Strong generalization from small brains

Bruno Olshausen
Helen Wills Neuroscience Institute, School of Optometry
Redwood Center for Theoretical Neuroscience
UC Berkeley

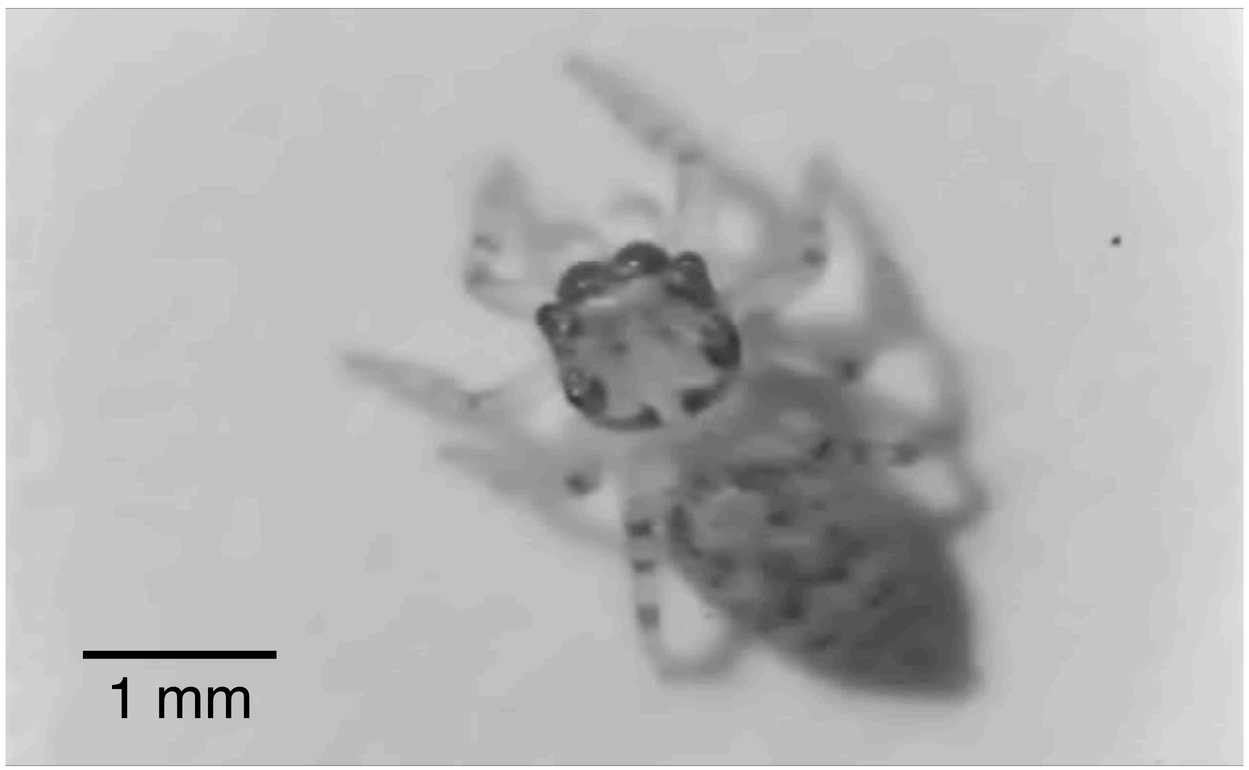

EARL WARREN HALL
Pentti Kanerva
Chris Kymn
Sonia Mazelet
Fritz Sommer
Anthony Thomas
VANS
Redwood Center for Theoretical Neuroscience

Main points

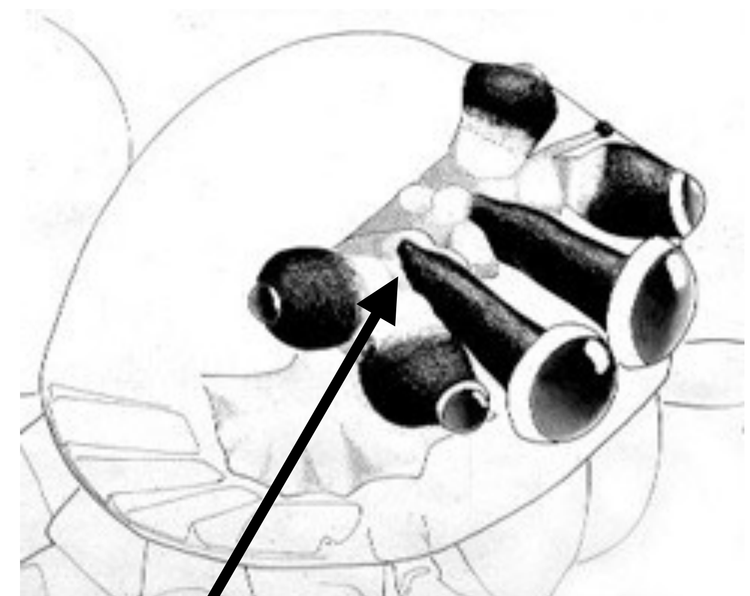
- Animal intelligence
- Physics of computation
- Perception as factorization
- Equivariant representation

Animal intelligence

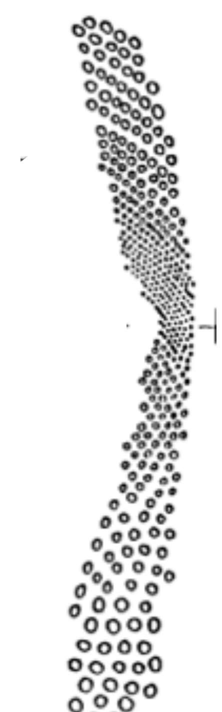
Jumping spider



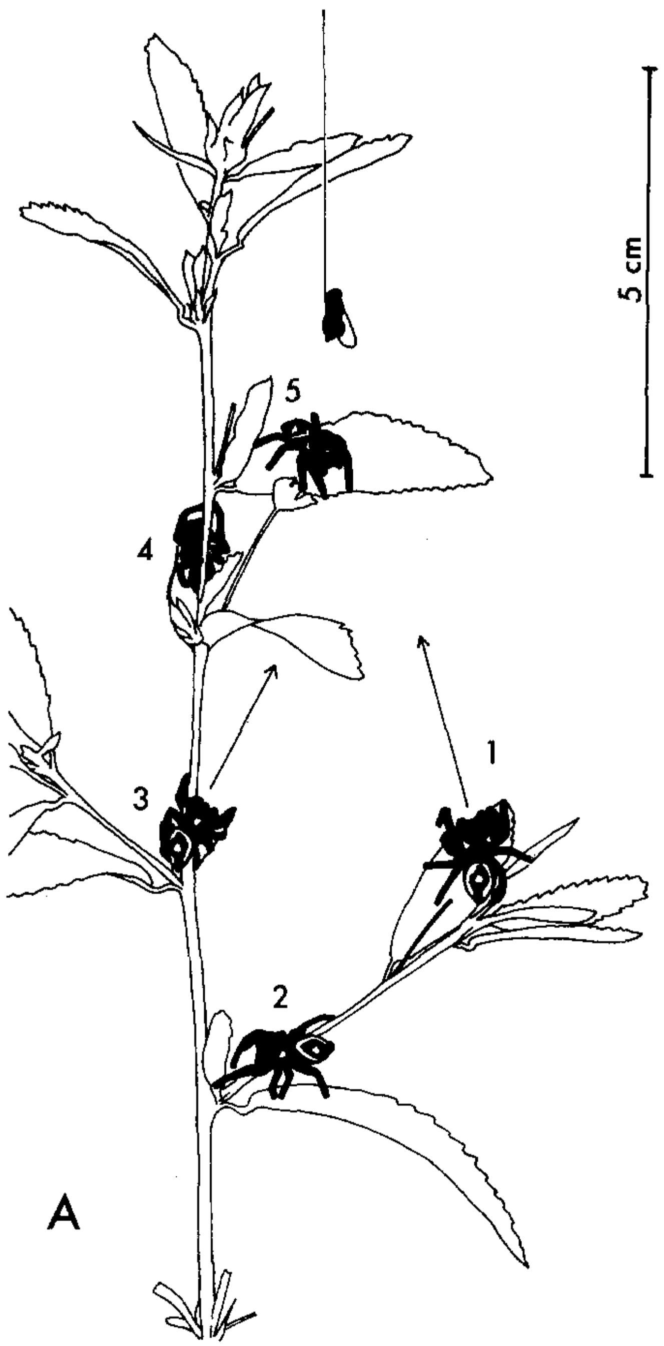
(Bair & Olshausen, 1991)

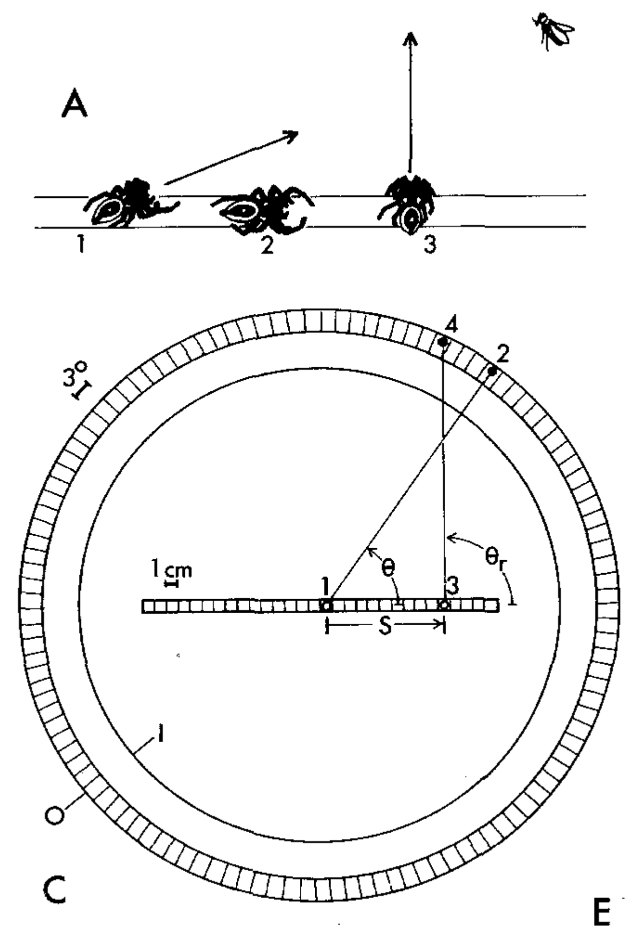


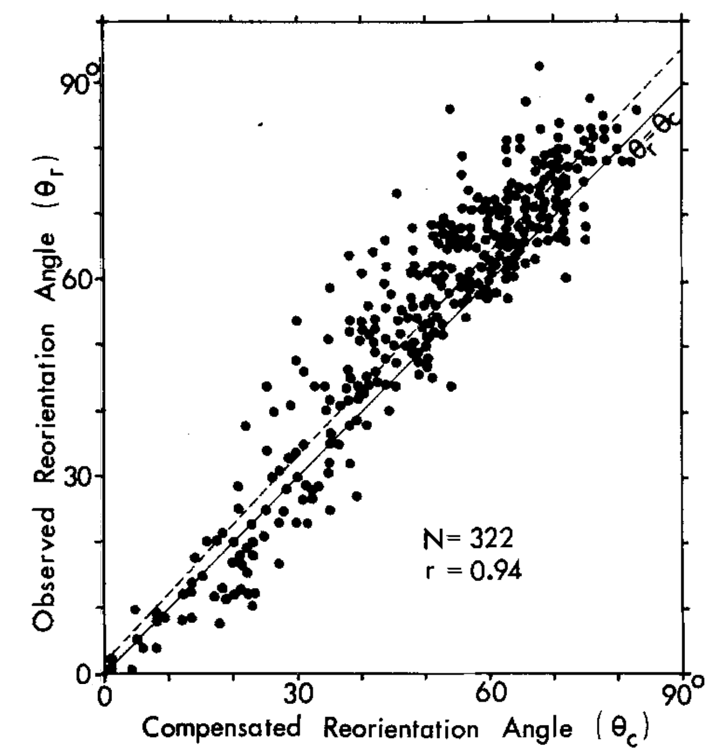
(Wayne Maddison)



Orientation by Jumping Spiders During the Pursuit of Prey

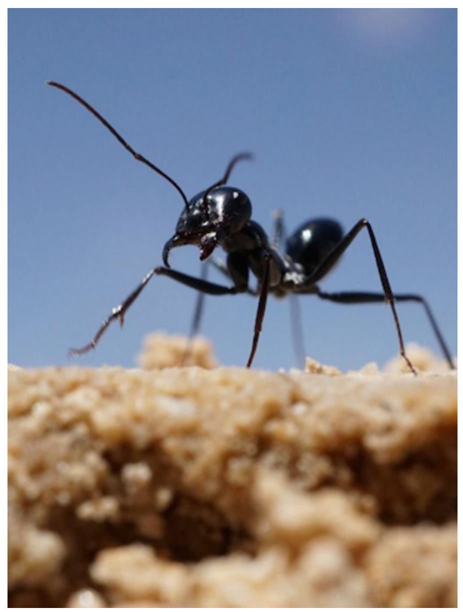


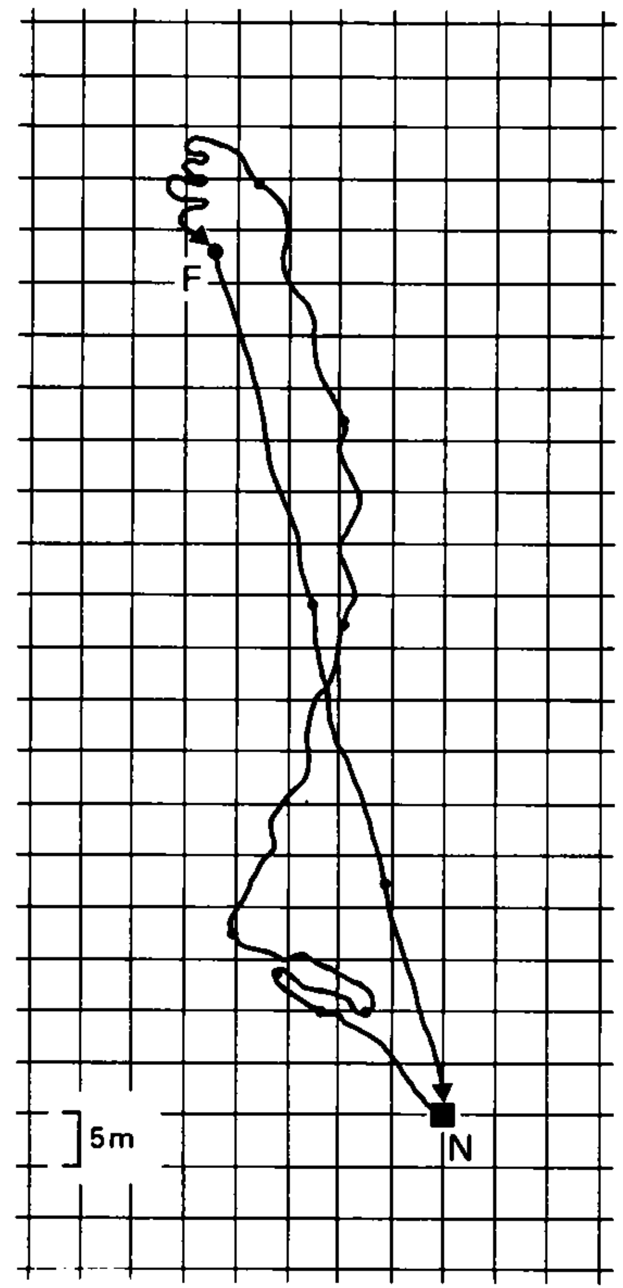




(D.E. Hill, 1979)

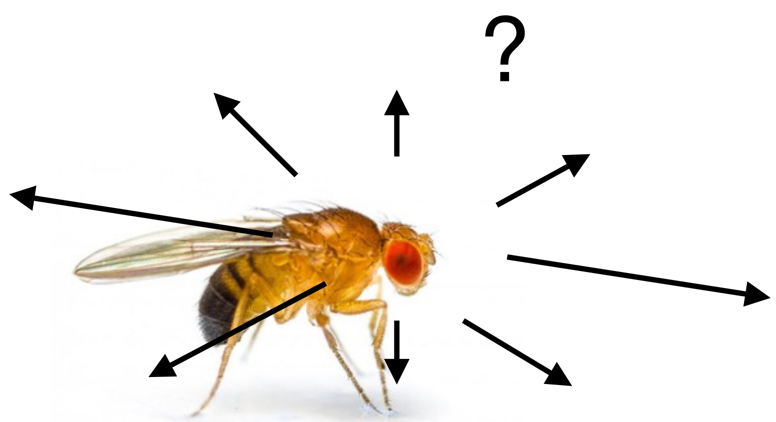
Path integration in desert ants

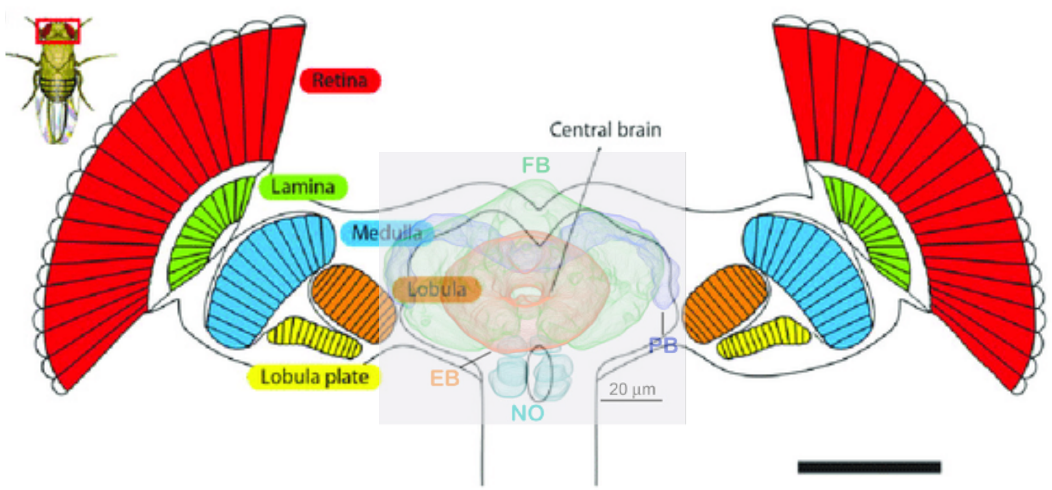




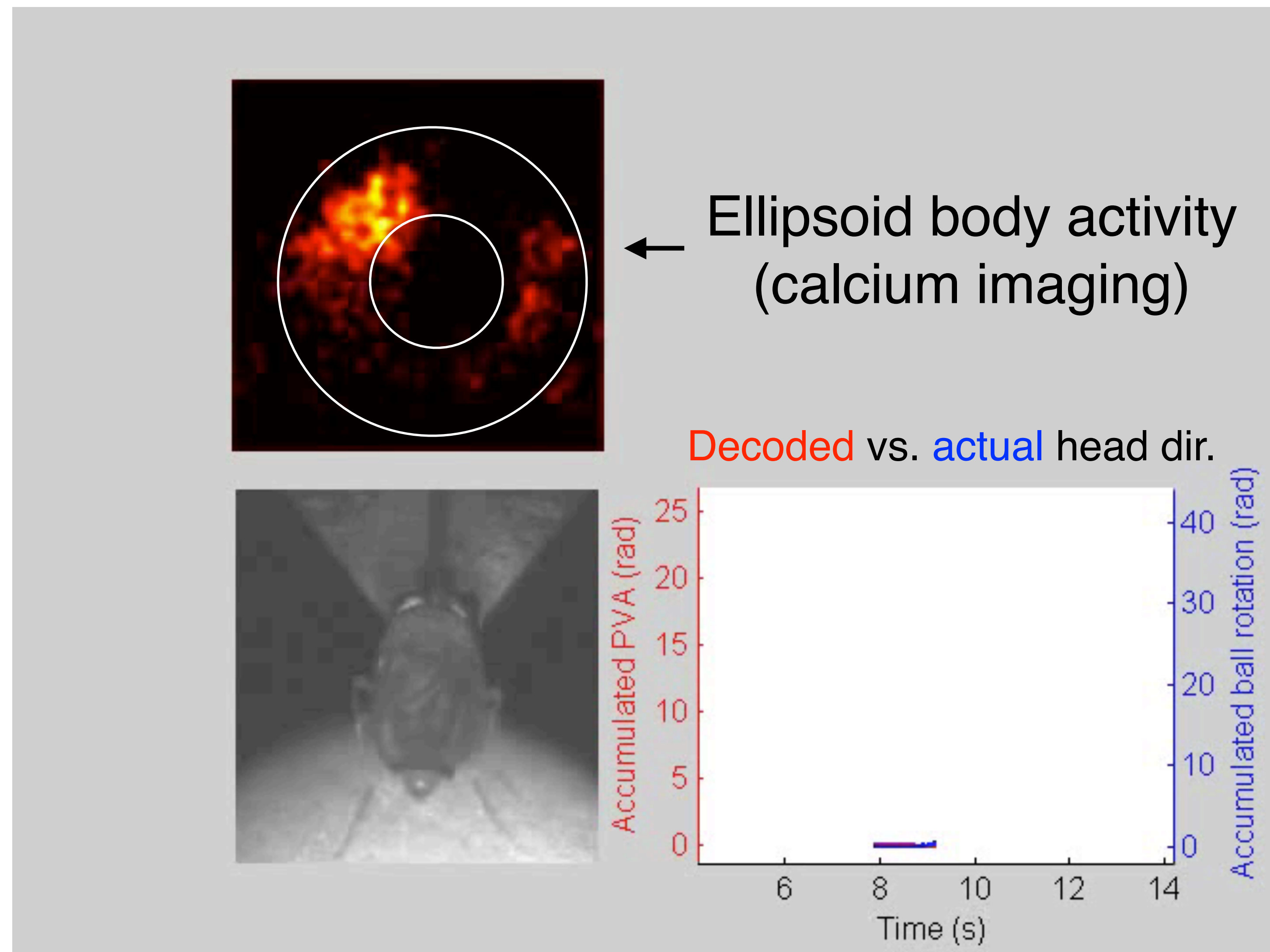
(R. Wehner, S. Wehner, 1986)

Navigation in fruit flies





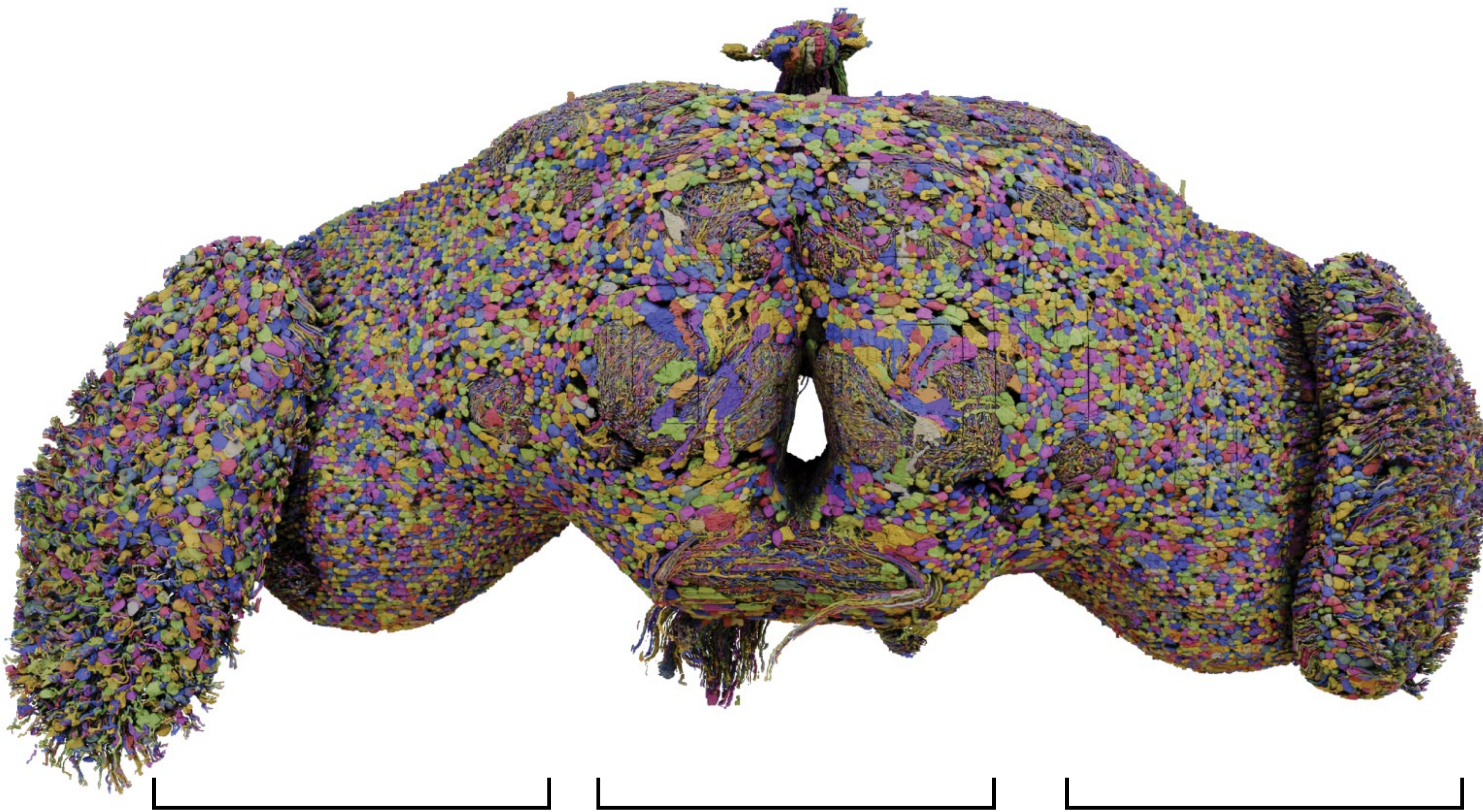
Head-direction cells in ellipsoid body of Drosophila



(Seelig & Jayaraman 2015)

Entire fly brain connectome (139,355 neurons)

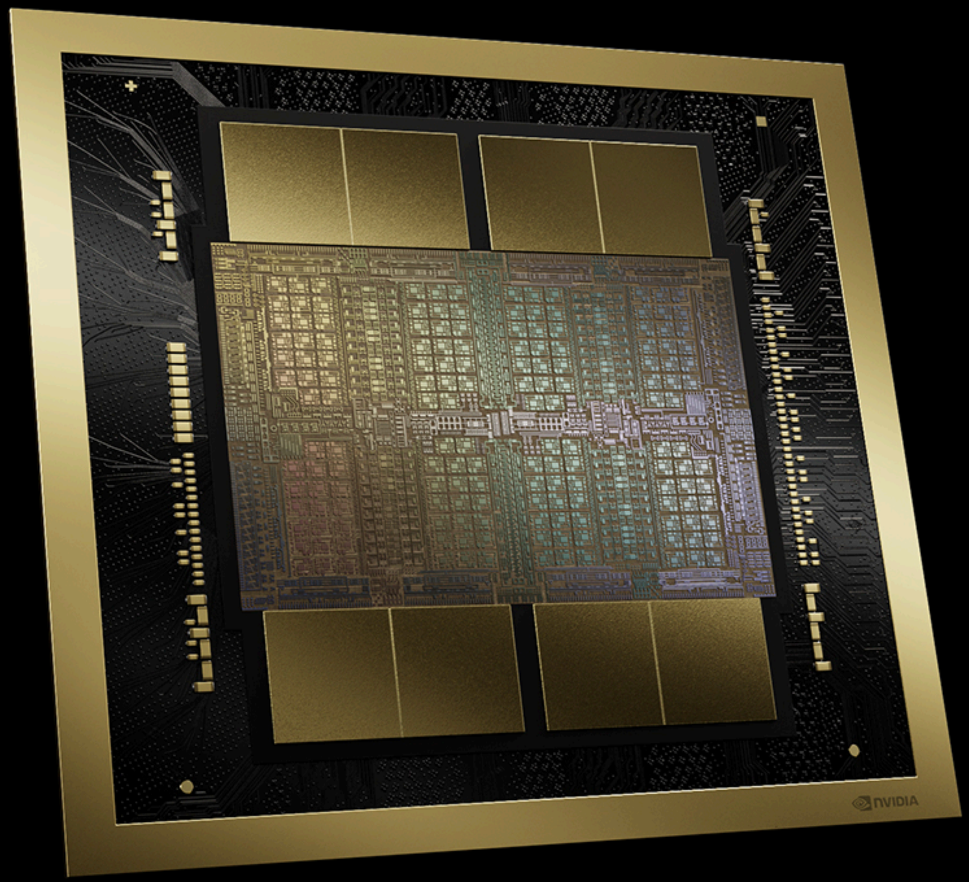
(Dorkenwald et al., 2024)



Physics of computation

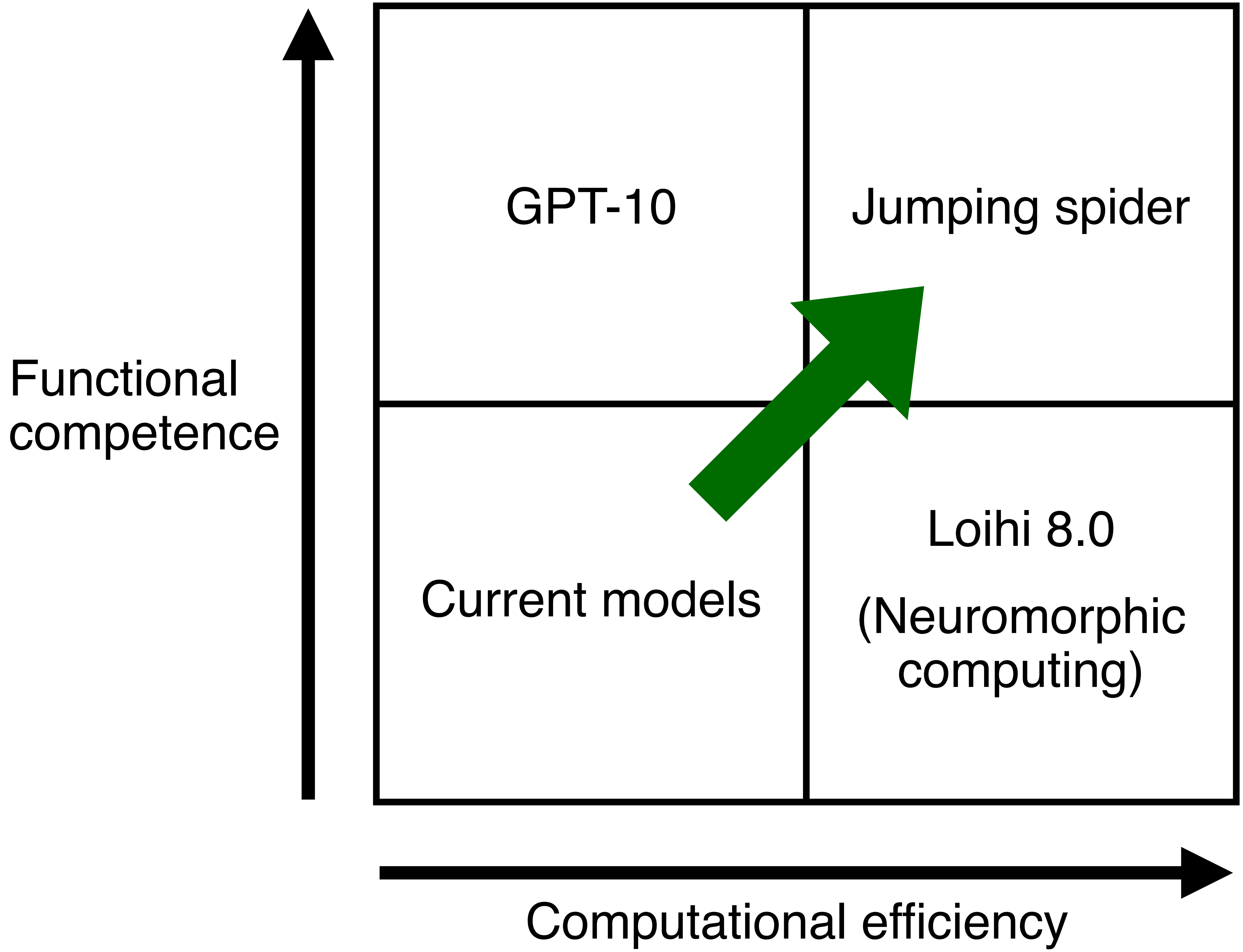
nVidia Blackwell GPU

200 billion transistors
1 kW



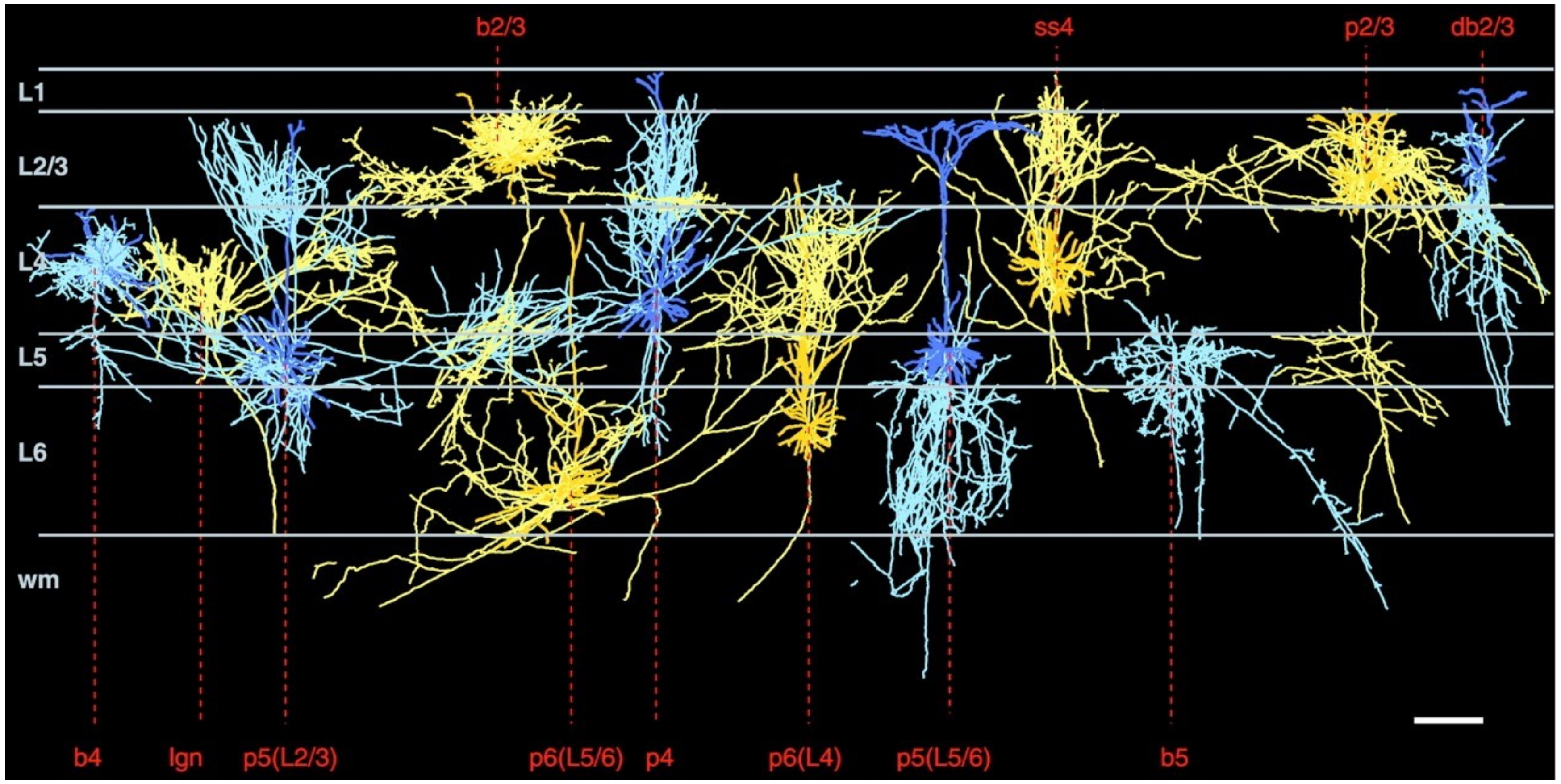
Jumping spider

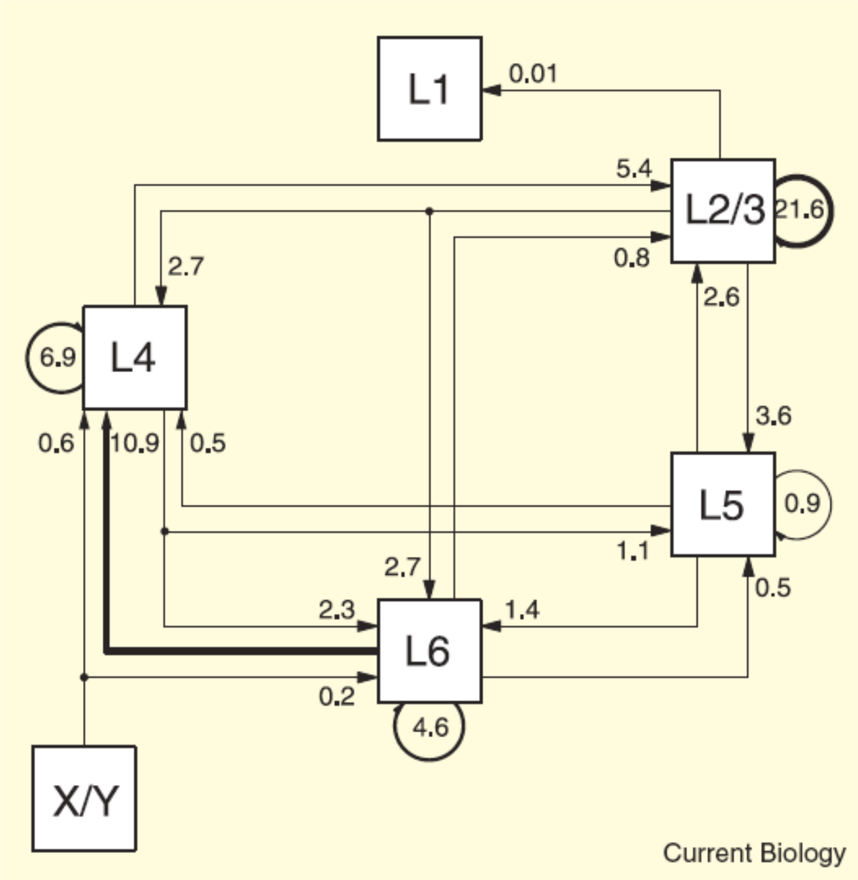
ca. 100,000 neurons
1 fly/day


GPT-10
Jumping spider
Functional competence
Current models
Loihi 8.0
(Neuromorphic computing)
Computational efficiency

Recurrent circuits are pervasive throughout cortex

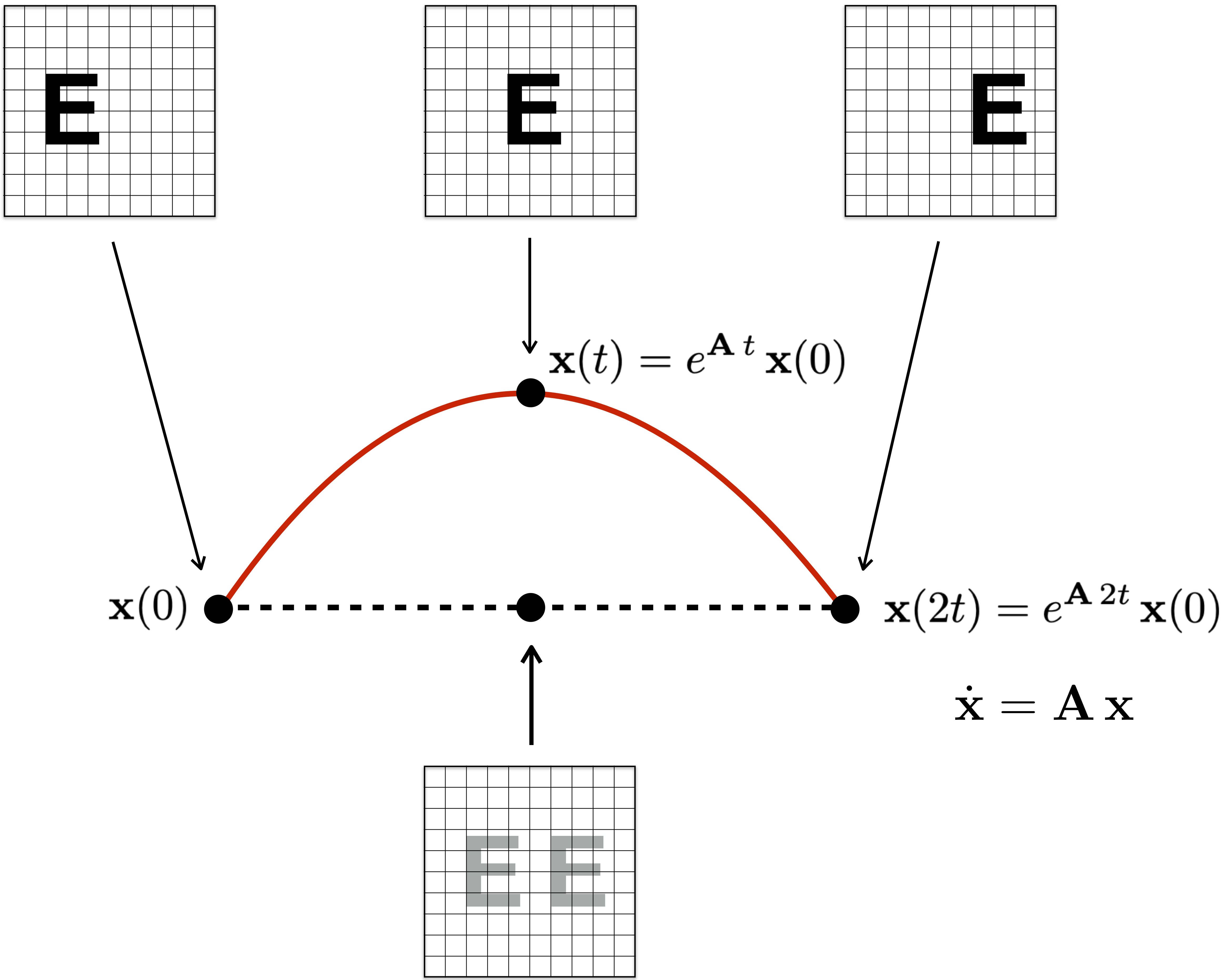
Cortical microcircuit





(Douglas and Martin, 2007)

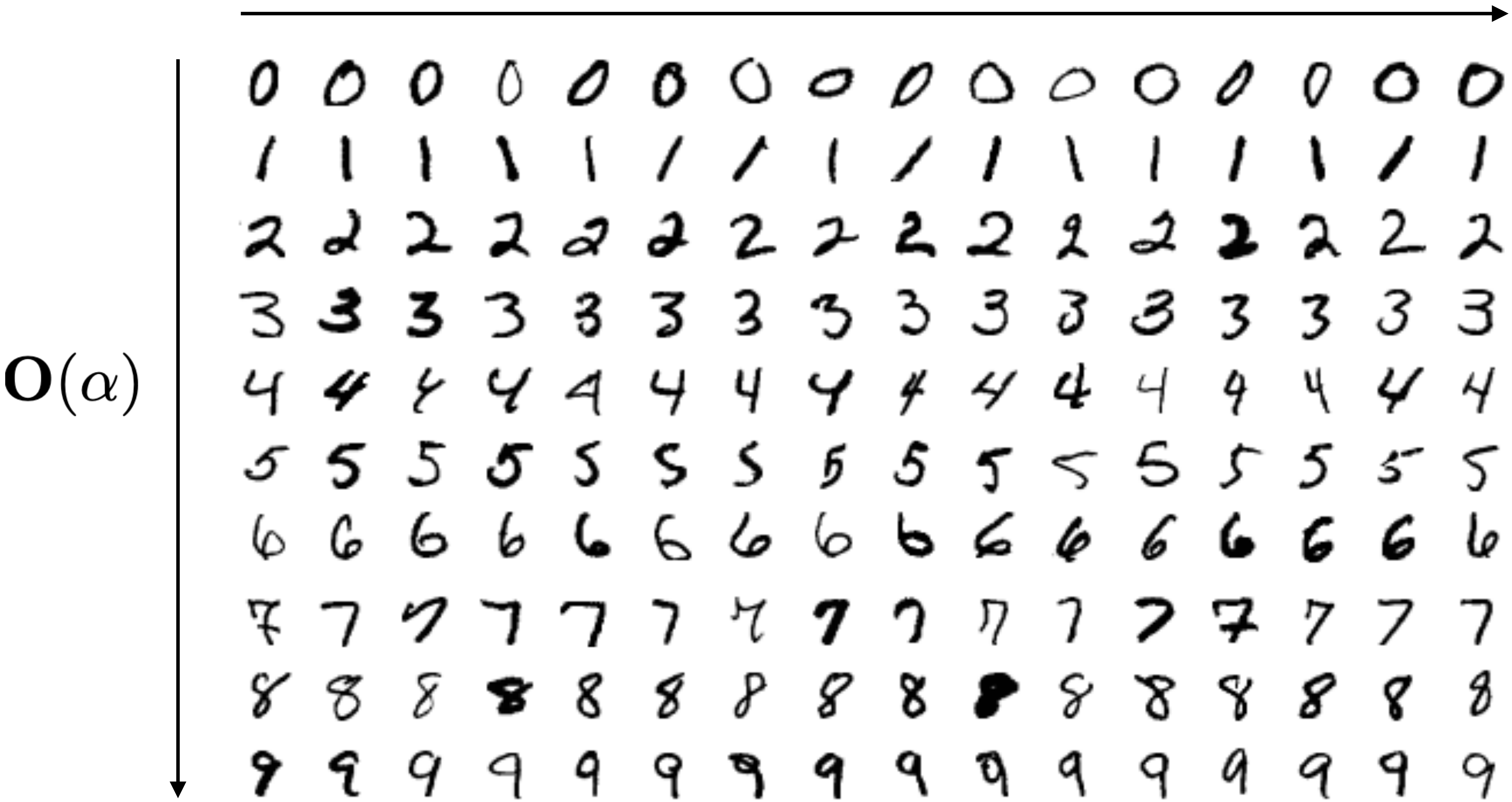
Perception as factorization



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

Example: MNIST dataset

$$\mathbf{T}(s) = e^{\mathbf{A}\,s}$$



$$\mathbf{I}(s, \alpha) = \mathbf{T}(s)\,\mathbf{O}(\alpha)$$

We can reformulate this as vector factorization

$$\mathbf{I} = \mathbf{T}(s)\,\mathbf{O}(\alpha)$$

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

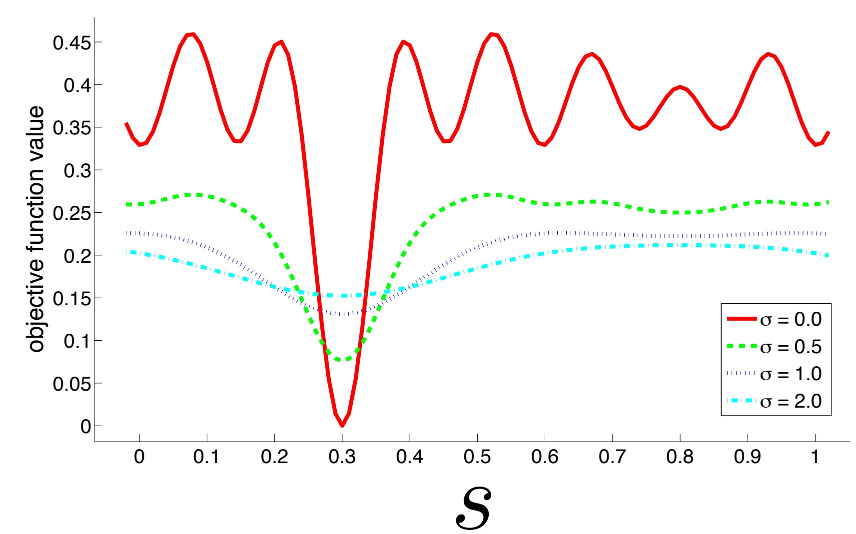
$$\mathbf{R}(s) = \begin{bmatrix} e^{i\omega_1 s} & & & \\ & e^{i\omega_2 s} & & \\ & & \ddots & \\ & & & e^{i\omega_{D/2} s} \end{bmatrix}$$

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

$$\tilde{\mathbf{I}} = \mathbf{R}(s)\,\tilde{\boldsymbol{\Phi}}\alpha$$

$$\tilde{\mathbf{I}} = \mathbf{z}(s) \odot \tilde{\mathbf{O}}(\alpha)$$

Equivariant part Invariant part



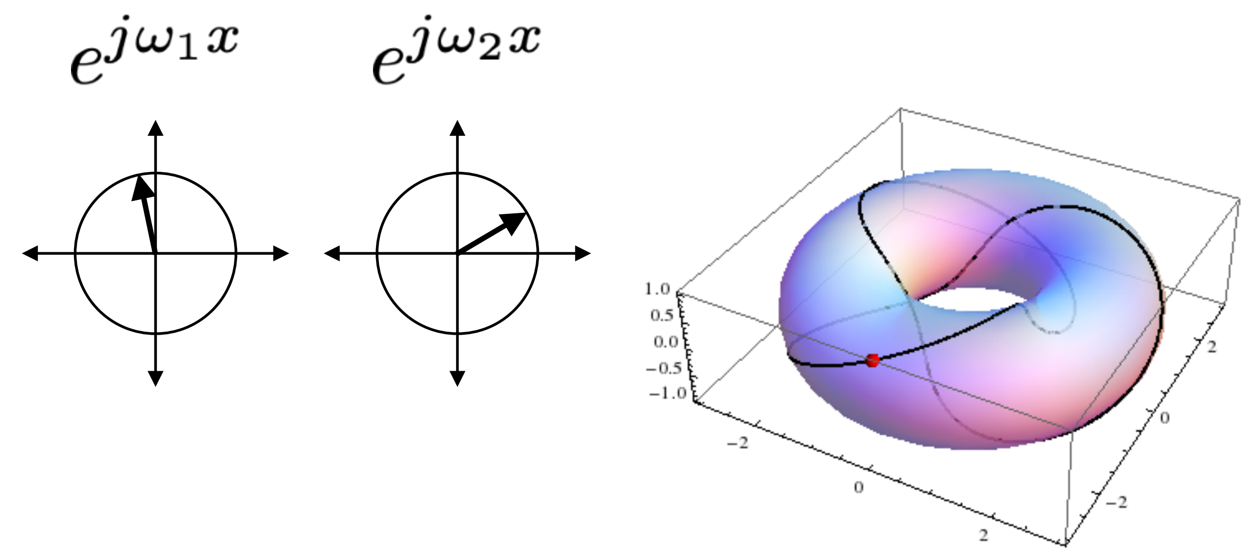
Representing position with complex-valued vectors

- Base vector:

$$\mathbf{z} = \begin{bmatrix} e^{j\omega_1} \\ e^{j\omega_2} \\ \vdots \\ e^{j\omega_N} \end{bmatrix}$$

- Value x is represented as:

$$\mathbf{z}(x) = \begin{bmatrix} e^{j\omega_1 x} \\ e^{j\omega_2 x} \\ \vdots \\ e^{j\omega_N x} \end{bmatrix}$$

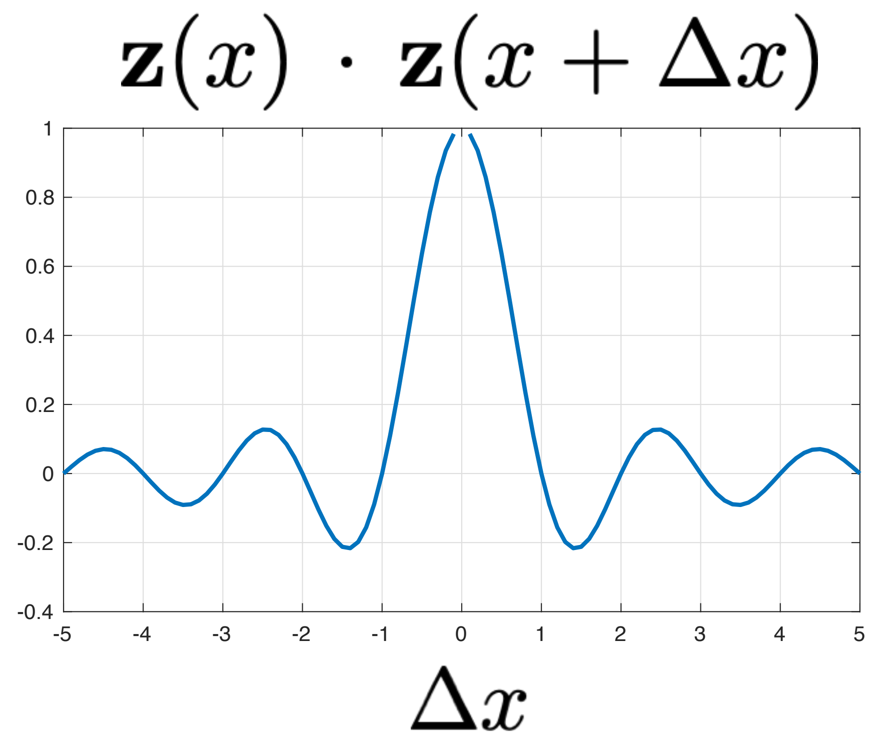


Frady EP, Kleyko D, Kymn CJ, Olshausen BA, Sommer FT (2022). Computing on Functions Using Randomized Vector Representations (in brief). In: *Proceedings of the 2022 Annual Neuro-Inspired Computational Elements Conference.*

Frady EP, Kleyko D, Kymn CJ, Olshausen BA, Sommer FT (2021). Computing on Functions Using Randomized Vector Representations. *arXiv:2109.03429*

Representing position with complex-valued vectors

Similarity kernel



Vector multiplication corresponds to variable addition

$$\mathbf{z}(x) \odot \mathbf{z}(y) = \mathbf{z}(x + y)$$

Frady EP, Kleyko D, Kymn CJ, Olshausen BA, Sommer FT (2022). Computing on Functions Using Randomized Vector Representations (in brief). In: *Proceedings of the 2022 Annual Neuro-Inspired Computational Elements Conference.*

Frady EP, Kleyko D, Kymn CJ, Olshausen BA, Sommer FT (2021). Computing on Functions Using Randomized Vector Representations. *arXiv:2109.03429*

Attractor dynamics

$$\hat{\mathbf{z}}_{t+1} = \sigma(\mathbf{Z}\mathbf{Z}^{\dagger}\hat{\mathbf{z}}_t)$$

$$\hat{\mathbf{O}}_{t+1} = \sigma(\tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}^{\dagger}\hat{\mathbf{O}}_t)$$

$$\mathbf{Z} = \begin{bmatrix} | & | & & | \\ \mathbf{z}(0) & \mathbf{z}(1) & \ldots & \mathbf{z}(M) \\ | & | & & | \end{bmatrix}$$

$$\tilde{\mathbf{\Phi}} = \begin{bmatrix} | & | & & | \\ \tilde{\mathbf{O}}_1 & \tilde{\mathbf{O}}_2 & \ldots & \tilde{\mathbf{O}}_N \\ | & | & & | \end{bmatrix}$$

See: Noest (1987). Phasor neural networks. *NIPS proceedings.*

Attractor dynamics for factorization

Given $\tilde{\mathbf{I}} = \mathbf{z}(s) \odot \tilde{\mathbf{O}}_n$

$$\hat{\mathbf{z}}_{t+1} = \sigma(\mathbf{Z}\mathbf{Z}^\dagger\, \tilde{\mathbf{I}} \odot \hat{\mathbf{O}}_t^\dagger)$$

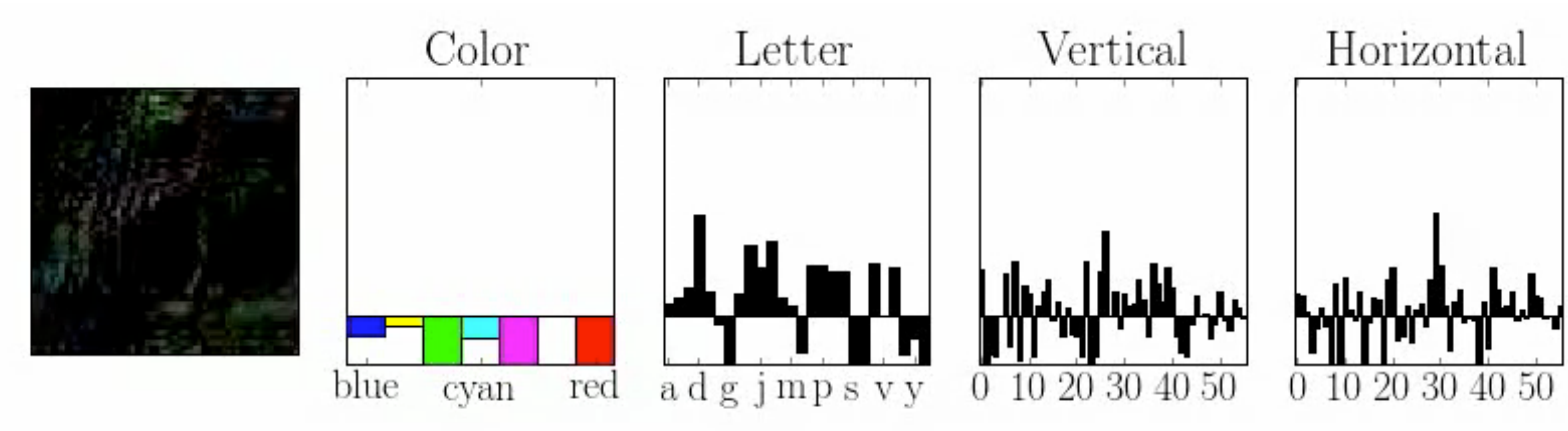
$$\hat{\mathbf{O}}_{t+1} = \sigma(\tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}^\dagger\, \tilde{\mathbf{I}} \odot \hat{\mathbf{z}}_t^\dagger)$$

"resonator network"

Frady EP, Kent S, Olshausen BA & Sommer FT (2020) Resonator Networks for factoring distributed representations of data structures. *Neural Computation* (in press) https://arxiv.org/abs/2007.03748

Kent S, Frady EP, Sommer FT & Olshausen BA (2020) Resonator Networks outperform optimization methods at solving high-dimensional vector factorization. *Neural Computation* (in press) https://arxiv.org/abs/1906.11684

Visual scene analysis via vector factorization



7 colors x 26 letters x 50 vertical x 50 horizontal = 455,000 combinations per object

Complexity of representation and computation is 7 + 26 + 50 + 50

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

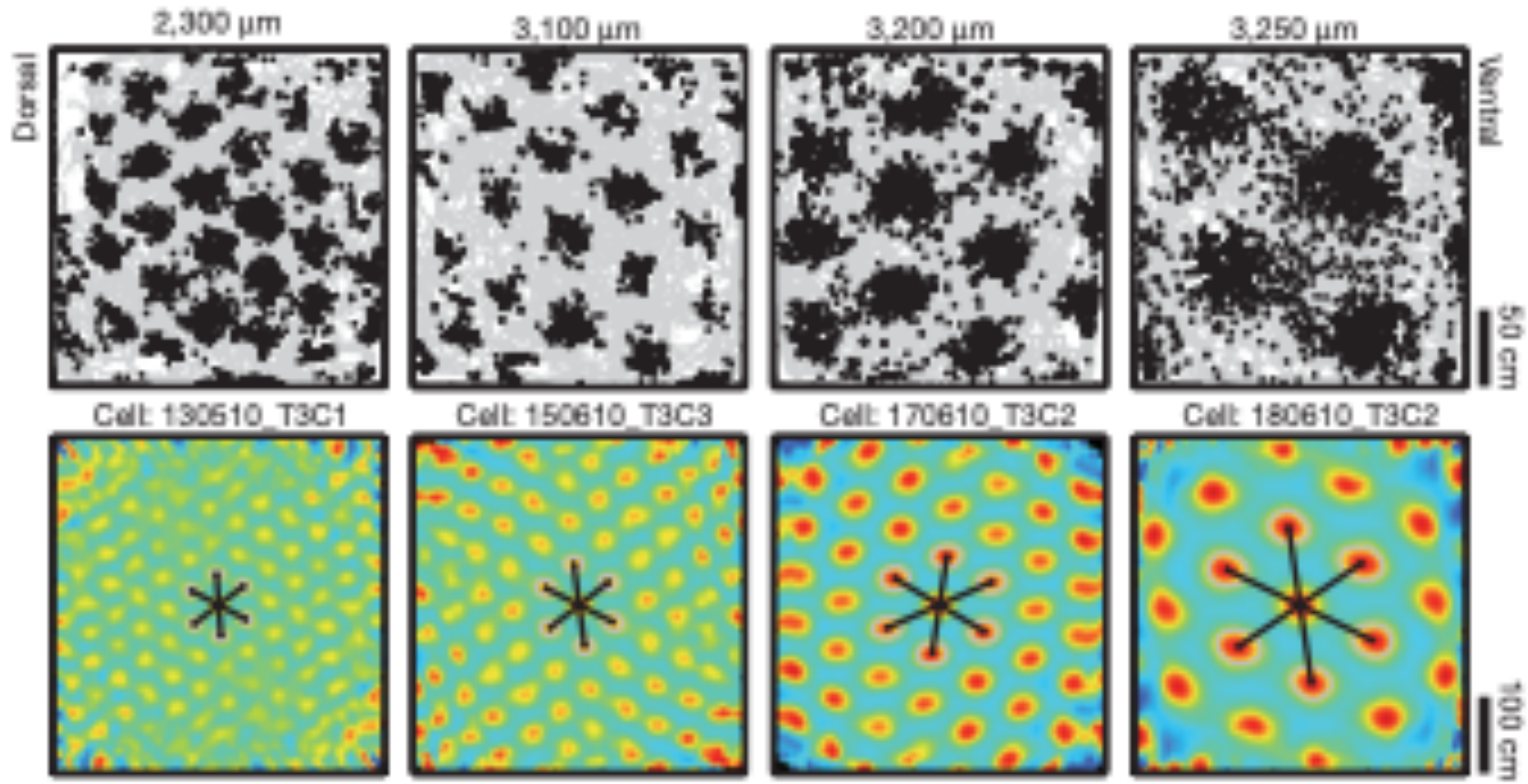
Equivariant representation

High-capacity, error-correcting representation of spatial position

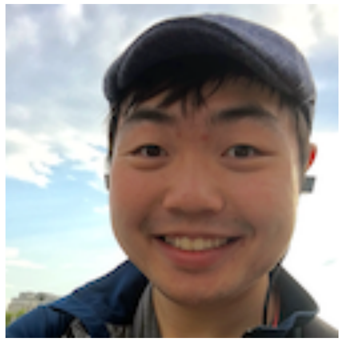
Grid cell responses

(entorhinal cortex)

Autocorrelogram



Recording of several neurons reveals multiple scales of encoding



Chris Kymn

Computing with Residue Numbers in High-Dimensional Representation.

arXiv:2311.04872. (*Neural Computation,* to appear)

- Key idea: Represent an integer in terms of its remainder relative to a set of pairwise co-prime integers $\{m_1, m_2 \ldots m_k\}$

  Example: 41 = {2, 1, 6} (mod 3, 5, 7)

- Chinese remainder theorem: Residue numbers are unique for all values of x, $0 \leq x \leq M - 1$, with $M = m_1 \times m_2 \times \ldots \times m_k$.

- Arithmetic operations are element-wise: (no carry)

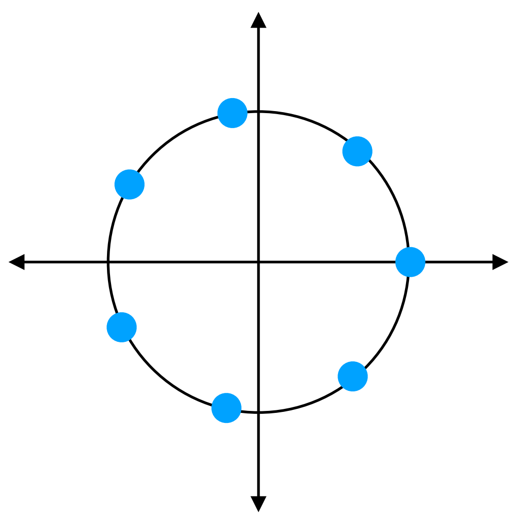
  Example:

$$\begin{array}{rr} 41 & \{2\,1\,6\} \\ +\ 26 & \{2\,1\,5\} \\ \hline 67 & \{1\,2\,4\} \end{array}$$

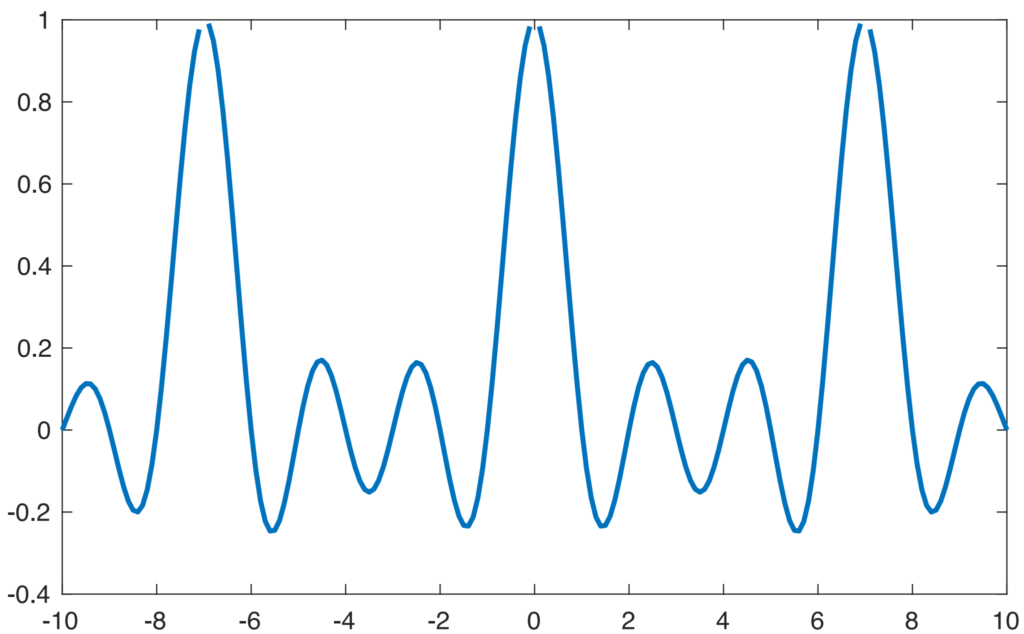
How to represent residue numbers with HD vectors?

(Kymn C, et al. 2023, arXiv:2311.04872)

When phasor distribution is discrete (sampled from n^{th} roots of unity) → Similarity kernel is modulo n



$$\mathbf{z}(x) \cdot \mathbf{z}(x + \Delta x)$$



Δx

$$\mathbf{z} = \begin{bmatrix} e^{j\theta_1} \\ e^{j\theta_2} \\ \vdots \\ e^{j\theta_N} \end{bmatrix} \quad \mathbf{z}(x) = \begin{bmatrix} e^{j\theta_1 x} \\ e^{j\theta_2 x} \\ \vdots \\ e^{j\theta_N x} \end{bmatrix}$$

How to represent residue numbers with HD vectors?

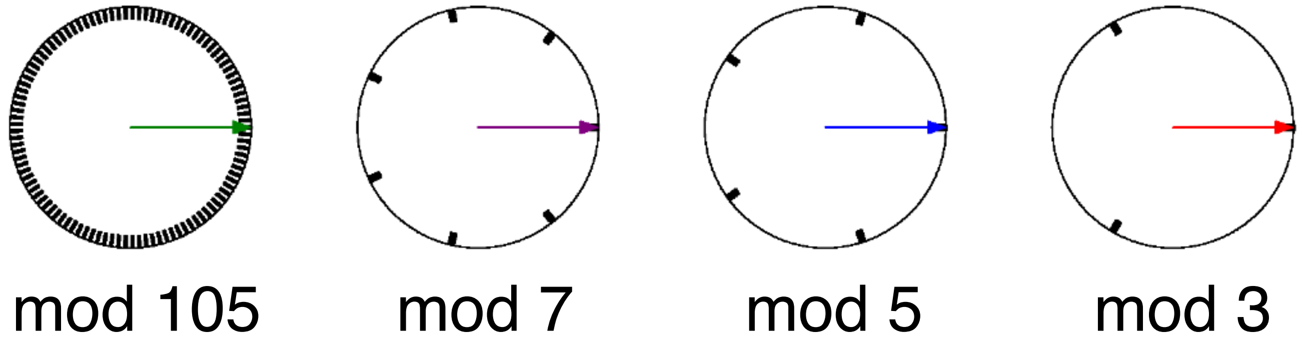
(Kymn C, et al. 2023, arXiv:2311.04872)

- Choose base vectors with phasors drawn from *m*-th roots of unity to represent numbers modulo *m.*

- A residue number representation of x can then be represented by *binding together* the vector representation of x for each of the moduli.

- For example, for the {3,5,7} RNS we have:

$$\mathbf{p}(x) = \mathbf{z}_1(x) \odot \mathbf{z}_2(x) \odot \mathbf{z}_3(x)$$

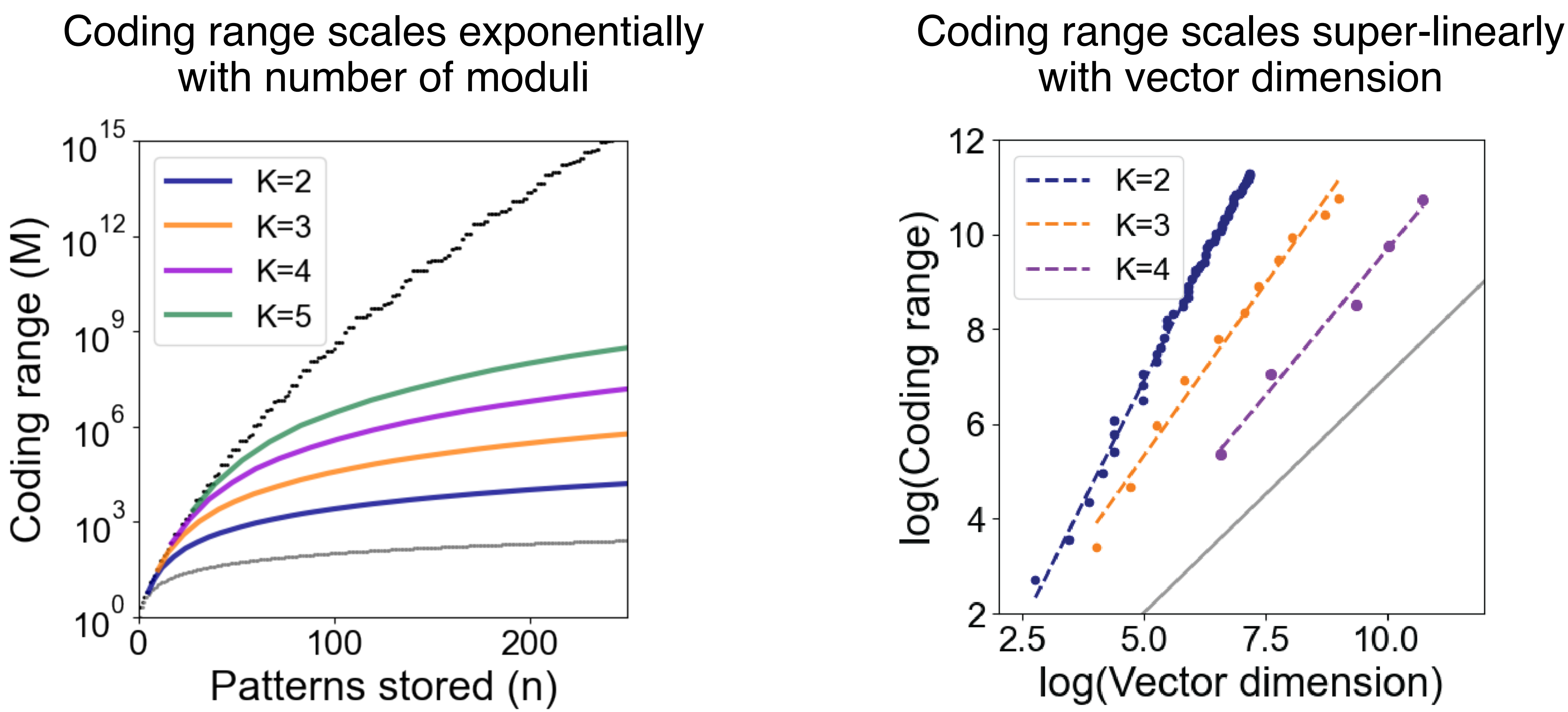


Given $\mathbf{p}$, how to compute its RNS components $\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{z}_3$?

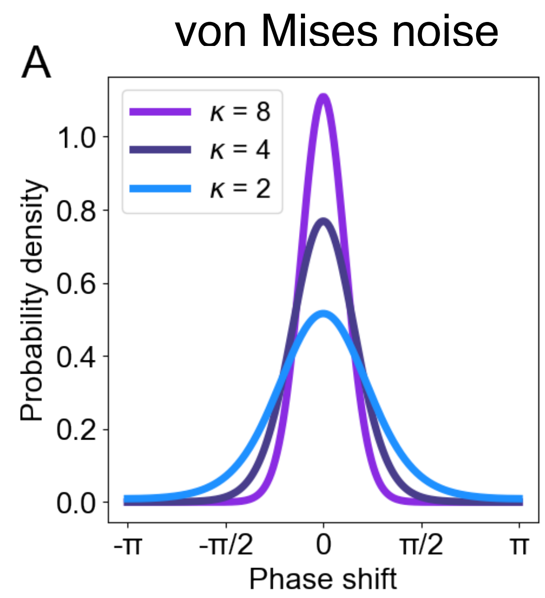
Factorize via

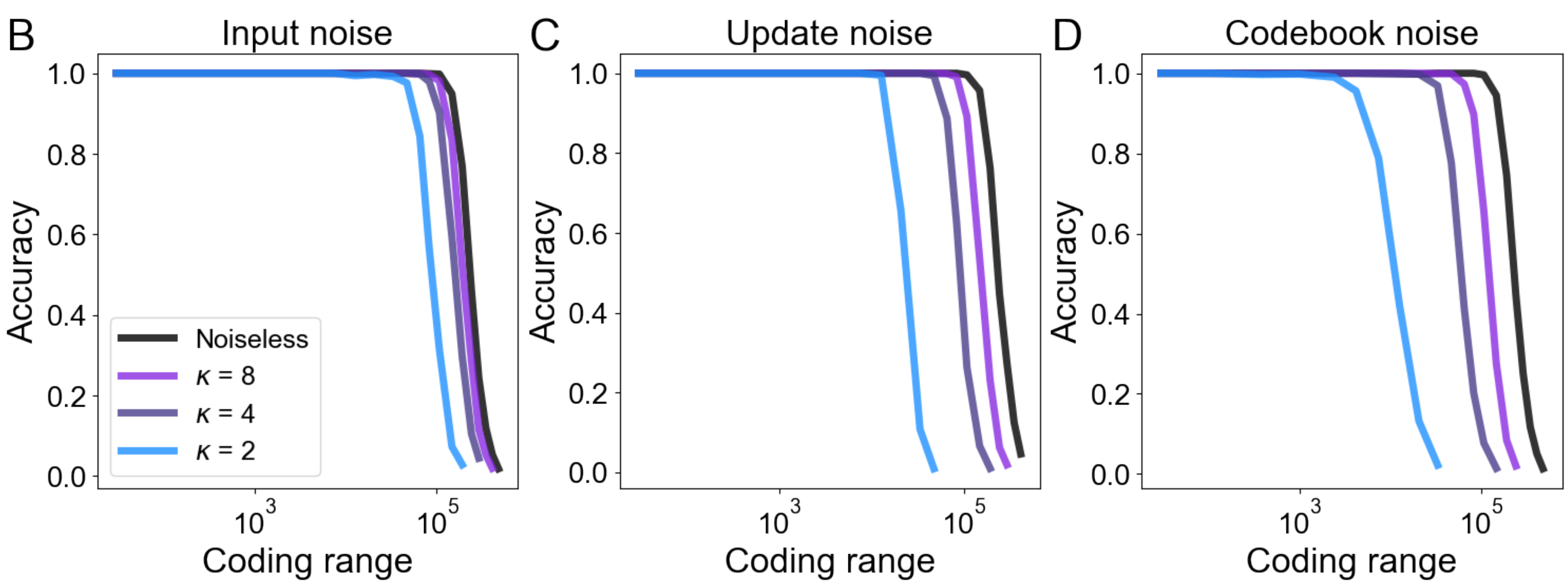
$$\hat{\mathbf{z}}_i(t+1) = \sigma\left(\mathbf{Z}_i \mathbf{Z}_i^\dagger (\mathbf{p} \bigodot_{j \neq i}^{K} \hat{\mathbf{z}}_j^\dagger(t))\right) \quad \forall\ i$$

$$\mathbf{Z}_i = \begin{bmatrix} | & | & & | \\ \mathbf{z}_i(0) & \mathbf{z}_i(1) & \dots & \mathbf{z}_i(m_i - 1) \\ | & | & & | \end{bmatrix}$$

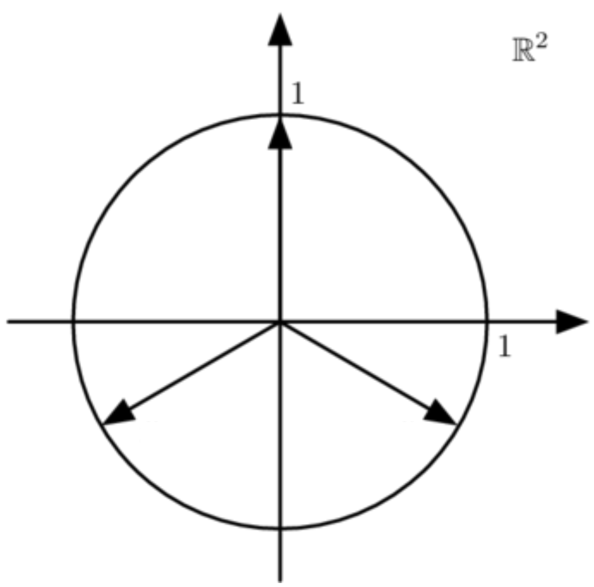

Coding range scales exponentially with number of moduli
Coding range (M)
Patterns stored (n)
K=2
K=3
K=4
K=5
10^{15}
10^{12}
10^{9}
10^{6}
10^{3}
10^{0}
0
100
200
Coding range scales super-linearly with vector dimension
log(Coding range)
log(Vector dimension)
K=2
K=3
K=4
12
10
8
6
4
2
2.5
5.0
7.5
10.0

Performance is robust to noise



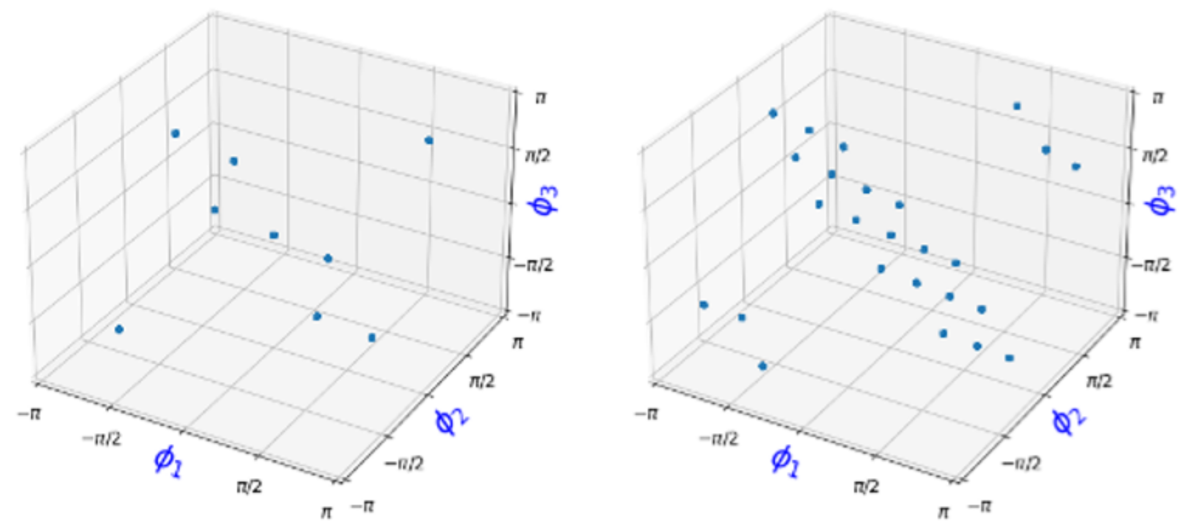


Representing 2D position in the ‘Mercedes Benz’ frame



‘Mercedes Benz’ frame

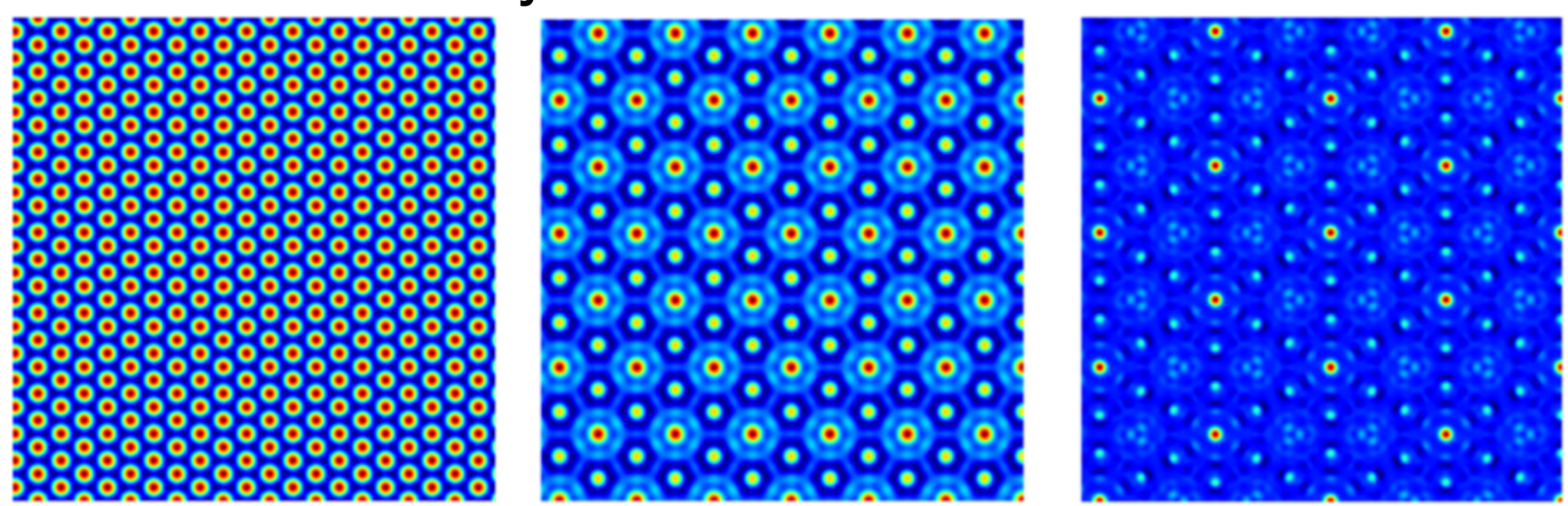
Phase distribution



Joint phase distribution constrained so that

$$\phi_1 + \phi_2 + \phi_3 = 0$$

Similarity kernel



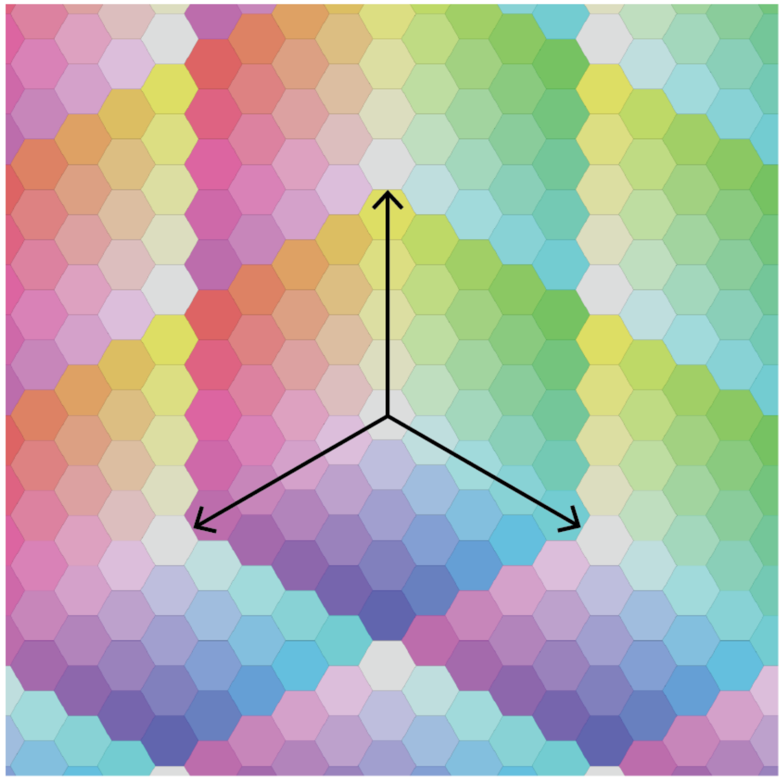
$\mathbf{g}_1(x)$ (mod 3)

$\mathbf{g}_2(x)$ (mod 5)

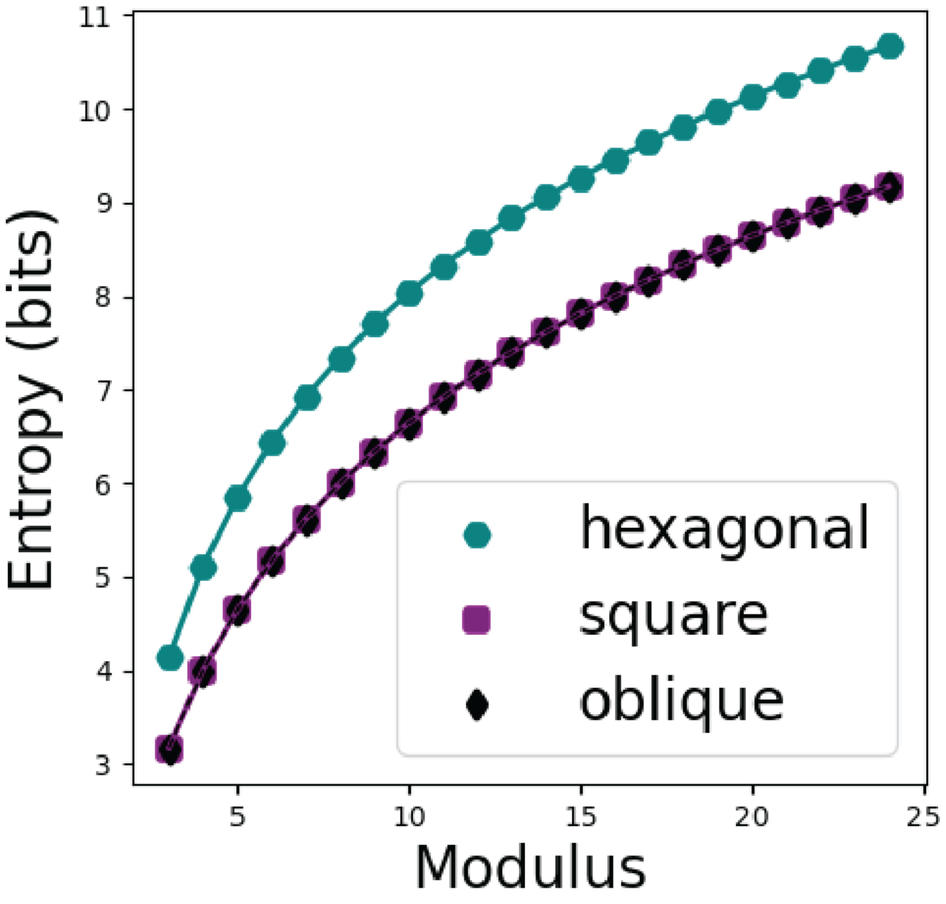
$\mathbf{g}_1(x) \odot \mathbf{g}_2(x)$ (mod 15)

Triangular coding improves spatial resolution

Voronoi tessellation for m=5



Triangular frame conveys more spatial information than rectangular frames



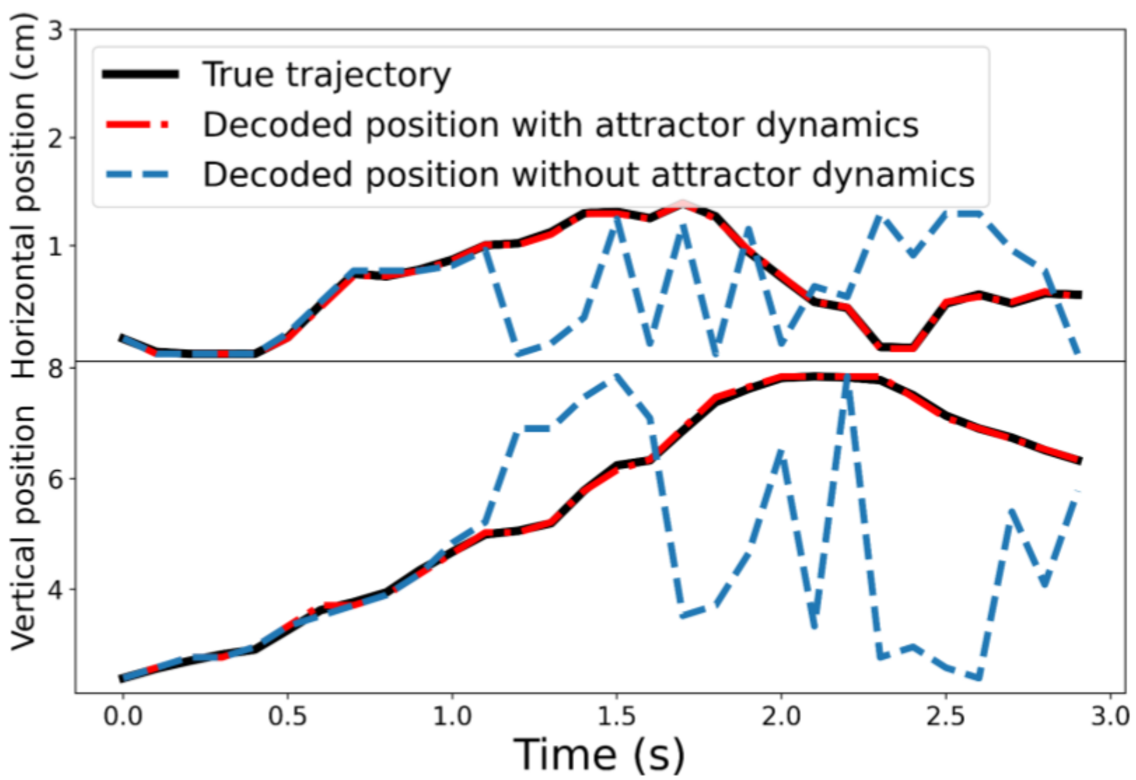
Path integration is accomplished by binding to instantaneous velocity

$$\hat{\mathbf{z}}_i(t+1) = \mathbf{q}_i(v_t) \odot \sigma\left(\mathbf{Z}_i \mathbf{Z}_i^\dagger (\tilde{\mathbf{p}}(x_t) \bigodot_{j \neq i}^{K} \hat{\mathbf{z}}_j^\dagger(t))\right) \quad \forall\ i$$

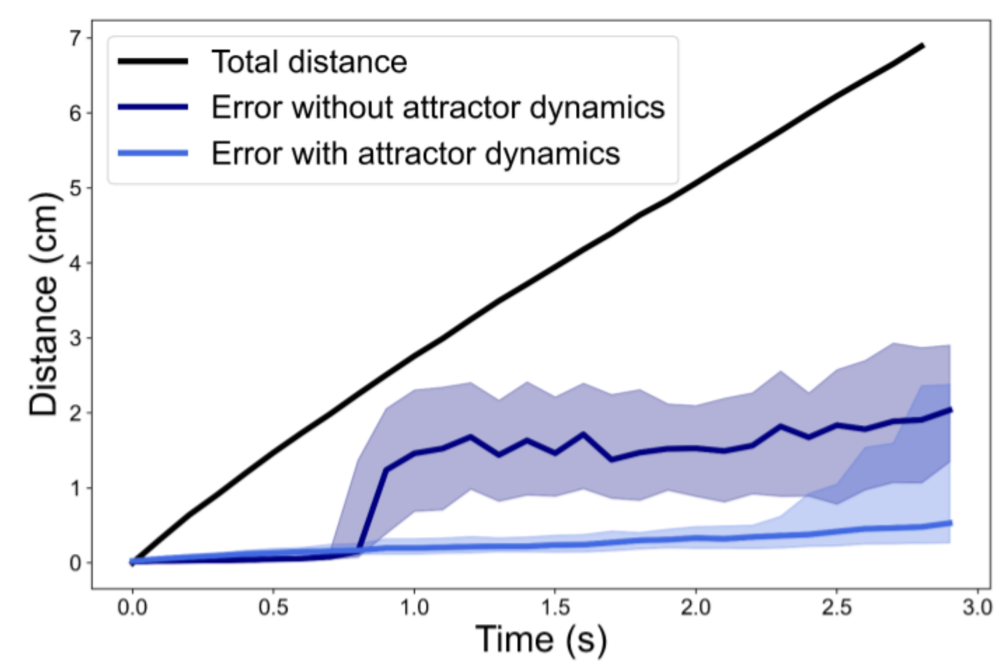
$$\tilde{\mathbf{p}}(x_{t+1}) = \bigodot_{i=1}^{K} \hat{\mathbf{z}}_i(t+1)$$

Robustness to noise

A



B



Main points

- Animal intelligence
- Physics of computation
- Perception as factorization
- Equivariant representation