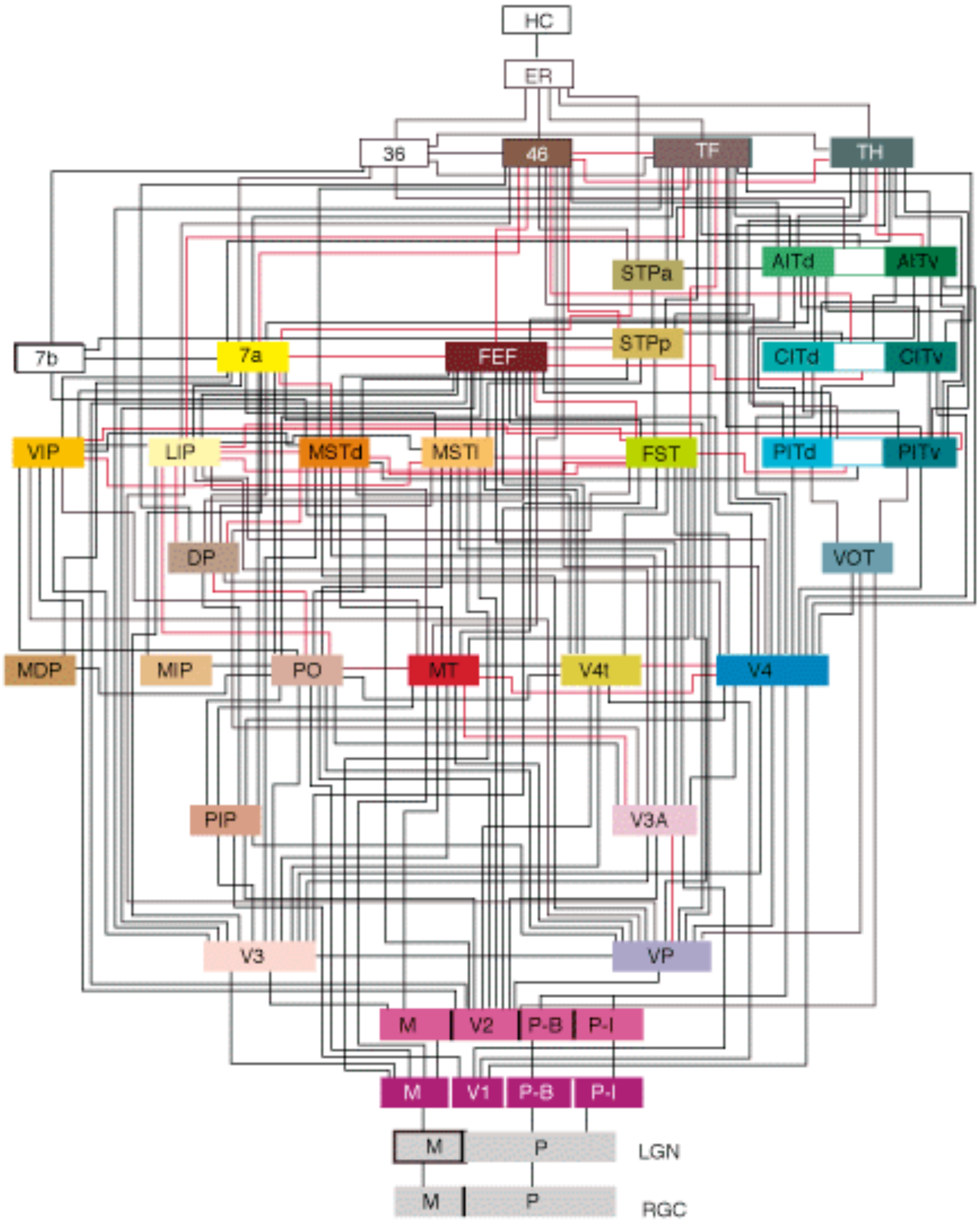


place cells
grid cells

.

.

face cells

.

invariant repr.
complex motion

.

.

?

.

.

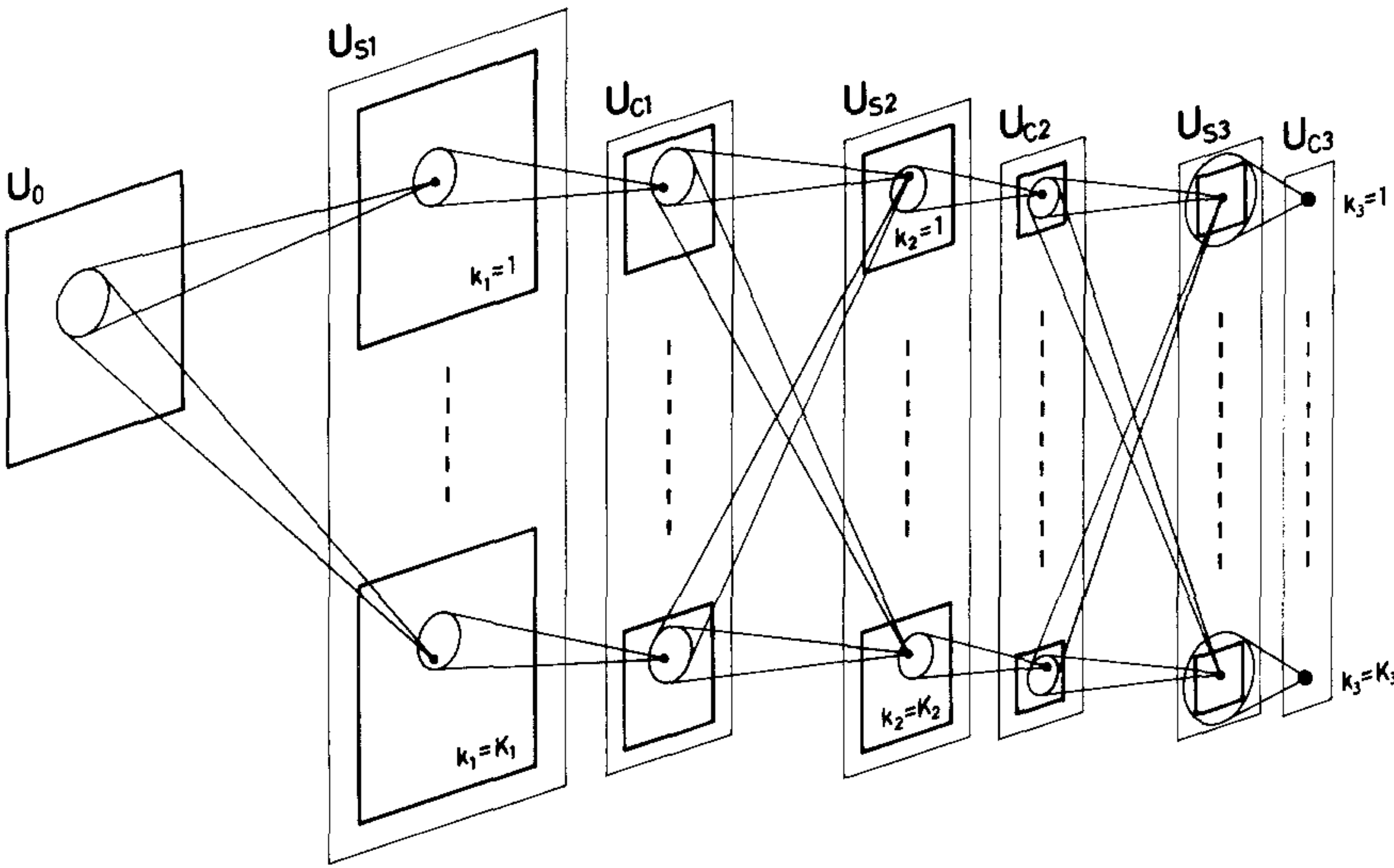
.

‘Gabor filters’

Biol. Cybernetics 36, 193–202 (1980)

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima
NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan



Neocognitron: rationale

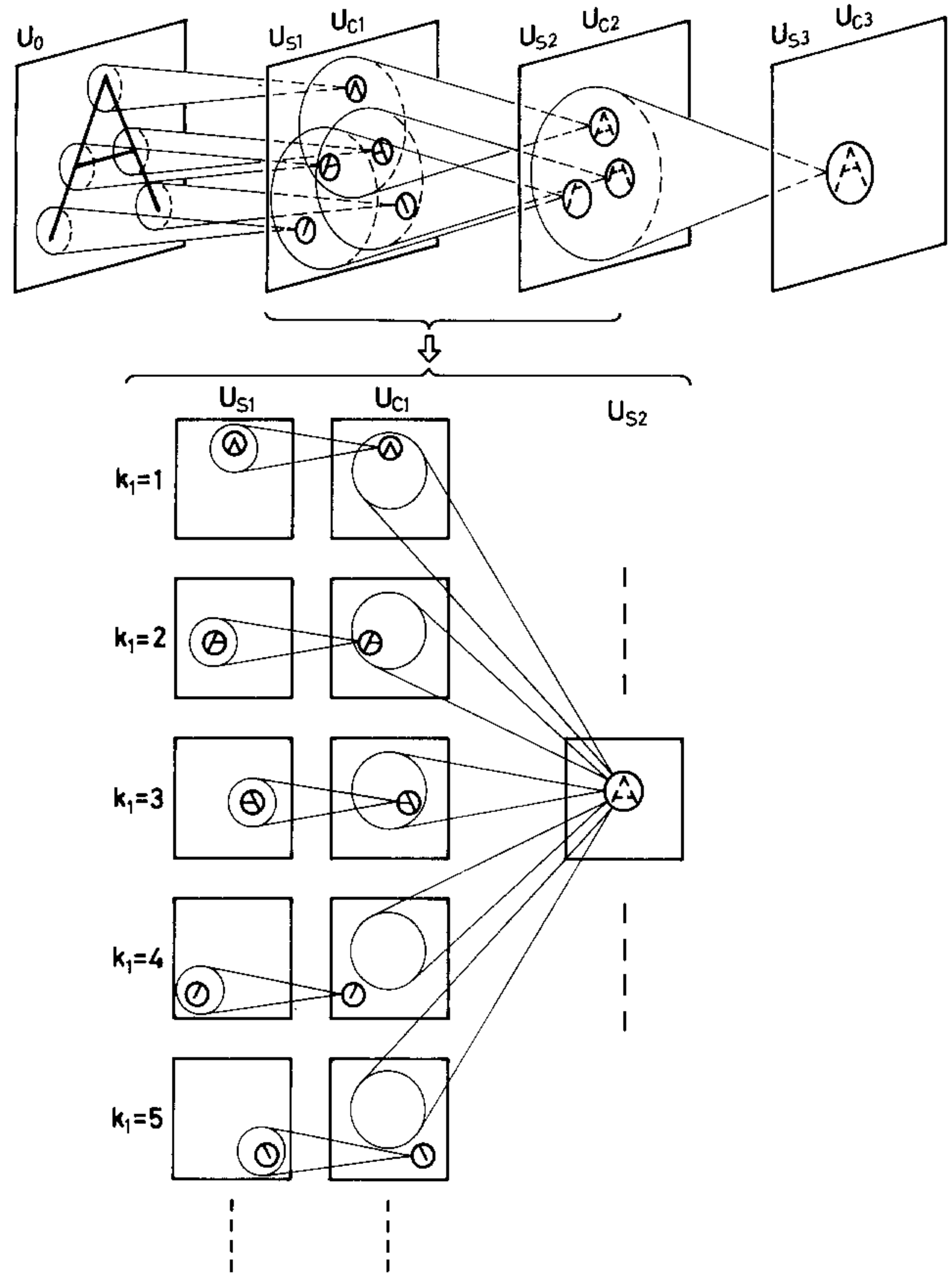


Fig. 5. An example of the interconnections between cells and the response of the cells after completion of self-organization

Neocognitron: activation rule

convolution

$$u_{Sl}(k_l, \mathbf{n}) = r_l \cdot \varphi\left[\frac{1 + \sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\mathbf{v} \in S_l} a_l(k_{l-1}, \mathbf{v}, k_l) \cdot u_{Cl-1}(k_{l-1}, \mathbf{n}+\mathbf{v})}{1 + \frac{2r_l}{1+r_l} \cdot b_l(k_l) \cdot v_{Cl-1}(\mathbf{n})} - 1 \right]$$

divisive normalization

where

$$\varphi[x] = \begin{cases} x & x \geqq 0 \\ 0 & x < 0. \end{cases}$$

Relu

$$v_{Cl-1}(\mathbf{n}) = \sqrt{\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\mathbf{v} \in S_l} c_{l-1}(\mathbf{v}) \cdot u_{Cl-1}^2(k_{l-1}, \mathbf{n}+\mathbf{v})},$$

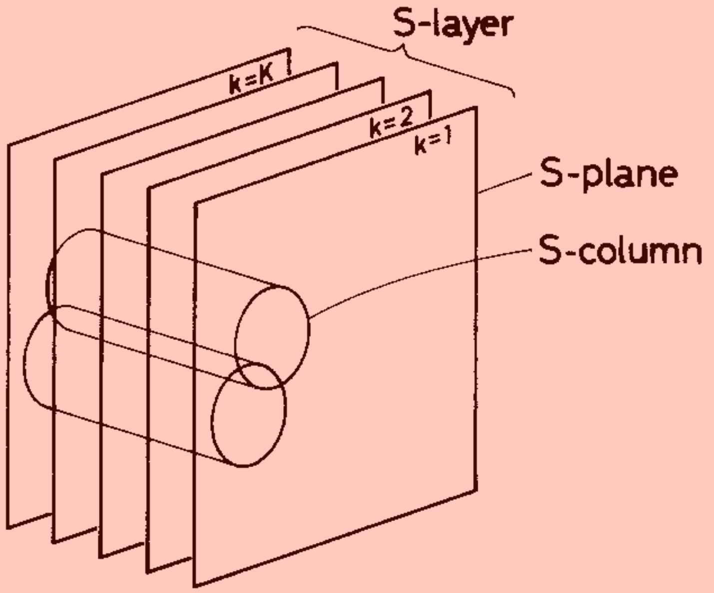
Neocognitron: learning rule

Let cell $u_{Sl}(\hat{k}_l, \hat{\mathbf{n}})$ be selected as a representative.

$$\Delta a_l(k_{l-1}, \mathbf{v}, \hat{k}_l) = q_l \cdot c_{l-1}(\mathbf{v}) \cdot u_{Cl-1}(k_{l-1}, \hat{\mathbf{n}} + \mathbf{v}),$$

Hebbian learning

From each S-column, every time when a stimulus pattern is presented, the S-cell which is yielding the largest output is chosen as a candidate for the representatives. Hence, there is a possibility that a number of candidates appear in a single S-plane. If two or more candidates appear in a single S-plane, only the one which is yielding the largest output among them is selected as the representative from that S-plane. In



Local WTA

Neocognitron: performance

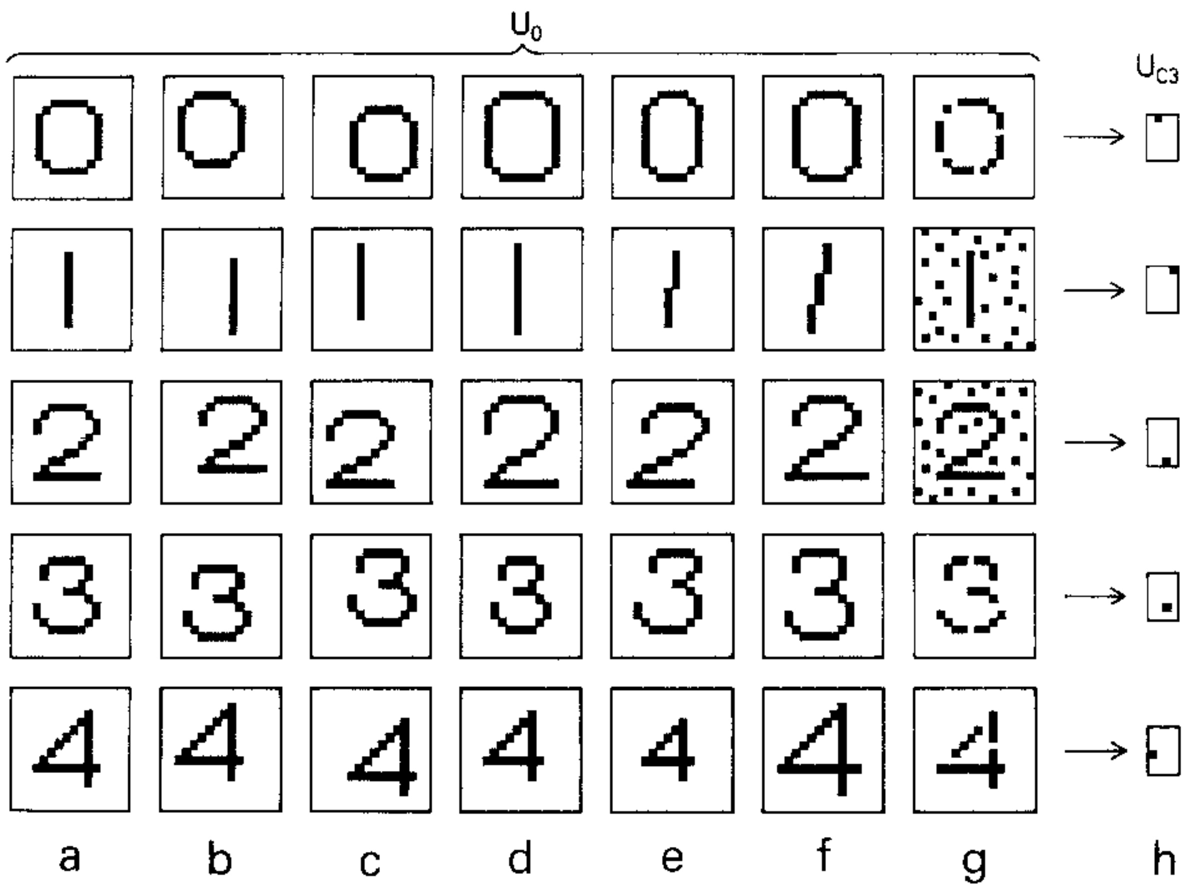


Fig. 6. Some examples of distorted stimulus patterns which the neocognitron has correctly recognized, and the response of the final layer of the network

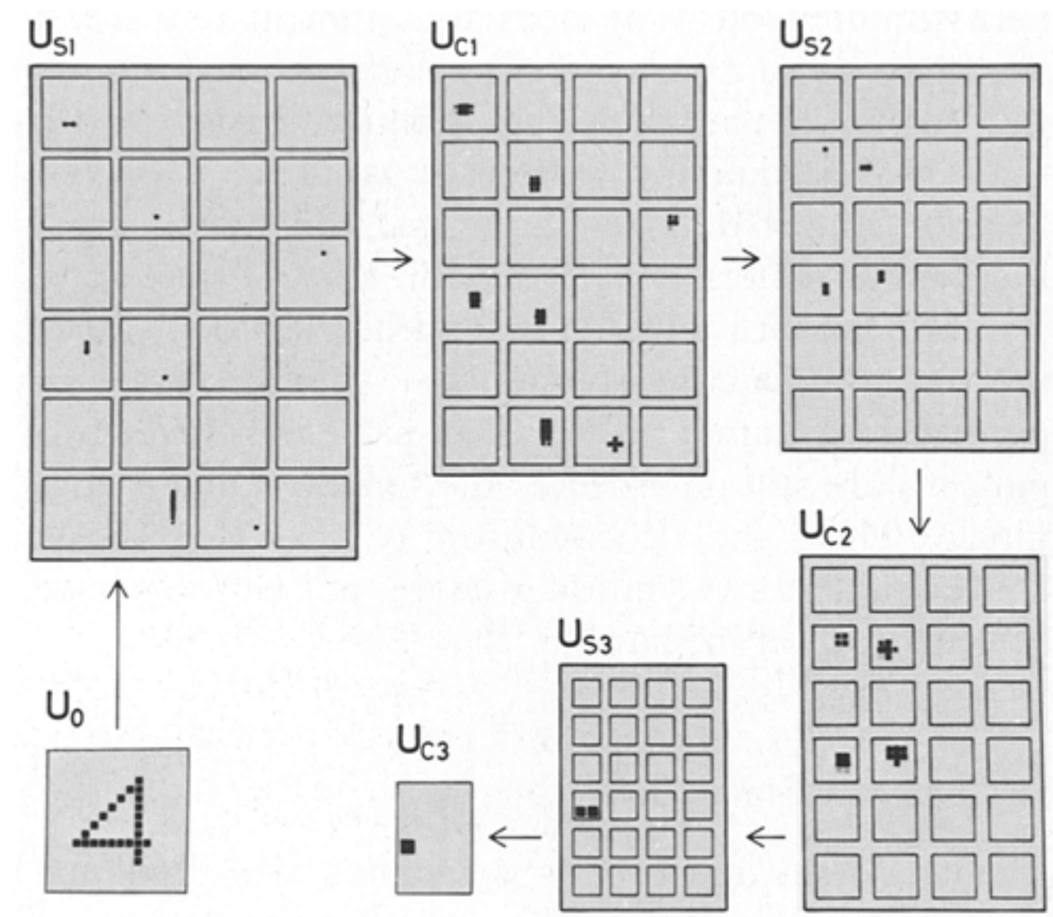


Fig. 7. A display of an example of the response of all the individual cells in the neocognitron

‘AlexNet’

(Krizhevsky, Sutskever & Hinton 2012)

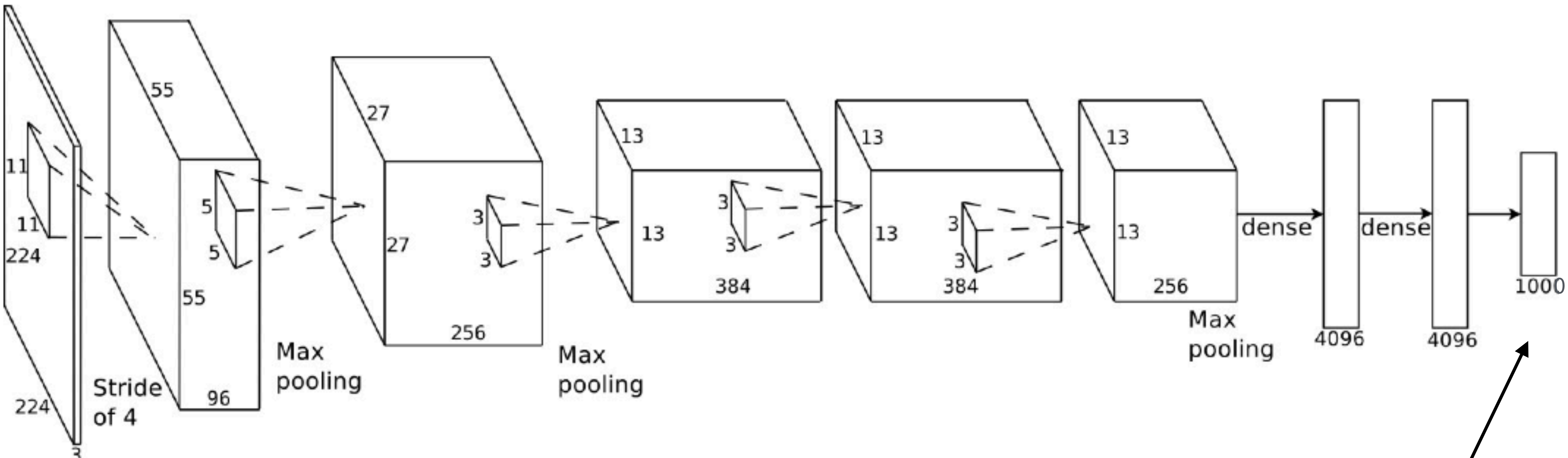
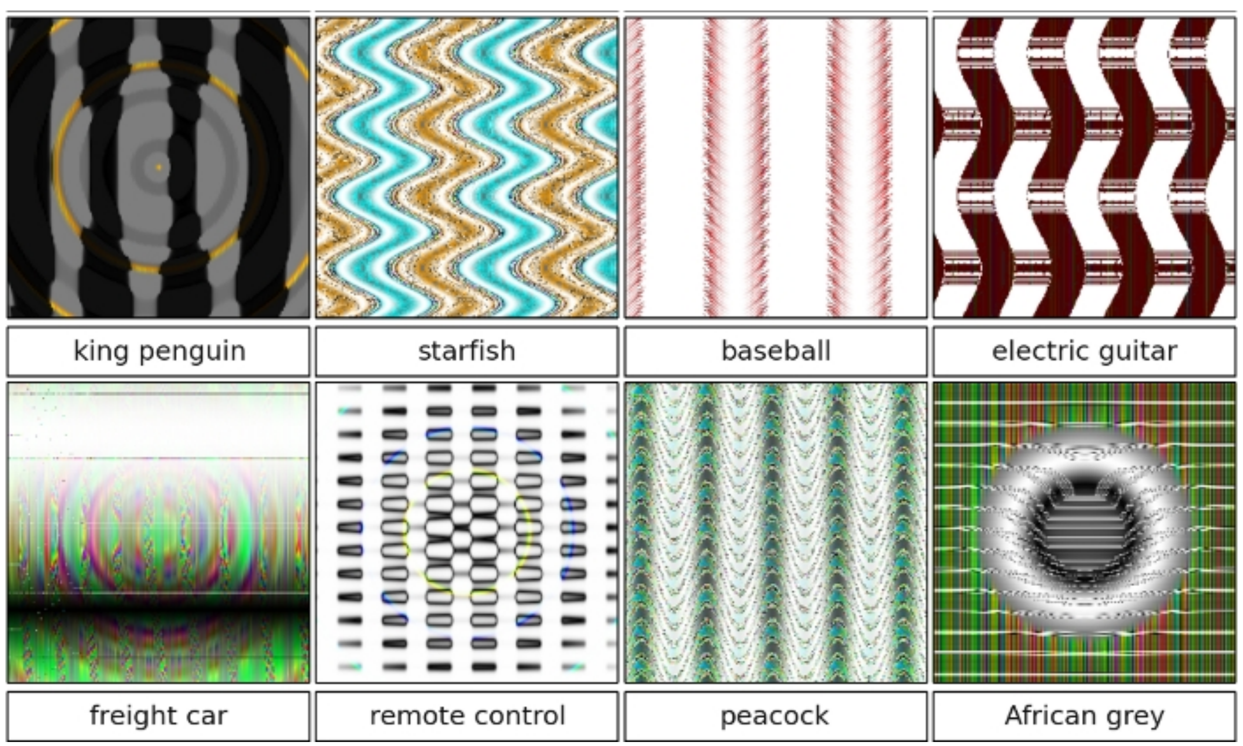


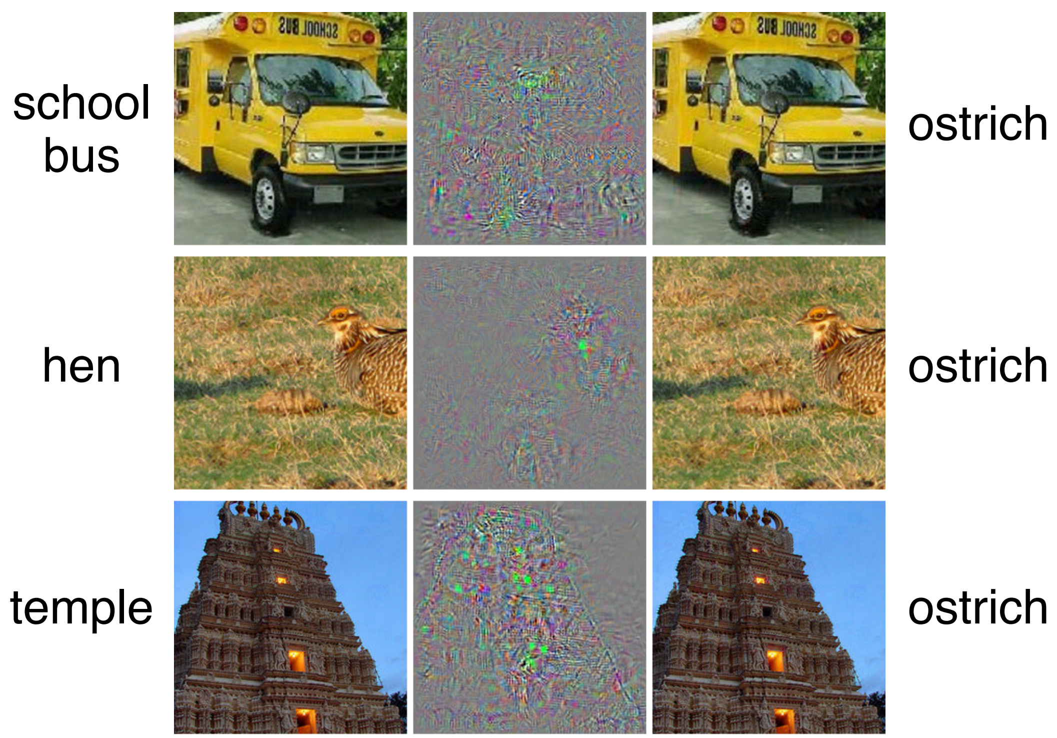
image feature extraction and pooling classification

This isn’t a good model of perception

The invariant representations produced by deep convnets are…

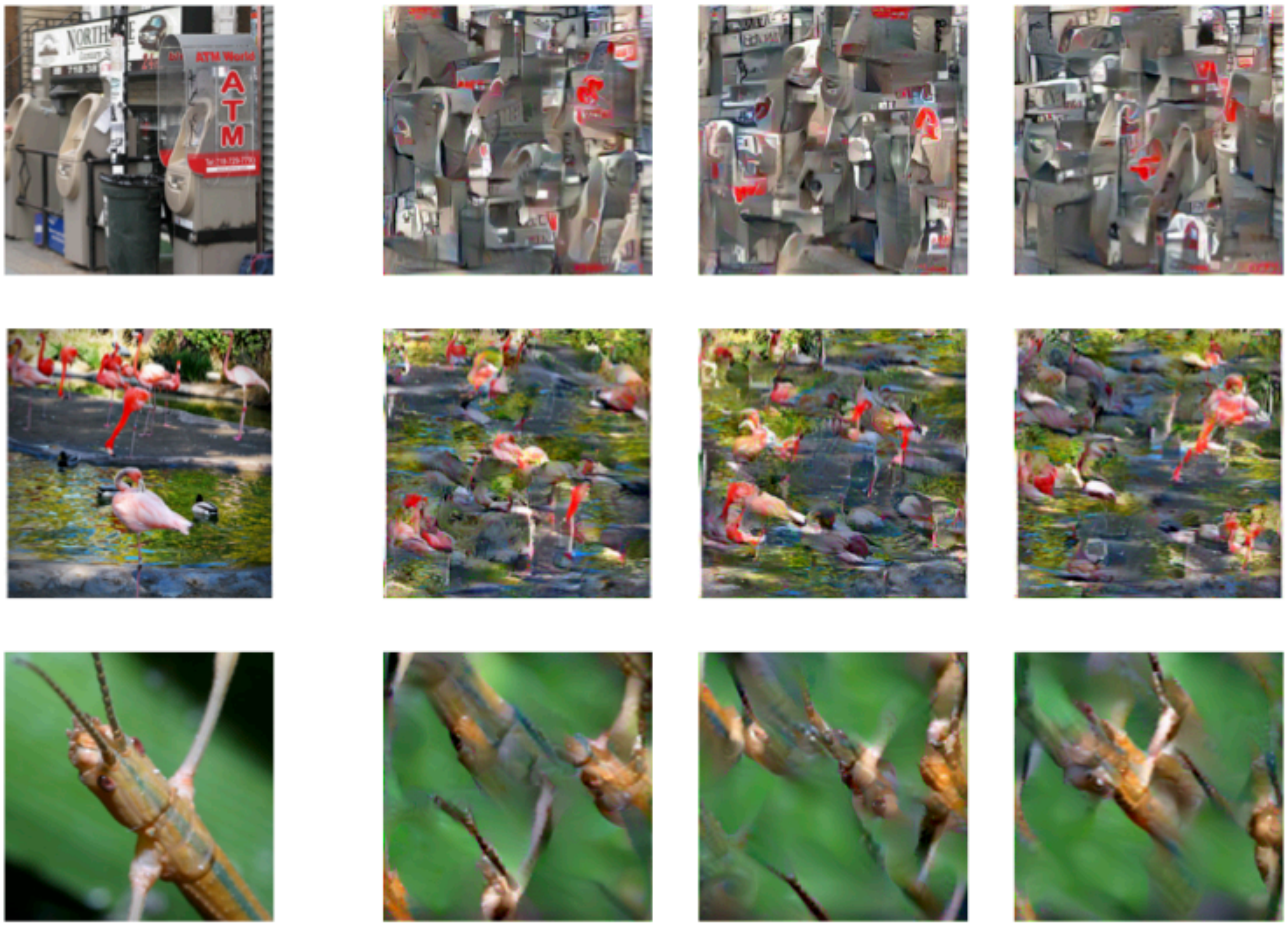
easily fooled and brittle



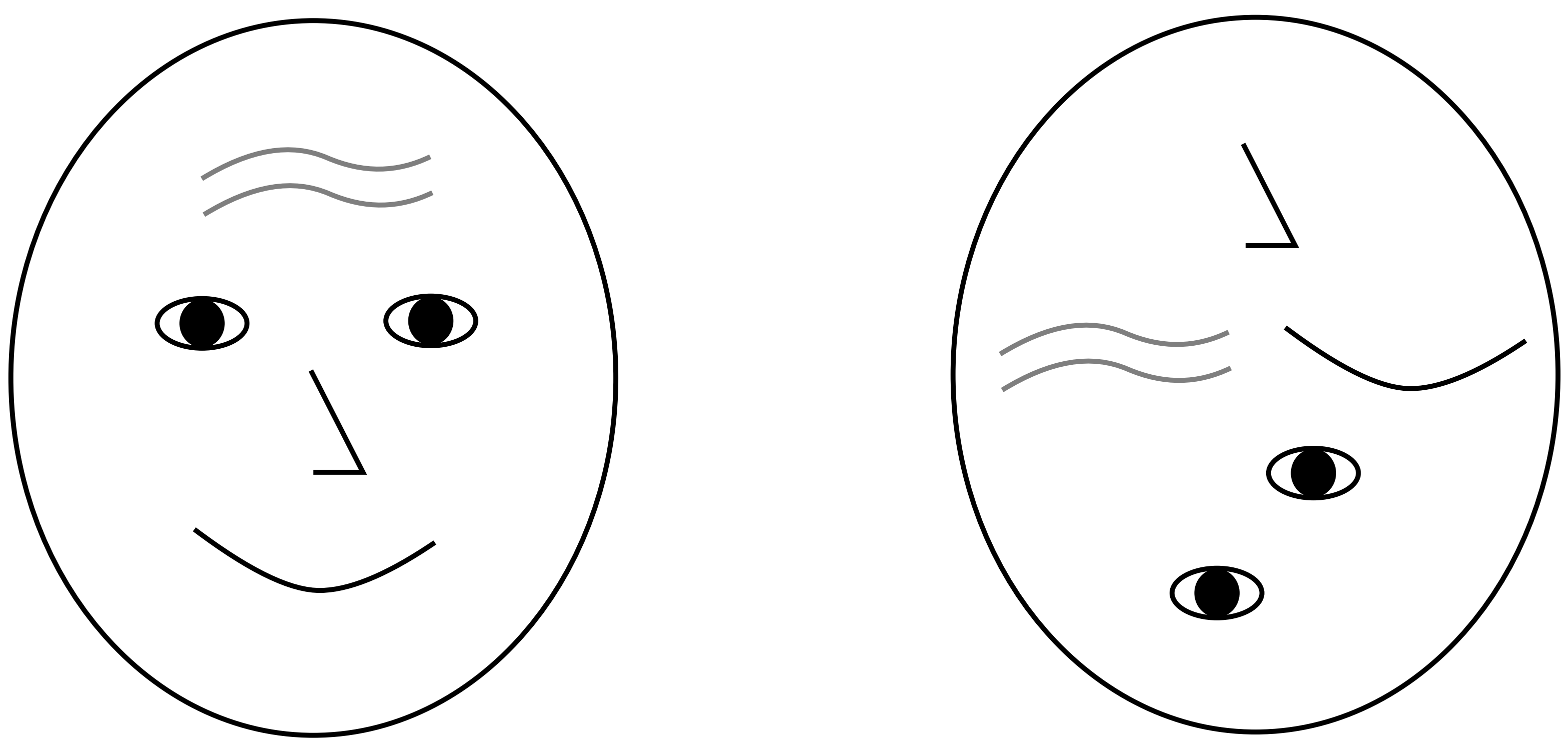


Images are not bags of features
(BagNet - Brendel & Bethge 2019)

original texturised images



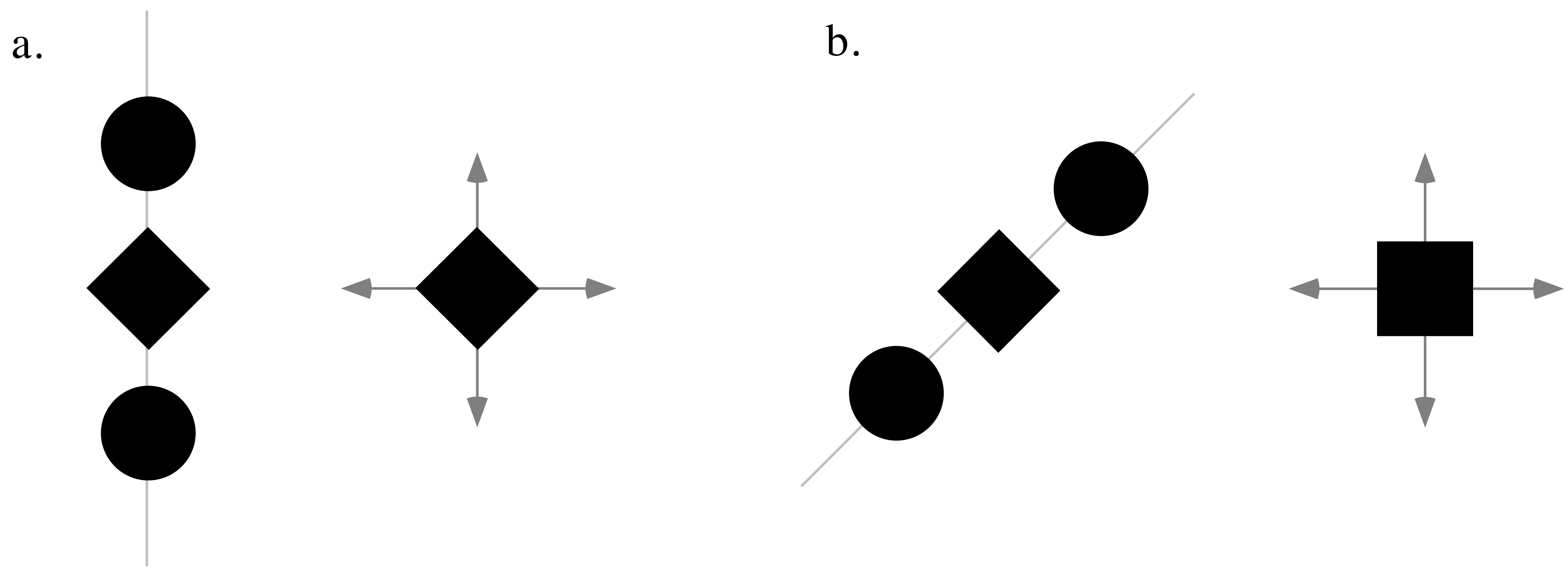
Relative spatial relationships are important



Pareidolia

Reference frame effects in perception

Diamond or square?



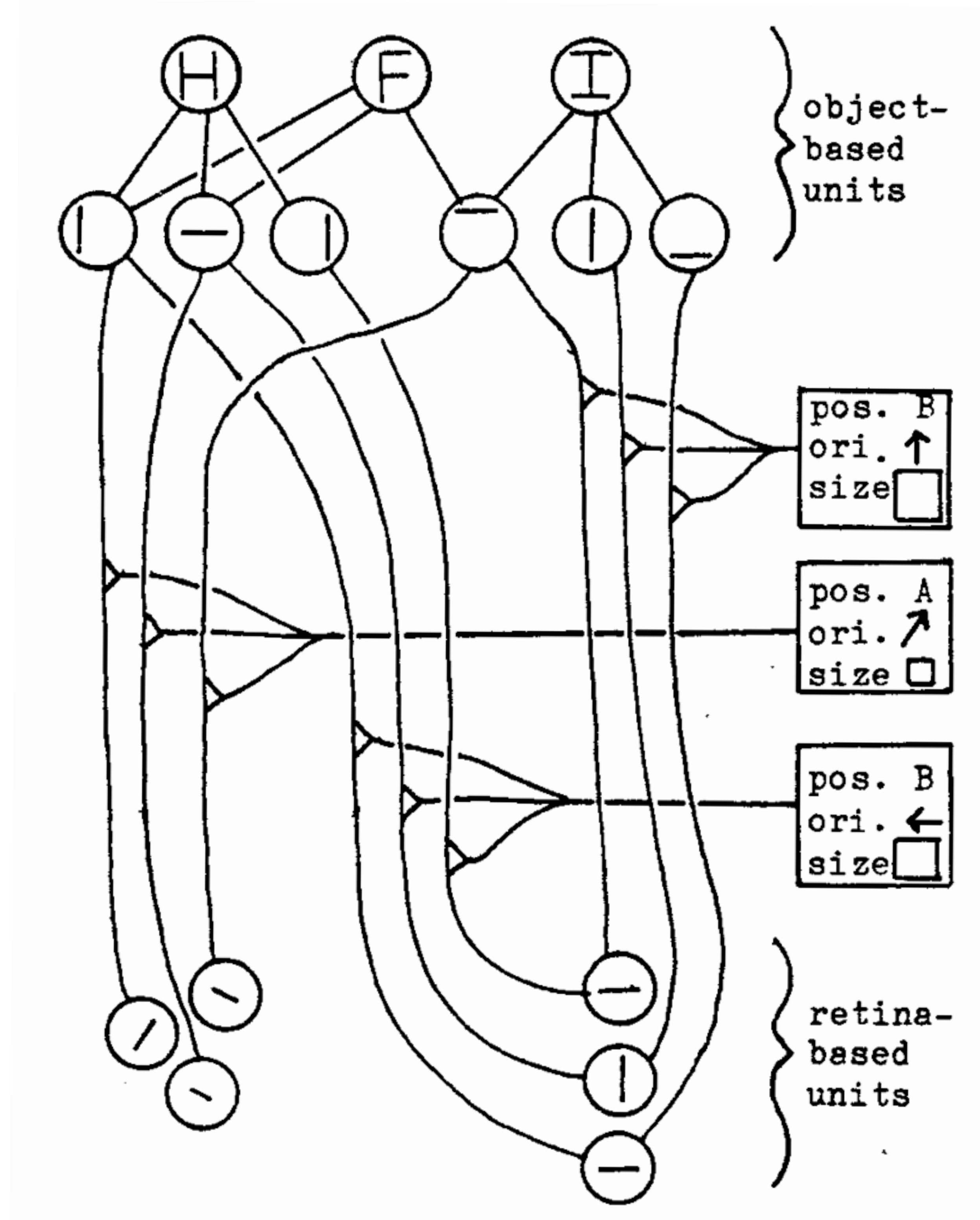
How to form invariant object representations?

Reference frames require *structured representations*

The meaning of the triangular symbol in fig. 1 is quite complex. It stands for two rules:

1. Multiply the activity level in the retina-based unit by the activity level in the mapping unit and send the product to the object-based unit.

2. Multiply the activity level in the retina-based unit by the activity level in the object-based unit and send the product to the mapping unit.

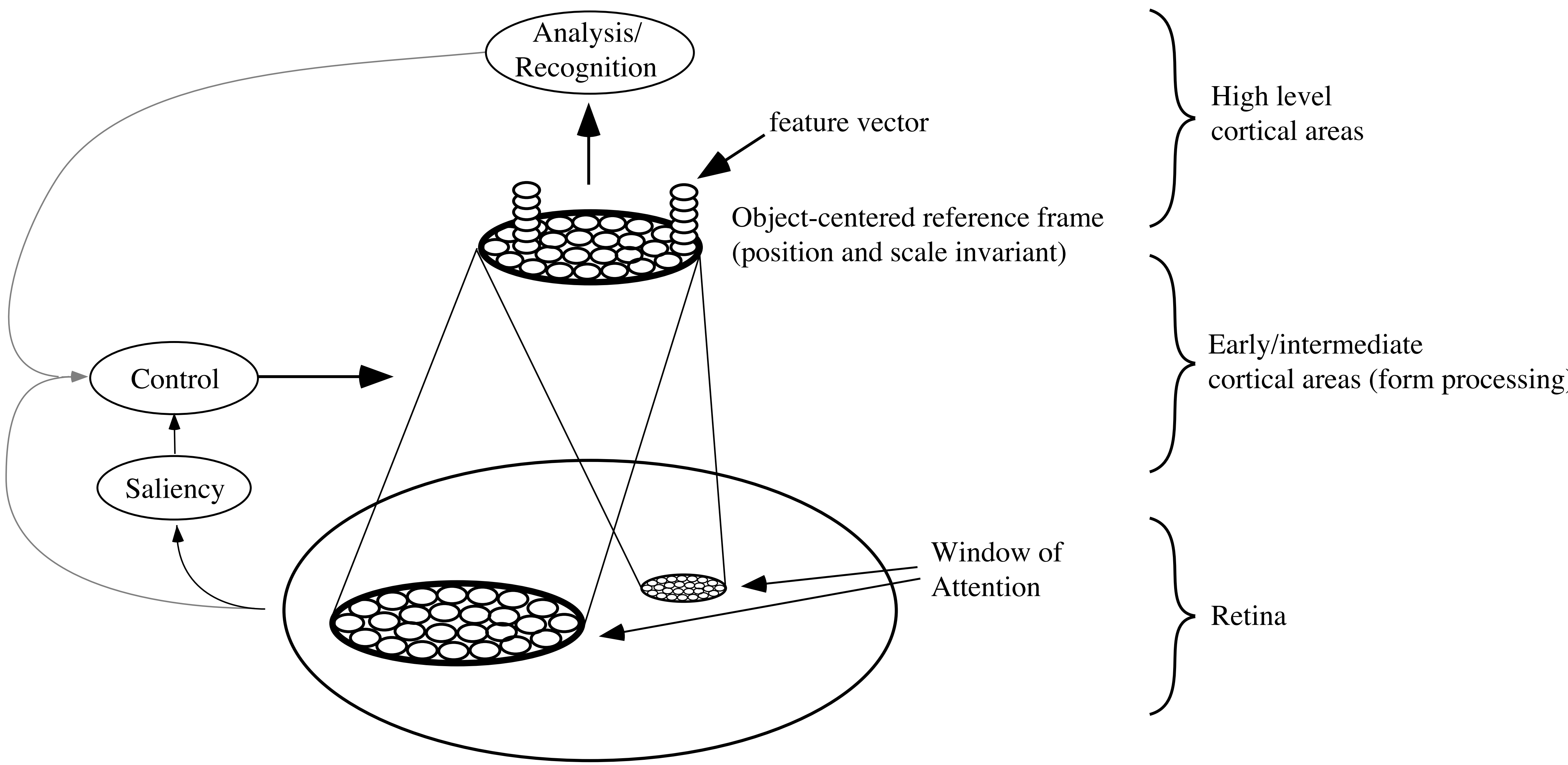


mapping units

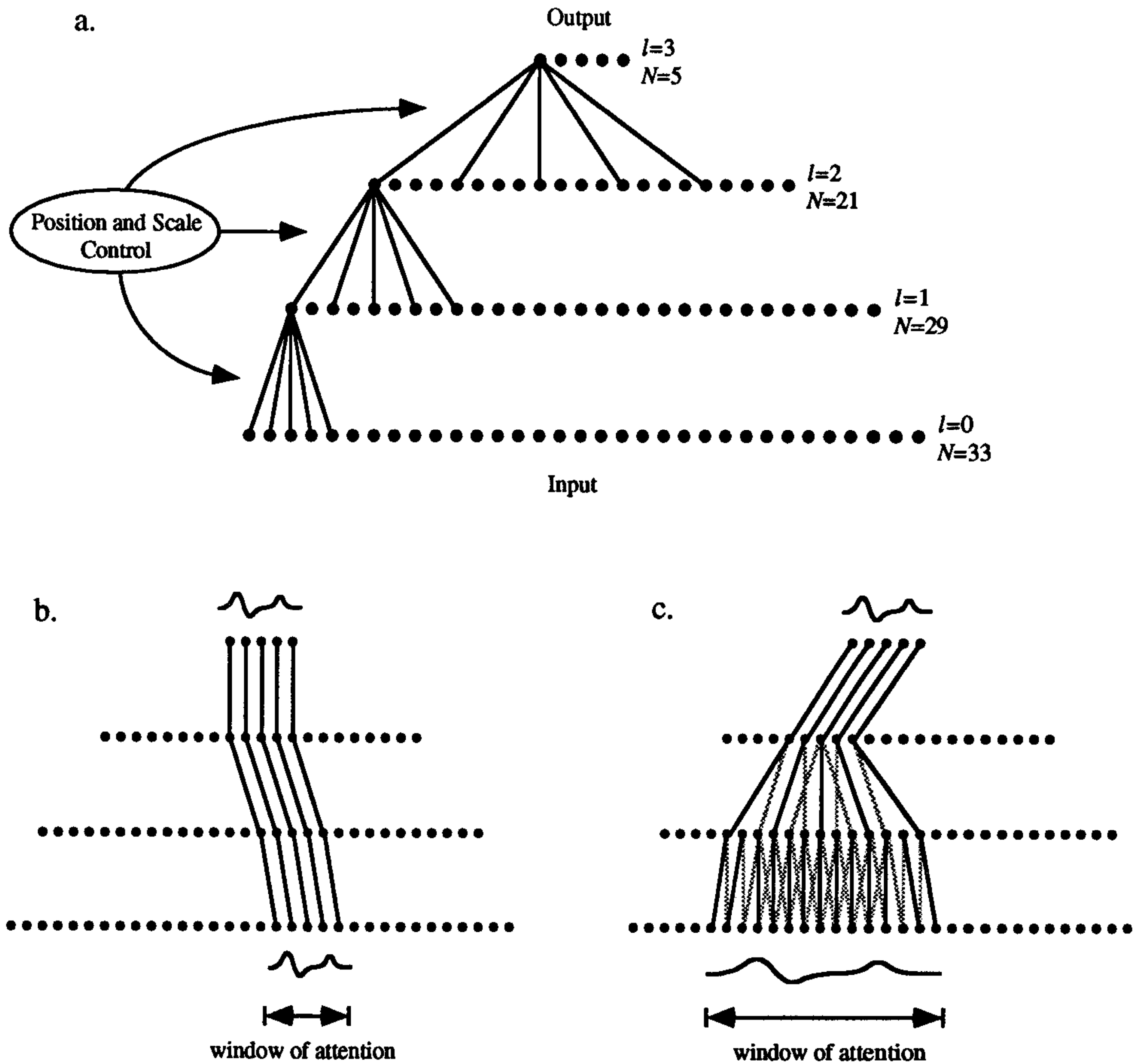
Hinton (1981)

Dynamic routing

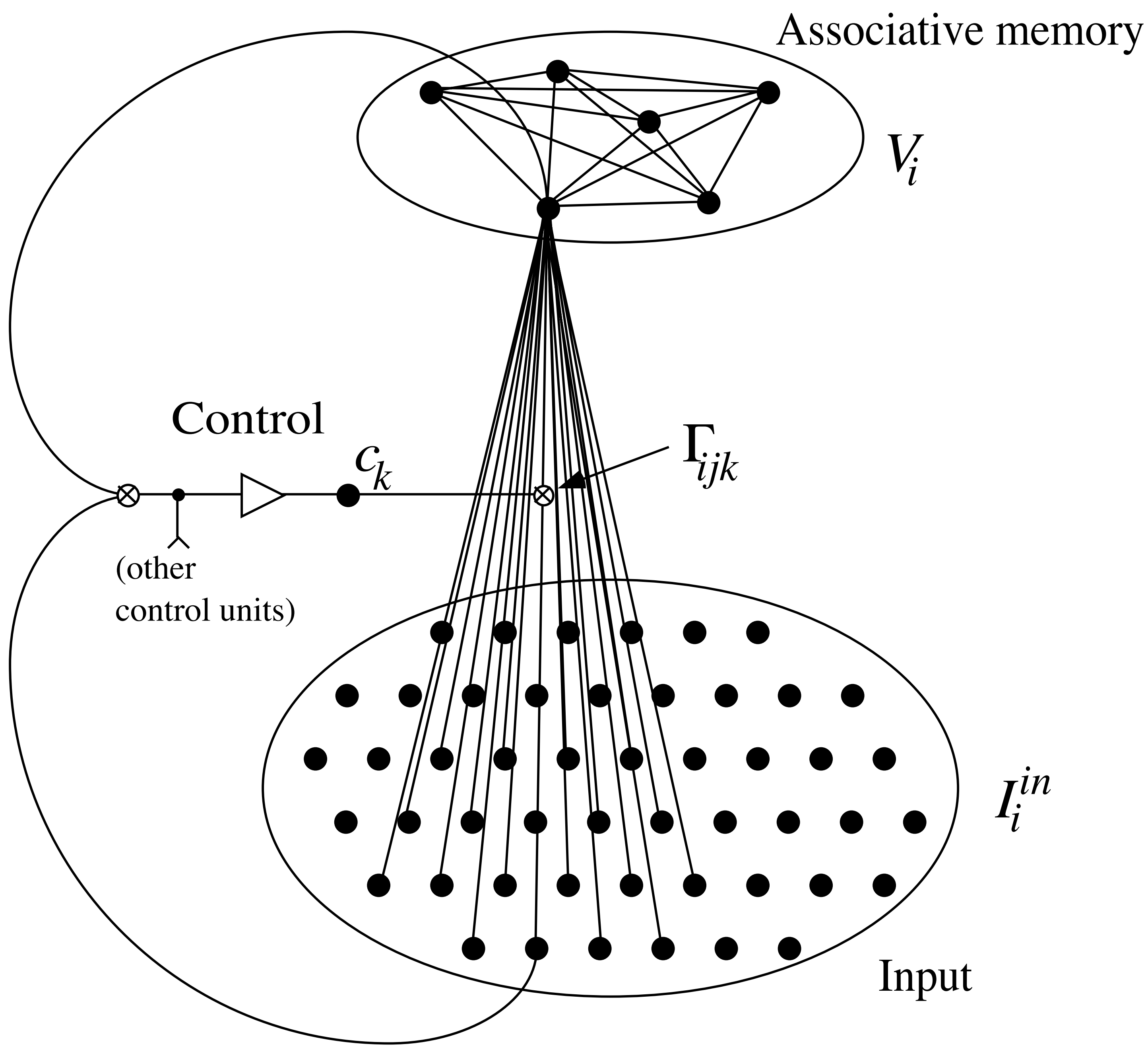
(Olshausen, Anderson, Van Essen 1993)



Dynamic routing circuit

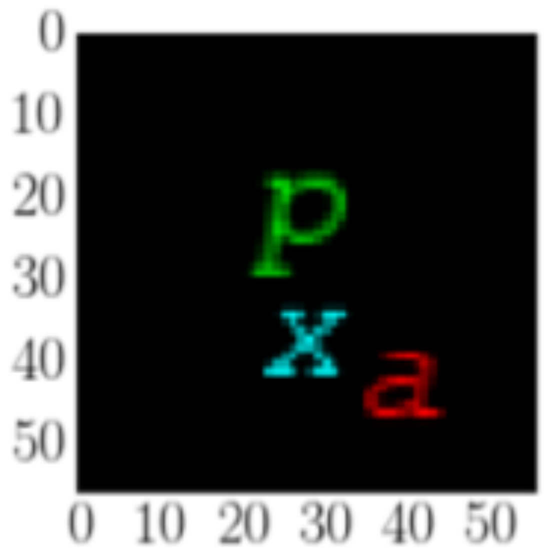


Pattern matching via dynamic routing



Factorization of shape, color and position

(Paxon Frady)



$\mathbf{u}^{x_i}$ = horizontal position x_i

$\mathbf{v}^{y_i}$ = vertical position y_j

$\mathbf{w}_c$ = color channel c

$$\mathbf{s} = \sum_{i,j,c} I(x_i, y_j, c)\, \mathbf{u}^{x_i}\, \mathbf{v}^{y_j}\, \mathbf{w}_c$$

Given $\mathbf{s}$, find $\mathbf{x}$, $\mathbf{y}$, $\mathbf{c}$ and $\mathbf{p}$ via resonator:

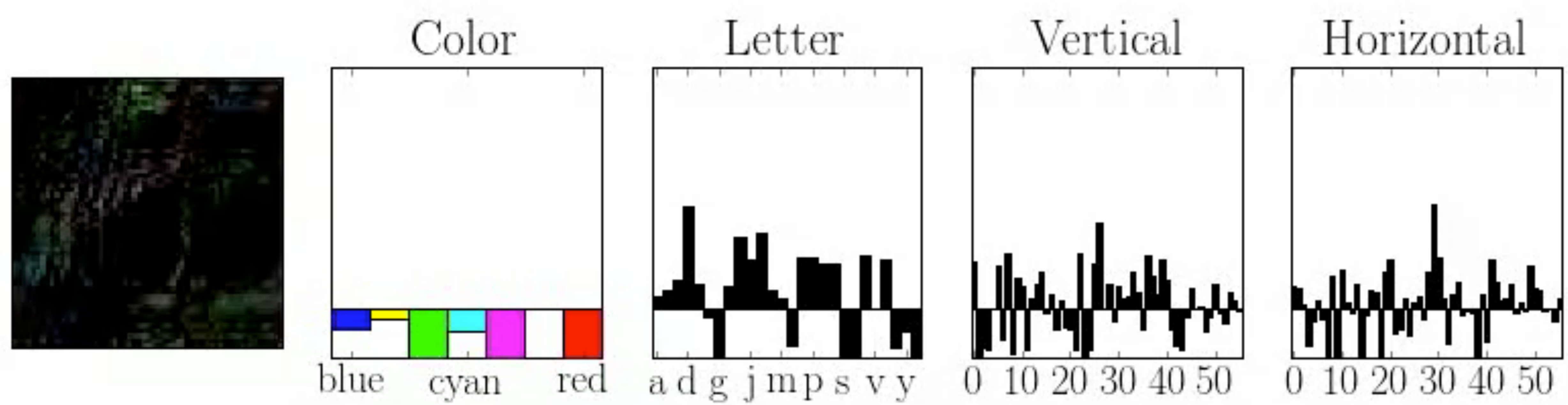
$$\hat{\mathbf{x}}_{t+1} = g(\mathbf{X}\mathbf{X}^\top(\mathbf{s} \otimes \hat{\mathbf{y}}_t^{-1} \otimes \hat{\mathbf{c}}_t^{-1} \otimes \hat{\mathbf{p}}_t^{-1}))$$ horizontal position

$$\hat{\mathbf{y}}_{t+1} = g(\mathbf{Y}\mathbf{Y}^\top(\mathbf{s} \otimes \hat{\mathbf{x}}_t^{-1} \otimes \hat{\mathbf{c}}_t^{-1} \otimes \hat{\mathbf{p}}_t^{-1}))$$ vertical position

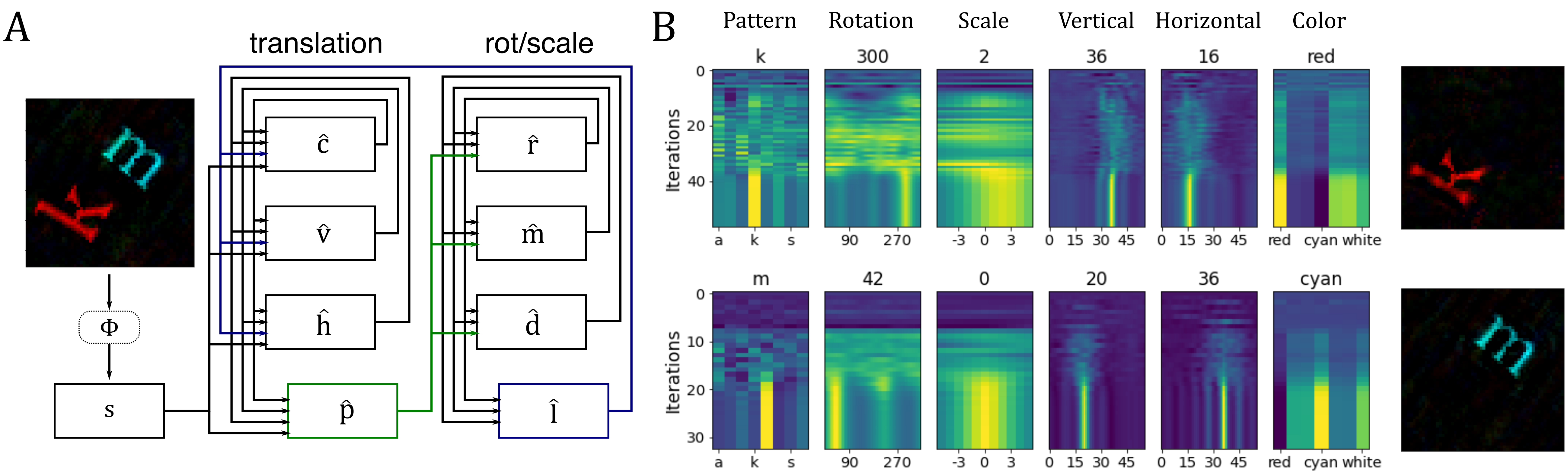
$$\hat{\mathbf{c}}_{t+1} = g(\mathbf{C}\mathbf{C}^\top(\mathbf{s} \otimes \hat{\mathbf{x}}_t^{-1} \otimes \hat{\mathbf{y}}_t^{-1} \otimes \hat{\mathbf{p}}_t^{-1}))$$ color

$$\hat{\mathbf{p}}_{t+1} = g(\mathbf{P}\mathbf{P}^\top(\mathbf{s} \otimes \hat{\mathbf{x}}_t^{-1} \otimes \hat{\mathbf{y}}_t^{-1} \otimes \hat{\mathbf{c}}_t^{-1}))$$ pattern

Visual scene analysis via factorization of HD vectors
(Paxon Frady)



Extension to translation, rotation and scaling



$$\mathbf{s} = \sum_i \mathbf{c}_{c_i} \odot \mathbf{h}^{x_i} \odot \mathbf{v}^{y_i} \odot \mathbf{\Lambda}^{-1}(\mathbf{r}^{r_i} \odot \mathbf{m}^{m_i} \odot \mathbf{d}_{p_i})$$

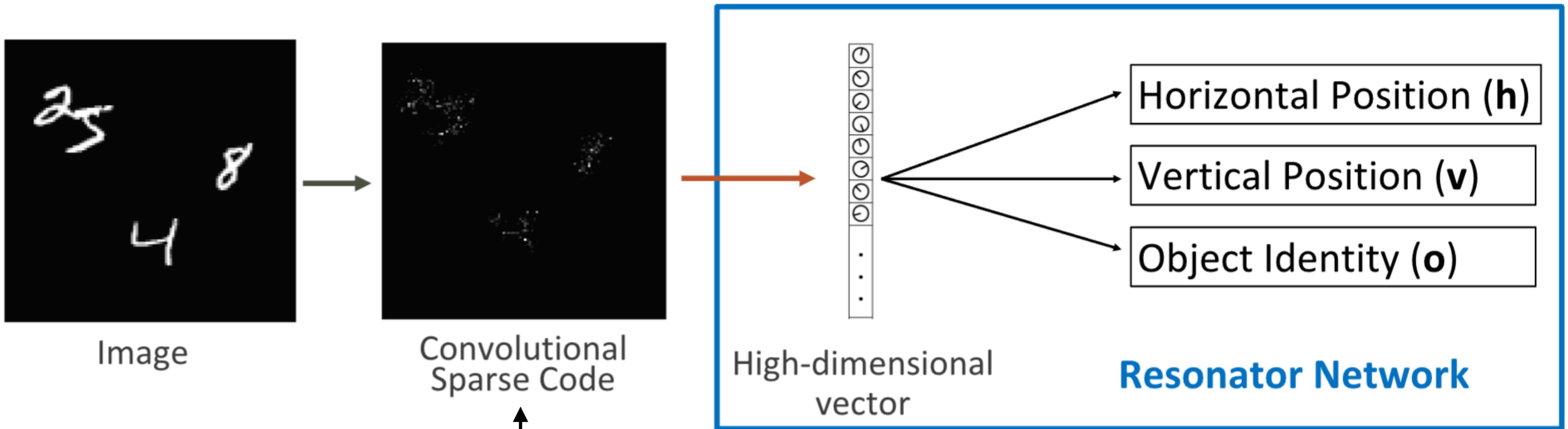
$$\hat{\mathbf{l}}(t+1) = \mathbf{\Lambda}^{-1}\big(\hat{\mathbf{r}}(t) \odot \hat{\mathbf{m}}(t) \odot \hat{\mathbf{d}}(t)\big),$$

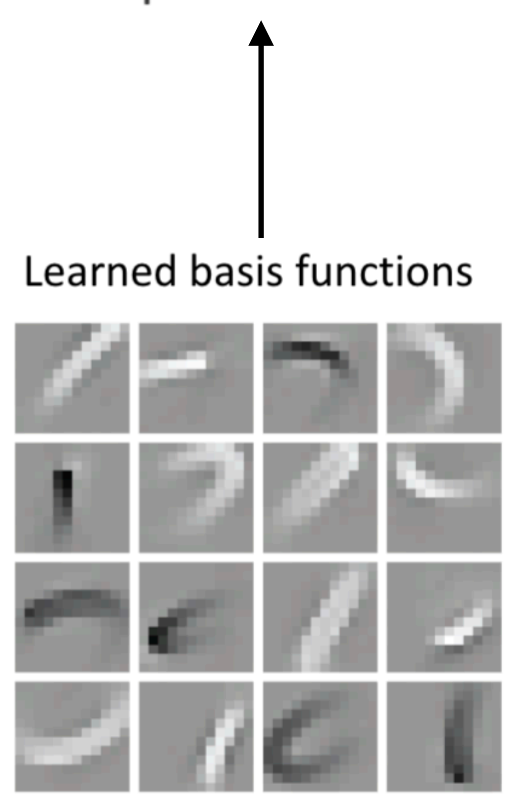
$$\hat{\mathbf{p}}(t+1) = \mathbf{\Lambda}\big(\mathbf{s} \odot \hat{\mathbf{c}}^*(t) \odot \hat{\mathbf{h}}^*(t) \odot \hat{\mathbf{v}}^*(t)\big)$$

Renner, et al. (2024). Neuromorphic visual scene understanding with resonator networks. *Nature Machine Intelligence*.

Factorization of Visual Scenes with Convolutional Sparse Coding and Resonator Networks.

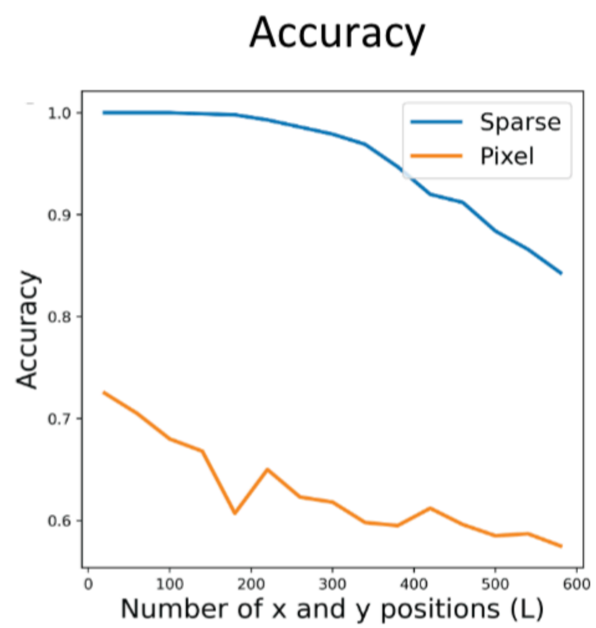
(Kymn, Mazelet, Kleyko & Olshausen. NICE 2024 Proceedings)

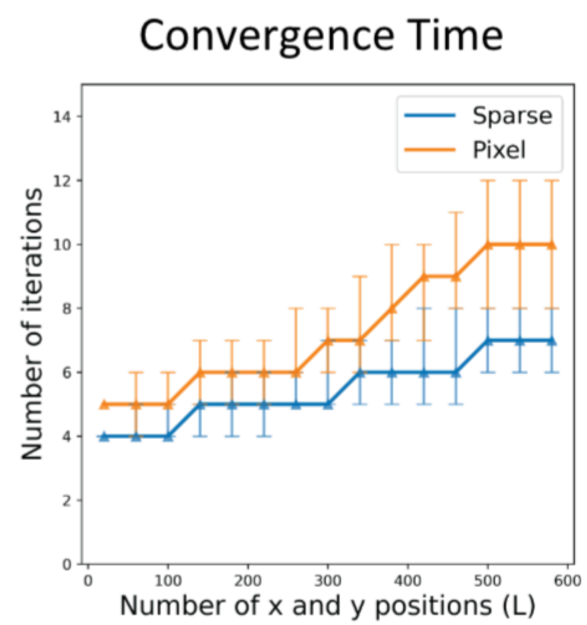


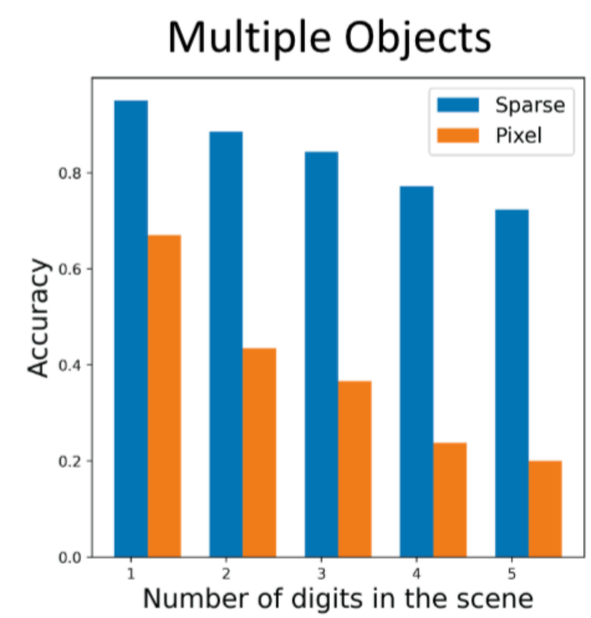


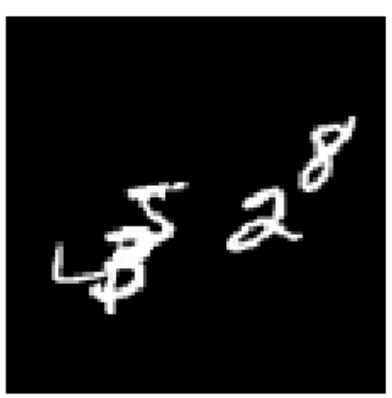
$$
\begin{aligned}
\hat{\mathbf{h}}_{t+1} &= g(\mathbf{H}\mathbf{H}^\dagger(\mathbf{z} \odot \hat{\mathbf{v}}_t^\dagger \odot \hat{\mathbf{o}}_t^\dagger)) \\
\hat{\mathbf{v}}_{t+1} &= g(\mathbf{V}\mathbf{V}^\dagger(\mathbf{z} \odot \hat{\mathbf{h}}_t^\dagger \odot \hat{\mathbf{o}}_t^\dagger)) \\
\hat{\mathbf{o}}_{t+1} &= g(\mathbf{O}\mathbf{O}^\dagger(\mathbf{z} \odot \hat{\mathbf{h}}_t^\dagger \odot \hat{\mathbf{v}}_t^\dagger))
\end{aligned}
$$

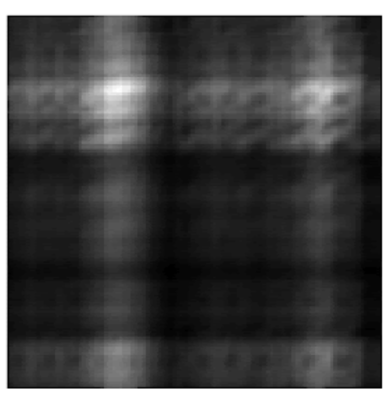
Performance

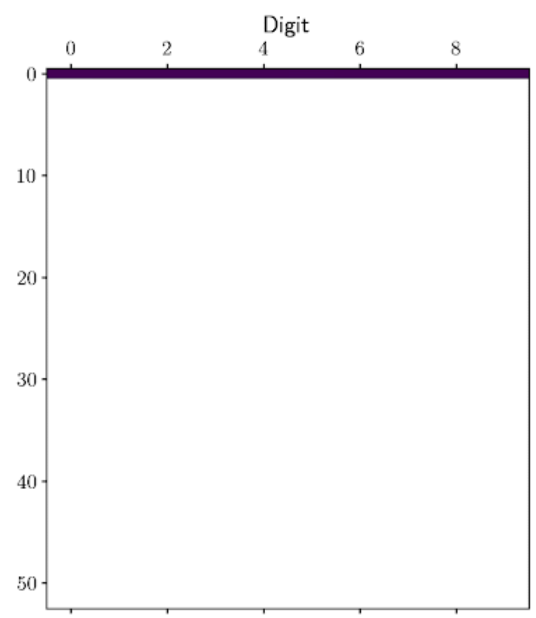


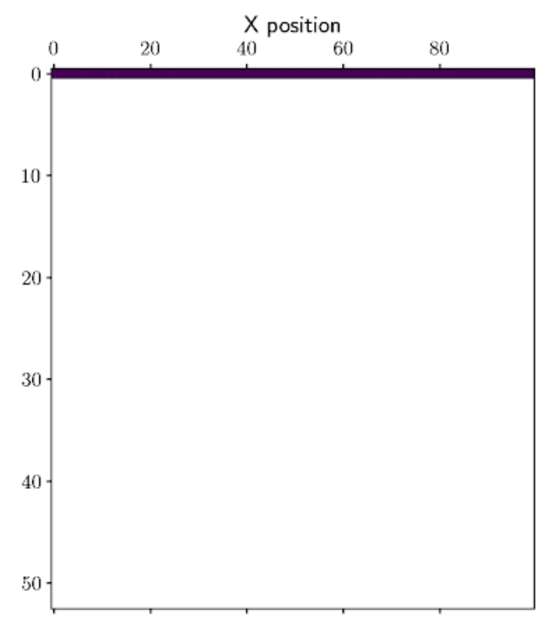


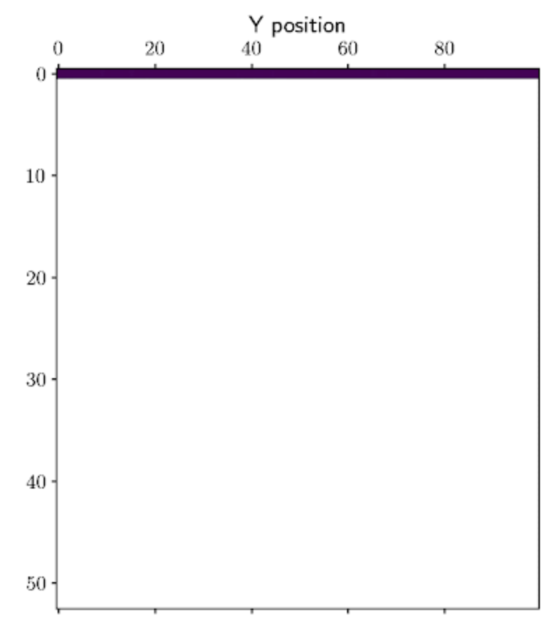


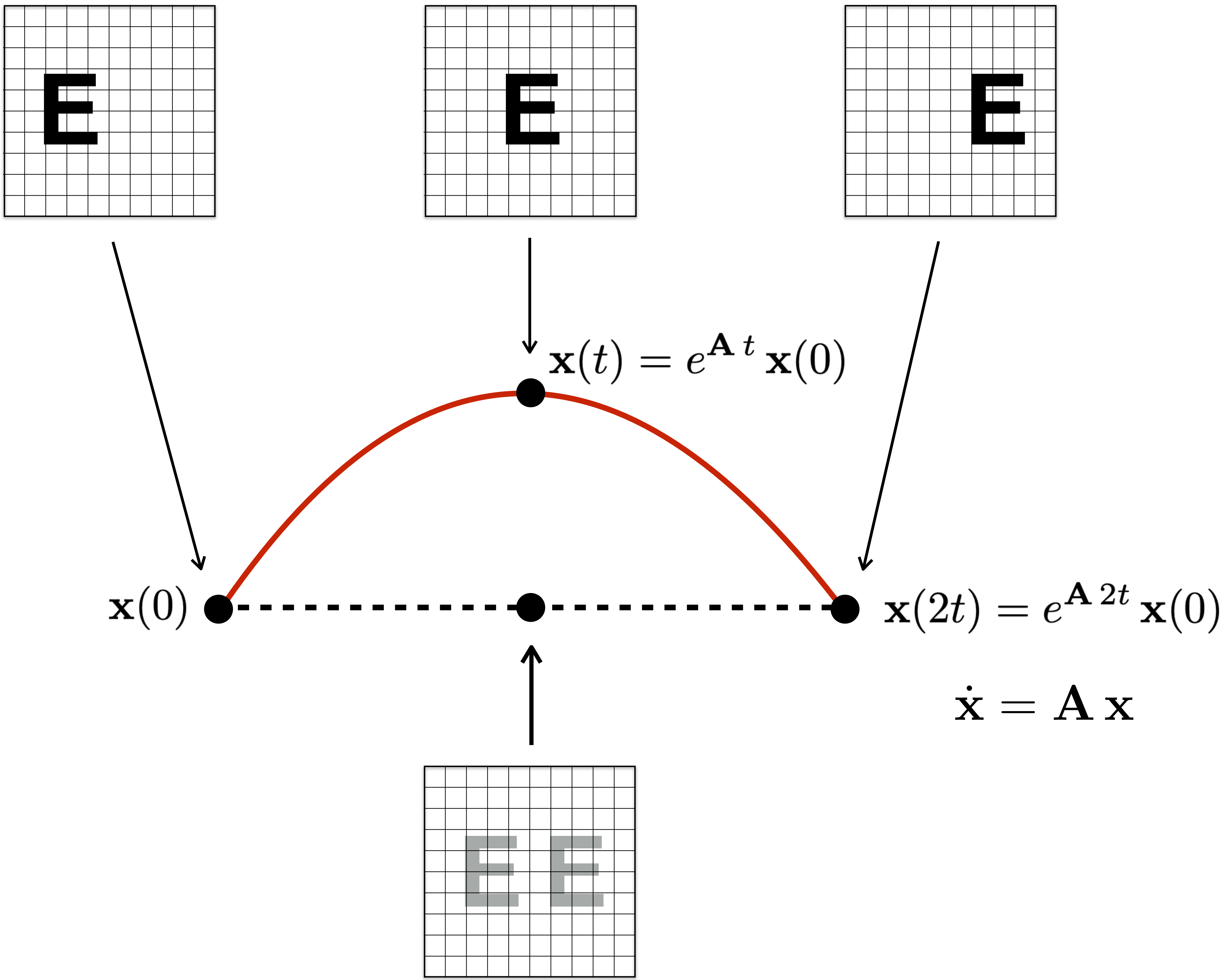










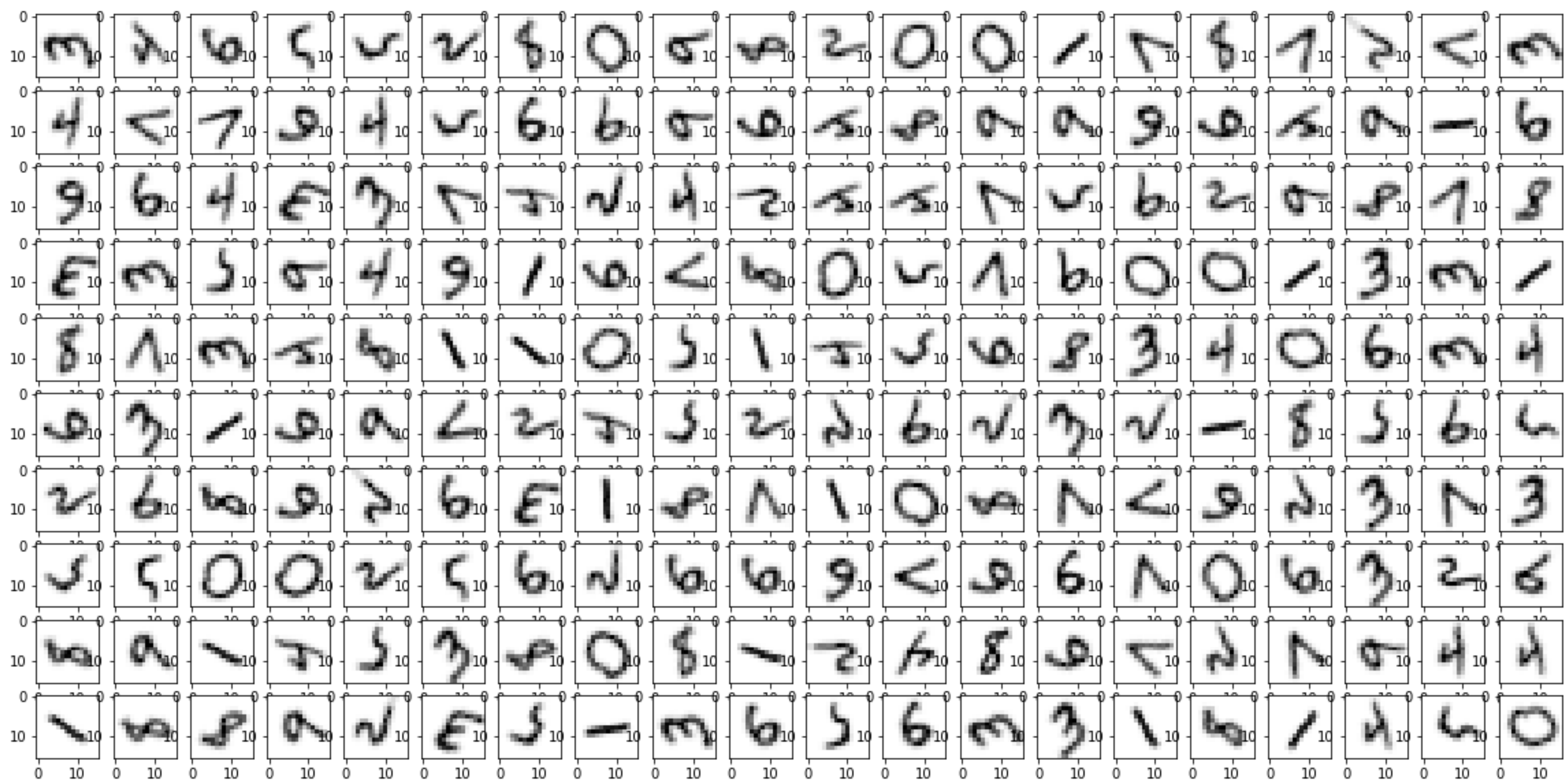


E
E
E
$\mathbf{x}(t) = e^{\mathbf{A}\,t}\,\mathbf{x}(0)$
$\mathbf{x}(0)$
$\mathbf{x}(2t) = e^{\mathbf{A}\,2t}\,\mathbf{x}(0)$
$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x}$
E E

Learning to separate shape and transformations via Lie group operators and sparse coding

(Ho Yin Chau, Yubei Chen, Frank Qiu)

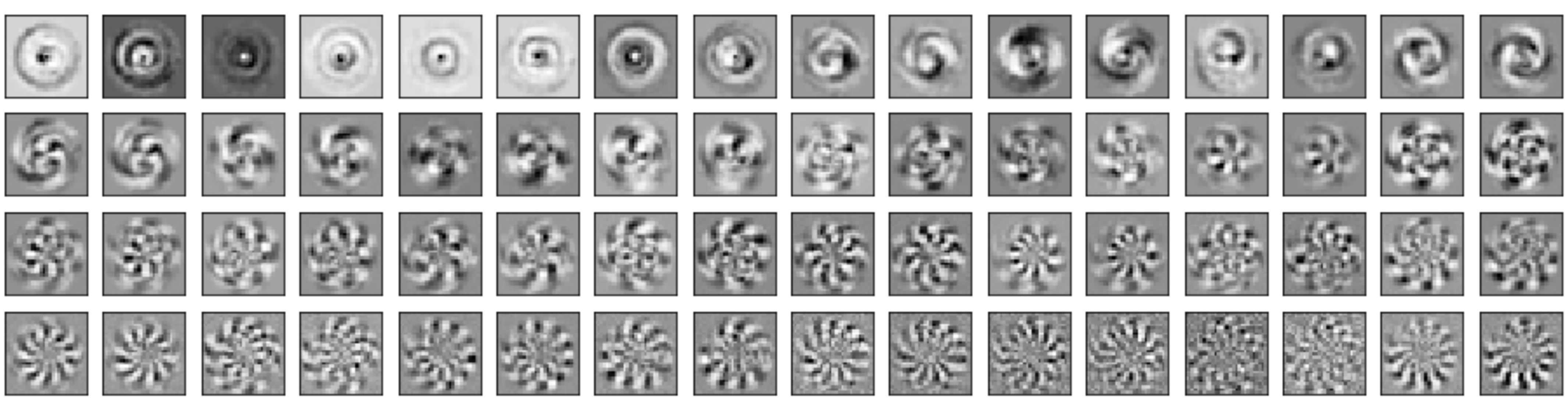
$$\mathbf{I} = \mathbf{T}(s)\,\boldsymbol{\Phi}\,\alpha + \epsilon$$
$$= e^{\mathbf{A}s}\,\boldsymbol{\Phi}\,\alpha + \epsilon$$



$$\begin{aligned}\mathbf{T}(s) &= e^{\mathbf{A}s} \\ &= \mathbf{W}\, e^{\boldsymbol{\Sigma} s}\, \mathbf{W}^T \;=\; \mathbf{W}\,\mathbf{R}(s)\,\mathbf{W}^T\end{aligned}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0 & -\omega_1 & & & \\ \omega_1 & 0 & & & \\ & & \ddots & & \\ & & & 0 & -\omega_{D/2} \\ & & & \omega_{D/2} & 0 \end{bmatrix} \qquad \mathbf{R}(s) = \begin{bmatrix} \cos(\omega_1 s) & -\sin(\omega_1 s) & & & \\ \sin(\omega_1 s) & \cos(\omega_1 s) & & & \\ & & \ddots & & \\ & & & \cos\left(\omega_{D/2} s\right) & -\sin\left(\omega_{D/2} s\right) \\ & & & \sin\left(\omega_{D/2} s\right) & \cos\left(\omega_{D/2} s\right) \end{bmatrix}$$

$$\mathbf{I} = \mathbf{W}\,\mathbf{R}(s)\,\mathbf{W}^T\,\boldsymbol{\Phi}\,\alpha + \epsilon$$



Results