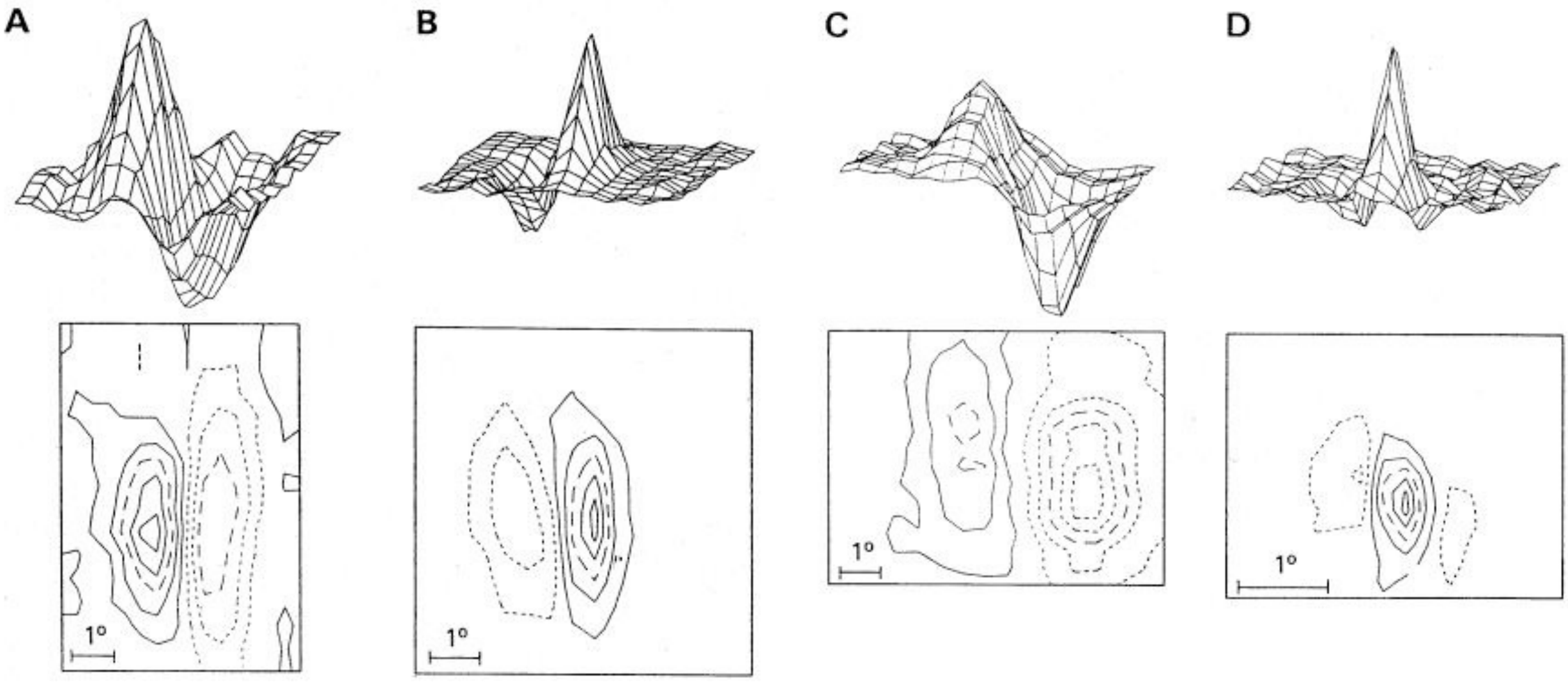
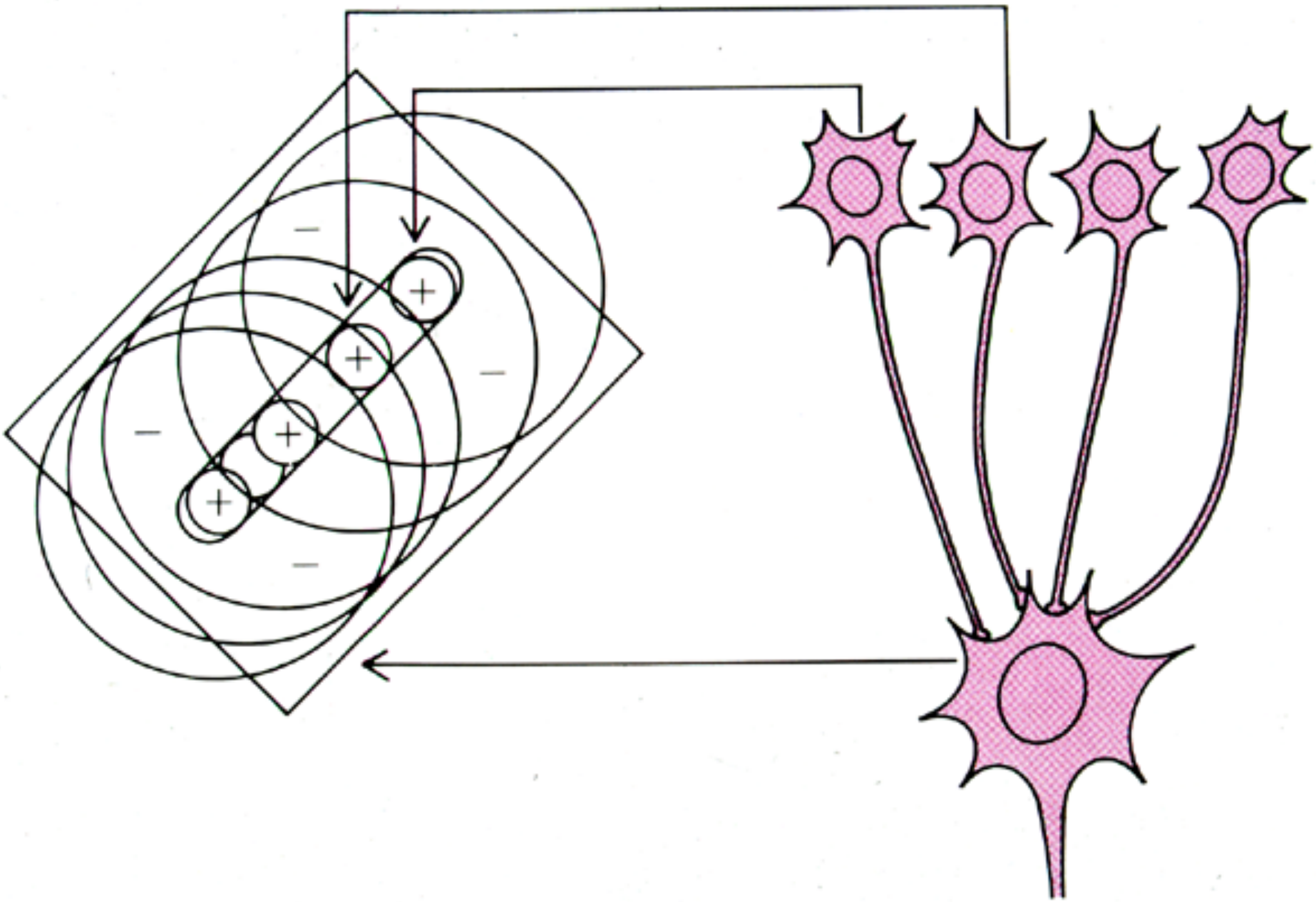
V1 - simple cell receptive fields

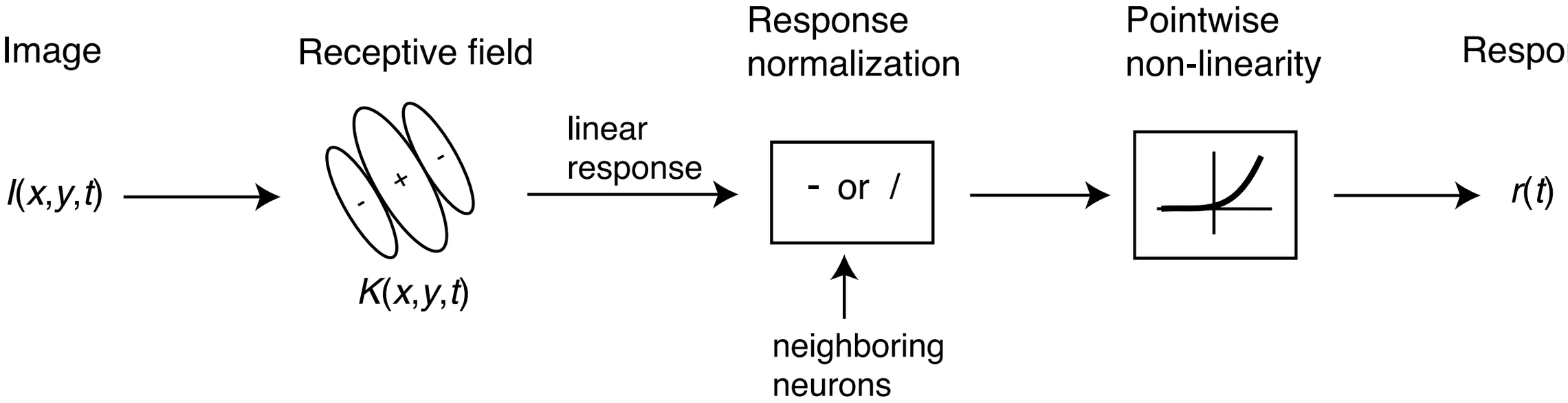


combine

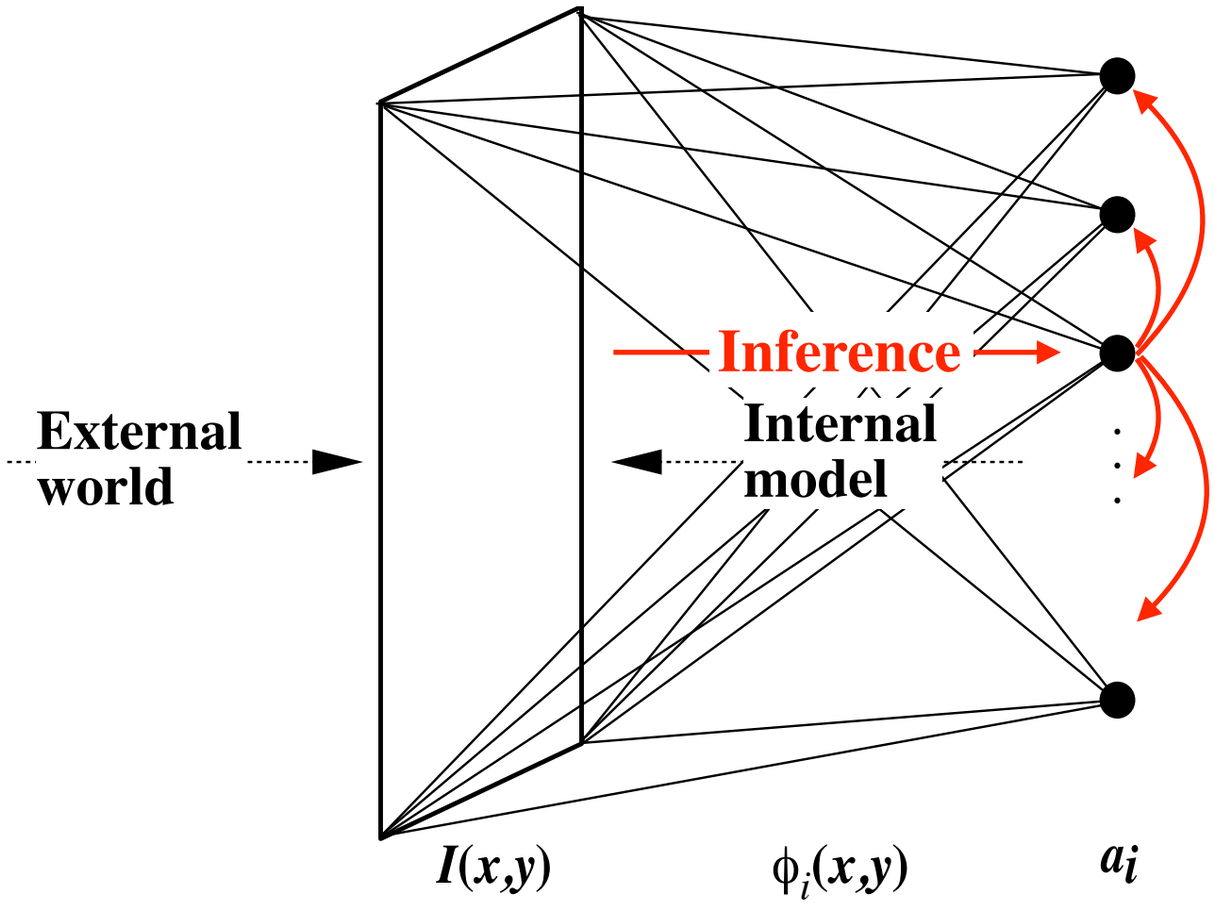
Simple cells ~~sum~~ LGN inputs

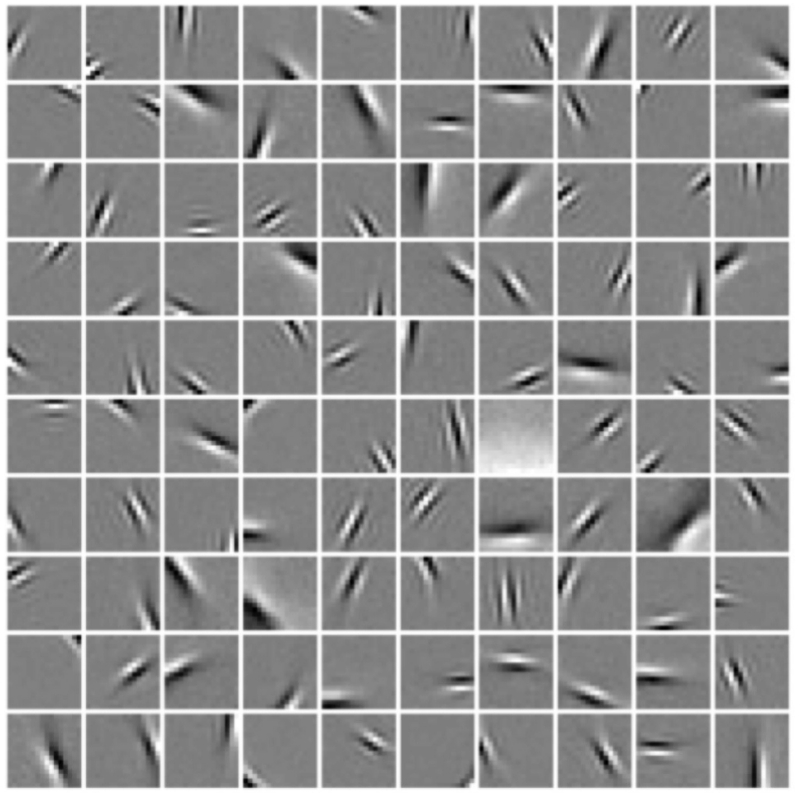


The “standard model” of V1



Sparse coding model





$\phi_i(x, y)$

$$P(\mathbf{a}|\mathbf{I};\mathbf{\Phi}) \propto P(\mathbf{I}|\mathbf{a};\mathbf{\Phi})\ P(\mathbf{a})$$

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} |\mathbf{I} - \Phi\,\mathbf{a}|^2 + \lambda \sum_i C(a_i)$$

Not ~~$\hat{a}_i = g(\sum_{x,y} \phi_i(x,y)\, I(x,y))$~~

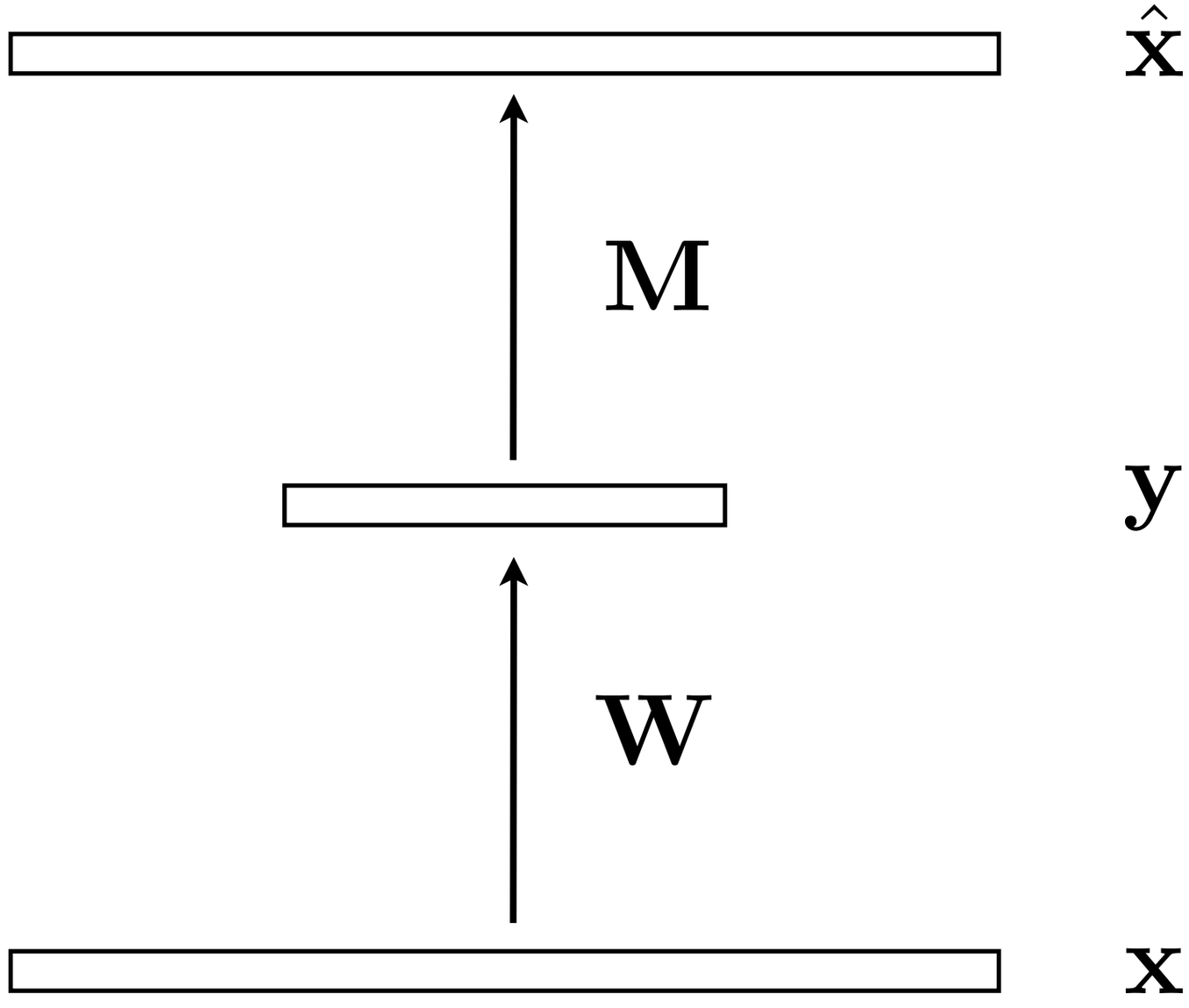
- Cortex is *not* doing PCA
  - PCA only captures pairwise correlations in data.
  - PCA assumes Gaussianity and orthogonal components.
  - Principal components do not resemble simple-cell RFs.

- Cortex is *not* doing compression
  - *expands* dimensionality of representation.
  - goal is to *interpret* image data, not to simply encode or compress it.

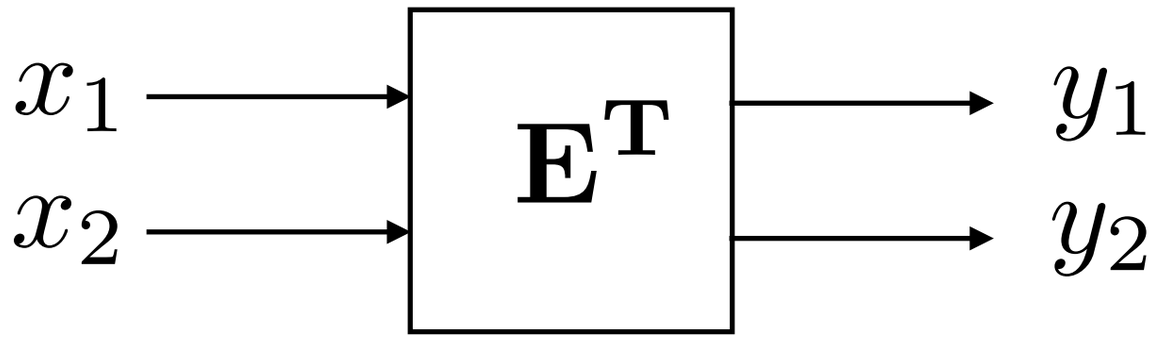
- Cortex seems to be something akin to sparse coding
  - …and probably much more.

Autoencoder networks

$$\min_{\mathbf{W},\mathbf{M}} |\mathbf{x} - \hat{\mathbf{x}}|^2$$



PCA
(Principal Components Analysis)



$$\langle x_1\, x_2 \rangle = c_{12} \neq 0$$

$$y_1 = \mathbf{e}_1 \cdot \mathbf{x}$$

$$y_2 = \mathbf{e}_2 \cdot \mathbf{x}$$

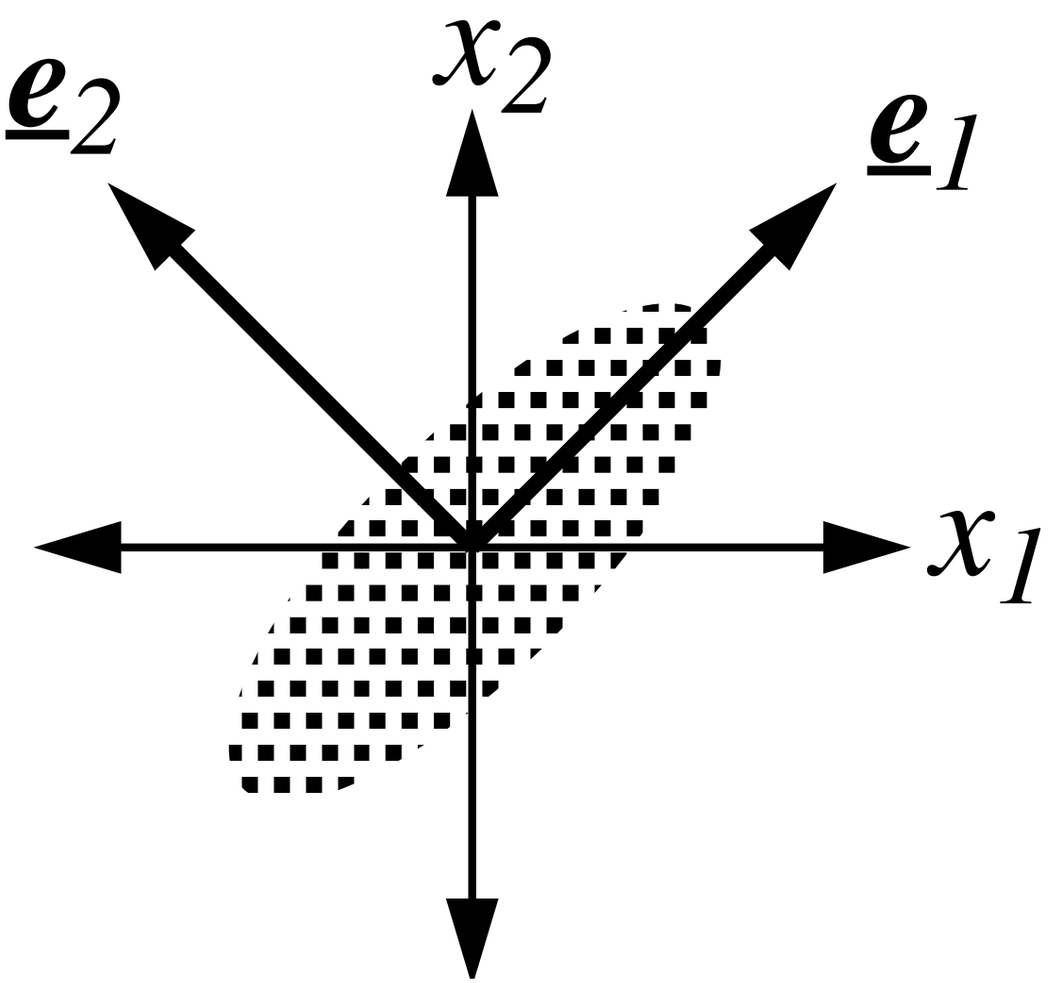
$$\langle y_1\, y_2 \rangle = \langle y_1 \rangle \langle y_2 \rangle = 0$$

$$\mathbf{E} = \begin{bmatrix} | & | \\ \mathbf{e}_1 & \mathbf{e}_2 \\ | & | \end{bmatrix}$$

$$\mathbf{e}_1 \cdot \mathbf{e}_2 = 0$$

$$|\mathbf{e}_1| = |\mathbf{e}_2| = 1$$

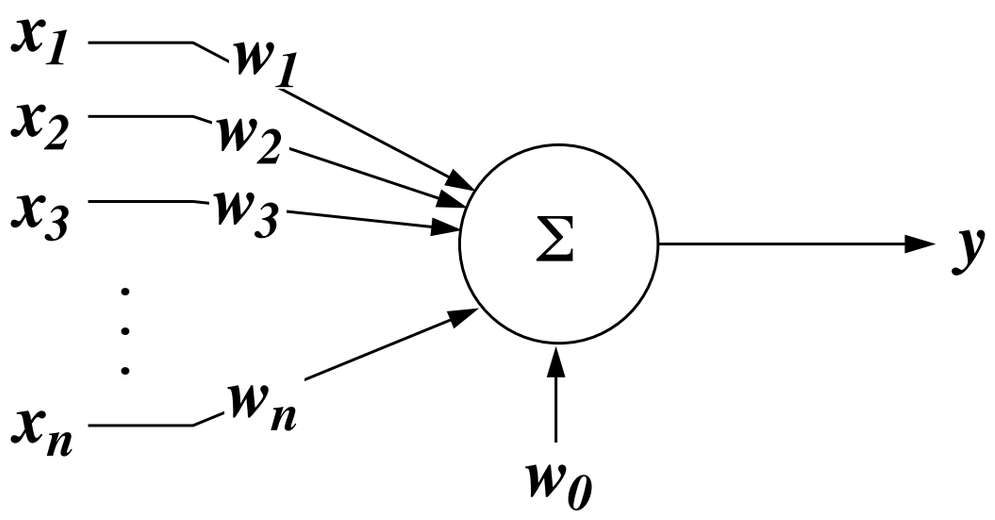
PCA
(Principal Components Analysis)



Hebbian learning

$$\Delta W_{AB} \propto < A\,B >$$

Linear Hebbian learning



$$\dot{w}_i \propto \langle y\, x_i \rangle$$

$$y = \sum_j w_j\, x_j$$

$$\dot{w}_i \propto \left\langle \sum_j w_j\, x_j\; x_i \right\rangle$$

$$= \sum_j w_j \langle\, x_j\; x_i \rangle$$

$$\boxed{\dot{\mathbf{w}} \propto \mathbf{C}\,\mathbf{w}} \qquad C_{ij} = \langle x_i\, x_j \rangle$$

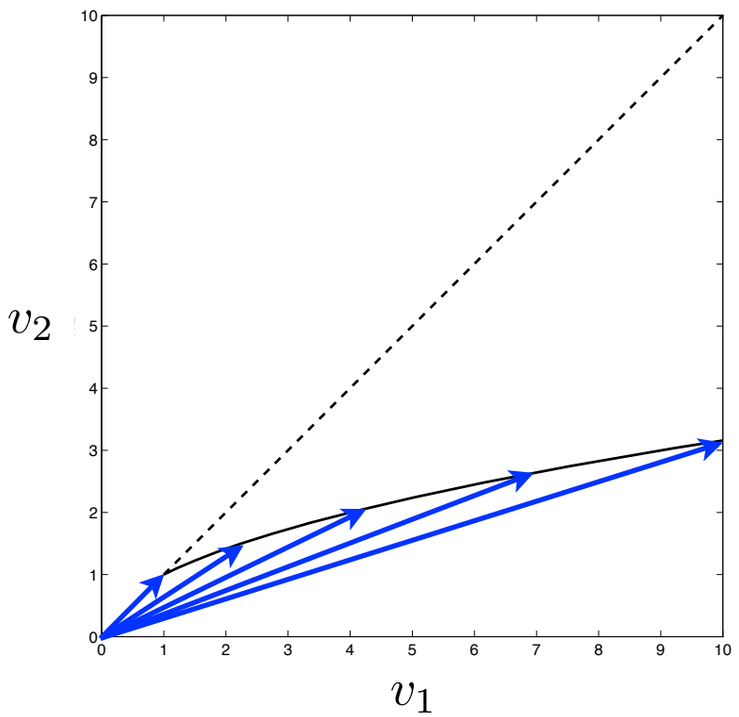
$$\mathbf{C} = \mathbf{E}\,\mathbf{\Lambda}\,\mathbf{E}^T$$

$$\mathbf{E} = \begin{bmatrix} | & | & & | \\ \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \\ | & | & & | \end{bmatrix} \qquad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}$$

define: $\mathbf{v} = \mathbf{E}^{\mathbf{T}}\,\mathbf{w}$

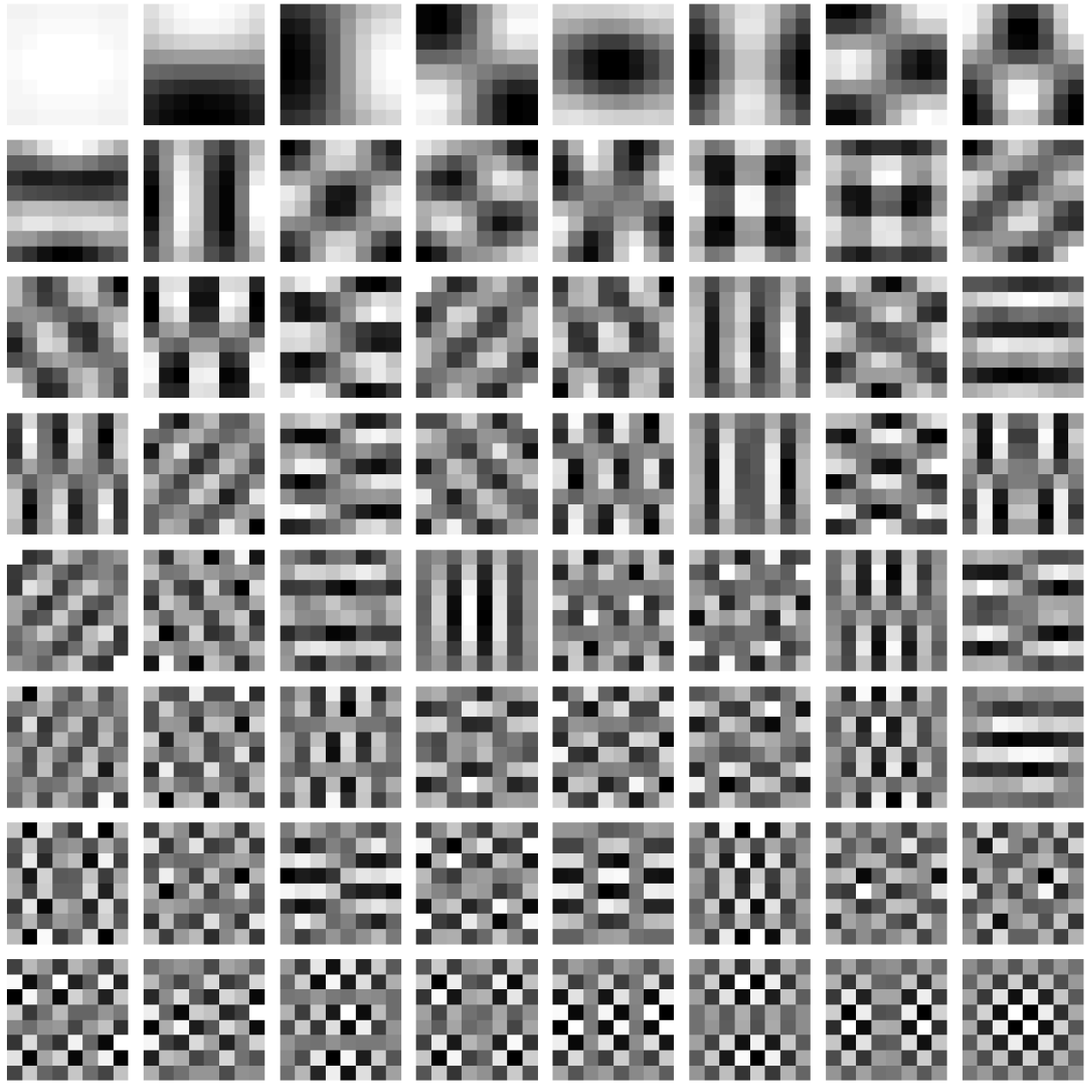
$$
\begin{aligned}
\dot{\mathbf{w}} &\propto \mathbf{C}\,\mathbf{w} \\
&= \mathbf{E}\,\mathbf{\Lambda}\,\mathbf{E}^{T}\,\mathbf{w} \\
&= \mathbf{E}\,\mathbf{\Lambda}\,\mathbf{v} \\
\mathbf{E}^{T}\,\dot{\mathbf{w}} &\propto \mathbf{E}^{T}\,\mathbf{E}\,\mathbf{\Lambda}\,\mathbf{v} \\
\dot{\mathbf{v}} &\propto \mathbf{\Lambda}\,\mathbf{v}
\end{aligned}
$$

$$
\begin{aligned}
\dot{v}_1 &\propto \lambda_1\,v_1 \\
\dot{v}_2 &\propto \lambda_2\,v_2
\end{aligned}
$$



Principal components of natural image patches (8 x 8 pixels)

$$\mathbf{W} = \mathbf{E}^T$$

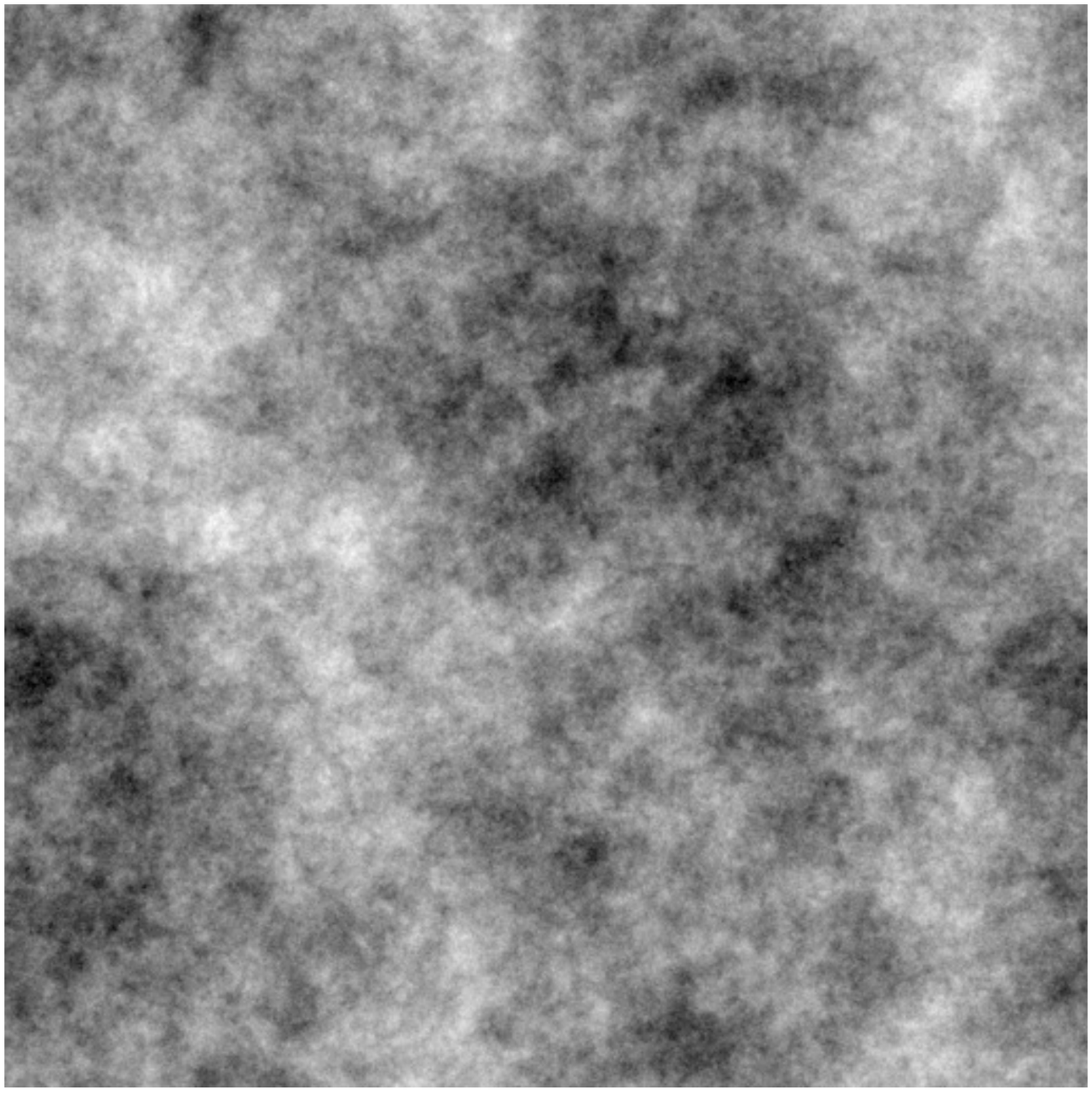


- Not localized
- Not oriented

PCA is incapable of learning about localized, oriented structure in images.

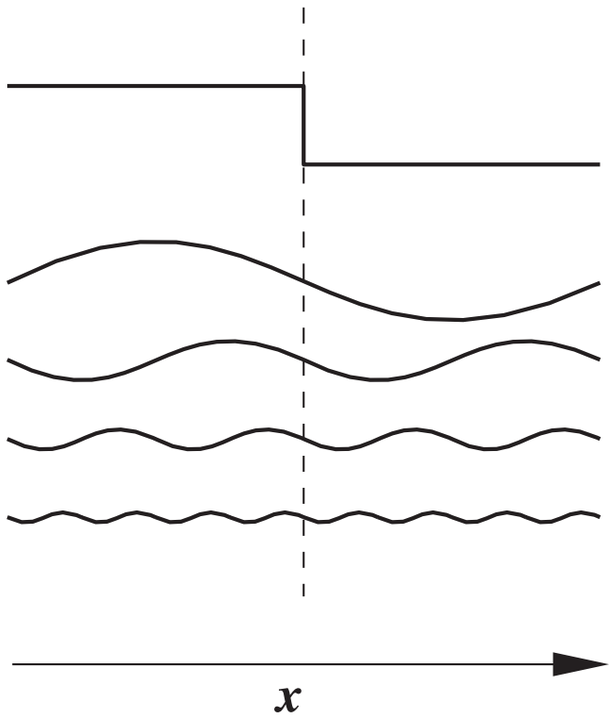
1/*f* noise

(what the world looks like if all you care about are pairwise correlations)

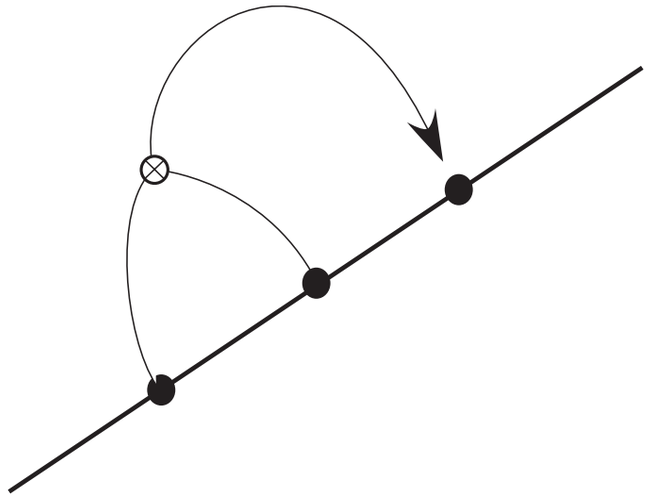


Higher-order image statistics

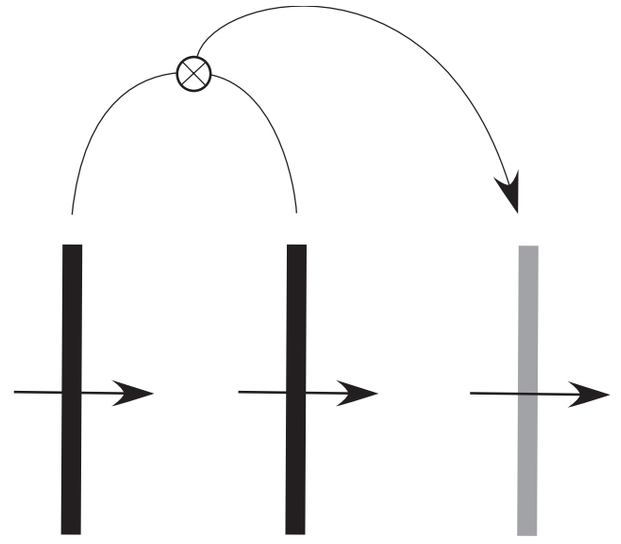
phase alignment



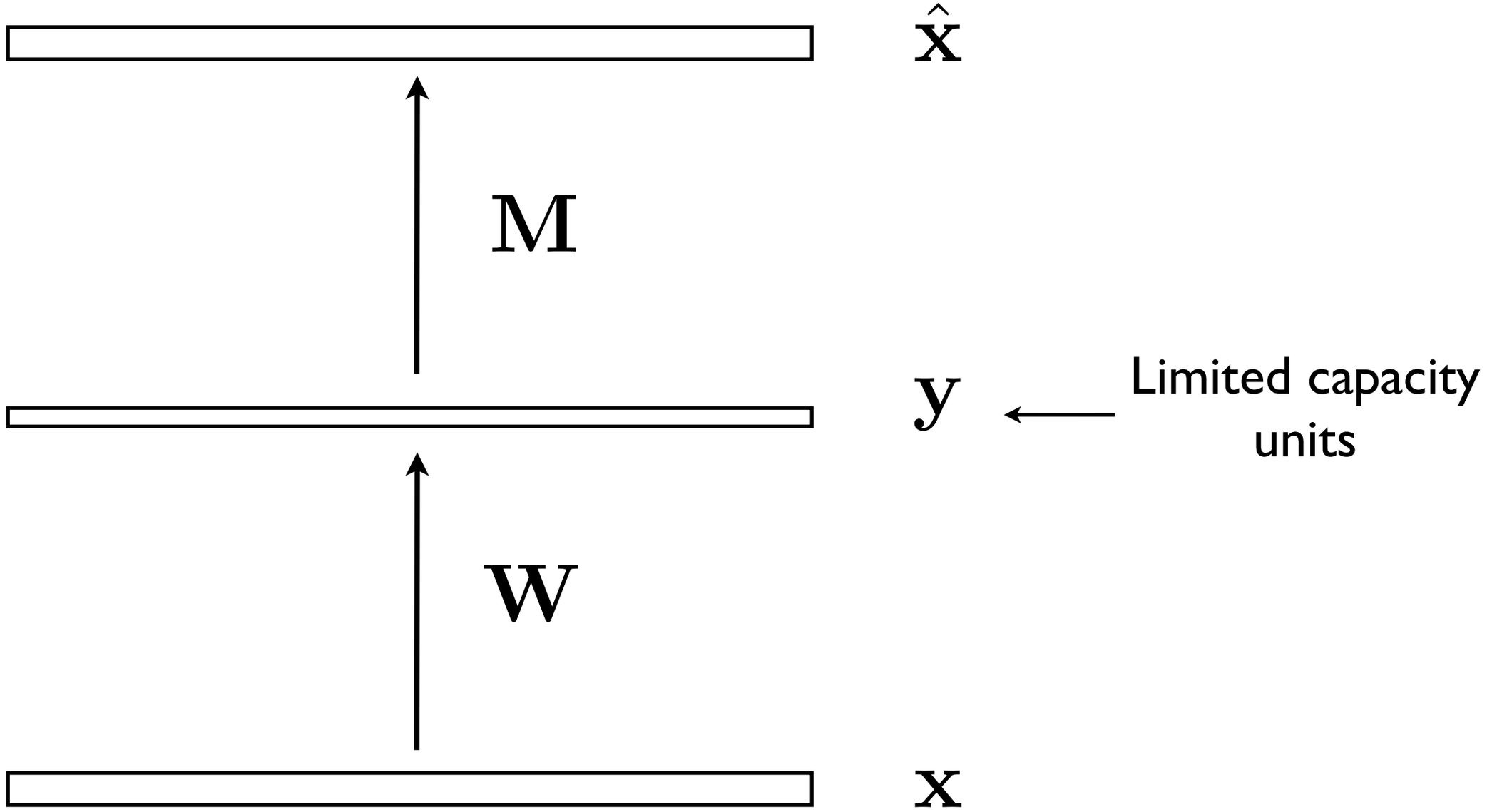
orientation



motion

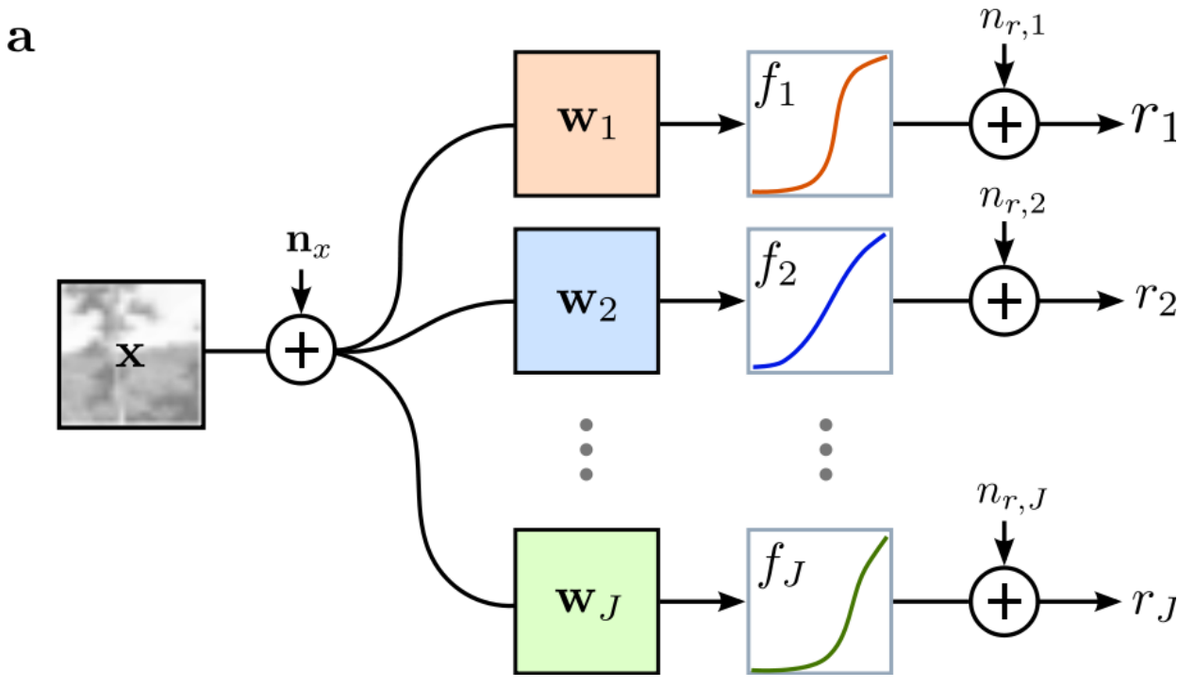


Bottleneck may also be in the form of limited capacity units.
Optimal strategy in this case is to whiten.



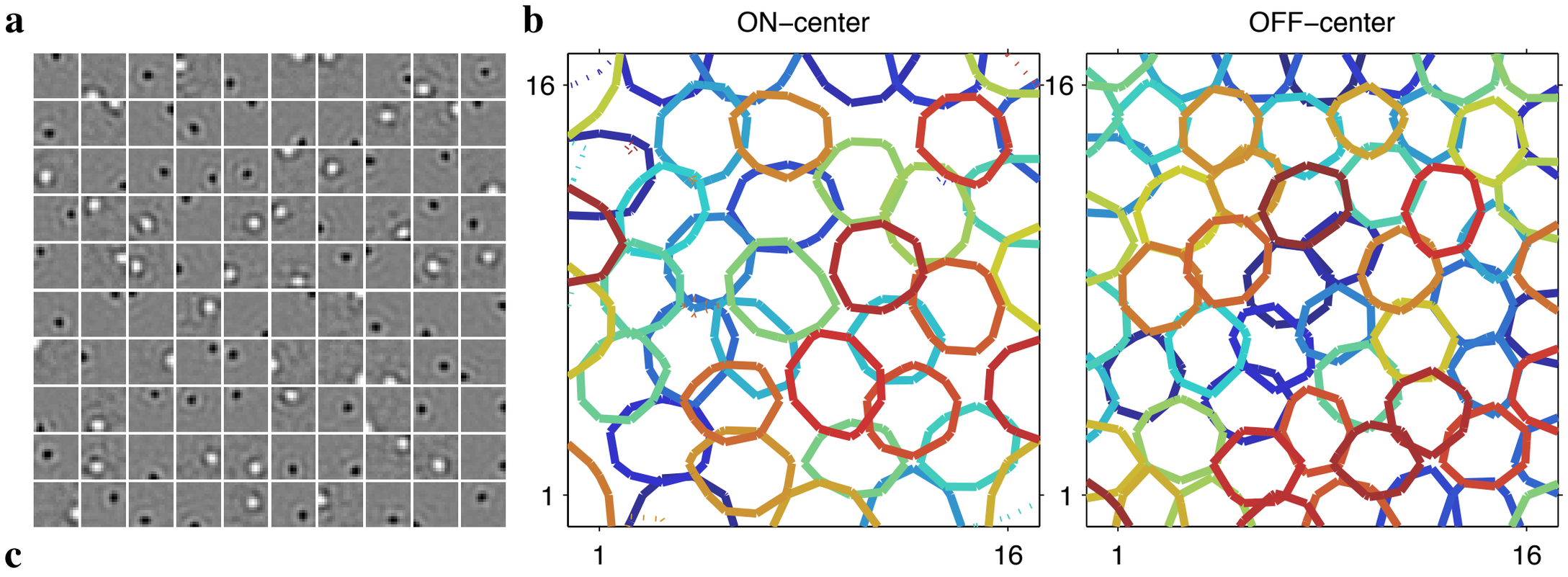
Efficient coding model of retina

(Karklin & Simoncelli 2012)

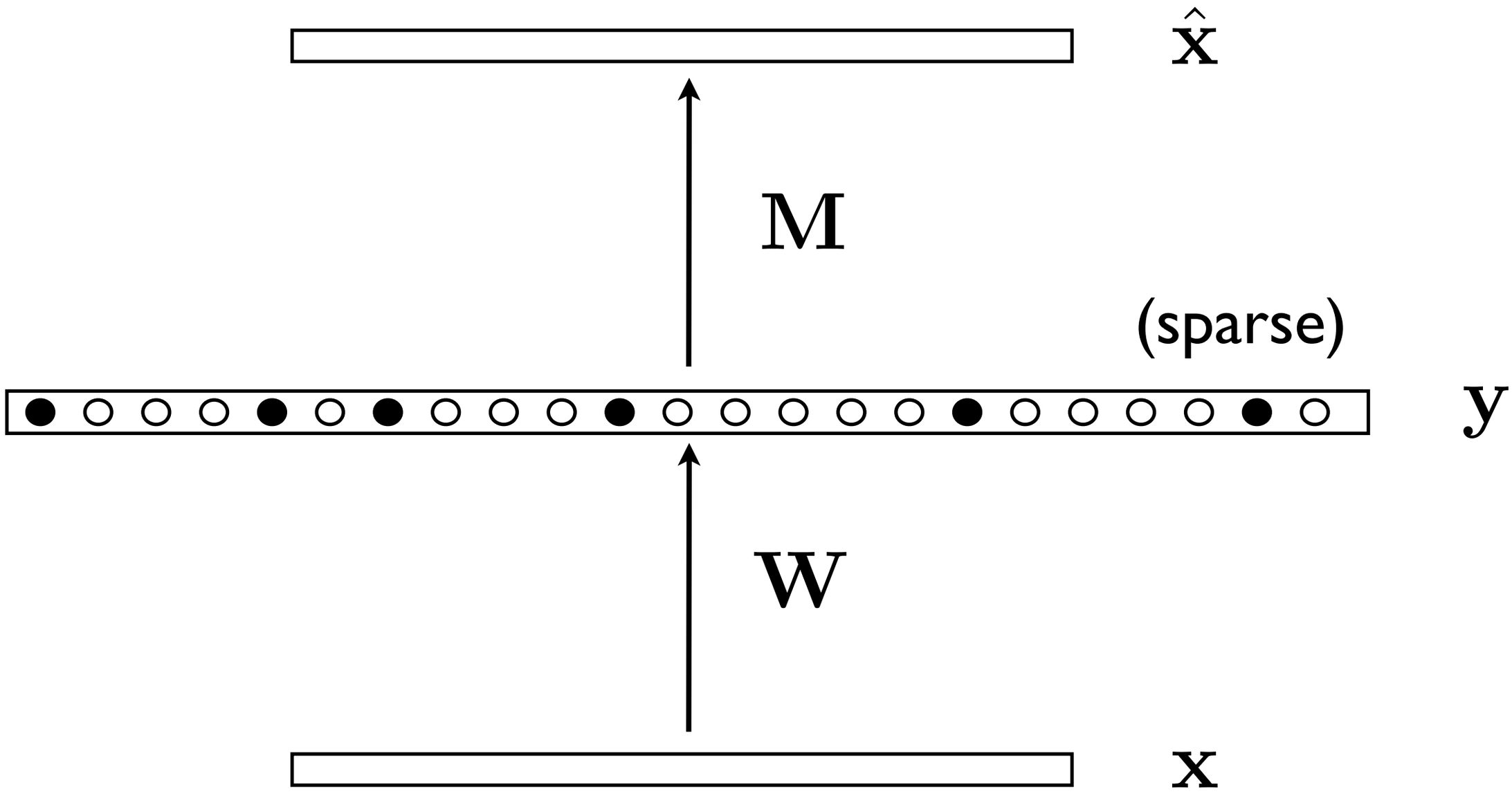


Objective function:

$$I(X;R) - \sum_j \lambda_j \langle r_j \rangle$$



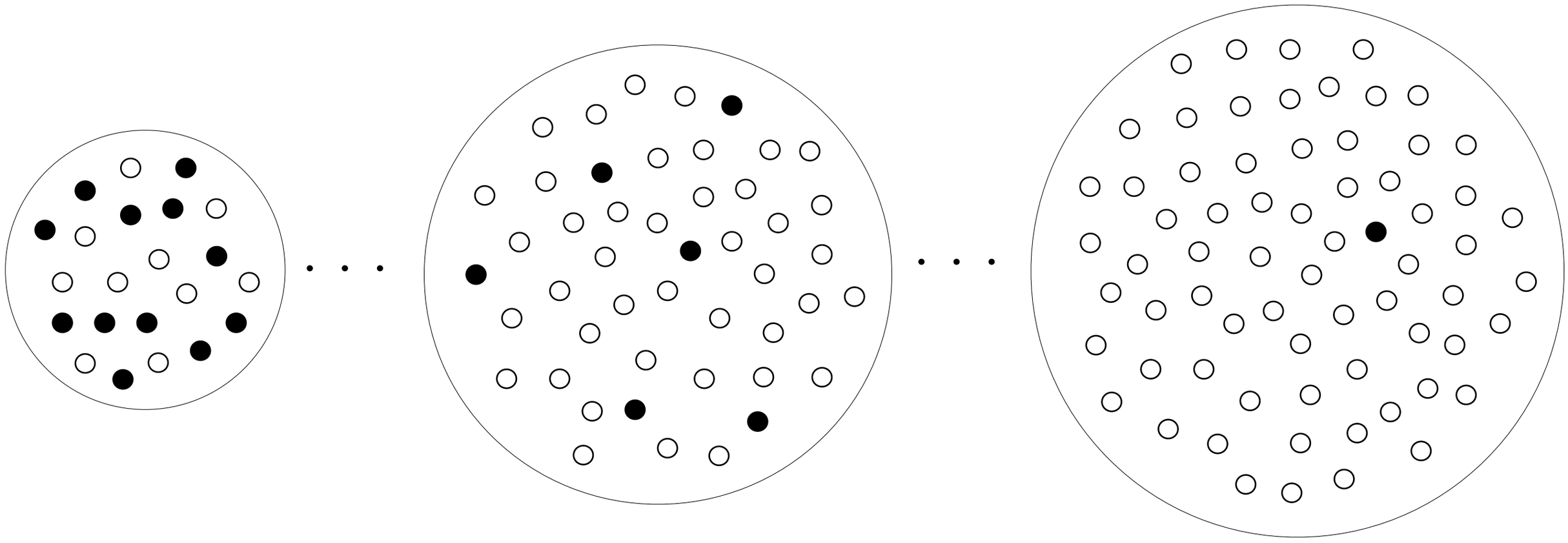
Sparse codes impose a different type of bottleneck by limiting the number of active units



Dense codes
(e.g., ascii)

Sparse, distributed codes

Local codes
(e.g., grandmother cells)



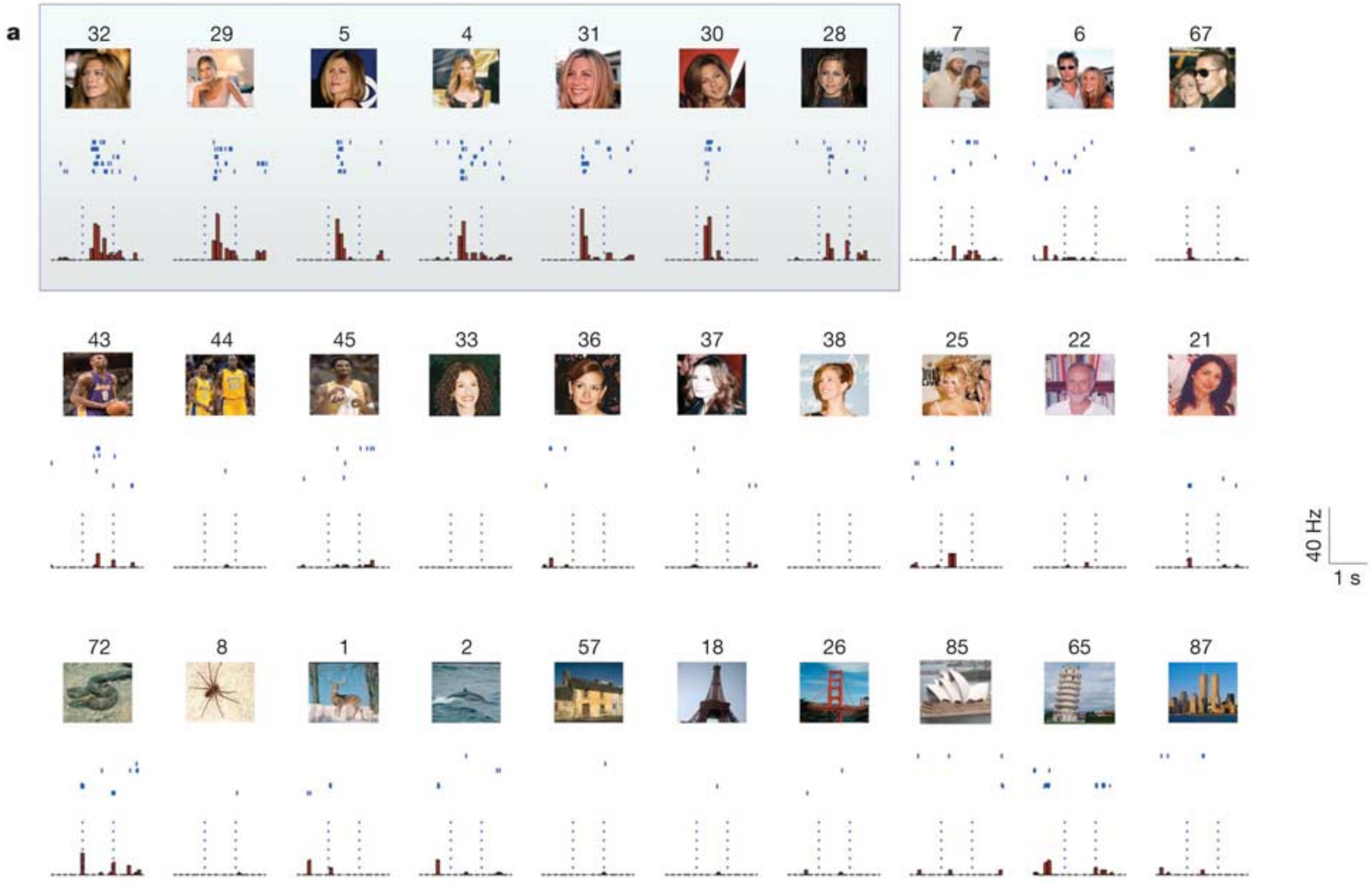
2^N

$\binom{N}{K}$

N

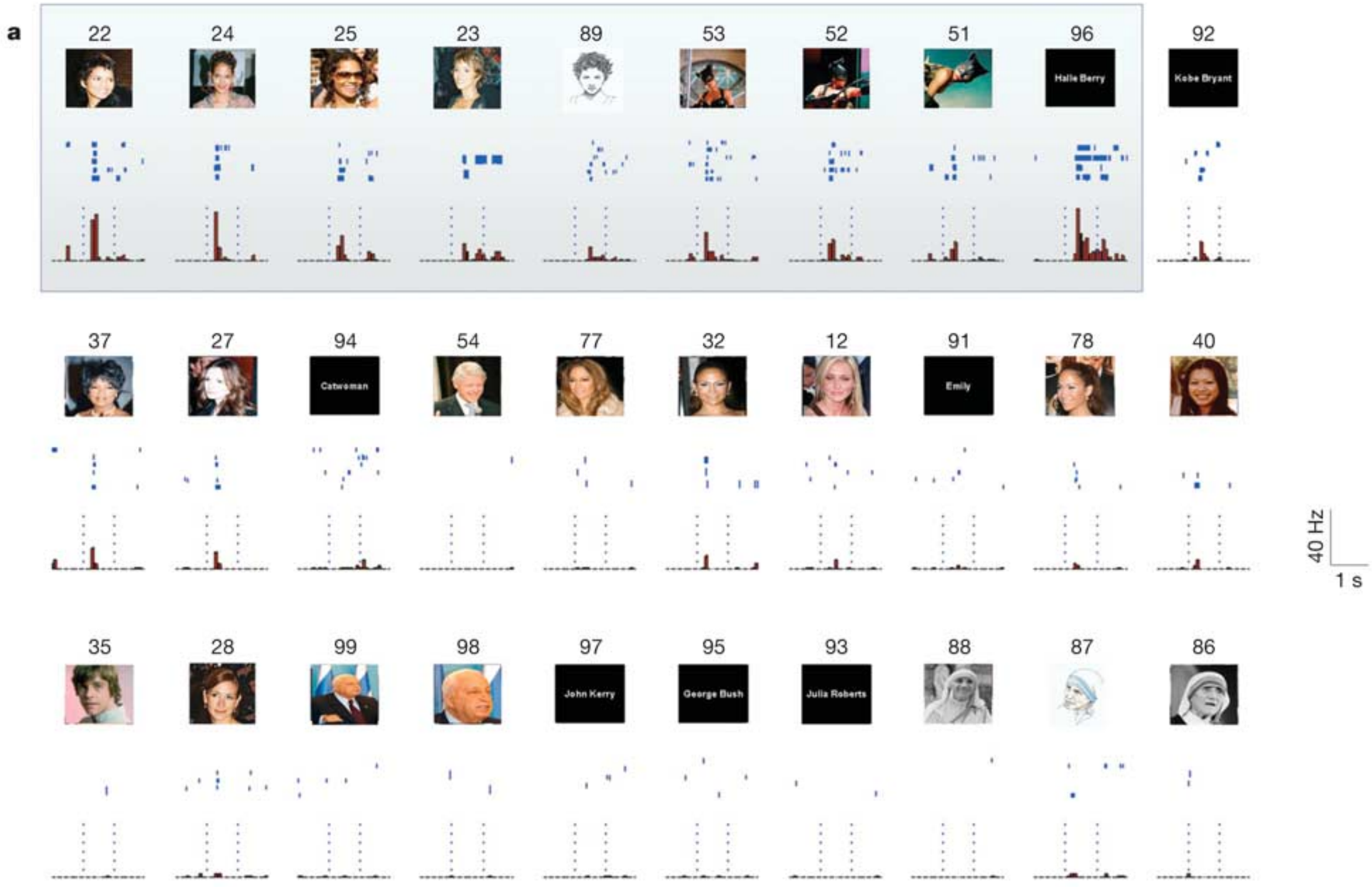
Evidence for grandmother cells?

(Quiroga, Reddy, Kreiman, Koch & Fried, *Nature* 2005)



Evidence for grandmother cells?

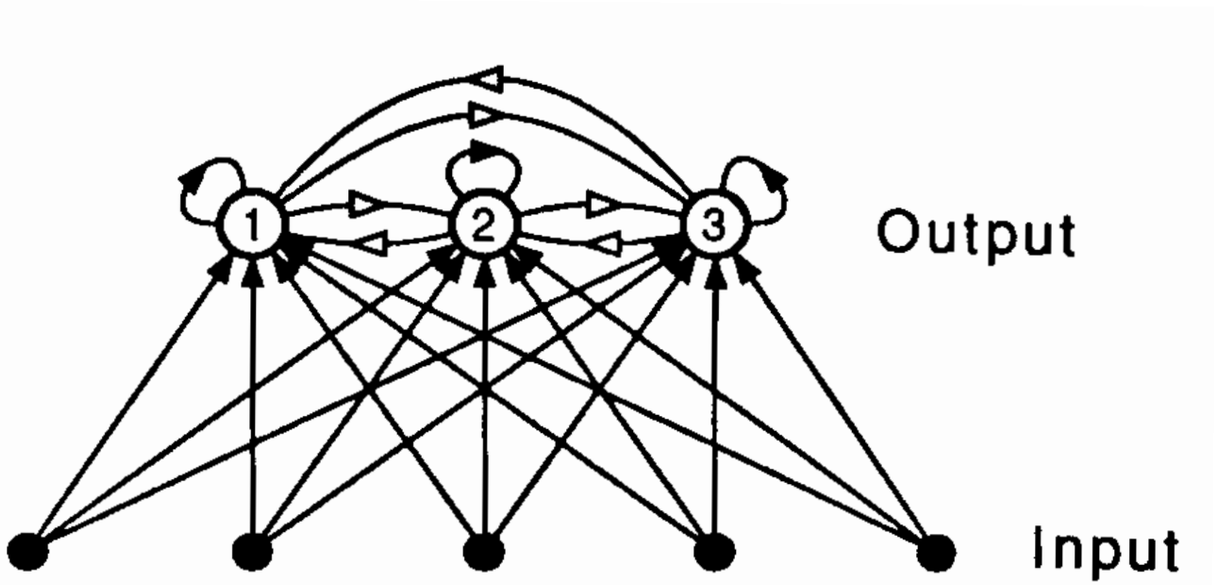
(Quiroga, Reddy, Kreiman, Koch & Fried, *Nature* 2005)



Evidence for grandmother cells?

(Quiroga, Reddy, Kreiman, Koch & Fried, *Nature* 2005)

Winner-take-all learning



Network:

$$u_i = \sum_j w_{ij}\, x_j$$

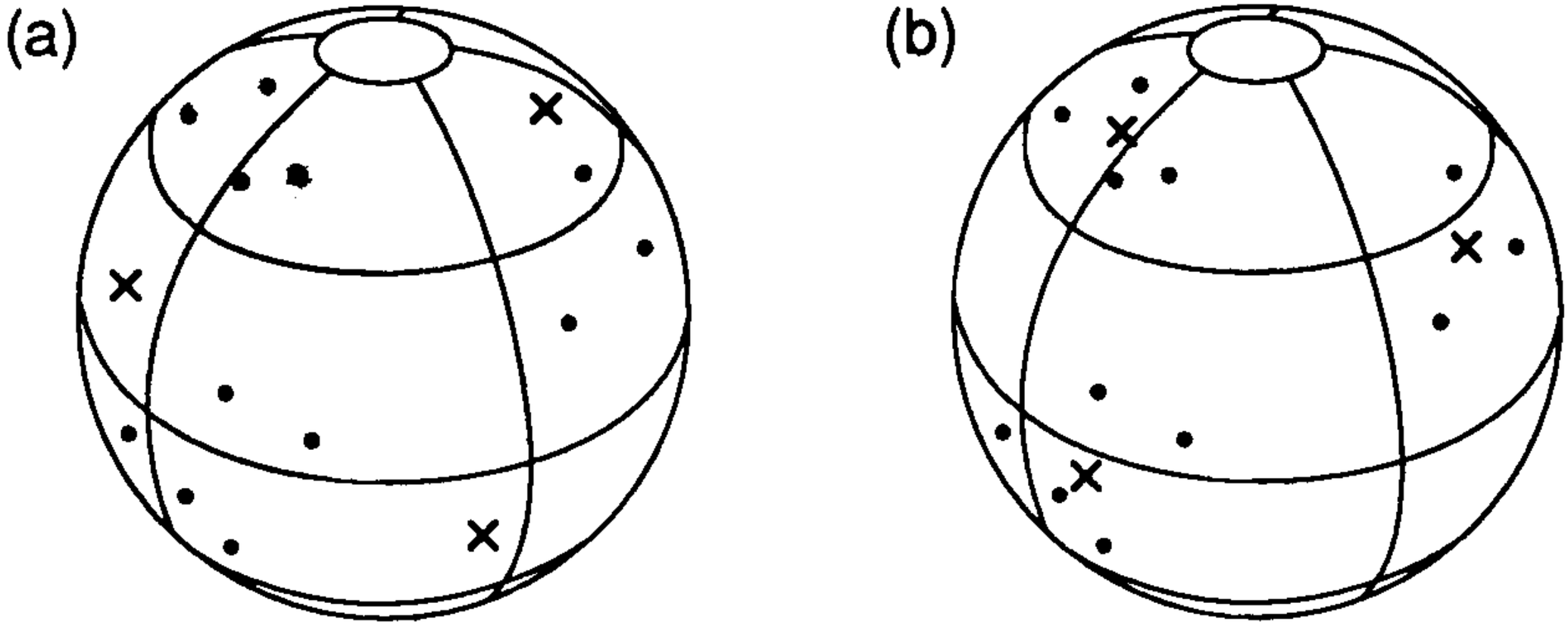
$$y_i = \begin{cases} 1 & u_i > u_j \;\; \forall j \neq i \\ 0 & \text{otherwise} \end{cases}$$

Learning rule: $$\Delta w_{ij} = \eta\, y_i\, (x_j - w_{ij})$$

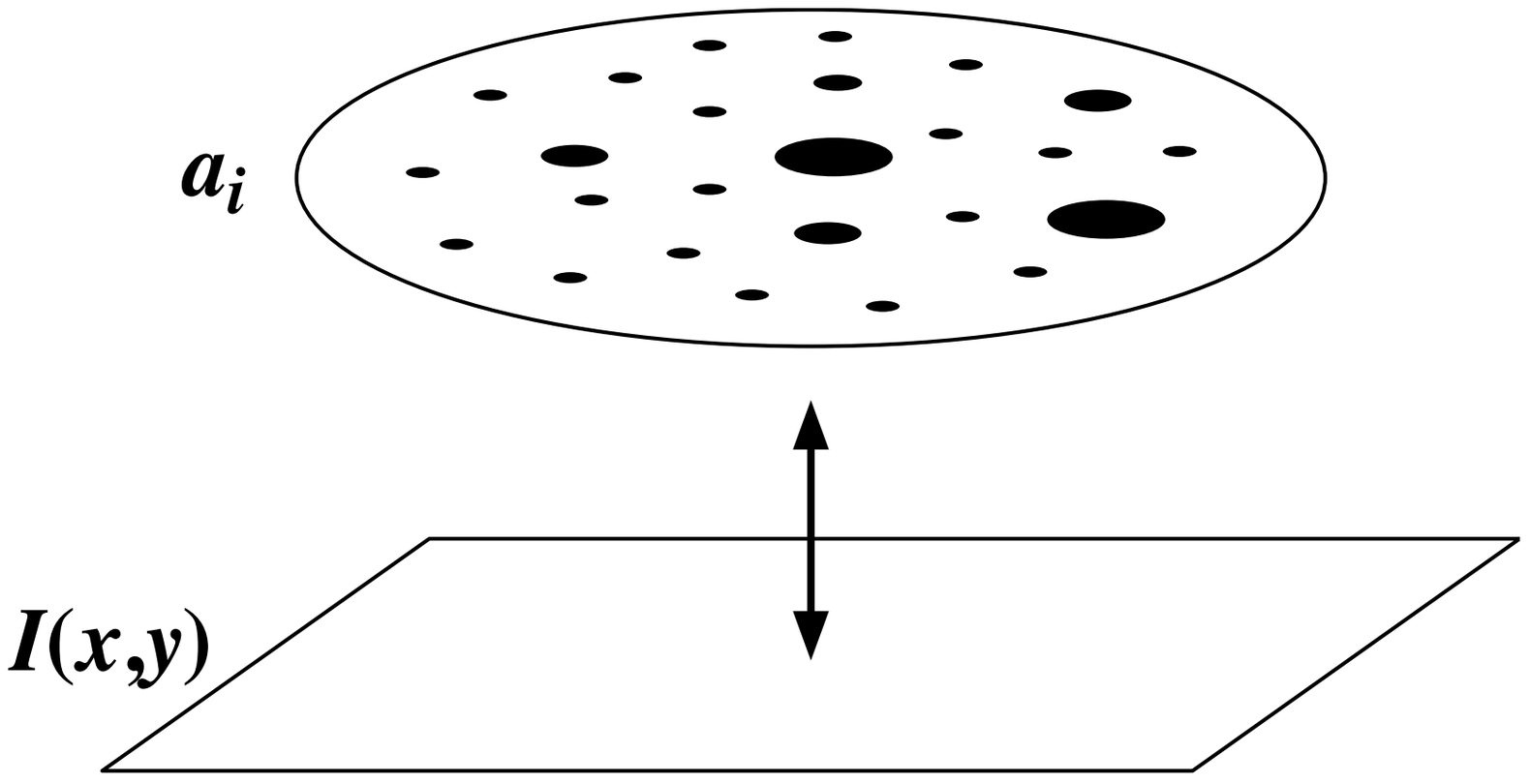
Winner-take-all learning

before learning

after learning



Sparse, *distributed* representation



Biol. Cybern. 64, 165–170 (1990)

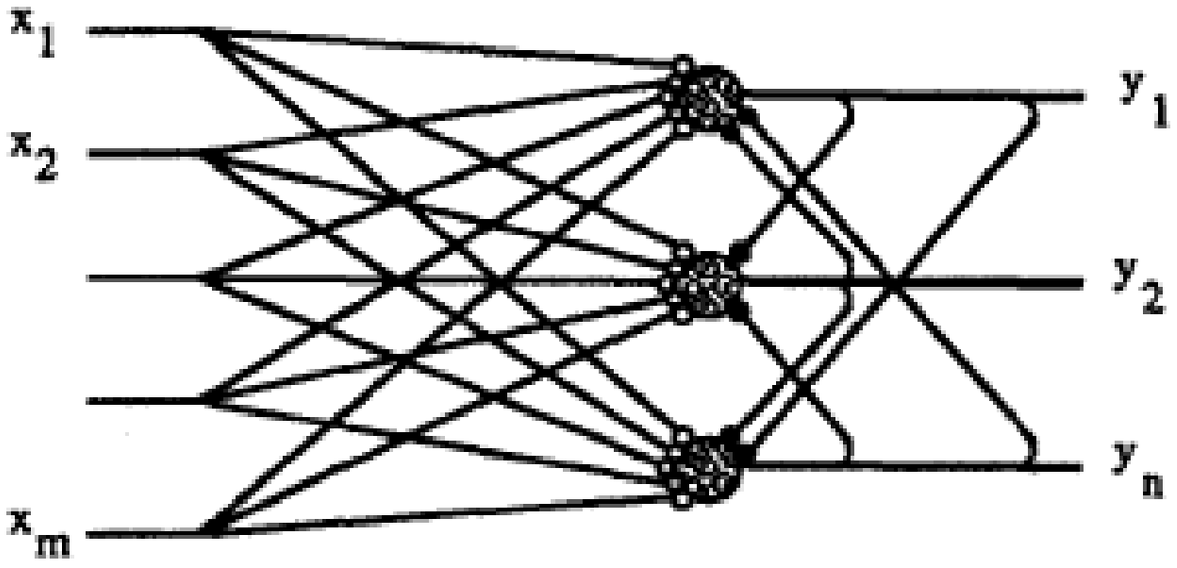
Biological Cybernetics

Forming sparse representations by local anti-Hebbian learning

P. Földiák

Physiological Laboratory, University of Cambridge, Downing Street, Cambridge CB2 3EG, United Kingdom

$$\frac{dy_i^*}{dt} = f\left(\sum_{j=1}^{m} q_{ij}x_j + \sum_{j=1}^{n} w_{ij}y_j^* - t_i\right) - y_i^*$$



anti-Hebbian rule–

$$\Delta w_{ij} = -\alpha(y_i y_j - p^2)$$

(if $i = j$ or $w_{ij} > 0$ then $w_{ij} := 0$)

Hebbian rule–

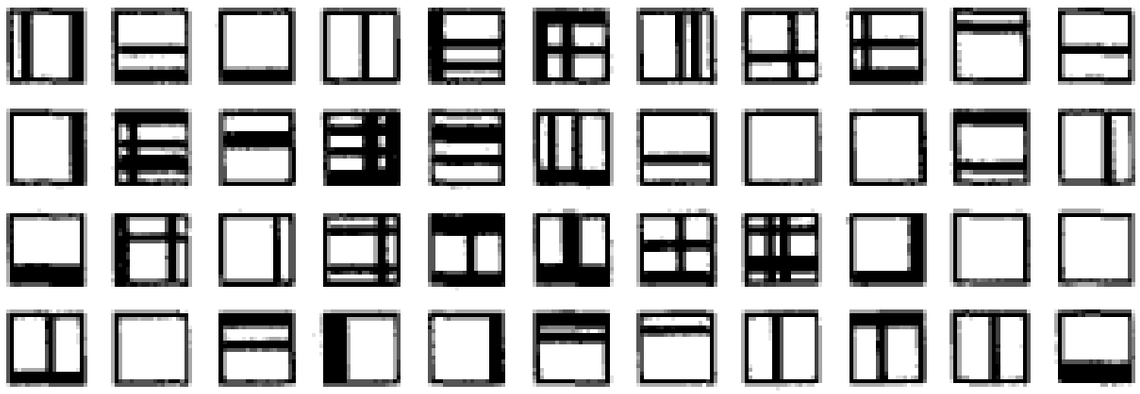
$$\Delta q_{ij} = \beta y_i (x_j - q_{ij})$$

threshold modification–

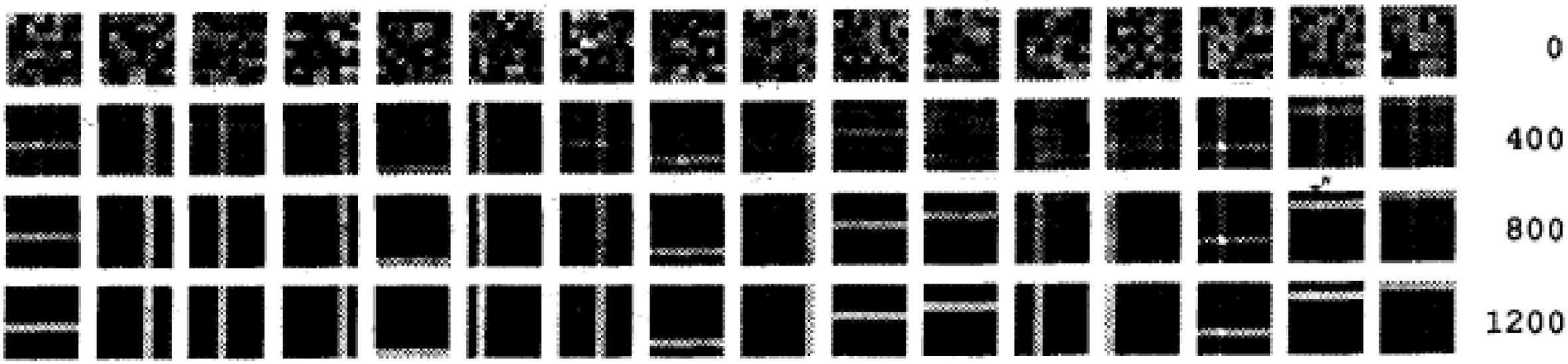
$$\Delta t_i = \gamma(y_i - p)\,.$$

Learning lines

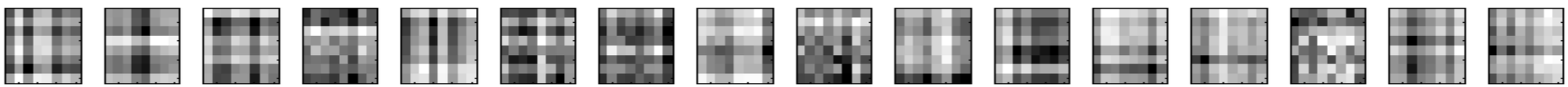
Input patterns:



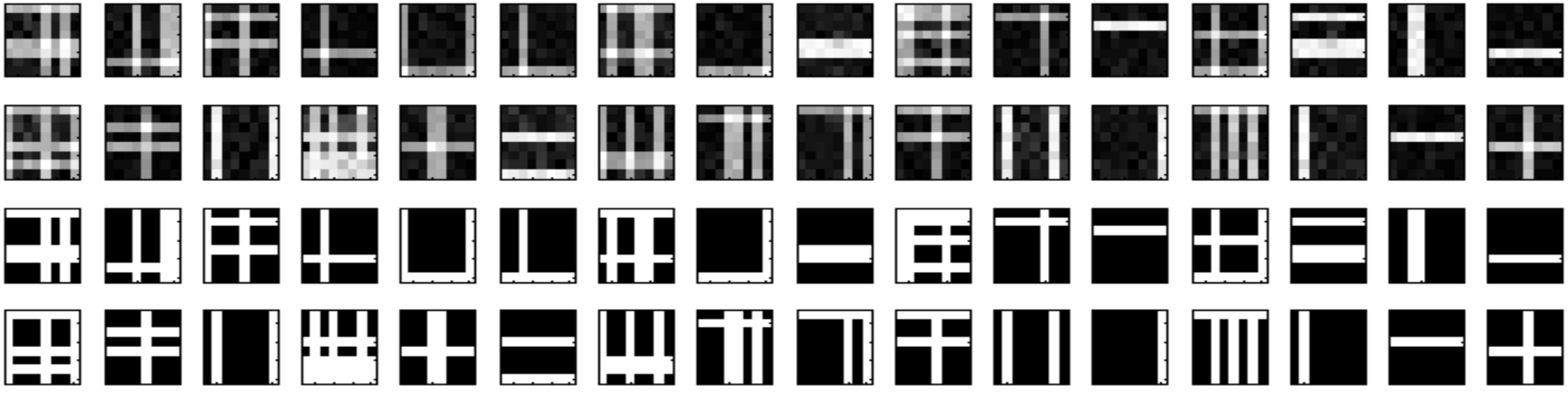
Learned weights:



PCA solution



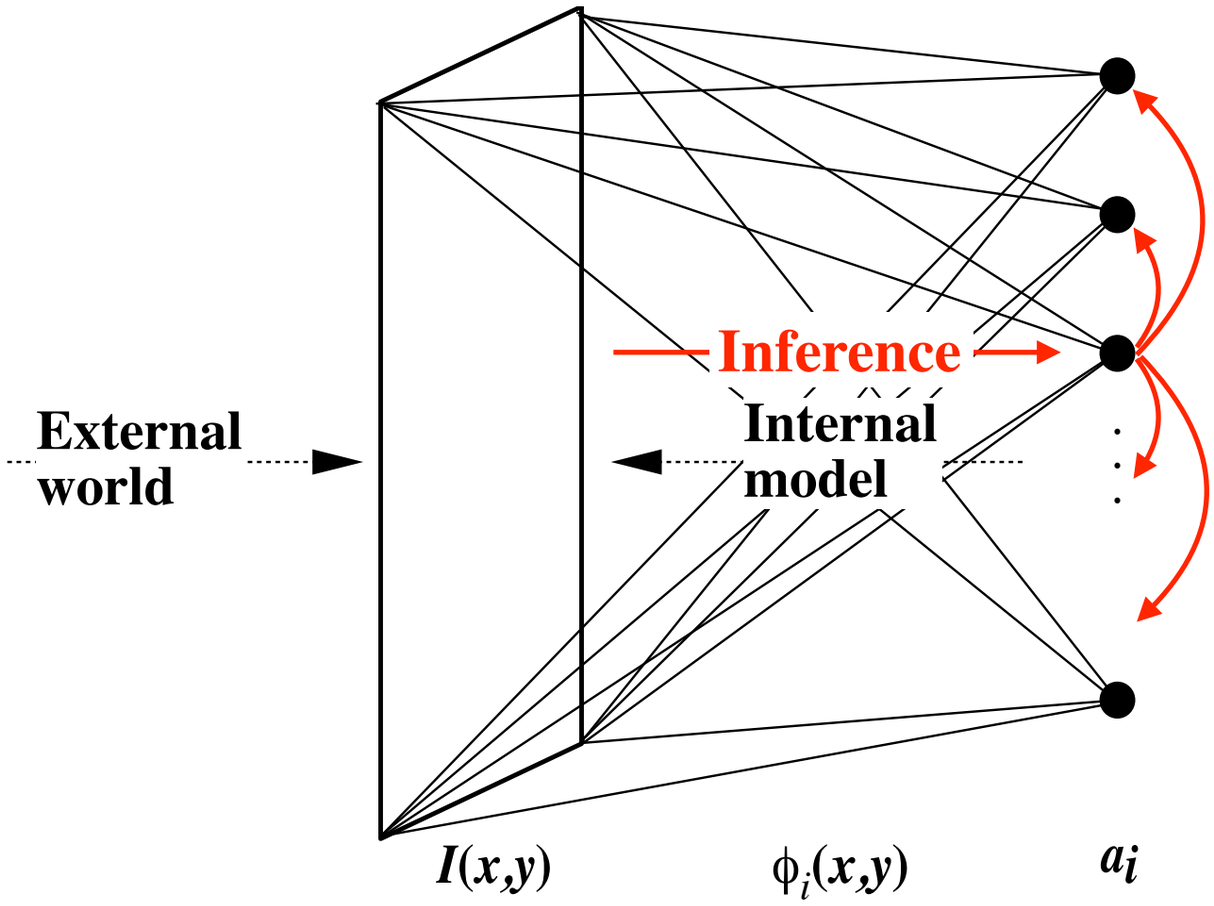
Reconstructions

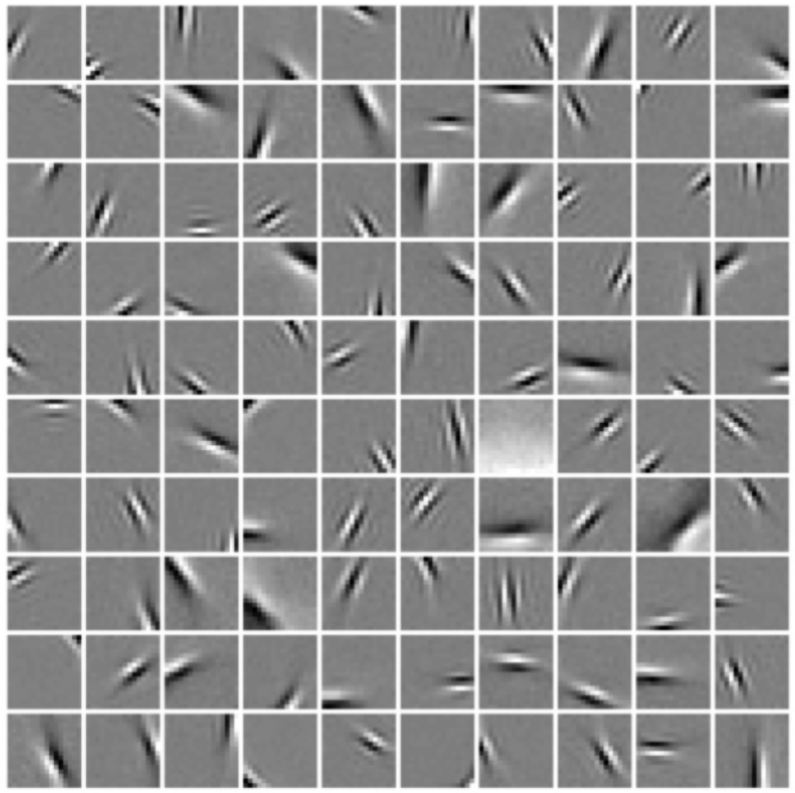


Problems

- How to deal with graded input signals? (i.e., real images)
- No objective function

Sparse coding model
(Olshausen & Field 1996, 1997)





$\phi_i(x, y)$

$$I(x,y) = \sum_i a_i \, \phi_i(x,y) + \epsilon(x,y)$$

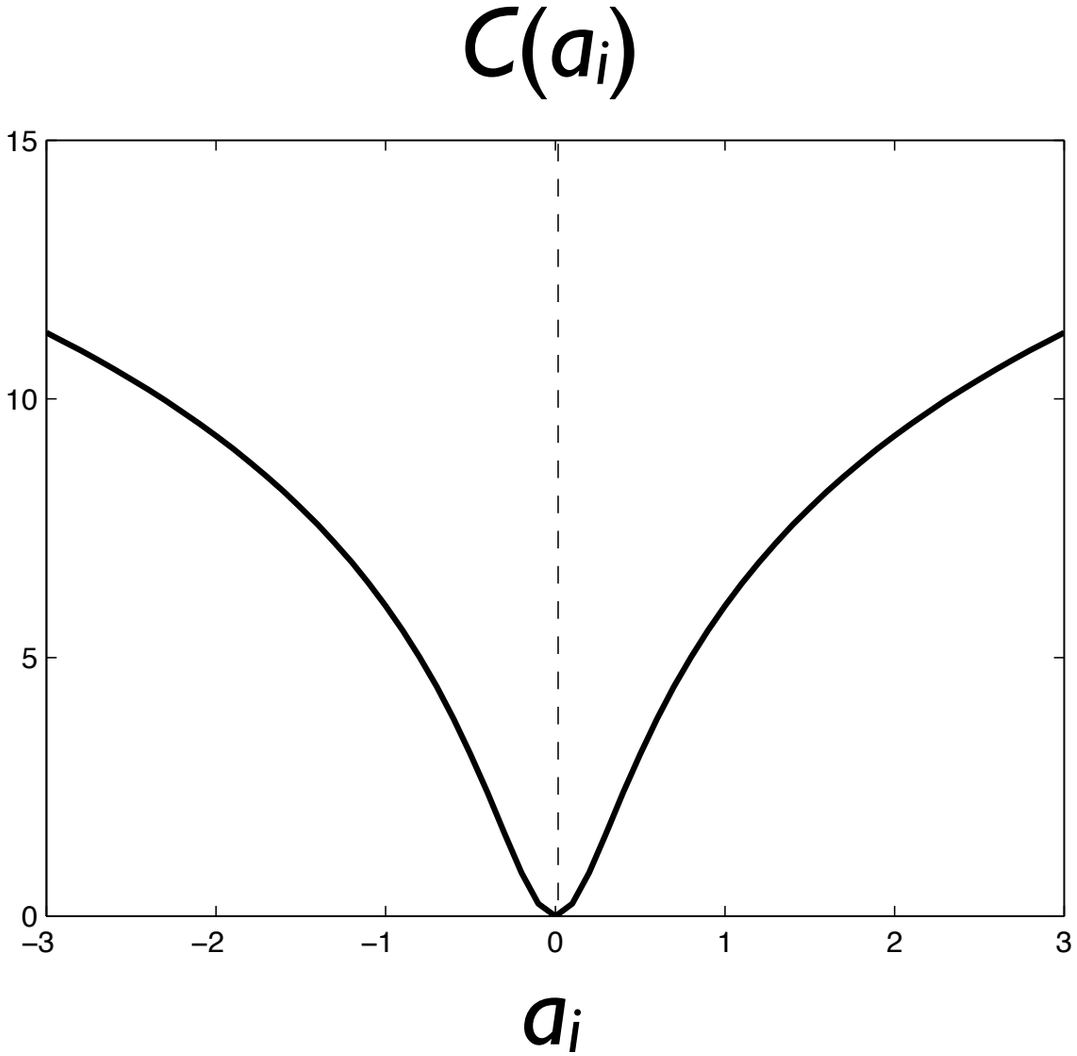
$$P(\mathbf{a}|\mathbf{I};\mathbf{\Phi}) \;\propto\; P(\mathbf{I}|\mathbf{a};\mathbf{\Phi})\; P(\mathbf{a})$$

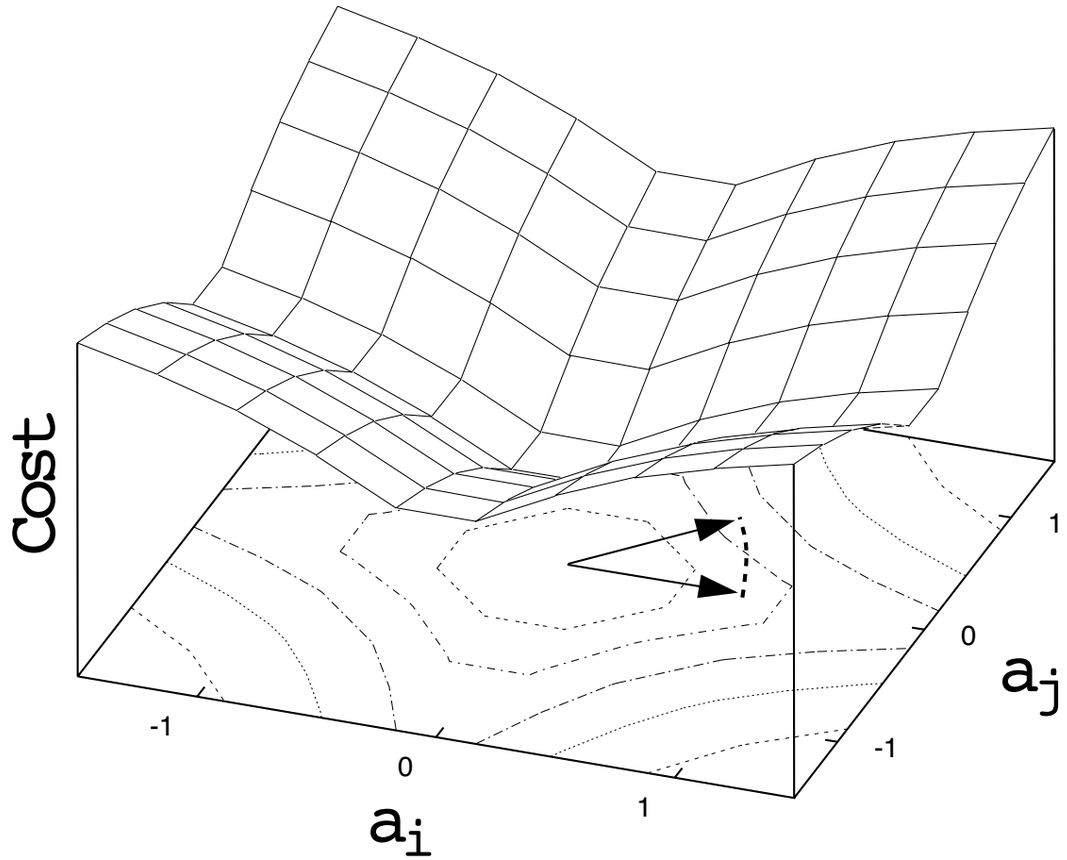
$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \; |\mathbf{I} - \Phi\,\mathbf{a}|^2 + \lambda \sum_i C(a_i)$$

$$P(\mathbf{a}|\mathbf{I};\mathbf{\Phi}) \;\propto\; P(\mathbf{I}|\mathbf{a};\mathbf{\Phi})\; P(\mathbf{a})$$

Cost function

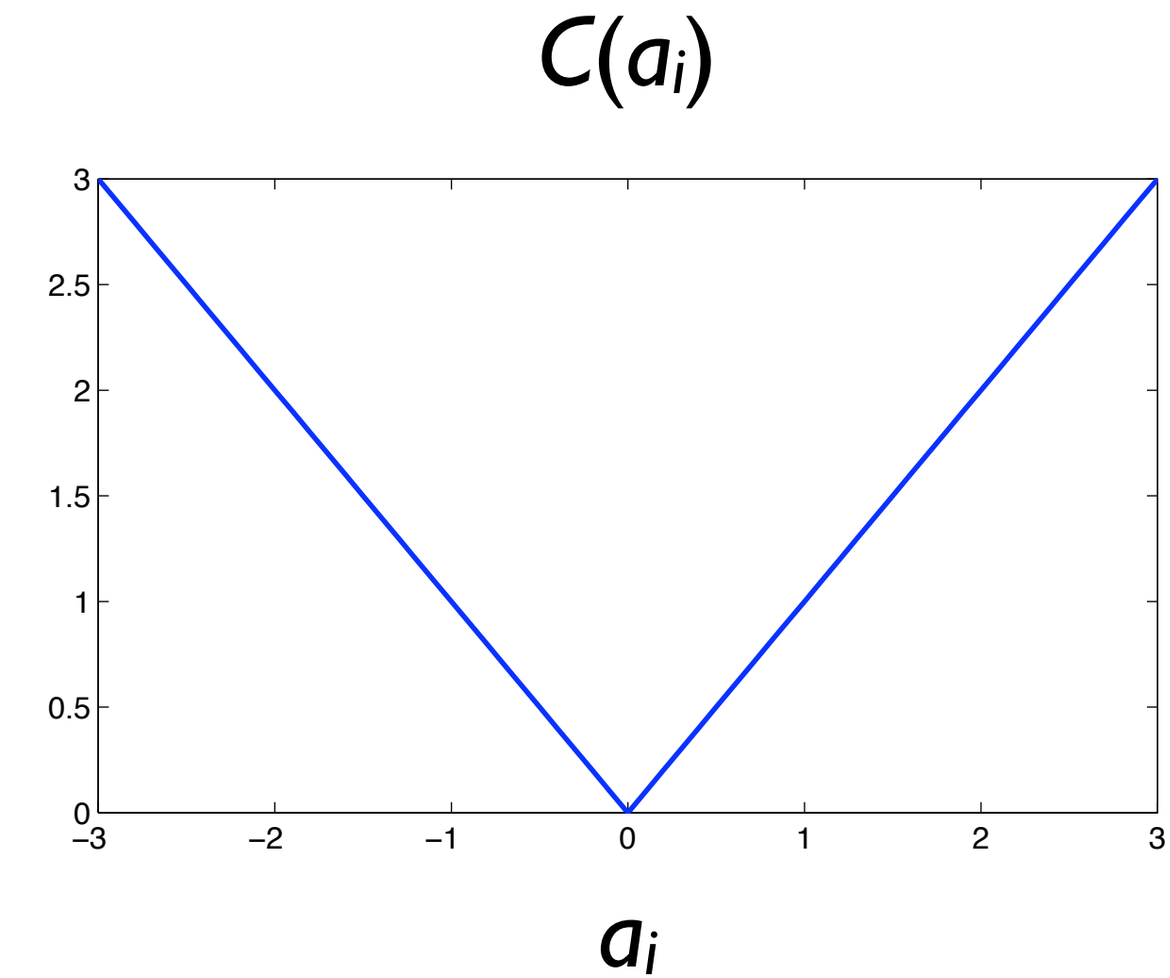
$$C(a_i) = \log(1 + a_i^2)$$

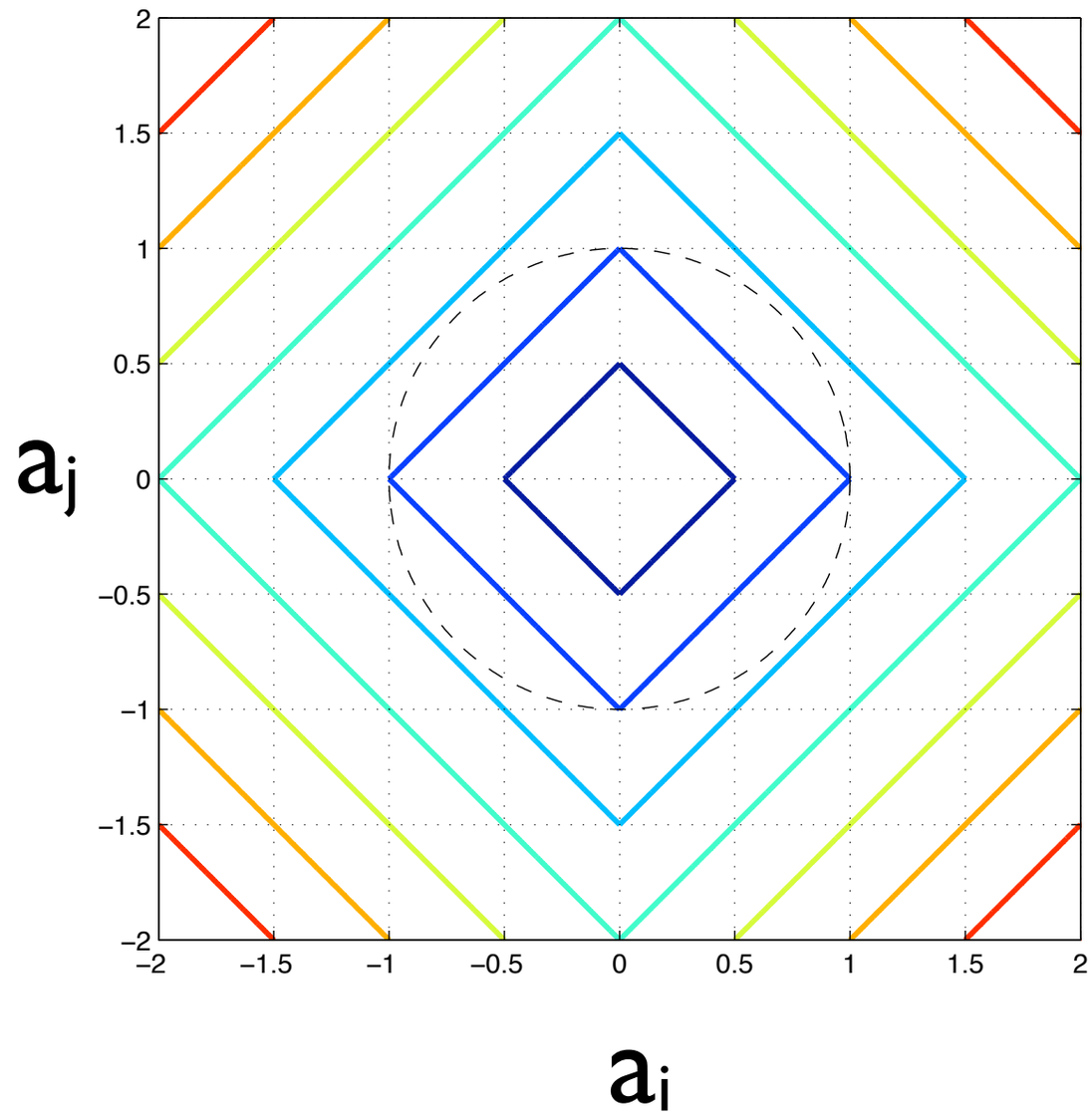




Cost function

$$C(a_i) = |a_i|$$





Compute coefficients via gradient descent

$$\begin{aligned}
\tau\,\dot{a}_i &= -\frac{dE}{da_i} \\
&= b_i - \sum_{j \neq i} G_{ij}\, a_j - f_\lambda(a_i)
\end{aligned}$$

Where

$$b_i = \sum_{x,y} \phi_i(x,y)\, I(x,y)$$

$$G_{ij} = \sum_{x,y} \phi_i(x,y)\, \phi_j(x,y)$$

$$f_\lambda(a_i) = a_i + \lambda\, C'(a_i)$$

Alternative formulation (the Hopfield trick)

Let

$$u_i = f_\lambda(a_i), \quad \text{or} \quad a_i = f_\lambda^{-1}(u_i) \equiv \; g(u_i)$$

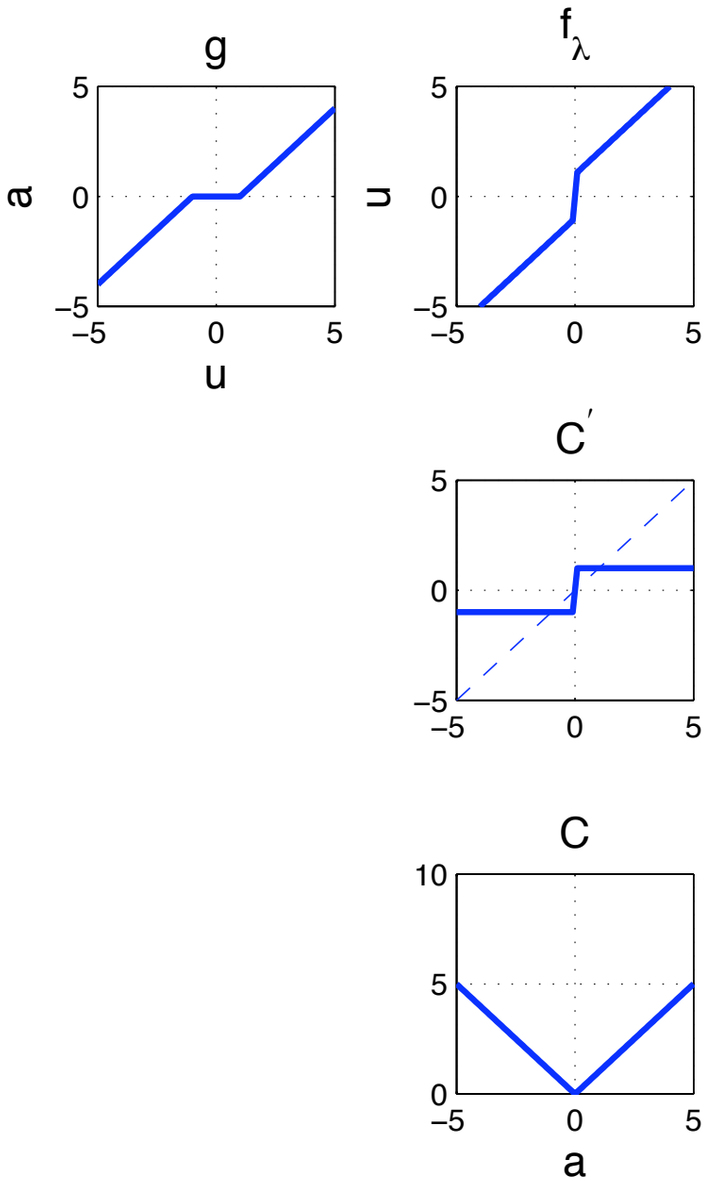
$$\begin{aligned} \tau\, \dot{u}_i &= -\frac{dE}{da_i} \\ &= b_i - \sum_{j \neq i} G_{ij}\, a_j - u_i \end{aligned}$$

Thus

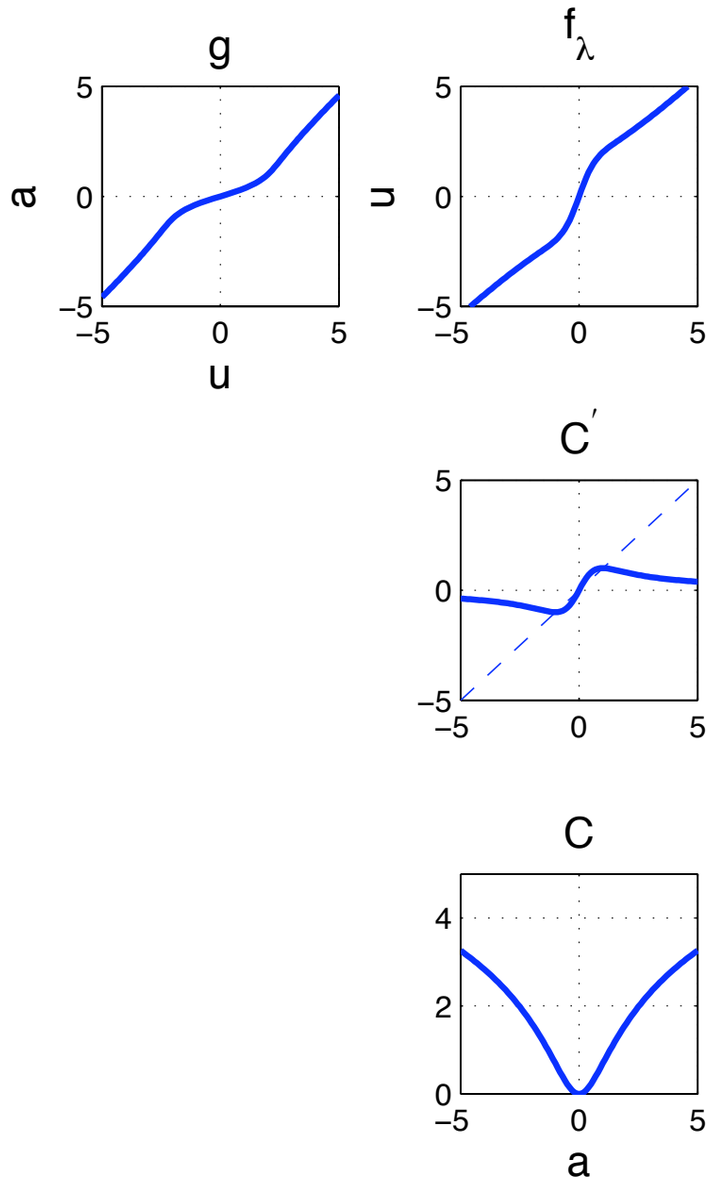
$$\boxed{\begin{aligned} \tau\, \dot{u}_i + u_i &= b_i - \sum_{j \neq i} G_{ij}\, a_j \\ a_i &= g(u_i) \end{aligned}}$$

Relation between the thresholding function *g* and cost function *C*

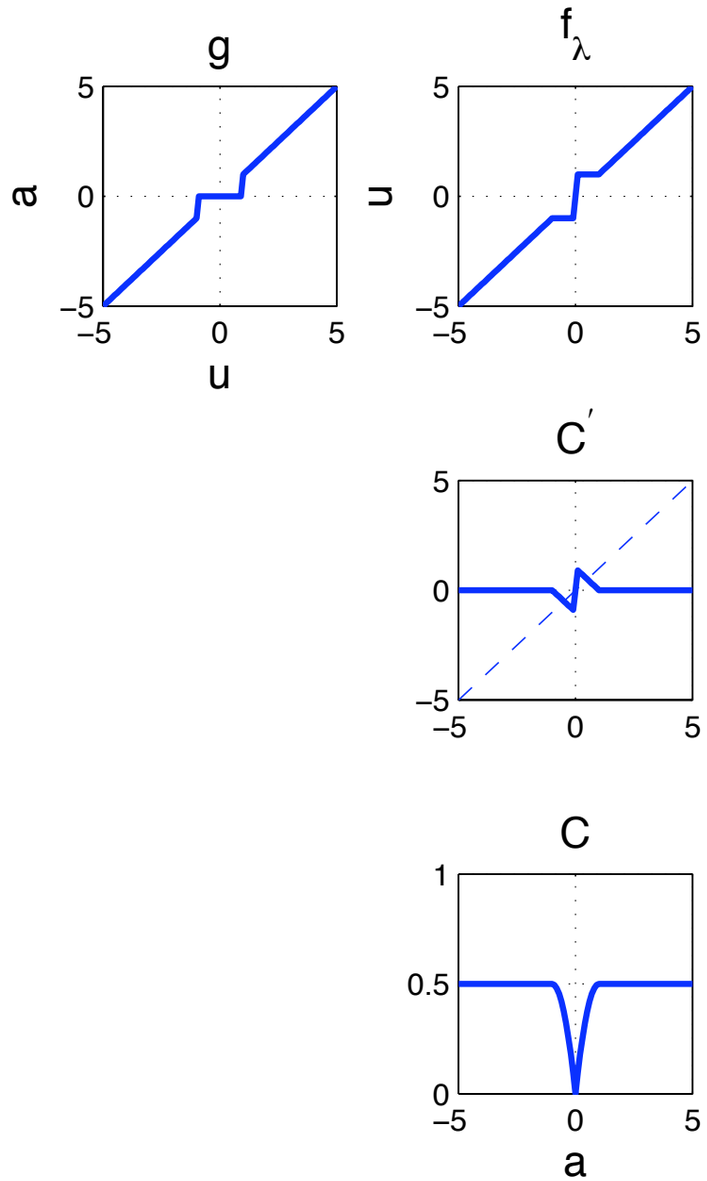
L1



$\log(1 + a^2)$

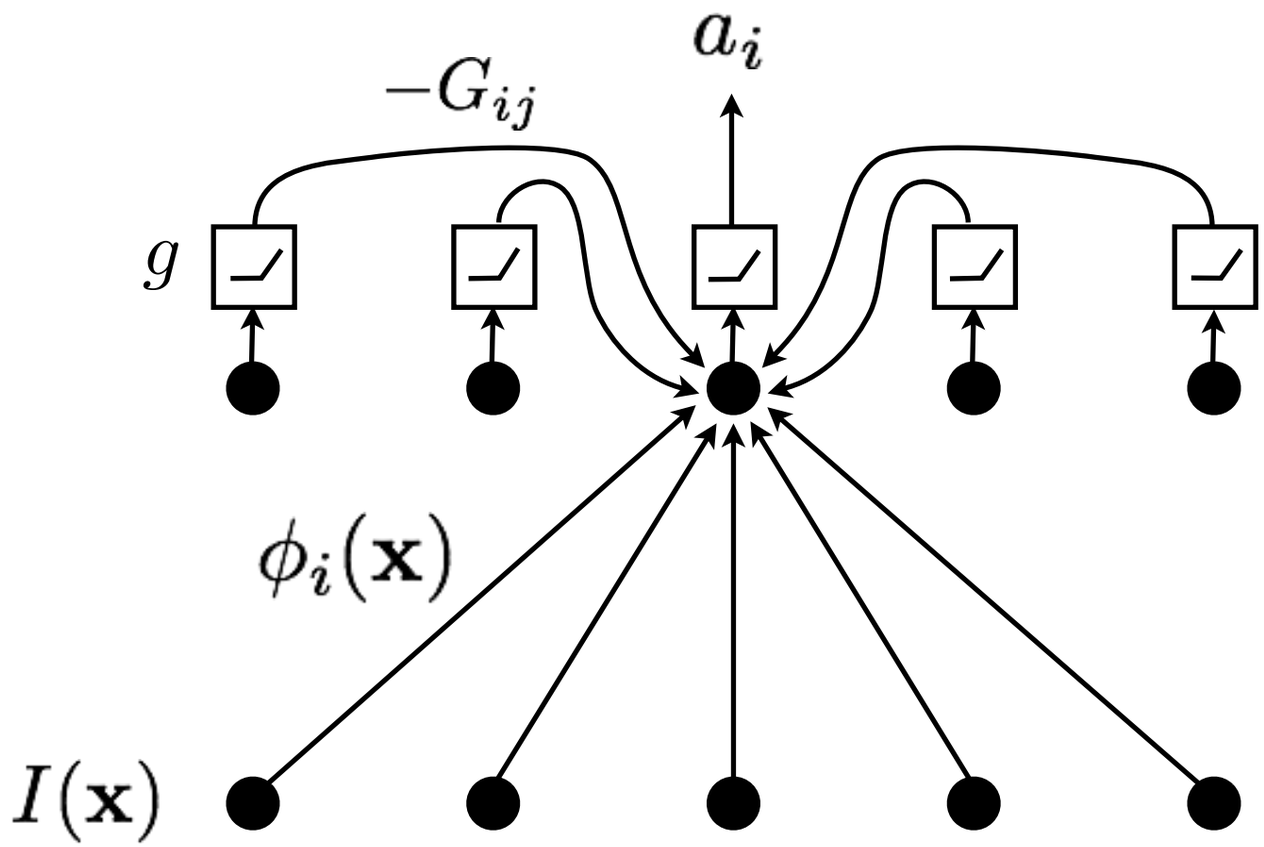


L0-like



Sparse inference via lateral inhibition and thresholding

(Rozell, Johnson, Baraniuk & Olshausen, 2008)



$$\tau\,\dot{u}_i + u_i \quad = \quad b_i - \sum_{j \neq i} G_{ij}\,a_j$$

$$a_i \quad = \quad g(u_i)$$

$$b_i = \sum_{\mathbf{x}} \phi_i(\mathbf{x})\, I(\mathbf{x})$$

$$G_{ij} = \sum_{\mathbf{x}} \phi_i(\mathbf{x})\, \phi_j(\mathbf{x})$$

Energy function

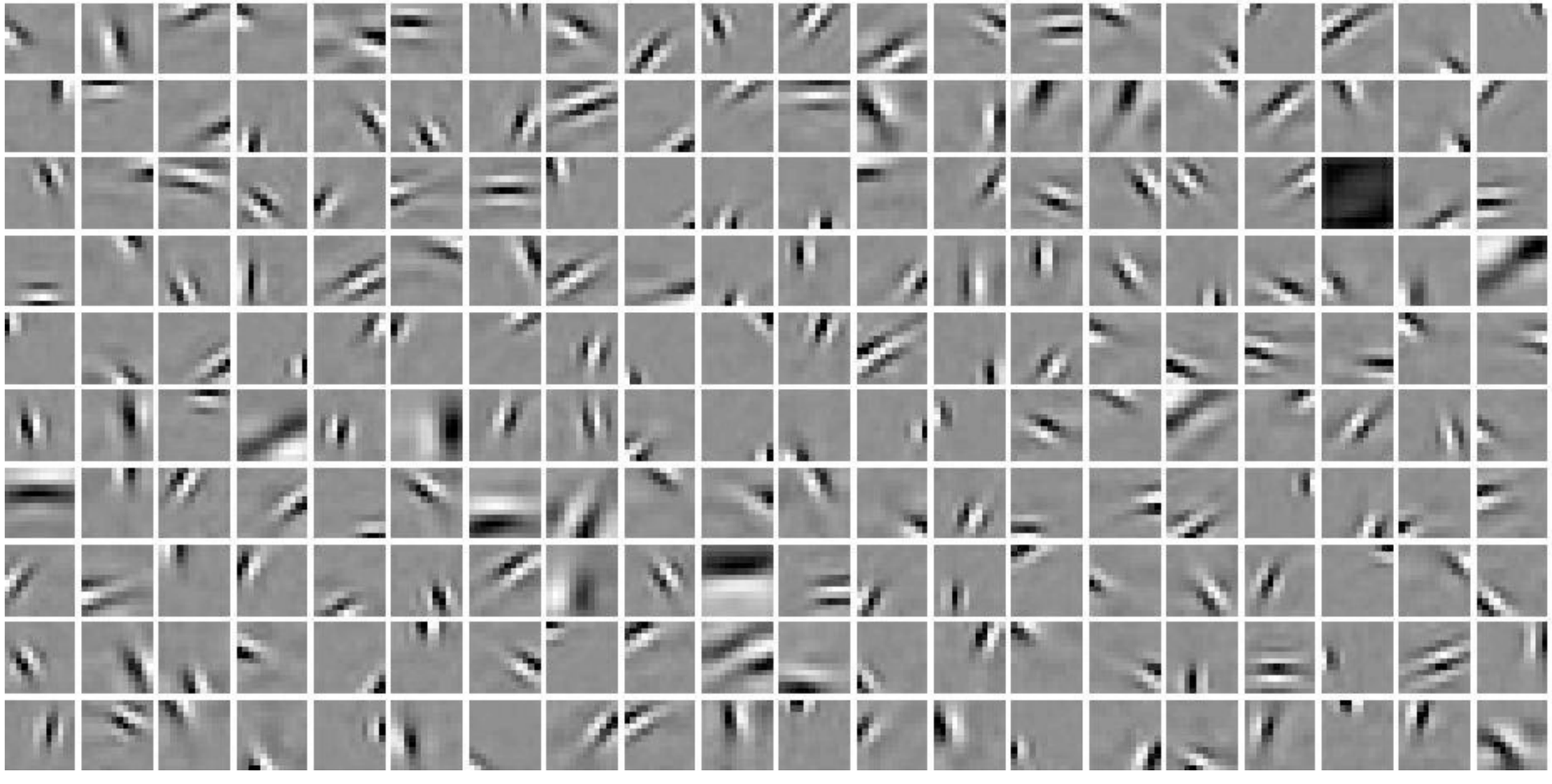
$$E = \frac{1}{2}|\mathbf{I} - \Phi\,\mathbf{a}|^2 + \lambda \sum_i C(a_i)$$

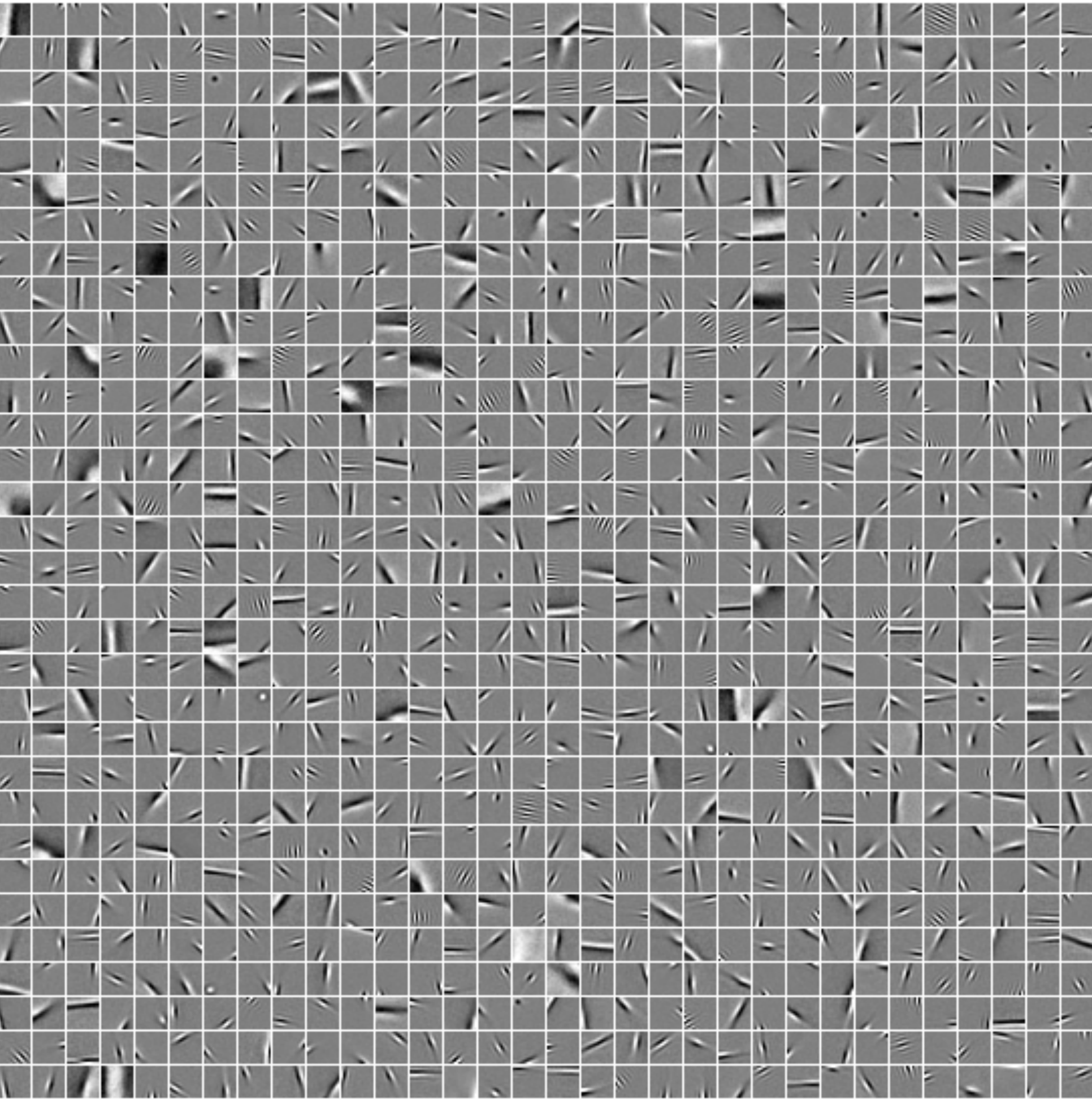
Learning rule

$$\begin{aligned} \Delta\phi_i &= -\eta\,\frac{\partial E}{\partial \phi_i} \\ &= \left[\mathbf{I} - \Phi\,\hat{\mathbf{a}}\right]\hat{a}_i \end{aligned}$$

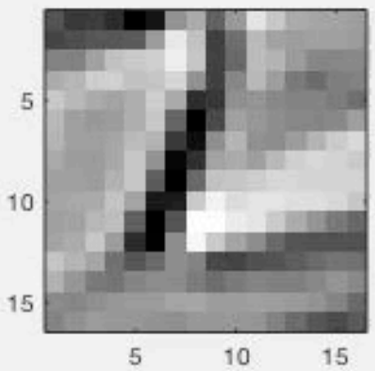
$\Phi_i(x, y)$ learned from natural images

(200, 12x12 pixels)

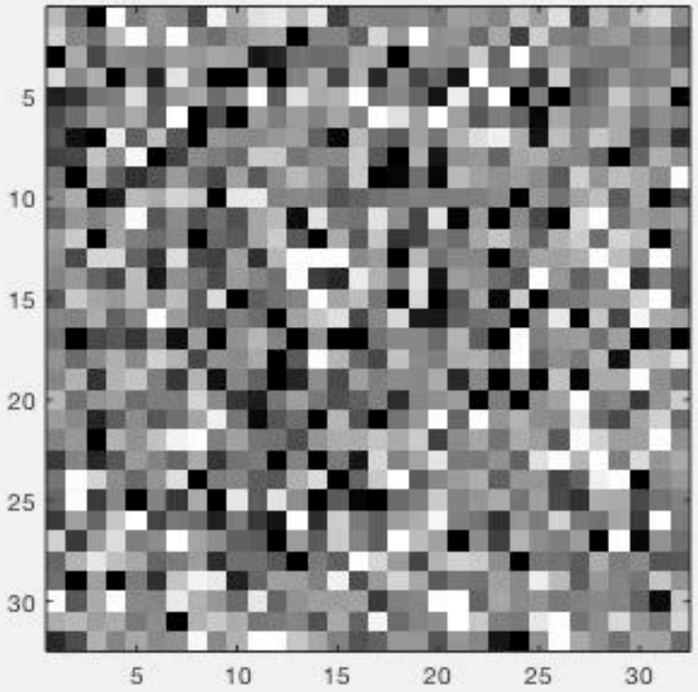




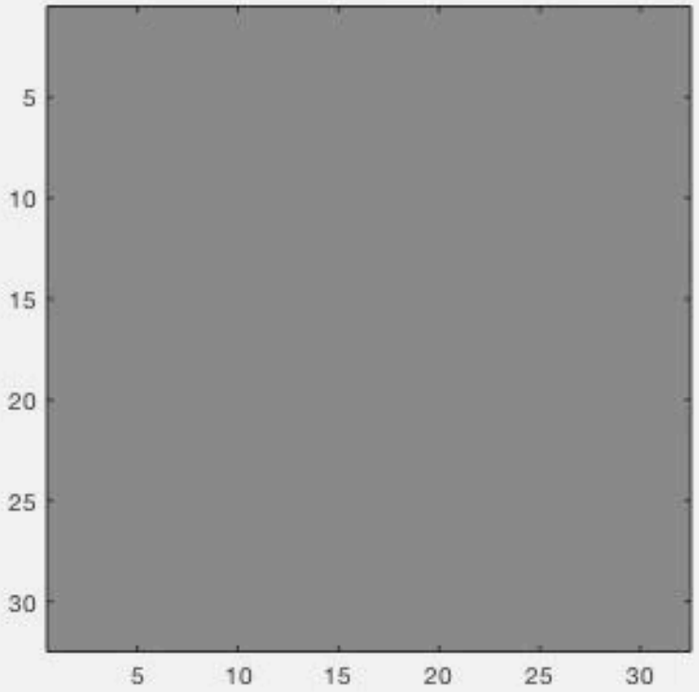
image



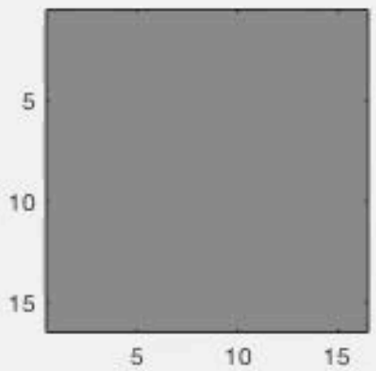
linear projection

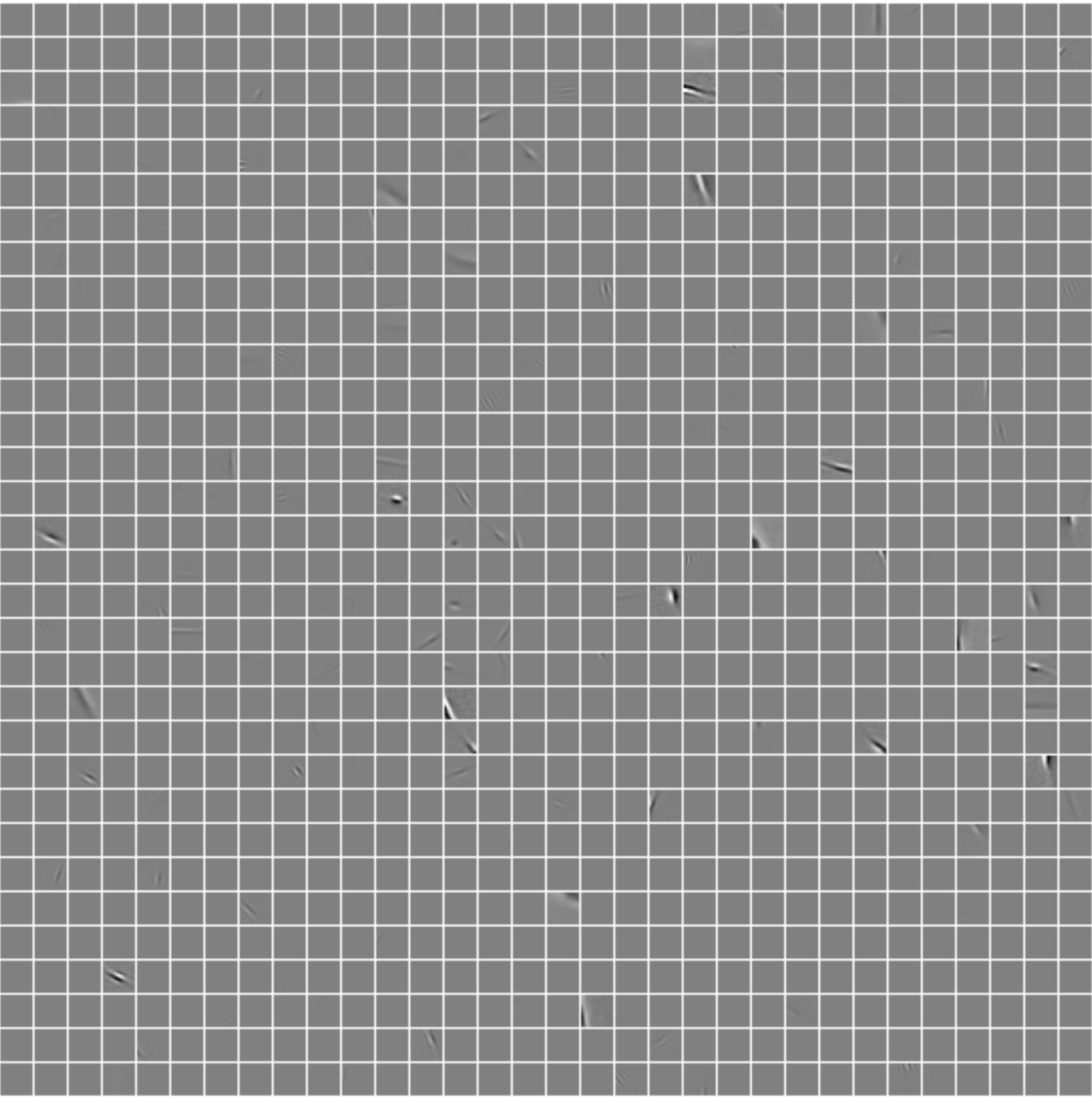


sparsified coefficients

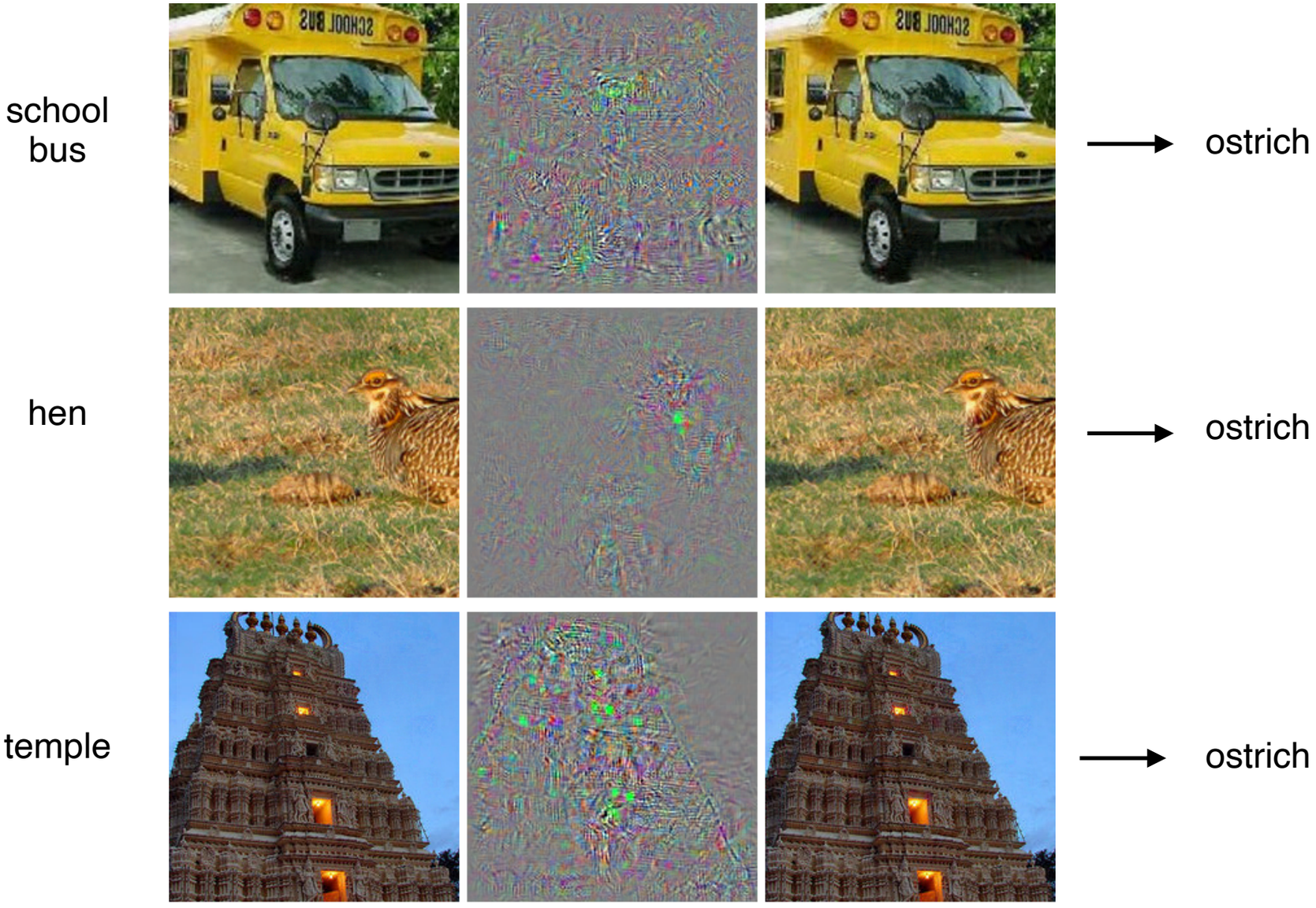


reconstruction





Deep convnets are easily fooled by imperceptible perturbations (adversarial examples)



Szegedy et al. (2013)

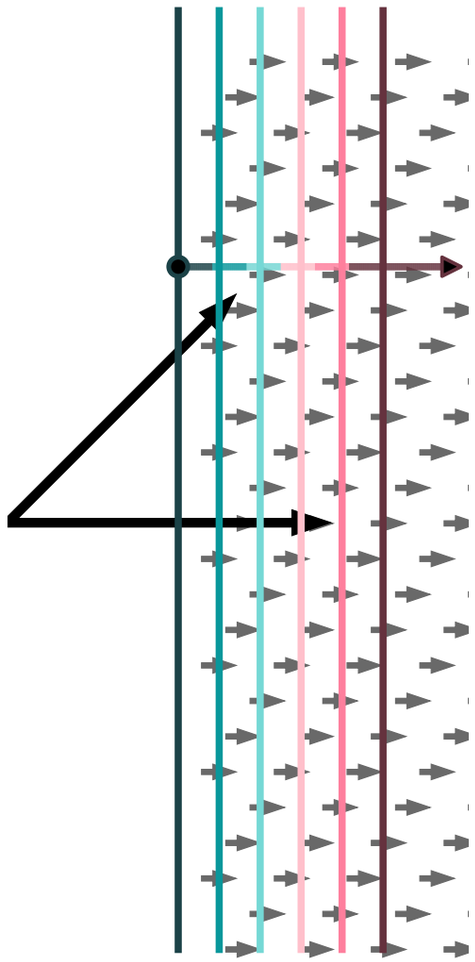
Sparse inference protects against adversarial attack

(Paiton, Frye, Lundquist, Bowen, Zarcone & Olshausen 2020)

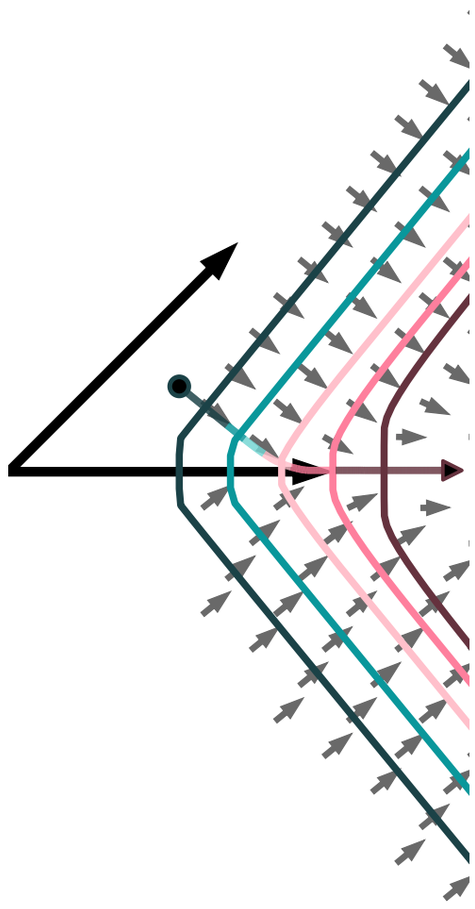
Dylan Paiton

Joel Bowen

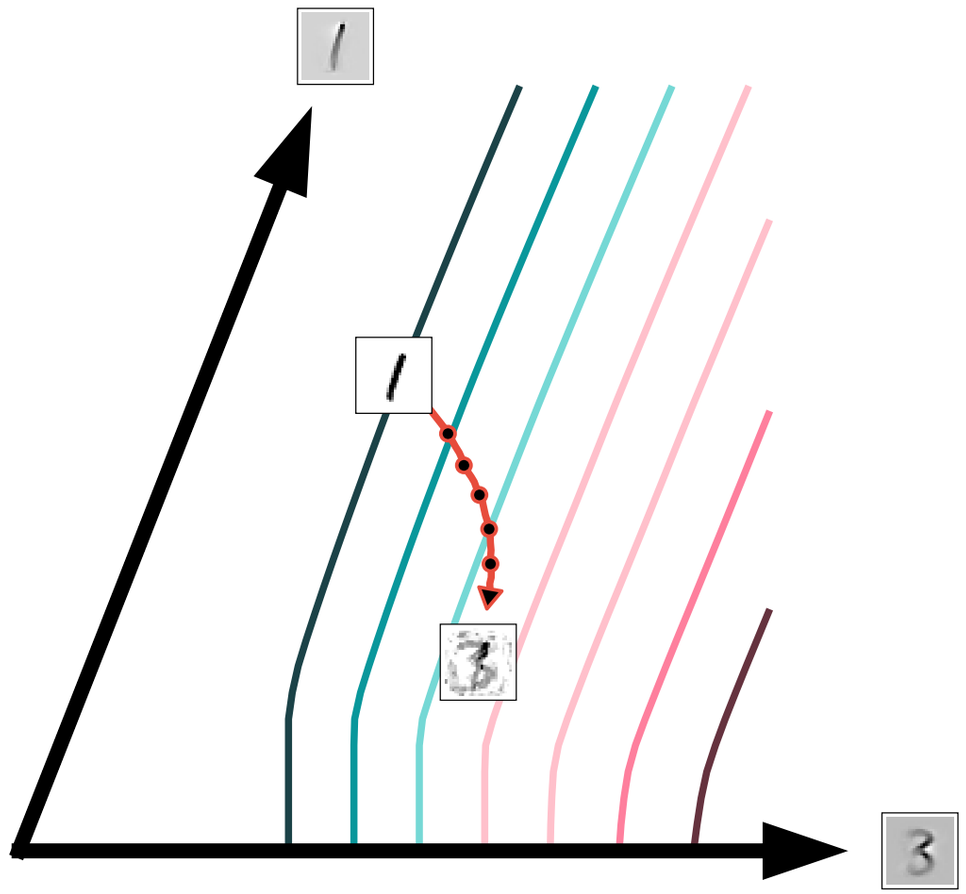
iso-response contours



linear projection



sparsified



Relationship between Sparse Coding and ICA

- No noise

$$\mathbf{x} = \mathbf{A}\,\mathbf{s}$$

- Invertible **A** matrix

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

Thus $$p(\mathbf{x}|\mathbf{s}) = \delta(\mathbf{x} - \mathbf{A}\,\mathbf{s})$$

$$\begin{aligned} p(\mathbf{x}) &= \int \delta(\mathbf{x} - \mathbf{A}\,\mathbf{s})\, p_s(\mathbf{s})\, d\mathbf{s} \\ &= p_s(\mathbf{A}^{-1}\mathbf{x})/|\det \mathbf{A}| \end{aligned}$$

$$\log p(\mathbf{x}) = -\sum_i C(s_i) - \log\det \mathbf{A}$$

$$\Delta \mathbf{A} \propto \frac{\partial}{\partial \mathbf{A}} \langle \log p(\mathbf{x}) \rangle$$

$$= \frac{\partial}{\partial \mathbf{A}} \left[-\sum_i C(s_i) - \log \det \mathbf{A} \right]$$

$$= \boxed{\langle [\mathbf{A}^T]^{-1} \mathbf{z}(\mathbf{s}) \, \mathbf{s}^T - [\mathbf{A}^T]^{-1} \rangle}$$

Its the ICA learning rule!

Pre-multiplying by $\boldsymbol{A}\,\boldsymbol{A}^T$ (natural gradient) yields:

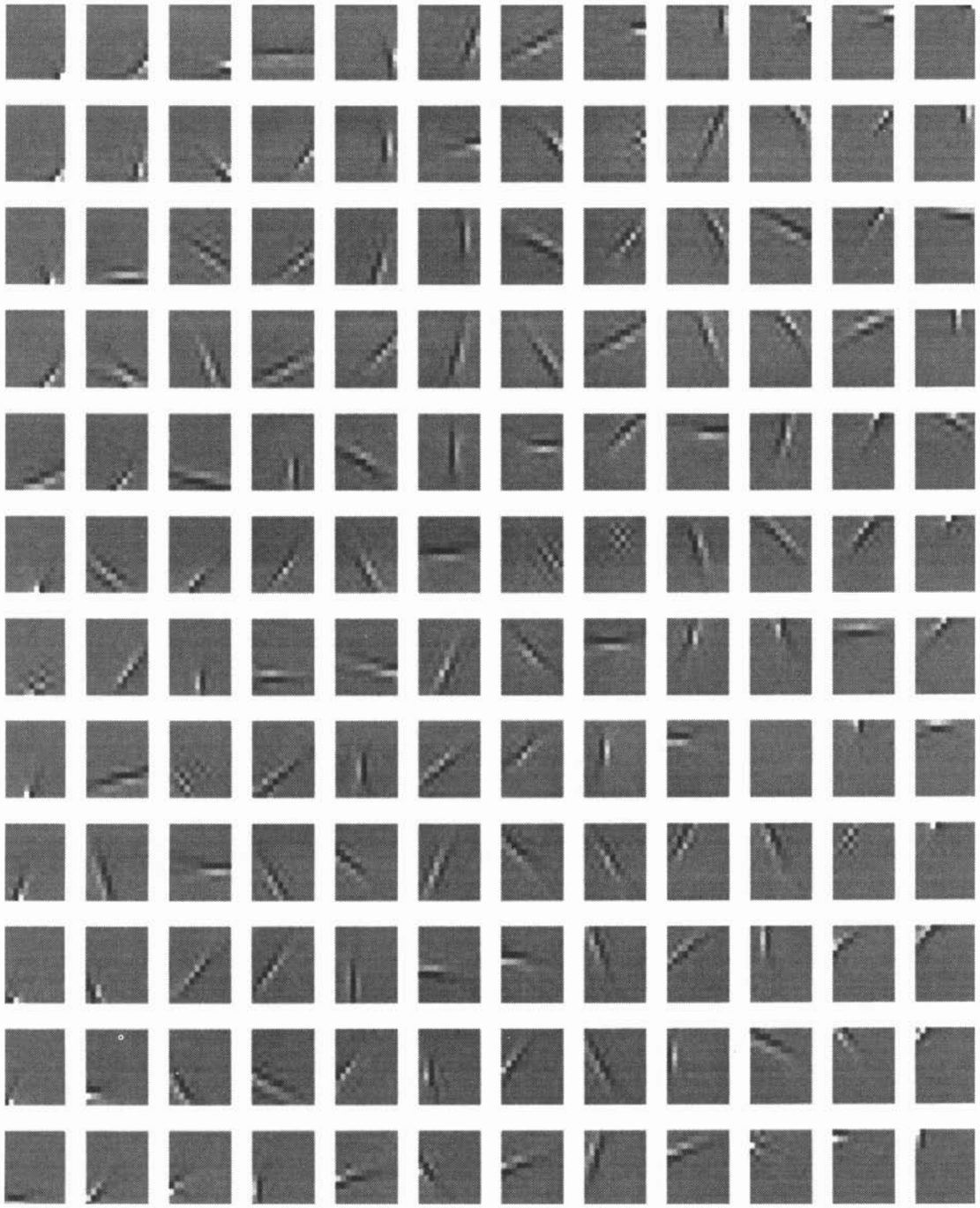
$$\Delta \mathbf{A} \propto \langle \mathbf{A} \, \mathbf{z} \, \mathbf{s}^T - \mathbf{A} \rangle$$

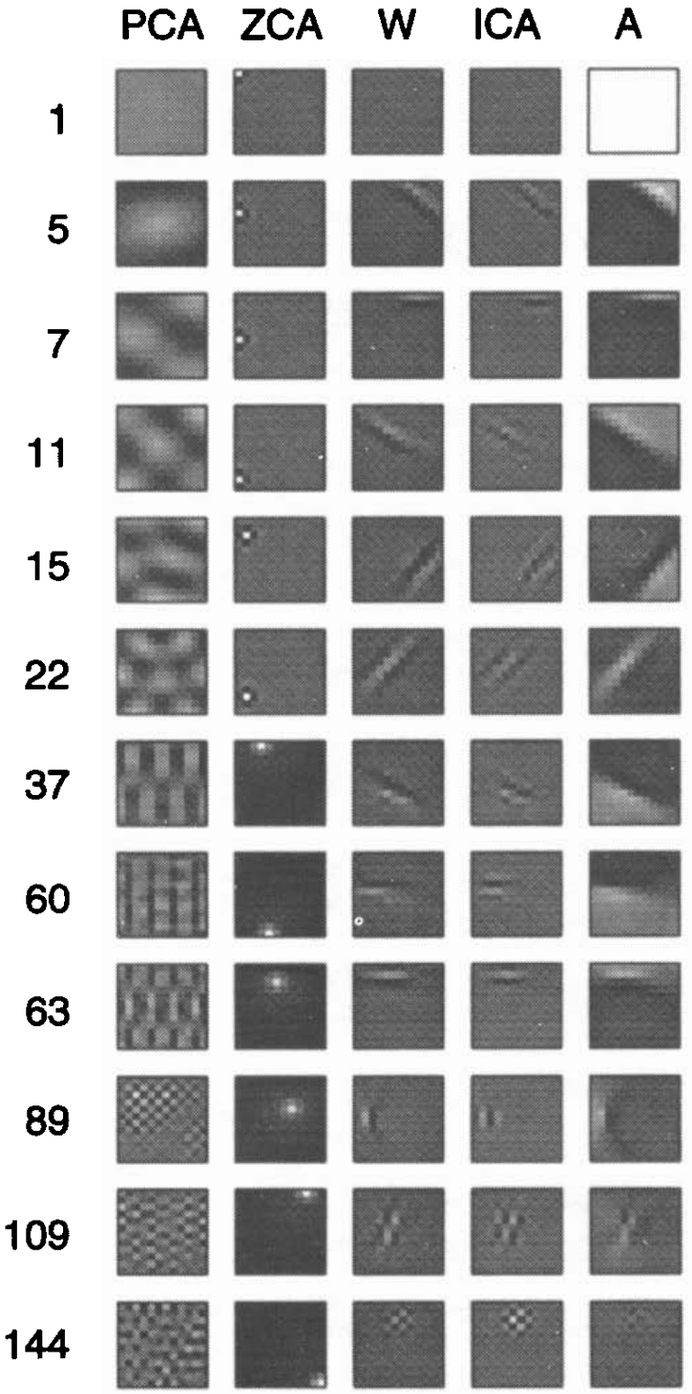
$$= \langle [\mathbf{x} - \mathbf{A}(\mathbf{s} - \mathbf{z})] \, \mathbf{s}^T - \mathbf{A} \rangle$$

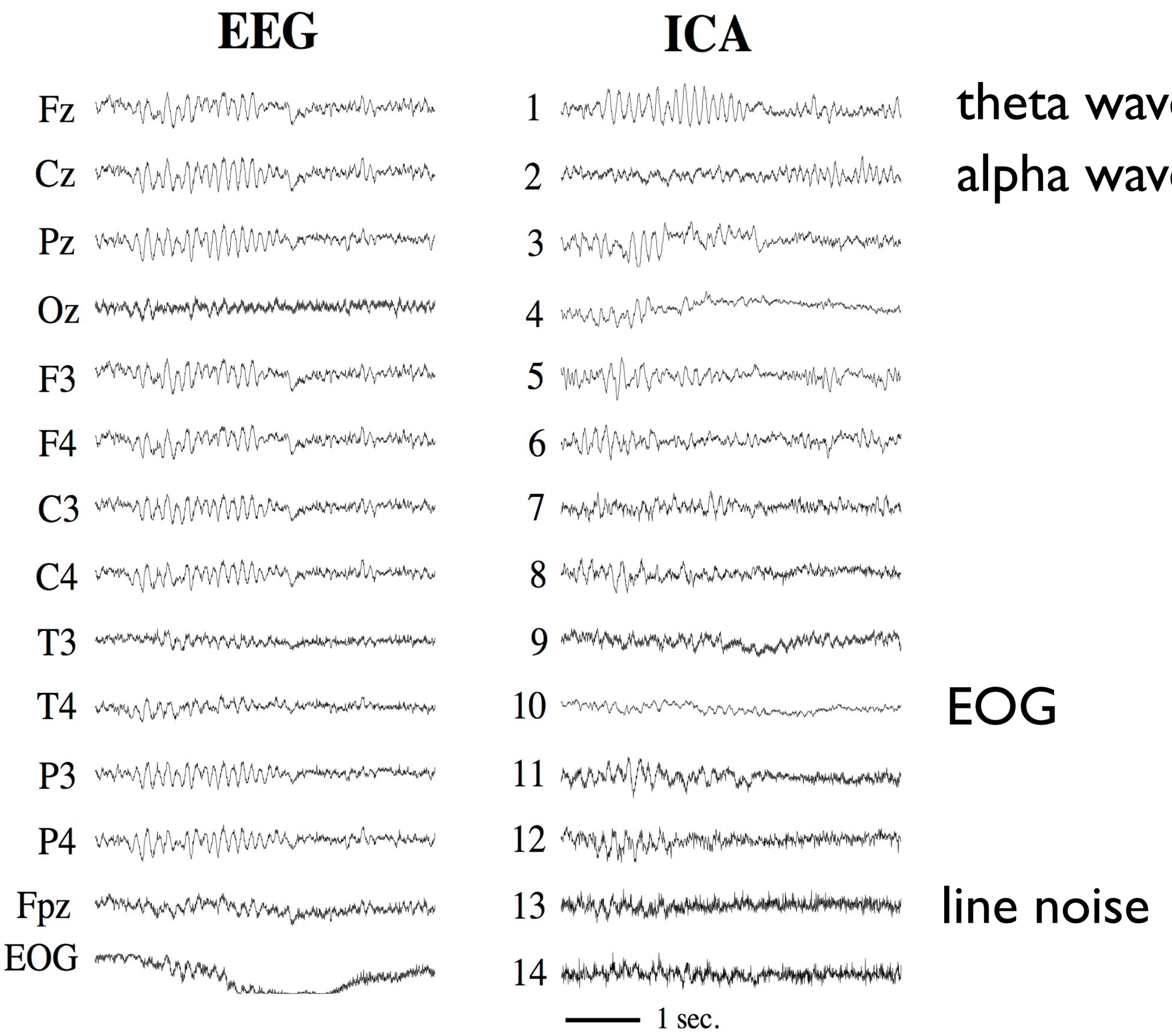
The "Independent Components" of Natural Scenes are Edge Filters

ANTHONY J. BELL,*† TERRENCE J. SEJNOWSKI*

Received 16 July 1996; in revised form 9 April 1997






EEG
ICA
Fz
Cz
Pz
Oz
F3
F4
C3
C4
T3
T4
P3
P4
Fpz
EOG
1
2
3
4
5
6
7
8
9
10
11
12
13
14
1 sec.
theta wave
alpha wave
EOG
line noise

ICA is not a general solution for finding independent components of data

Assumptions of the model:

1. linear superposition: $\mathbf{x} = \mathbf{A}\mathbf{s}$

2. shape of the prior over each of the components:

$$p(s_i) \propto e^{-C(s_i)}$$

3. factorial prior over the entire set of components:

$$p_s(\mathbf{s}) \propto \prod_i p_{s_i}$$