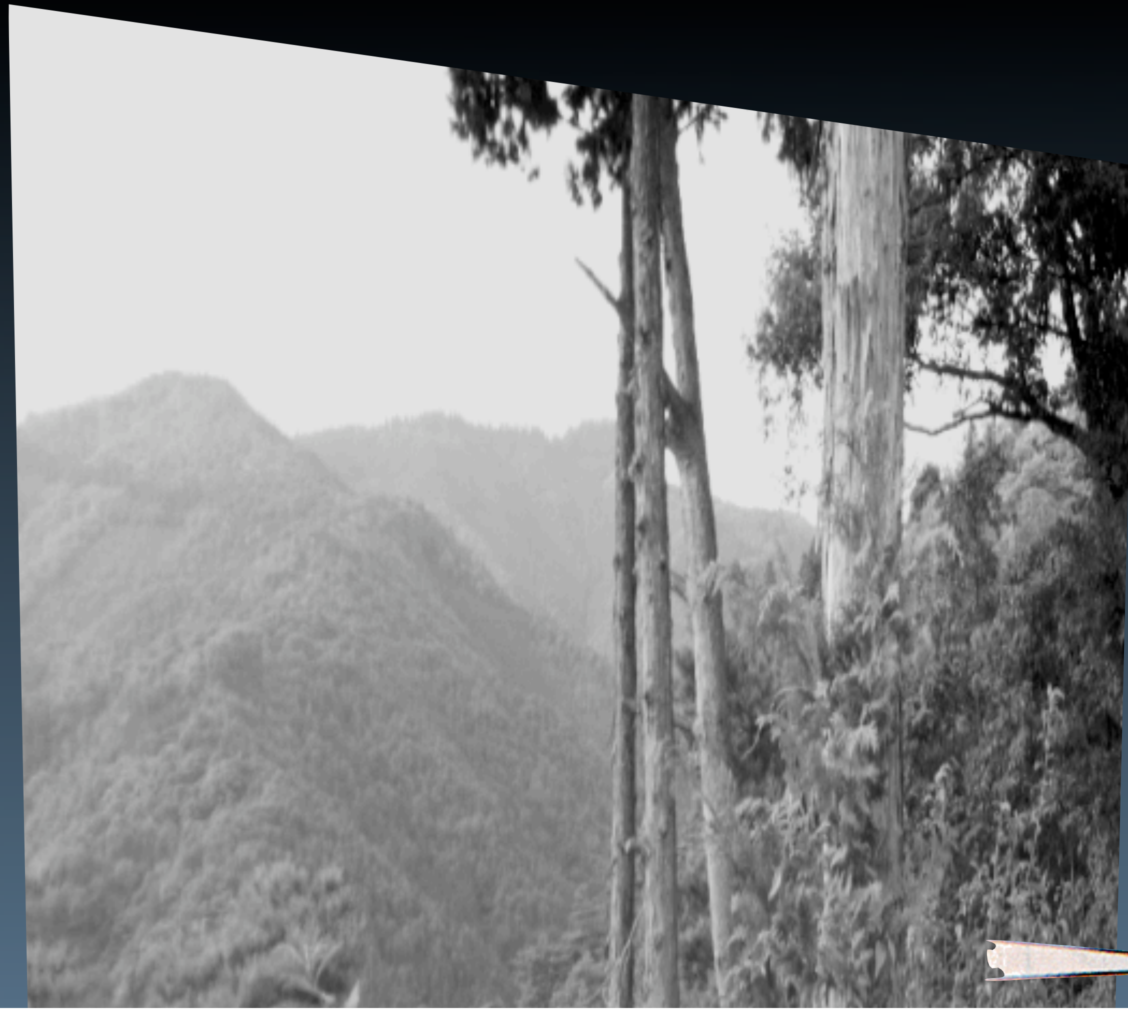


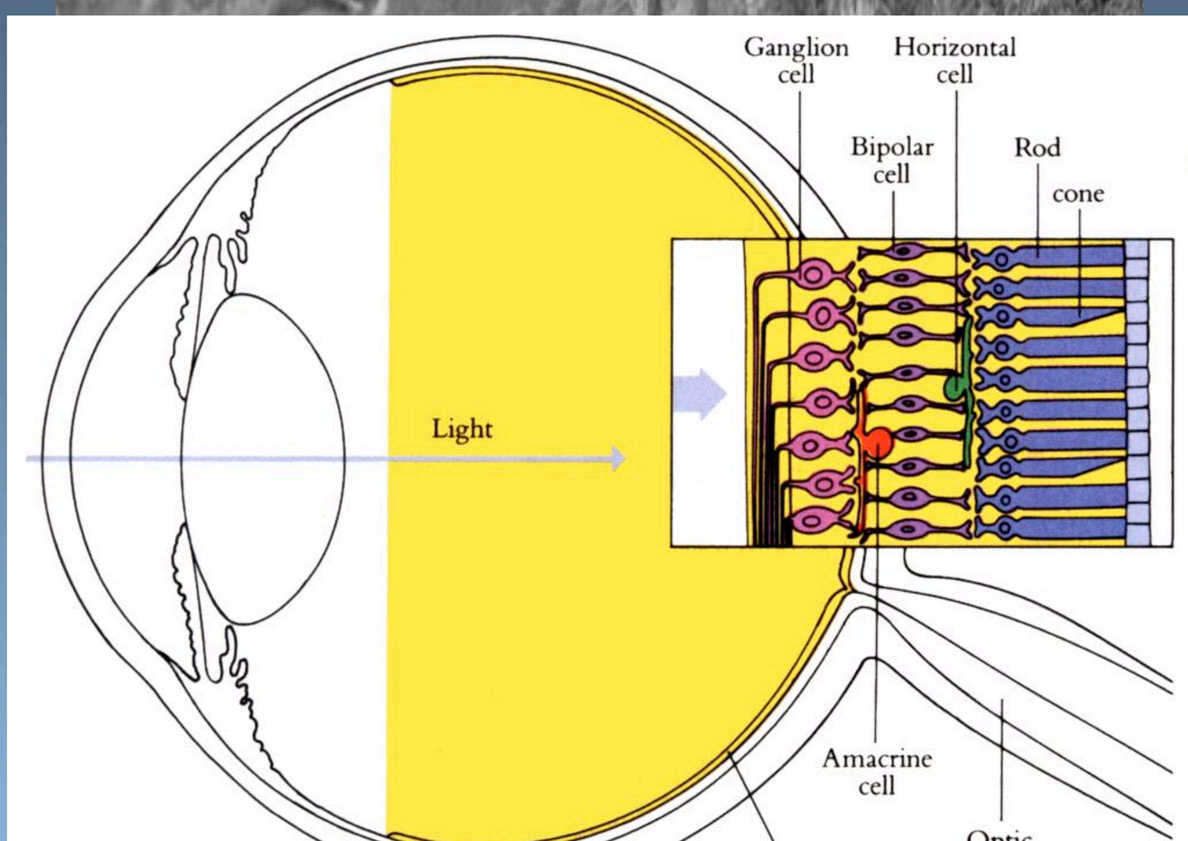
from Lewicki and Olshausen, 1999

A gap in the theory?

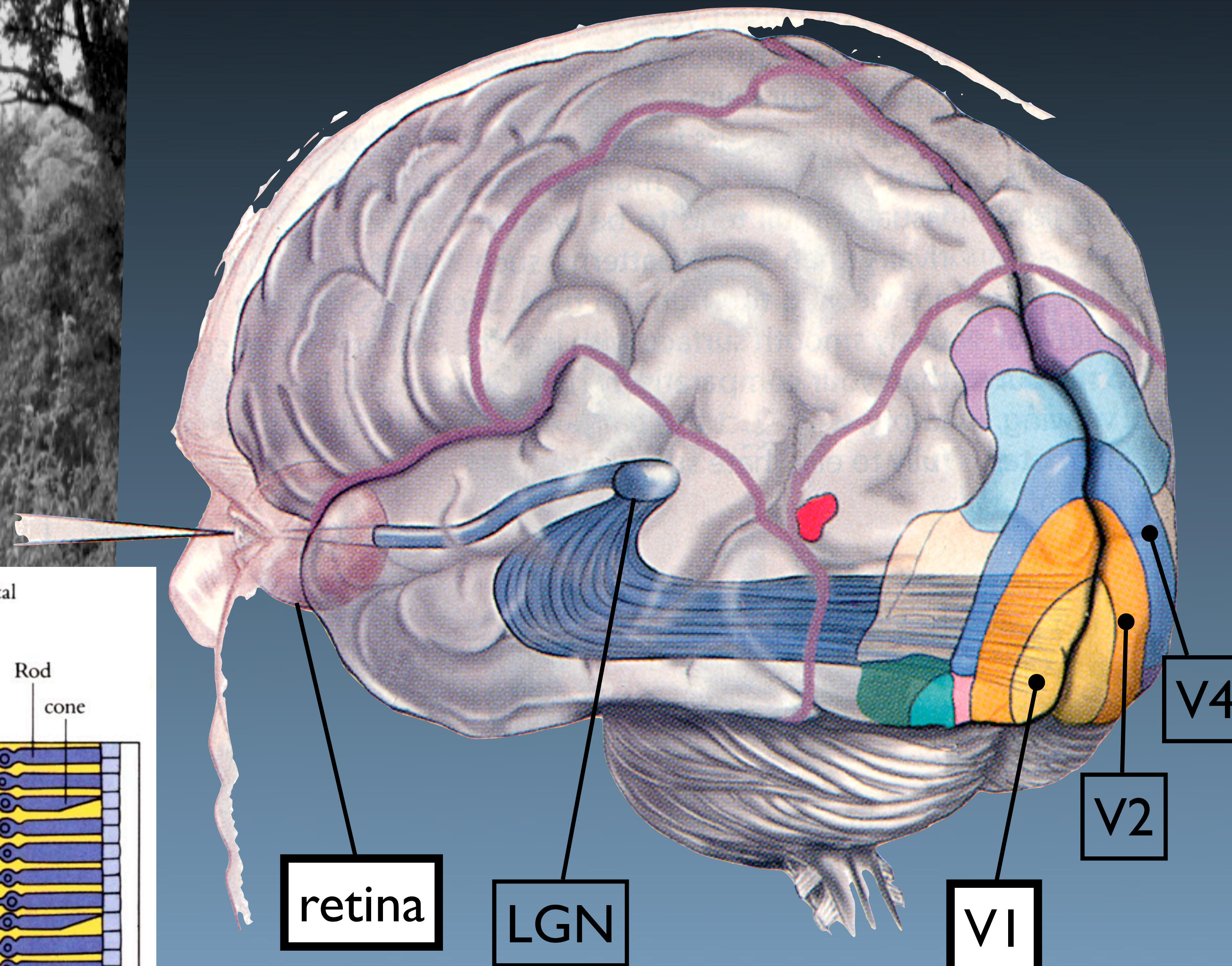


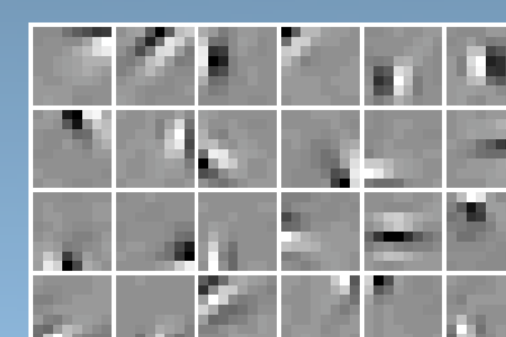
A gap in the theory?





from Hubel, 1995

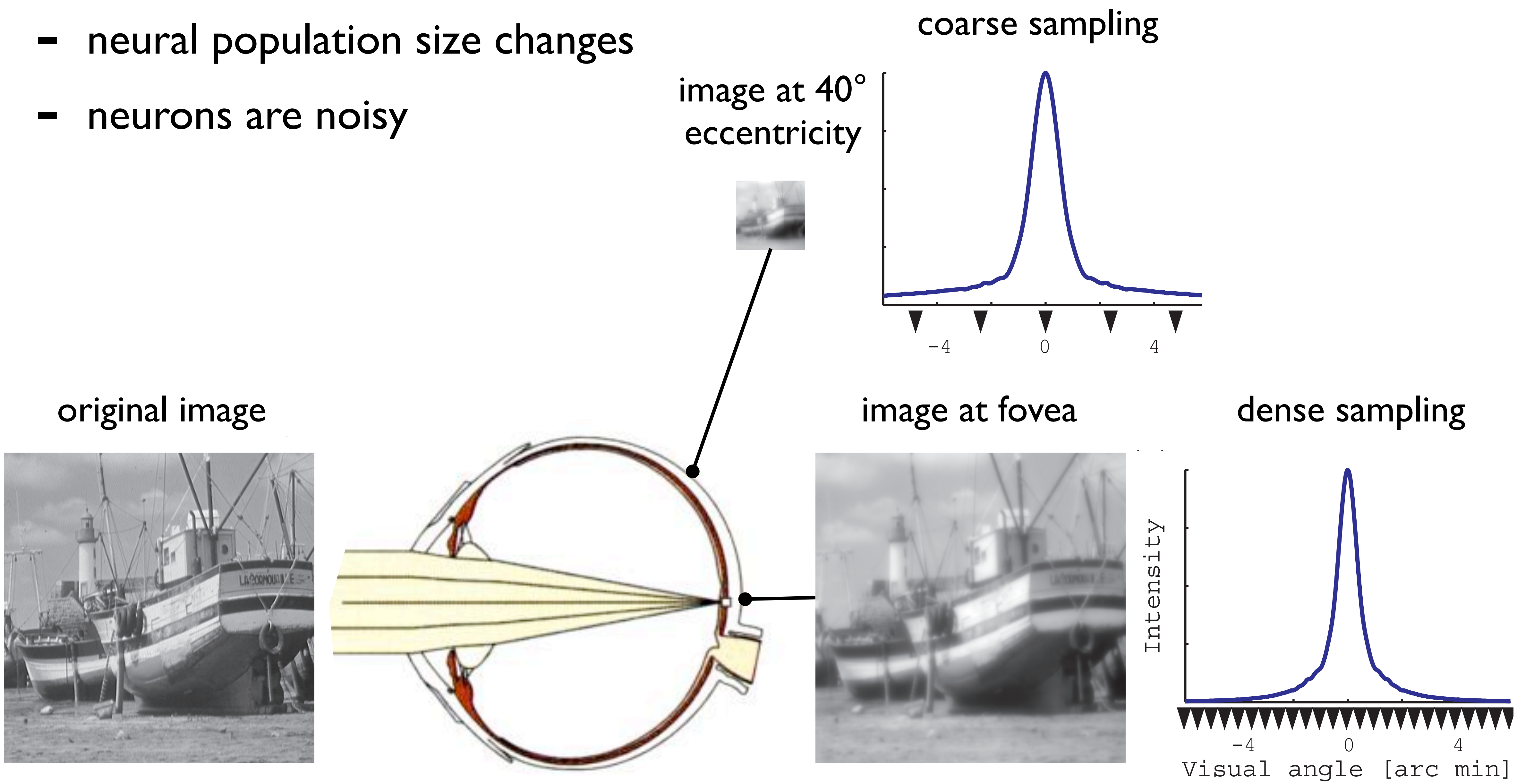




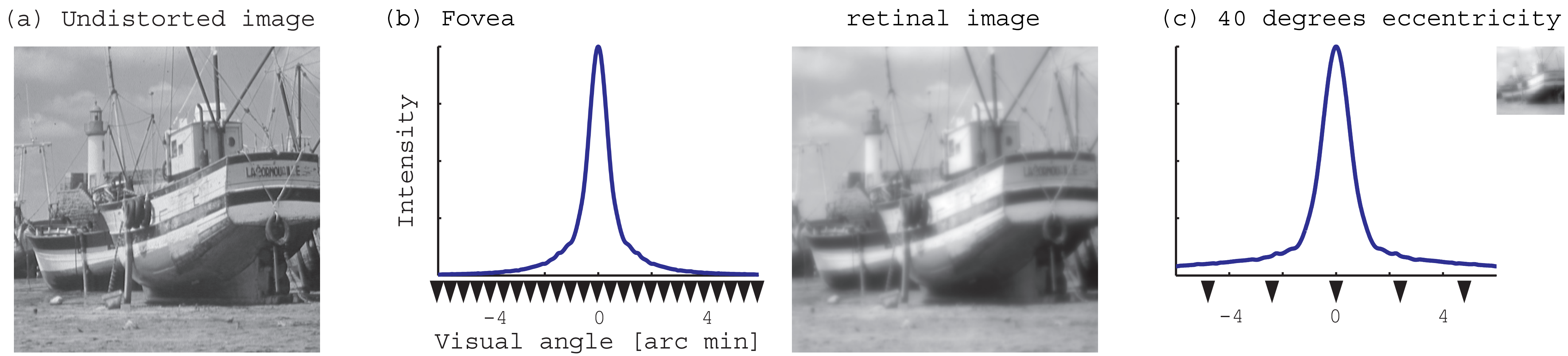
Robust coding of natural images

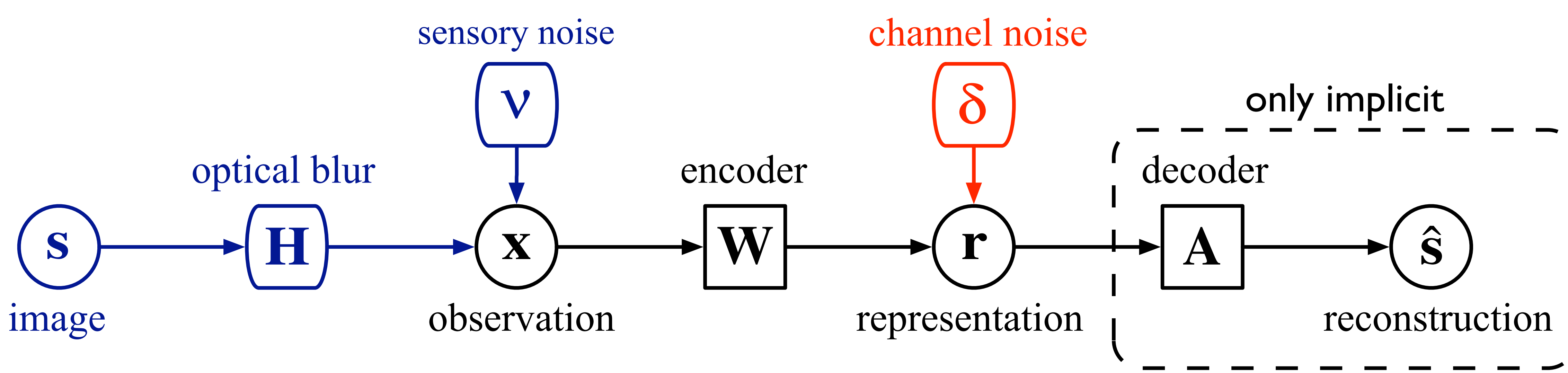
Doi and Lewicki (2005, 2006, 2007)

- Theory refined:
  - image is noisy and blurred
  - neural population size changes
  - neurons are noisy



Generalizing the model: sensory noise and optical blur



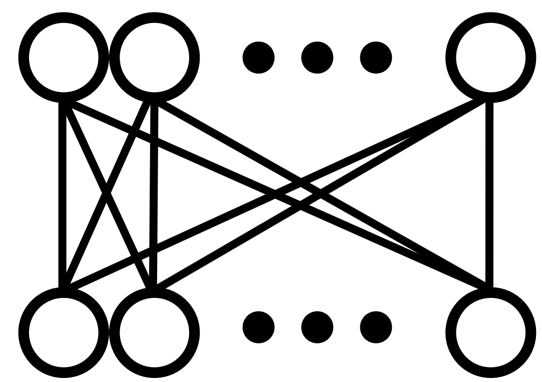


Can also add sparseness and resource constraints

Traditional codes are *not* robust

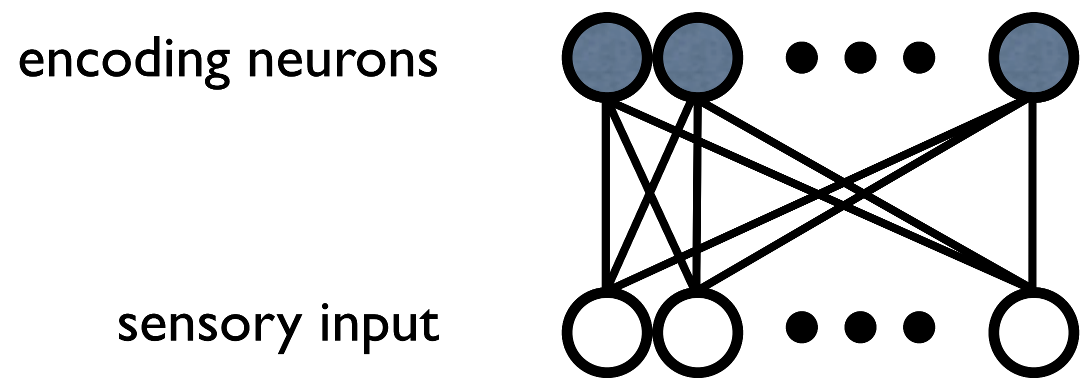
encoding neurons

sensory input



Original

Traditional codes are *not* robust



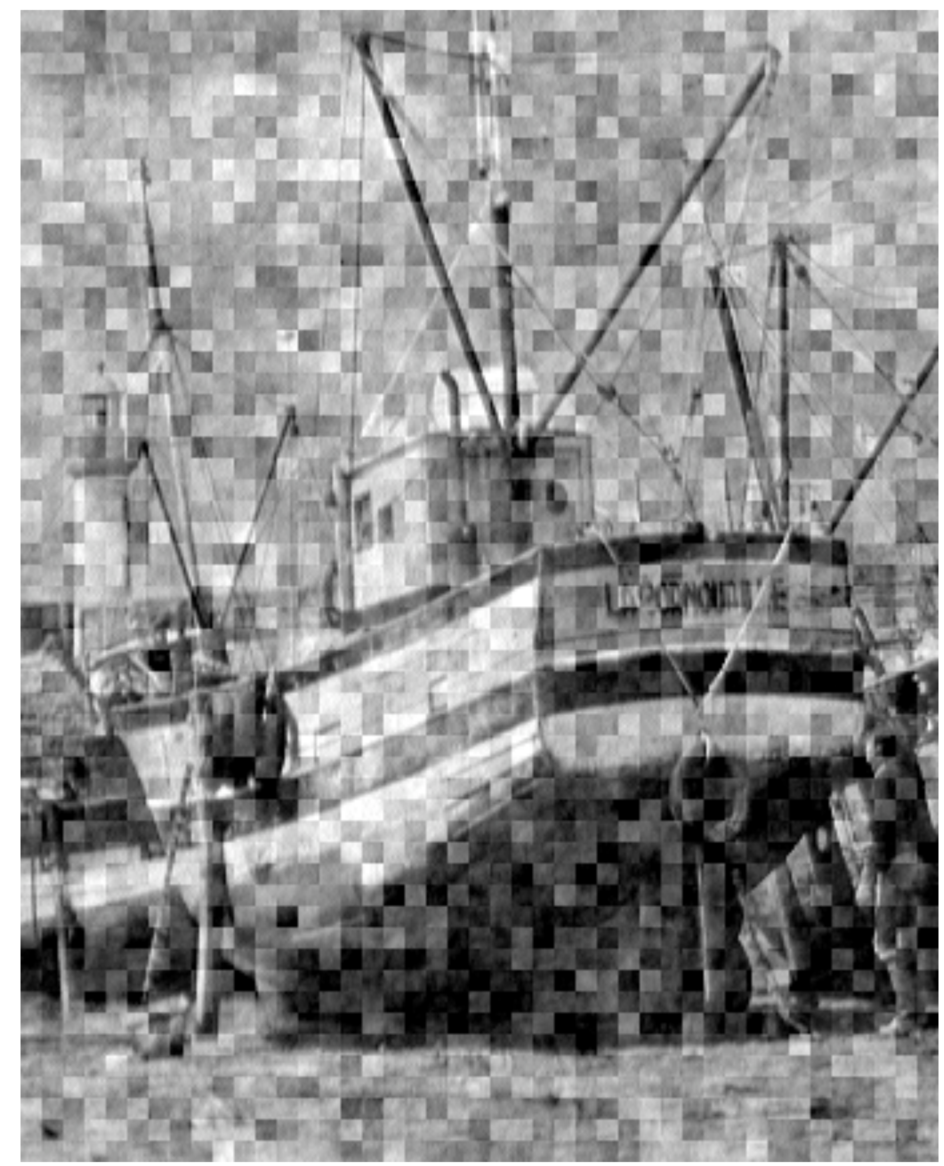
Add
noise equivalent
to 1 bit precision

Original

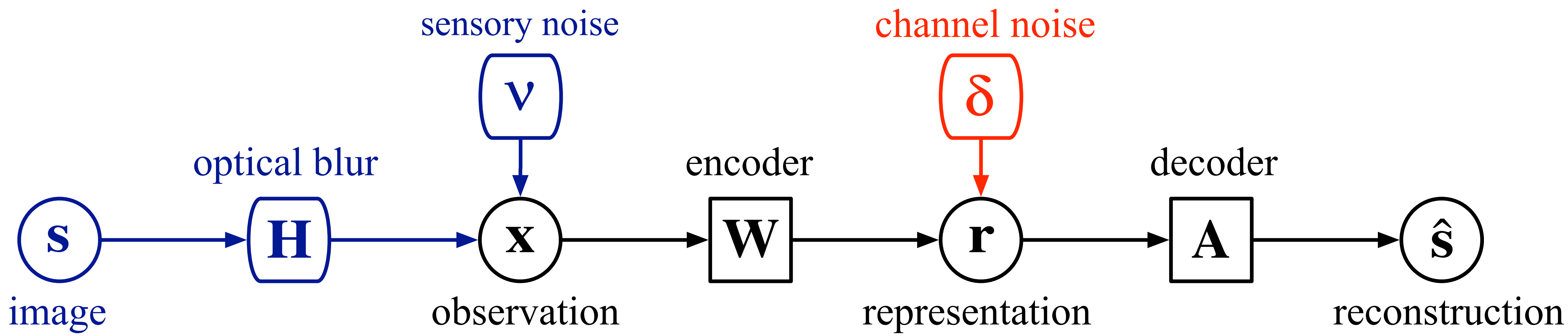
1x efficient coding

1 bit precision

reconstruction (34% error)



How do we learn robust codes?



Objective:

Find **W** and **A** that minimize reconstruction error.

- Channel capacity of the i^{th} neuron:

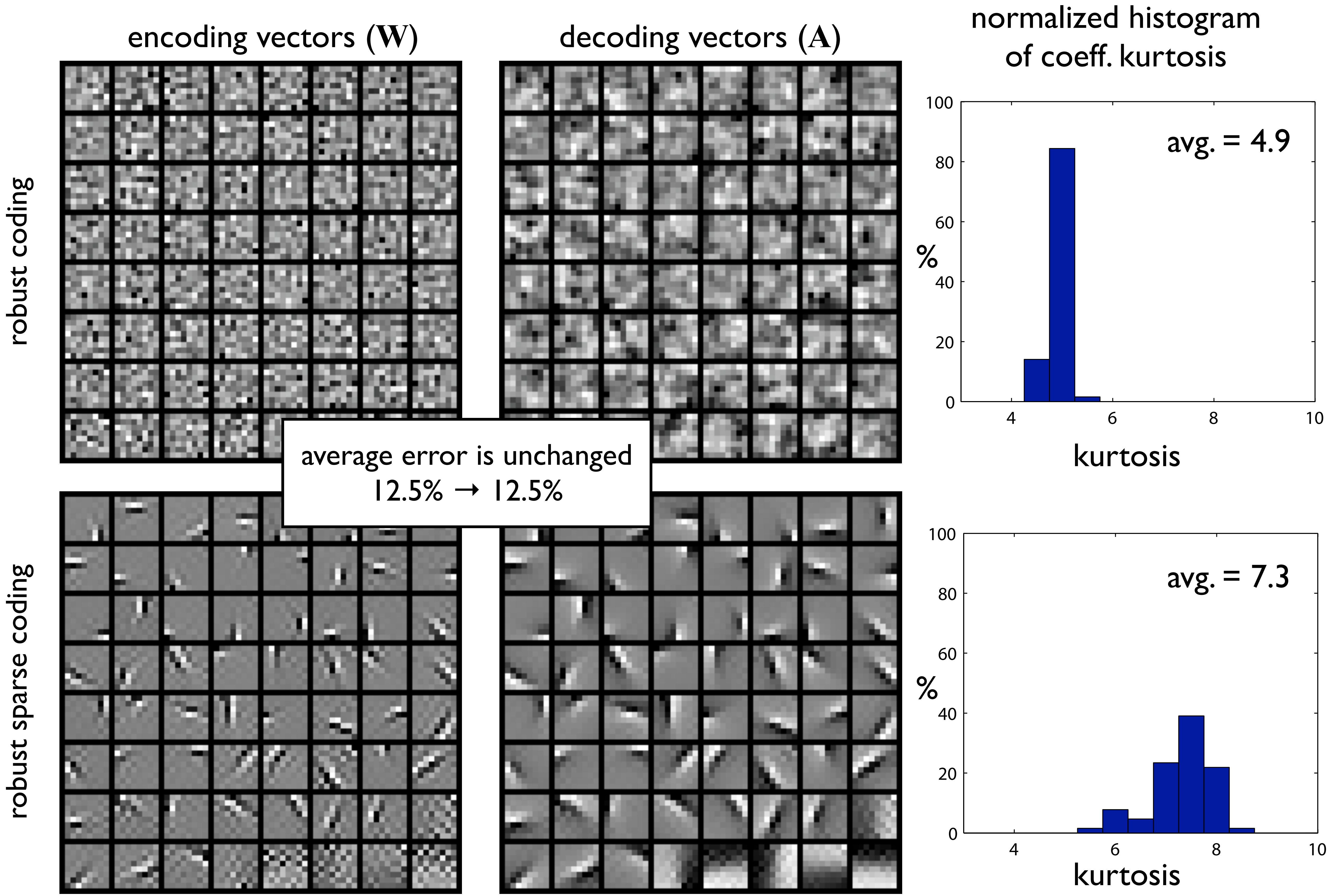
$$C_i = \frac{1}{2}\ln(\mathrm{SNR}_i + 1)$$

- To limit capacity, fix the coefficient signal to noise ratio:

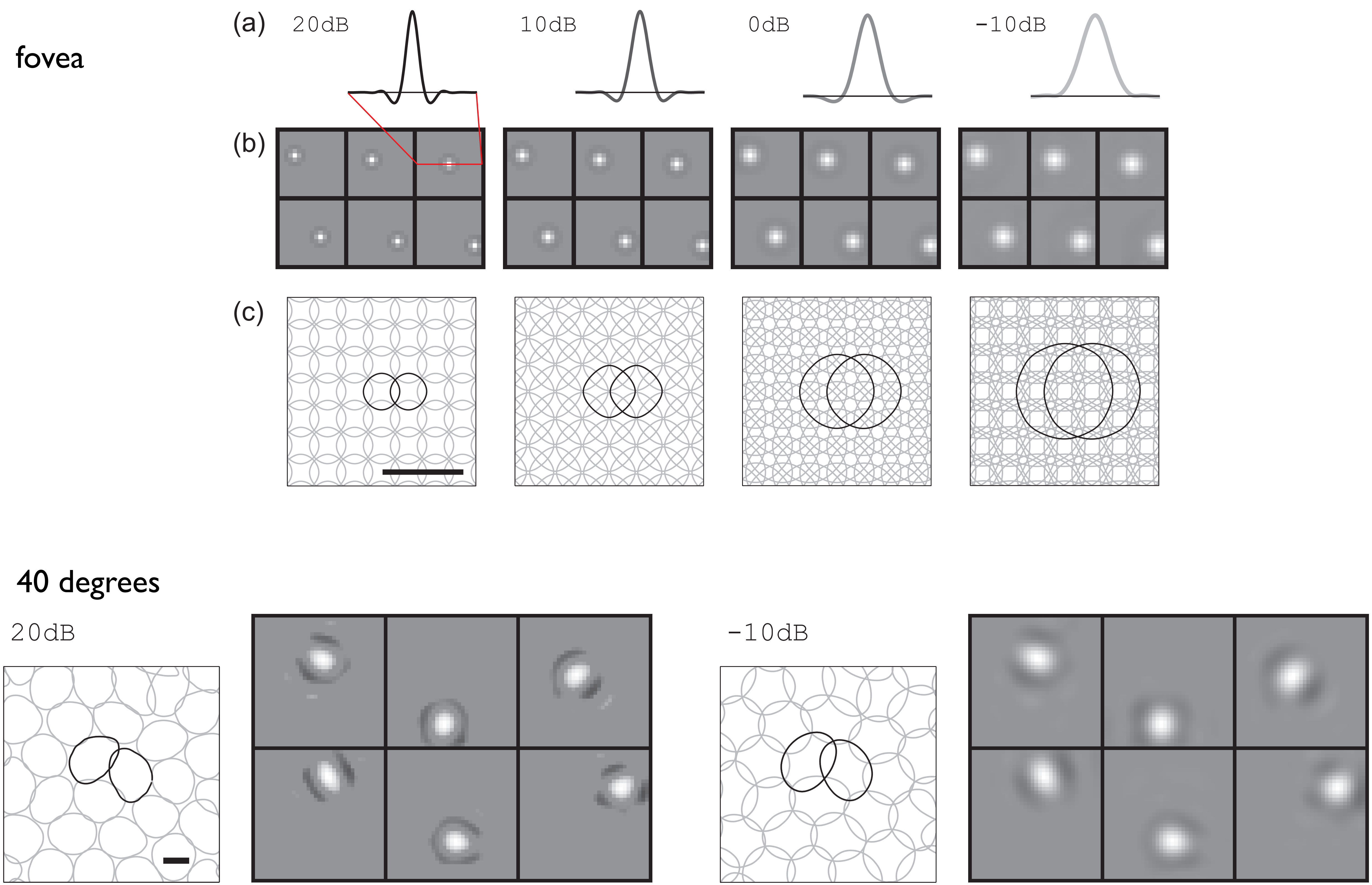
$$\mathrm{SNR}_i = \frac{\langle u_i^2 \rangle}{\sigma_n^2}$$

Now robust coding is formulated as a *constrained optimization problem.*

Sparseness localizes the vectors and increases coding efficiency



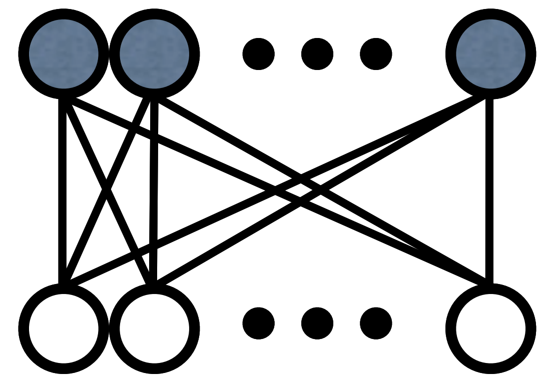
Optimal weights match retinal code and response properties



Robust coding of natural images

encoding neurons

sensory input



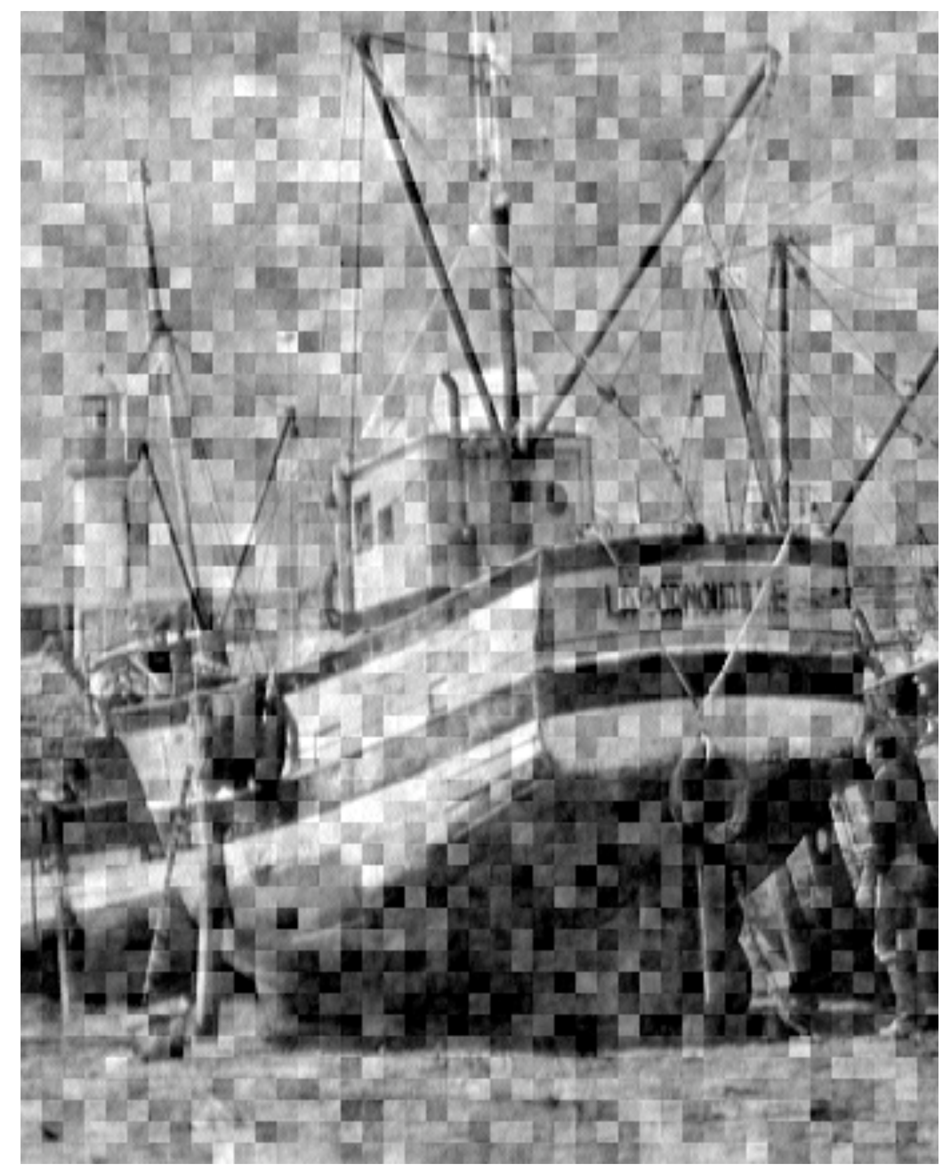
Add
noise equivalent
to 1 bit precision

Original

1x efficient coding

1 bit precision

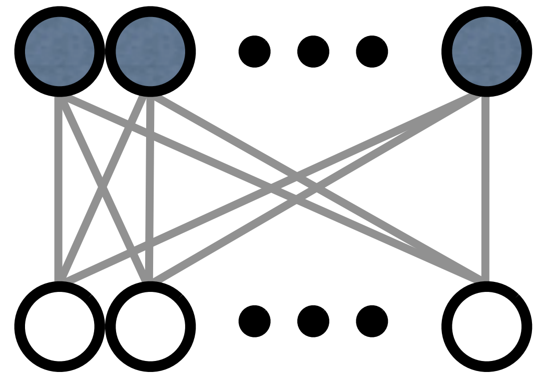
reconstruction (34% error)



Robust coding of natural images

encoding neurons

sensory input



Weights adapted for optimal robustness

Original

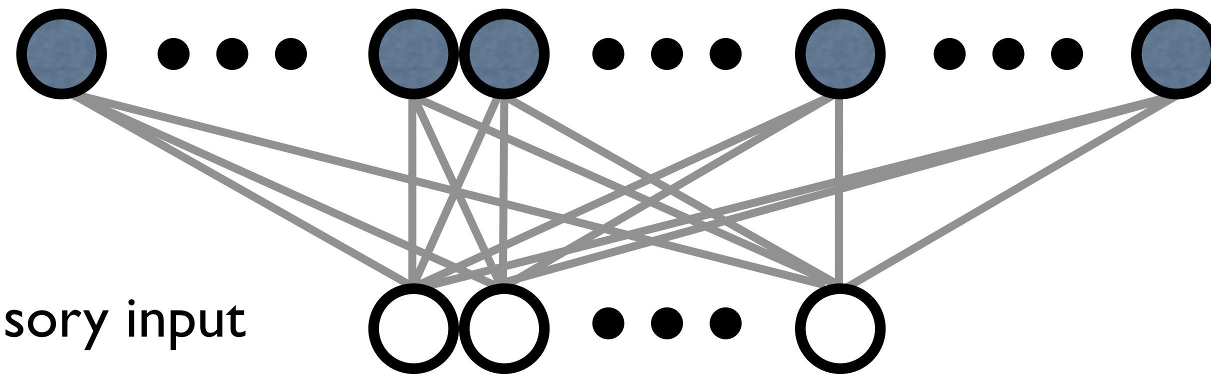
1x robust coding

1 bit precision

reconstruction (3.8% error)

Reconstruction improves by adding neurons

encoding neurons



Weights adapted for optimal robustness

sensory input

Original

8x robust coding

1 bit precision

reconstruction error: 0.6%

Can derive minimum theoretical average error bound

$$\mathcal{E} = \frac{1}{\frac{M}{N} \cdot \mathrm{SNR} + 1} \; \frac{1}{N} \left[\sum_{i=1}^{N} \sqrt{\lambda_i} \right]^2 \qquad \text{if SNR} \geq \text{SNR}_c$$

λ_i - ith eigenvalue of the data covariance

N - input dimensionality

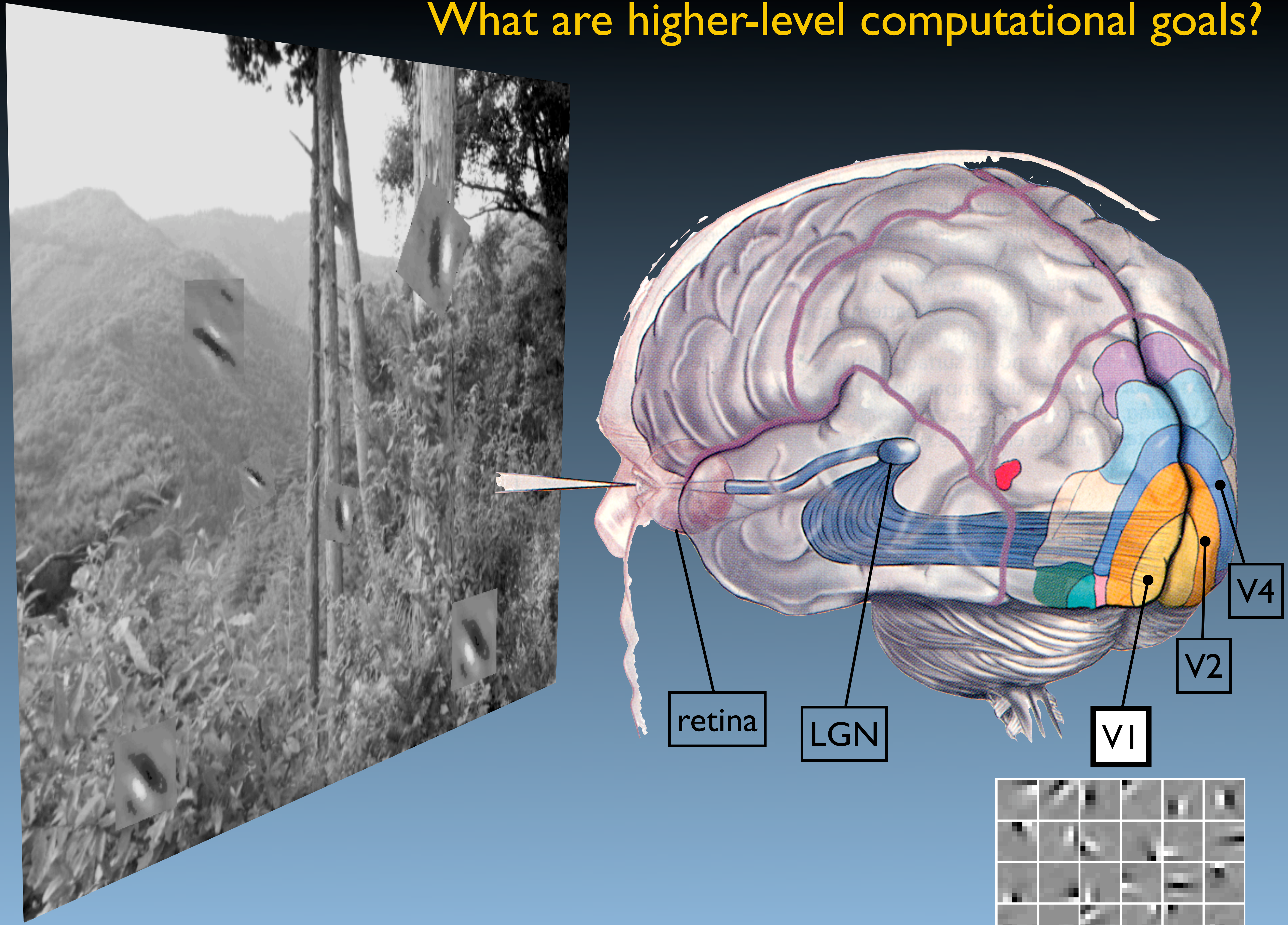
M - # of coding units (neurons)

Algorithm achieves theoretical lower bound

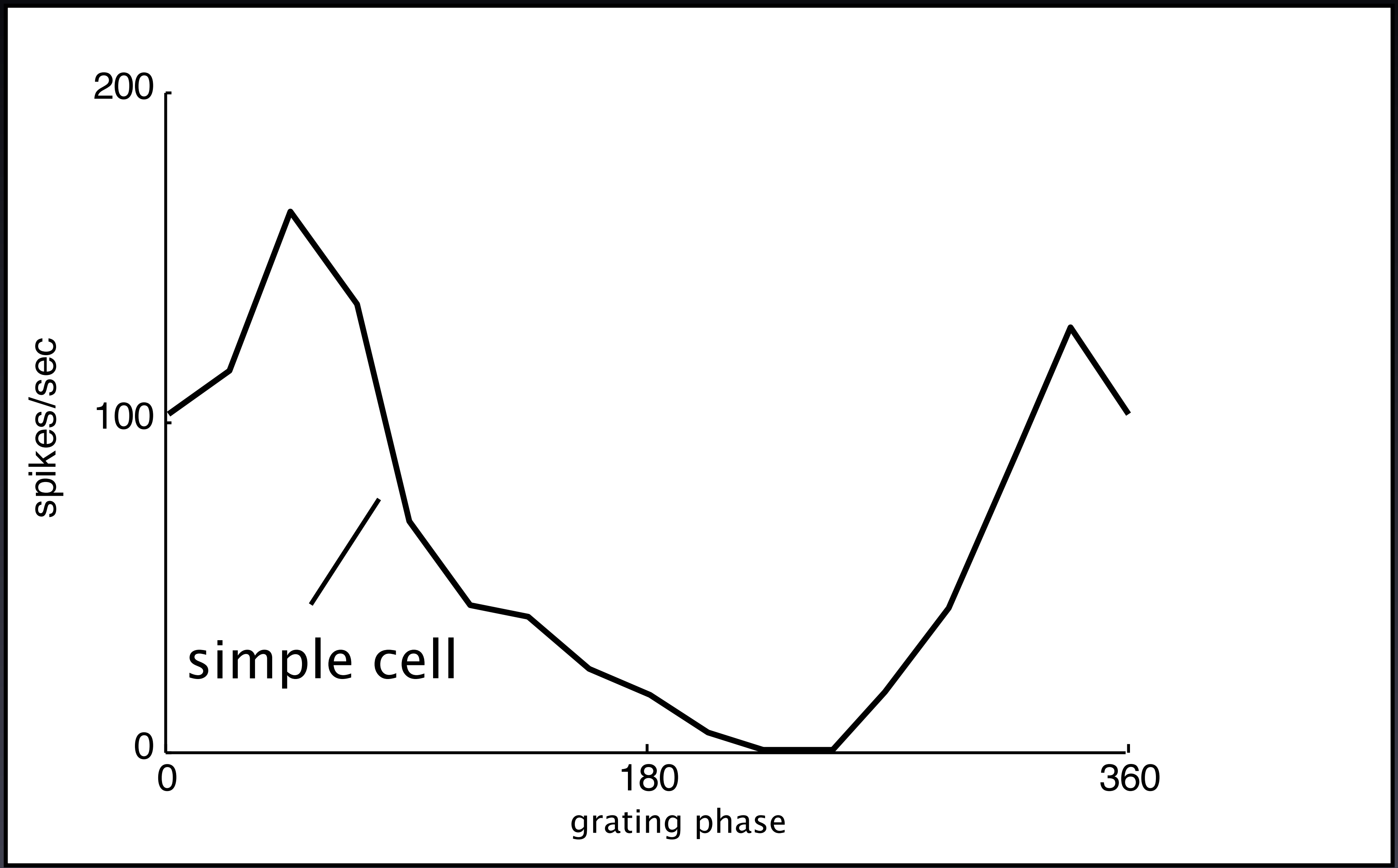
	Results	Bound
0.5x	19.9%	20.3%
1x	12.4%	12.5%
8x	2.0%	2.0%

Balcan, Doi, and Lewicki, 2007;
Balcan and Lewicki, 2007

What are higher-level computational goals?

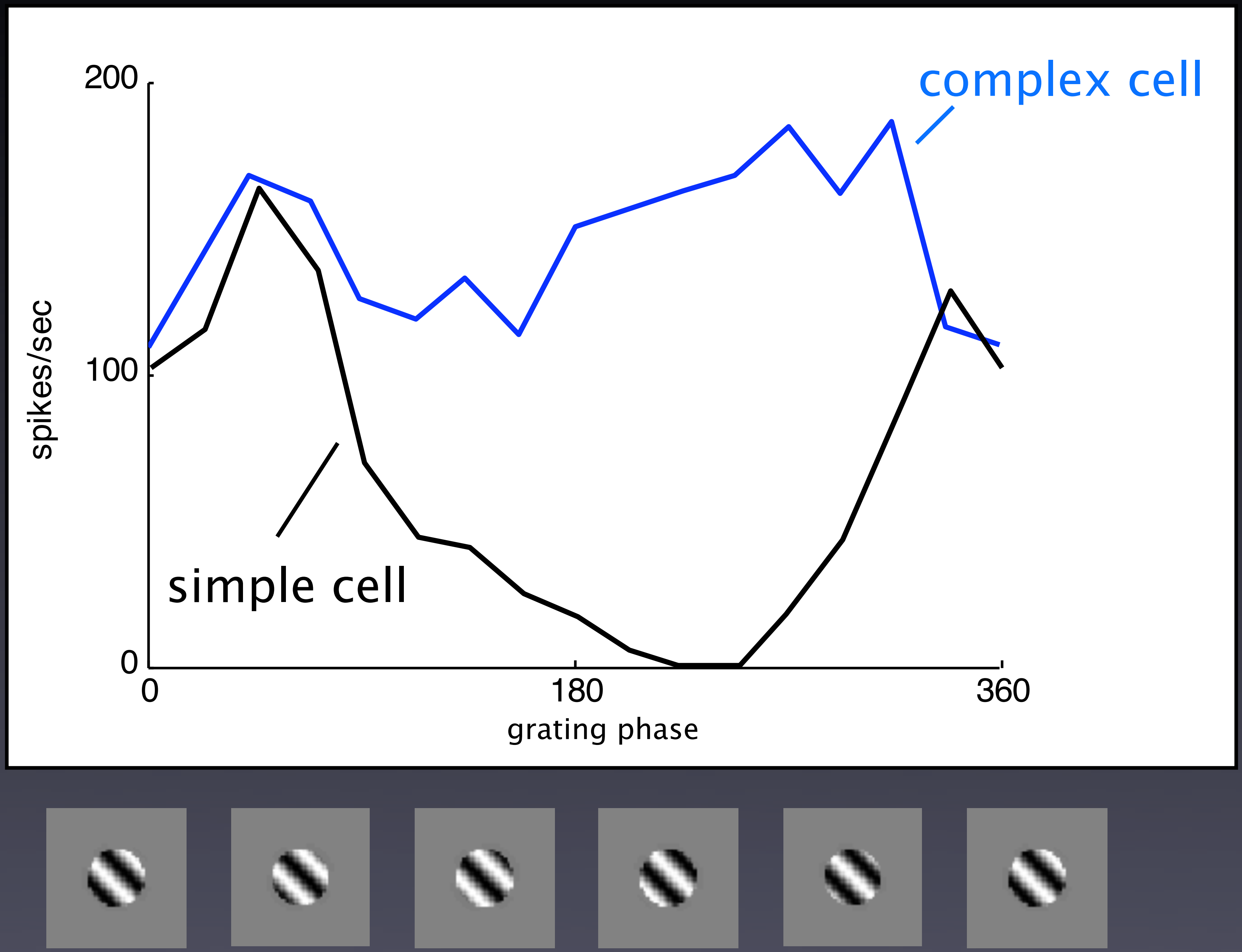


Response of a simple cell to translating grating



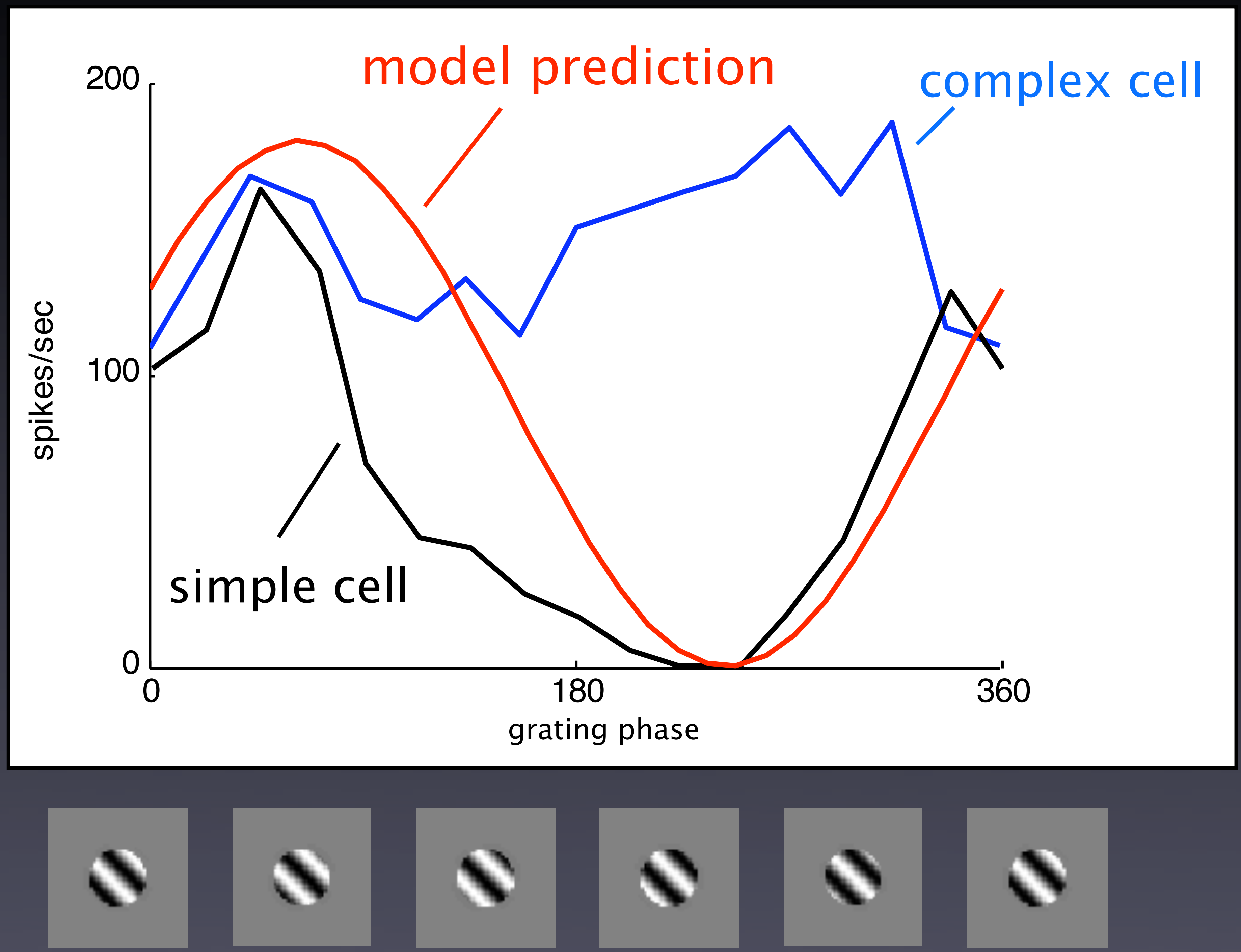
(Movshon et al, 1978)

Response of a simple cell to translating grating



(Movshon et al, 1978)

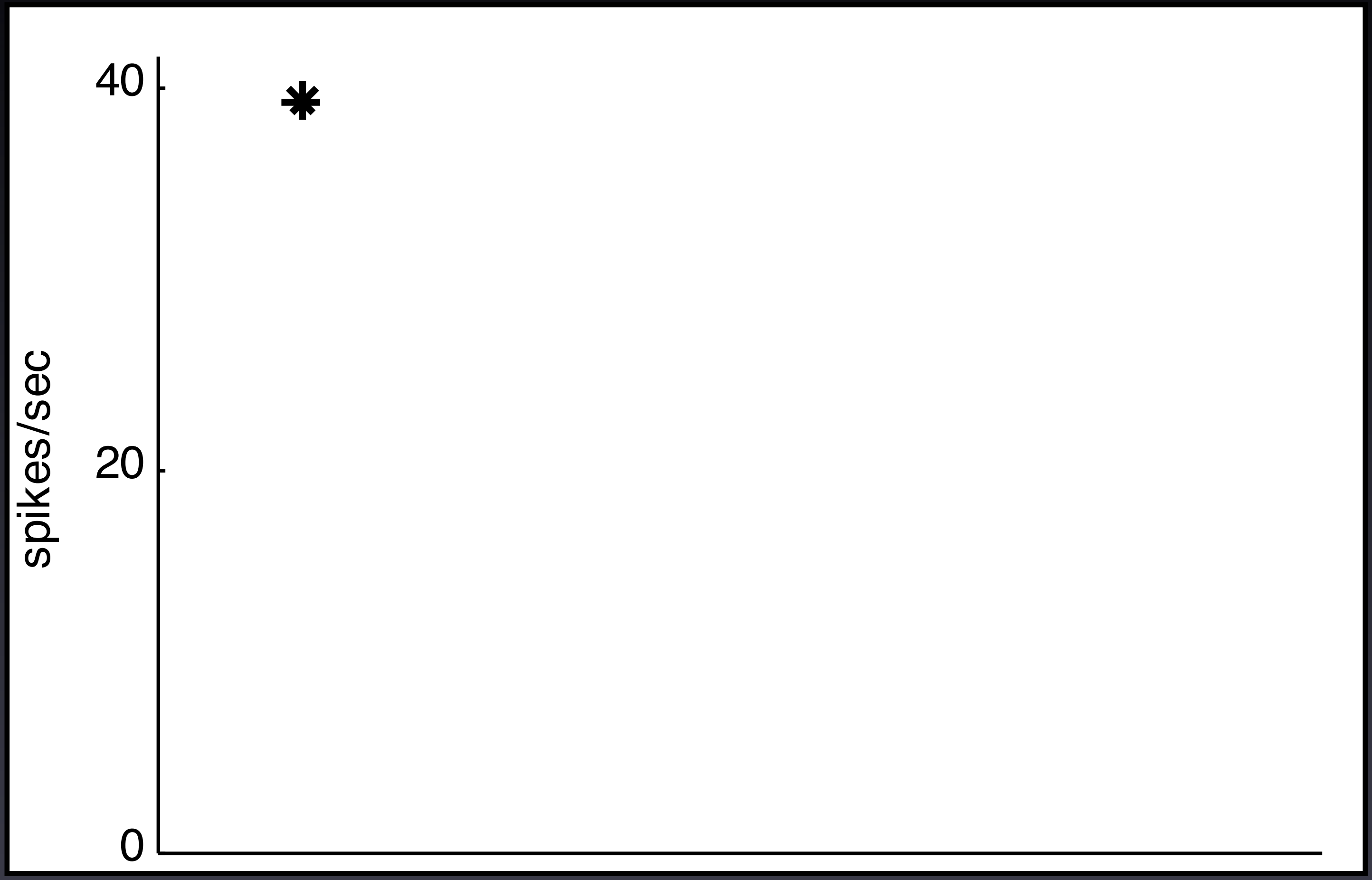
Response of a simple cell to translating grating



(Movshon et al, 1978)

V1 cells have many other unexplained properties

surround suppression in V1

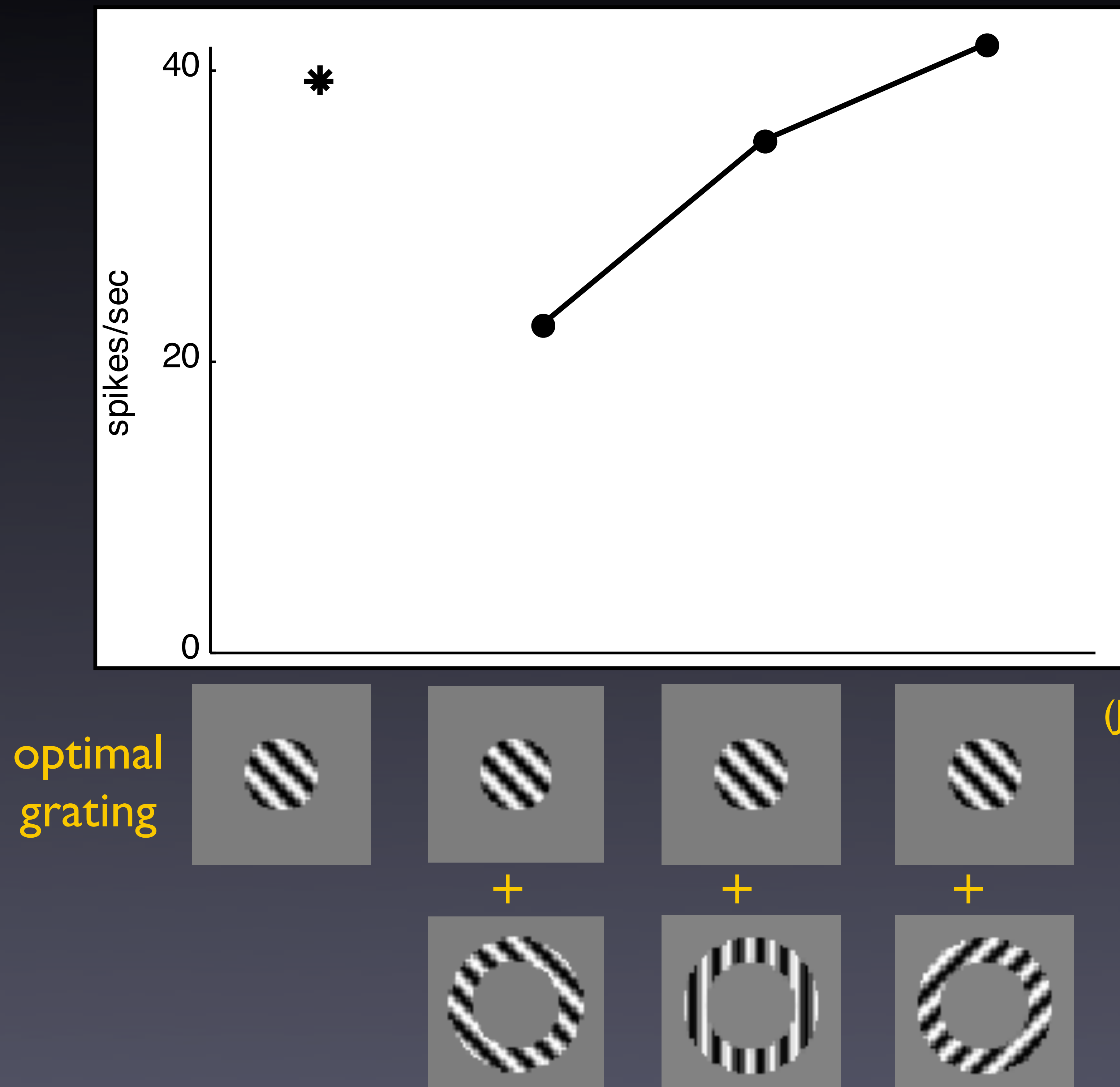


optimal grating

(Jones et al, 2002)

V1 cells have many other unexplained properties

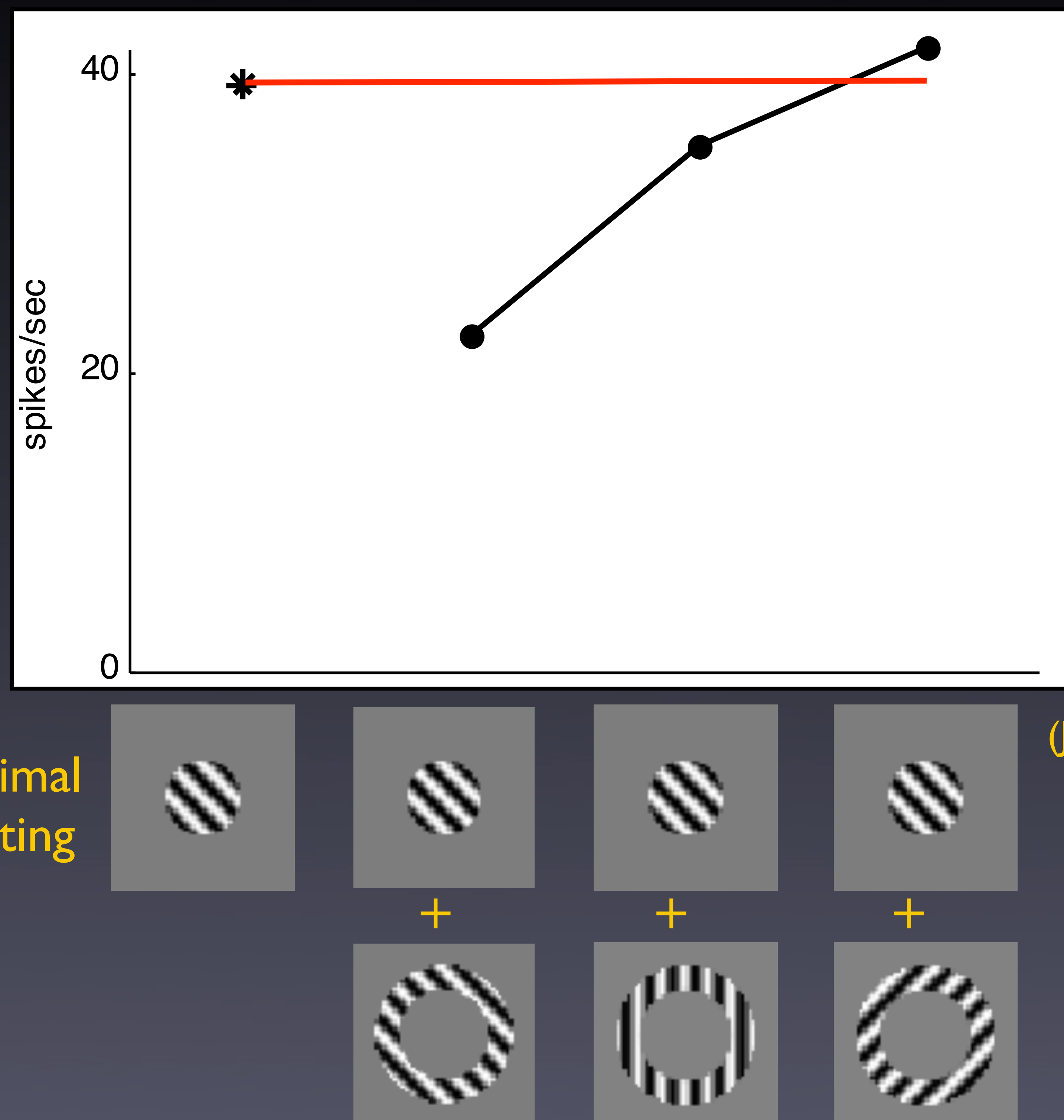
surround suppression in V1



(Jones et al, 2002)

V1 cells have many other unexplained properties

surround suppression in V1



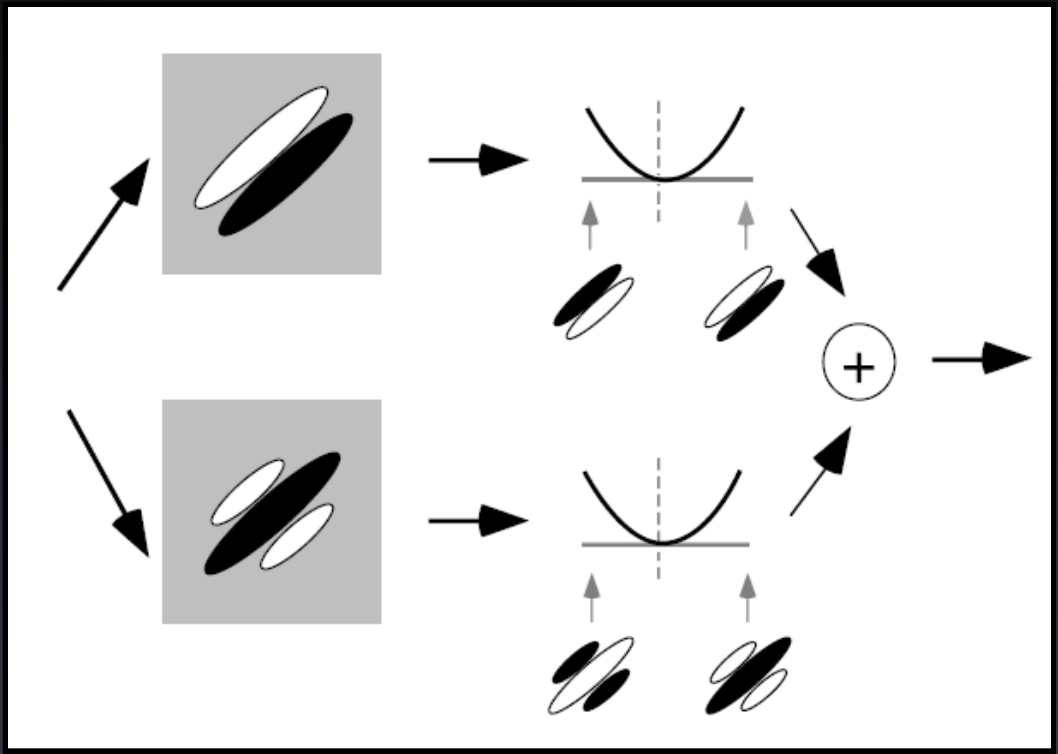
model prediction

(Jones et al, 2002)

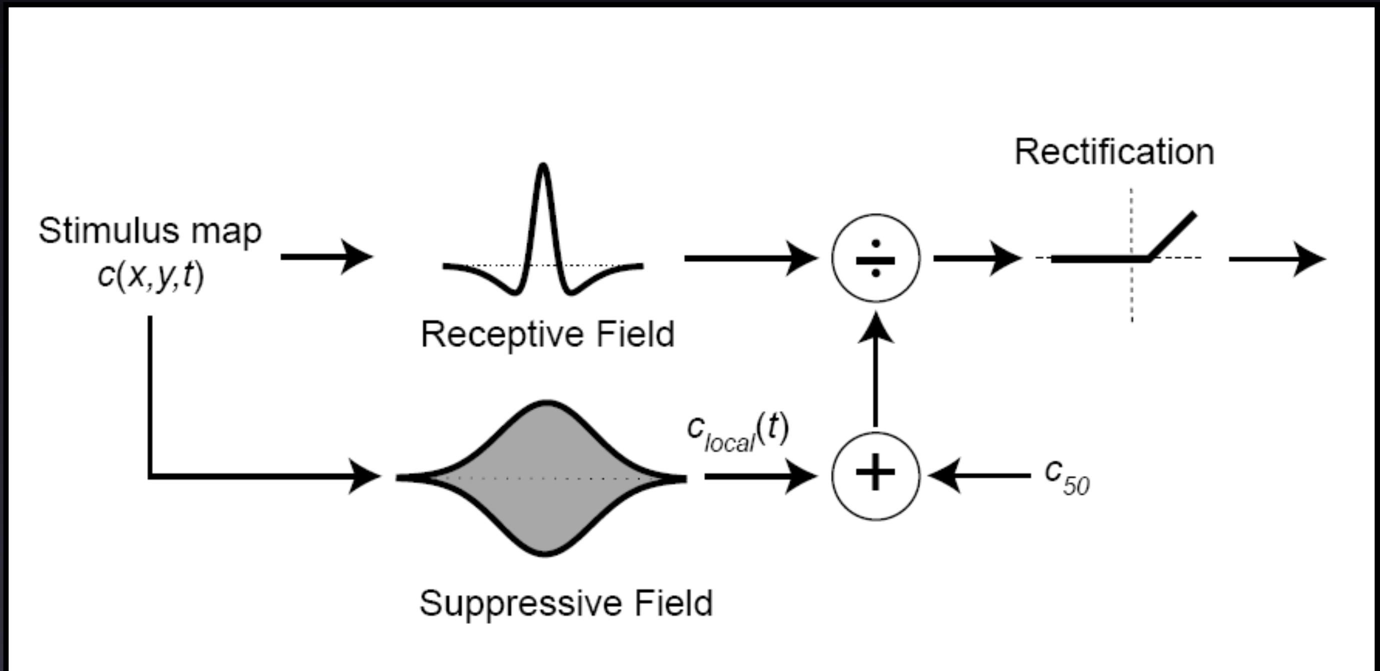
optimal grating

Models of V1 non-linear responses

models for complex cells



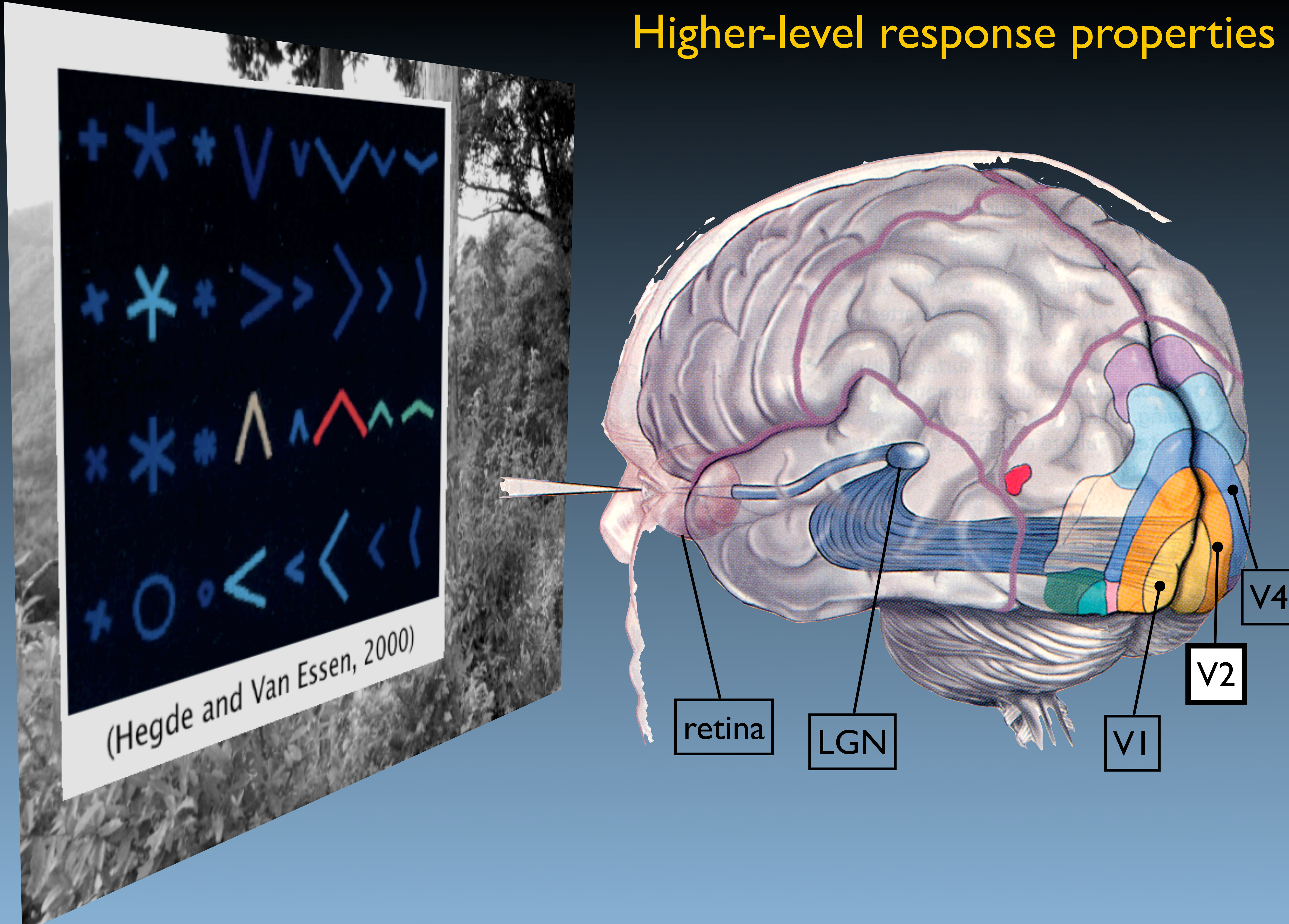
models for surround effects



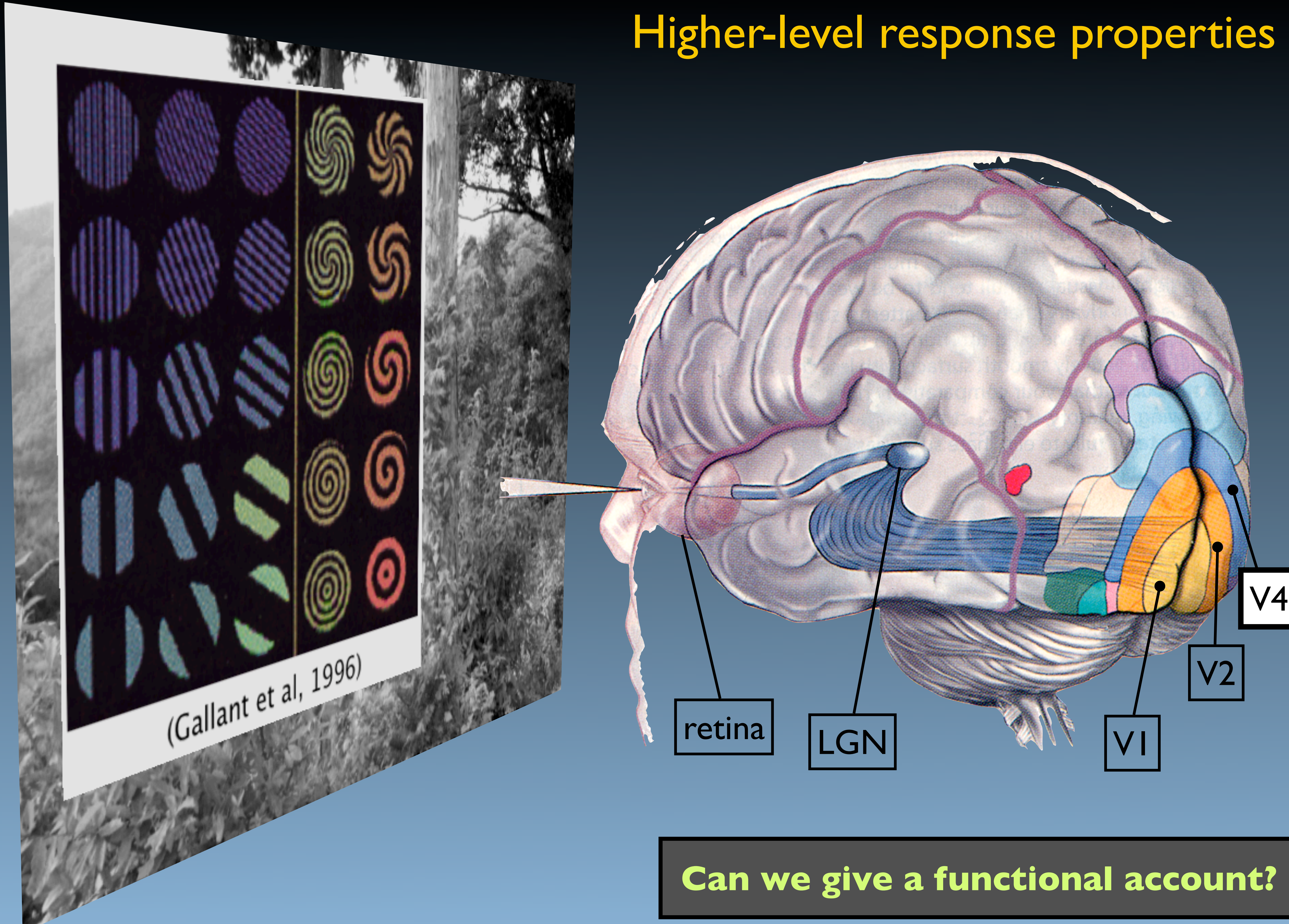
(Carandini et al, 2004,2005)

What is the functional significance?

Higher-level response properties



Higher-level response properties

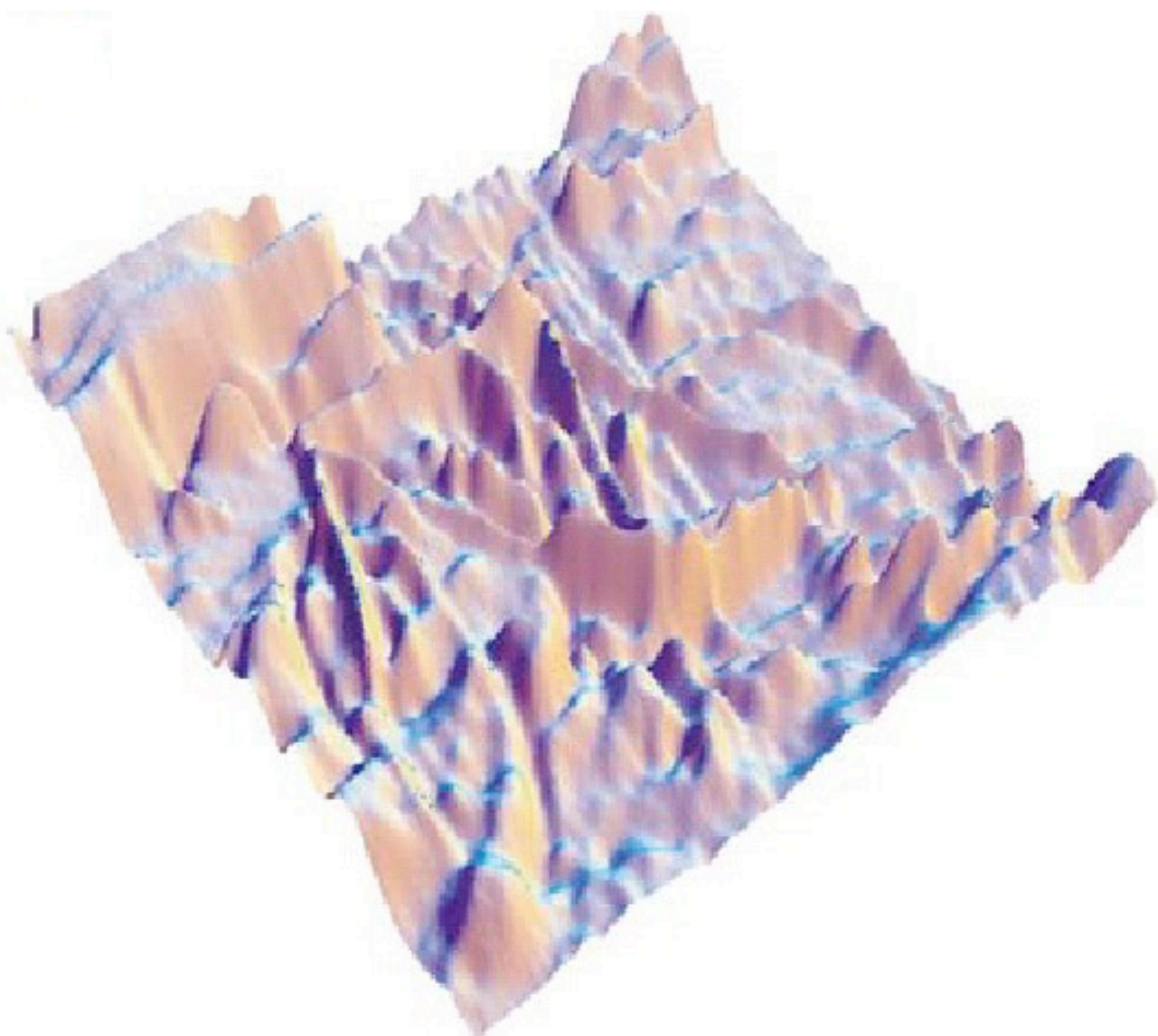


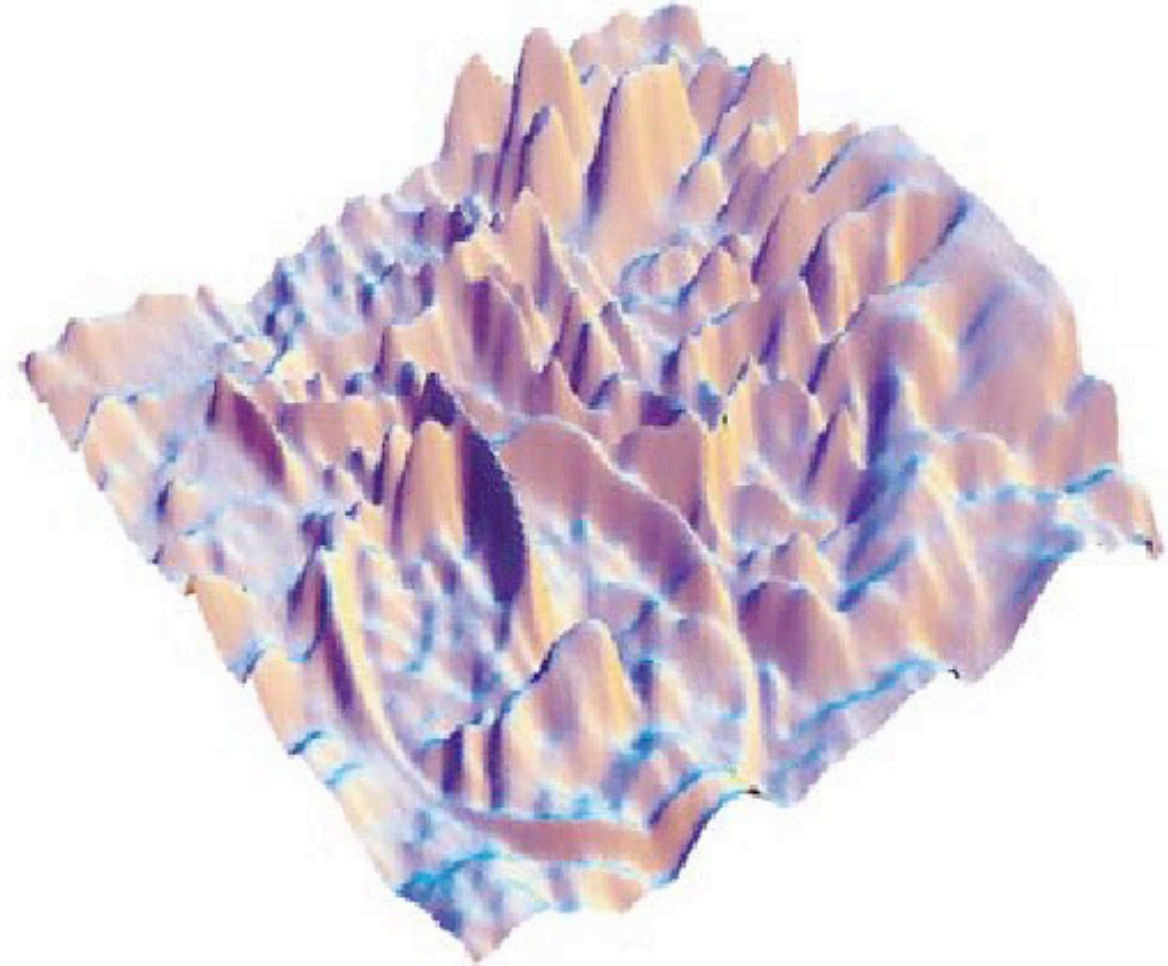
Can we give a functional account?

Perceptual organization in natural scenes

image of Kyoto, Japan from E. Doi

A different representation of a natural scene (Kersten and Yuille, 2003)

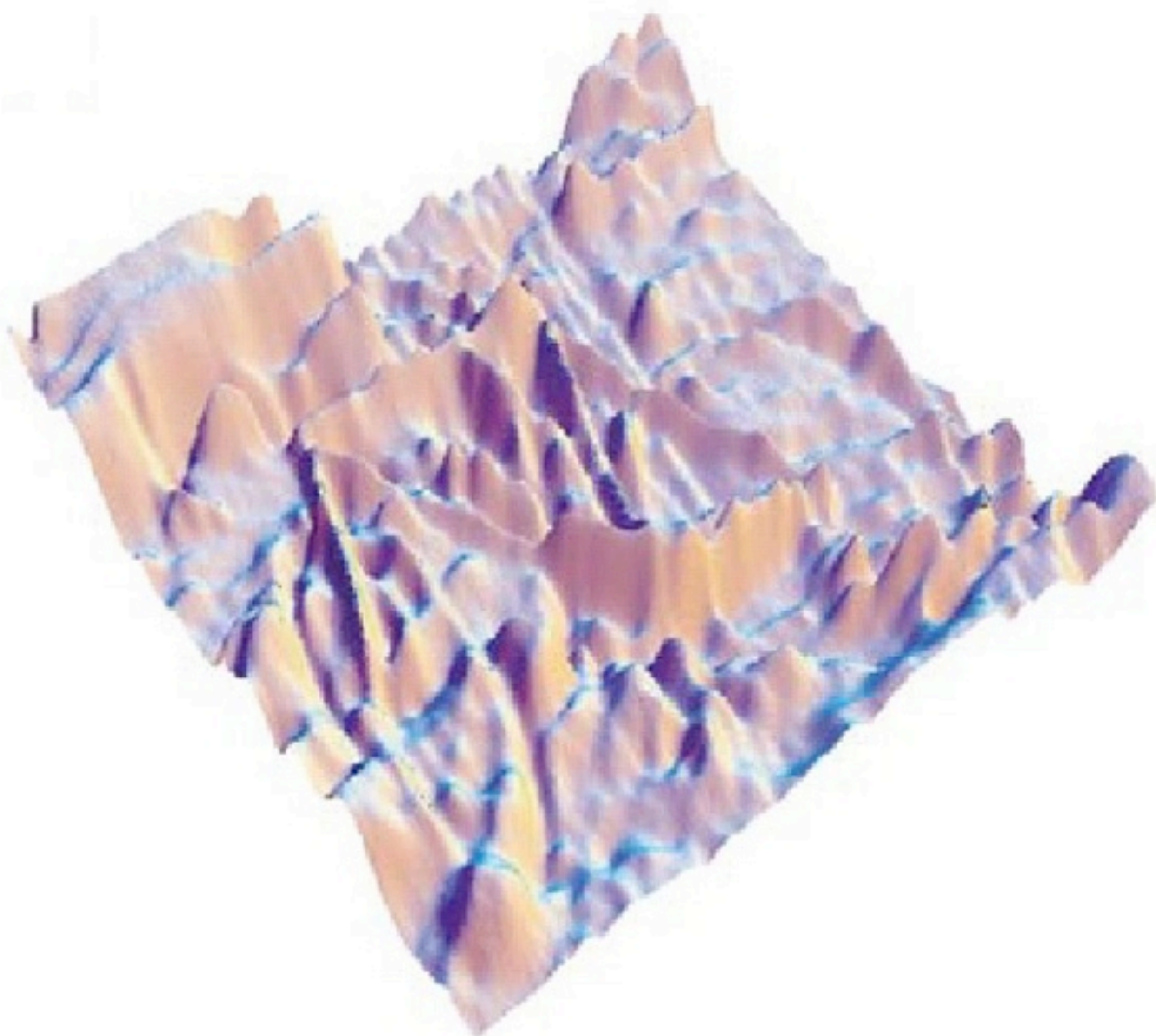


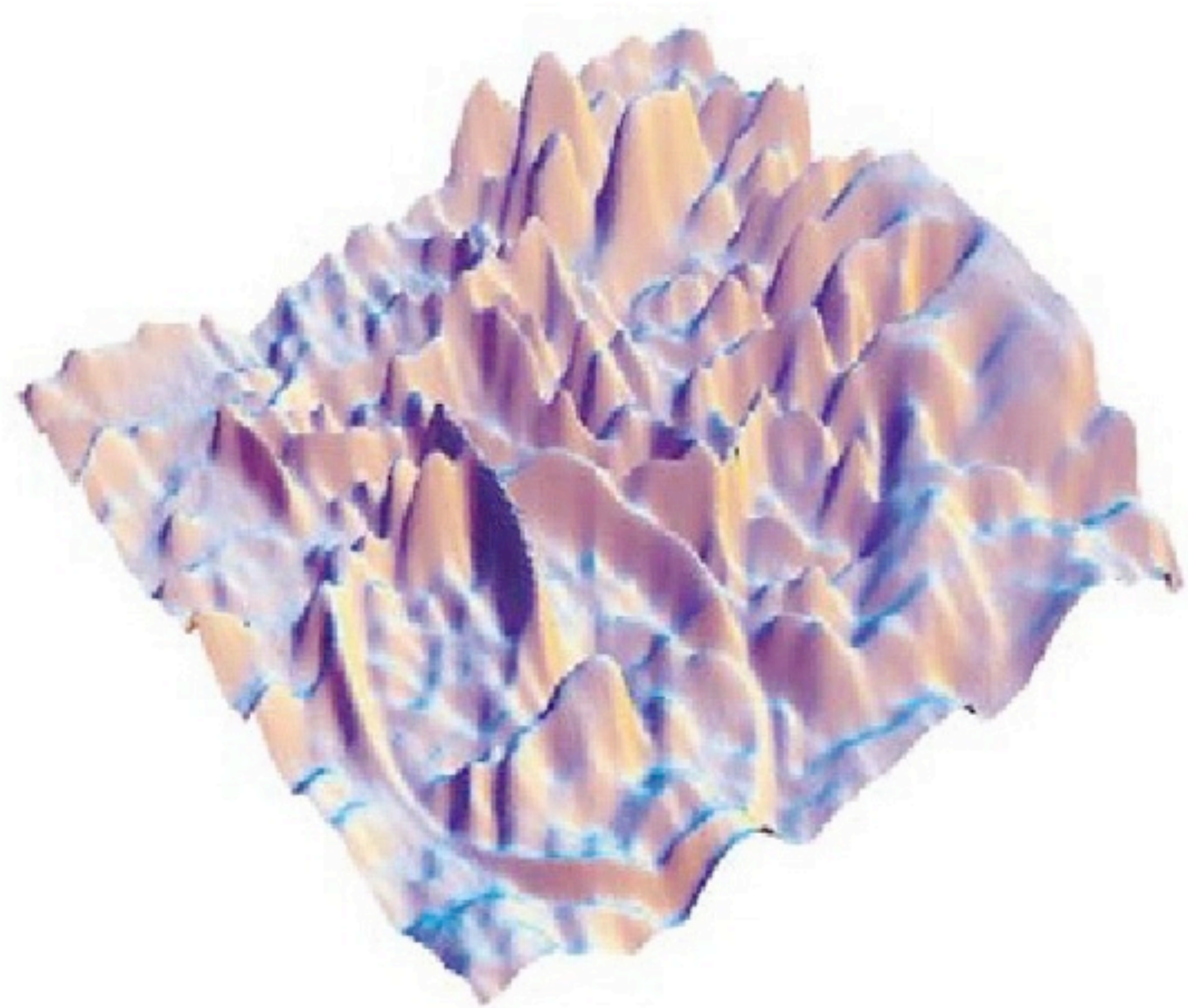


A representation we're more familiar with

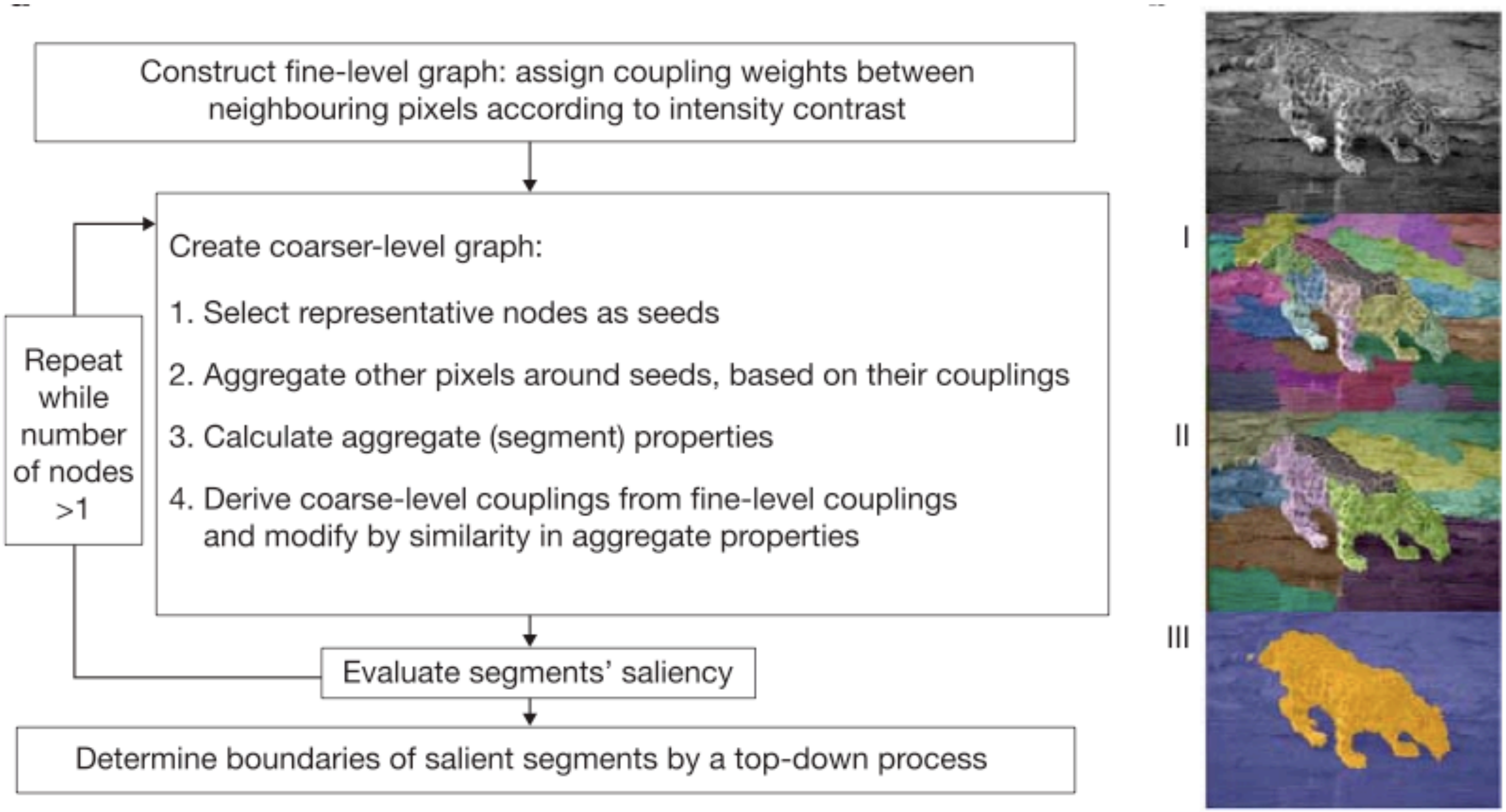
A representation we’re more familiar with

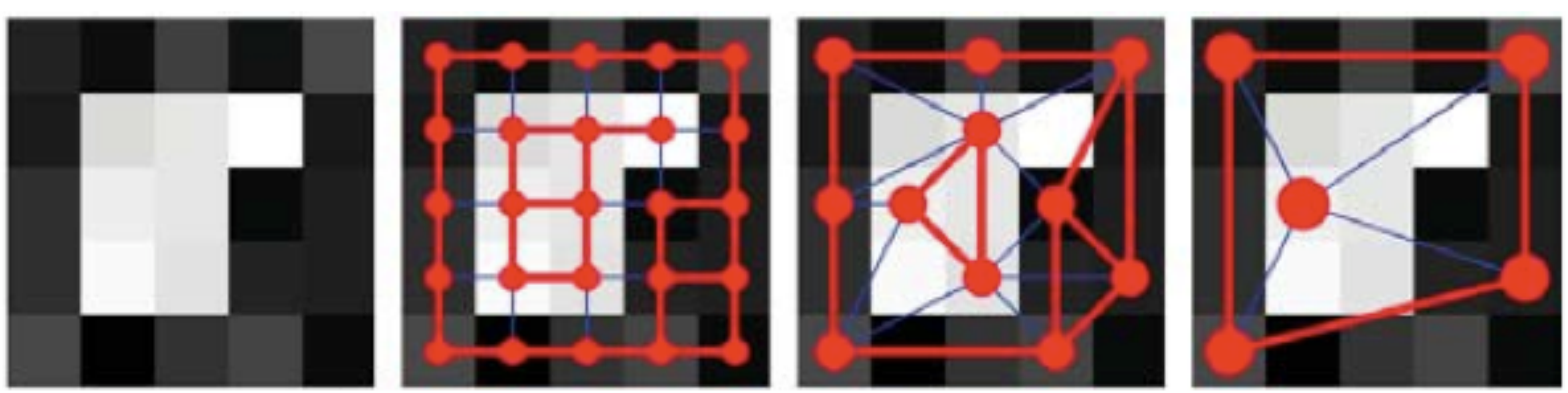
This is what our brain does





Modern segmentation algorithm using graph cuts



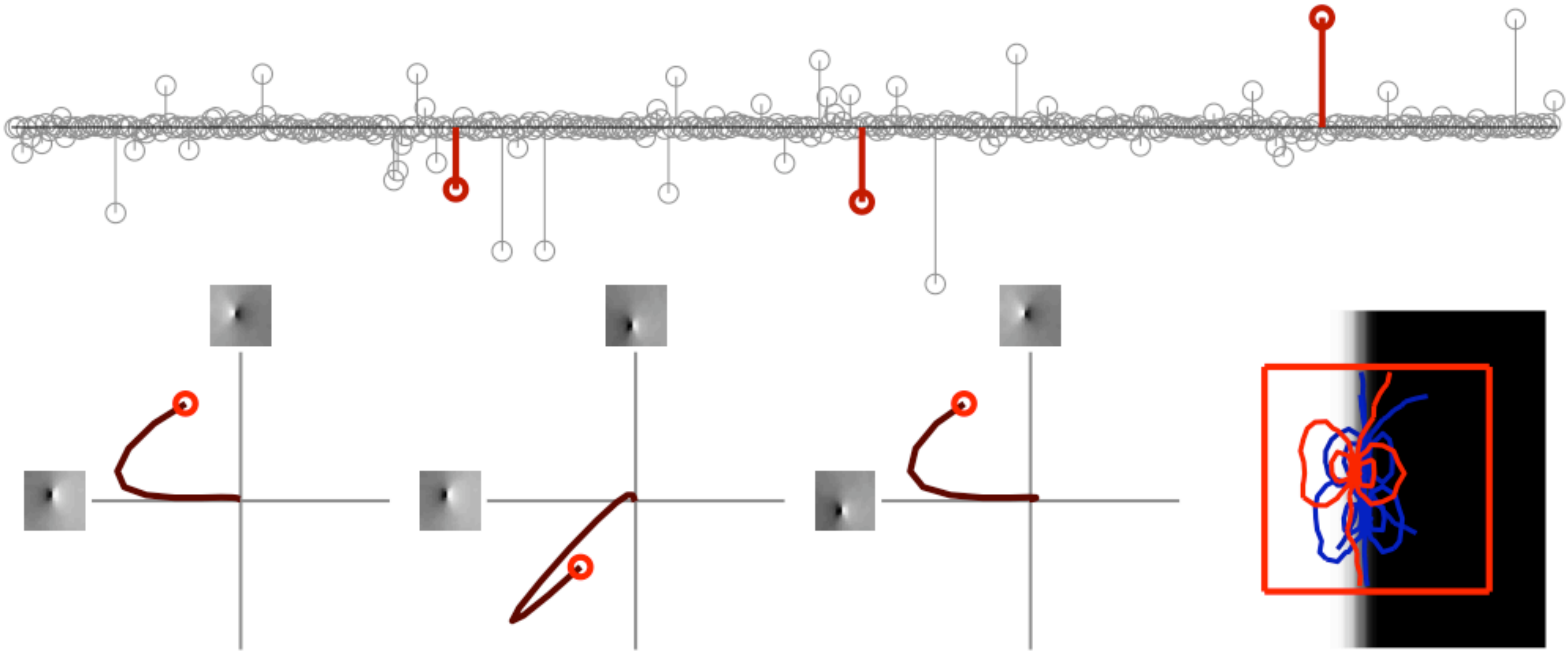


from Sharon et al, 2006

What do we know about the visual system?

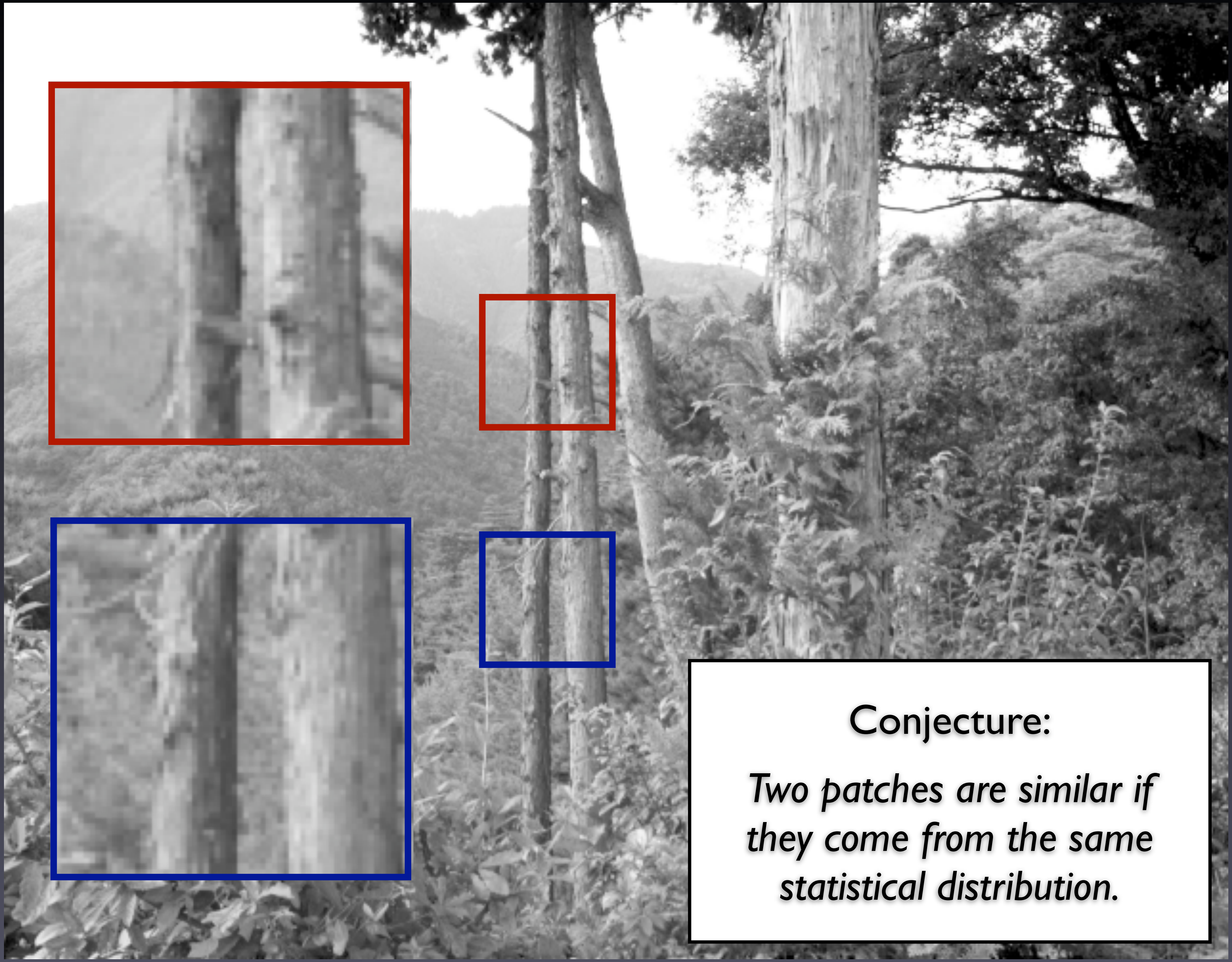
Perceptual generalization in natural scenes

Perceptual generalization in natural scenes



Perceptual generalization in natural scenes

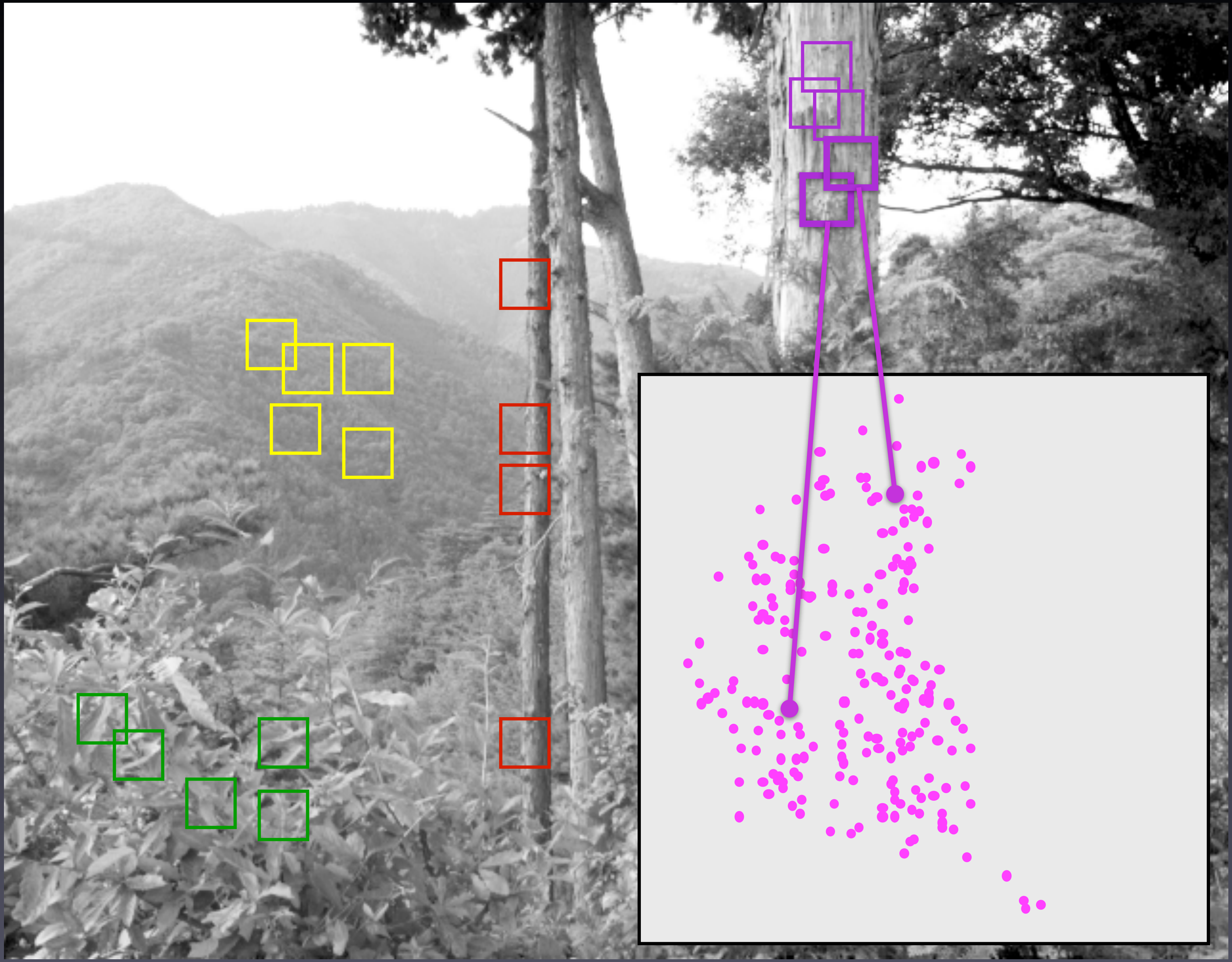
Perceptual generalization in natural scenes



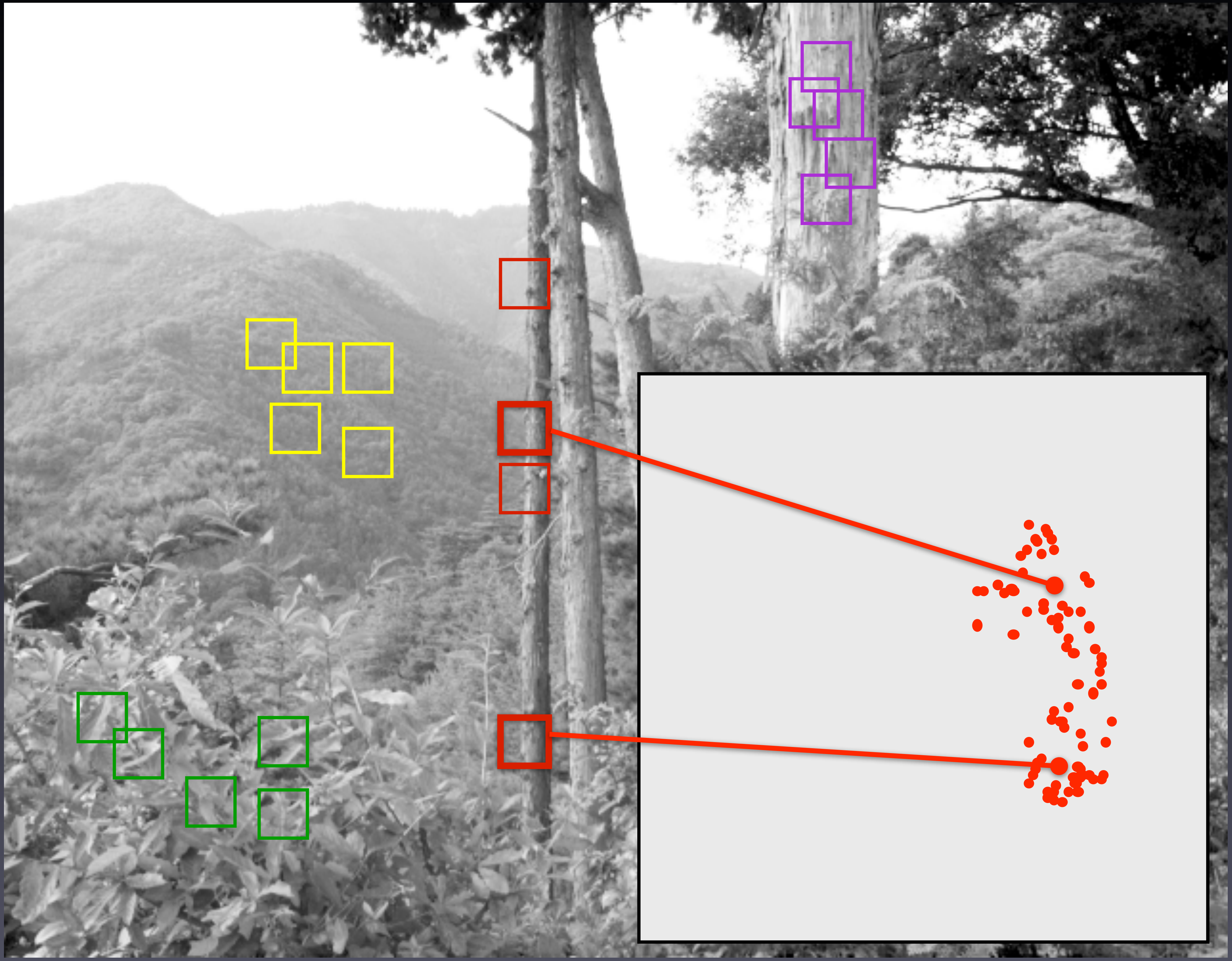
Conjecture:

Two patches are similar if they come from the same statistical distribution.

Generalization by distribution modeling

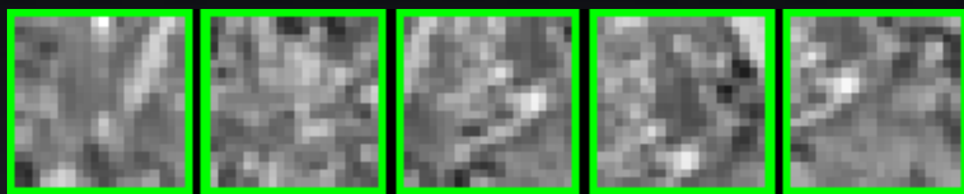


Generalization by distribution modeling



Linear representations do not separate the image classes

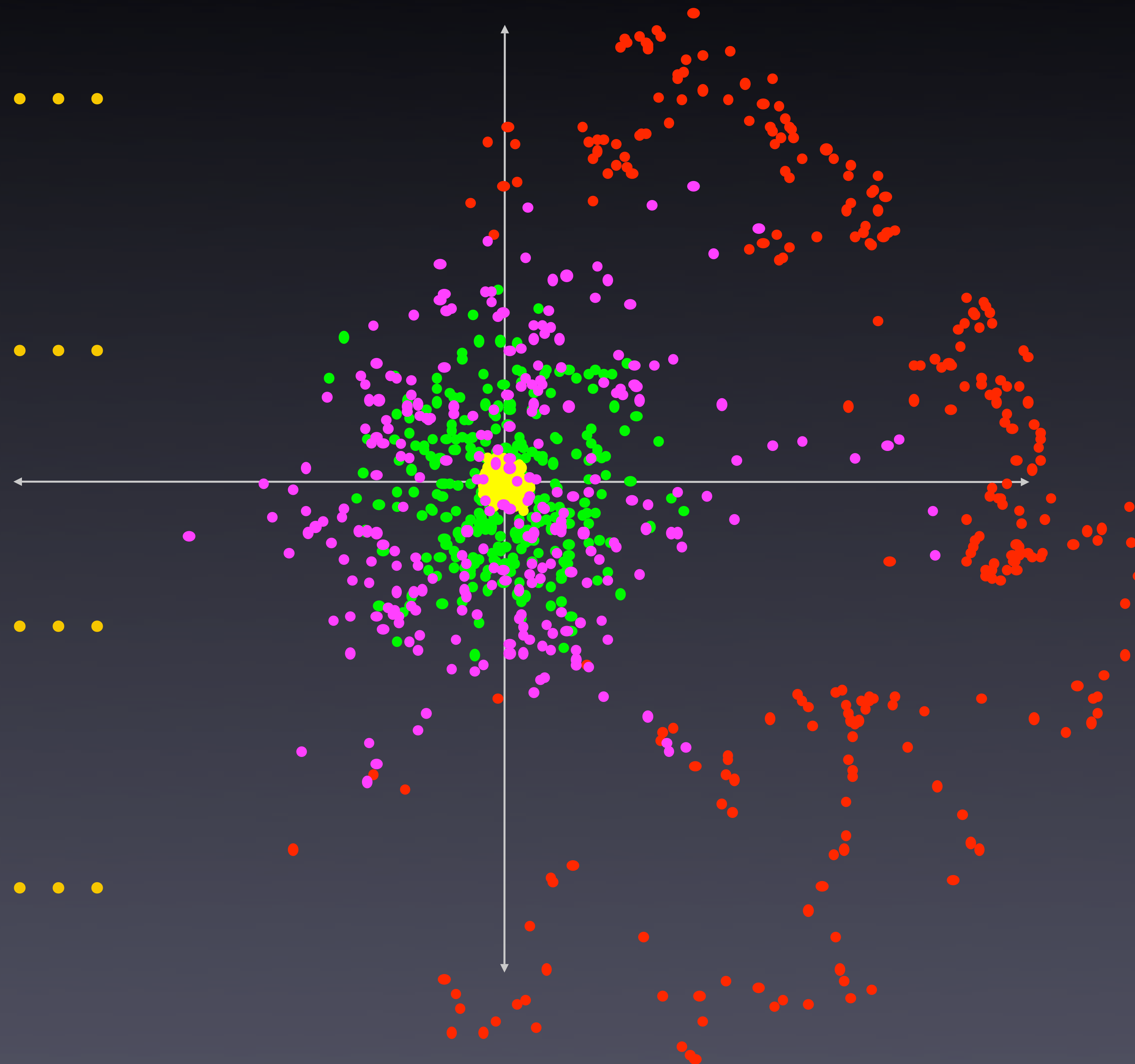
- bushes



- hillside

- tree edge

- tree bark

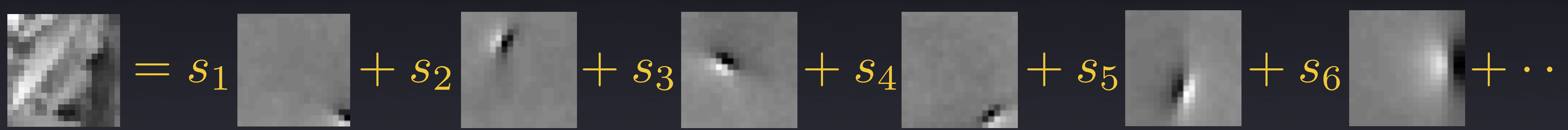


projection onto the first 2 principal components of the data

How can we describe the distribution of local regions?

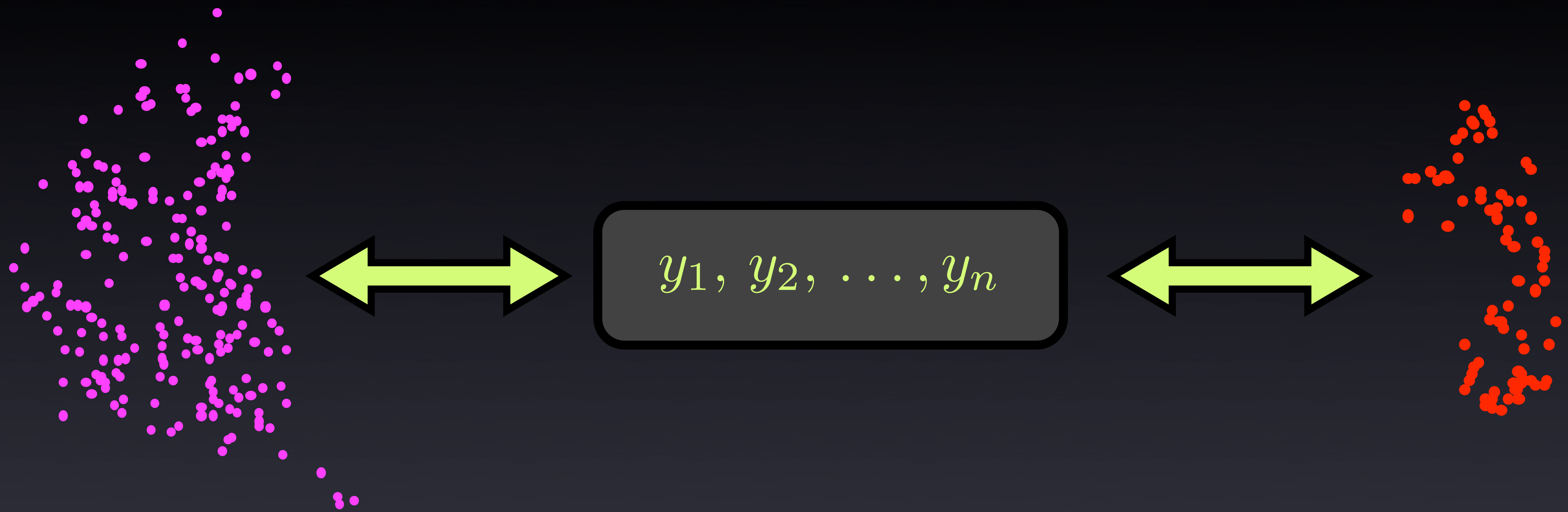
Distributed representations of an image

$$\boxed{} = s_1 \boxed{\phi_1} + s_2 \boxed{\phi_2} + s_3 \boxed{\phi_3} + s_4 \boxed{\phi_4} + s_5 \boxed{\phi_5} + s_6 \boxed{\phi_6} + \cdots$$



Can we have a distributed representation of a distribtion?

Distributed representations of image *distributions*



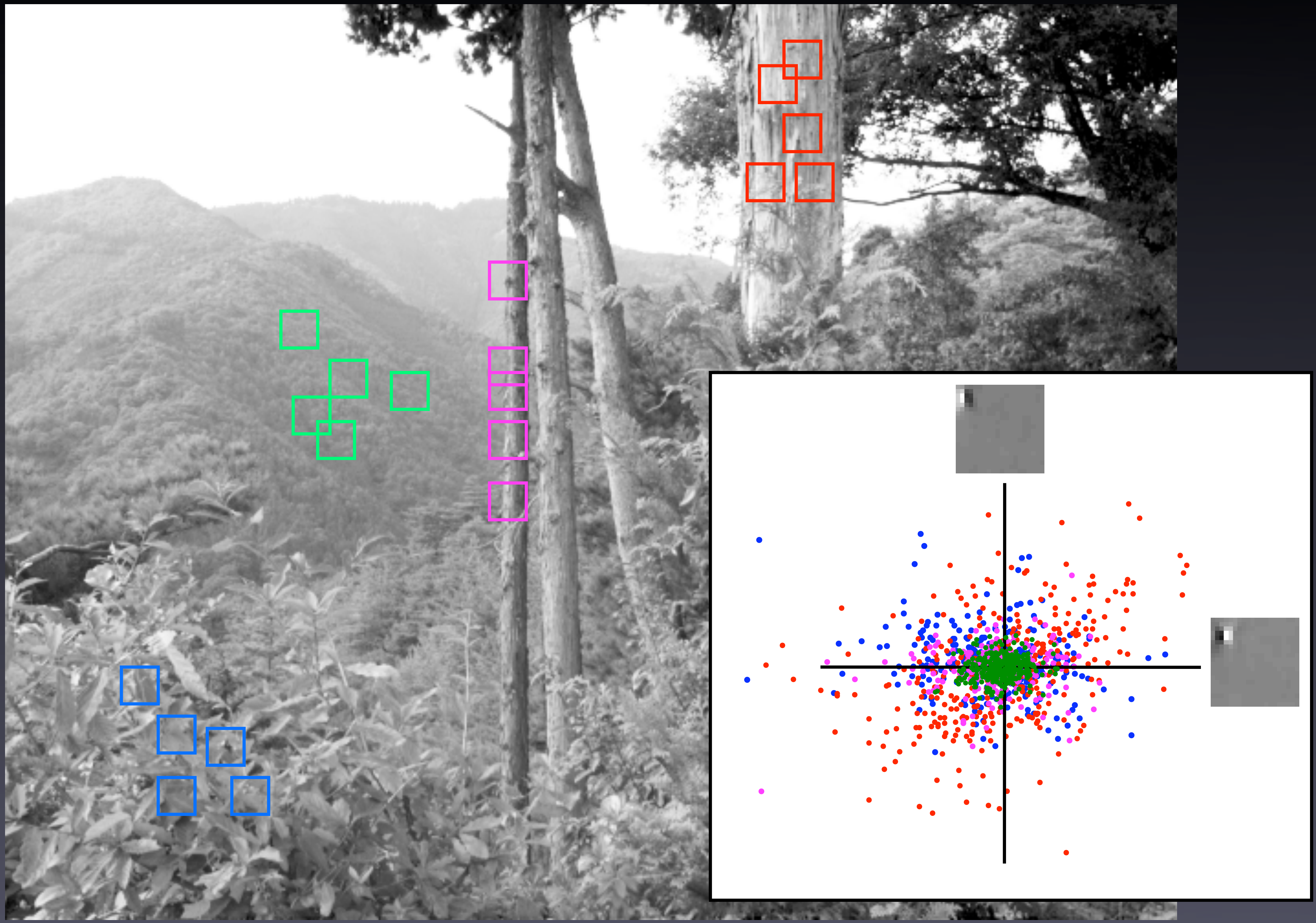
Points from each distribution will have the same representation

$\Rightarrow$ model will **generalize** over local scene region

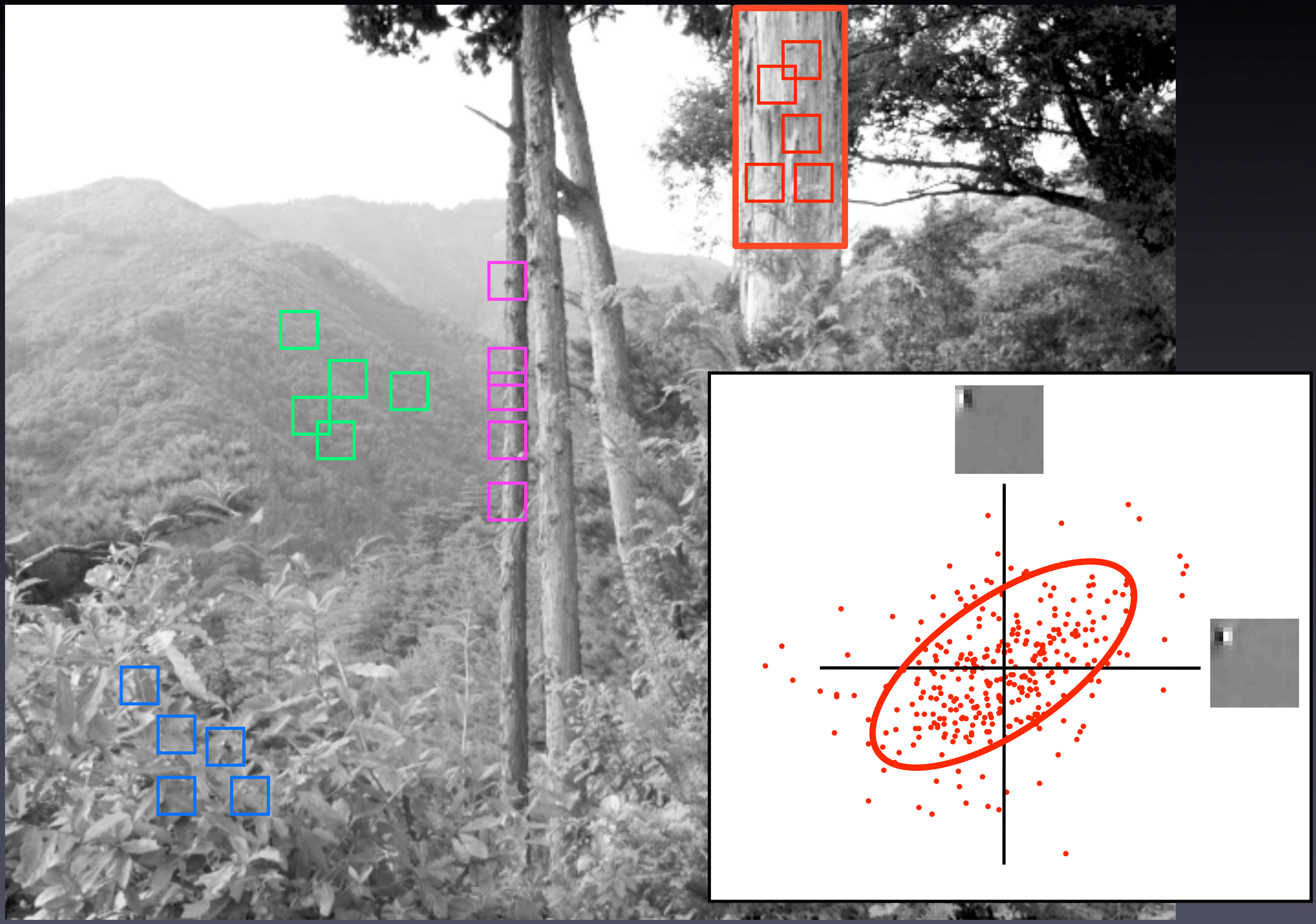
Modeling distributions of local scene regions

- model *local* scene structure, not average scene statistics
- model *all* structure
  - want a "complete" code
  - a universal "texture" model
- code should be *distributed* and statistically *efficient*

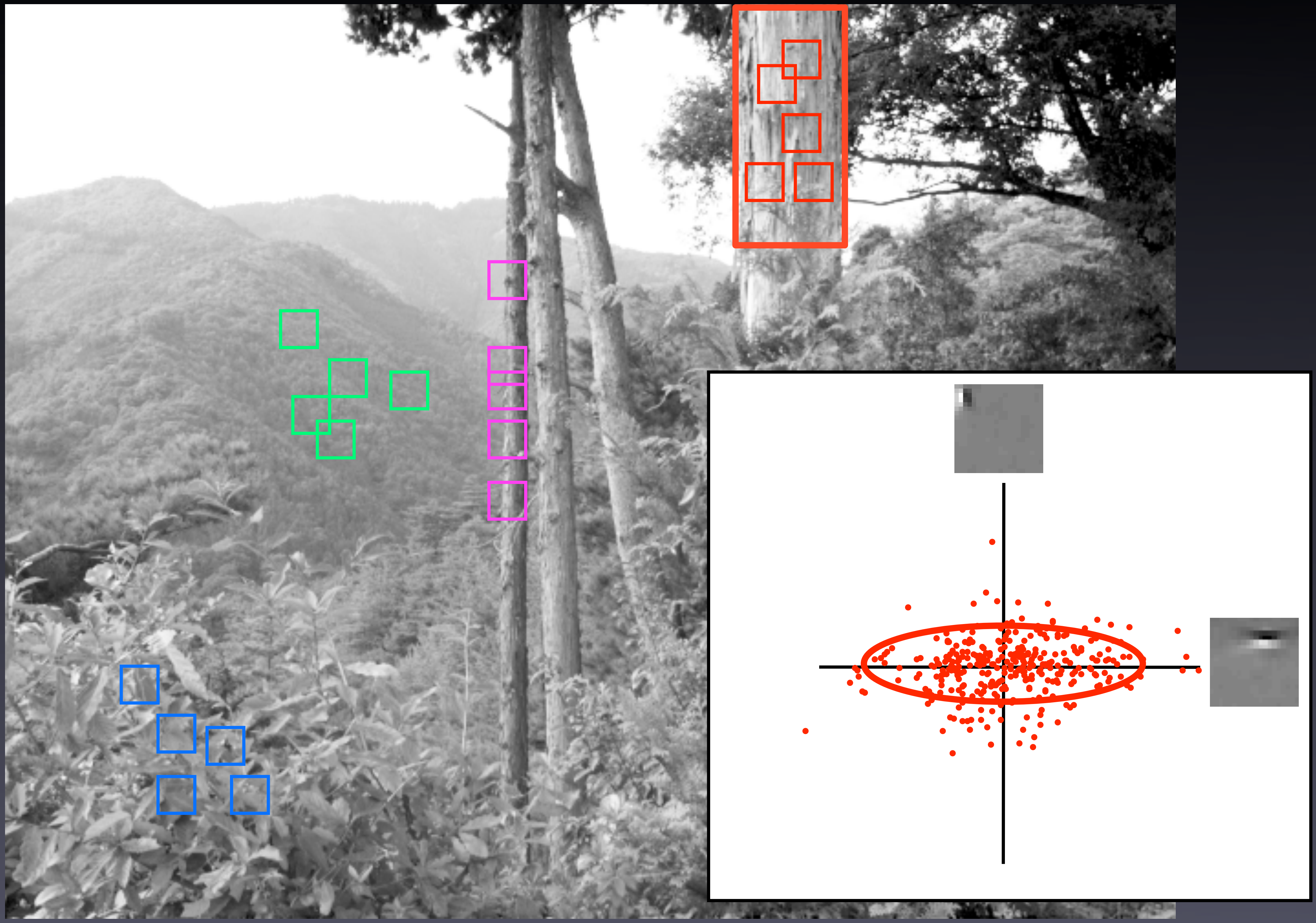
Modeling distributions of local scene regions



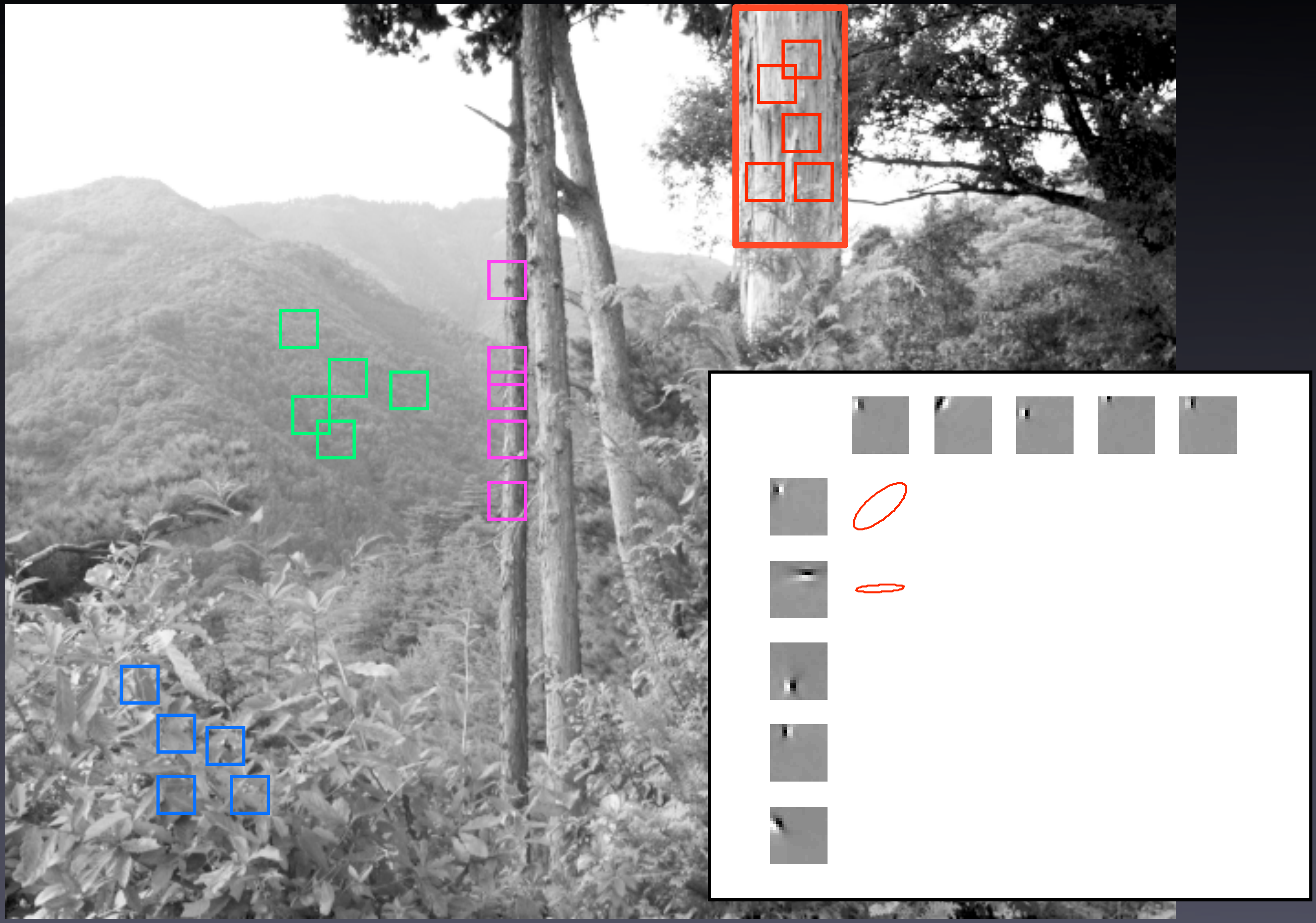
Specific regions have subtle and characteristic correlations



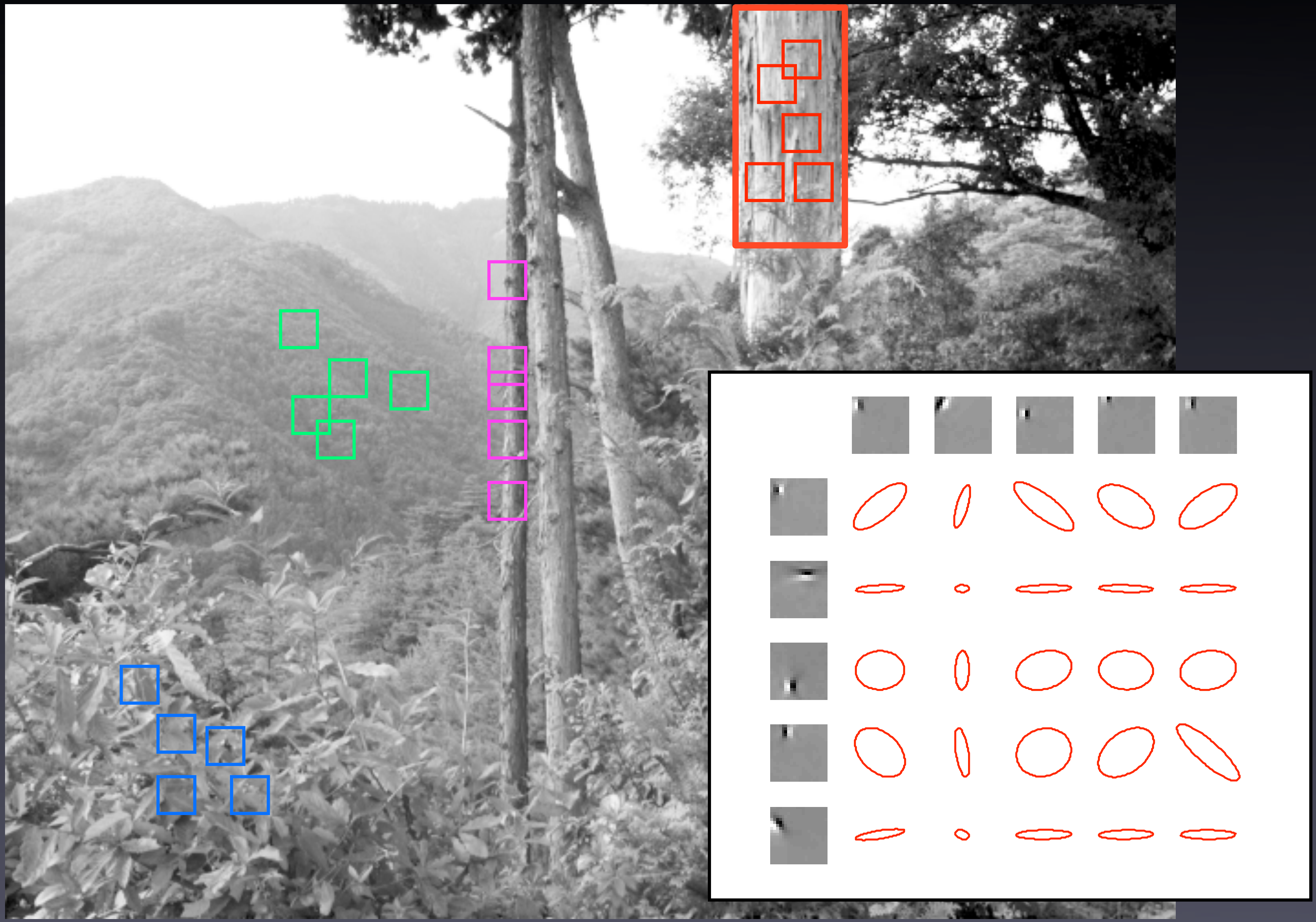
Summarize all pair-wise correlations, ie the covariance



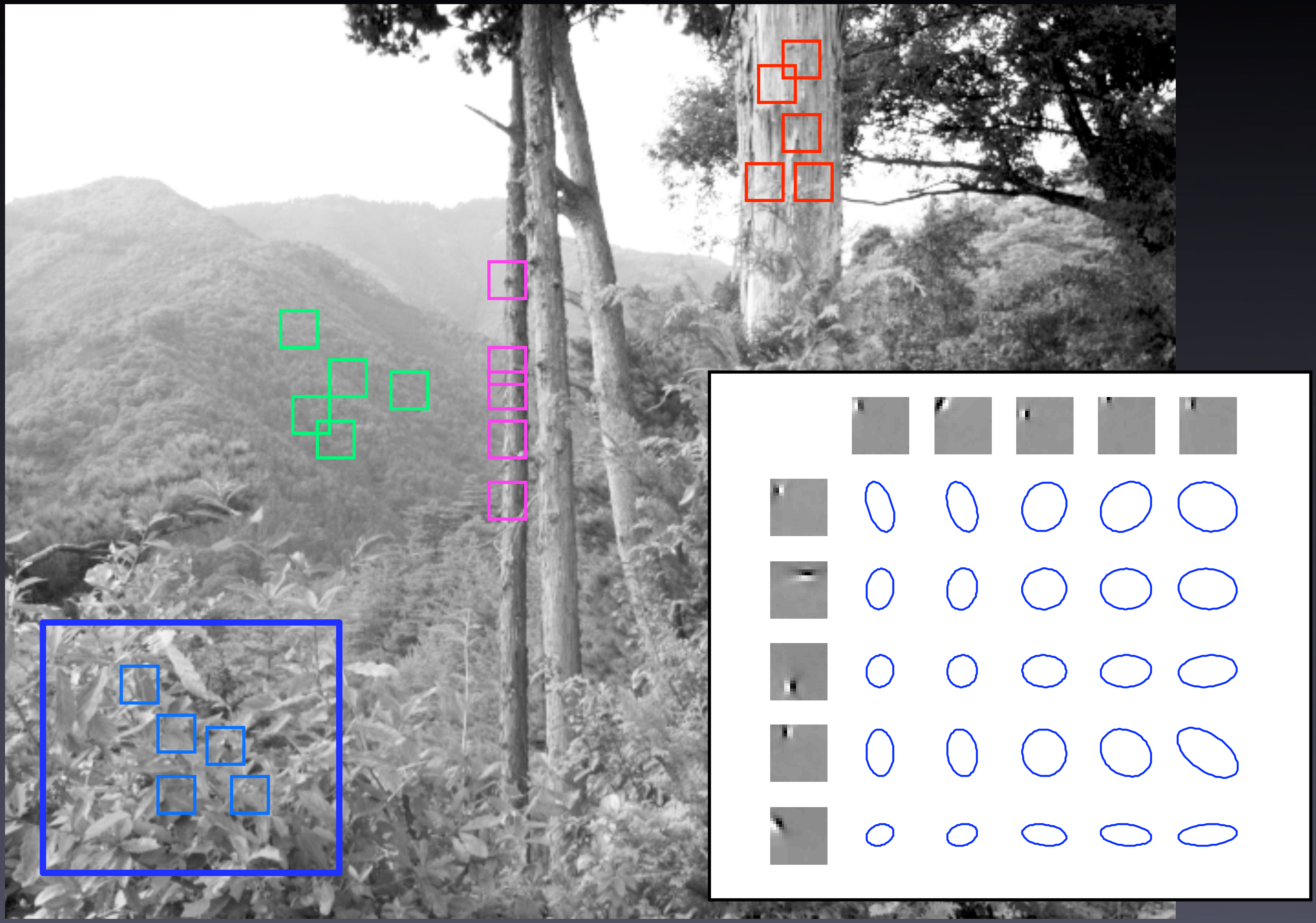
Region shows a characteristic pattern of correlations



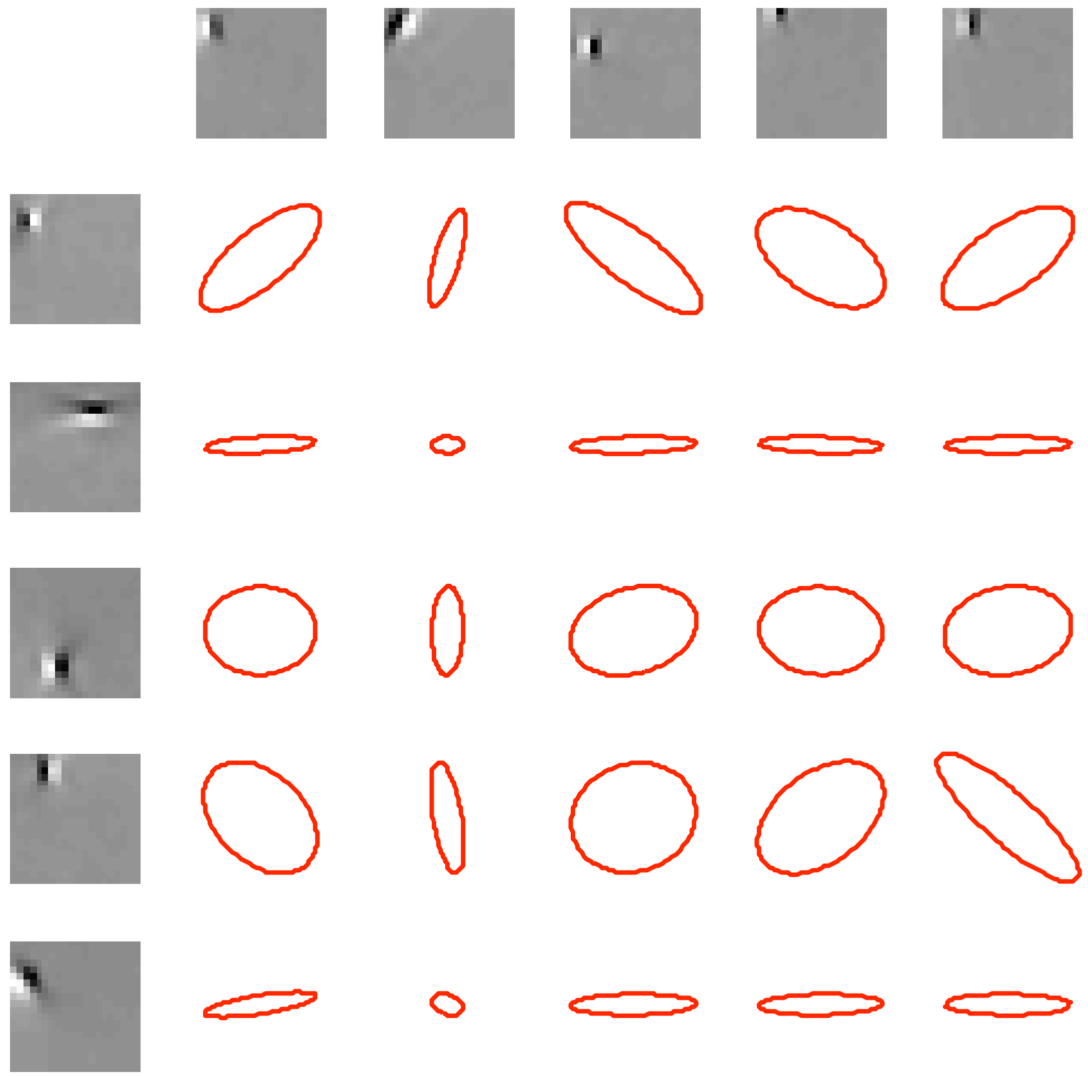
Region shows a characteristic pattern of correlations



Region shows a characteristic pattern of correlations



Patterns captured by the covariance matrix



multivariate Gaussian model

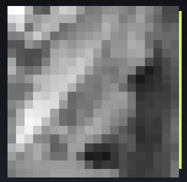
$$\mathbf{x} \sim \mathcal{N}(0, \mathbf{C})$$

neural activity describes the covariance

$$\mathbf{C} = f(\mathbf{y})$$

A distributed code for covariance matrices

An efficient image code: linear basis:

 $= s_1 \boxed{\phi_1} + s_2 \boxed{\phi_2} + s_3 \boxed{\phi_3} + s_4 \boxed{\phi_4} + s_5 \boxed{\phi_5} + \cdots$

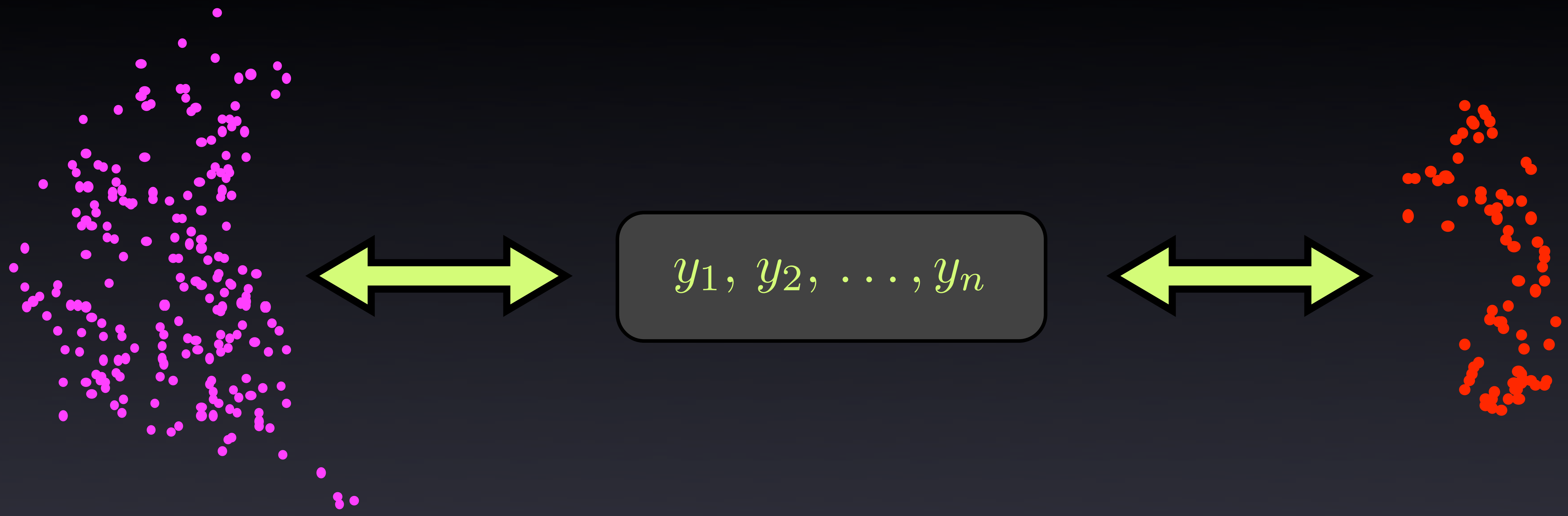
A basis for covariance matrices?

$$\boxed{\mathbf{C}} = y_1 \boxed{\mathbf{A}^1} + y_2 \boxed{\mathbf{A}^2} + y_3 \boxed{\mathbf{A}^3} + y_4 \boxed{\mathbf{A}^4} + \cdots$$

Represent basis using log covariance:

$$\boxed{\log \mathbf{C}} = y_1 \boxed{\mathbf{A}^1} + y_2 \boxed{\mathbf{A}^2} + y_3 \boxed{\mathbf{A}^3} + y_4 \boxed{\mathbf{A}^4} + \cdots$$

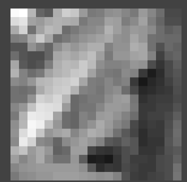
Distributed representations of image *distributions*



Points from each distribution will have the same representation

$\Rightarrow$ model will **generalize** over local scene region

Compare to linear basis coding

 $= s_1 \boxed{\phi_1} + s_2 \boxed{\phi_2} + \cdots$

$\boxed{\log \mathbf{C}} = y_1 \boxed{\mathbf{A}^1} + y_2 \boxed{\mathbf{A}^2} + \cdots$

- code a single image:

$$\mathbf{x} = \sum_i s_i \phi_i$$

- images represented exactly
- no activity $\Rightarrow$ blank image

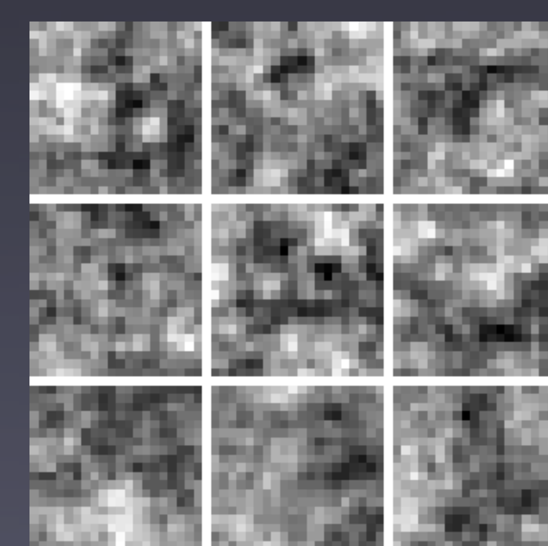
$$\mathbf{s} = \mathbf{0}$$
$$\Rightarrow$$
$$\mathbf{x} = \mathbf{0}$$

- code a single distribution:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{C})$$
$$\log \mathbf{C} = \sum_j y_j \mathbf{A}^j$$

- exact image not encoded
- no activity $\Rightarrow$ “canonical” distribution

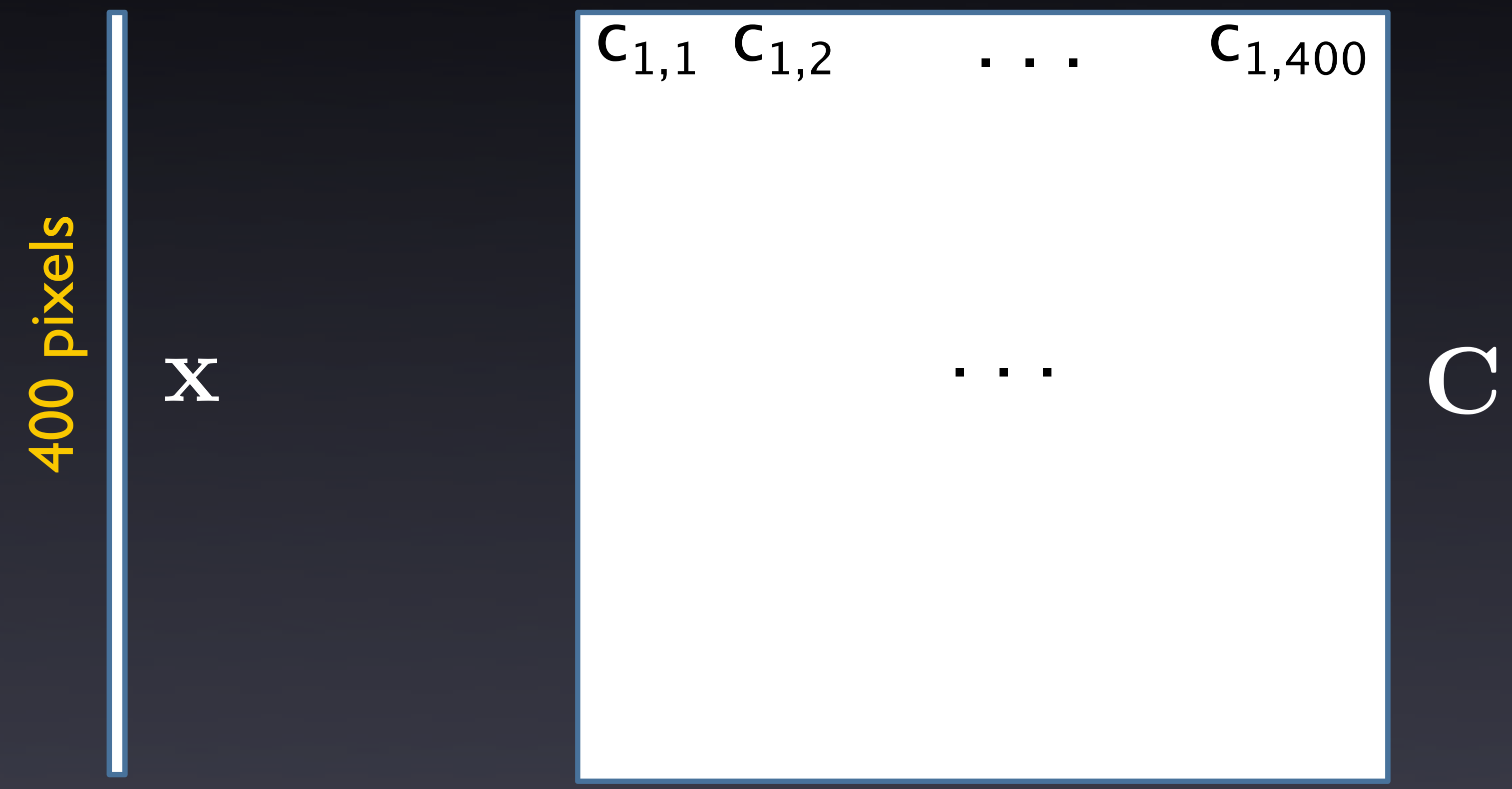
$$\mathbf{y} = \mathbf{0}$$
$$\Rightarrow$$
$$\log \mathbf{C} = \mathbf{0}$$
$$\Rightarrow$$
$$\mathbf{C} = \mathbf{I}$$



Size of models for 20x20 pixel image patches

linear model

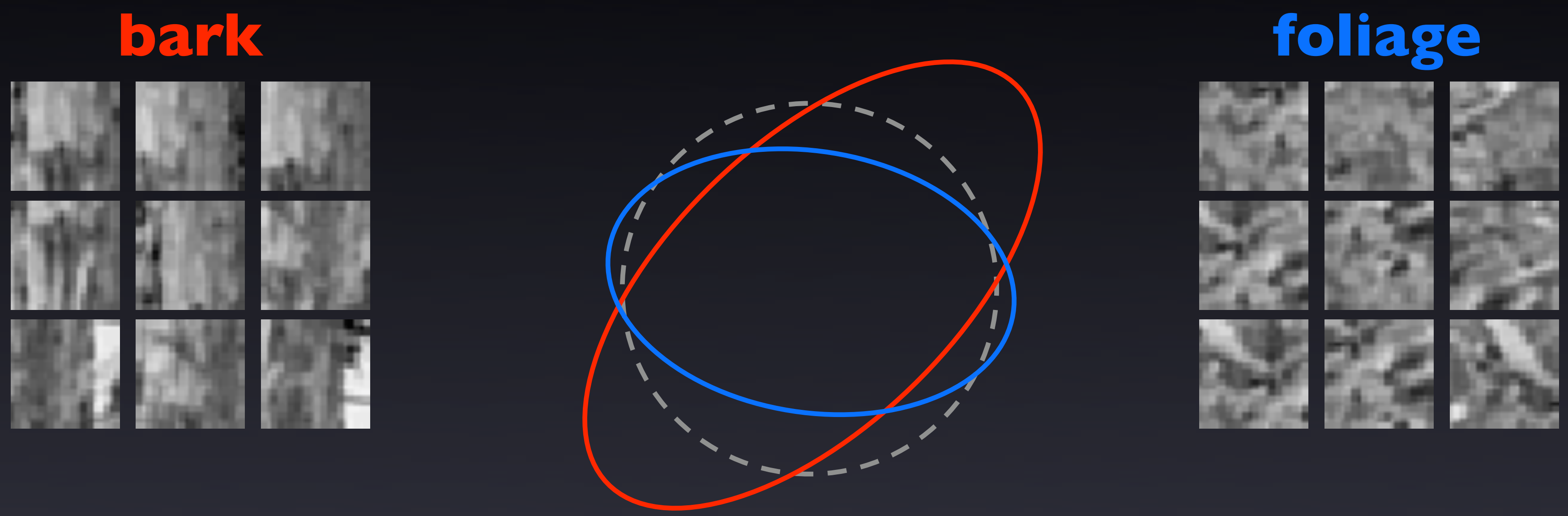
covariance model



	linear model	covariance model
# basis functions:	400	$\approx 400^2/2 = 80{,}000$
# parameters:	400^2	$80{,}000^2$

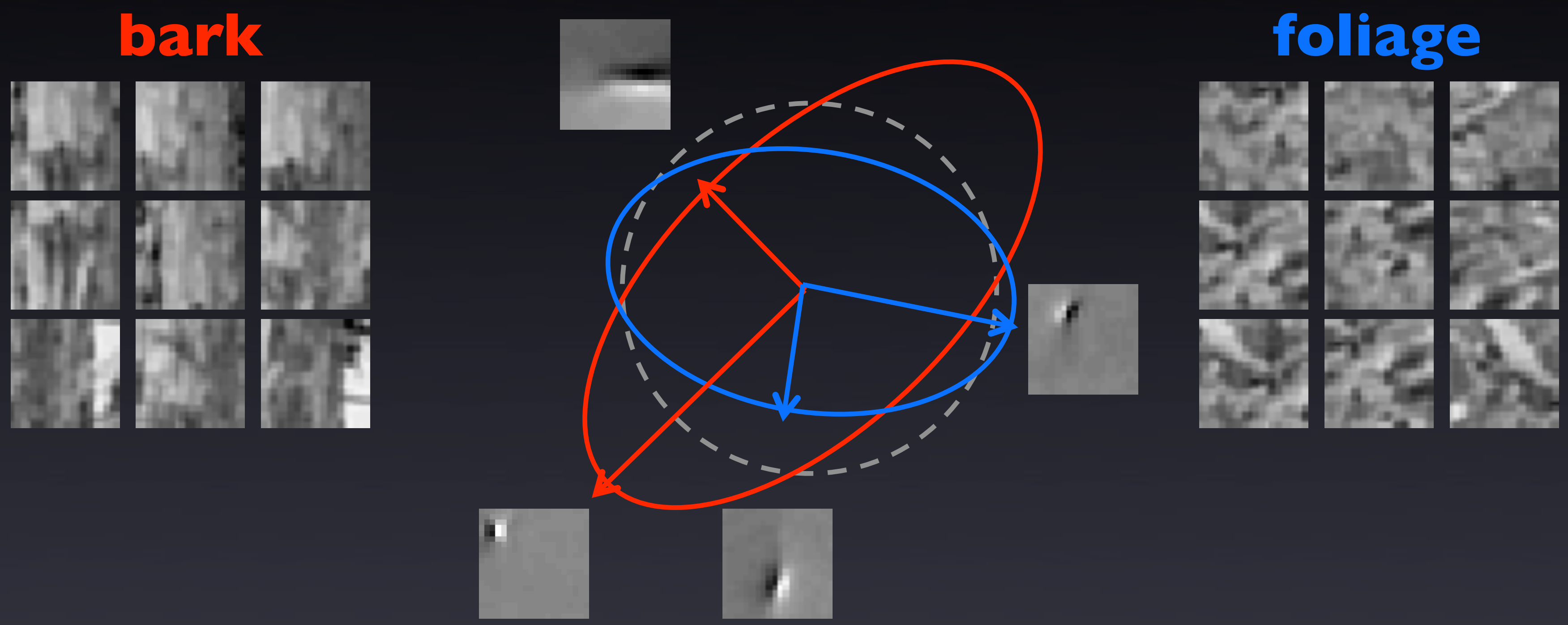
Learning covariance basis is infeasible ⇒ need to simplify

Parameterizing covariance components



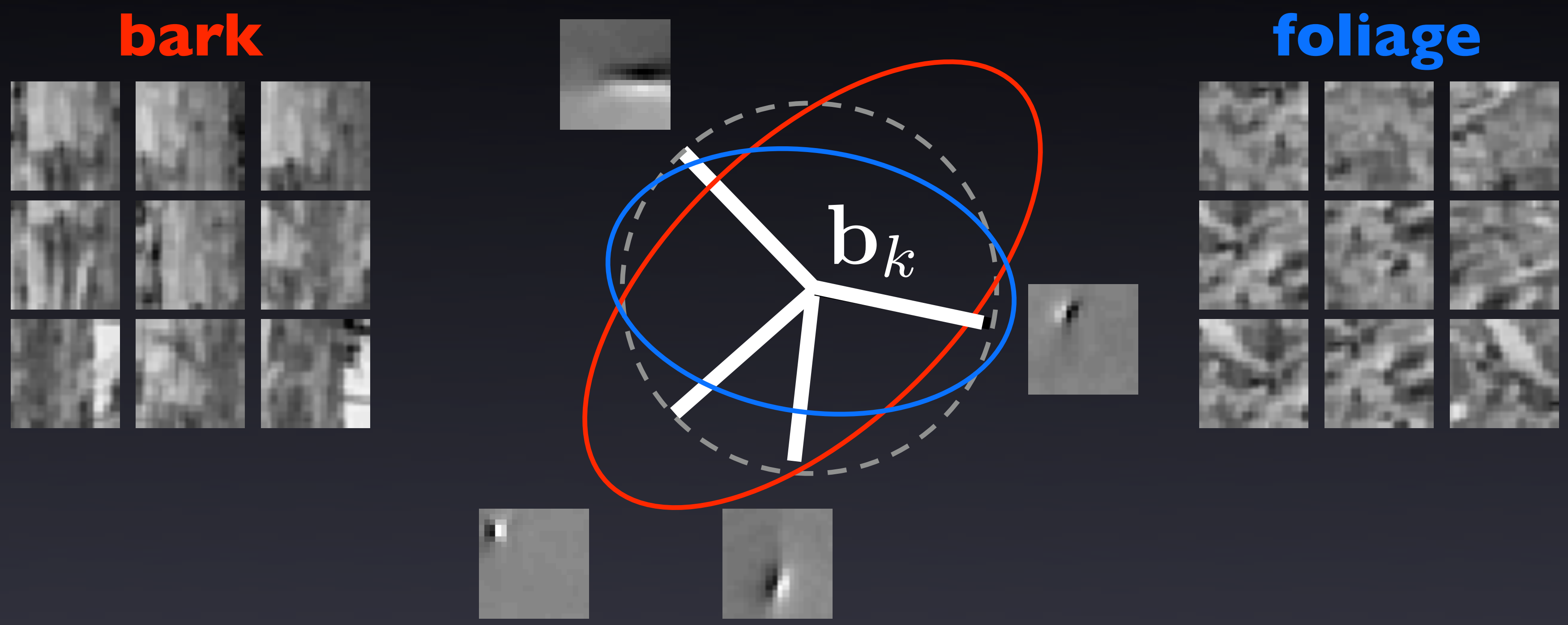
Do all patterns of co-variation occur in natural images?

Intuition behind the model parameterization



b_k: common *directions of change* in variation/correlation

Intuition behind the model parameterization

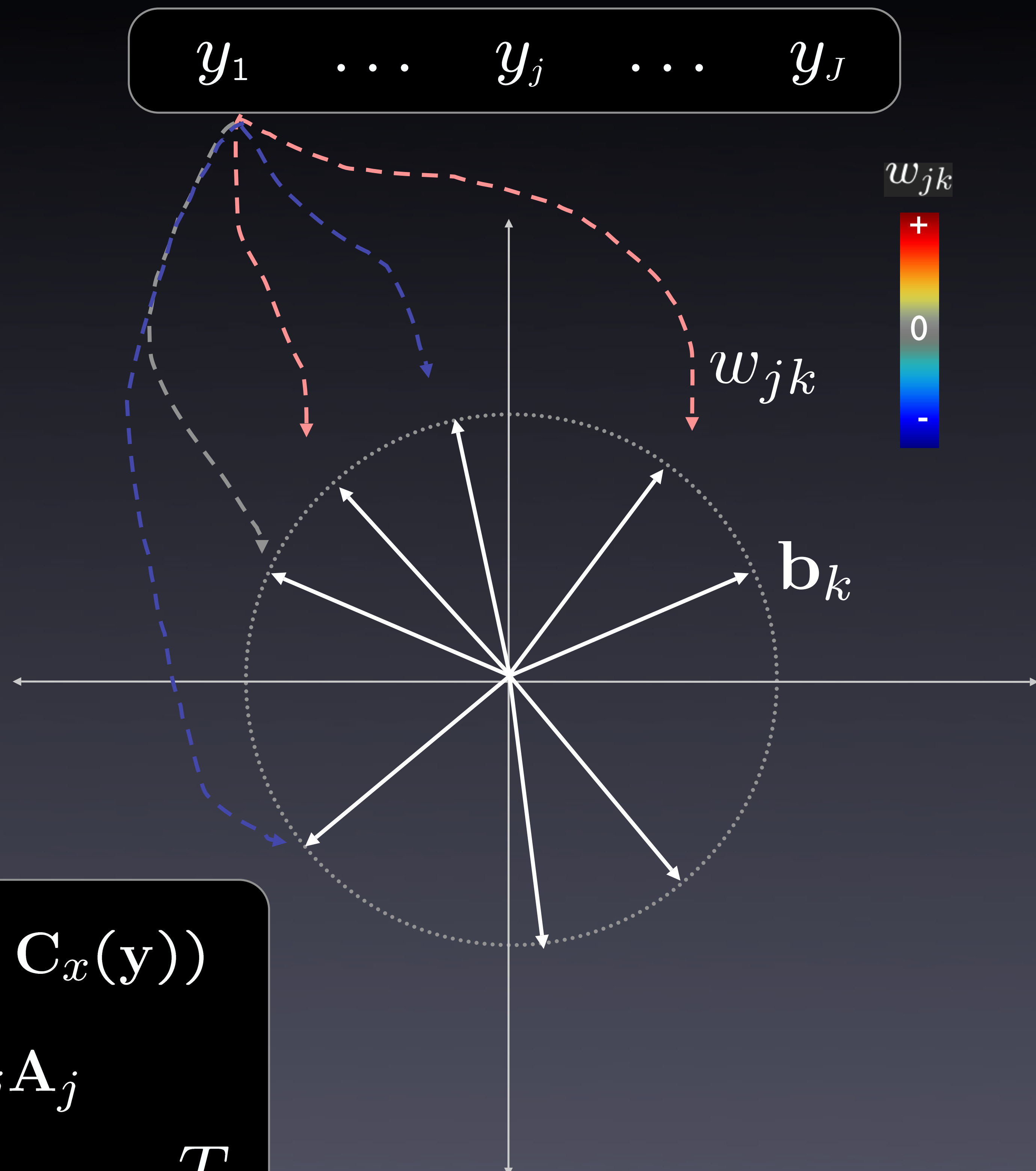


$\mathbf{b}_k$ vectors allow a *much* more compact description of the components

$$\mathbf{A}^j = \sum_k w_{jk} \mathbf{b}_k \mathbf{b}_k^T$$

1000 $\mathbf{b}_k$s: # params: *1000 x 400 = 4e5* *vs.* *80,000² = 6.4e9*

$\mathbf{b}_k$ vectors are shared

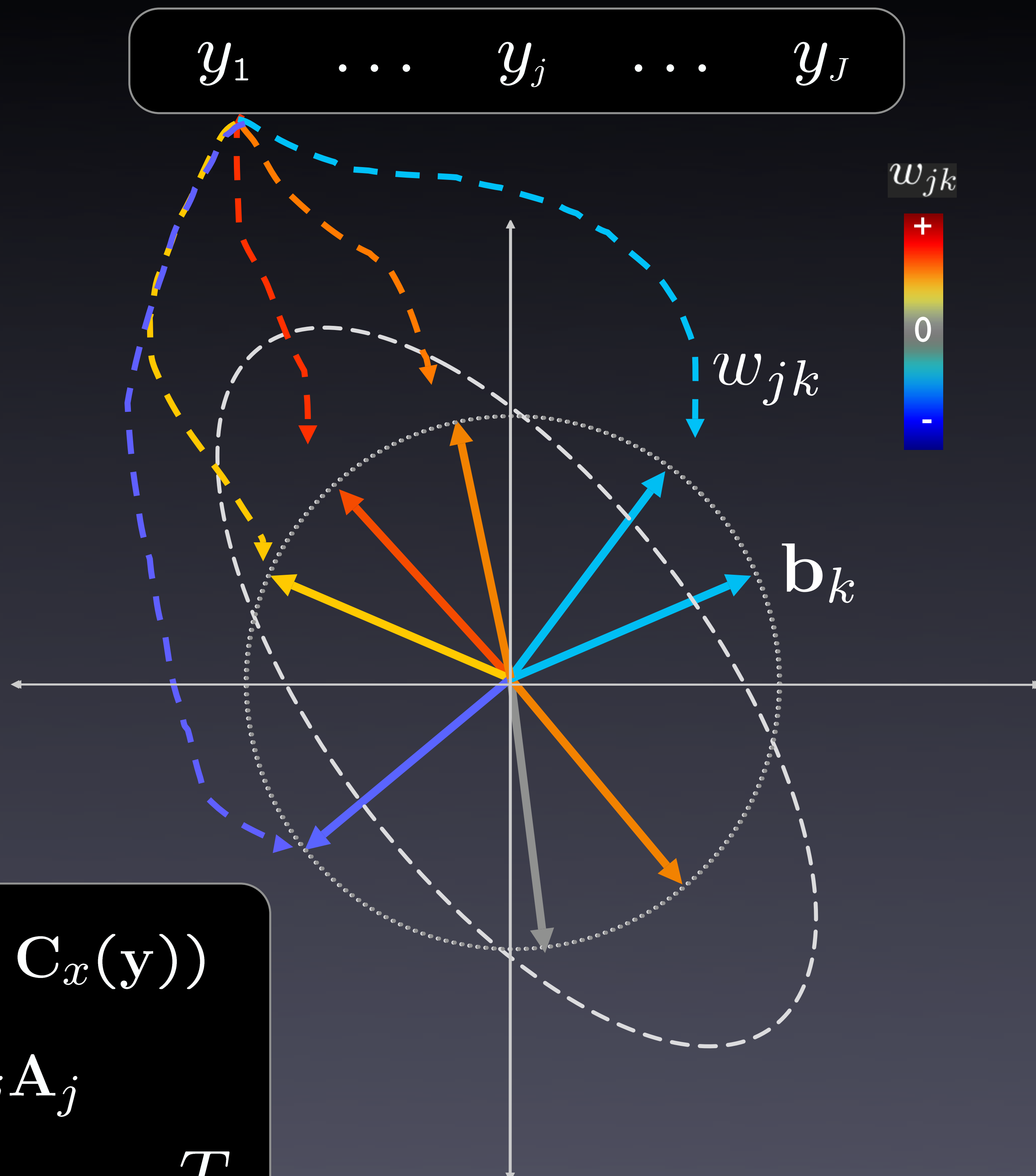


$$P(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{C}_x(\mathbf{y}))$$

$$\log \mathbf{C}_x = \sum_j y_j \mathbf{A}_j$$

$$\mathbf{A}_j = \sum_k w_{jk} \mathbf{b}_k \mathbf{b}_k^T$$

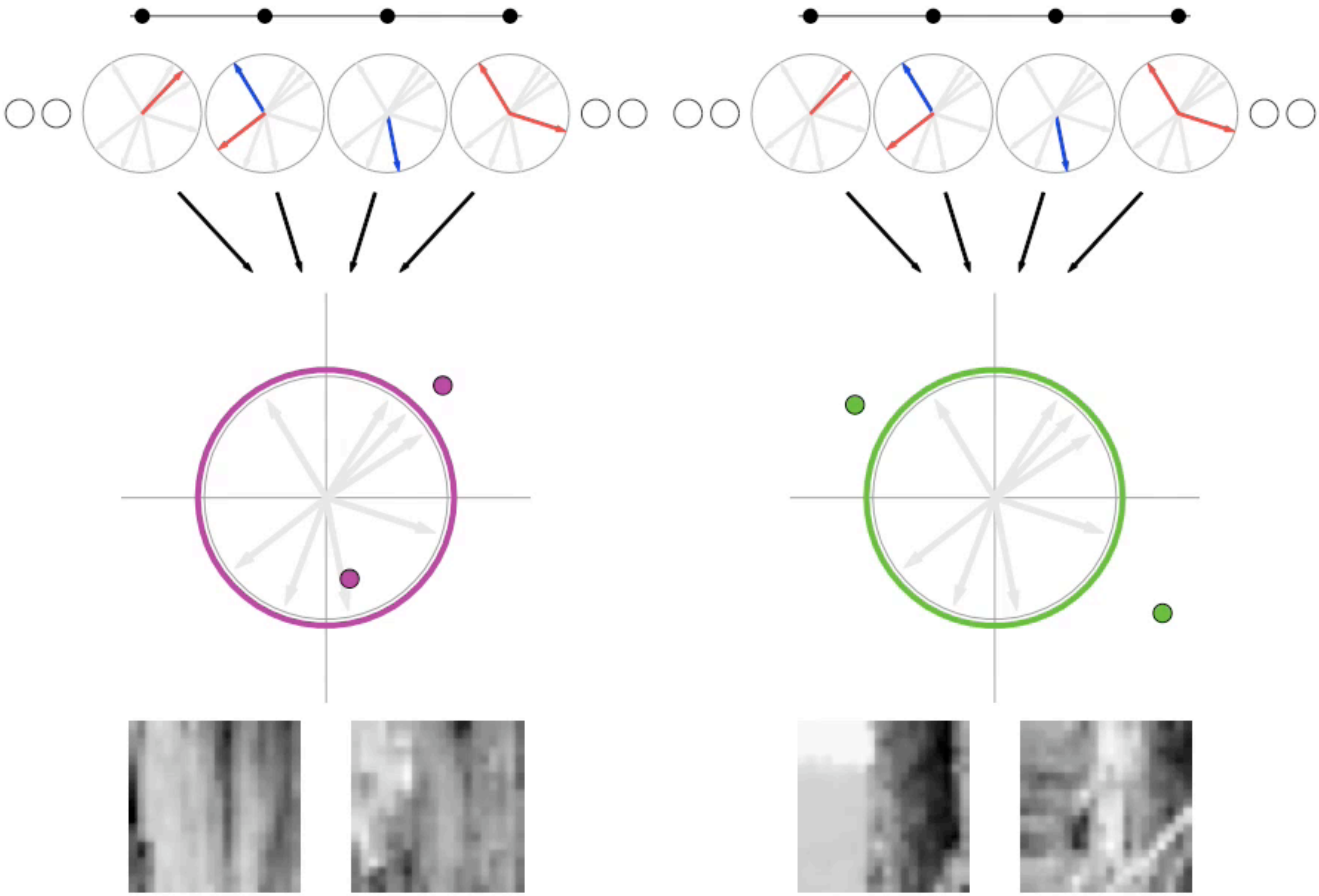
$\mathbf{b}_k$ vectors are shared

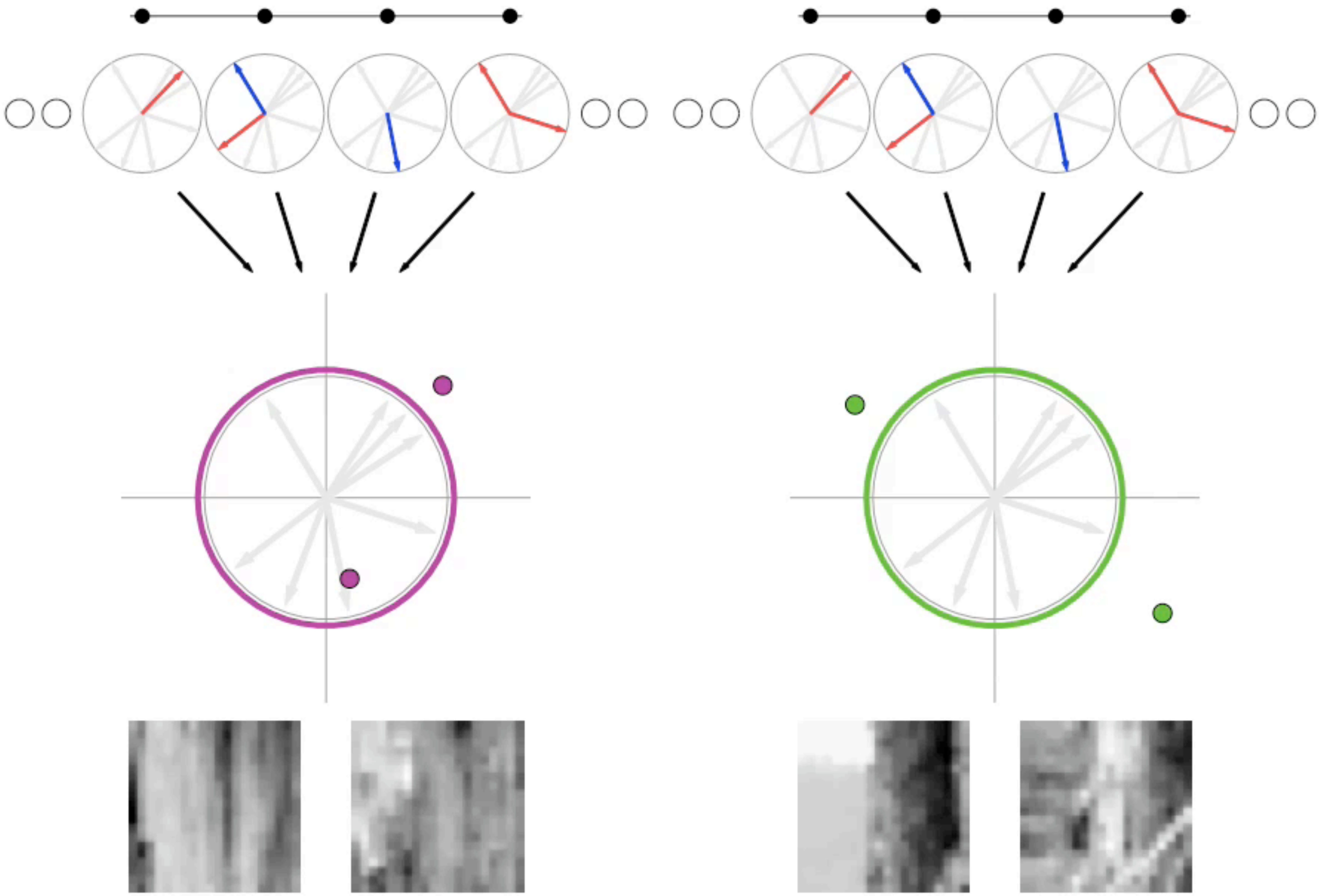


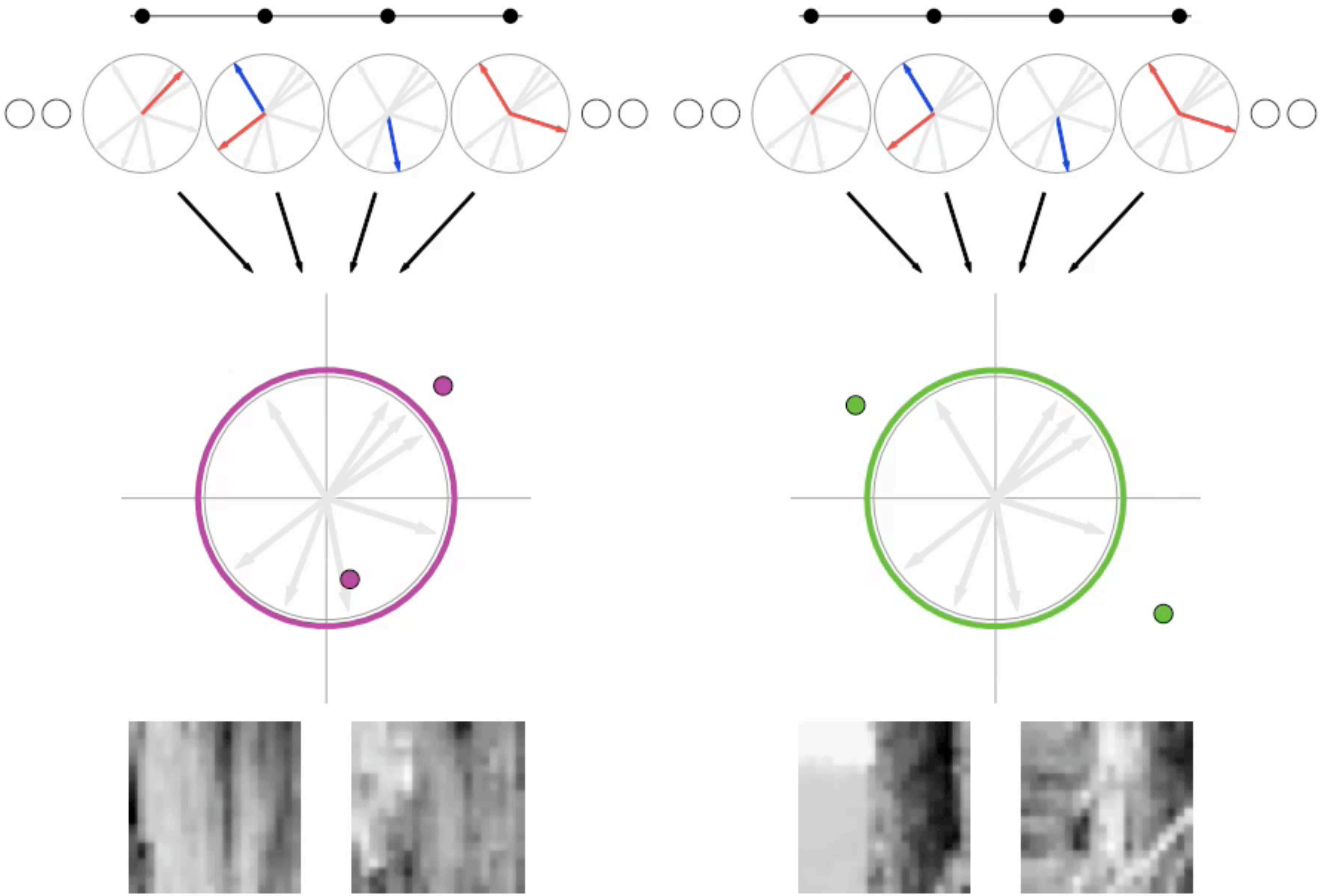
$$P(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{C}_x(\mathbf{y}))$$

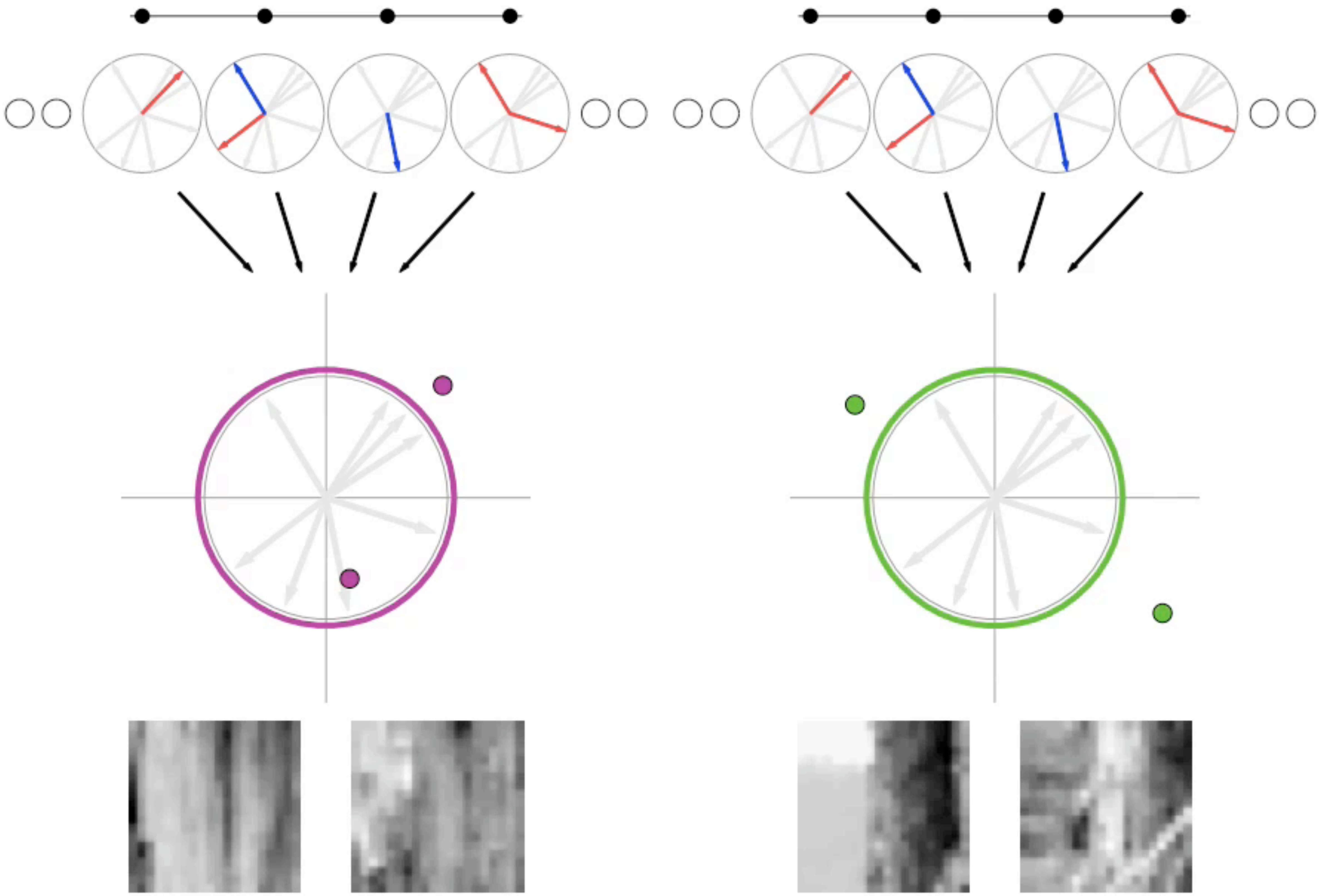
$$\log \mathbf{C}_x = \sum_j y_j \mathbf{A}_j$$

$$\mathbf{A}_j = \sum_k w_{jk} \mathbf{b}_k \mathbf{b}_k^T$$



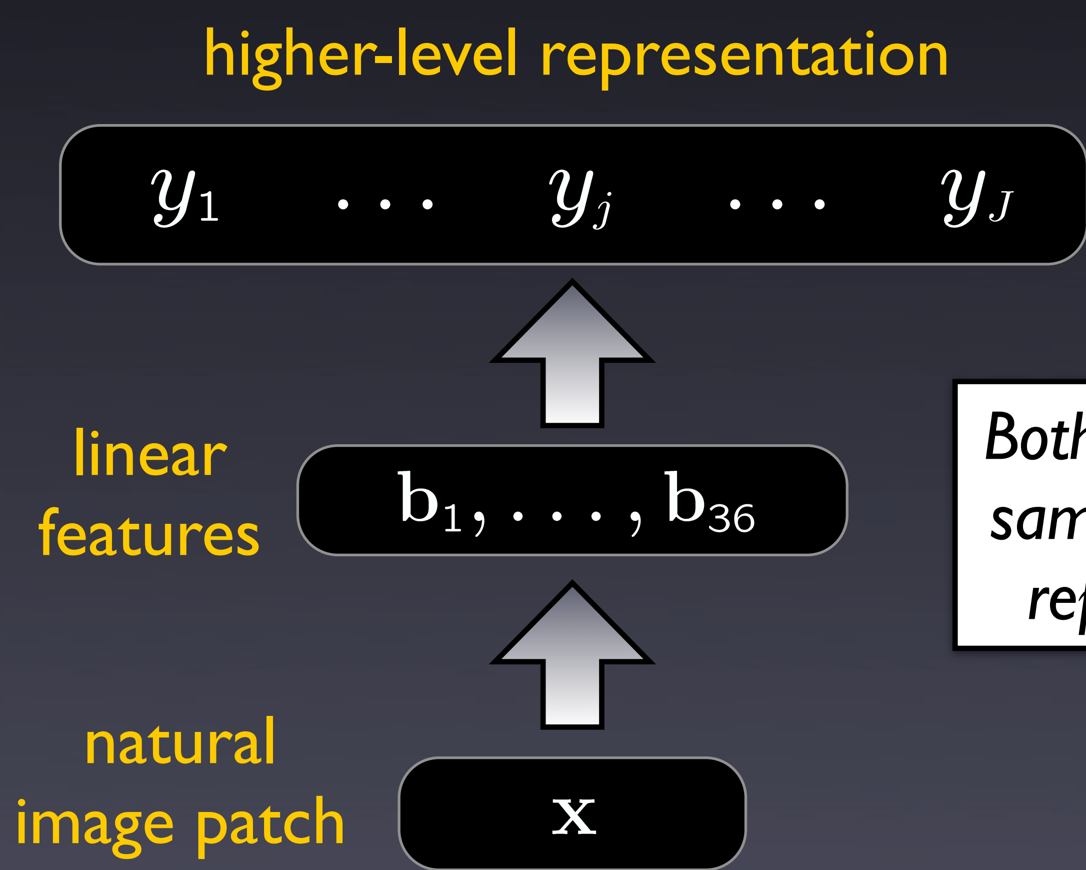


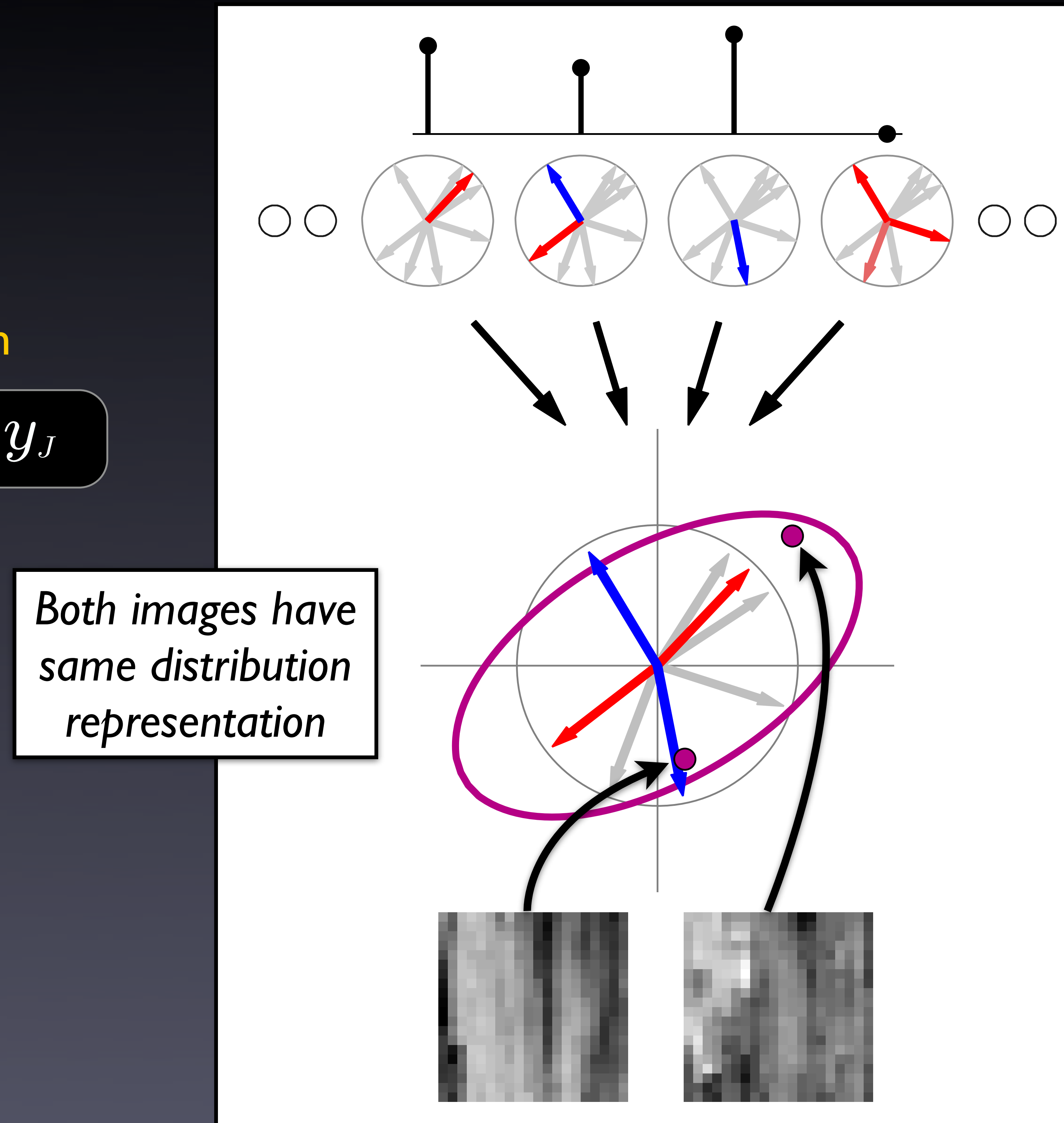




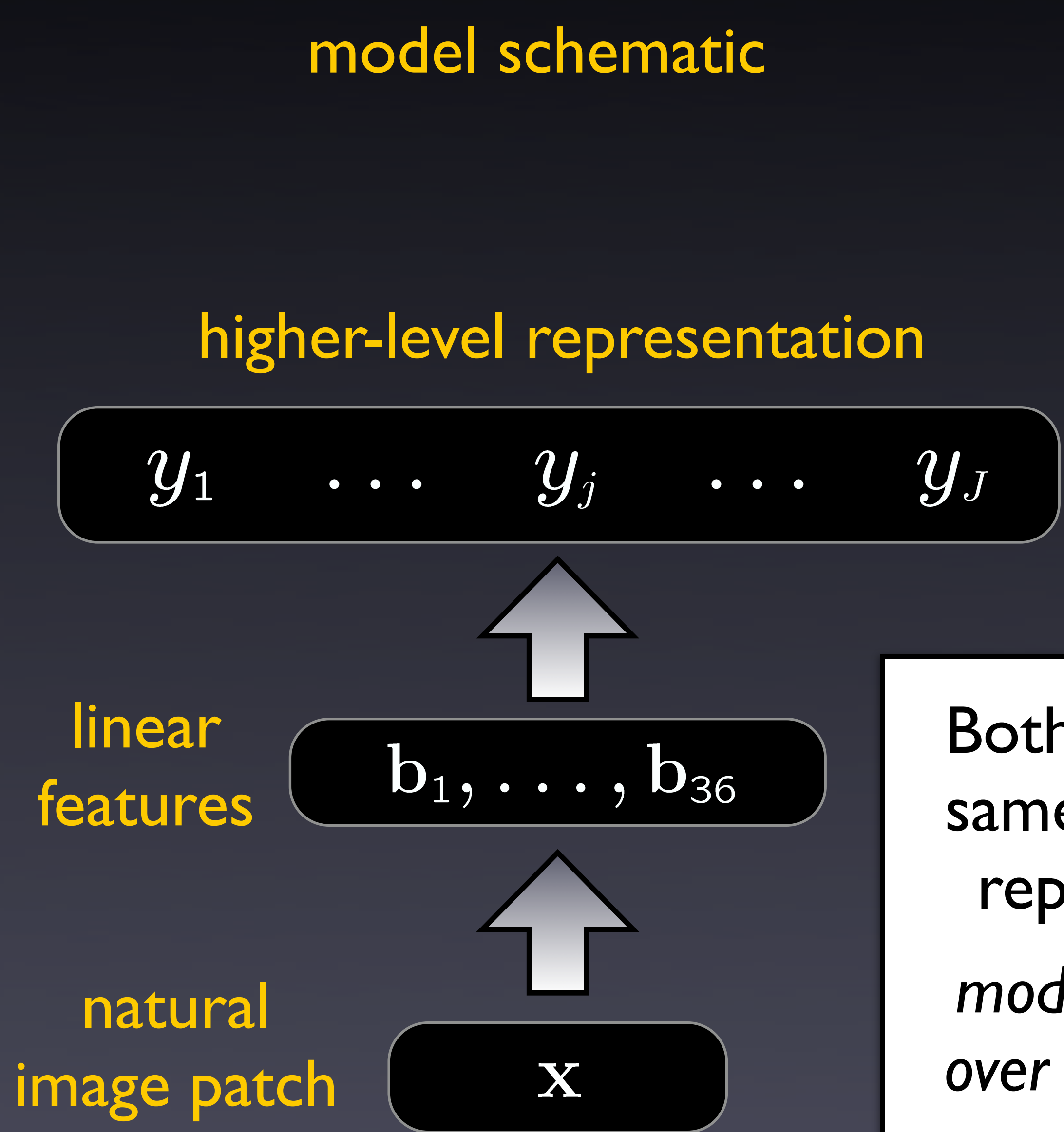
A distributed representation for covariance patterns

model schematic



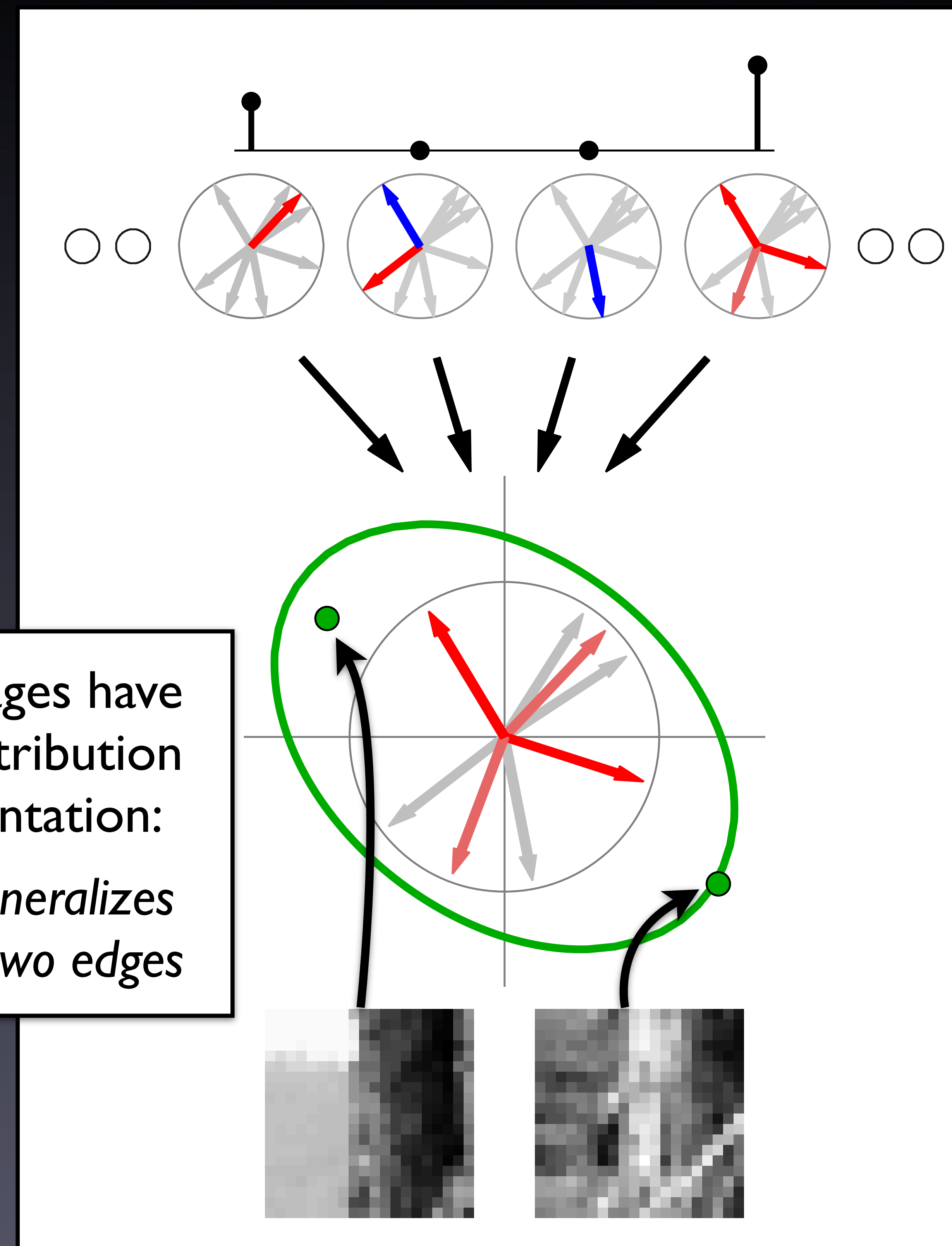


A distributed representation for covariance patterns

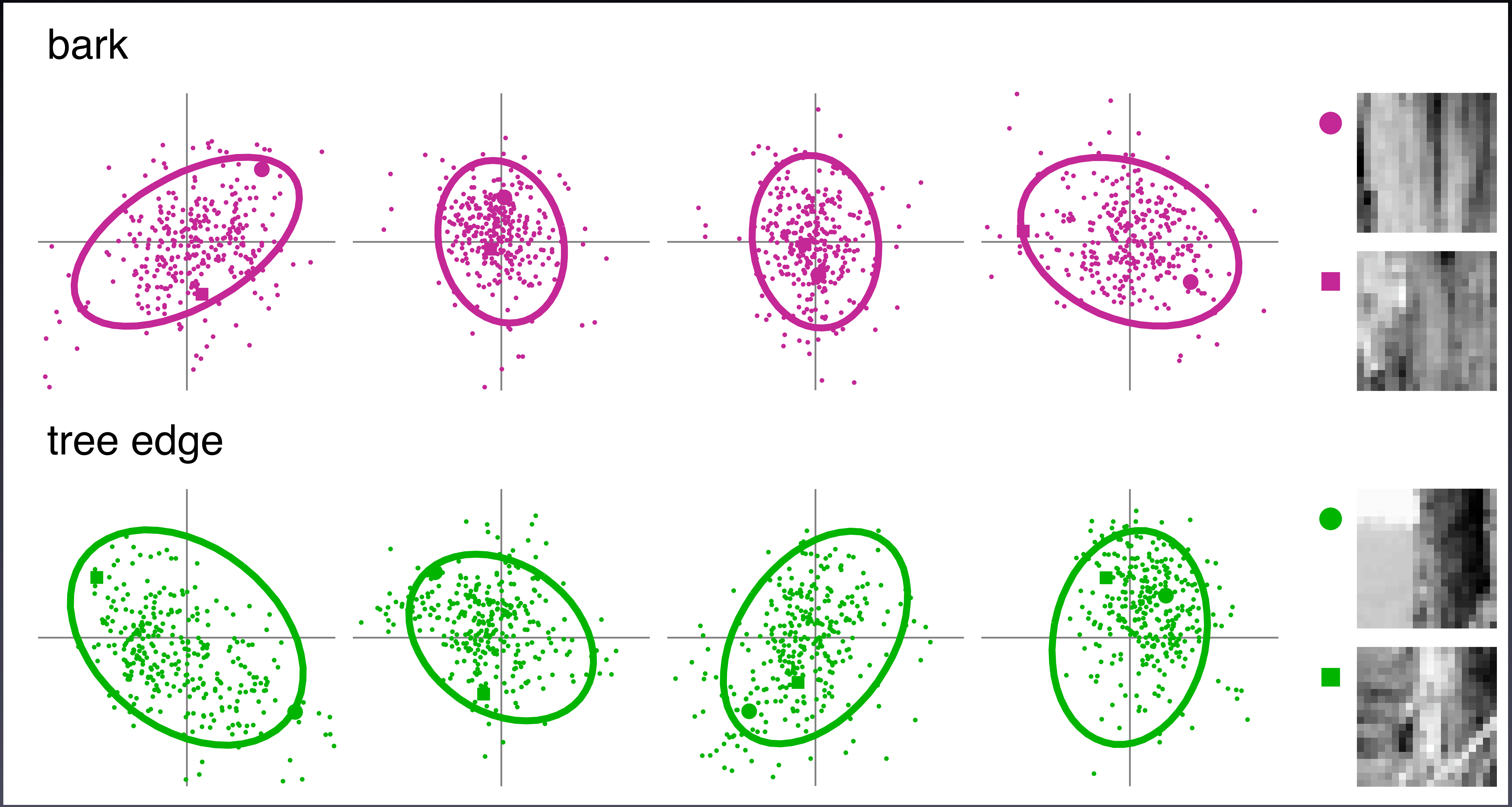


Both images have same distribution representation:

model generalizes over the two edges

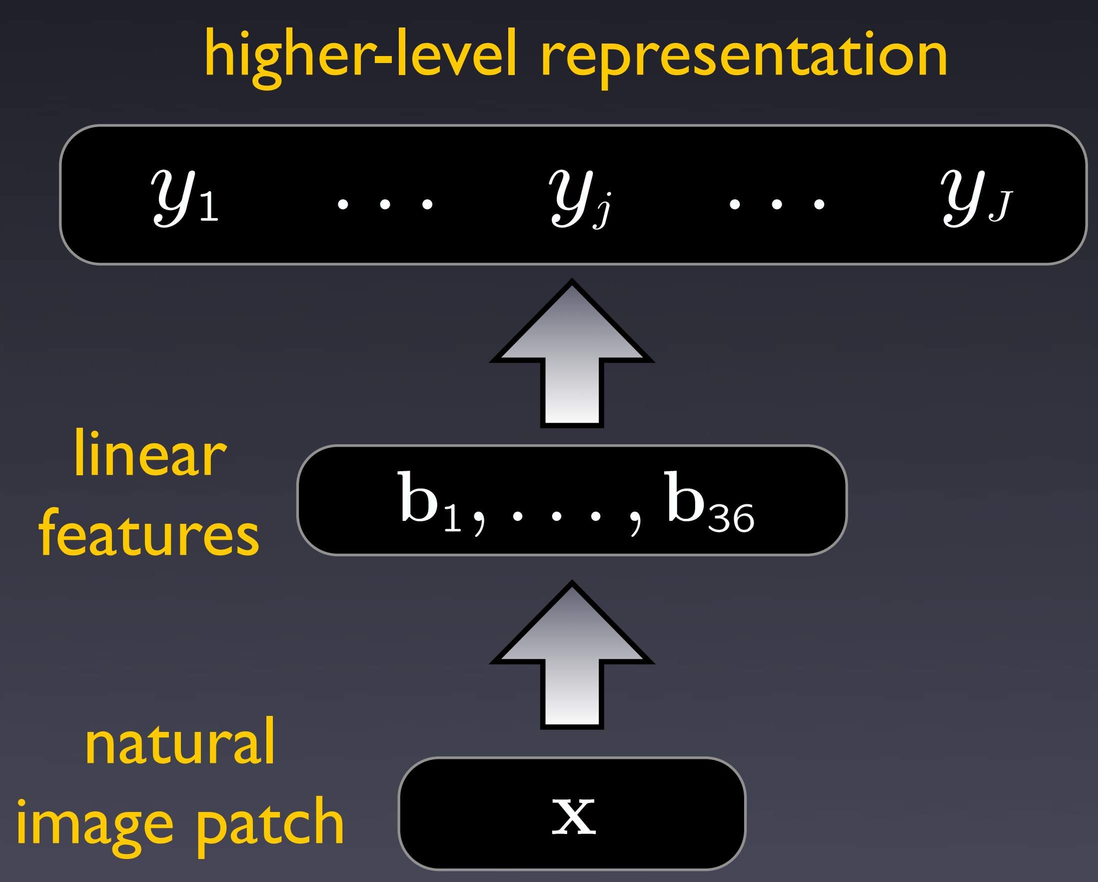


Inferring the distribution from a single image



A distributed representation for covariance patterns

model schematic



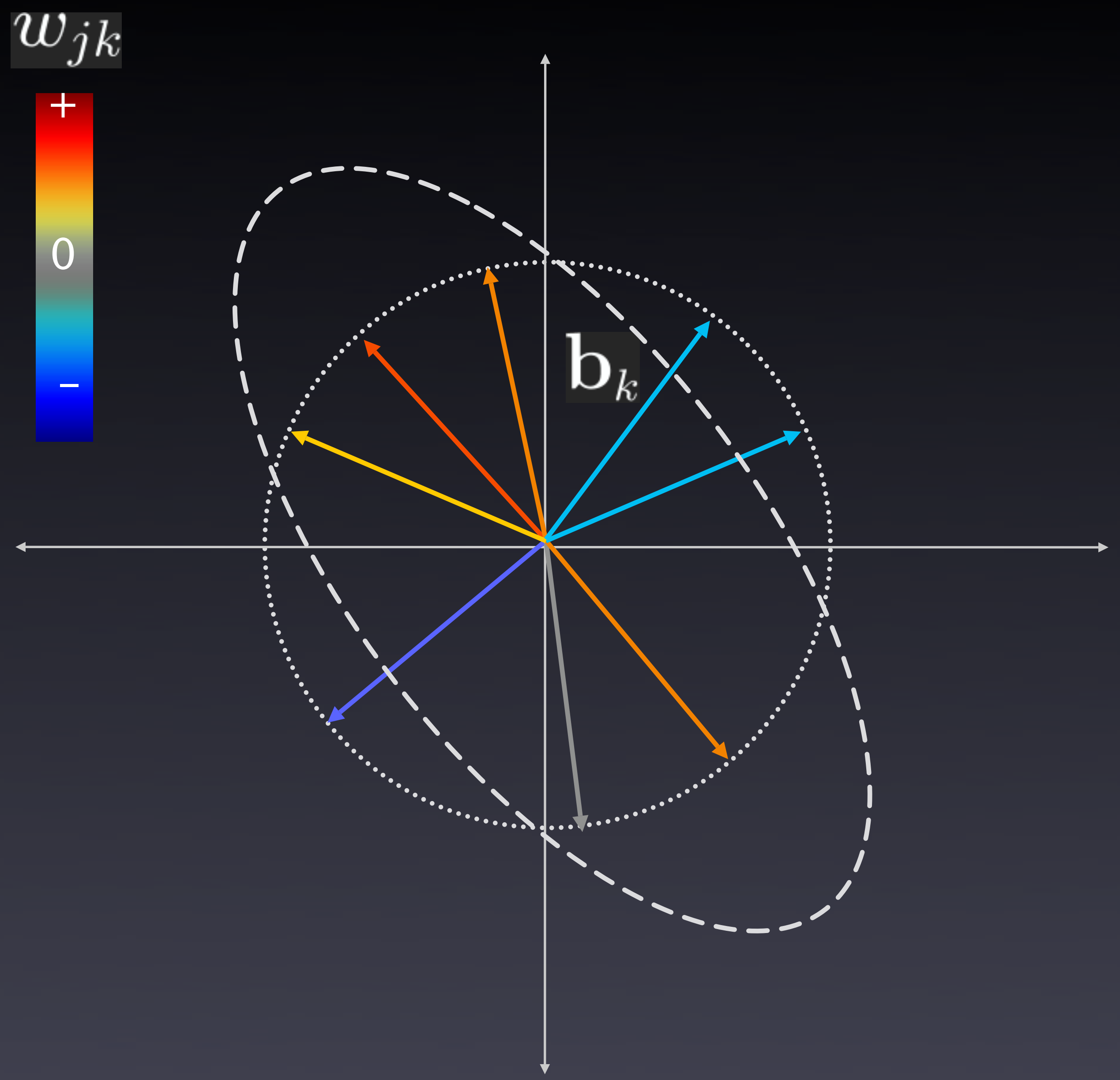
- Each pattern in top level represents a *distribution* of images
- Model
  - 20x20 image patches
  - 1000 correlation vectors (b_k's)
  - 150 high-level units (y's) each fully connected to all b_k's
- Adapt both correlation vectors and weights to natural images

$$P(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{C}_x(\mathbf{y}))$$

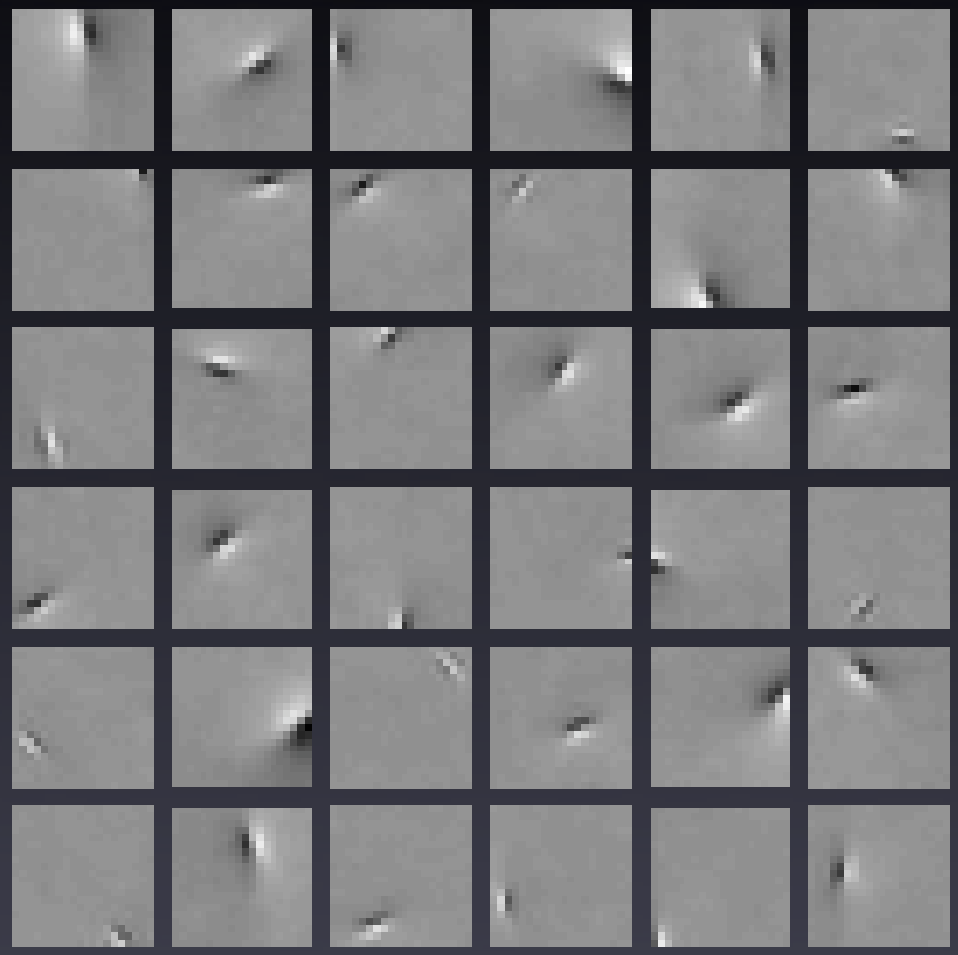
$$\log \mathbf{C}_x = \textstyle\sum_j y_j \mathbf{A}_j$$

$$\mathbf{A}_j = \textstyle\sum_k w_{jk} \mathbf{b}_k \mathbf{b}_k^T$$

Analyzing the model



36 (out of 1000) vectors b_k



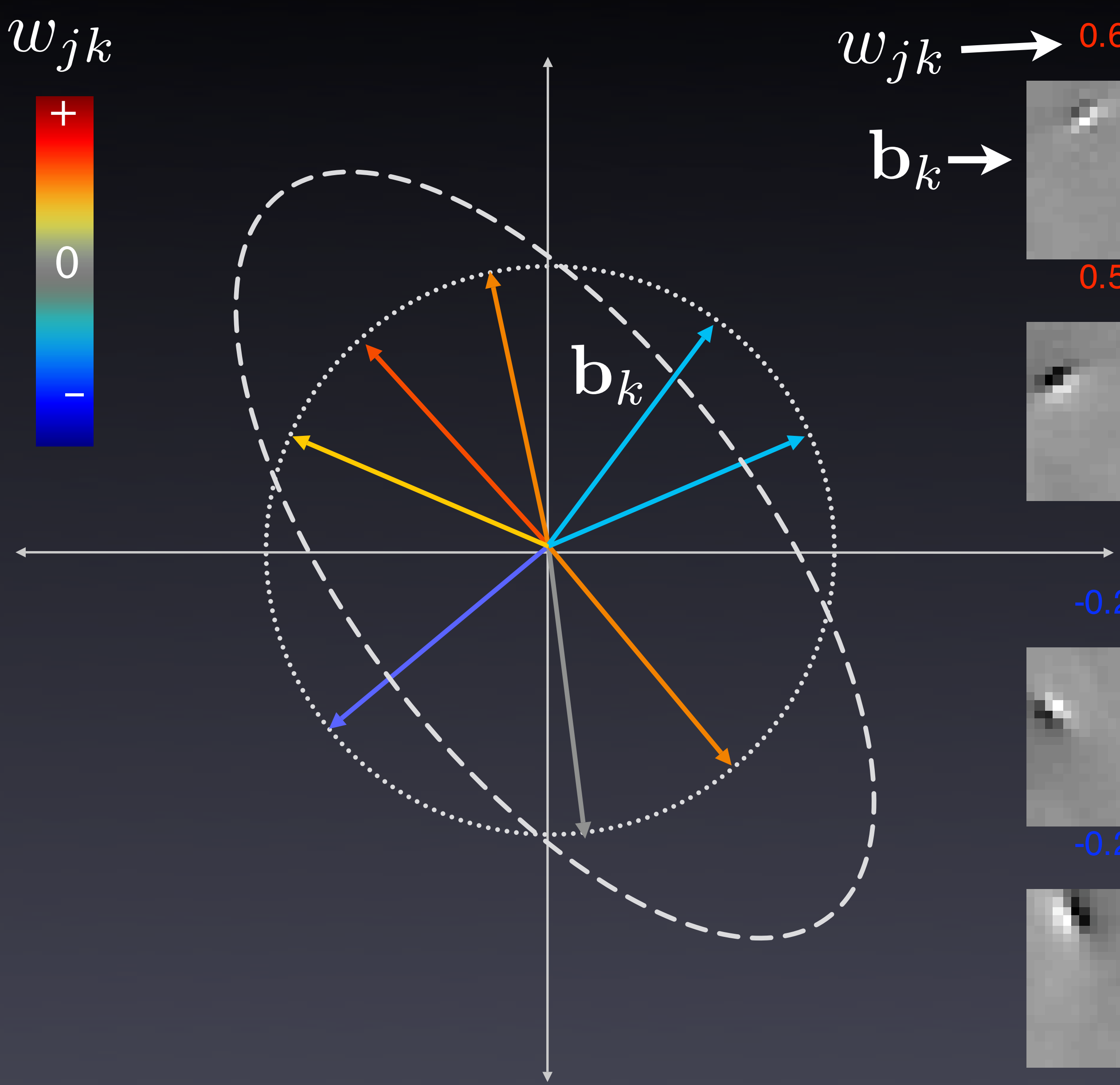
$$P(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{C}_x(\mathbf{y}))$$

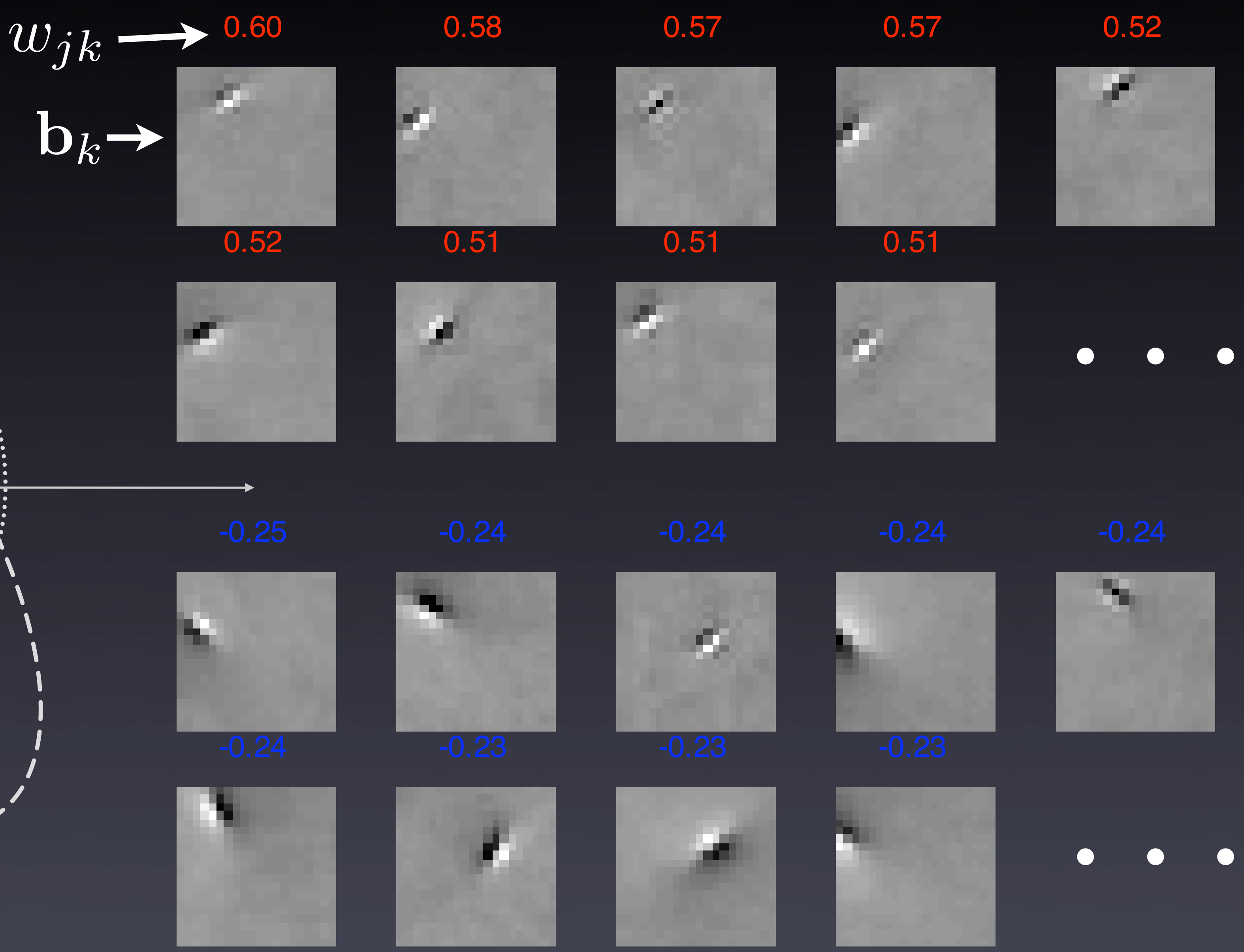
$$\log \mathbf{C}_x = \sum_j y_j \mathbf{A}_j$$

$$\mathbf{A}_j = \sum_k w_{jk} \mathbf{b}_k \mathbf{b}_k^T$$

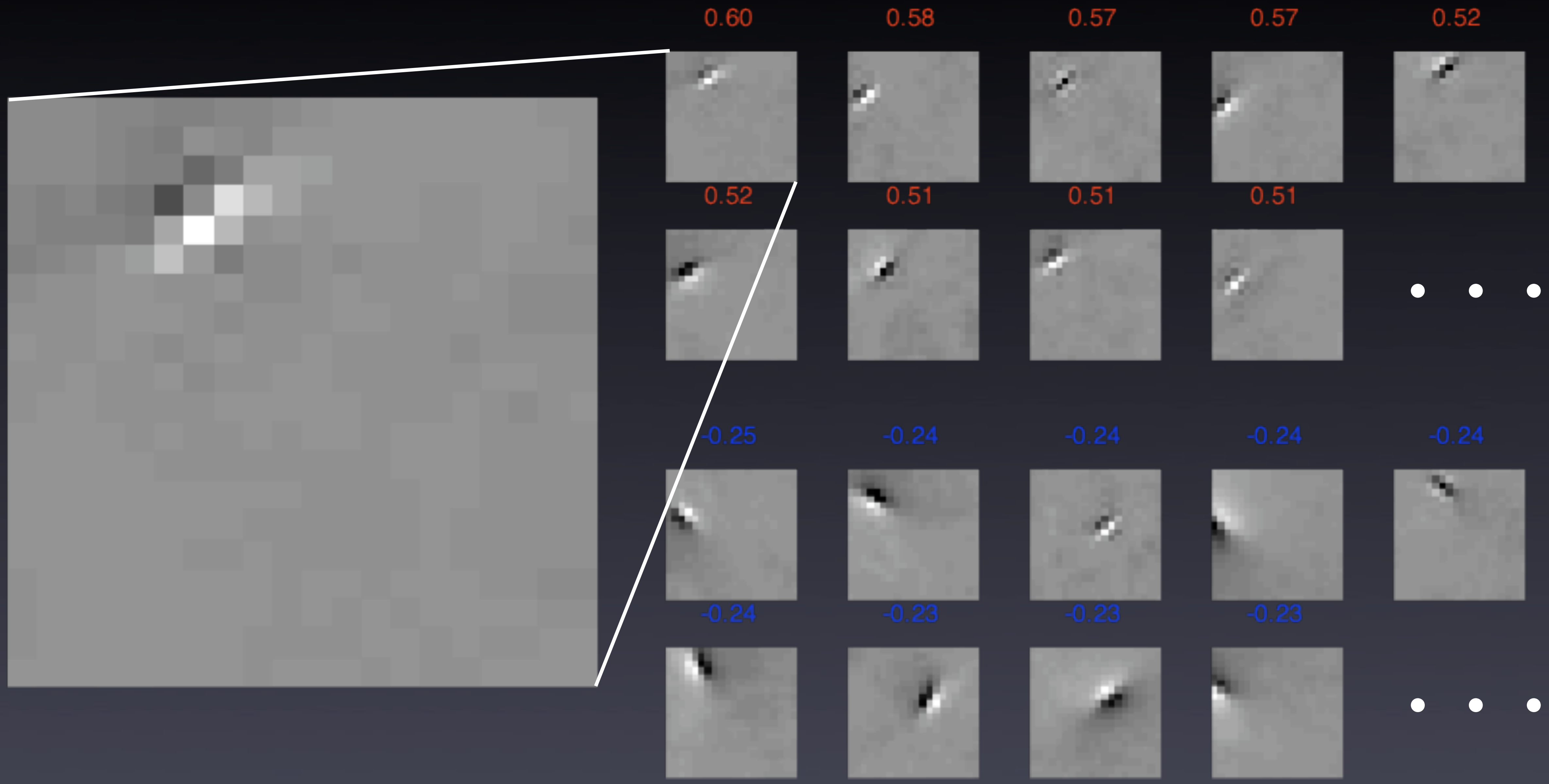
Alternative functional explanation for simple cells: *describing local distributions of images*

Analyzing a typical model neuron

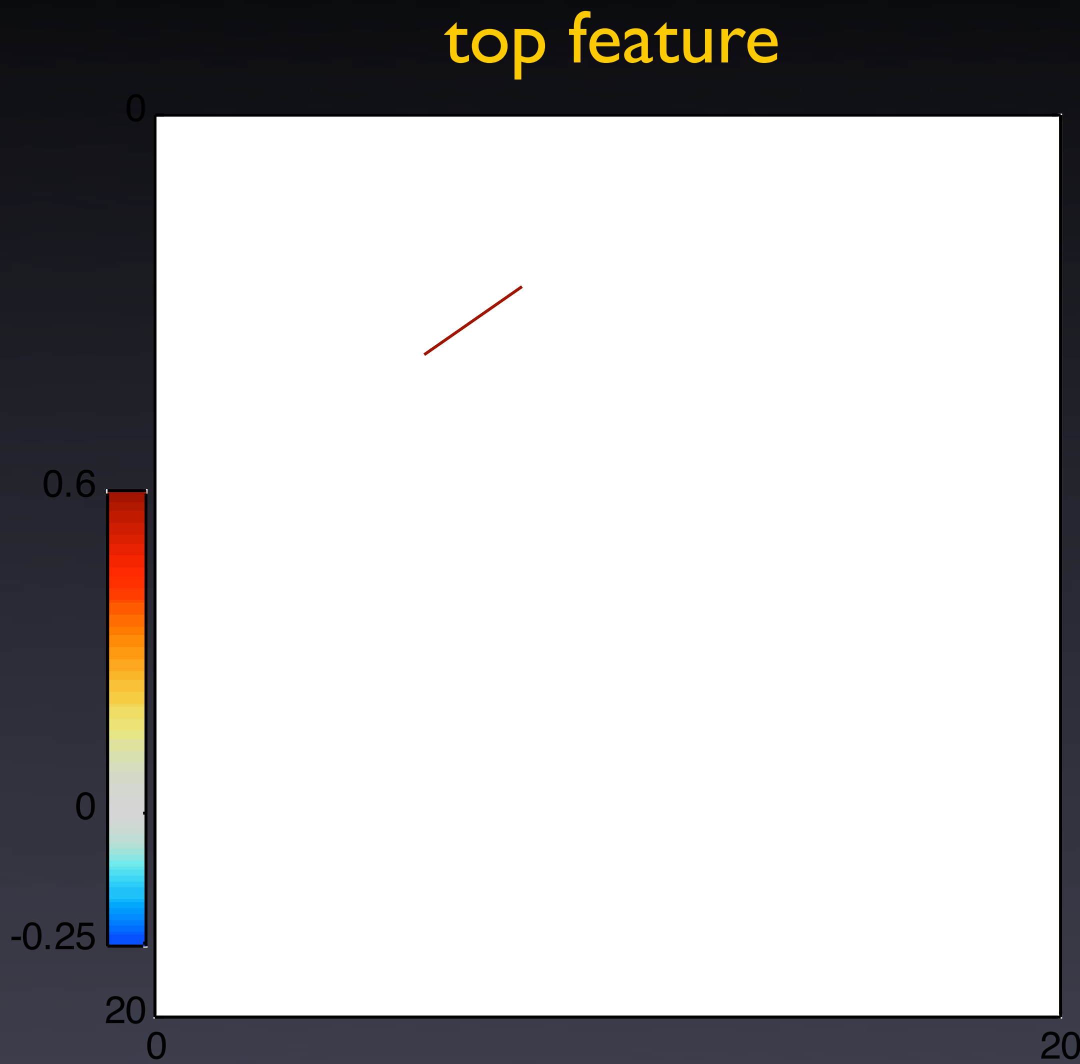


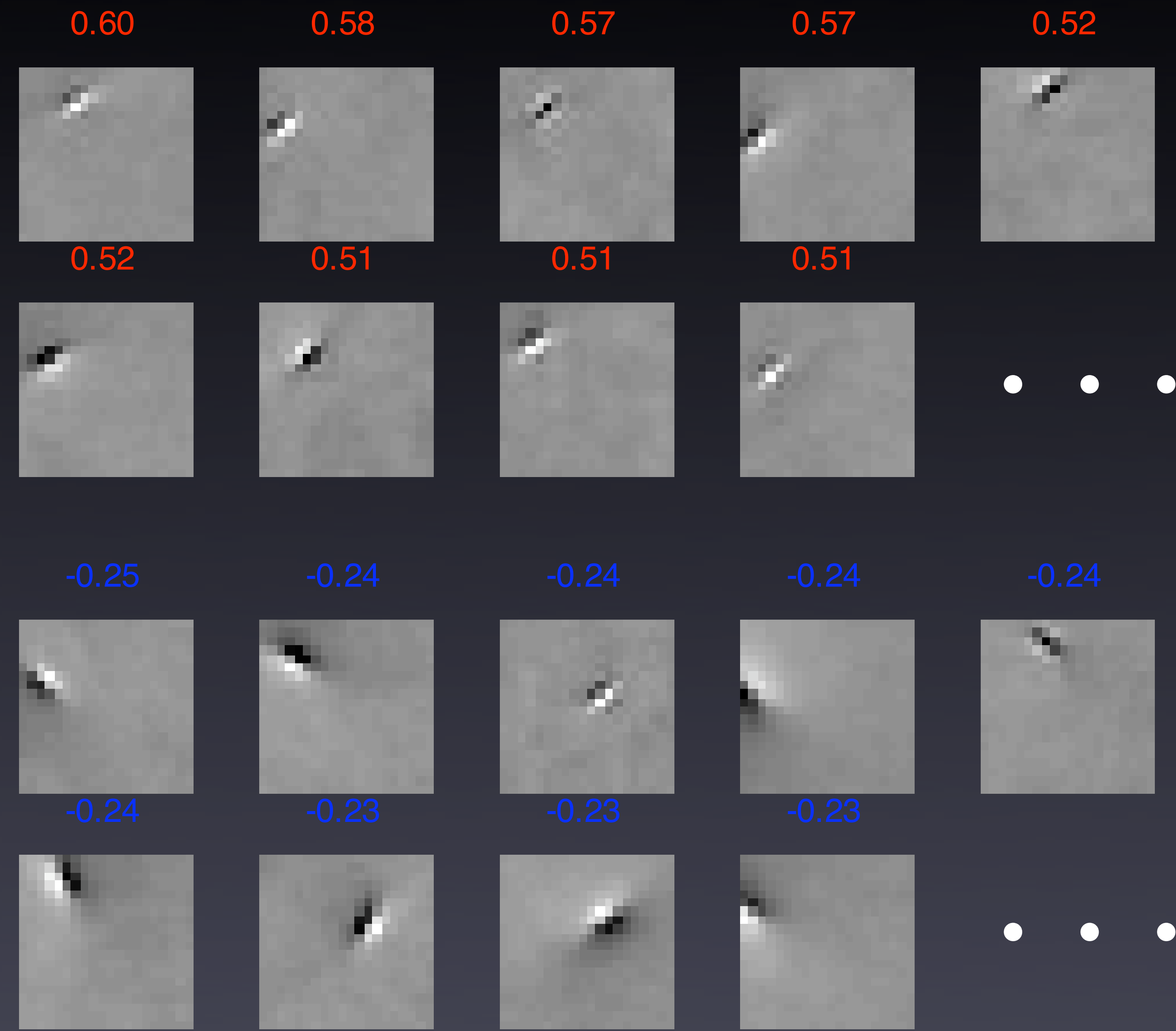


Analyzing a typical model neuron



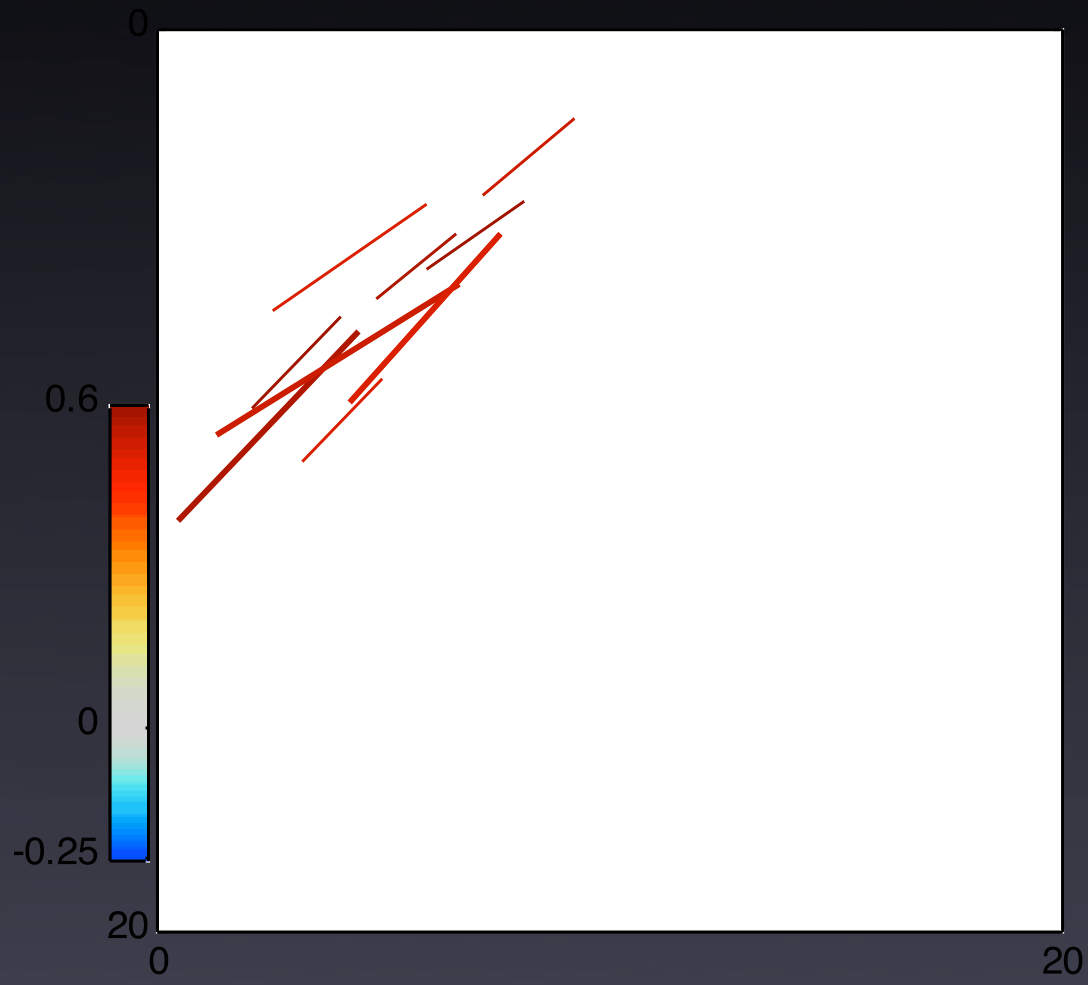
Analyzing a typical model neuron

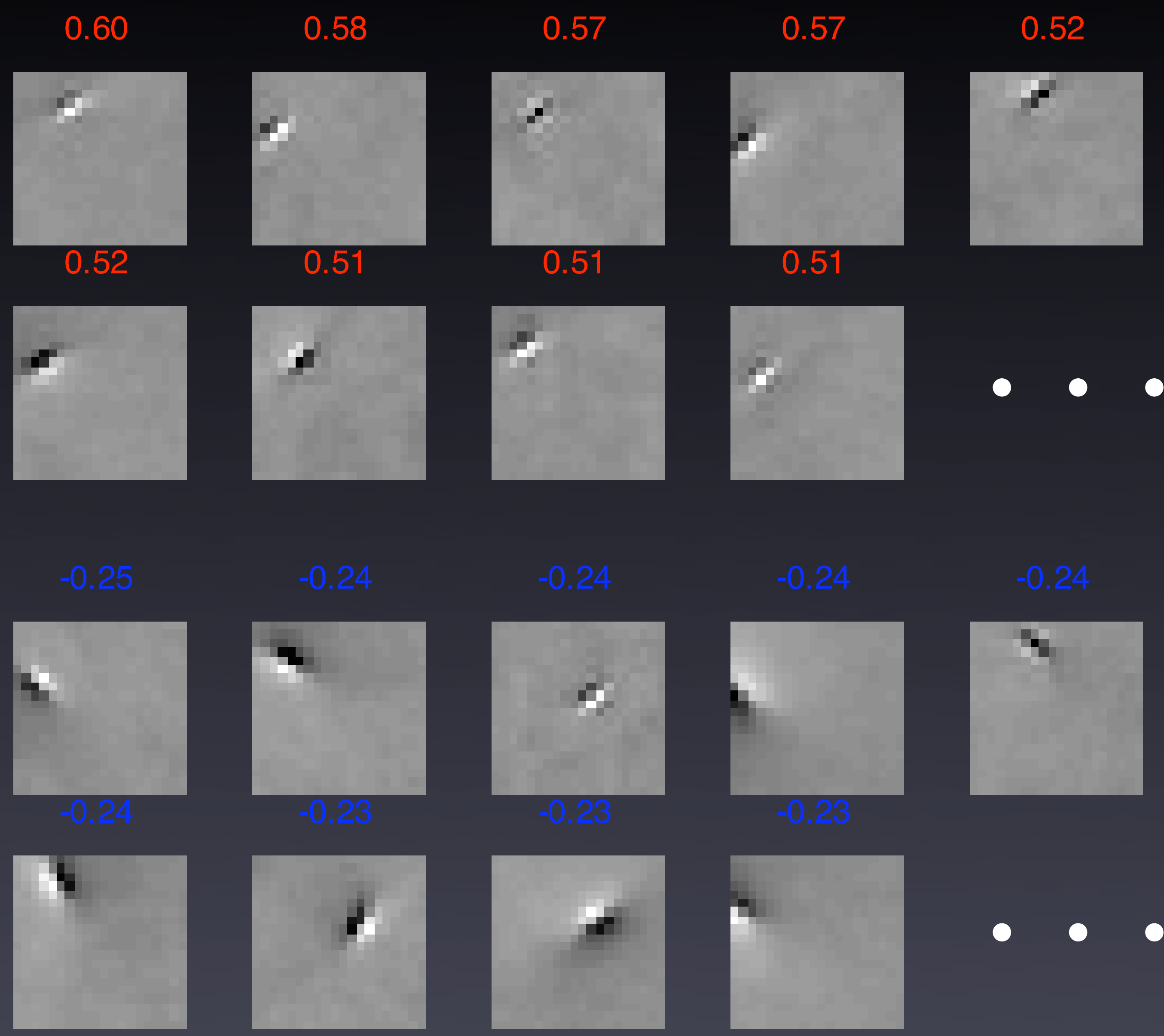




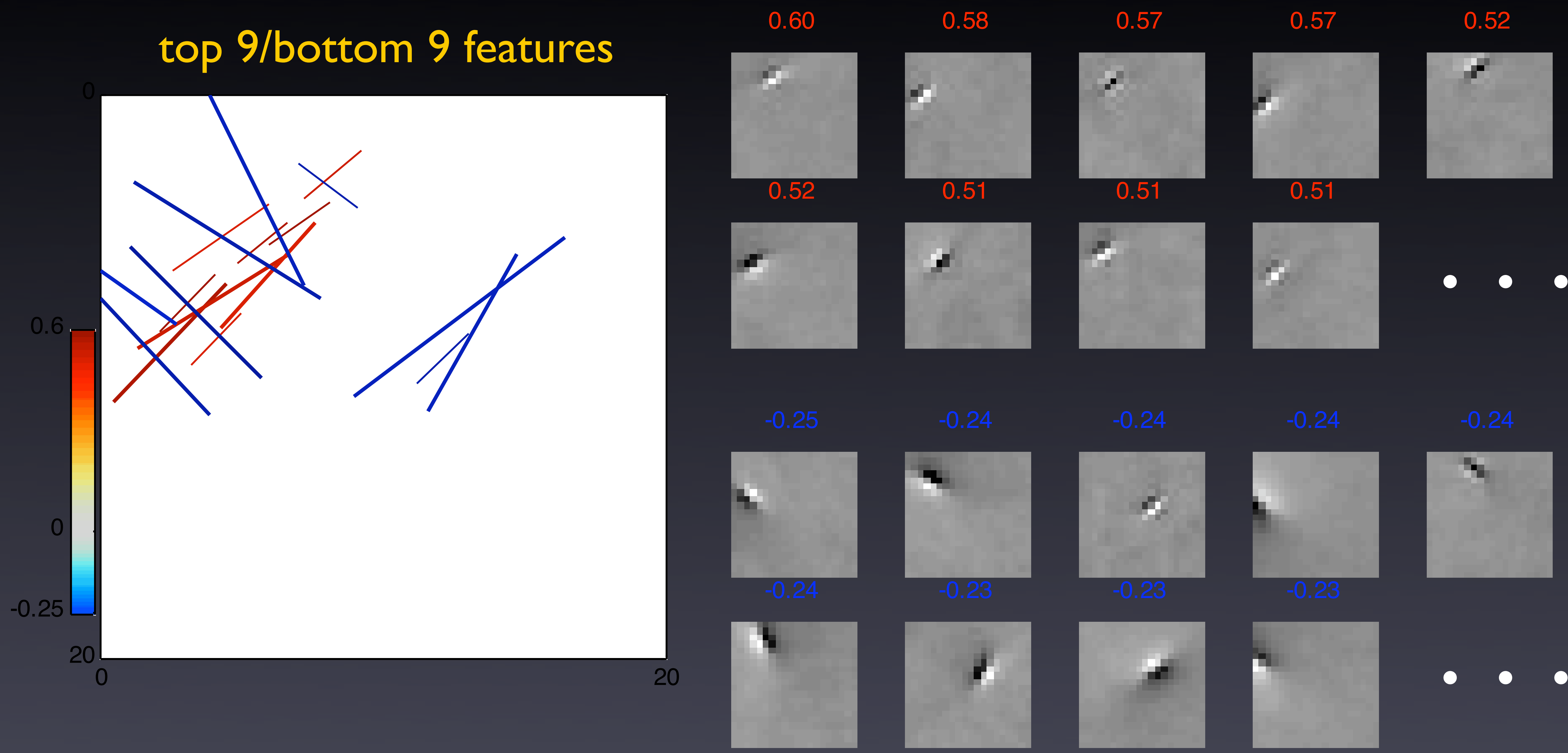
Analyzing a typical model neuron

top 9 features



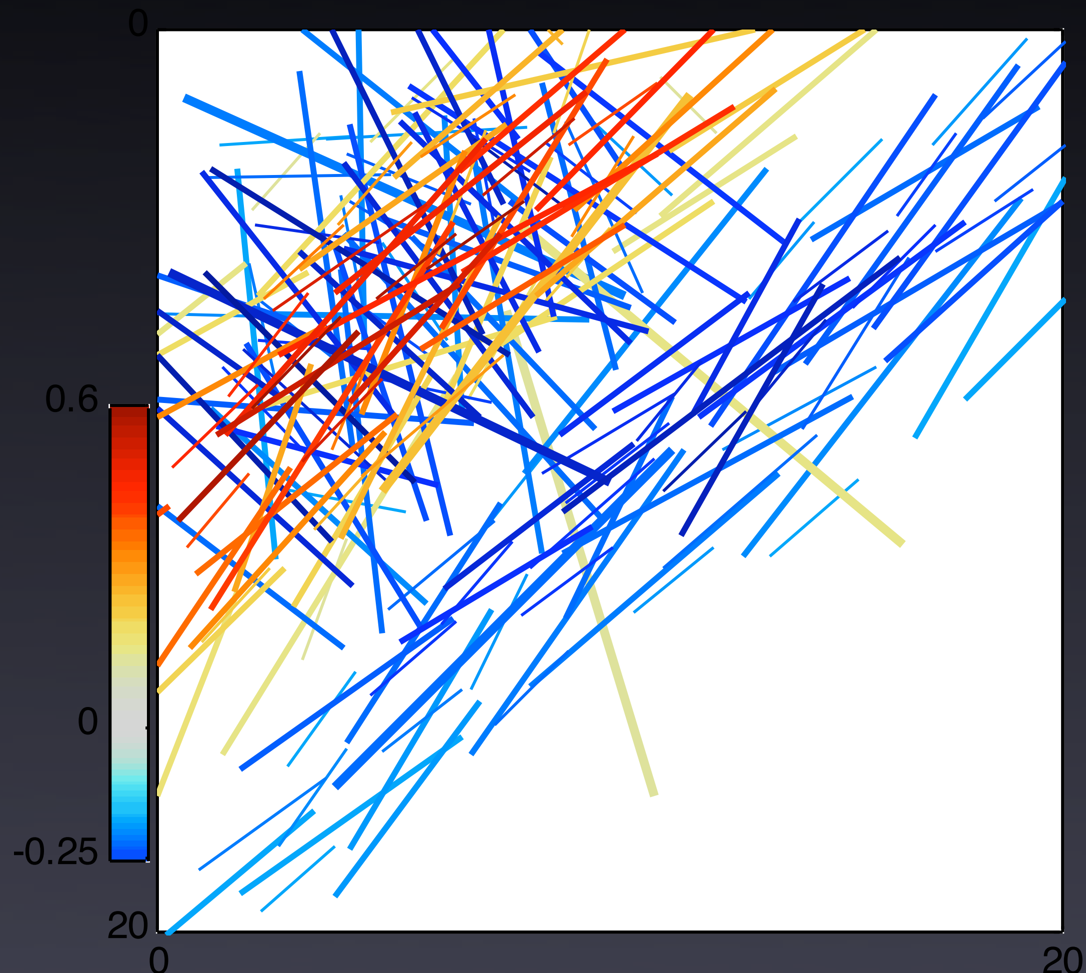


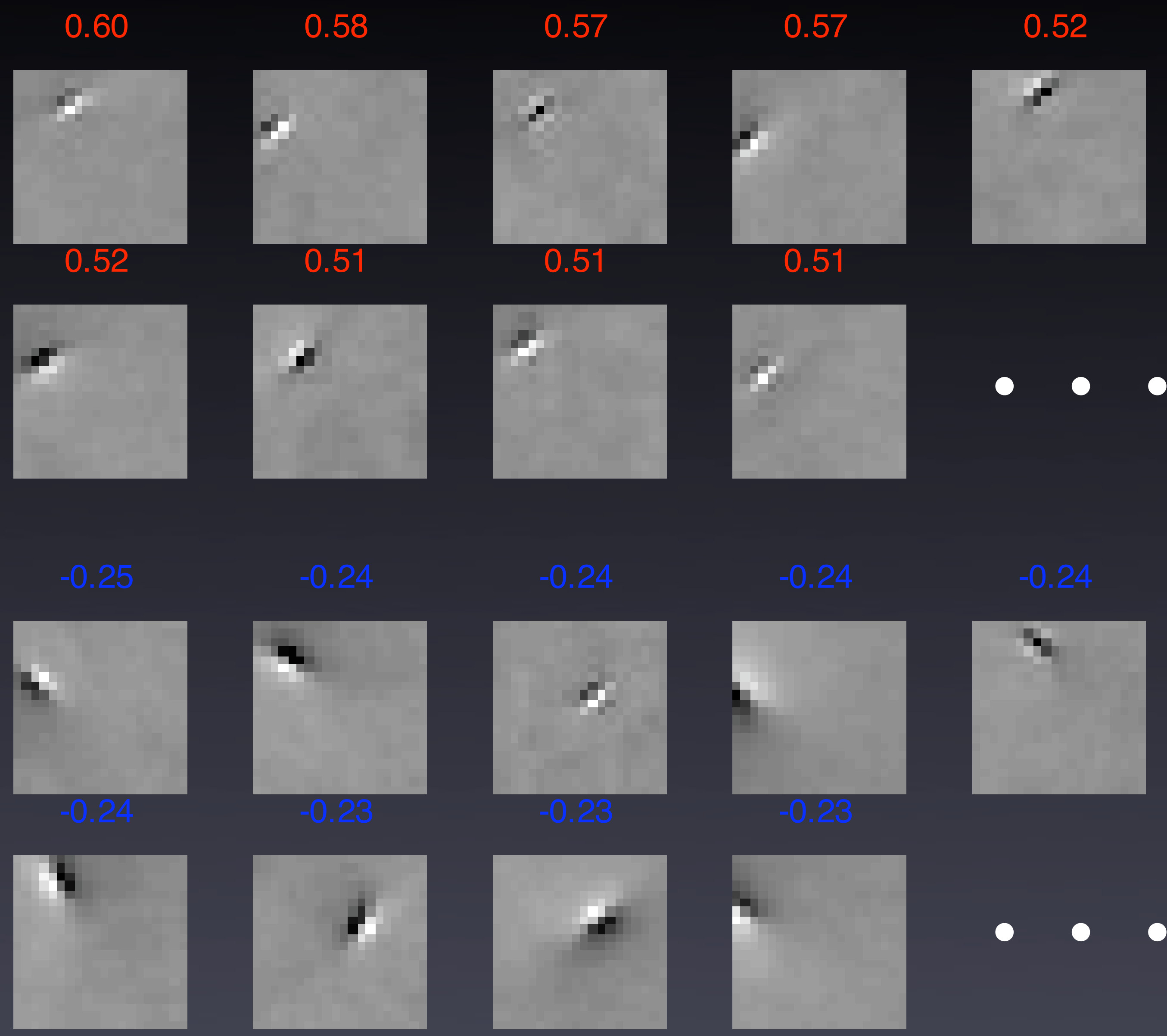
Analyzing a typical model neuron



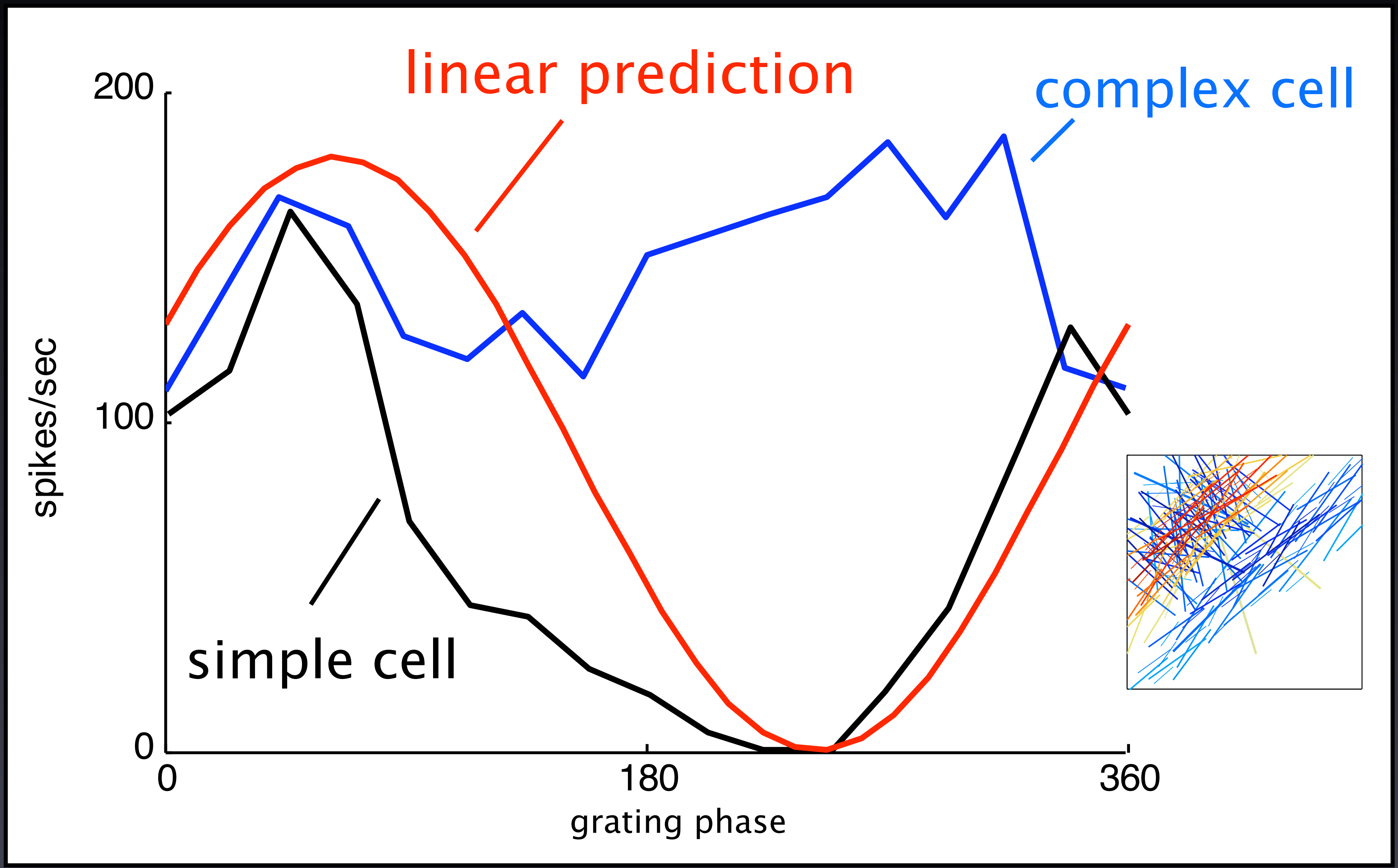
Analyzing a typical model neuron

all significant features



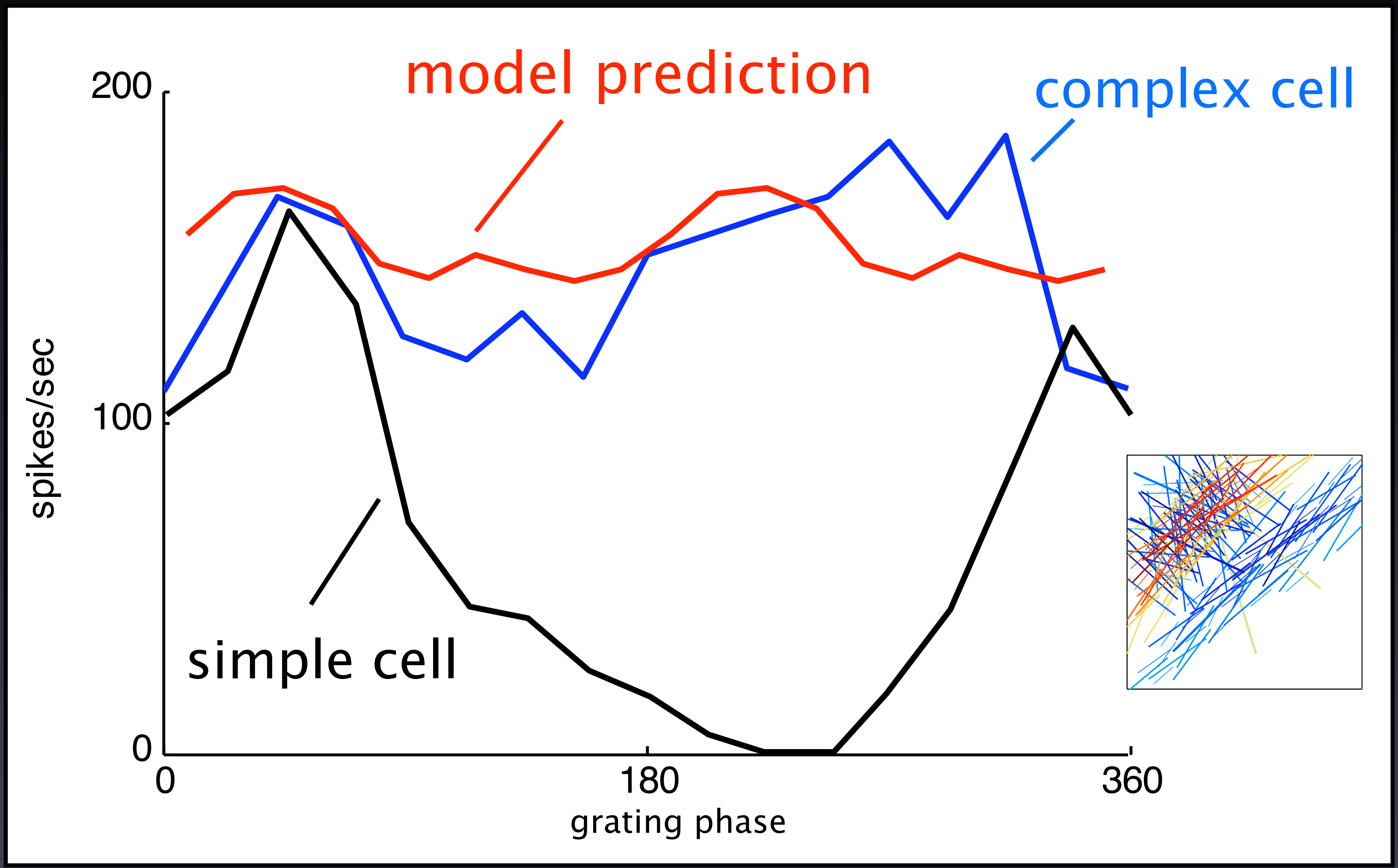


How does this cell respond?



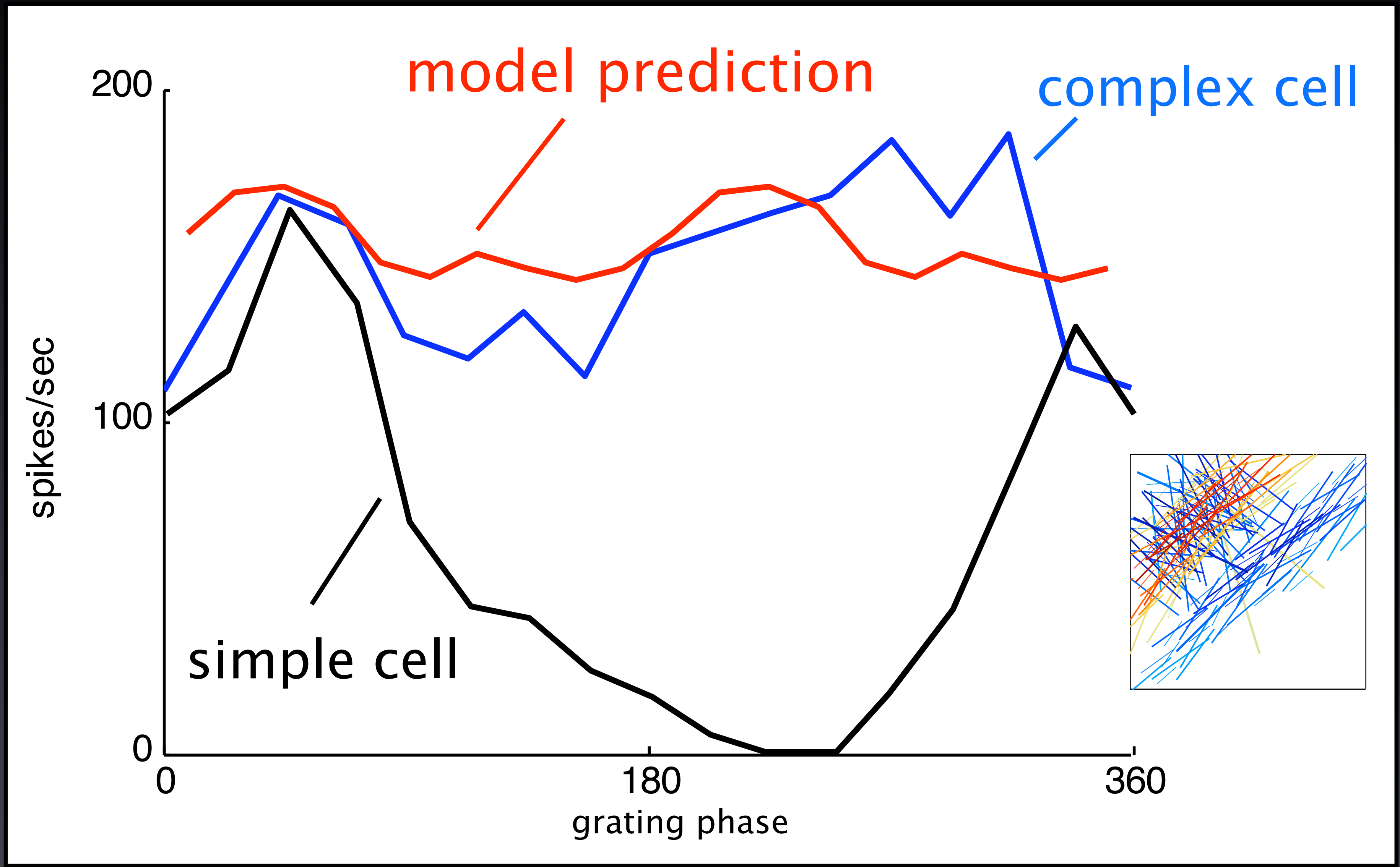
(Movshon et al, 1978)

How does this cell respond?



(Movshon et al, 1978)

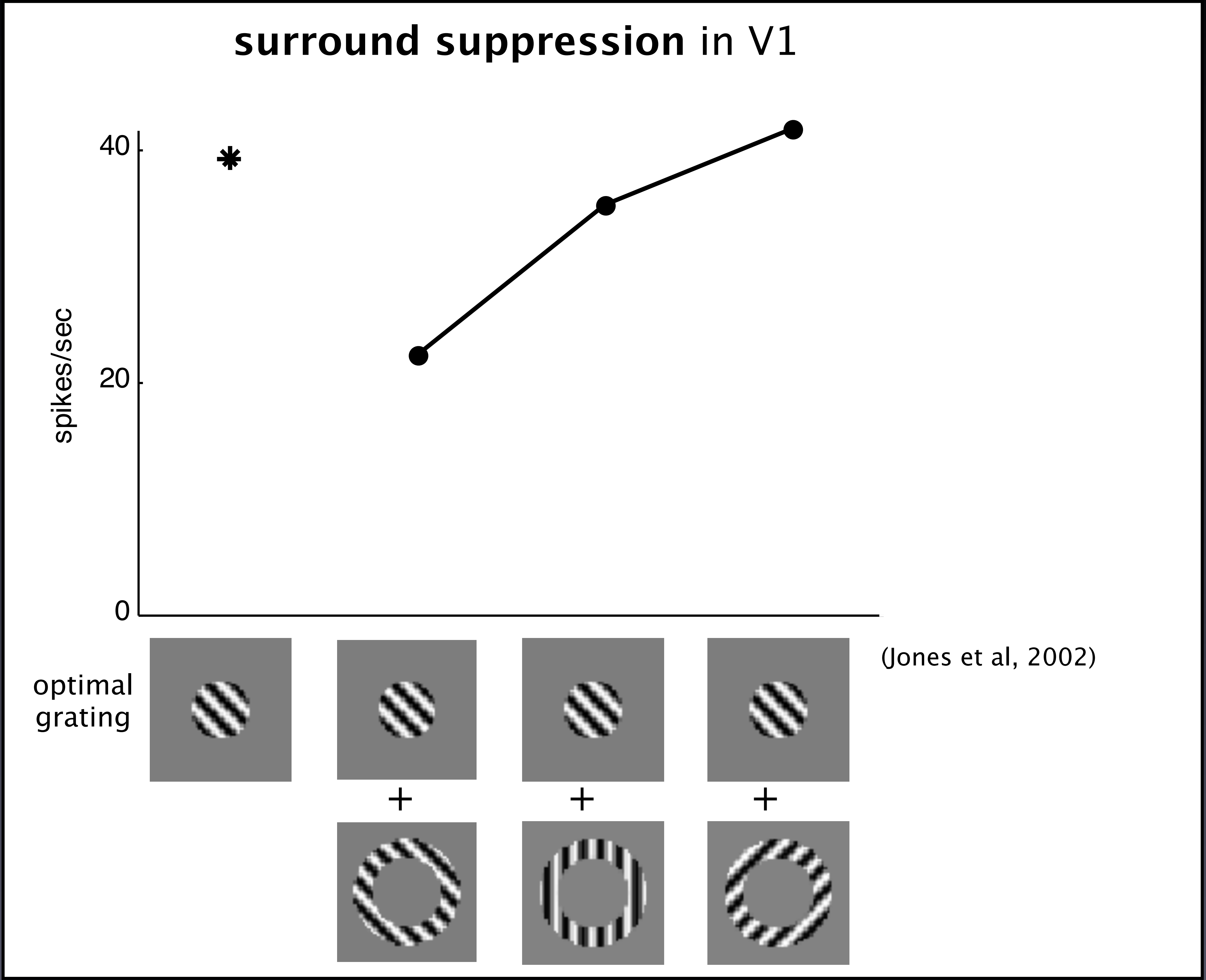
How does this cell respond?



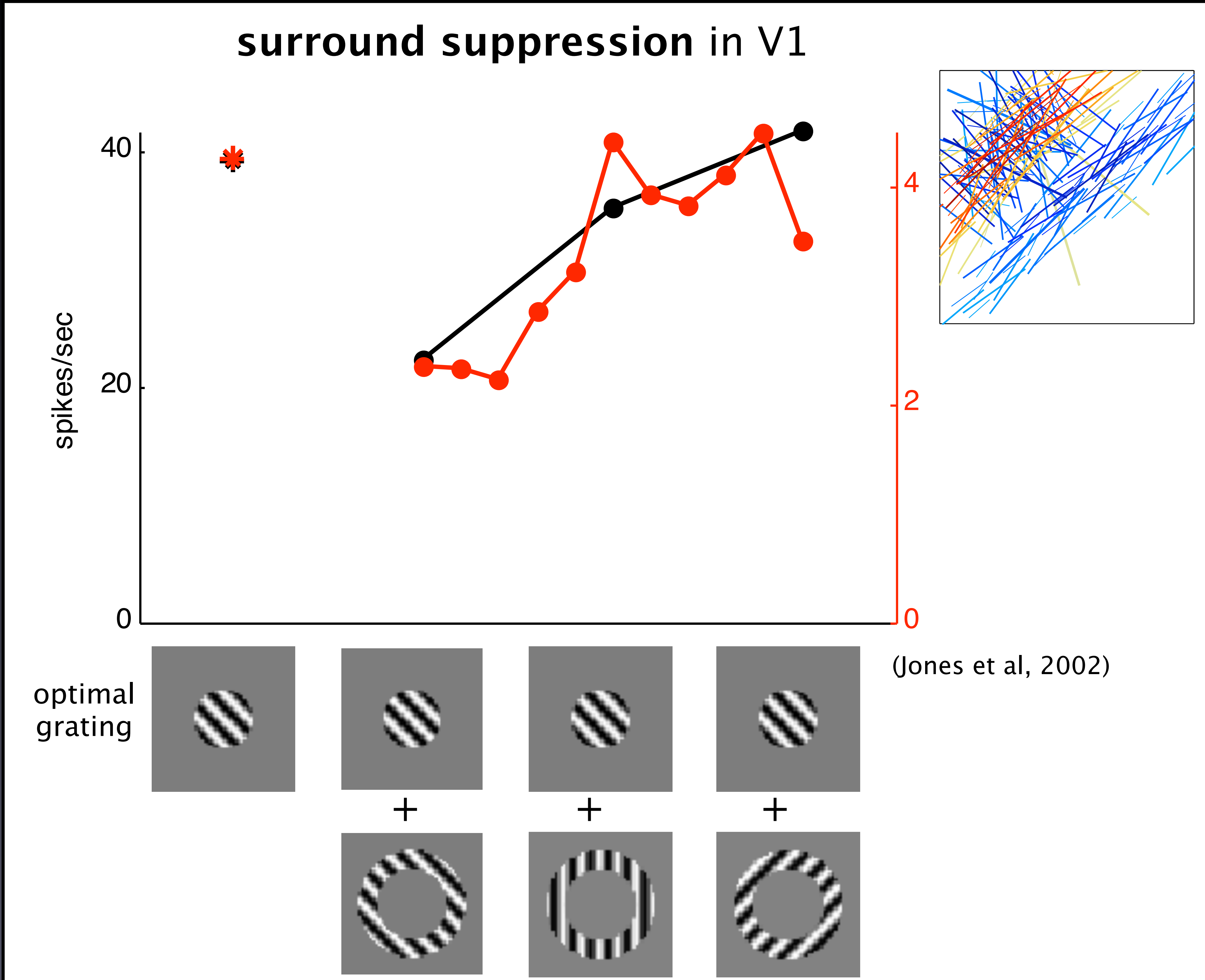
A functional role for complex cells:

generalization in natural scenes

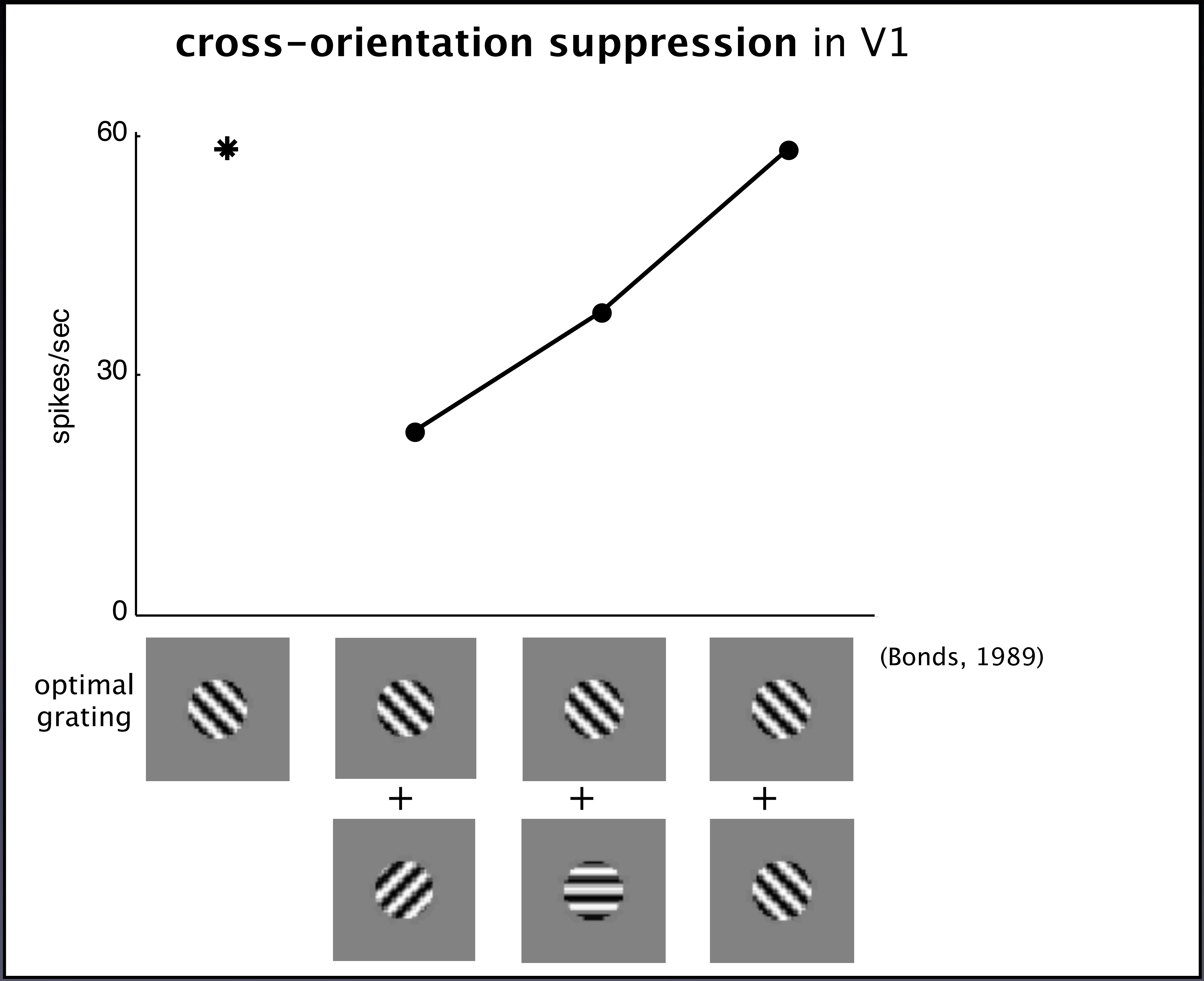
Many other properties of V1 cells are explained



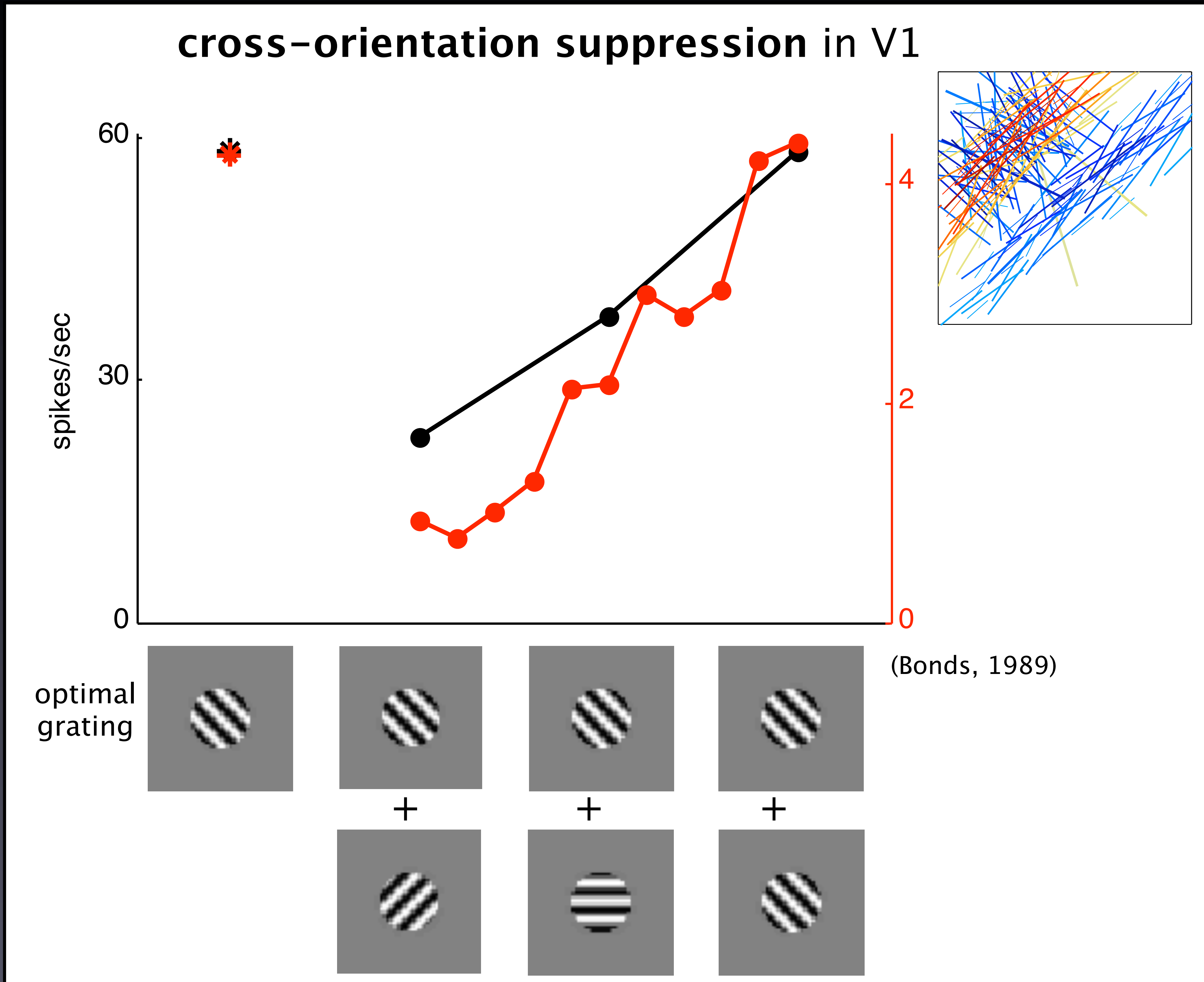
Many other properties of V1 cells are explained



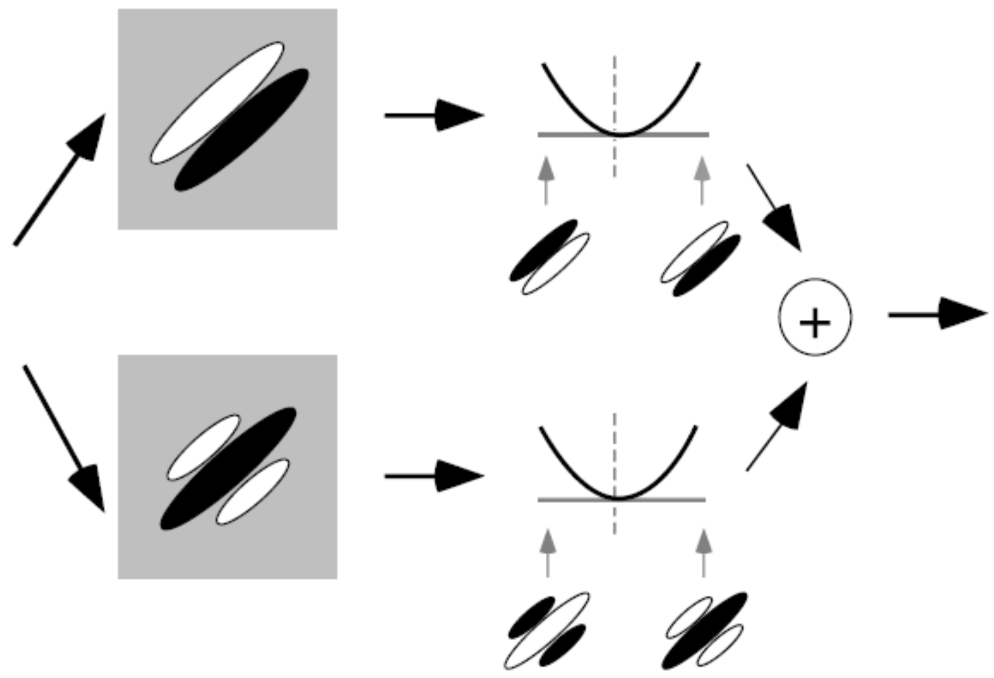
Many other properties of V1 cells are explained

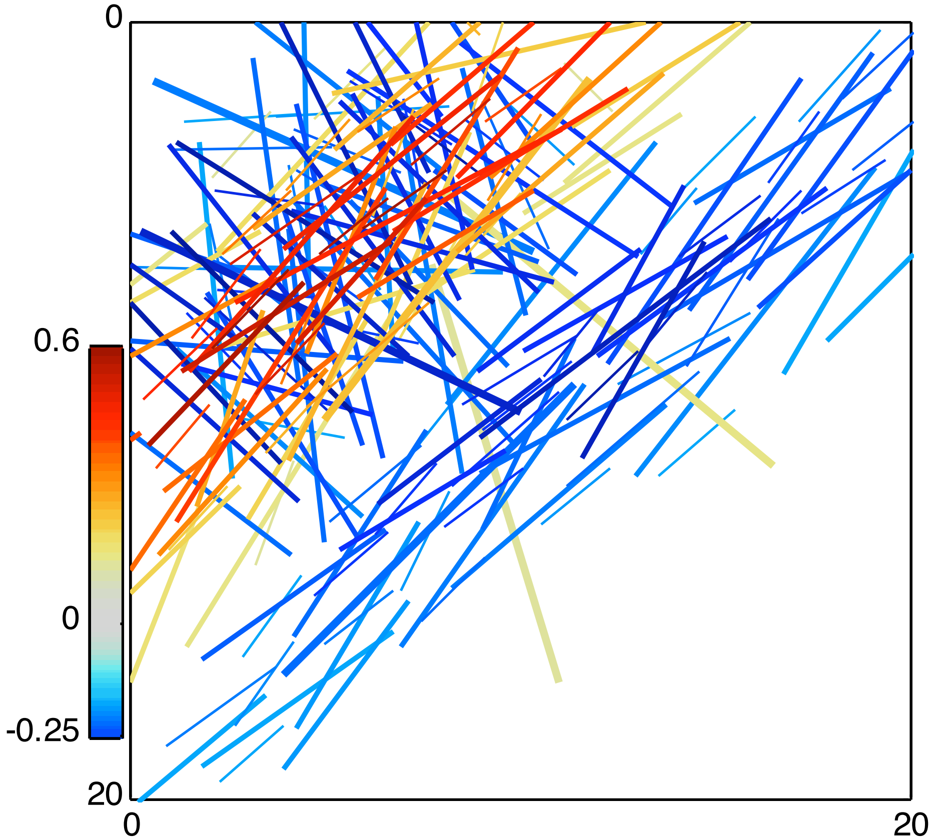


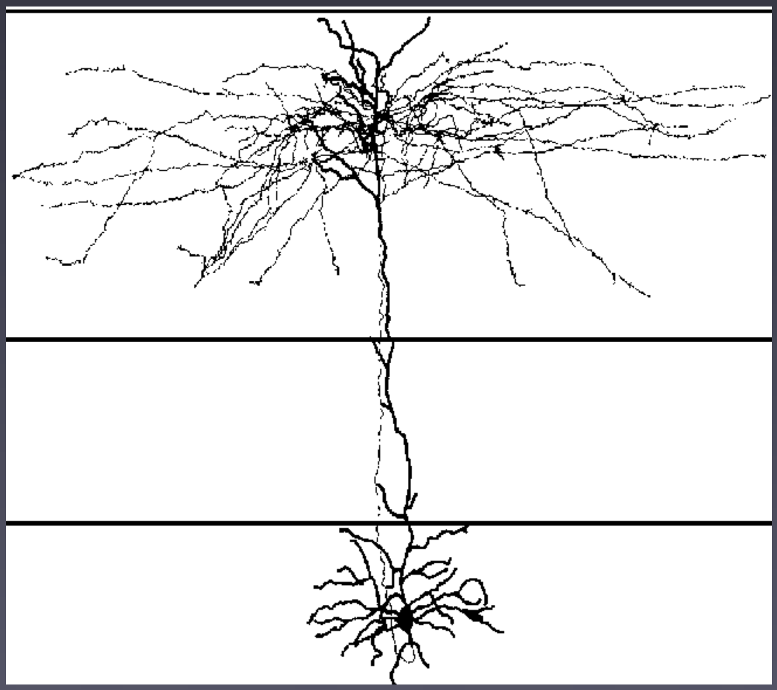
Many other properties of V1 cells are explained



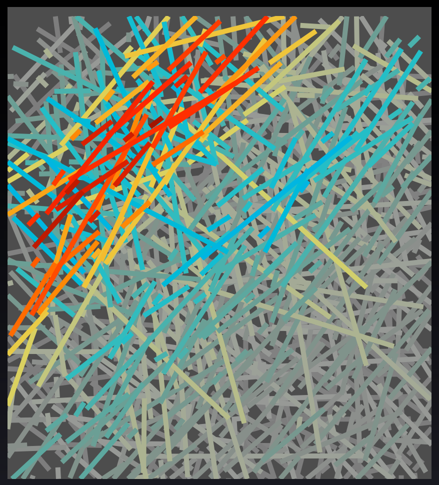
Compare to classical models of complex cells





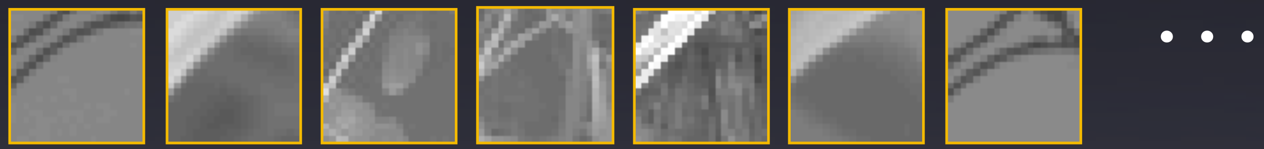


Cellular morphology of a V1 cell

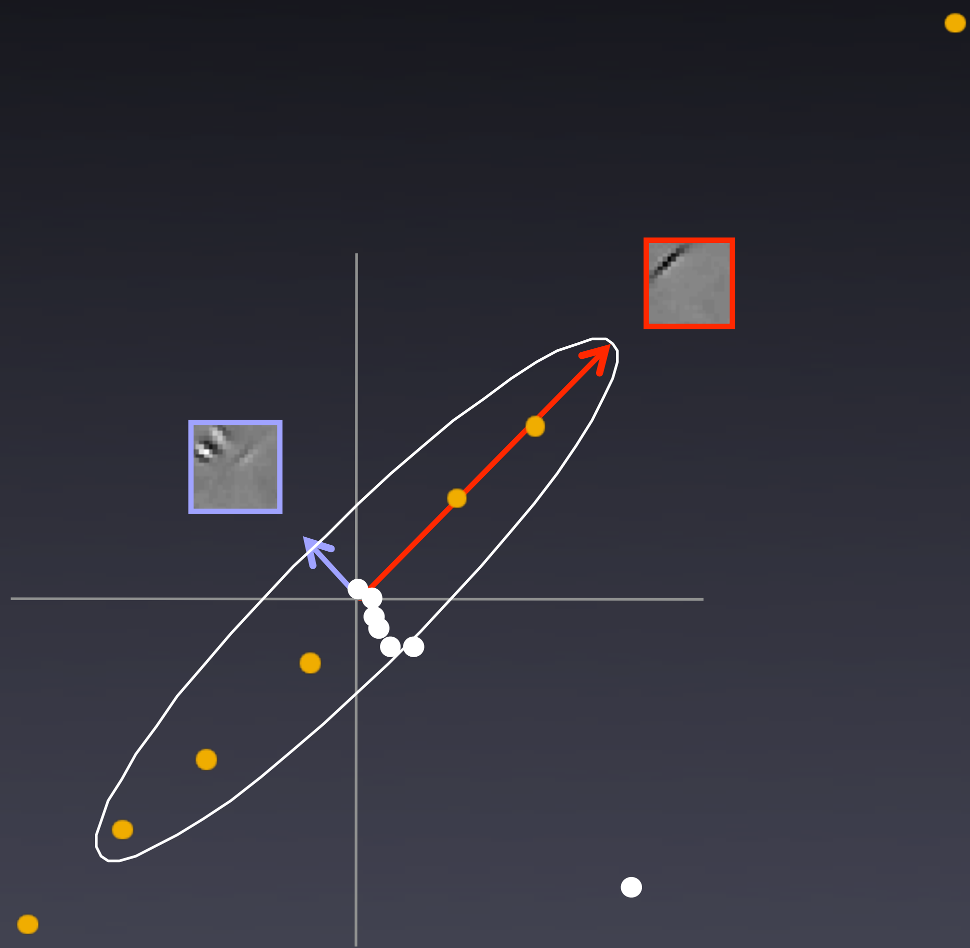


What kind of images activate this unit?

most activating stimuli

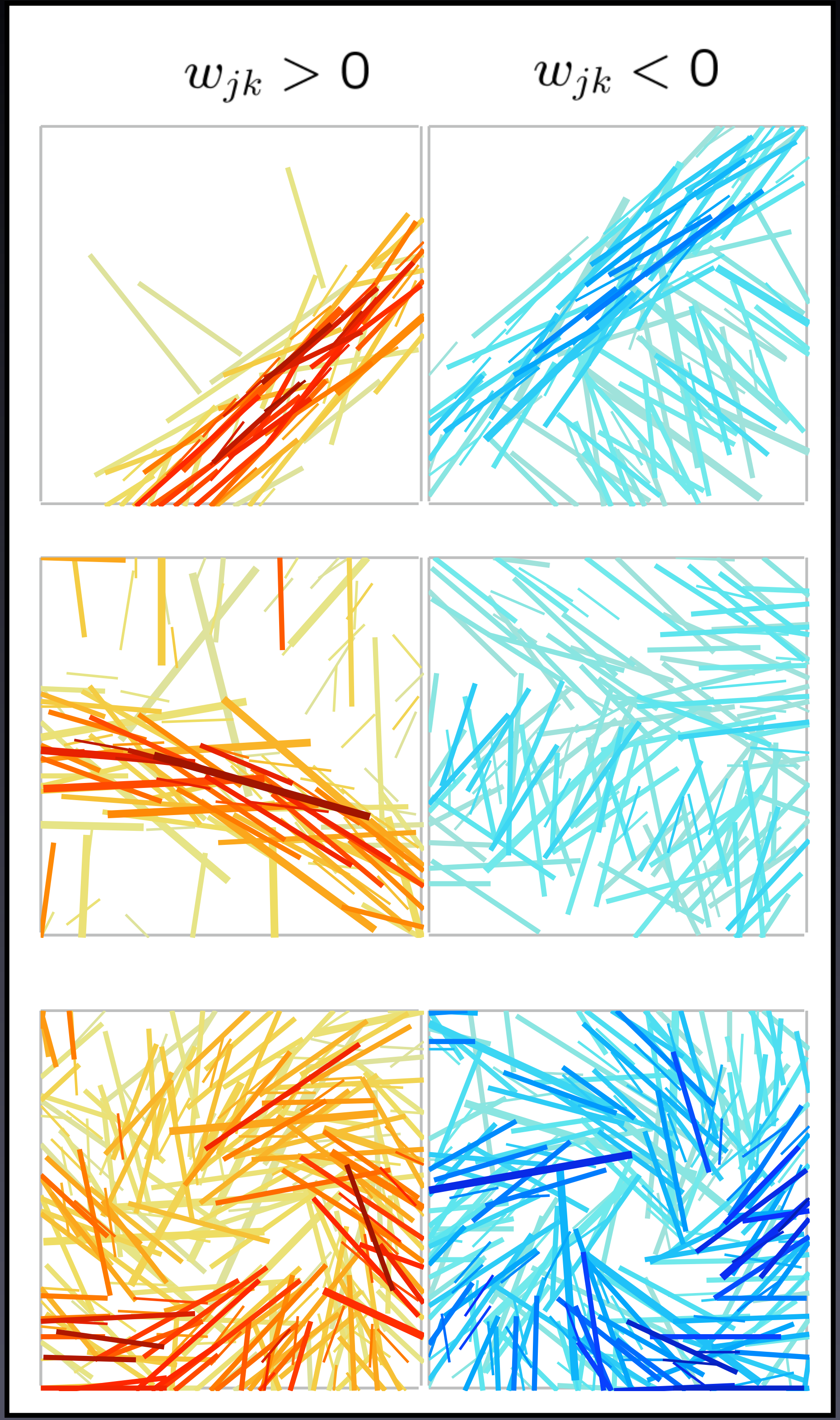


most suppressive stimuli



Main point: unit encodes *distributions* of images

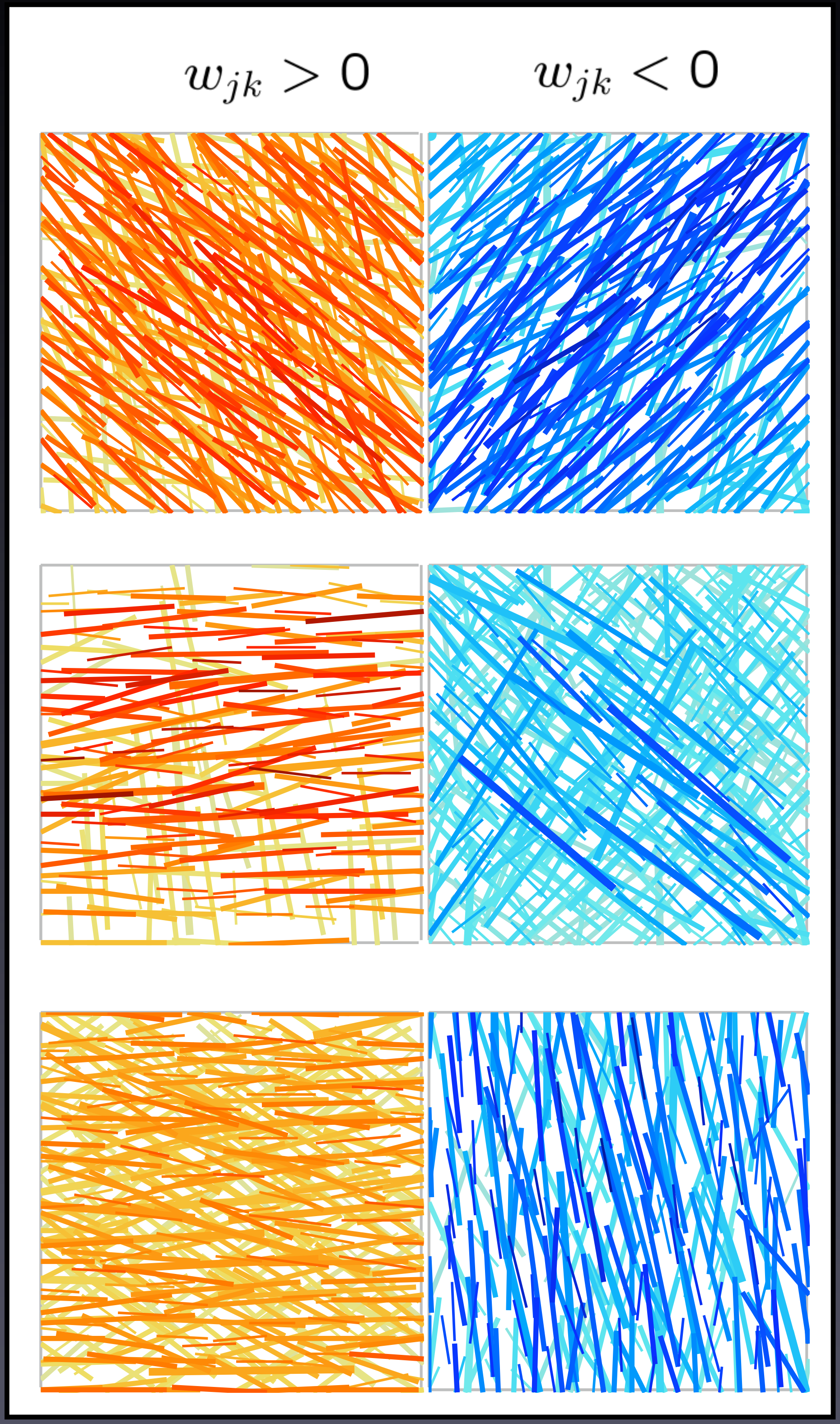
Other orientation-tuned model neurons



relationship to neurons in V1/V2/...

- complex cells?
- shape/curvature-encoding neurons?
- non-Cartesian pattern cells?

Some encode global frequency/orientation

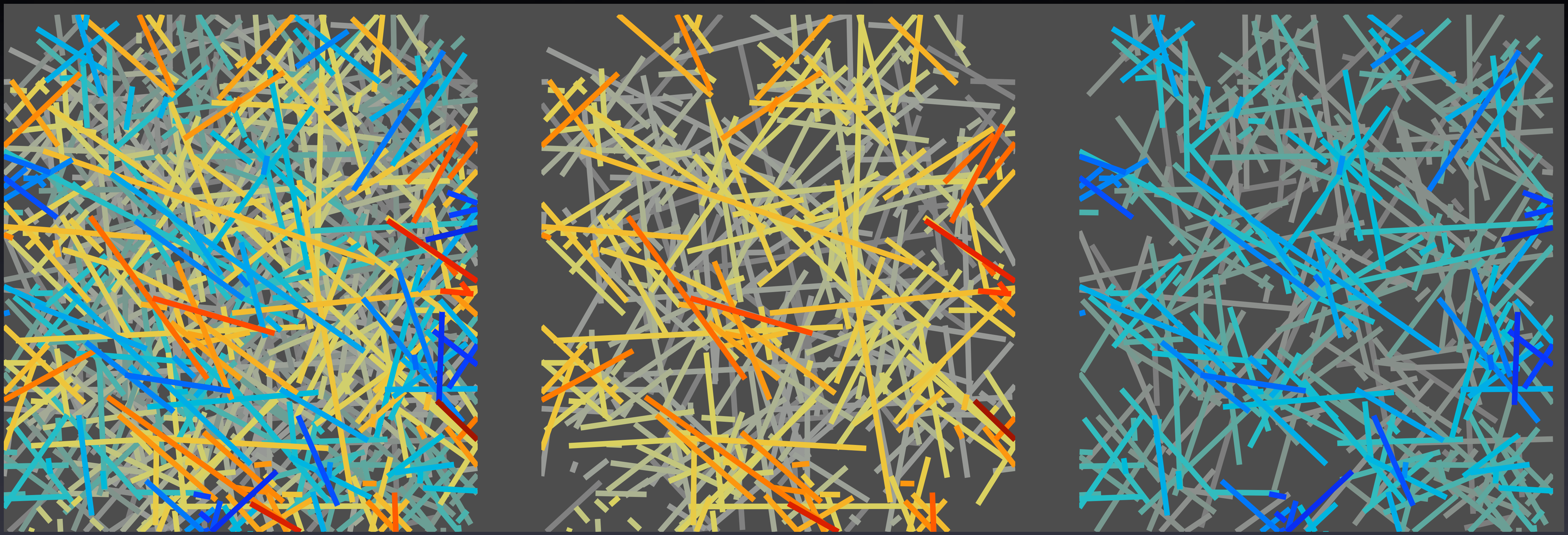


relationship to neurons in V1/V2/...

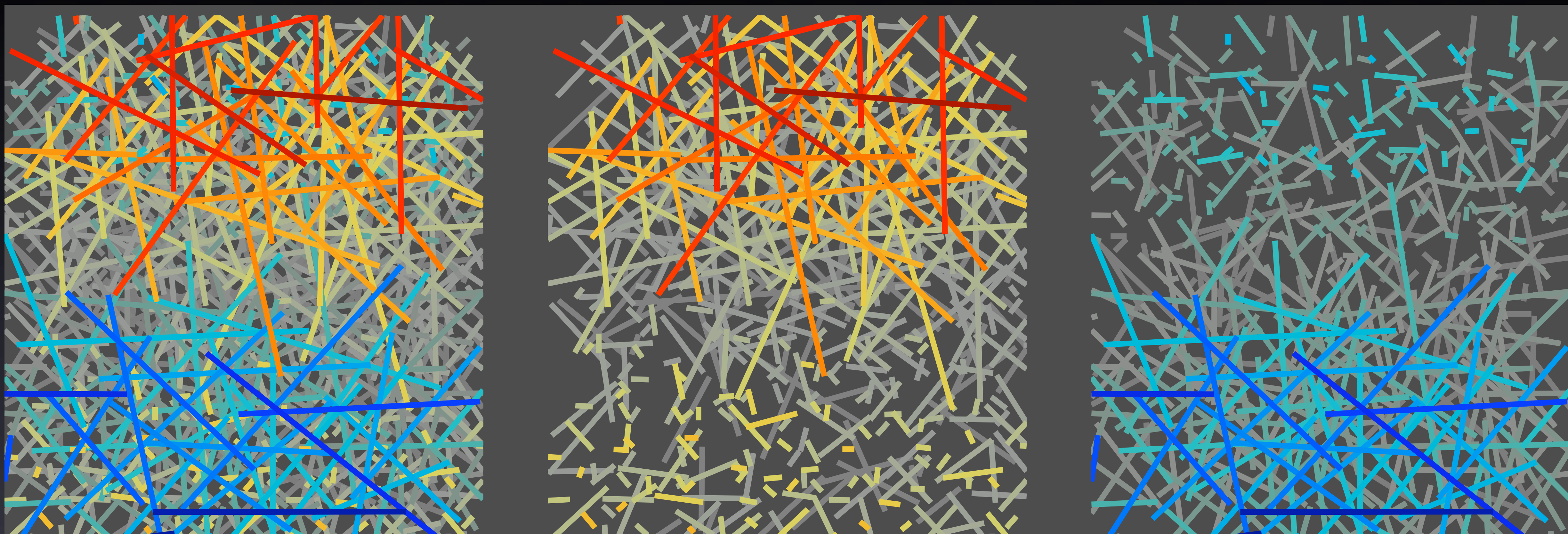
some neurons in V4 appear tuned to be broadly tuned, but selective for orientation

(David et al, 2006)

Another texture field unit

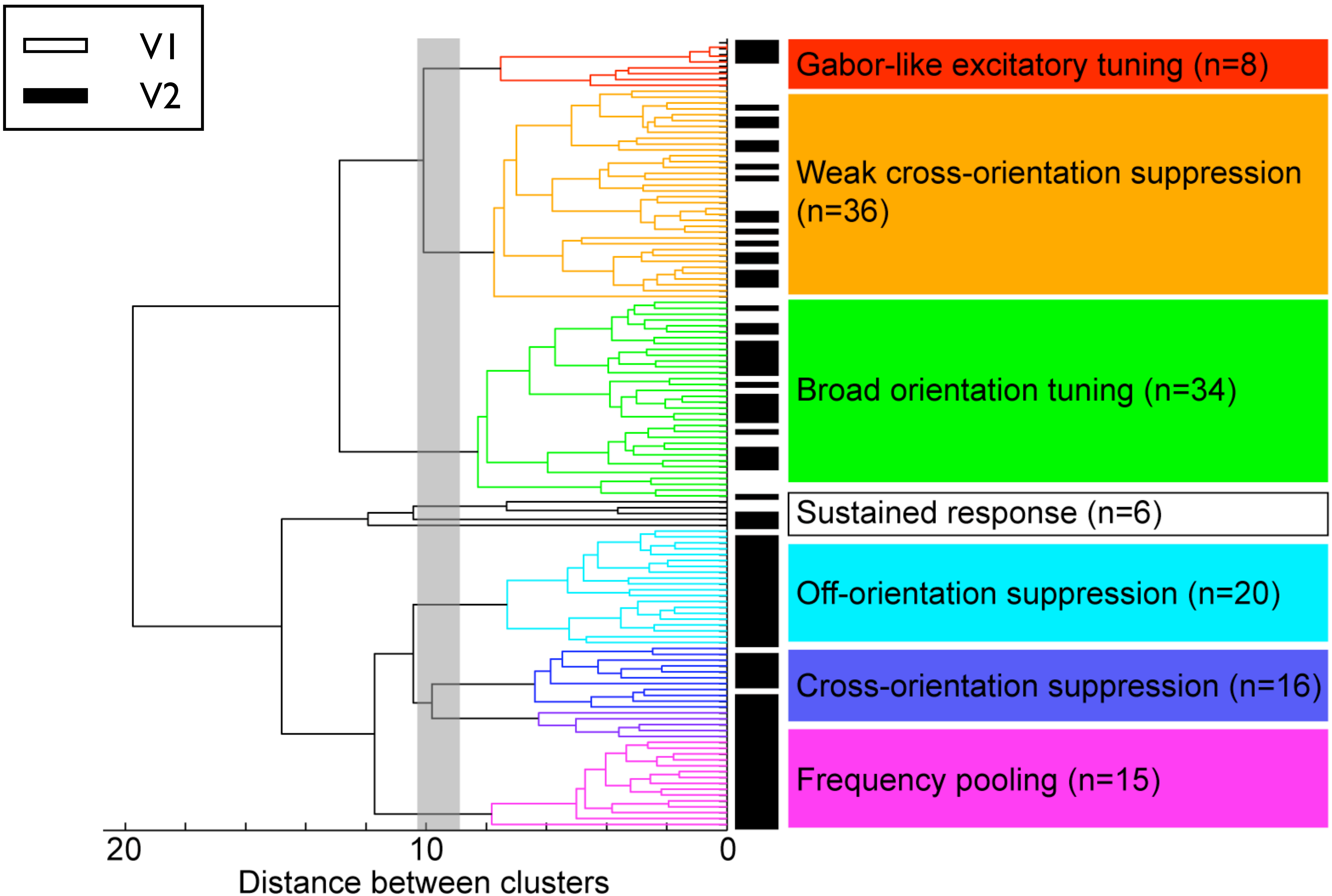


A spatial scale edge unit



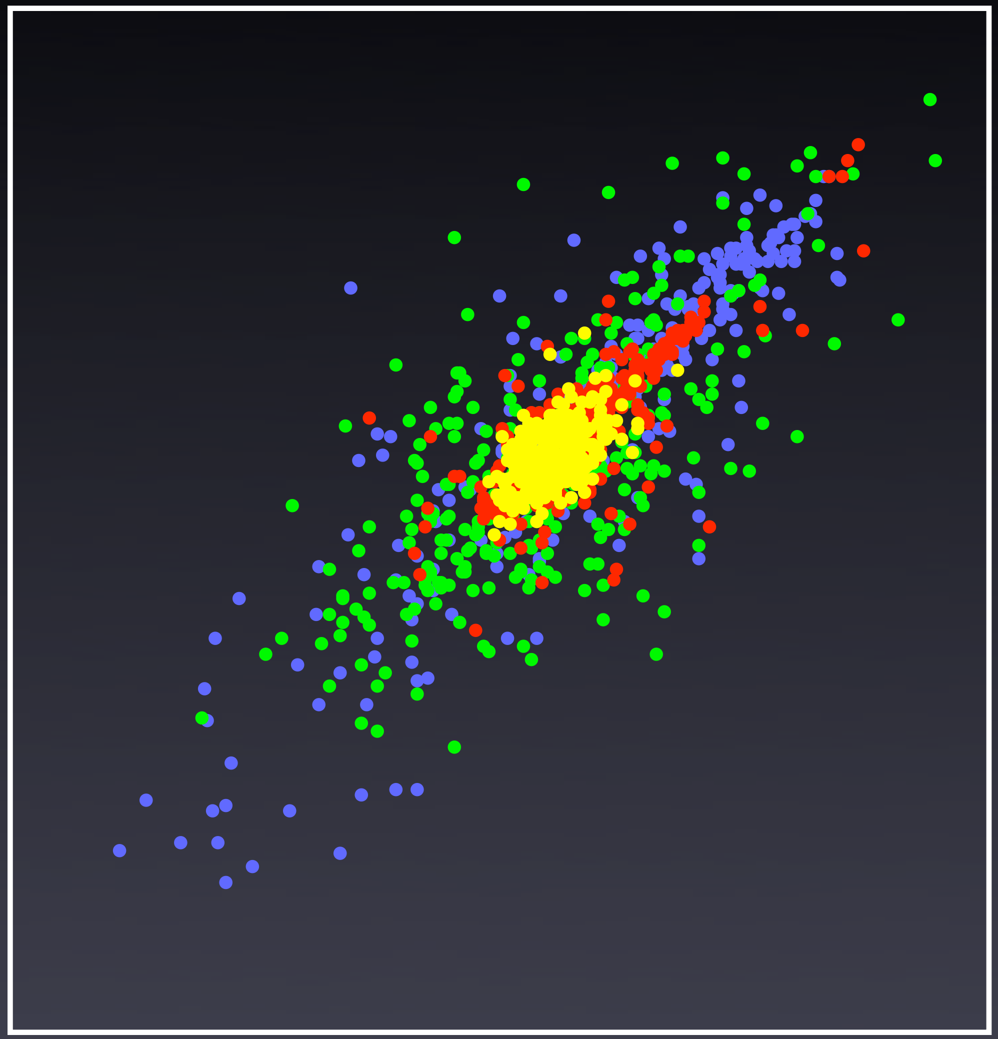
And 100's more ...

V1-V2 taxonomy (data from Willmore and Gallant)



How does the model encode the image regions?

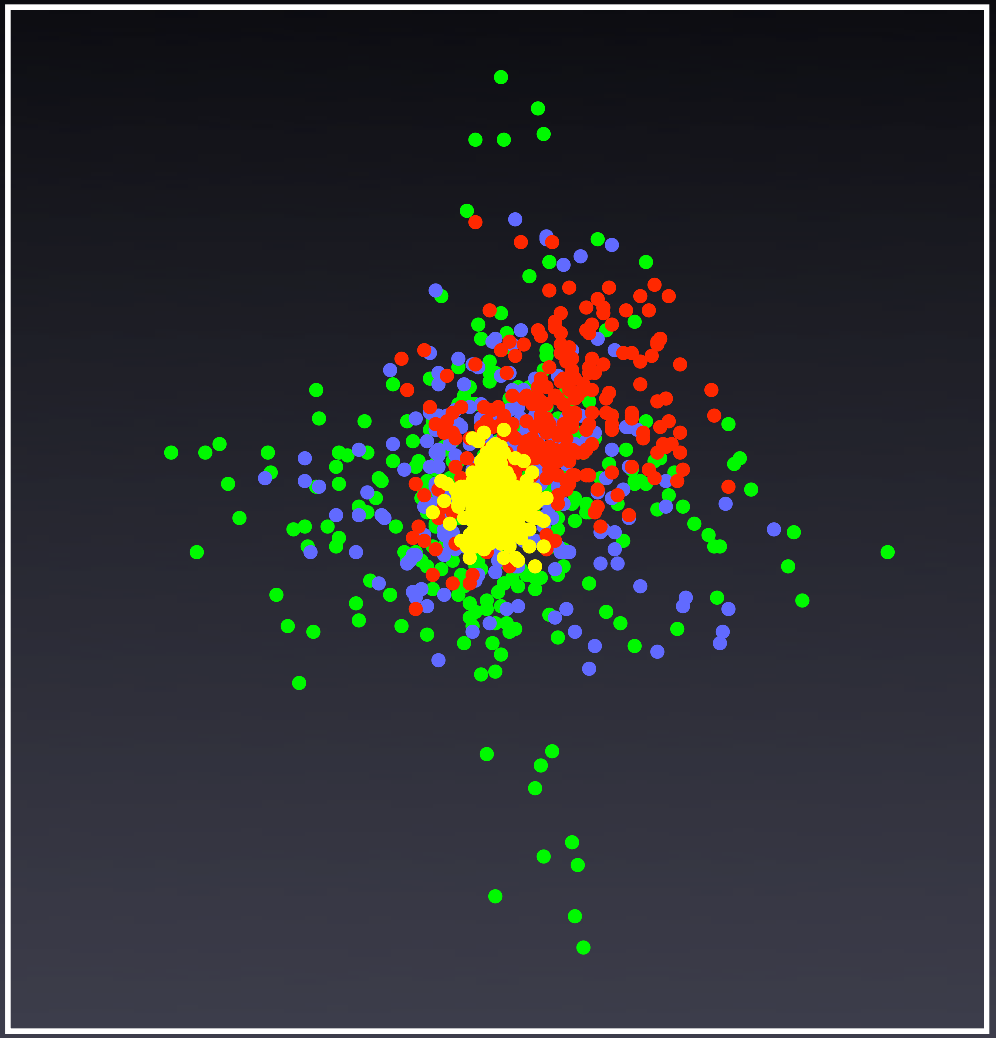
image distributions in pixel space



2D projection of 400D space

How does the model encode the image regions?

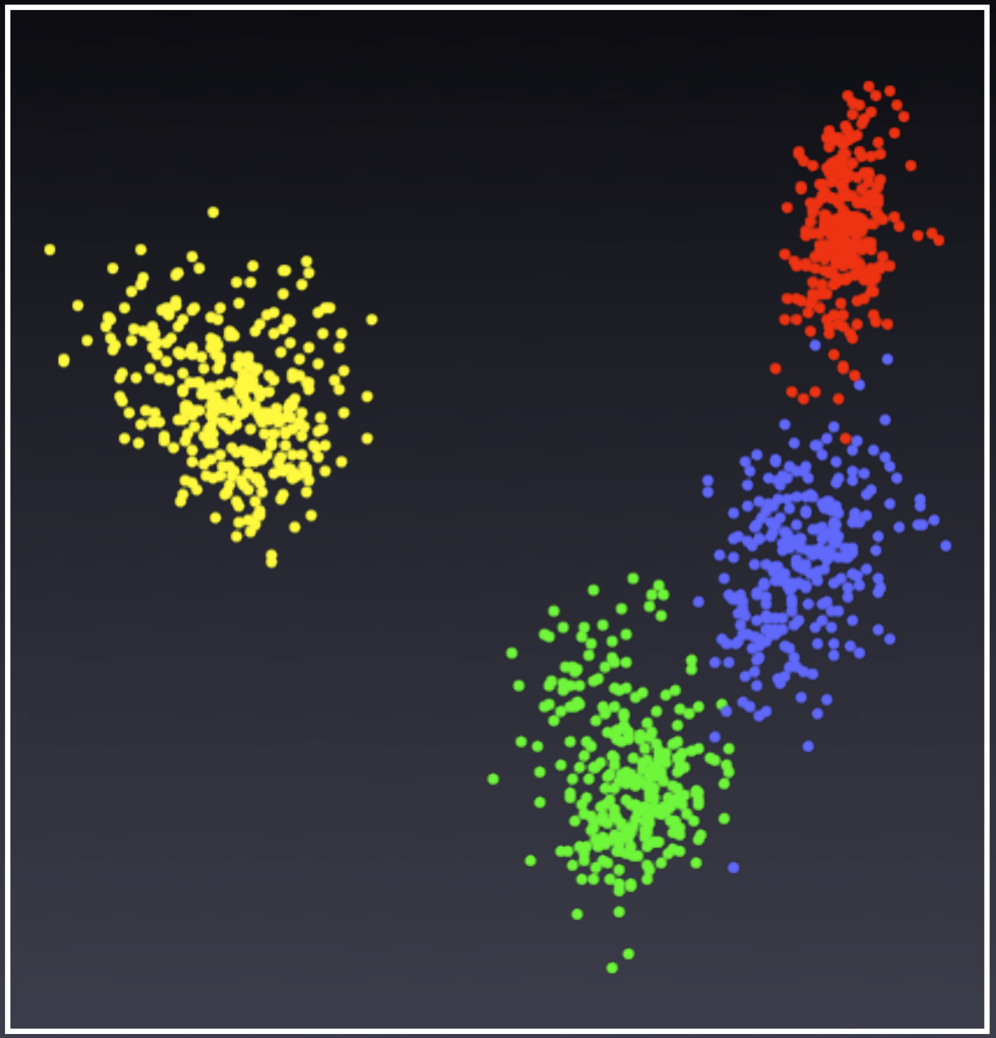
distributions in simple cell space (V1)



2D projection of 400D space

How does the model encode the image regions?

distributions in higher-order space



2D projection of 150D space

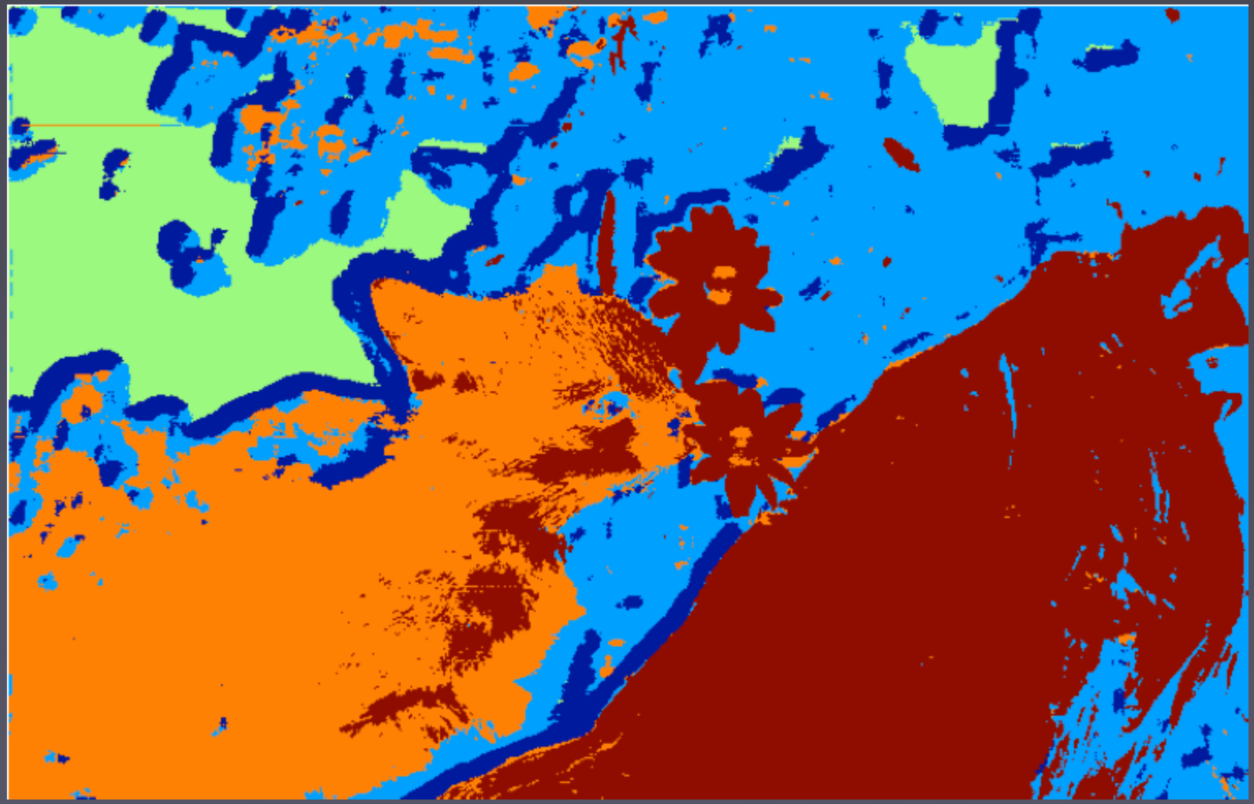
Model generalizes over regions while keeping them distinct.

All *unsupervised.*

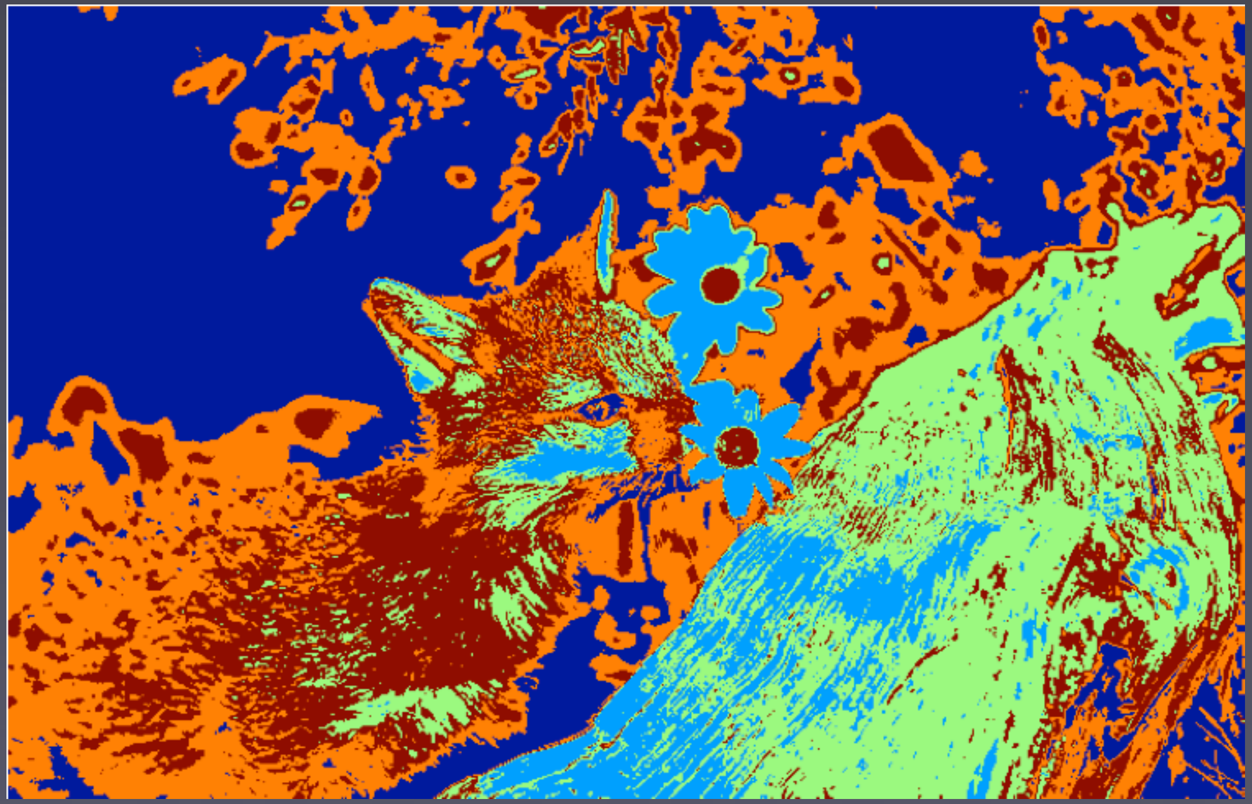
Winner maps

Clustering the higher-order representation yields segmentation

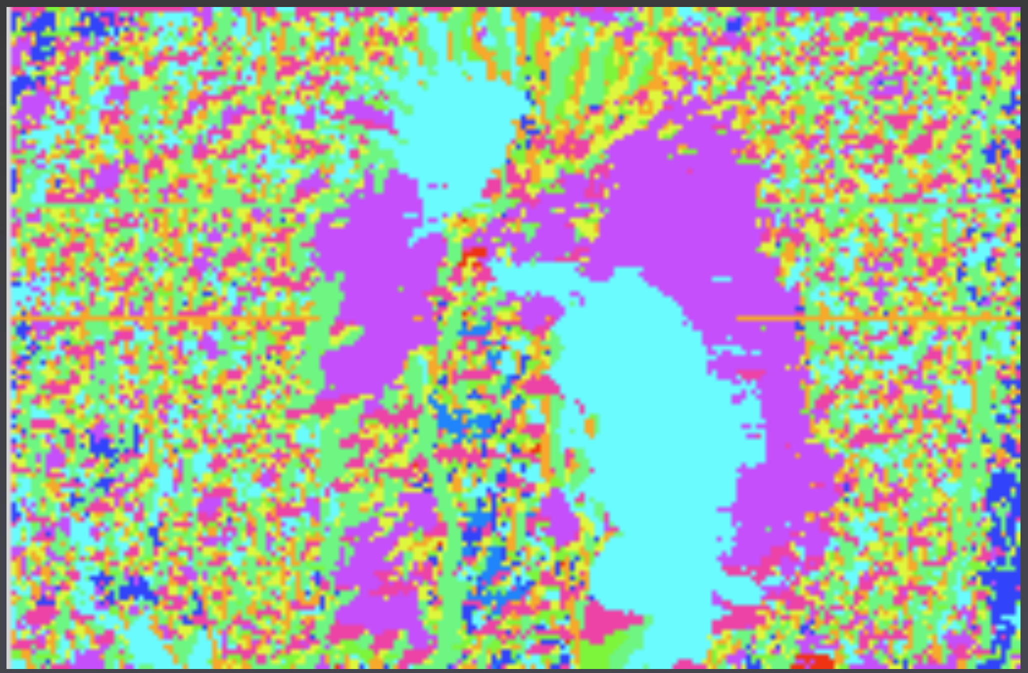
clustering y's

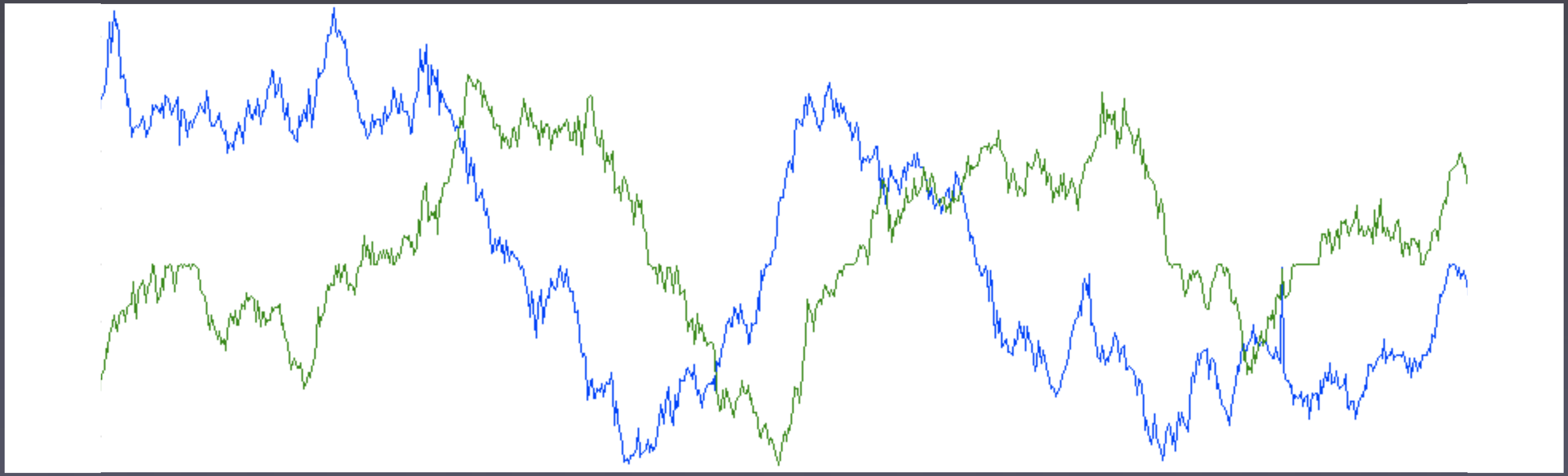


clustering color

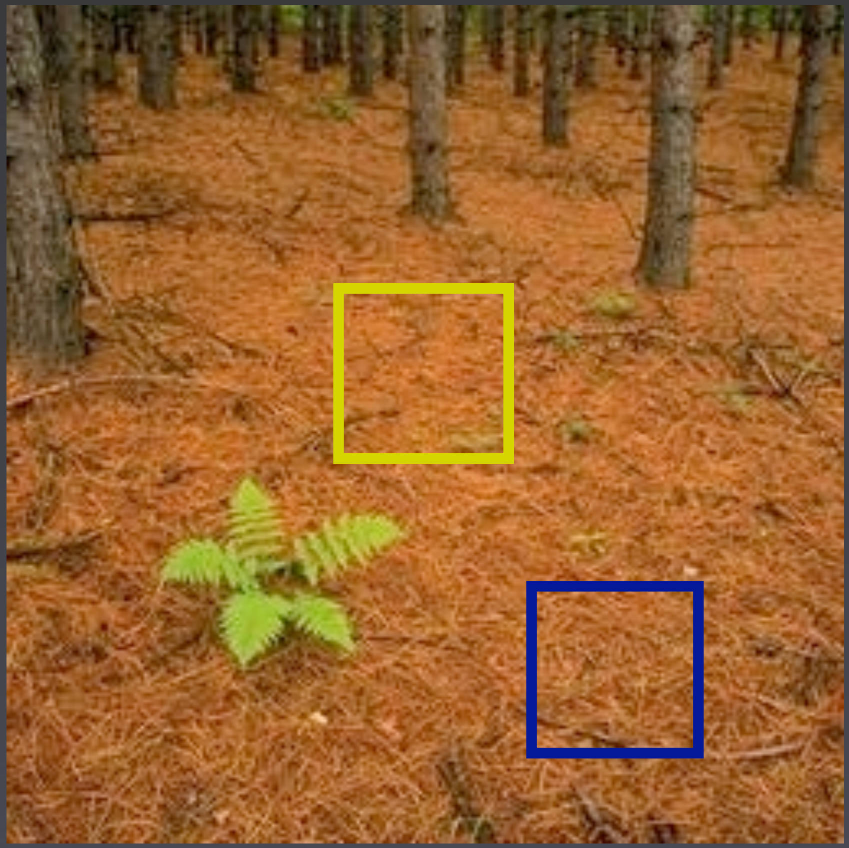


A distributed code for visual surfaces



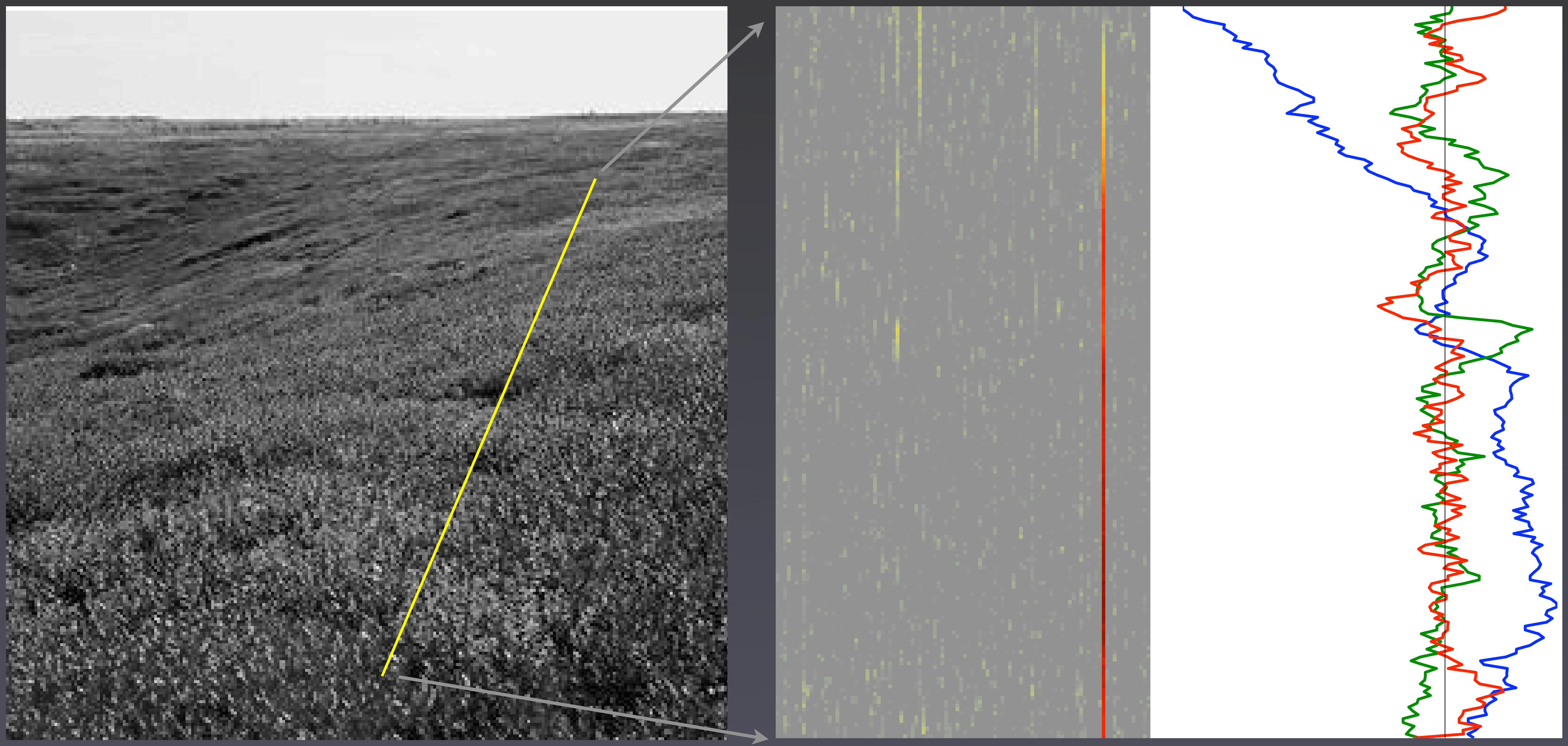


Texture gradients in natural scenes

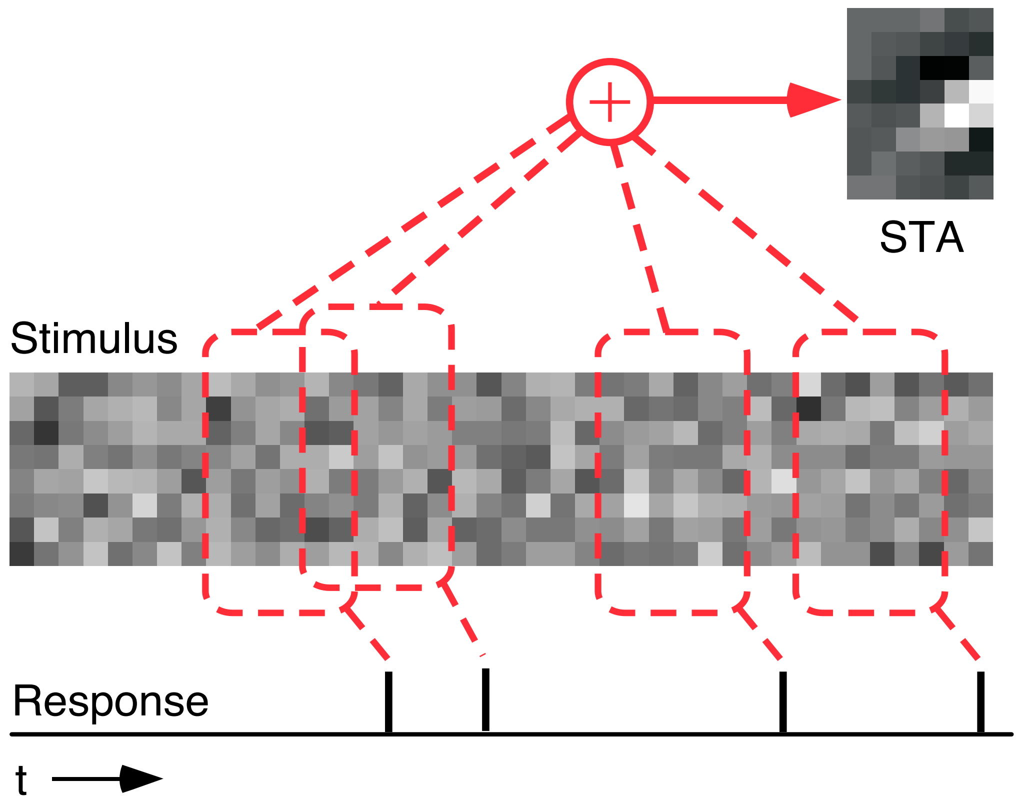


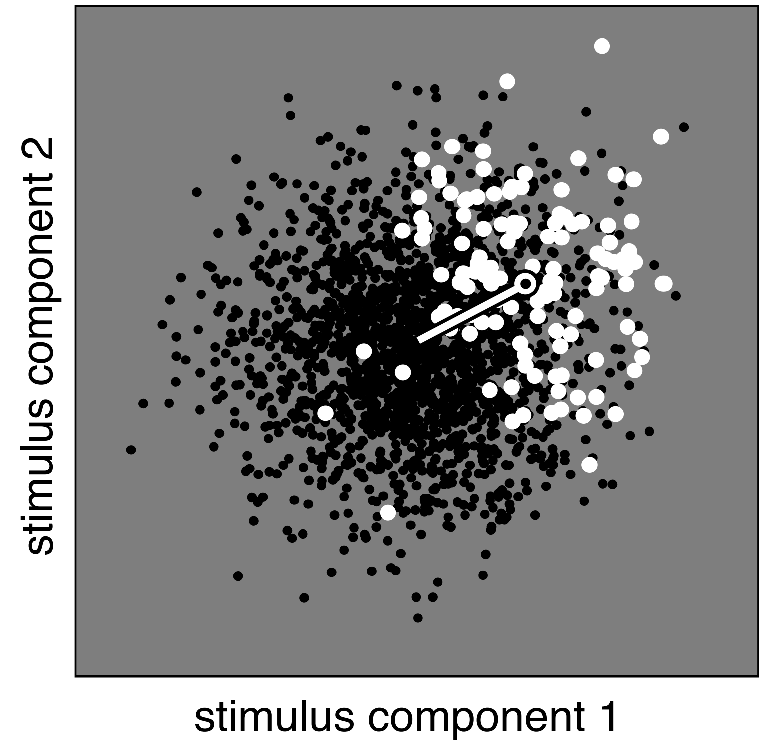
- image density changes continuously
- need to:
  - infer density from single image
  - model smooth & abrupt changes
- important for perceptual organization:
  - 3D scene structure
  - region grouping
  - texture boundaries
- How could this be modeled?

Smooth changes in representation for texture gradients

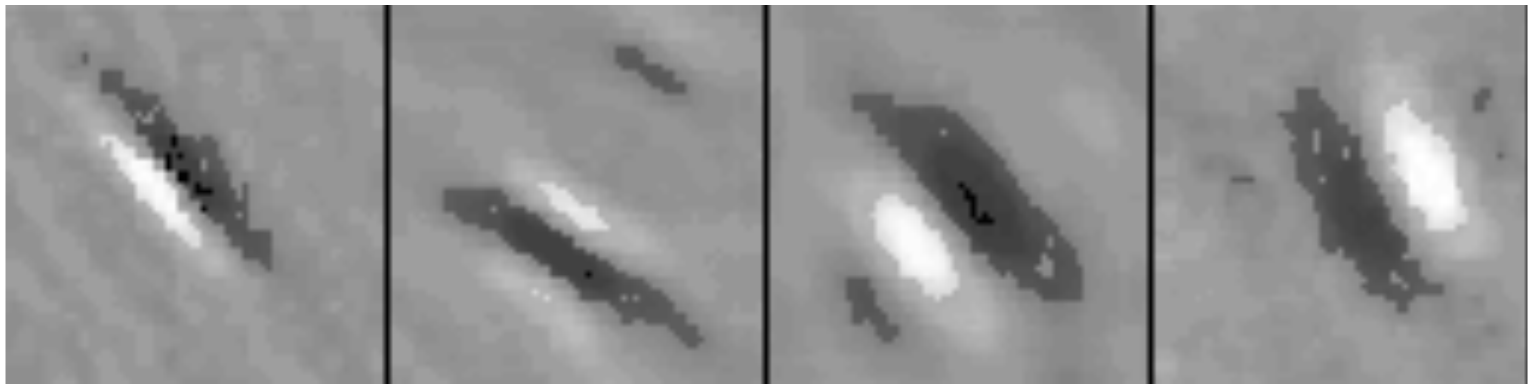


Using spike-triggered averaging to estimate simple cell RFs



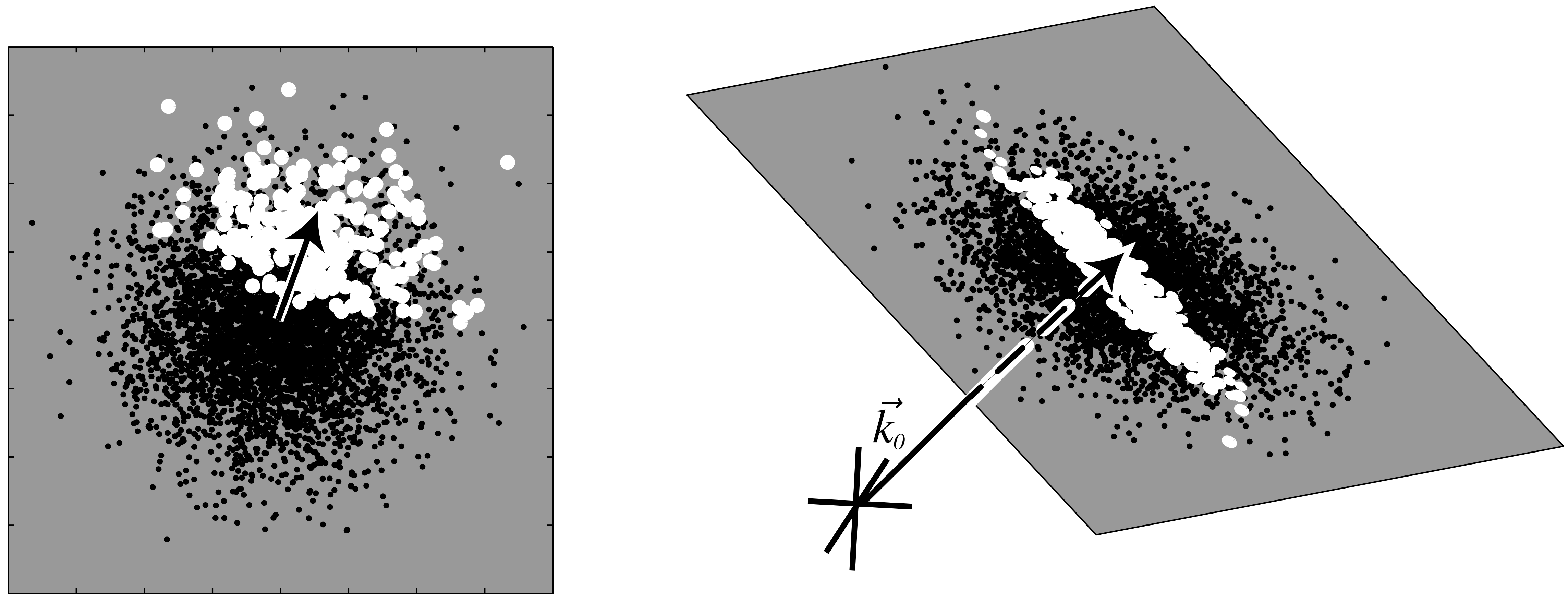


(Simoncelli etal 2004)

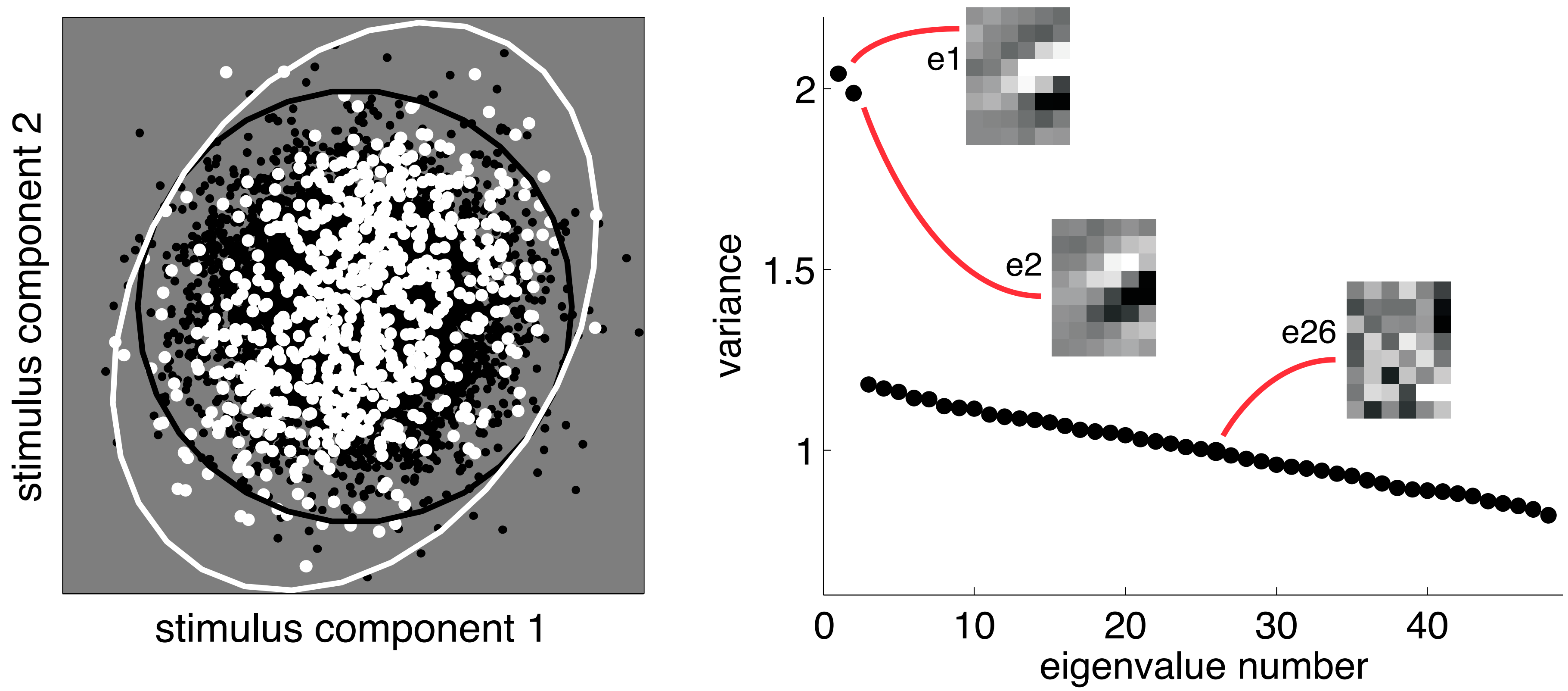


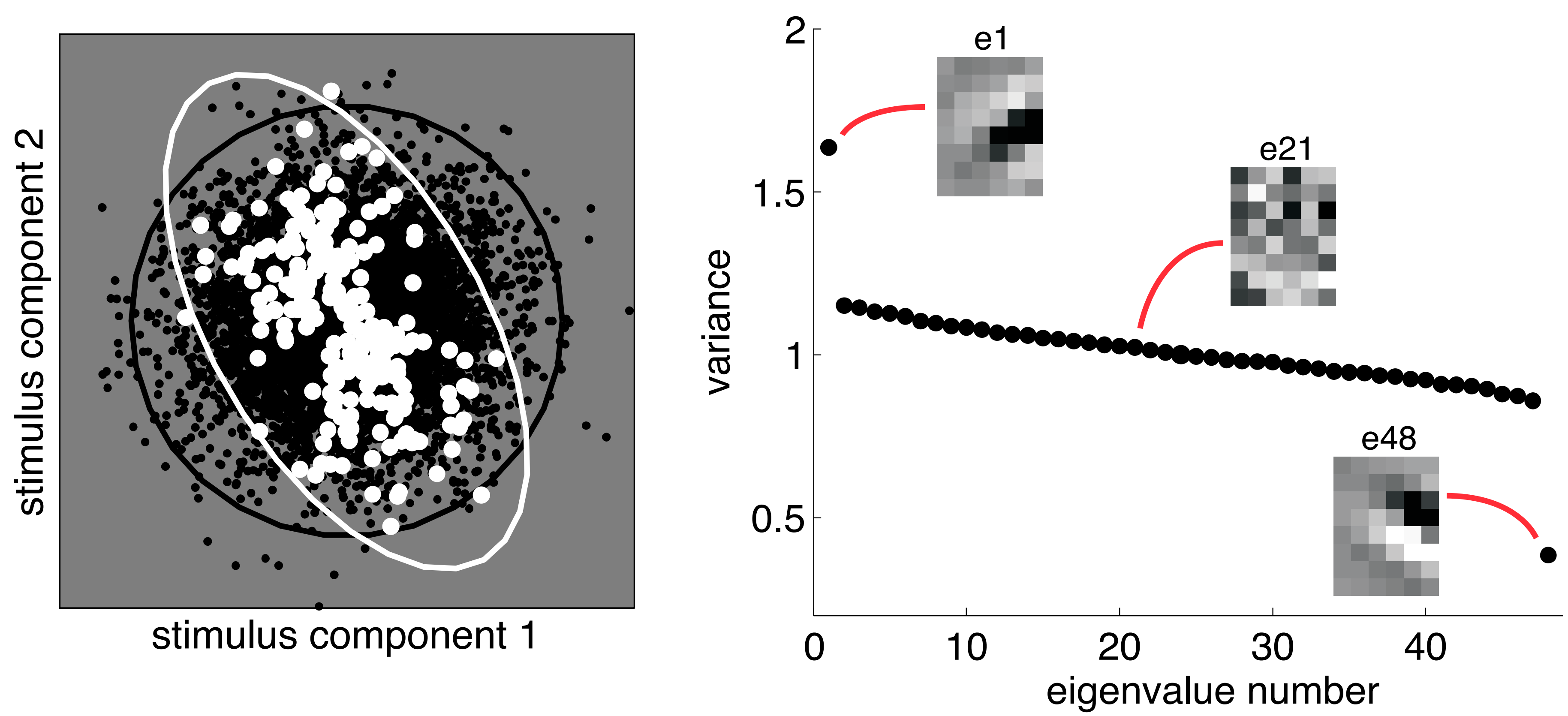
(Ringach 2002)

Spike-triggered covariance (Schwartz, Chichilnisky, and Simoncelli, 02)

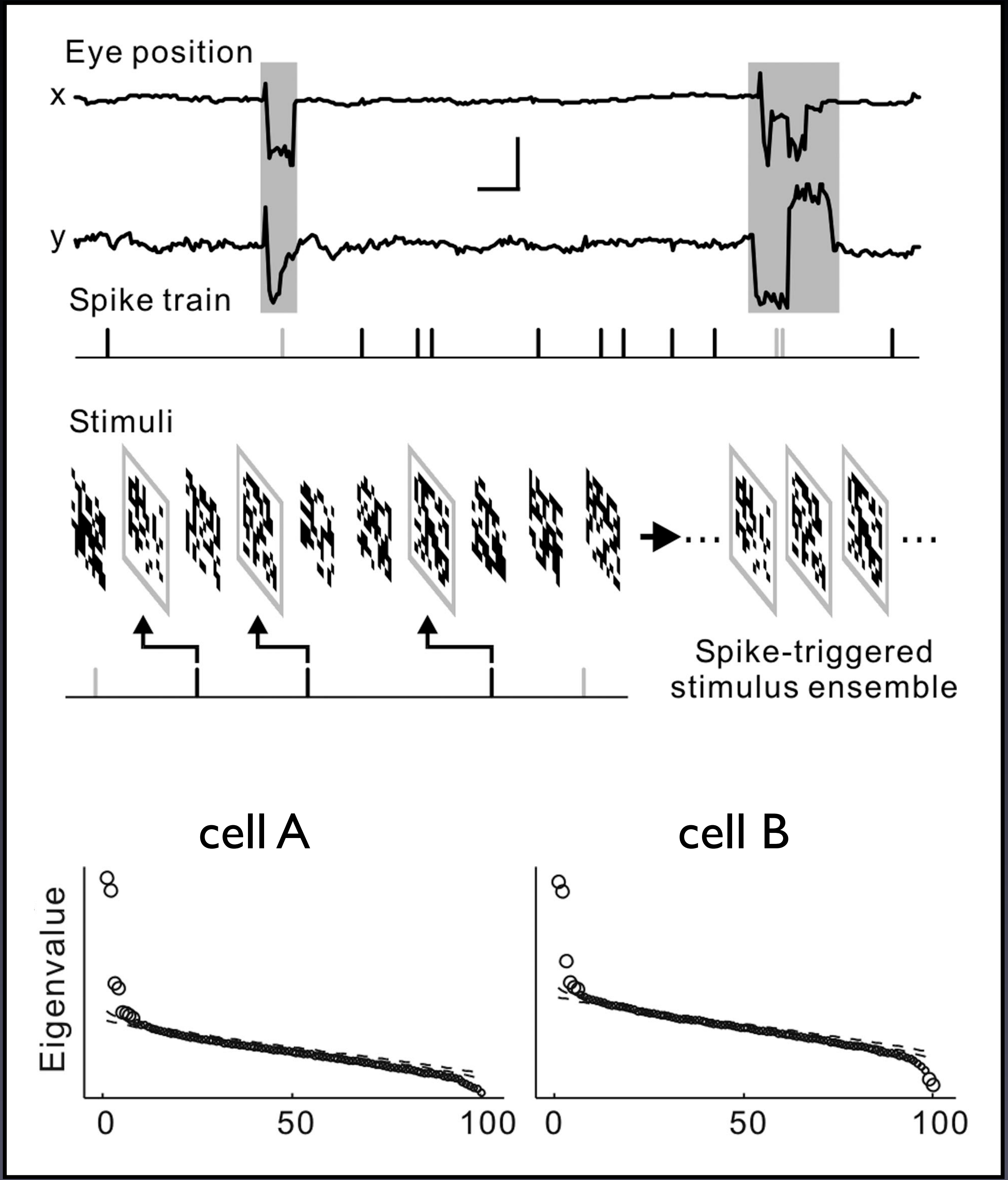


Spike-triggered covariance (Simoncelli, et al, 2004)



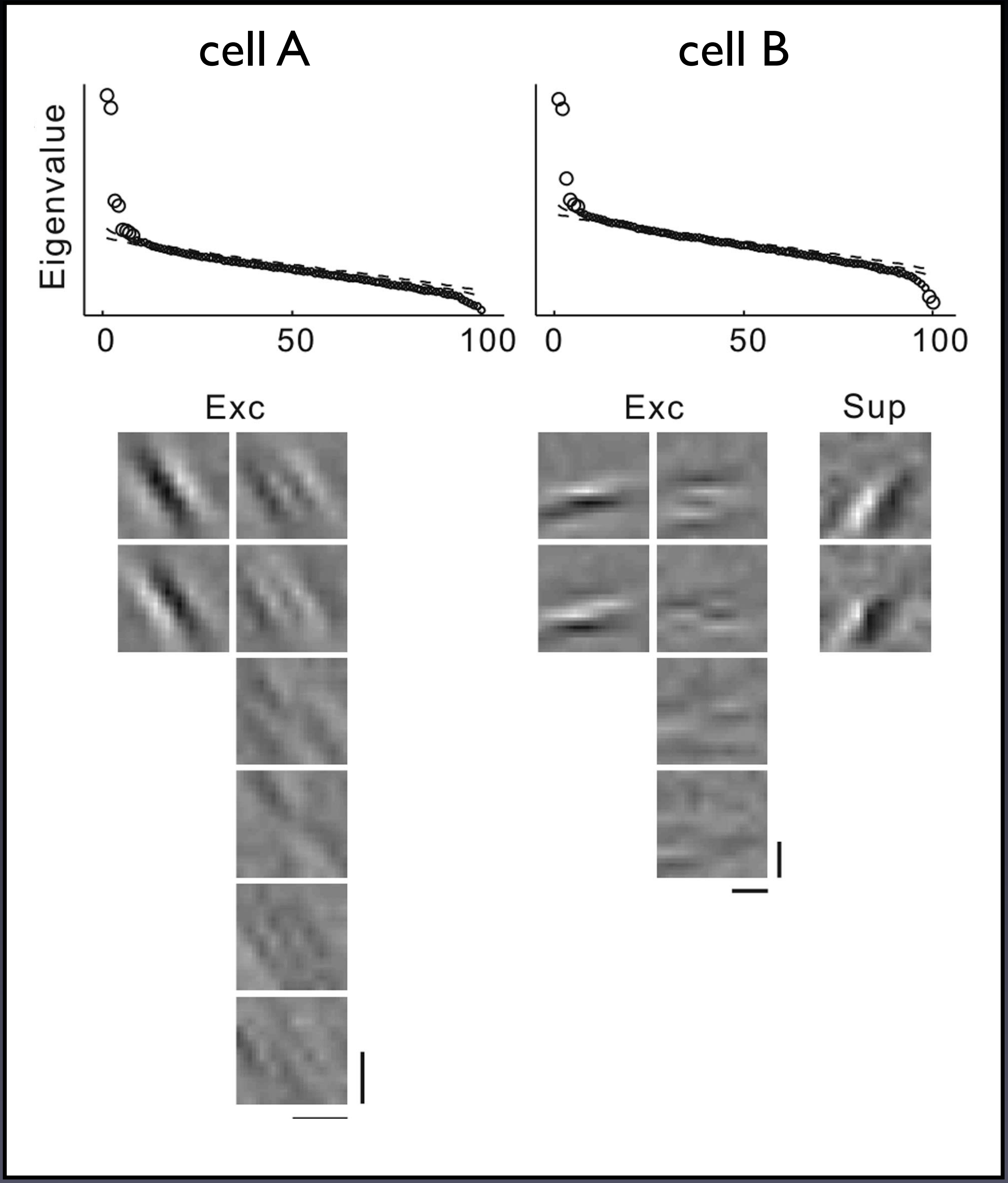


Spike-triggered covariance of macaque V1 neurons (Chen et al '07)



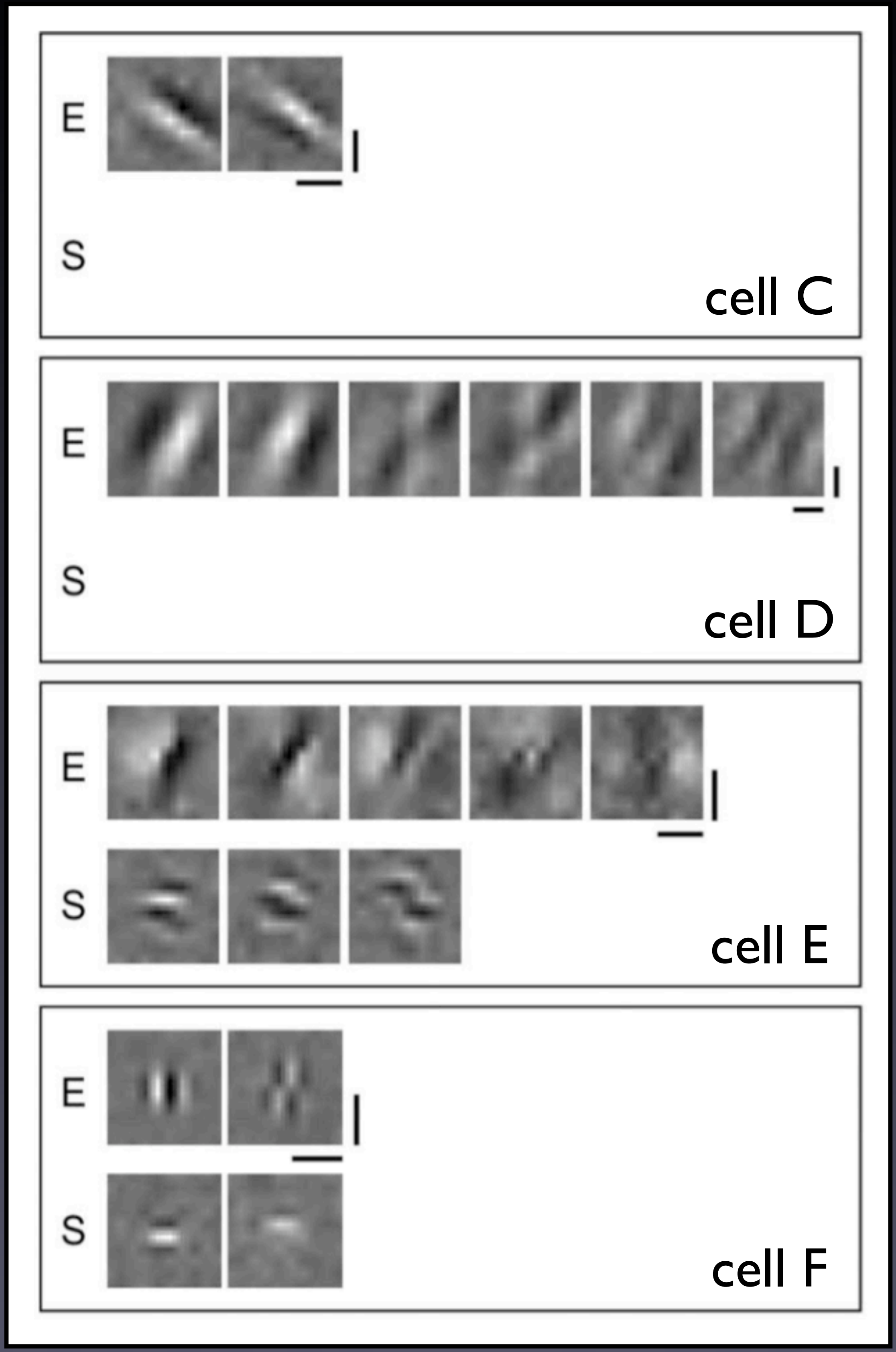
- collect spikes to random binary stimuli (10x10 to 12x12)
- throw out eye positions outside fixation window
- collect spike-triggered simulus ensemble
- spike-triggered average:
  - average ensemble
- spike-triggered covariance:
  - compute ensemble covariance
  - select significant eigenvectors

Spike-triggered covariance of macaque V1 neurons (Chen et al '07)



- eigenvectors represent excitatory and suppressive sub-units
  - not necesarily anatomical, eg simple cells
  - could be linear combinations of anatomical subunits
- some only excitatory
- suppressive usually weaker
- note off-orientation (non-orthogonal) suppressive subunits

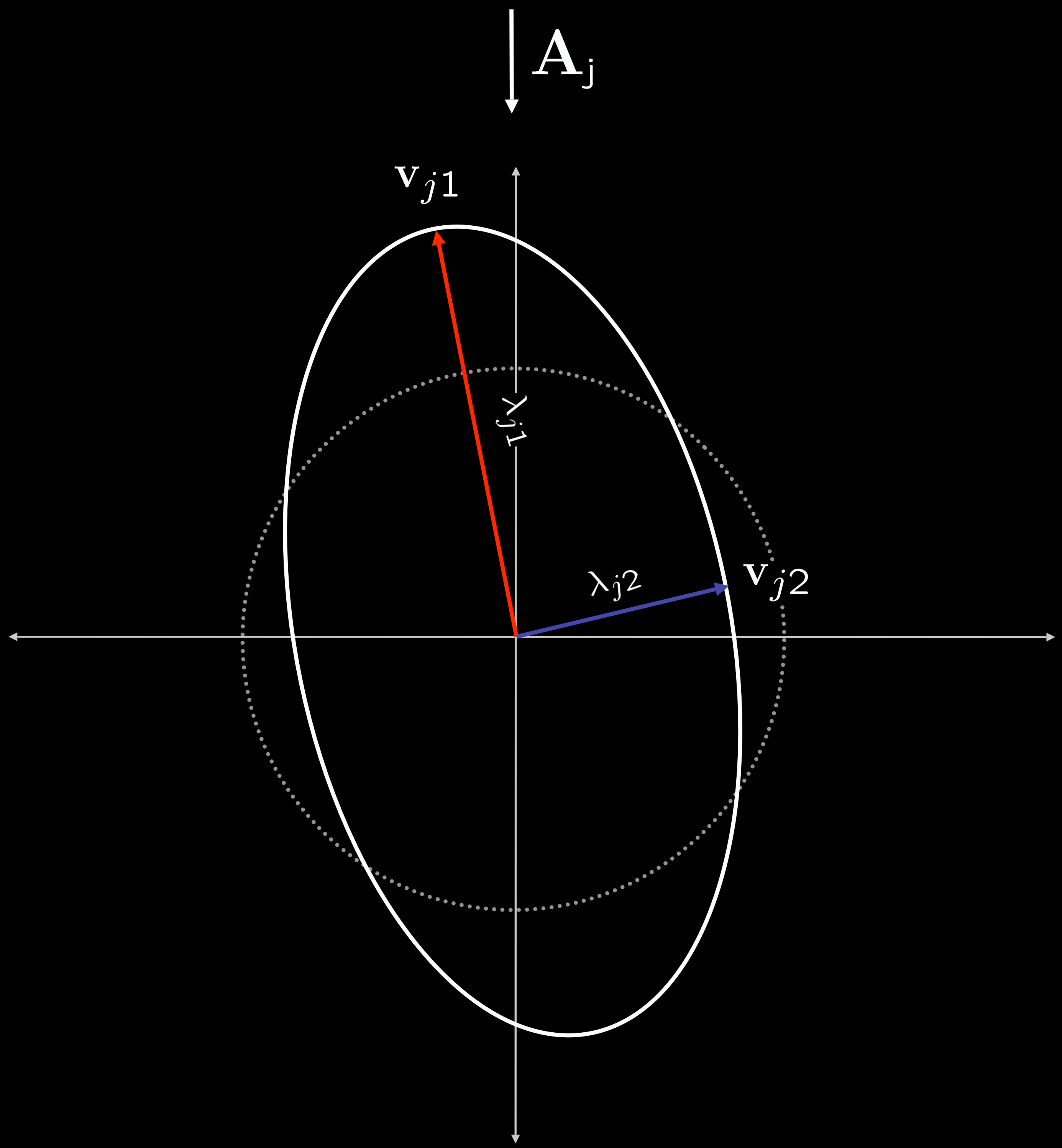
Spike-triggered covariance of macaque V1 neurons (Chen et al '07)



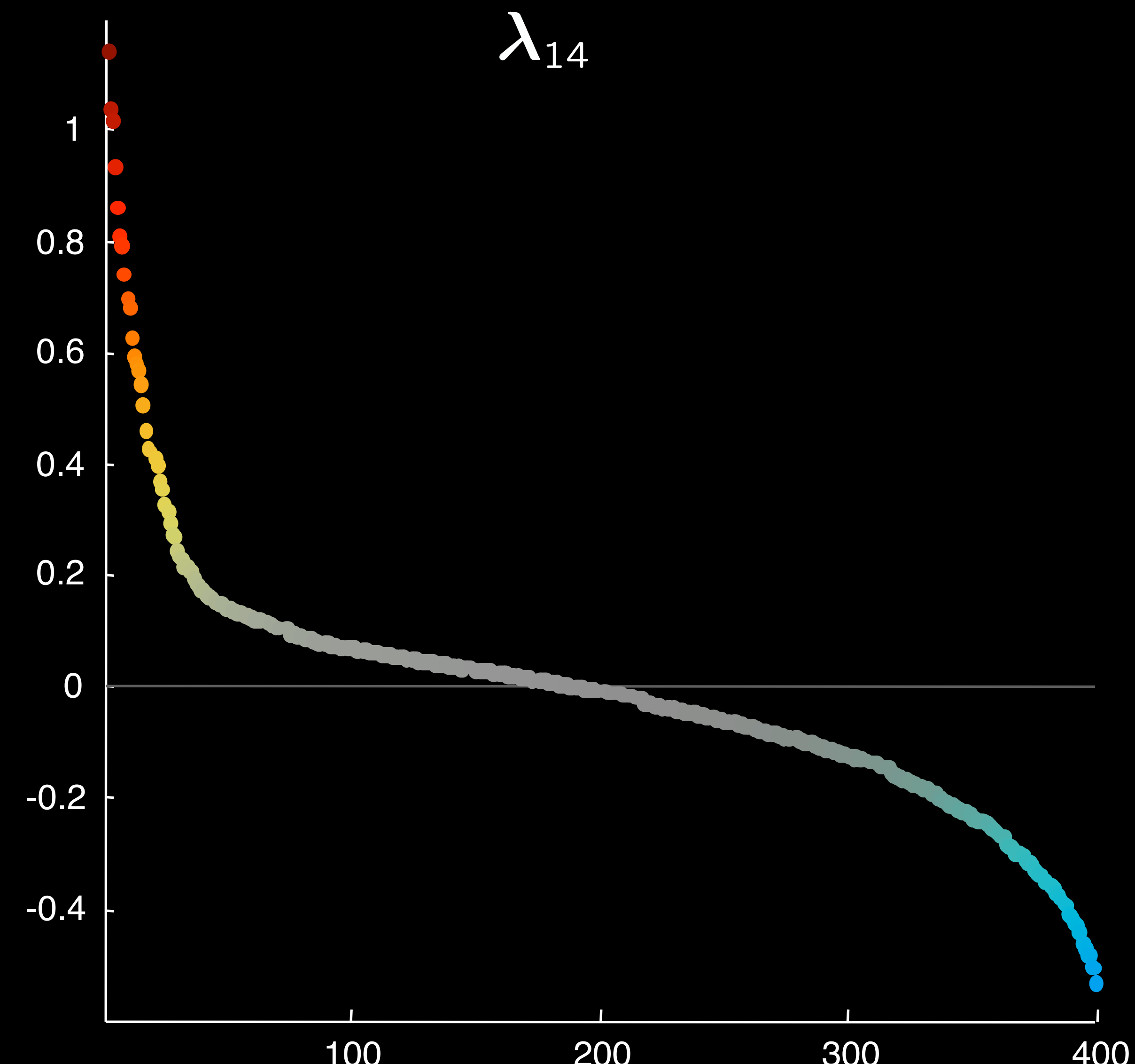
- other types of STC eigenvectors
- some properties:
  - quadrature Gabor-like pairs (from orthogonality)
  - cross-orientation suppression

Analysis of log-covariance components

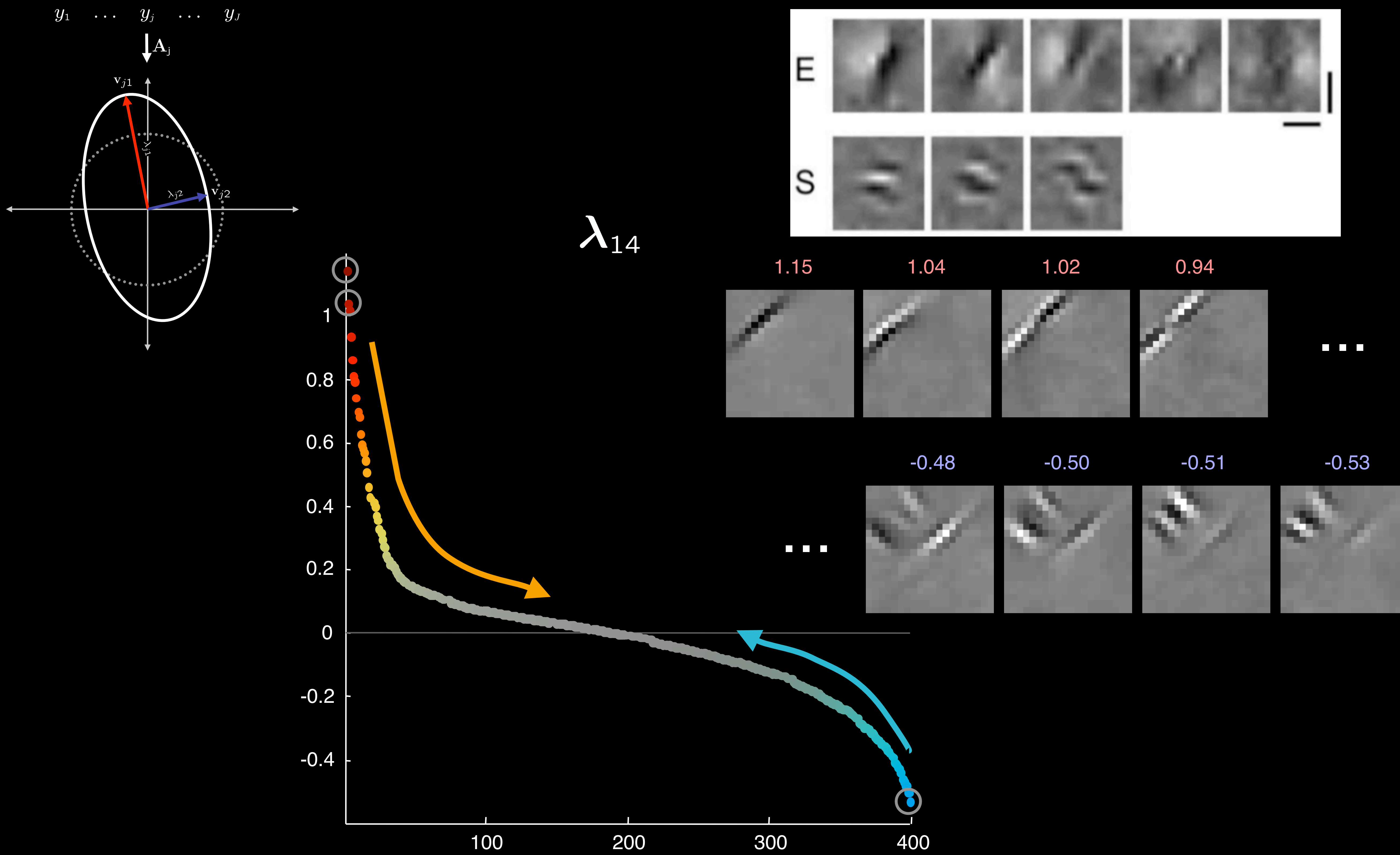
$$y_1 \quad \ldots \quad y_j \quad \ldots \quad y_J$$



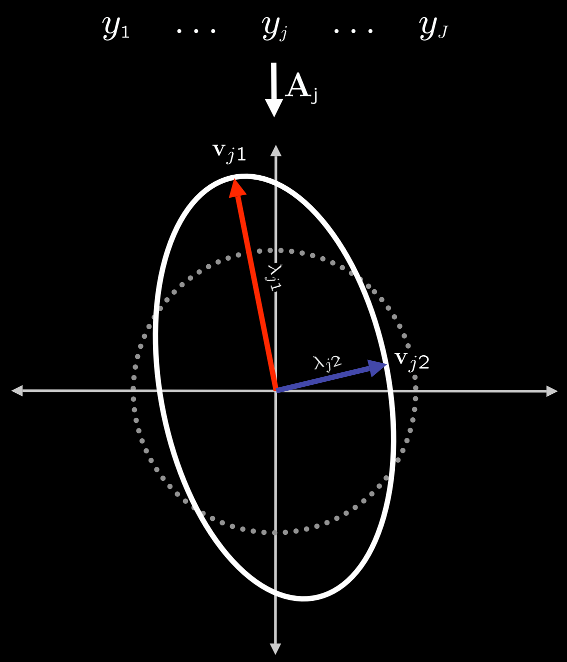
in all 400 dimensions:

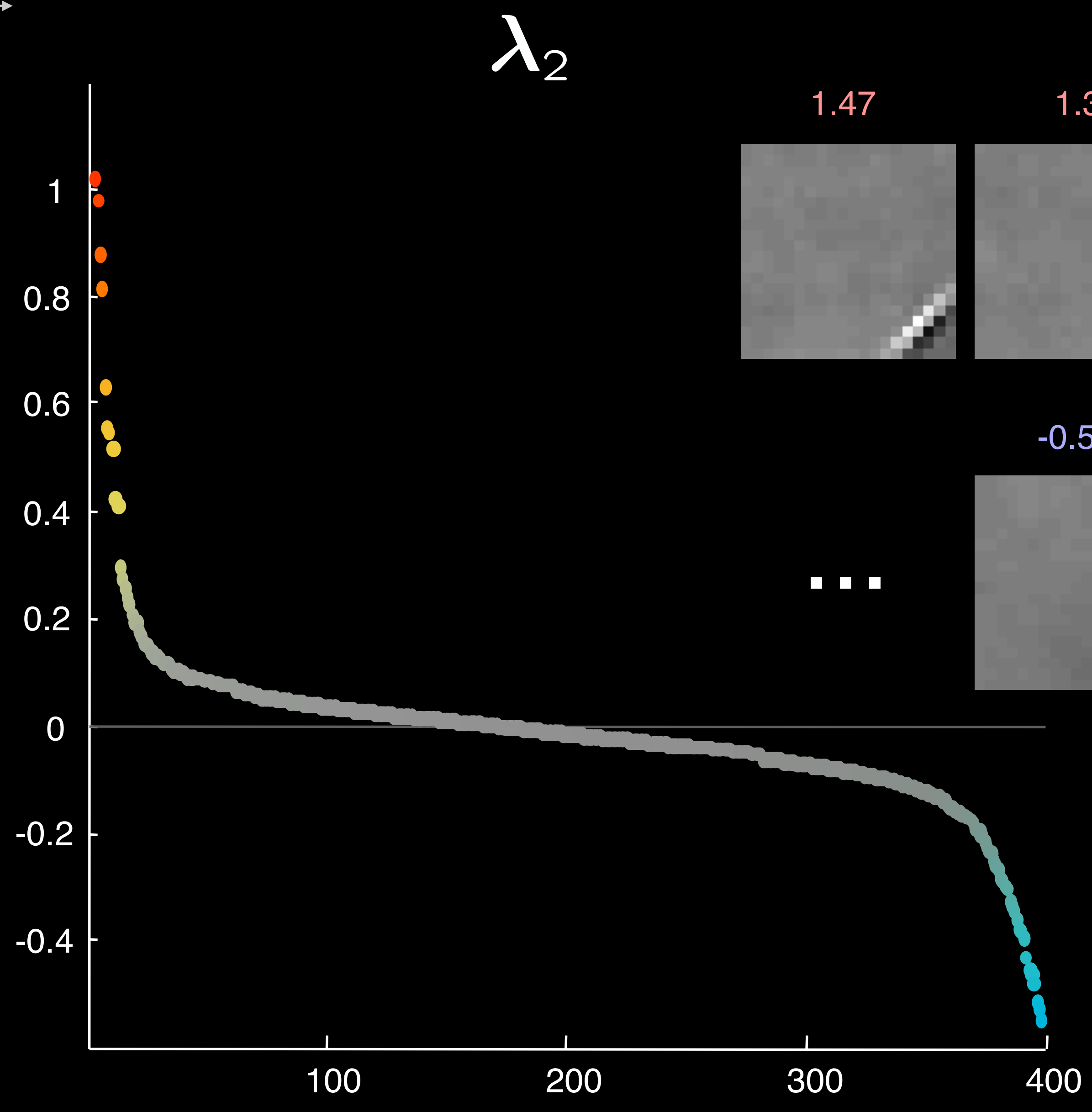


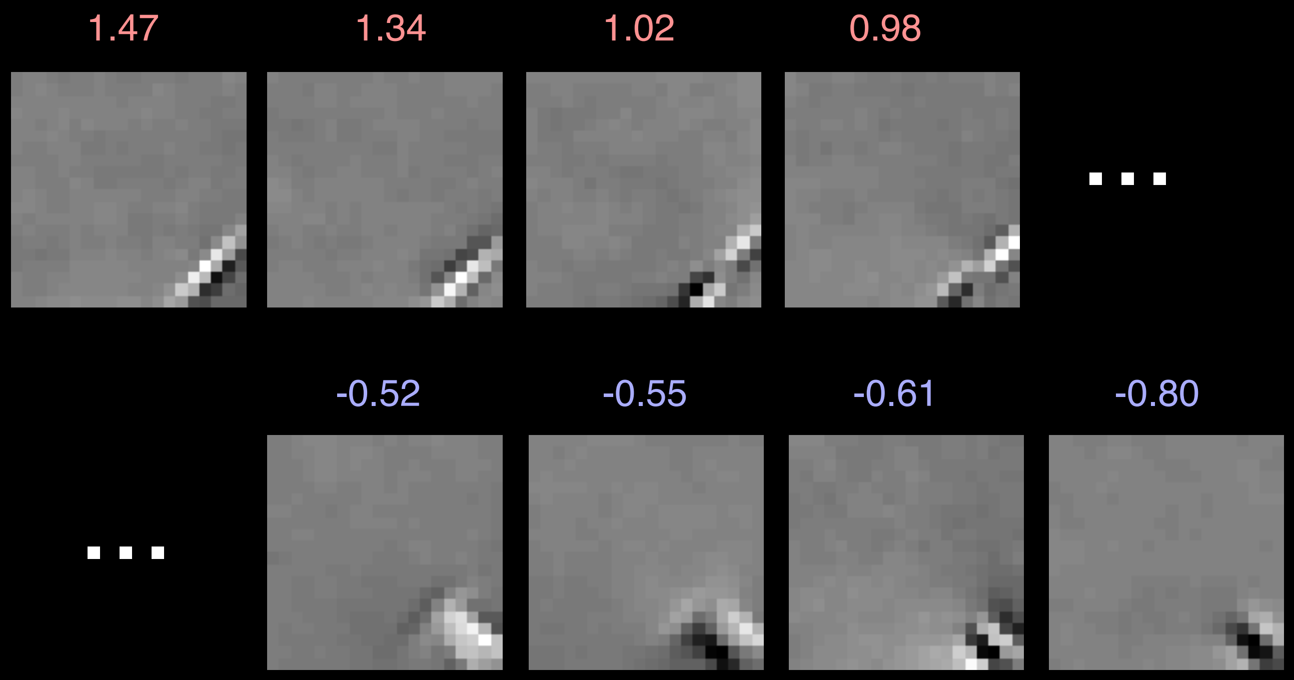
Analysis of log-covariance components



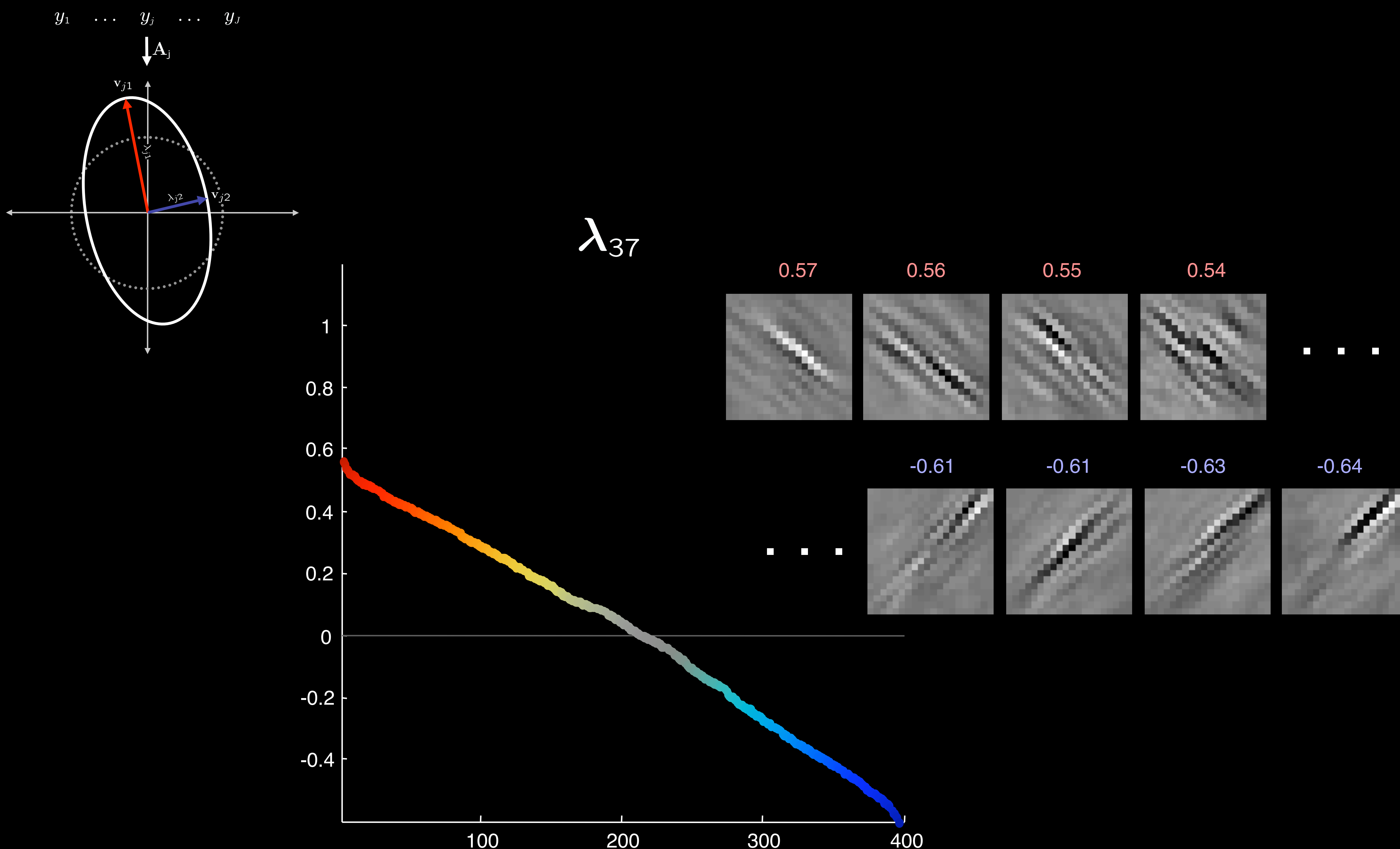
Analysis of log-covariance components

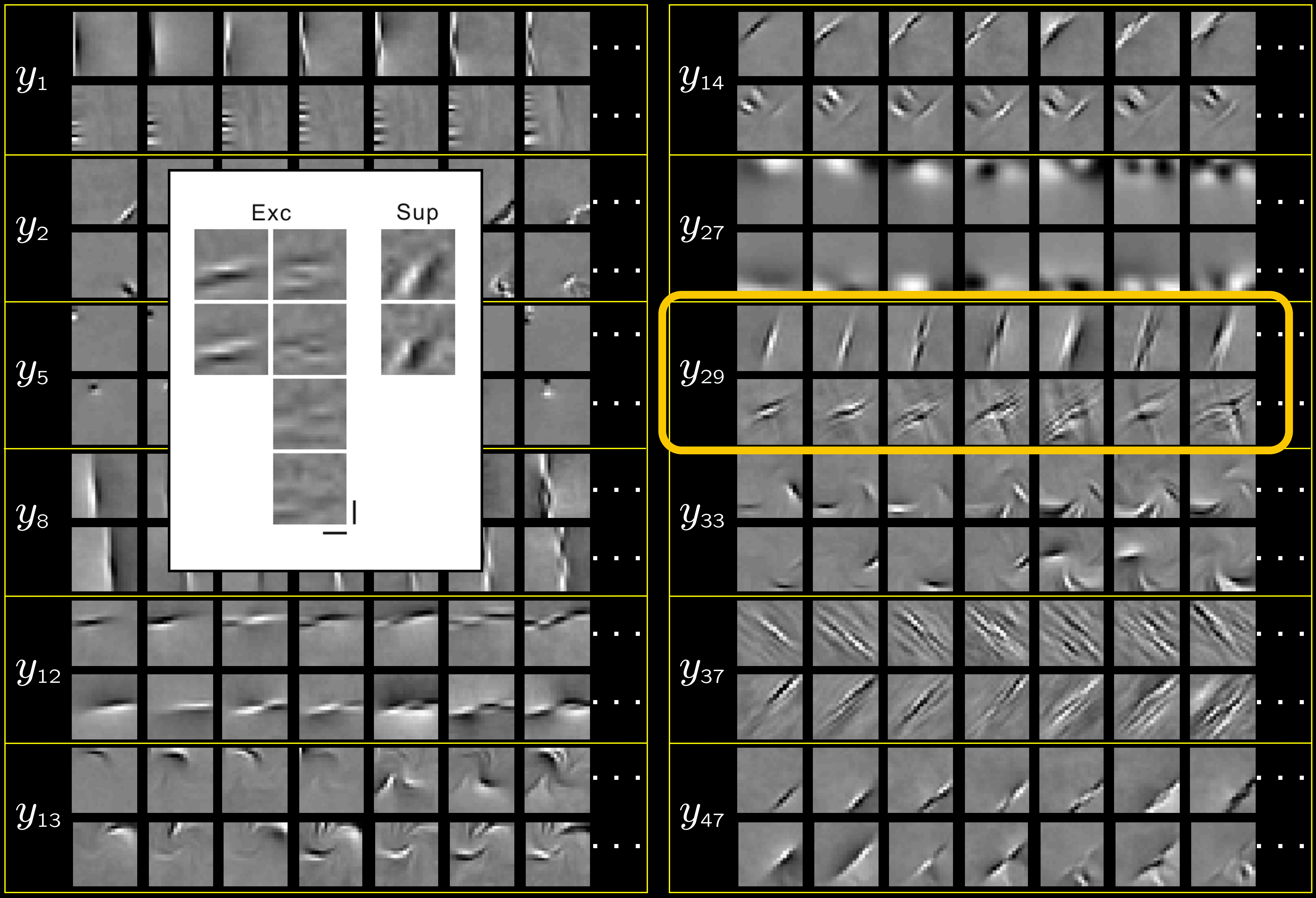




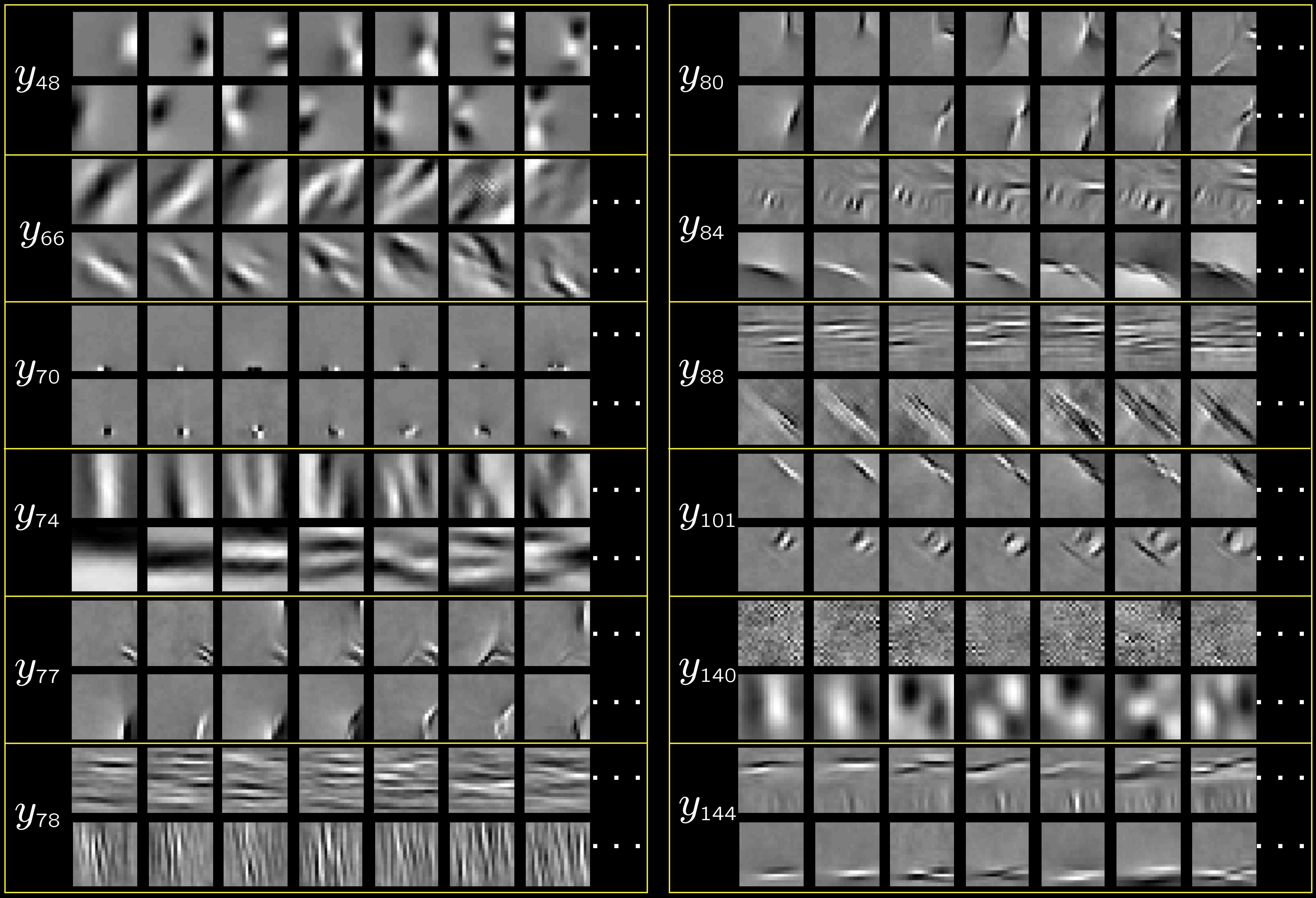


Analysis of log-covariance components





y_1
y_2
y_5
y_8
y_{12}
y_{13}
Exc
Sup
y_{14}
y_{27}
y_{29}
y_{33}
y_{37}
y_{47}



y_{48}
y_{66}
y_{70}
y_{74}
y_{77}
y_{78}
y_{80}
y_{84}
y_{88}
y_{101}
y_{140}
y_{144}

Summary

- We need to understand problem being solved:
  generalization in natural scenes

- Proposed model to solve it:
  - no a priori biological assumptions or constraints

- Right problem should give us insight
- Probabilities fundamental to solving this problem
- Good results:
  - solves interesting & relevant problem
  - many similarities to physiology
  - many novel predictions & interpretations