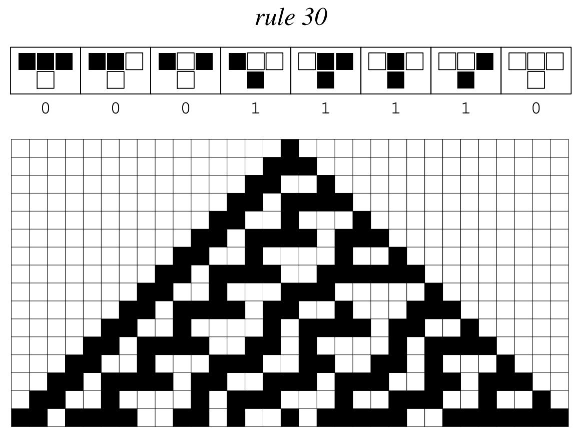
Introduction to Neurobiological Computation

Down the rabbit hole

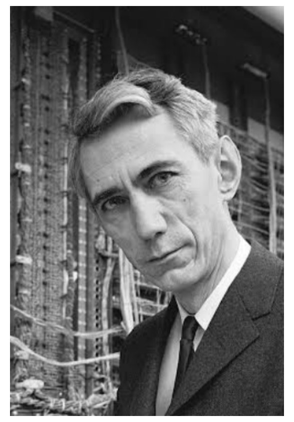
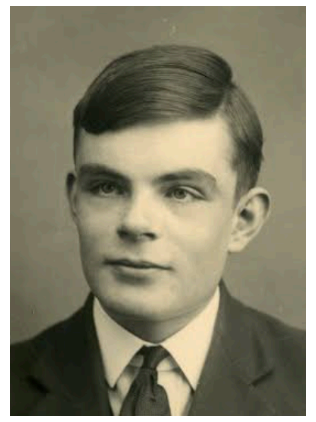
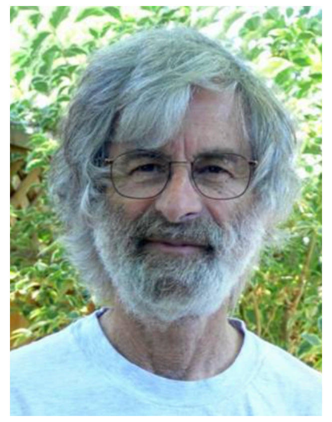
Aman, October 2024

My Heroes

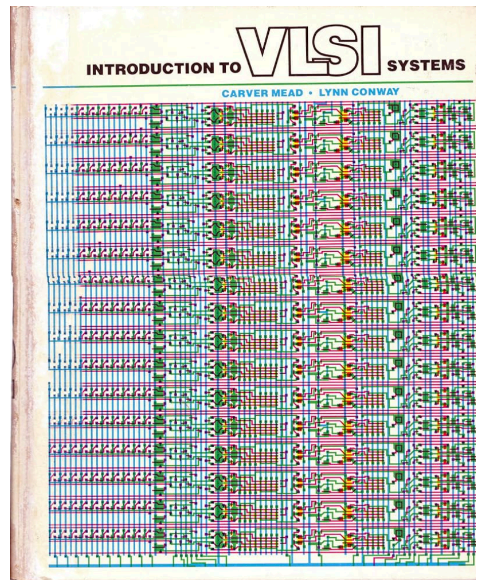
Toward Cellular Computing



Thinking Machines Corporation



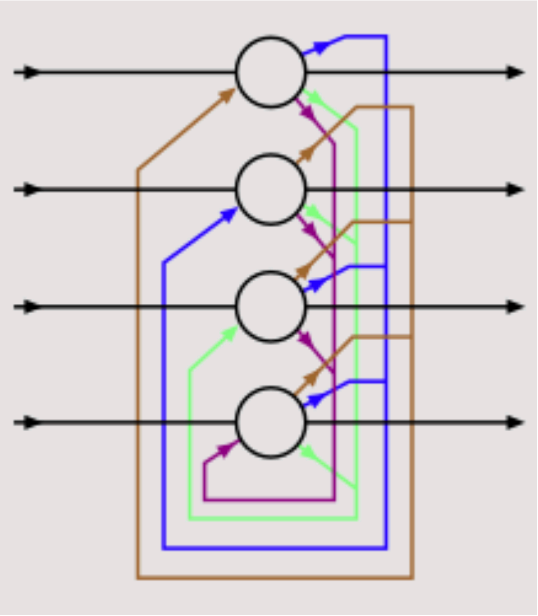



Mead/Conway Neuromorphics

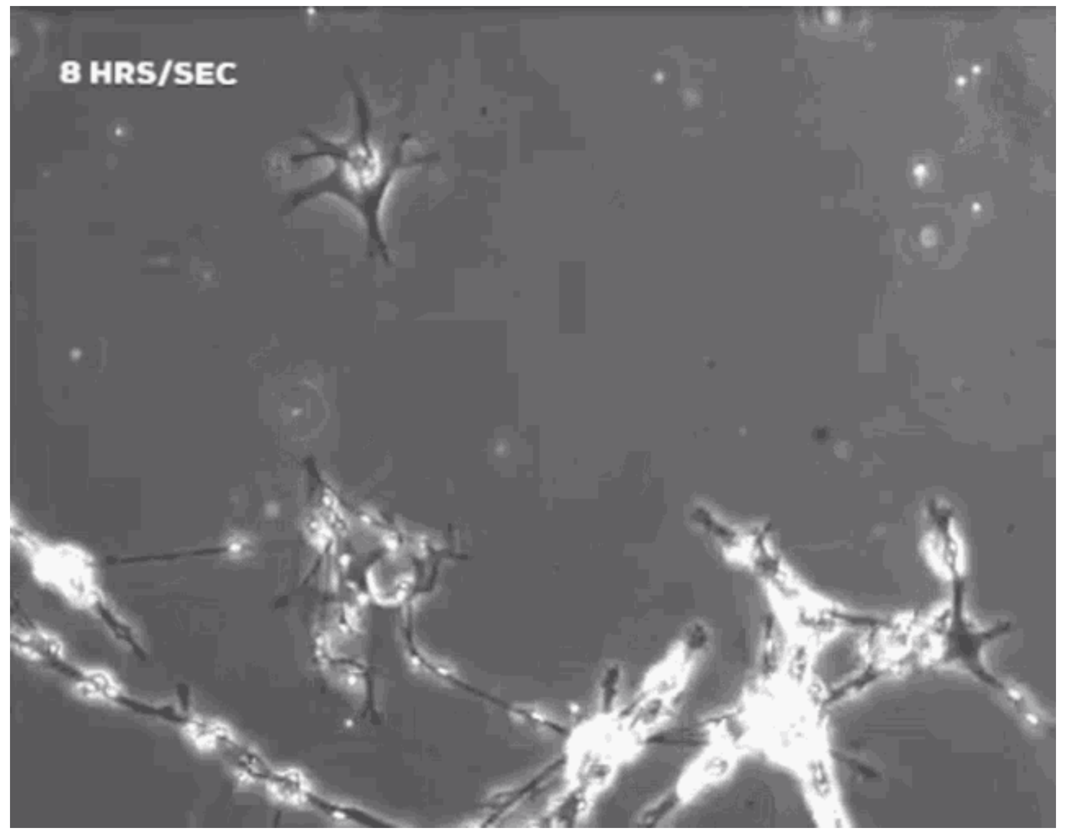


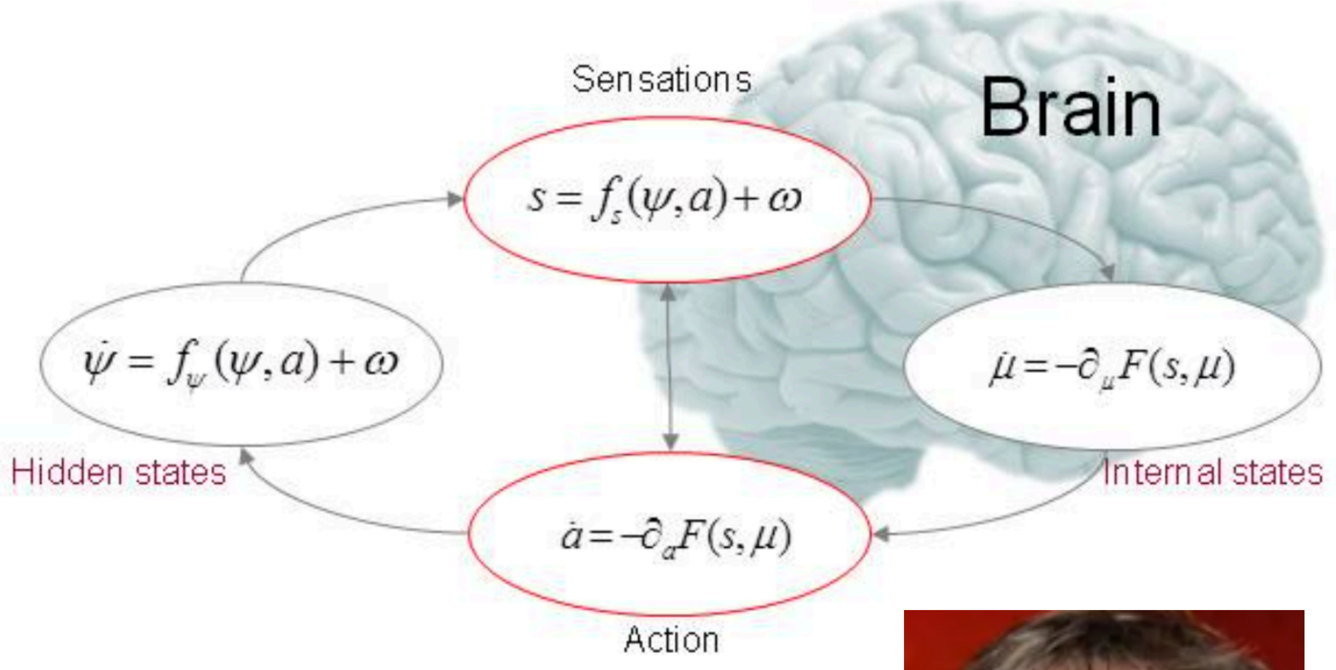


Feynman/Wolfram/Winfree/Levin/Thomson
Cellular/bio/chemical computation

Hopfield/Comp Neuro

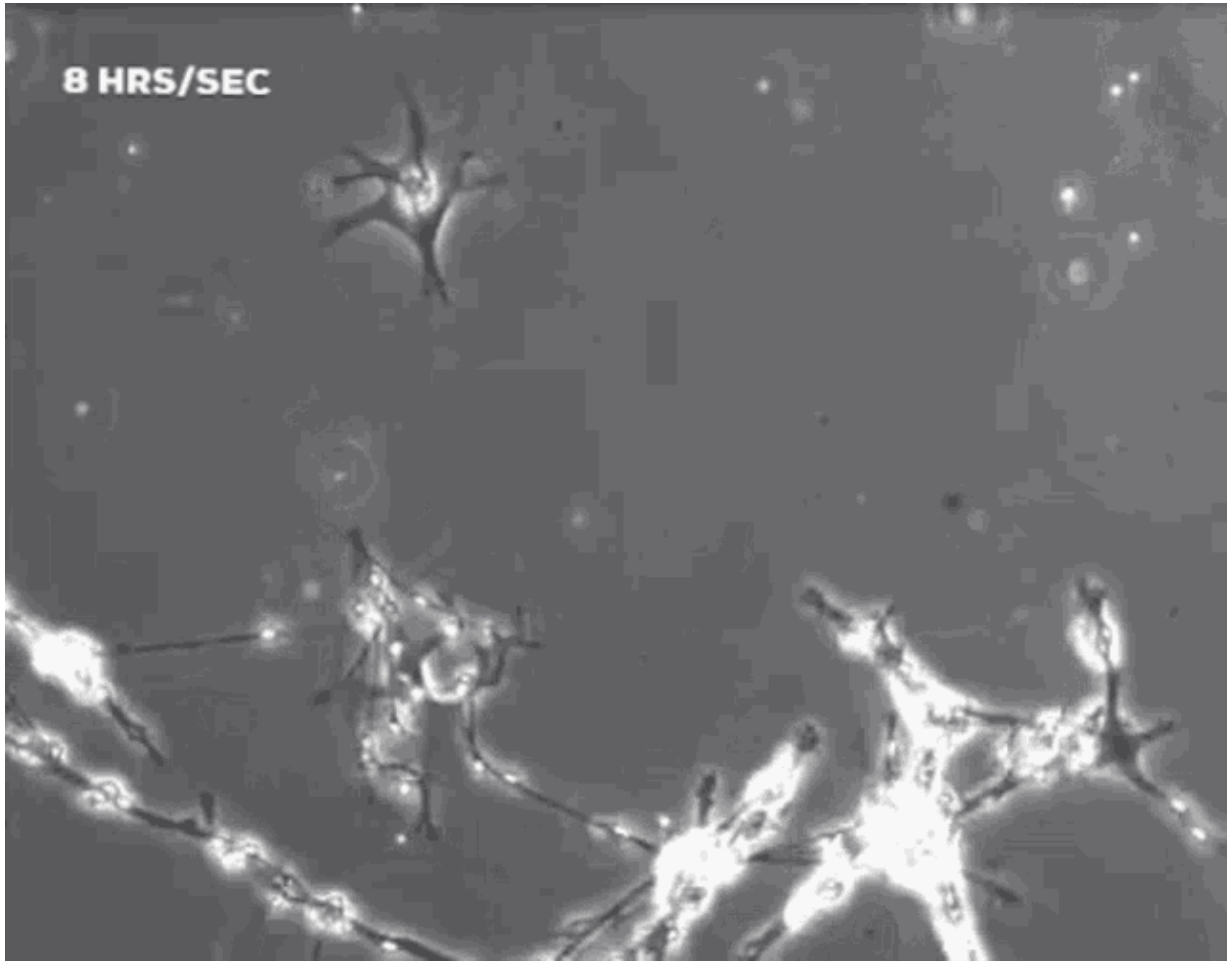


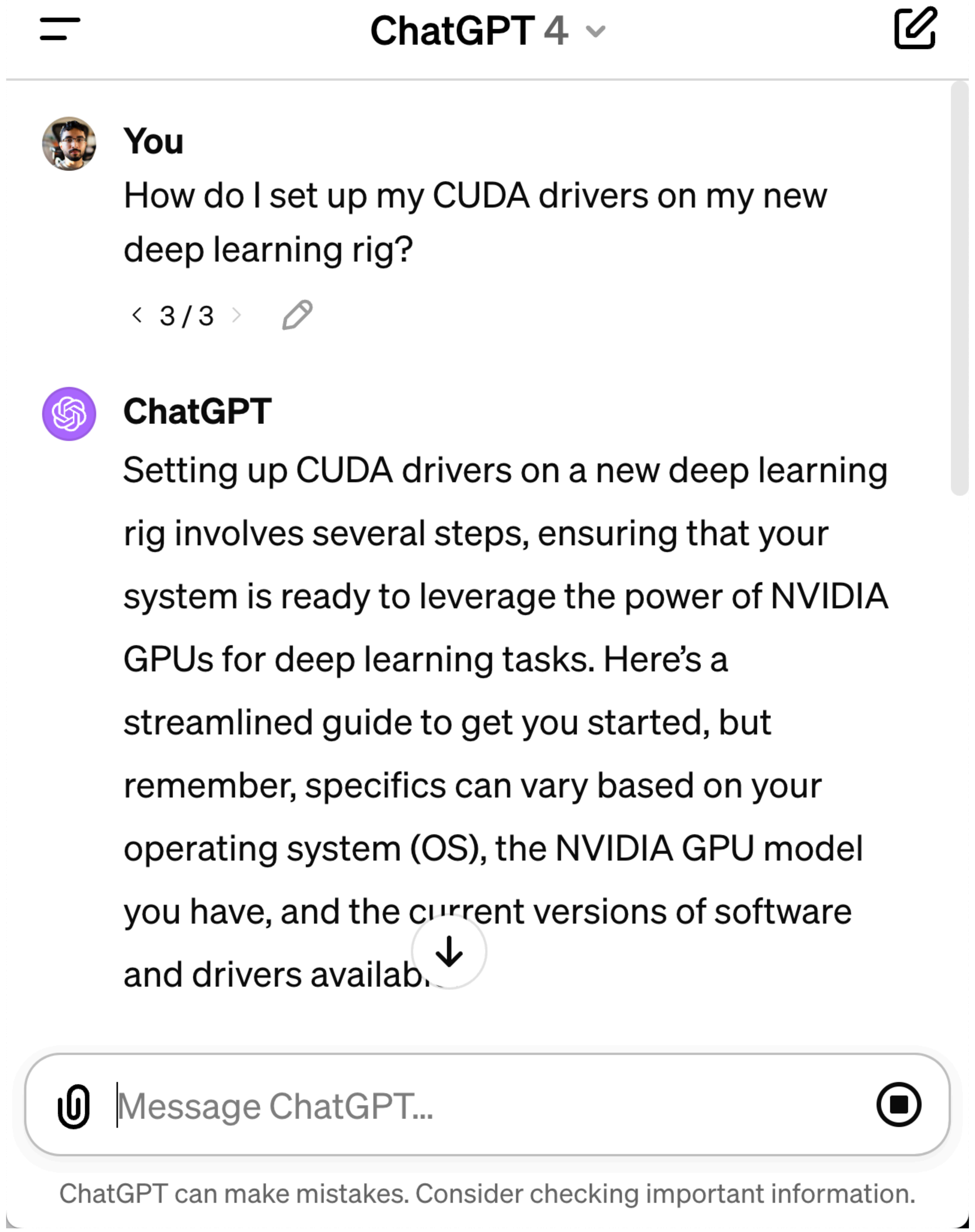



Friston/Free Energy Principle

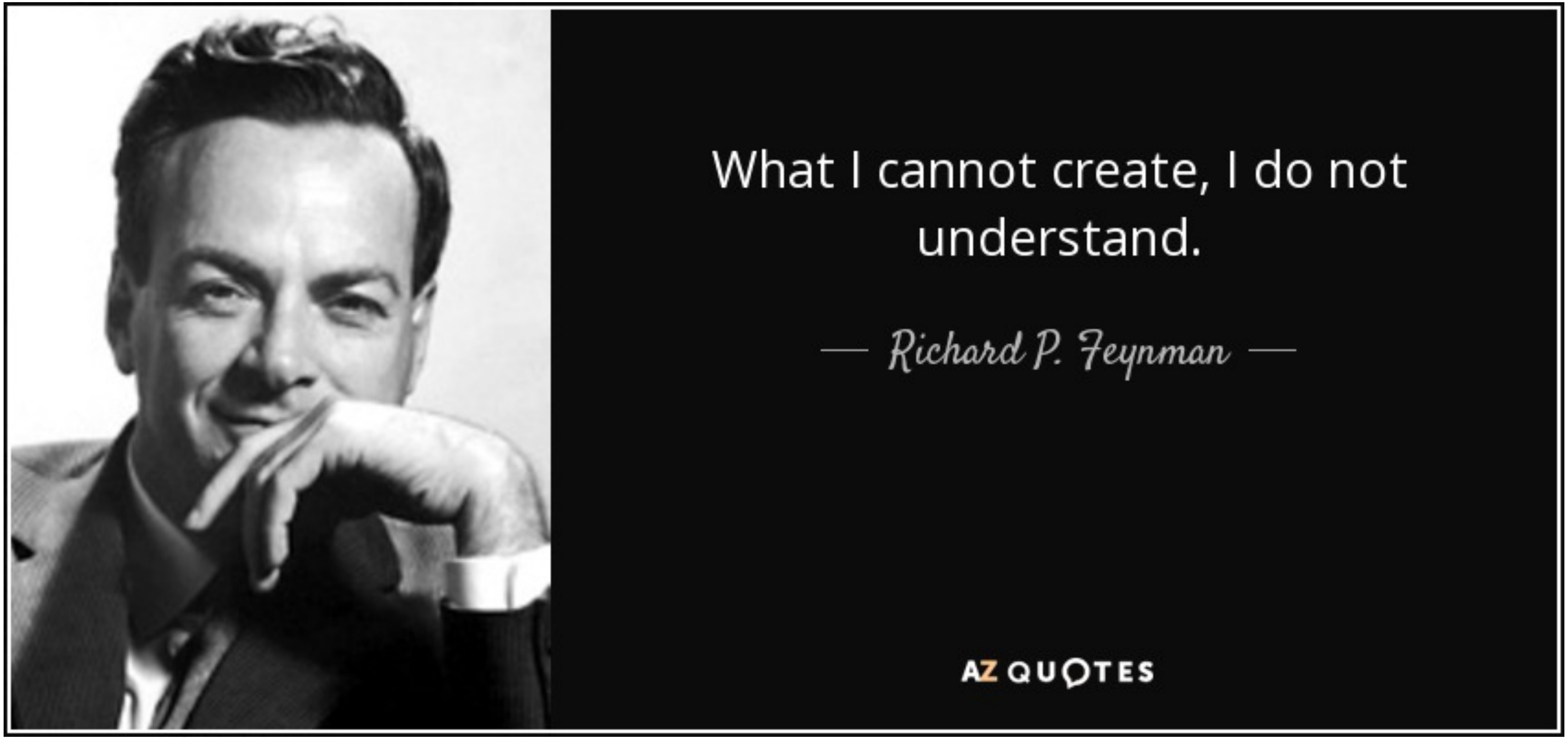
Big Jeff →

Intelligence is a fascinating phenomenon.





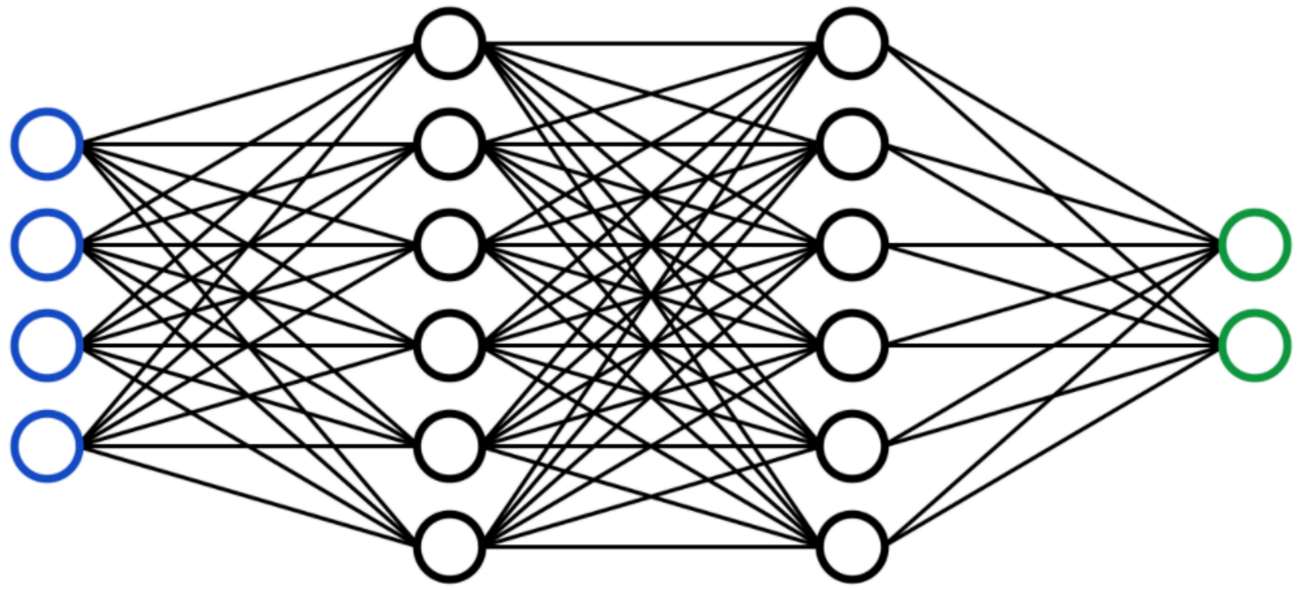
Building Systems can help us understand intelligence.

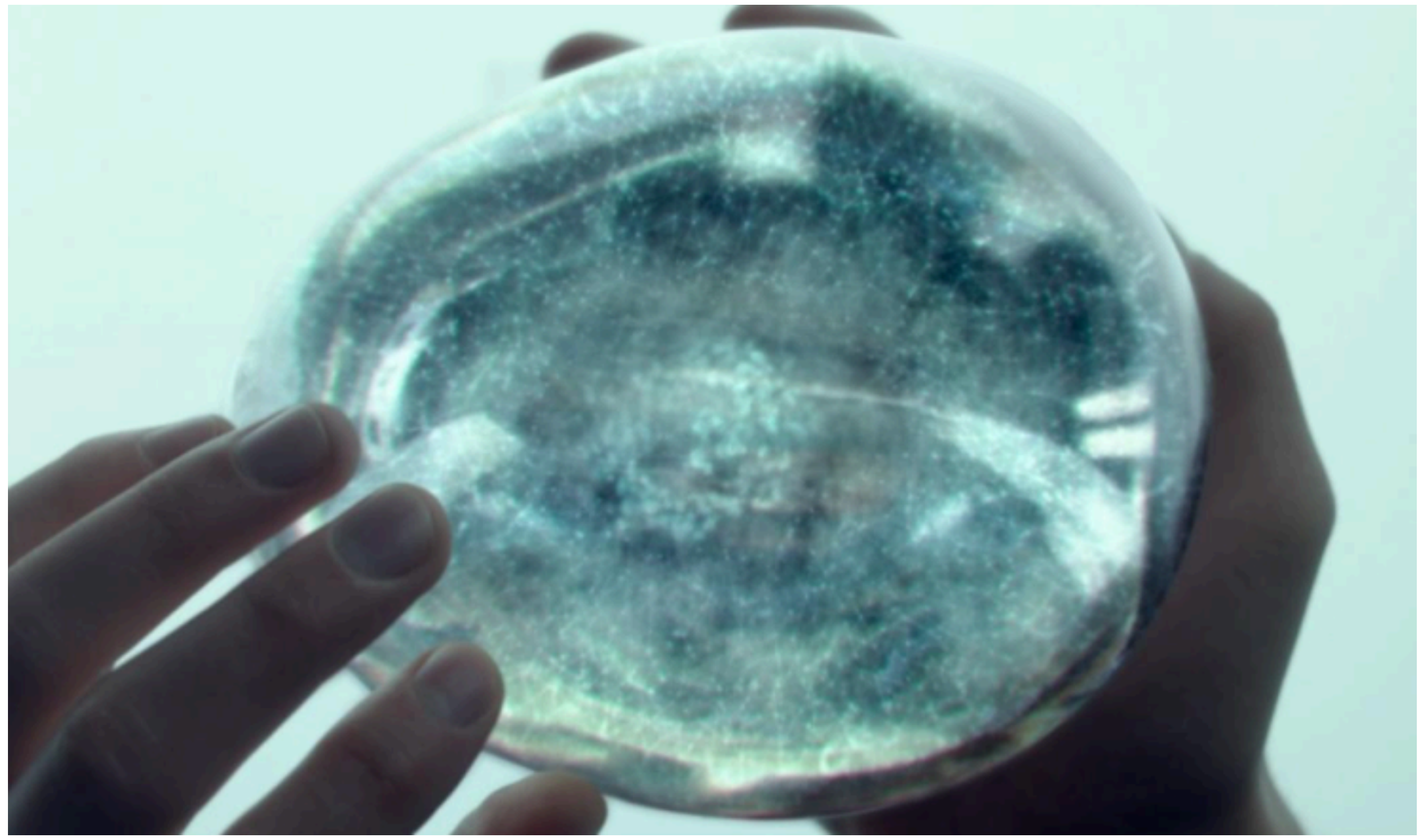


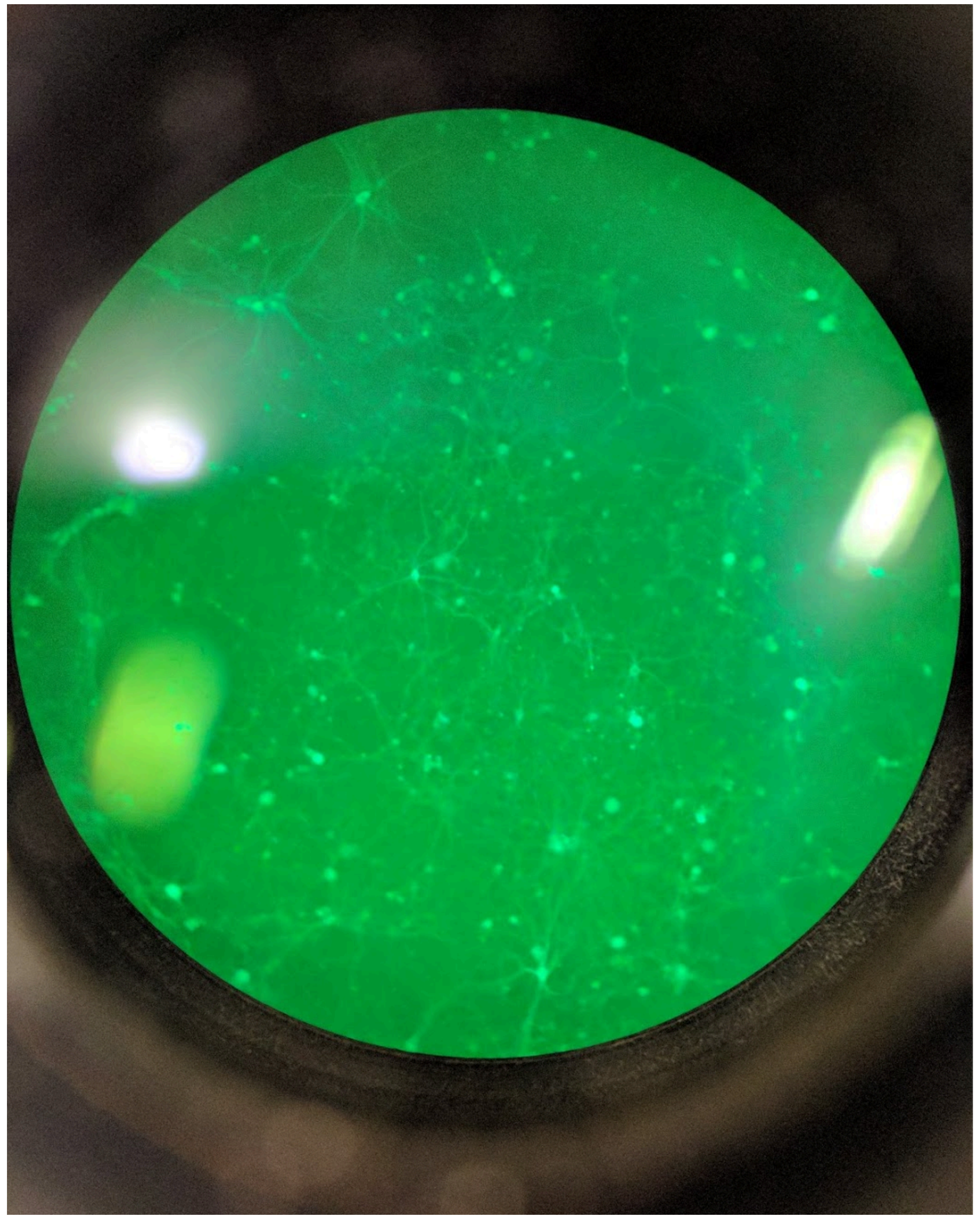
Building Systems can help us understand intelligence.

The Miracle of Biological Self-Organization
Motivation

- **Human brain:** <u>Billions</u> of neurons, <u>quadrillions</u> of synapses cooperate to create **intelligence.**
- **Neuroscience**: How does it work?
- **Artificial Intelligence**: How can we build one?





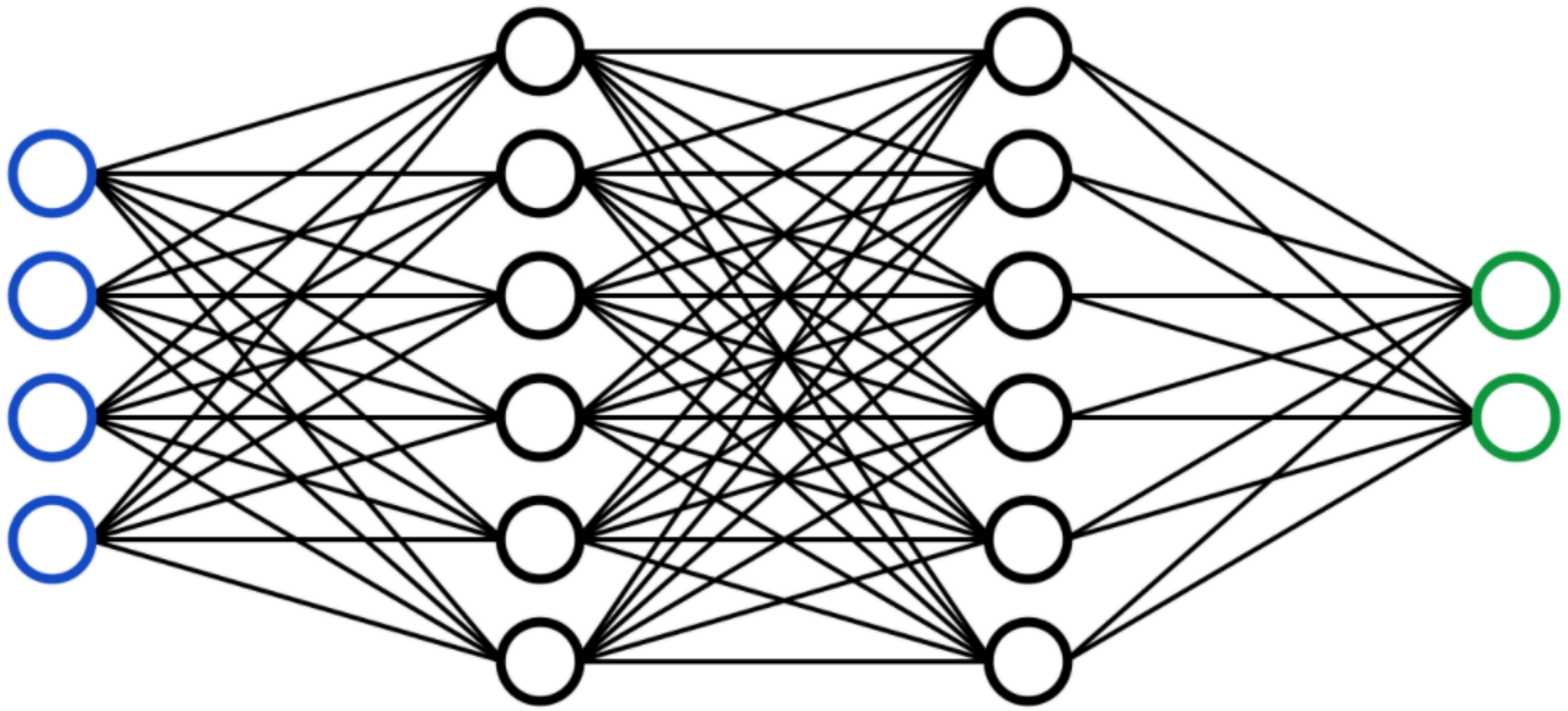


Learning = Changes in Neuron Connectivity

The centrality of synapse "weights" in learning & memory

- Eric Kandel — 2000 Nobel Prize
- Experimentally derived from **mouse & slug** models.
- Logical next question: **how are weights changed?**

How do Artificial Neural Networks Work?

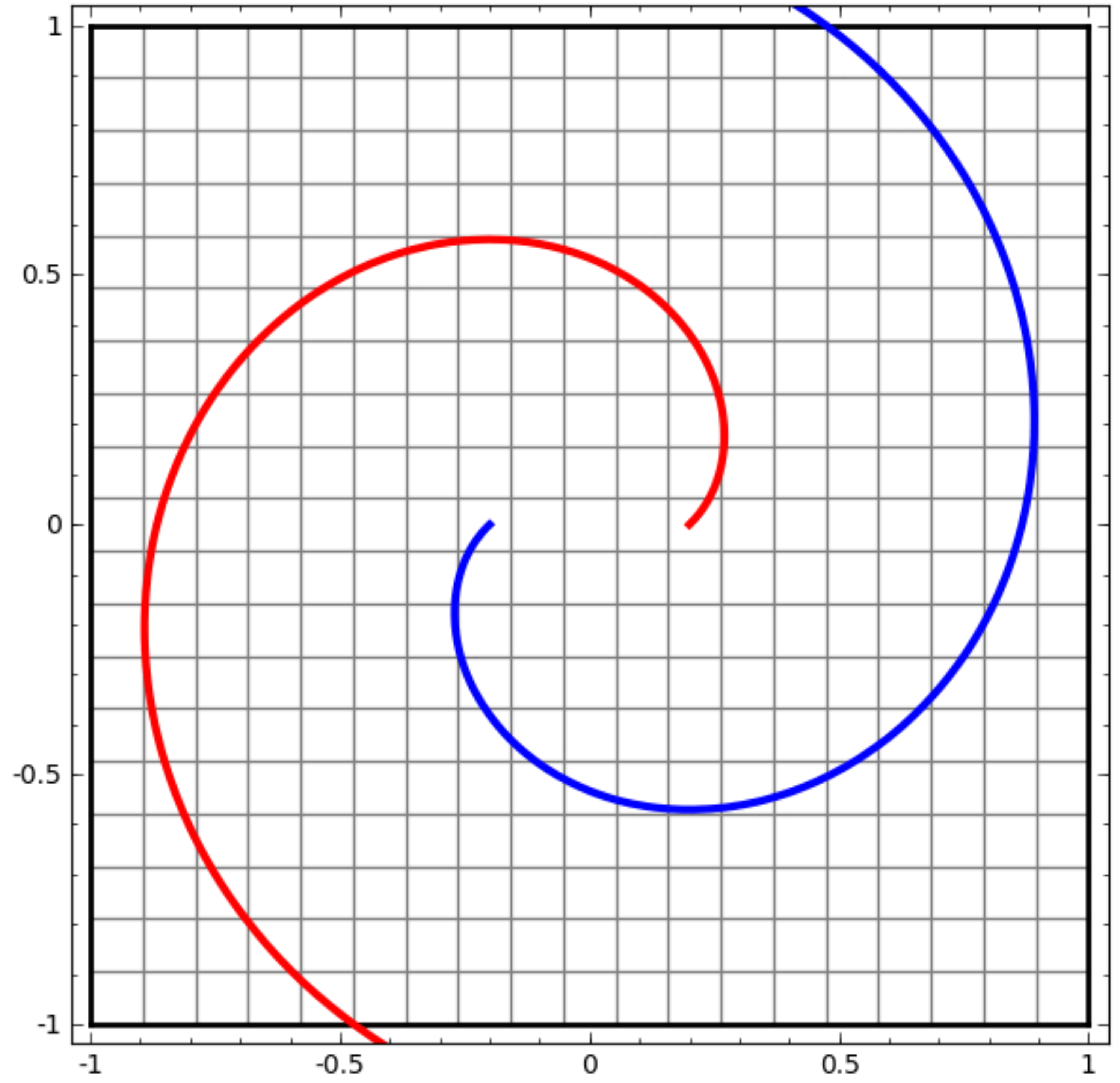


Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

$$\vec{x}^{(\ell)} = \begin{bmatrix} 1 \\ \theta(\vec{s}^{(\ell)}) \end{bmatrix}$$



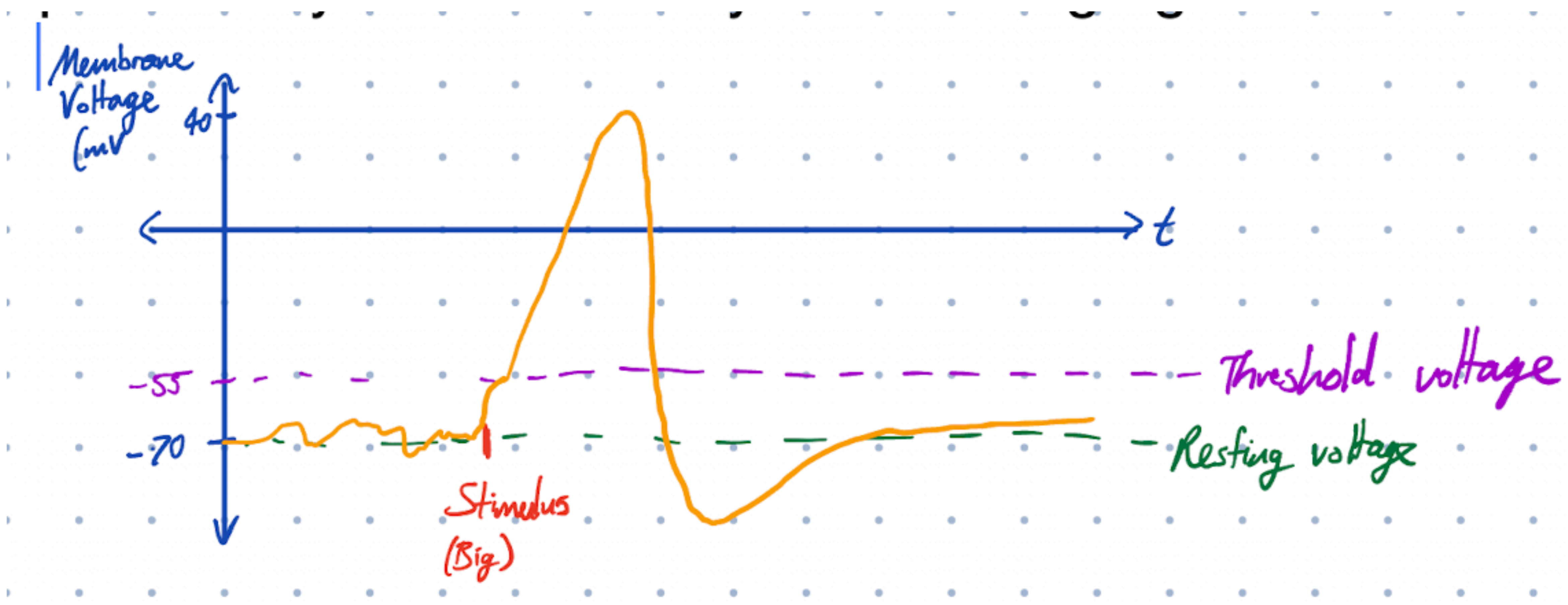
How do Artificial Neural Networks Work?

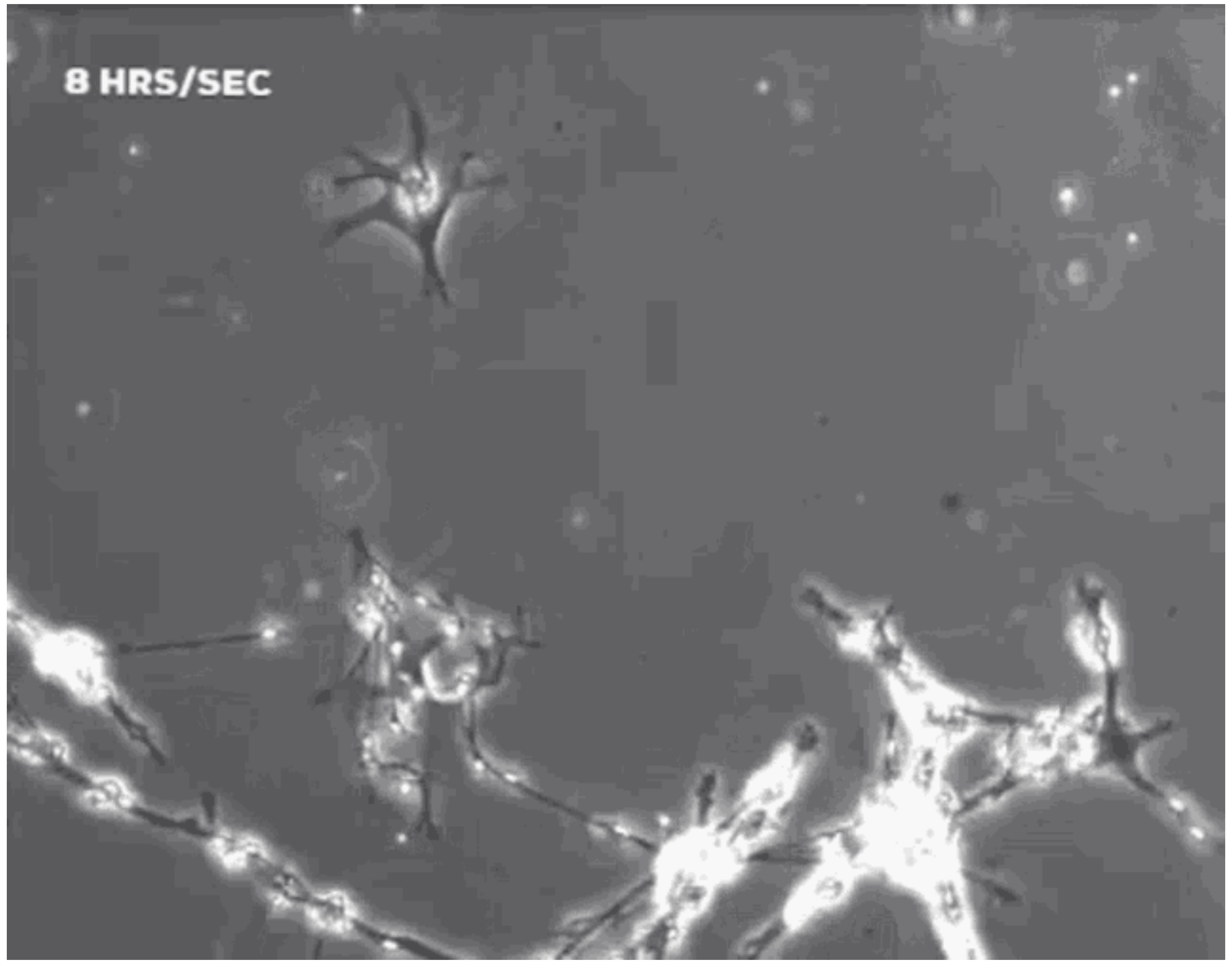
- **Feed-forward architecture**: Information flows only forward through layers.
- **Scalar activation values**: No internal state other than a scalar activation value (and backprop values).
- **Thresholded Linear Transformations**: $\mathbf{x}^{\ell+1} = \sigma(\mathbf{W}\mathbf{x}^{\ell})$
  - *Universal function approximation!*
- **Learning**: Gradient descent, chain rule, back propagation.

Miracle: ANN's with backprop <u>actually work</u>!

How do Biological Neural Networks Work?

- **Recurrent architecture**: Information flows all over the place.
- **High-dimensional cell states**: Neurons = cells = little bags of electrochemically active biomolecules (nanomachines).
- **Thresholded activations**:
- **Learning**: Can't do back propagation!

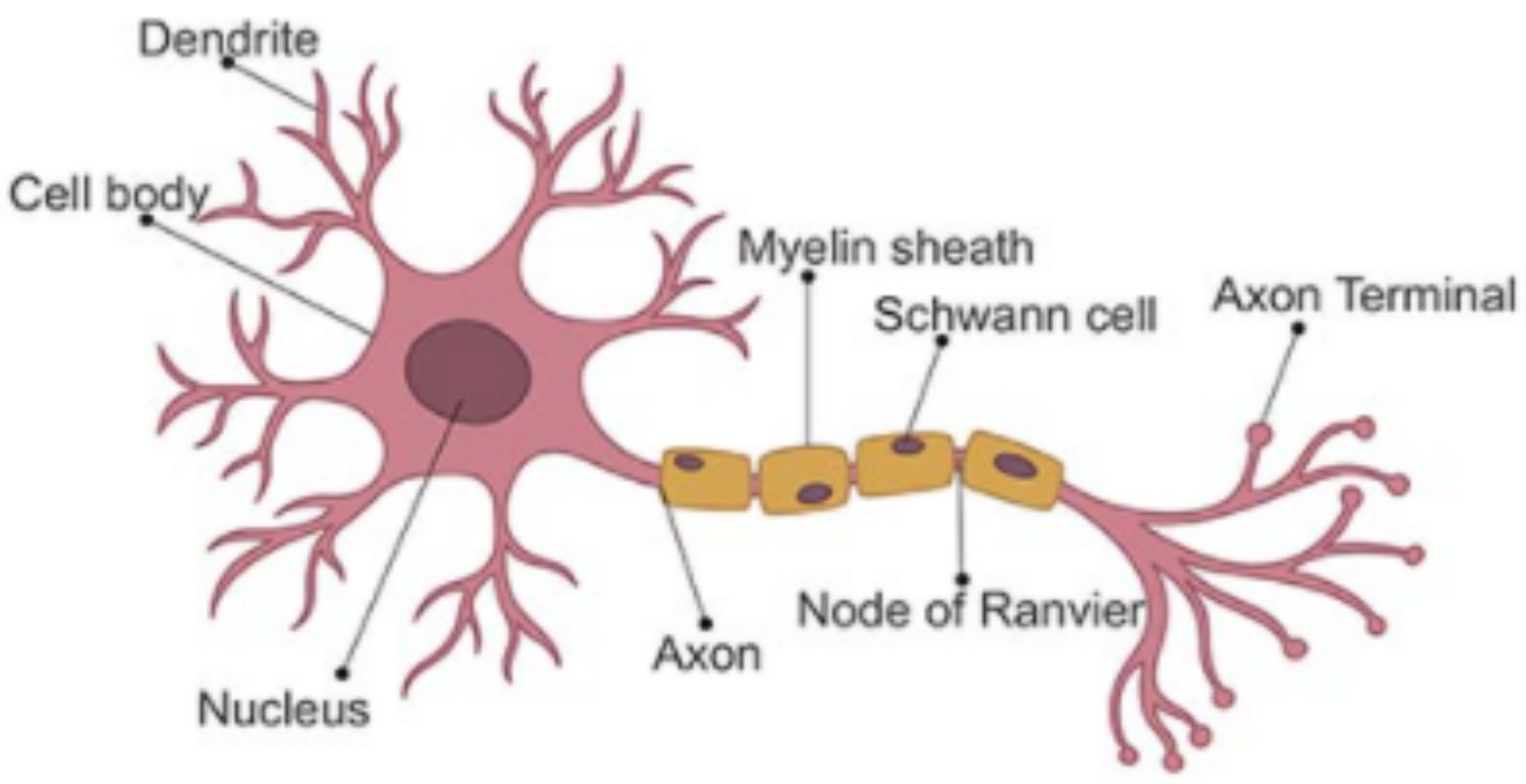


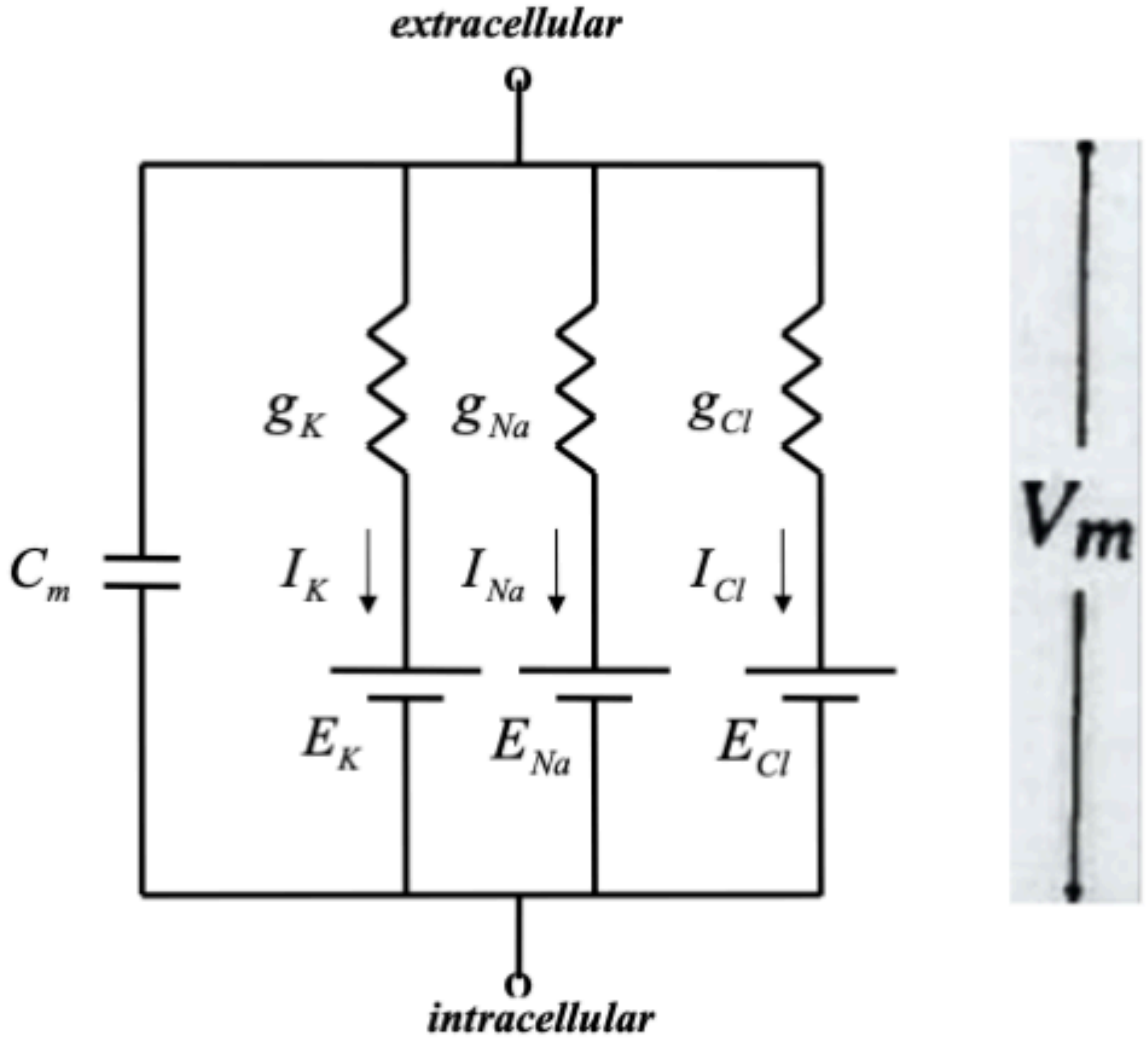


The Hassle of Biological Neurons

Hodgkin-Huxley Models

- Extremely realistic for action potentials, but **painfully slow** to simulate.
- Does not explain **learning**.





$$g_K(t; v_m) = \bar{g}_K \cdot n(t; v_m)^4$$

$$\frac{d}{dt} n(t, v_m) = (1 - n) \cdot \alpha_n(v_m) - n \cdot \beta_n(v_m)$$

$$n(t) = n_\infty - (n_\infty - n_0)e^{-t/\tau_n}$$
$$\tau_n = (\alpha_n + \beta_n)^{-1}$$
$$n_\infty = \alpha_n(\alpha_n + \beta_n)^{-1}$$
$$\therefore n'(t) = (n_\infty - n)/\tau_n$$

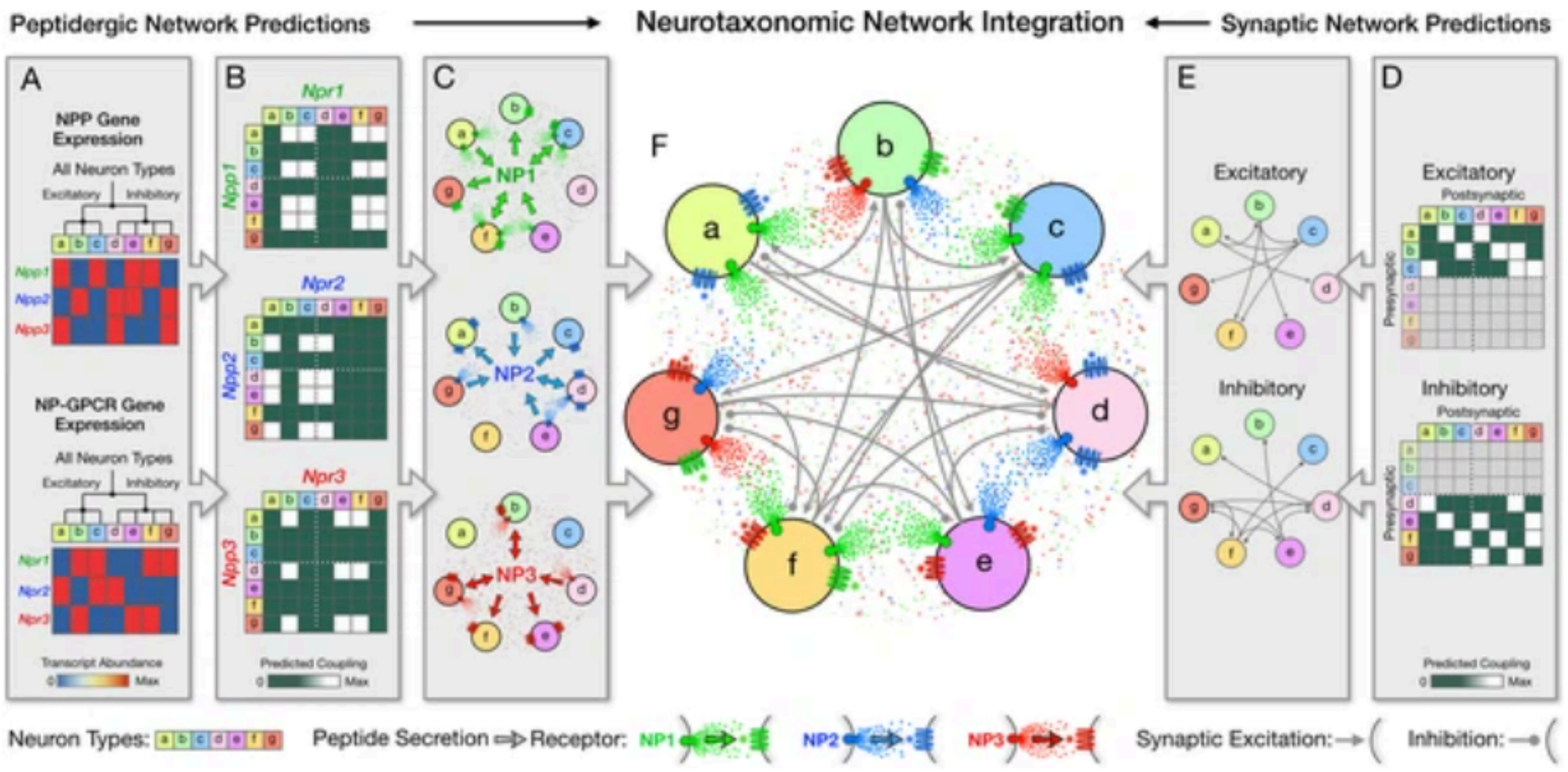
$$\alpha_n(v_{mi}) = \frac{n_\infty(v_{mi})}{\tau_n(v_{mi})} \qquad \alpha_n = \frac{0.01(10 - v_m)}{[\exp(\frac{10 - v_m}{10}) - 1]}$$

$$\beta_n(v_{mi}) = \frac{[1 - n_\infty(v_{mi})]}{\tau_n(v_{mi})} \qquad \beta_n = 0.125 \exp(\frac{-v_{mi}}{80})$$

The Hassle of Biological Neurons

Cells are Biological Microcomputers

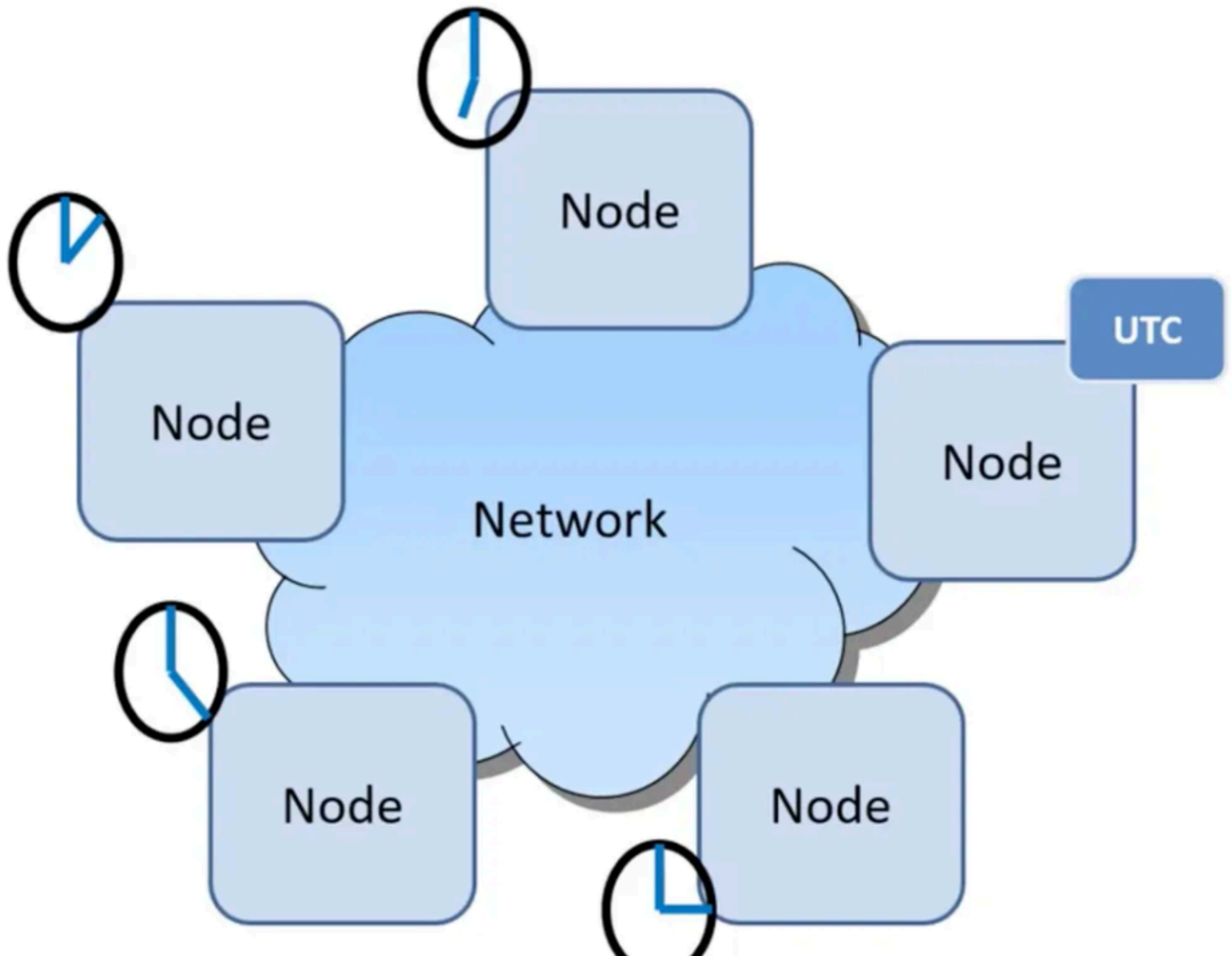
- Every cell is capable of **non-trivial computation** via biochemical reactions, gene regulation, neurotransmitter diffusion, etc.



The Hassle of Biological Neurons

Cells are Biological Microcomputers

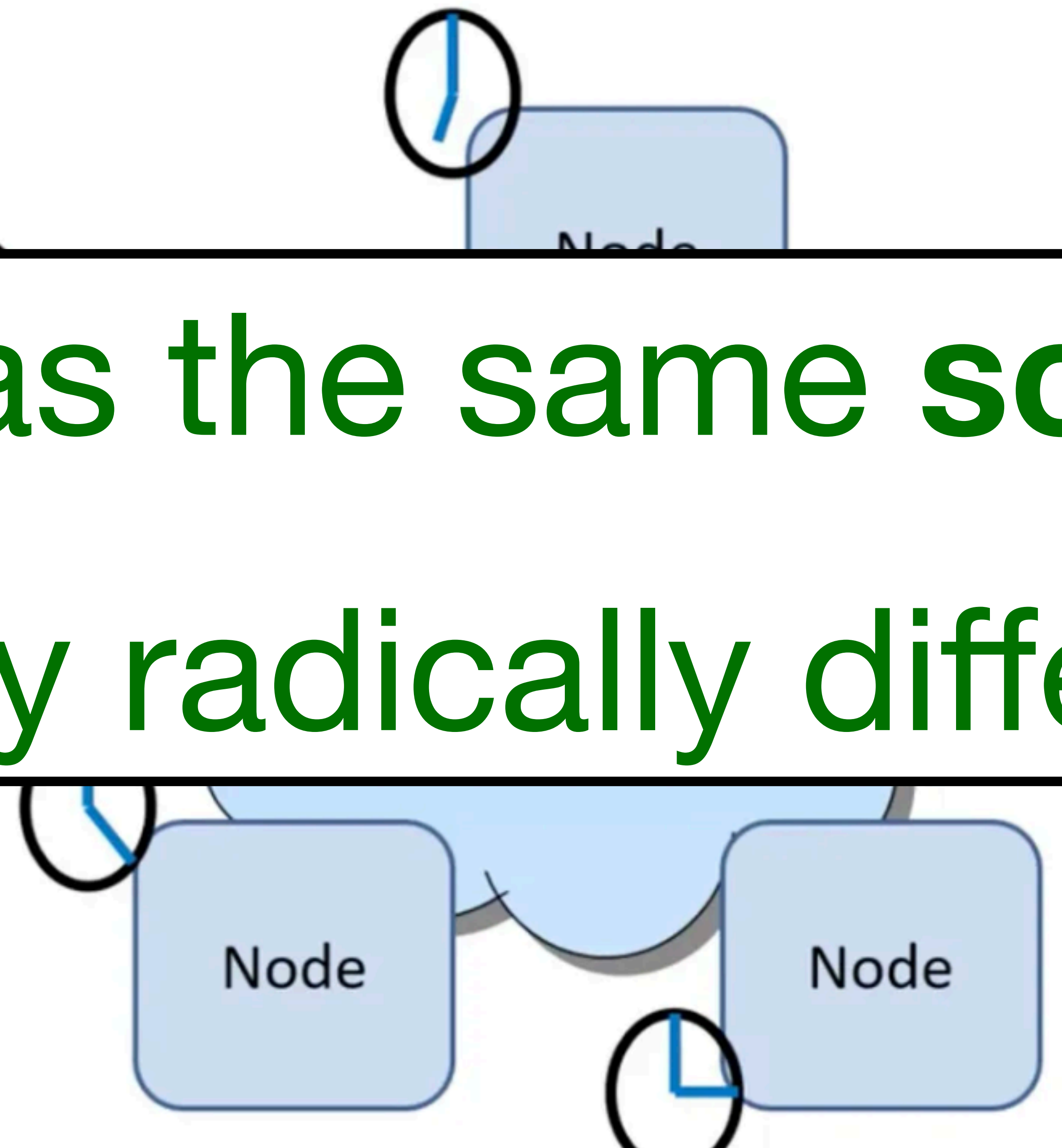
- Brain = **massively distributed, ASYNCHRONOUS system** of computers (neurons).
- NOT FEED FORWARD — the brain is highly **recurrent**.
- Each node runs **evolution-derived spaghetti code.**



The Hassle of Biological Neurons

Cells are Biological Microcomputers

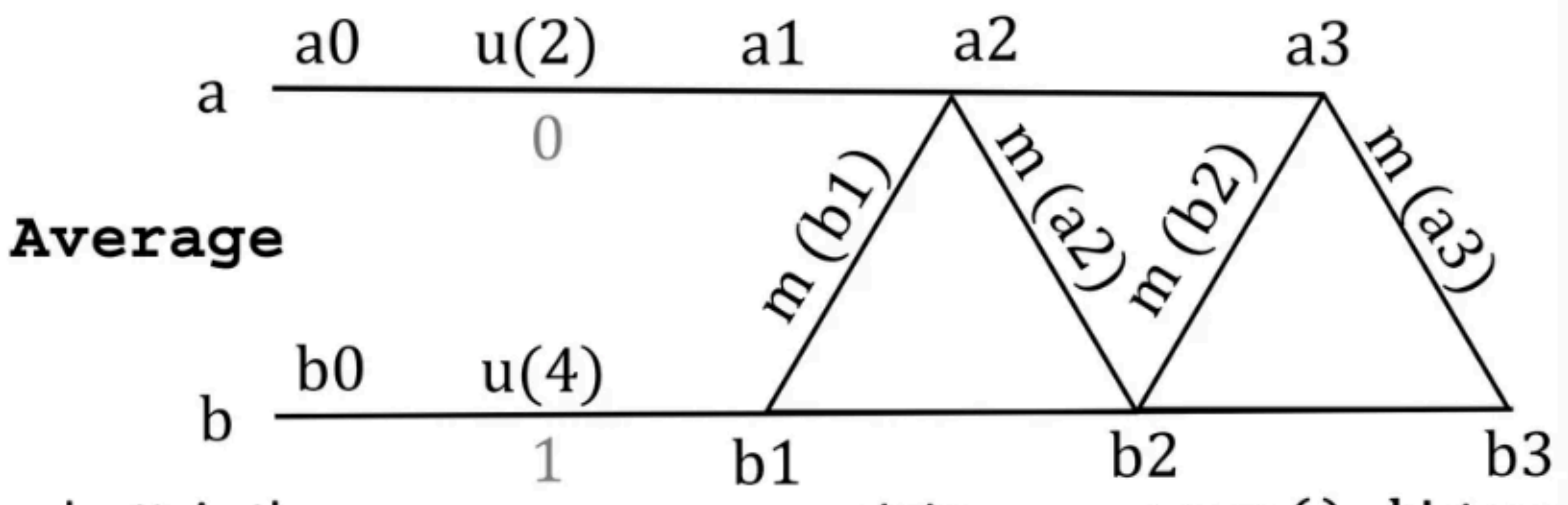
- Brain = **massively distributed, ASYNCHRONOUS system** of computers (neurons).
- NOT FEED FORWARD — the brain is highly **recurrent**.
- Each node runs **evolution-derived spaghetti code.**



Every node has the same **source code**, but potentially radically different **state**.

The Hassle of Biological Neurons

Engineering Distributed Systems is Hard

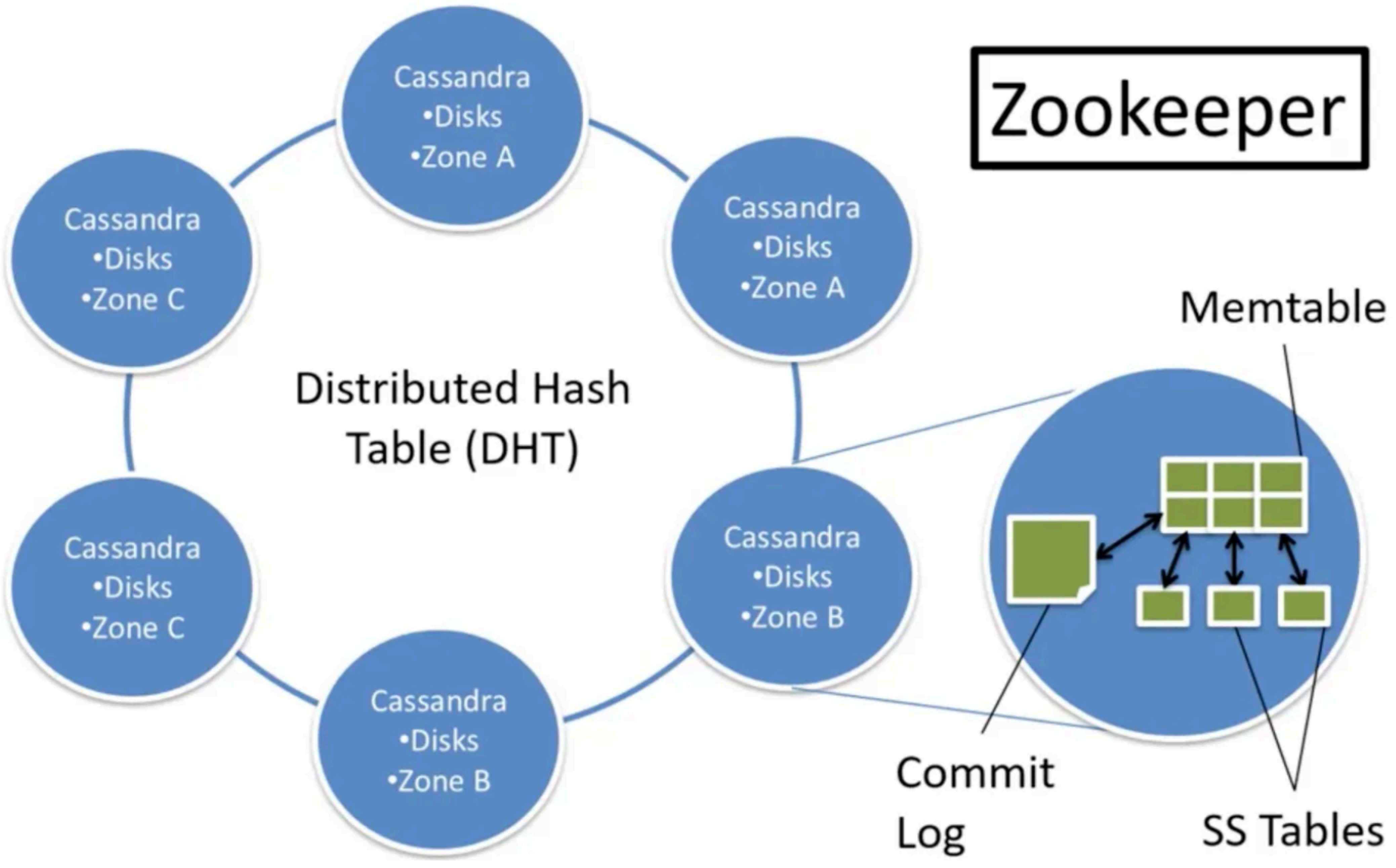


a, b attain the **same causal history** but **do not converge** to the **same internal state** – they **do not converge** at all**!**

Neither eventually consistent, nor strongly eventually consistent

	state		query ()	history
a0	sum:0,	cnt:0	0	{ }
a1	sum:2,	cnt:1	2	{0}
a2	sum:6,	cnt:2	3	{0,1}
a3	sum:16,	cnt:5	3.2	{0,1}
b0	sum:0,	cnt:0	0	{}
b1	sum:4,	cnt:1	4	{1}
b2	sum:10,	cnt:3	3.3	{0,1}
b3	sum:26,	cnt:8	3.25	{0,1}

Distributed Systems (Hans-Arno Jacobsen)



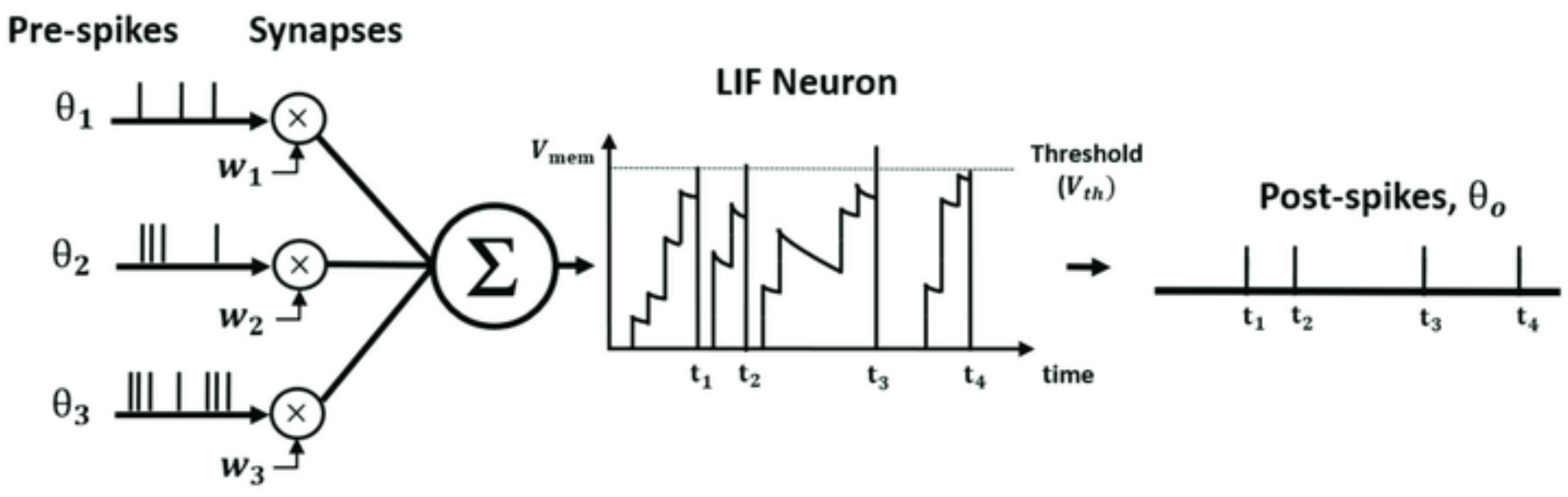
All this just to **agree** +

maintain the same info!

Spiking Neural Networks

A nice middle ground?

- Preserves behaviour of **Hodgkin-Huxley**: *Spike-based communication, spatiotemporal summation.*
- Applies simplification of **ANN's**: *Scalar state variables, scalar synapses.*
- **"Leaky integrate-and-fire"**



Spiking Neural Networks

A nice middle ground?

Efficient hardware exists, but **strong learning rules** do not.

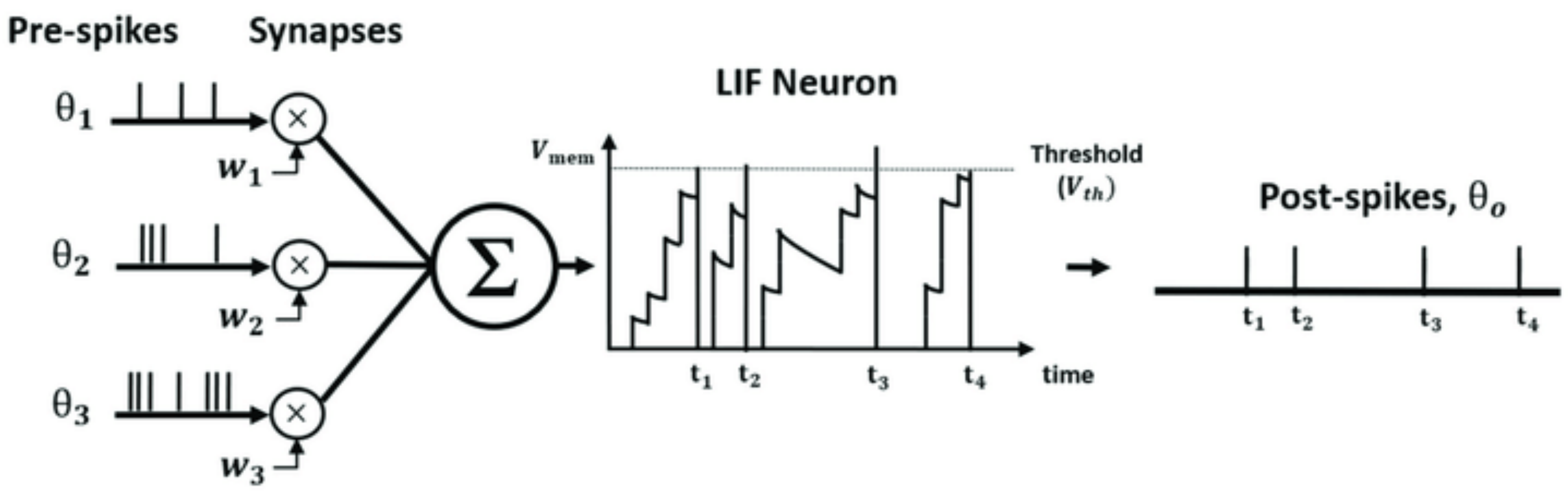
“Biologically feasible” learning rules

- Biological neurons cannot **back propagate** along synapses.
- **Question**: What is a “biologically feasible learning rule” for NNs that doesn’t “break the laws” of how neurons work?

$$\Delta w(\,\cdot\,) = ?$$

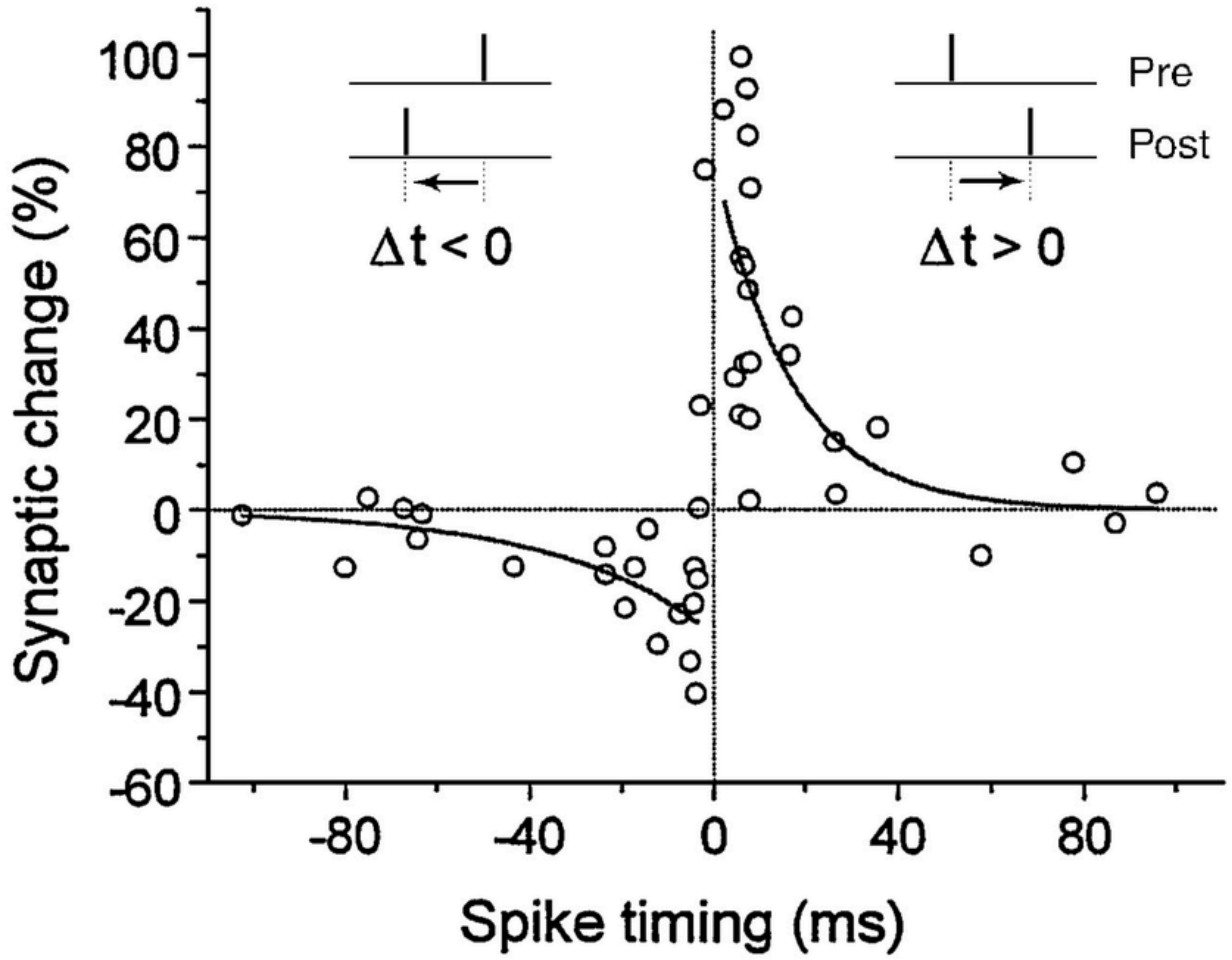
Hebbian Learning

“Neurons that fire together wire together”



Spike timing dependent plasticity (STDP)

“Neurons that <u>CAUSE EACH OTHER</u> to fire together wire together”



Reward correlation learning

The brain rewires itself to seek dopamine.

- Correlation between reward signal and synaptic connection change informs subsequent learning.
- **Credit assignment problem**: So many neurons, only 1 reward signal!

Hypotheses on Cellular Computation

An agent-based approach

- **Decisions**: Releasing neurotransmitters, adjusting synapse weights over time.
- **Memory**: Cells observe and maintain an **internal state** that informs computations over time.
- **Reward optimization**: Neurons appear to **cooperatively** seek reward via decisions.

$$\Delta w(\,\cdot\,) = ?$$

Hypotheses on ECONOMIC Computation

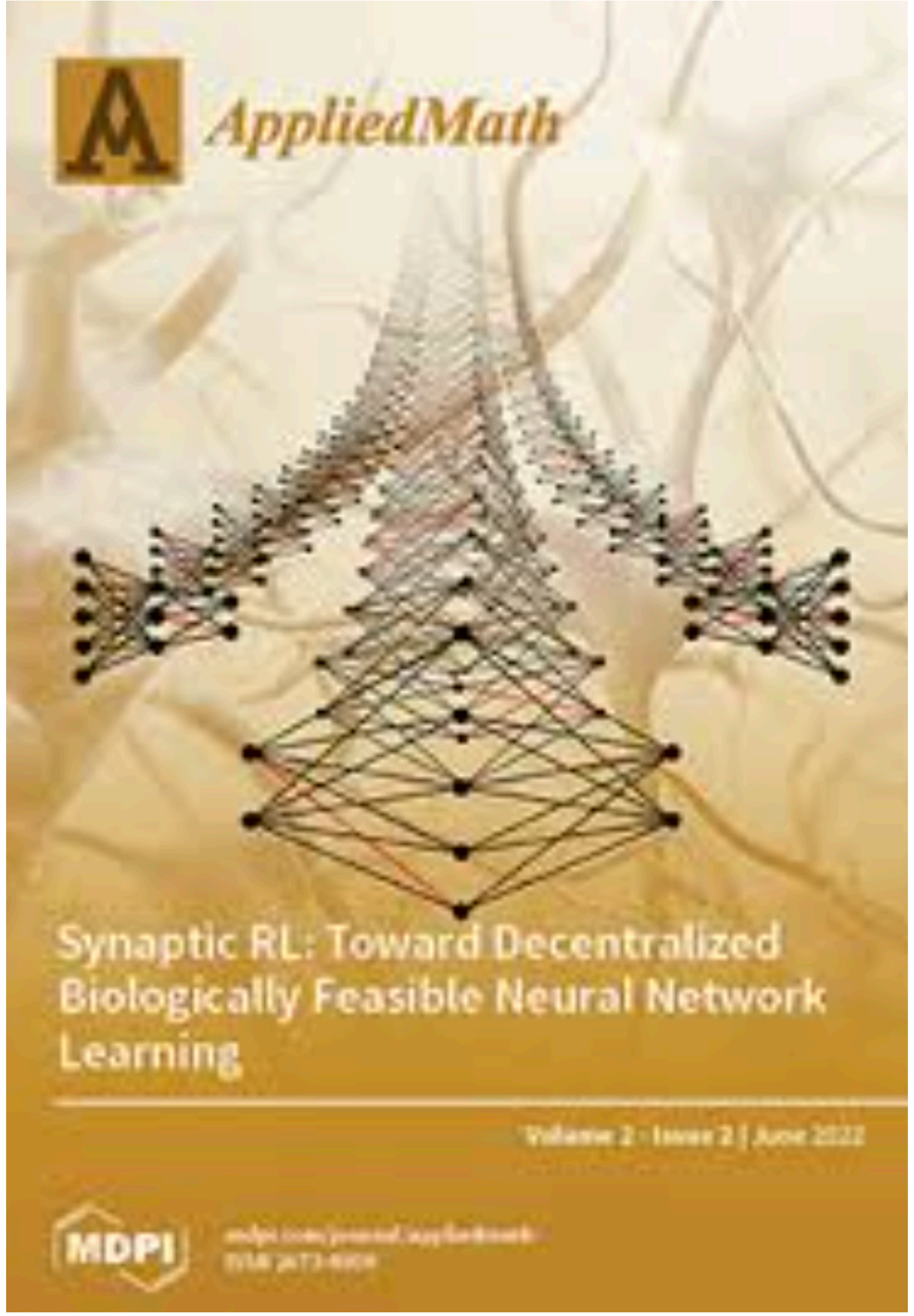
An agent-based approach

- **Decisions**: Releasing INFORMATION, adjusting MUTUAL CREDIT/CONNECTION/$ exchange over time.
- **Memory**: ECONOMIC AGENTS observe and maintain an **internal state** that informs computations over time.
- **Reward optimization**: ECONOMIC AGENTS appear to **cooperatively** seek reward via decisions.

$$\Delta w(\cdot) = ?$$

Hypotheses on Cellular Computation

An agent-based approach



AppliedMath

MDPI

Article

Gradient-Free Neural Network Training via Synaptic-Level Reinforcement Learning

Aman Bhargava [1,2], Mohammad R. Rezaei [2,3,4] and Milad Lankarany [2,3,4,5,*]

1 Division of Engineering Science, University of Toronto, Toronto, ON M5S 2E4, Canada; aman.bhargava@mail.utoronto.ca
2 Division of Clinical and Computational Neuroscience, Krembil Brain Institute, University Health Network, Toronto, ON M5G 1L7, Canada; mr.rezaei@mail.utoronto.ca
3 Institute of Biomedical Engineering, University of Toronto, Toronto, ON M5S 2E4, Canada
4 KITE, Toronto Rehabilitation Institute, University Health Network, Toronto, ON M5G 1L7, Canada
5 Department of Physiology, University of Toronto, Toronto, ON M5S 2E4, Canada
* Correspondence: milad.lankarany@uhnresearch.ca

Abstract: An ongoing challenge in neural information processing is the following question: how do neurons adjust their connectivity to improve network-level task performance over time (i.e., actualize learning)? It is widely believed that there is a consistent, synaptic-level learning mechanism in specific brain regions, such as the basal ganglia, that actualizes learning. However, the exact nature of this mechanism remains unclear. Here, we investigate the use of universal synaptic-level algorithms in training connectionist models. Specifically, we propose an algorithm based on reinforcement learning (RL) to generate and apply a simple biologically-inspired synaptic-level learning policy for neural networks. In this algorithm, the action space for each synapse in the network consists of a small increase, decrease, or null action on the connection strength. To test our algorithm, we applied it to a multilayer perceptron (MLP) neural network model. This algorithm yields a static synaptic learning policy that enables the simultaneous training of over 20,000 parameters (i.e., synapses) and consistent learning convergence when applied to simulated decision boundary matching and optical character recognition tasks. The trained networks yield character-recognition performance comparable to identically shaped networks trained with gradient descent. The approach has two significant advantages in comparison to traditional gradient-descent-based optimization methods.

check for updates

Citation: Bhargava, A.; Rezaei, M.R.; Lankarany, M. Gradient-Free Neural

The Problem with Modern Learning Rules Research

- There are very few **“feed-forward”** brain regions.
- No clear **output label/error signal** for most brain regions.
- Most proposed learning rules have ~3 terms in the cell state.

The Problem with Modern Learning Rules Research

- There are very few **“feed-forward”** brain regions.
- No clear **output label/error signal** for most brain regions.
- Most proposed learning rules have ~3 terms in the cell state.

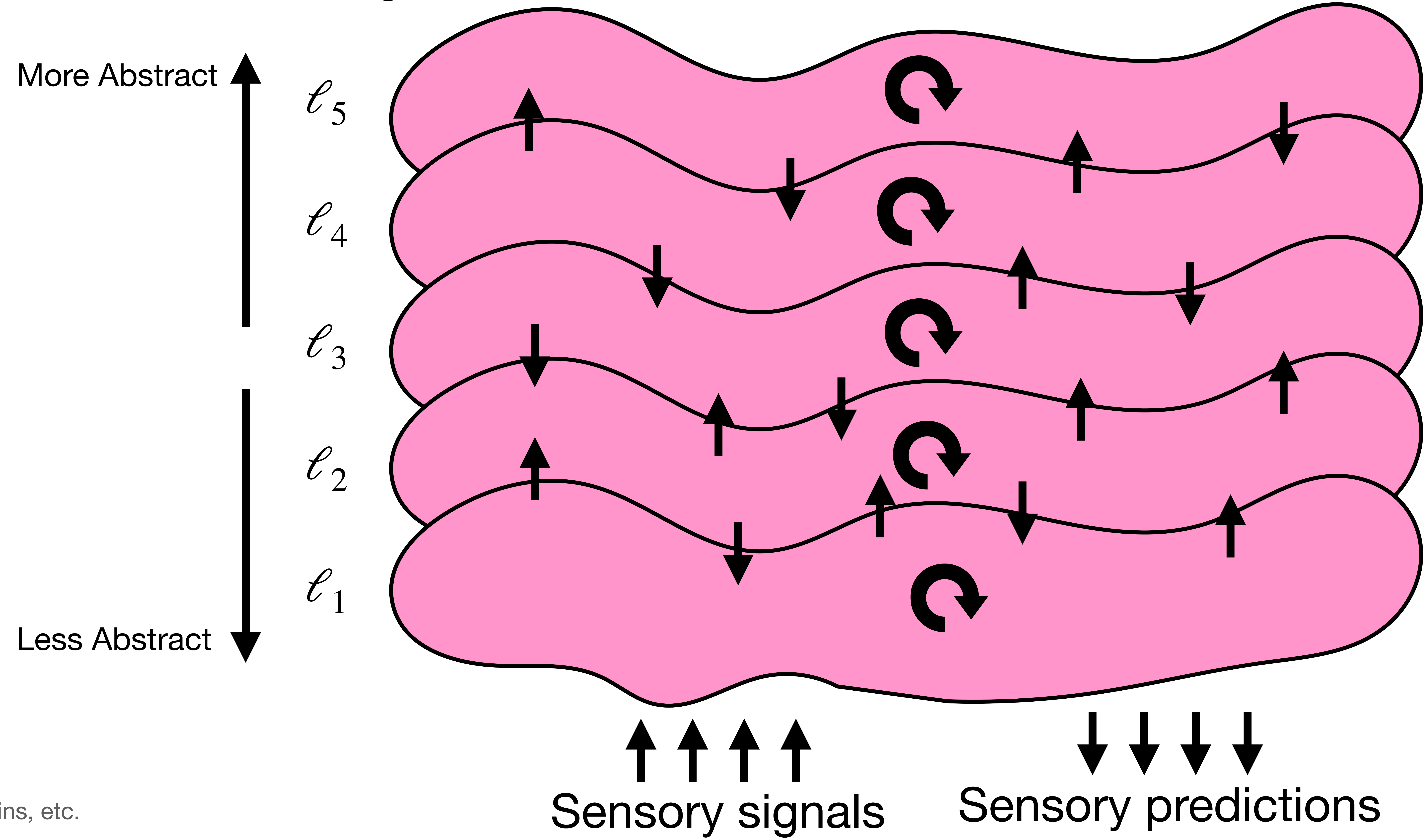
Real brains are a <u>distributed system</u> of billions of microcomputers running <u>the most confusing code</u> in the universe!

What aspects of biological computation are fundamental?

- **Locality** of information processing → low power requirements.
- **Simplicity** of each node (neuron) → easily build massive, scalable systems.
- **Asynchronous operation** → low latency.
- **Predictive coding** objective → highly general method for building world models + representations.

The cortex predicts what will happen next.

Cortical processing cartoon sketch



FEP, 1000 Brains, etc.

LLMs predict the next token.

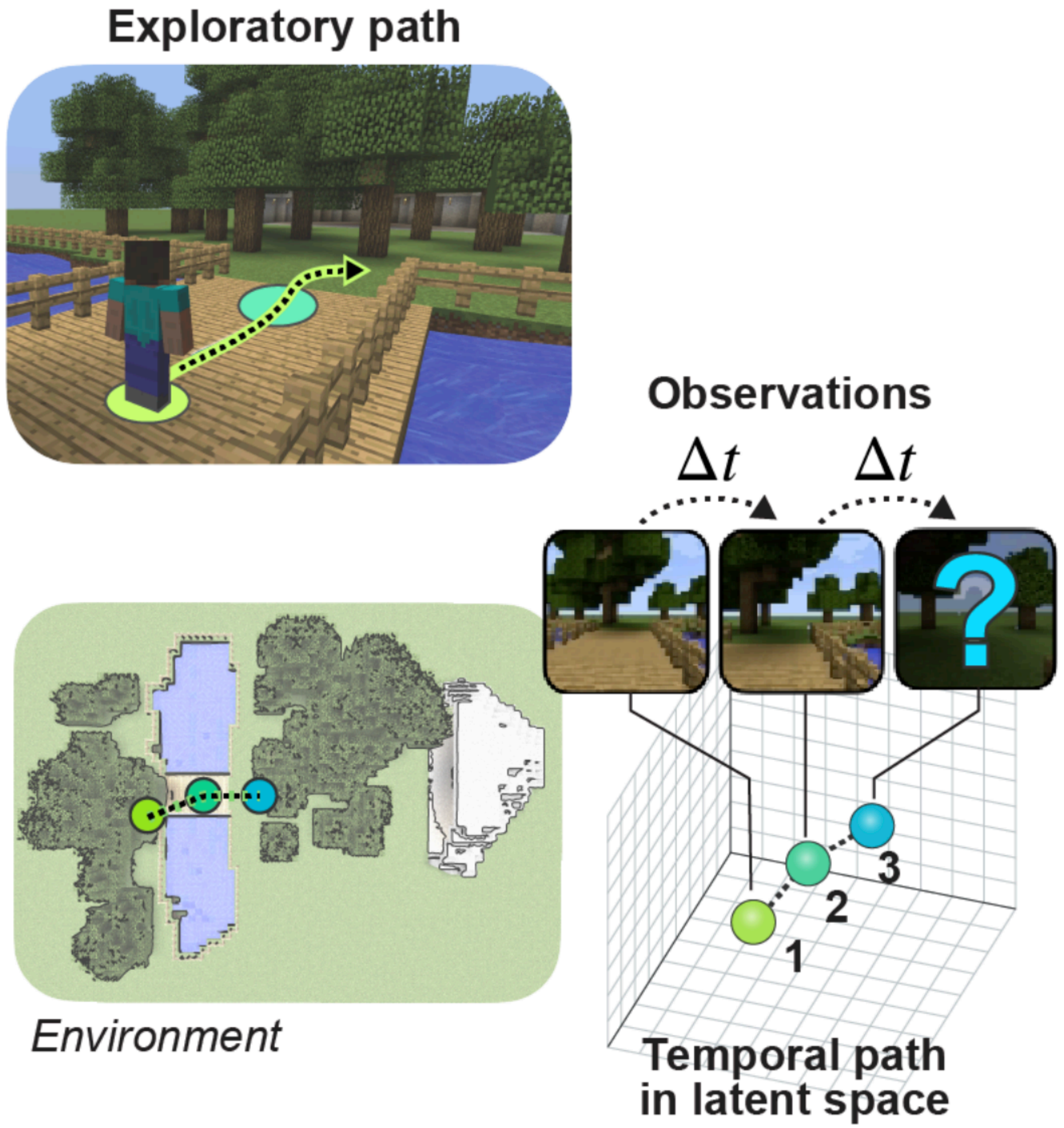
```
Try to predict the next token!

[22170, 311, 7168, 279, 1828, 4037, 0]
```

x_1 x_2 x_3 x_4 x_5 x_6 x_7

$$P_\theta(x_{t+1} \mid x_1, \ldots, x_t)$$

World Models emerge from Predictive Coding.

James Gornet (Thomson Lab, CNS)



James!

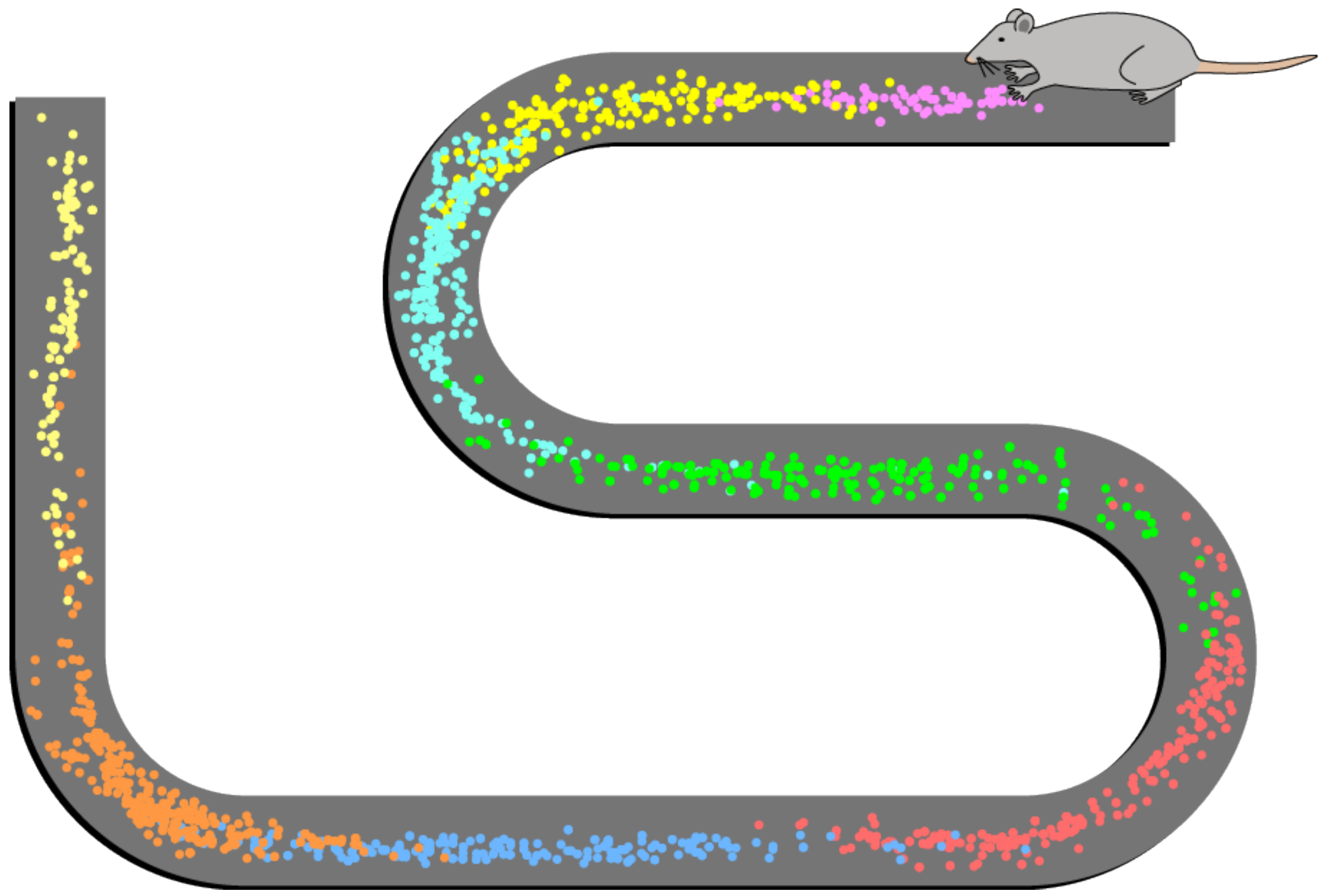
From “Automated mapping of virtual environments with visual predictive coding” (Gornet and Thomson) — arxiv:2308.10913

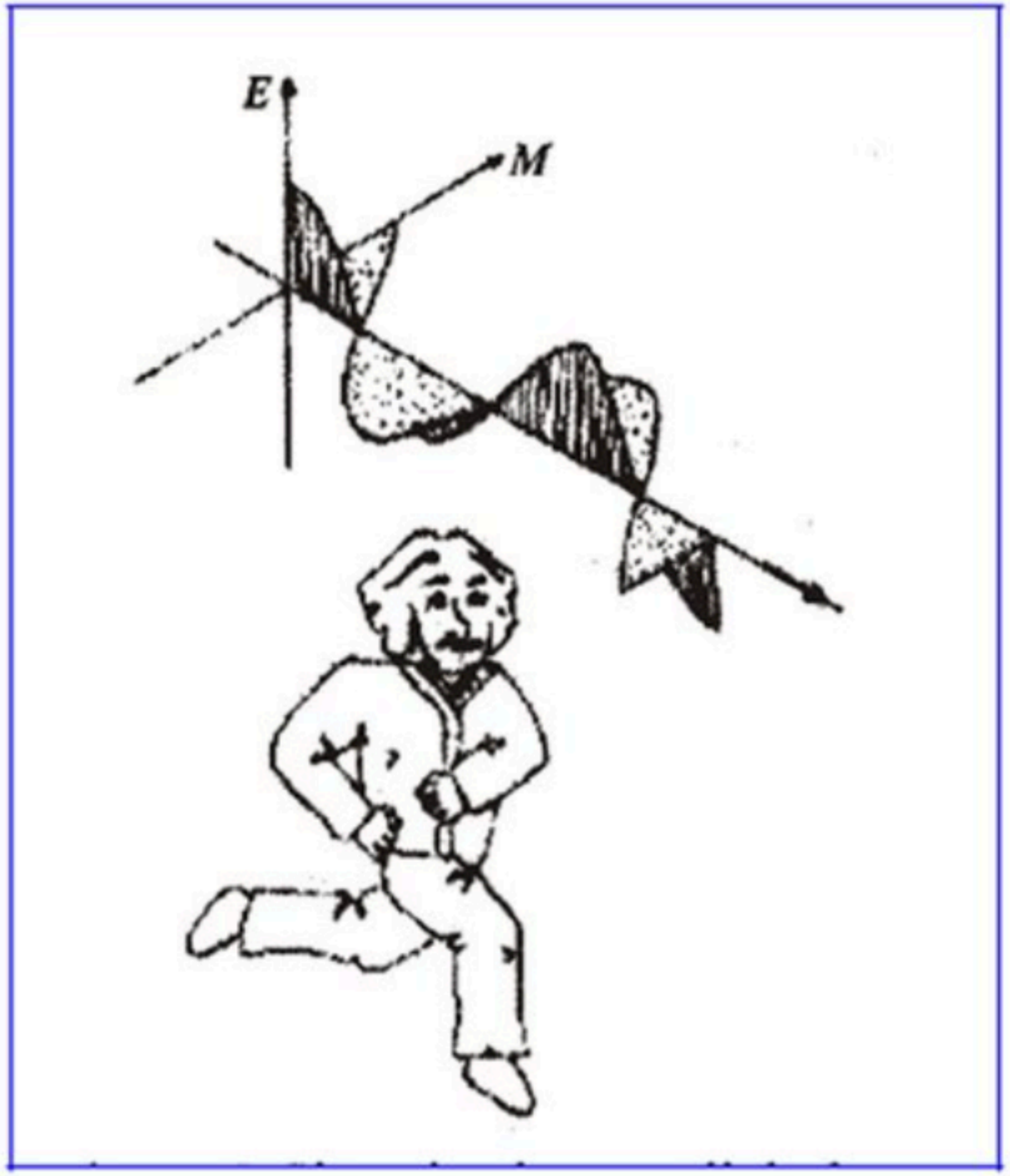
Thinking is internal world model exploration.

Leveraging *implicit understanding*.

- **Neuro**: We observe rodents "mentally exploring" a maze in place cell activations.

- **Phil of Mind**: Thought experiments, hypotheticals, imagined movement/ environments.

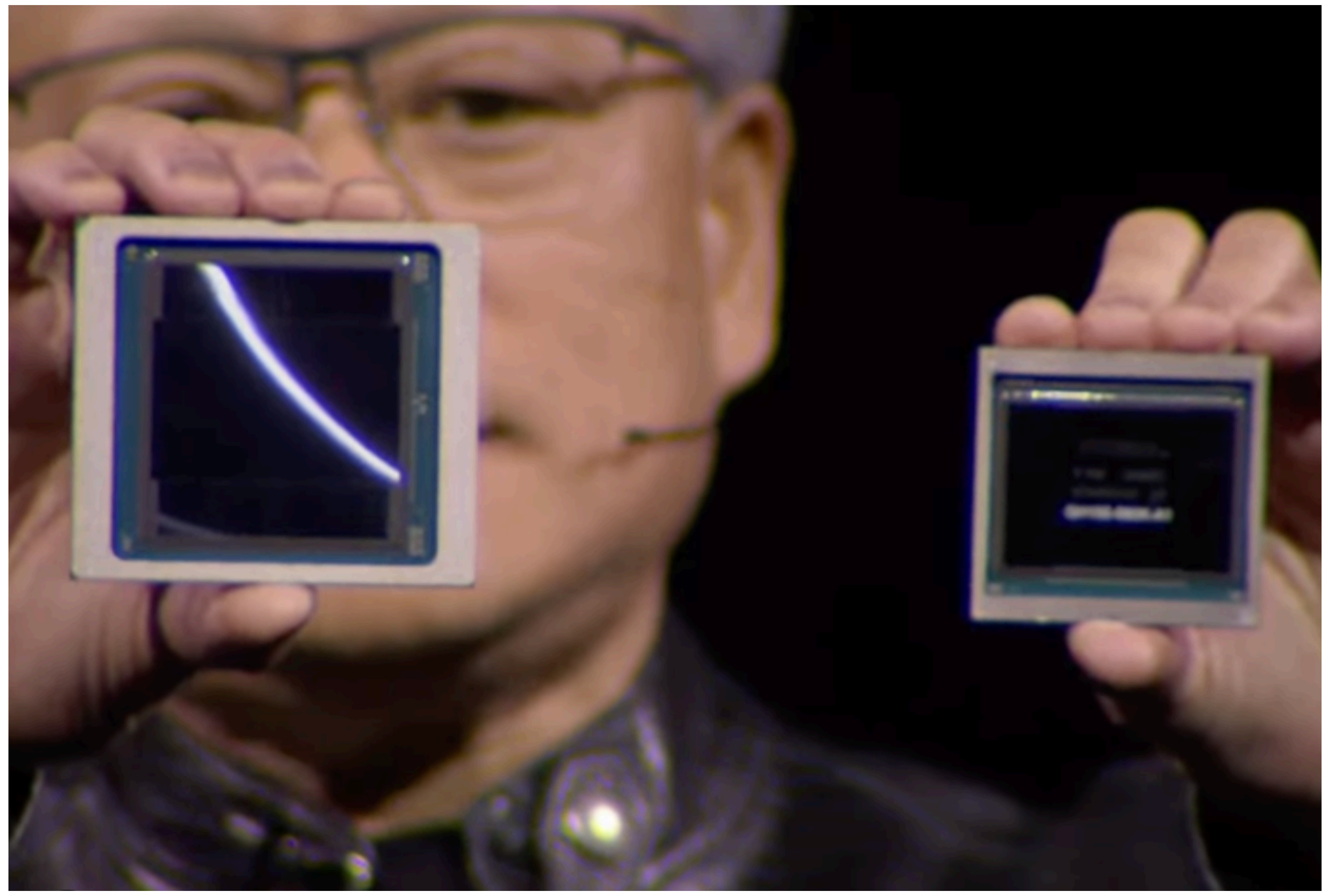


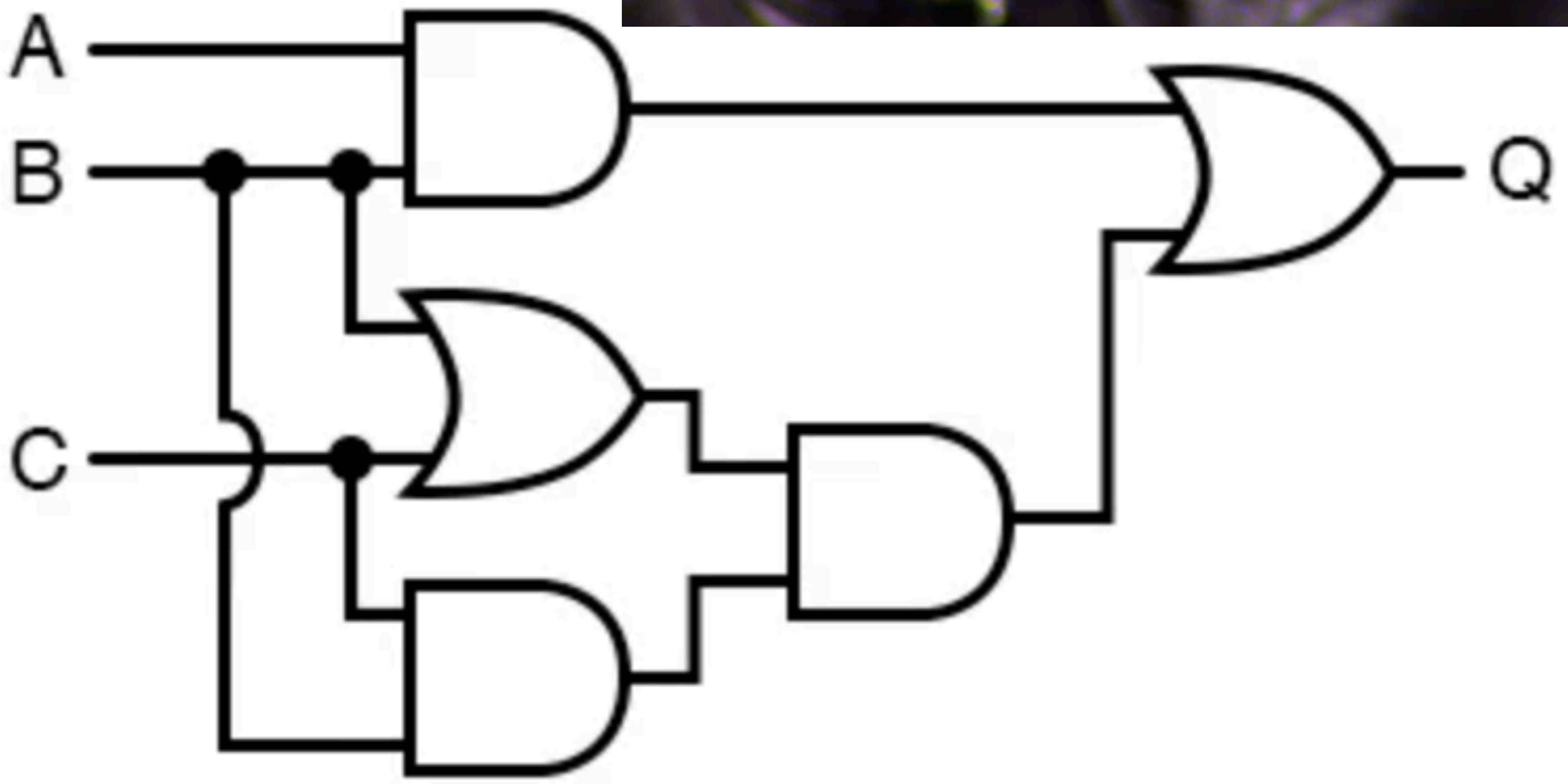


Beyond Modern Neuromorphics

Toward Cellular Computing

Sand

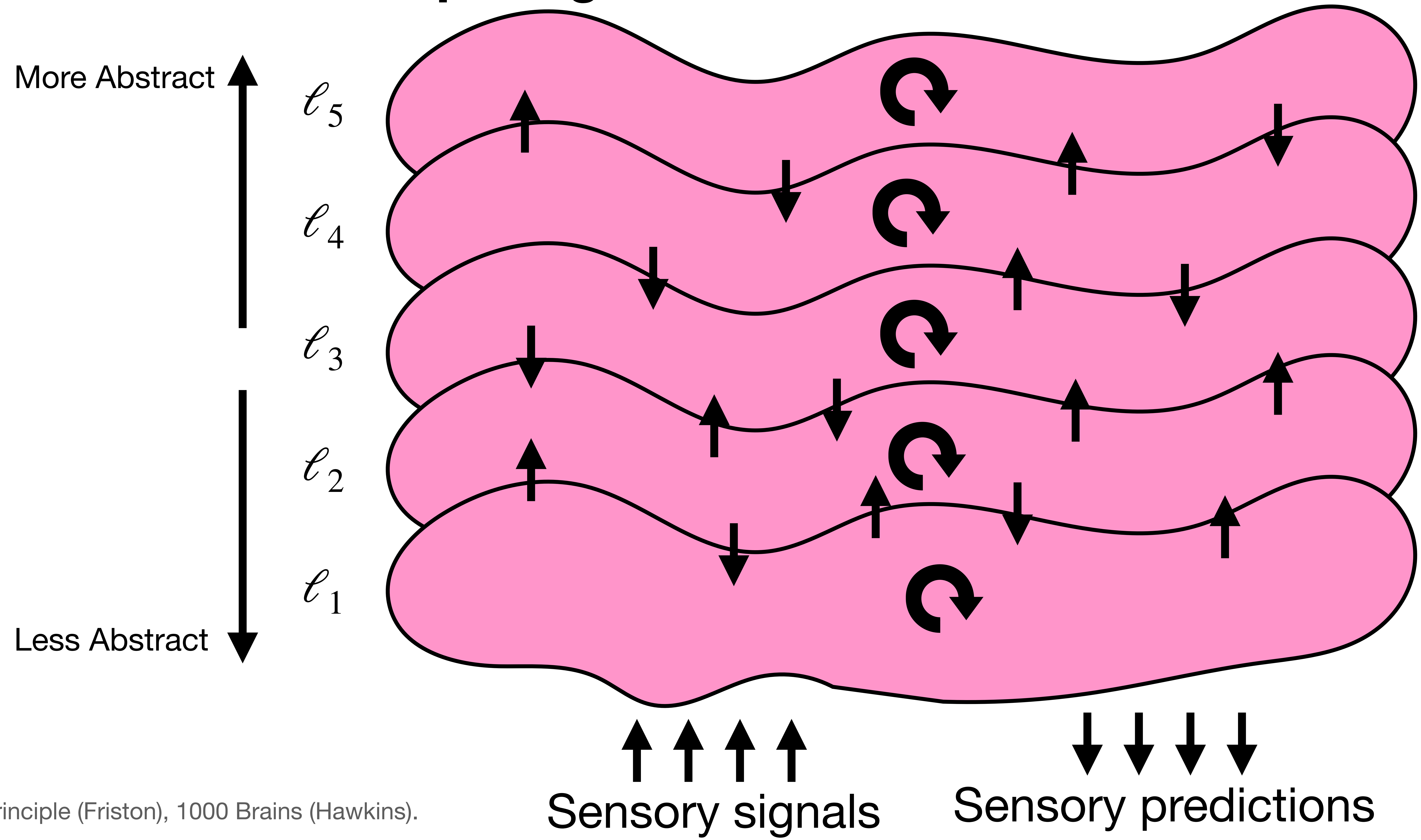




Thinking Sand

Beyond LLMs

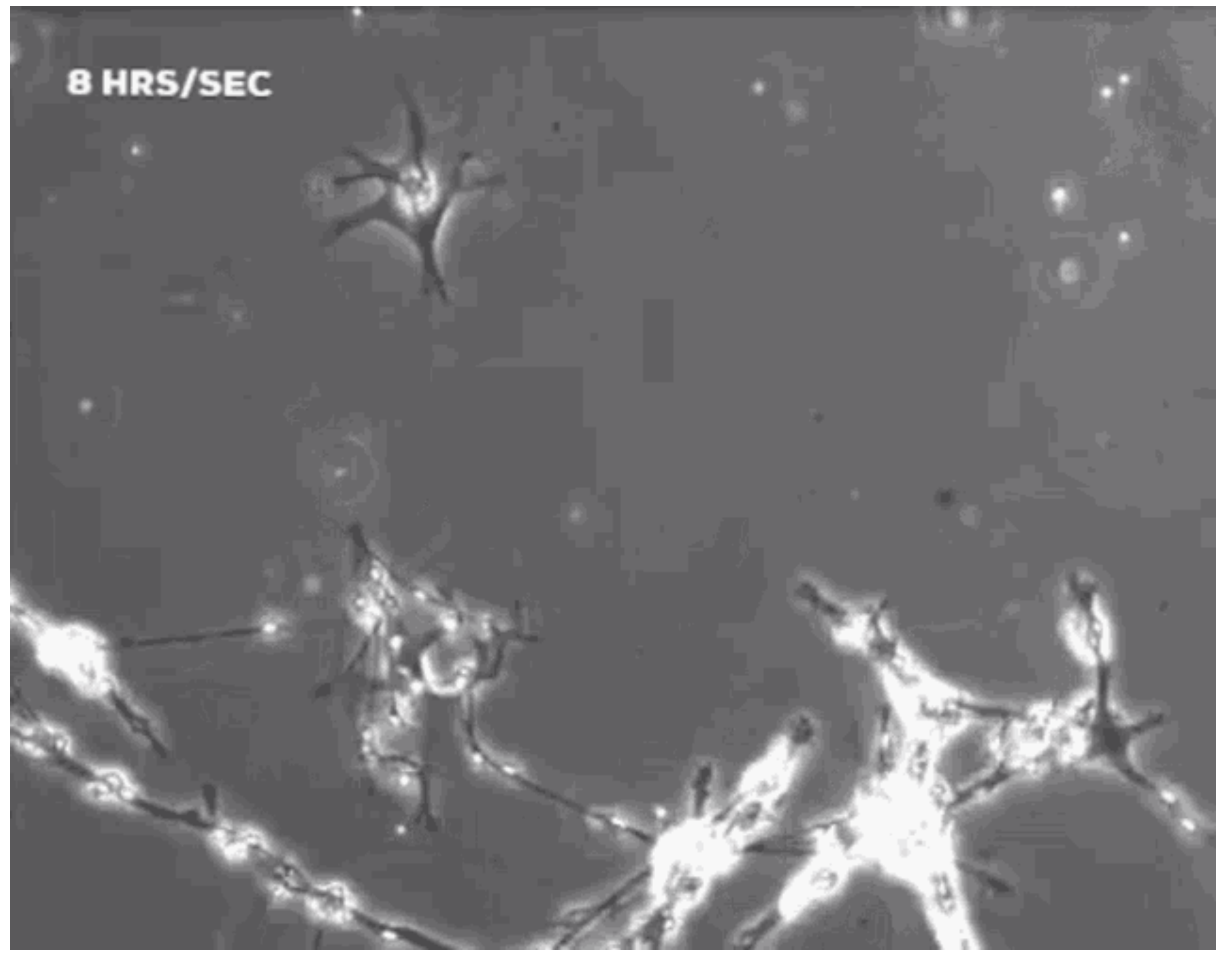
Toward Cellular Computing

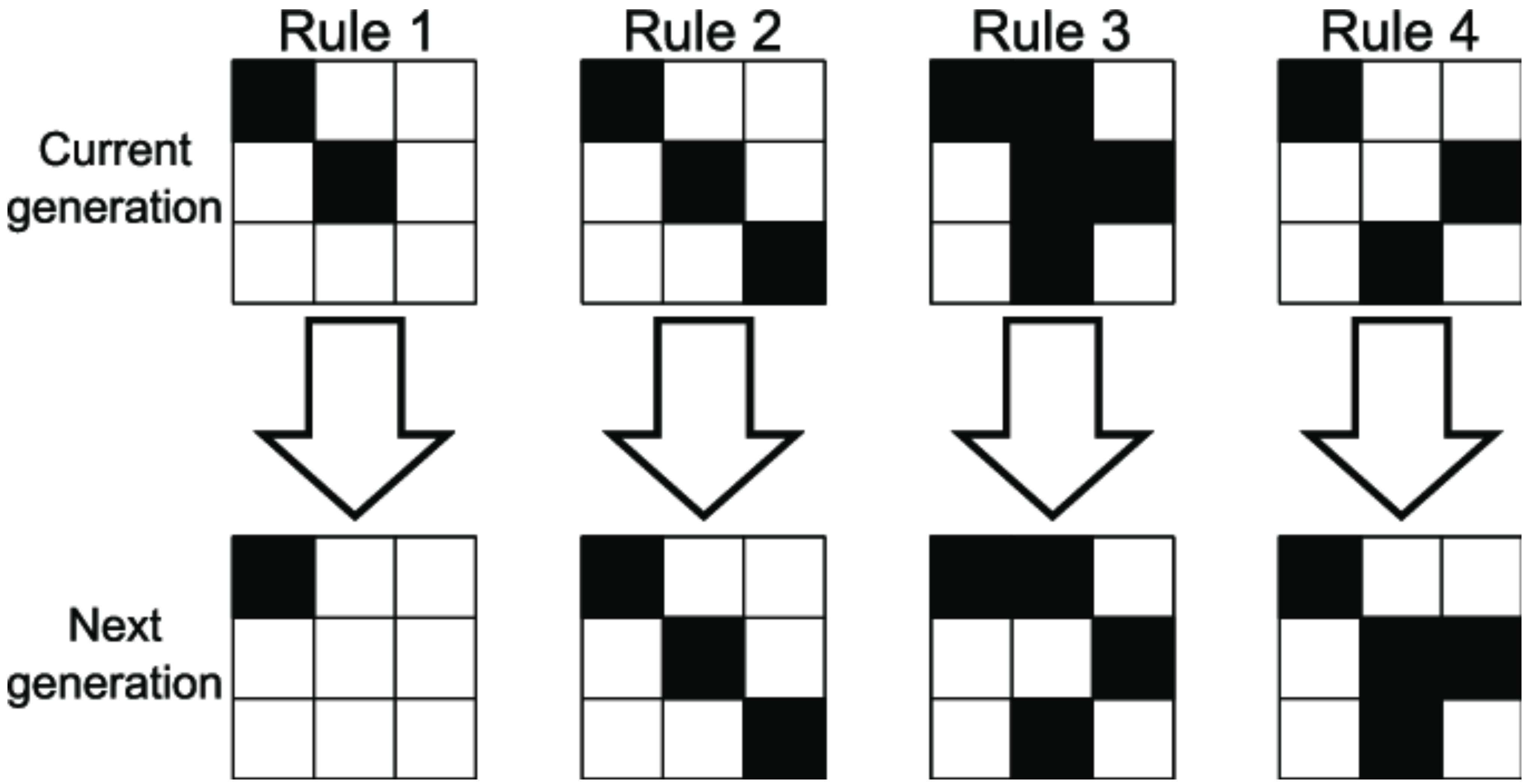


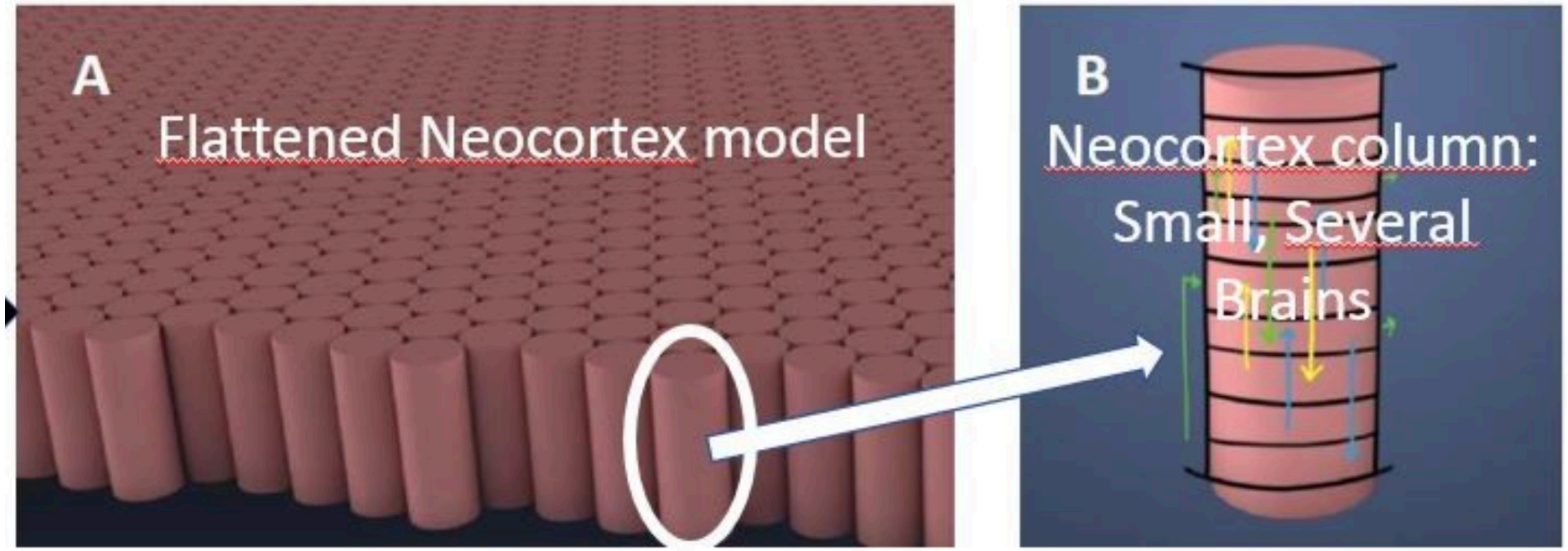
Free Energy Principle (Friston), 1000 Brains (Hawkins).

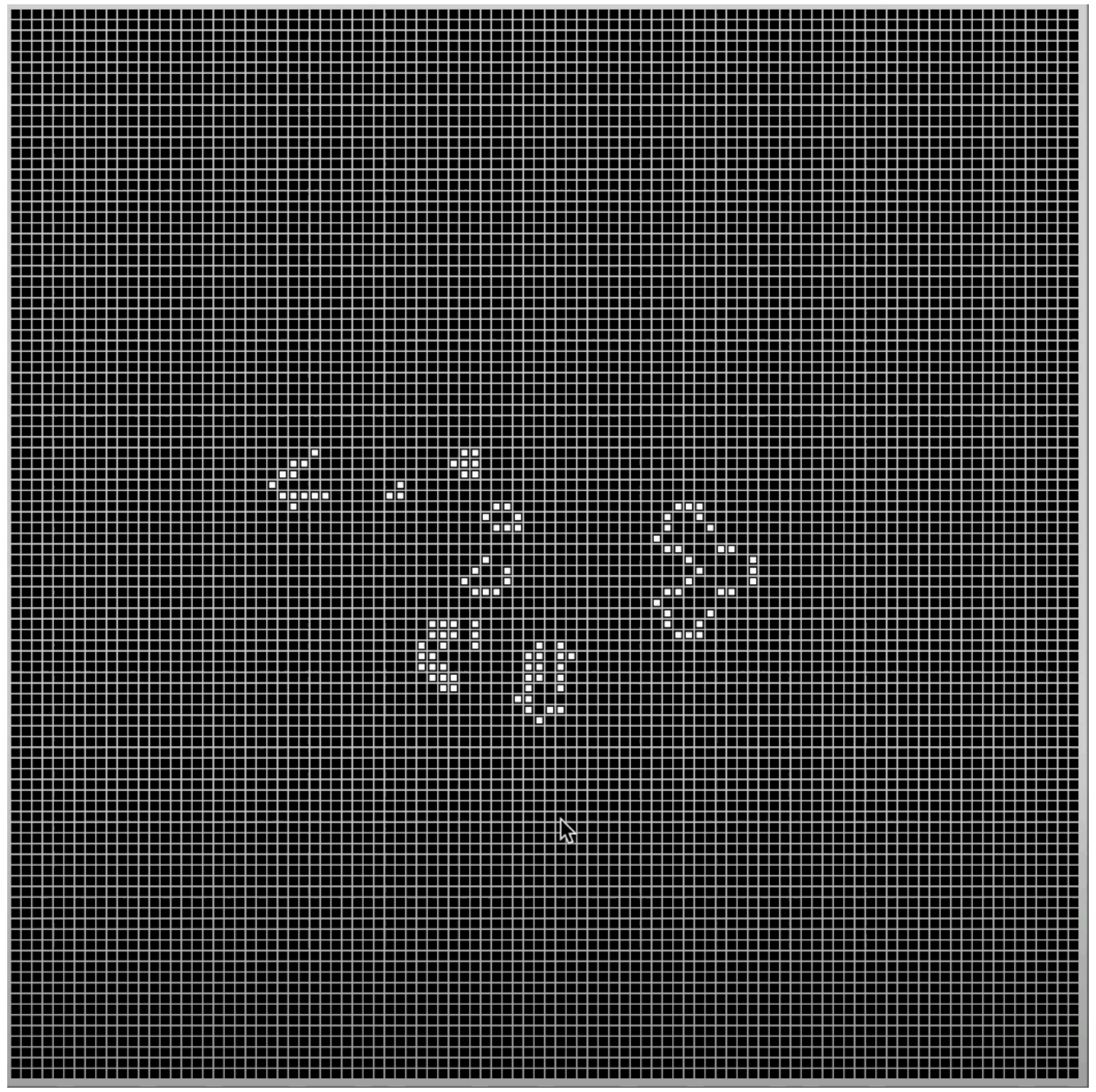
Beyond LLMs

Cortical Universal Update Rule





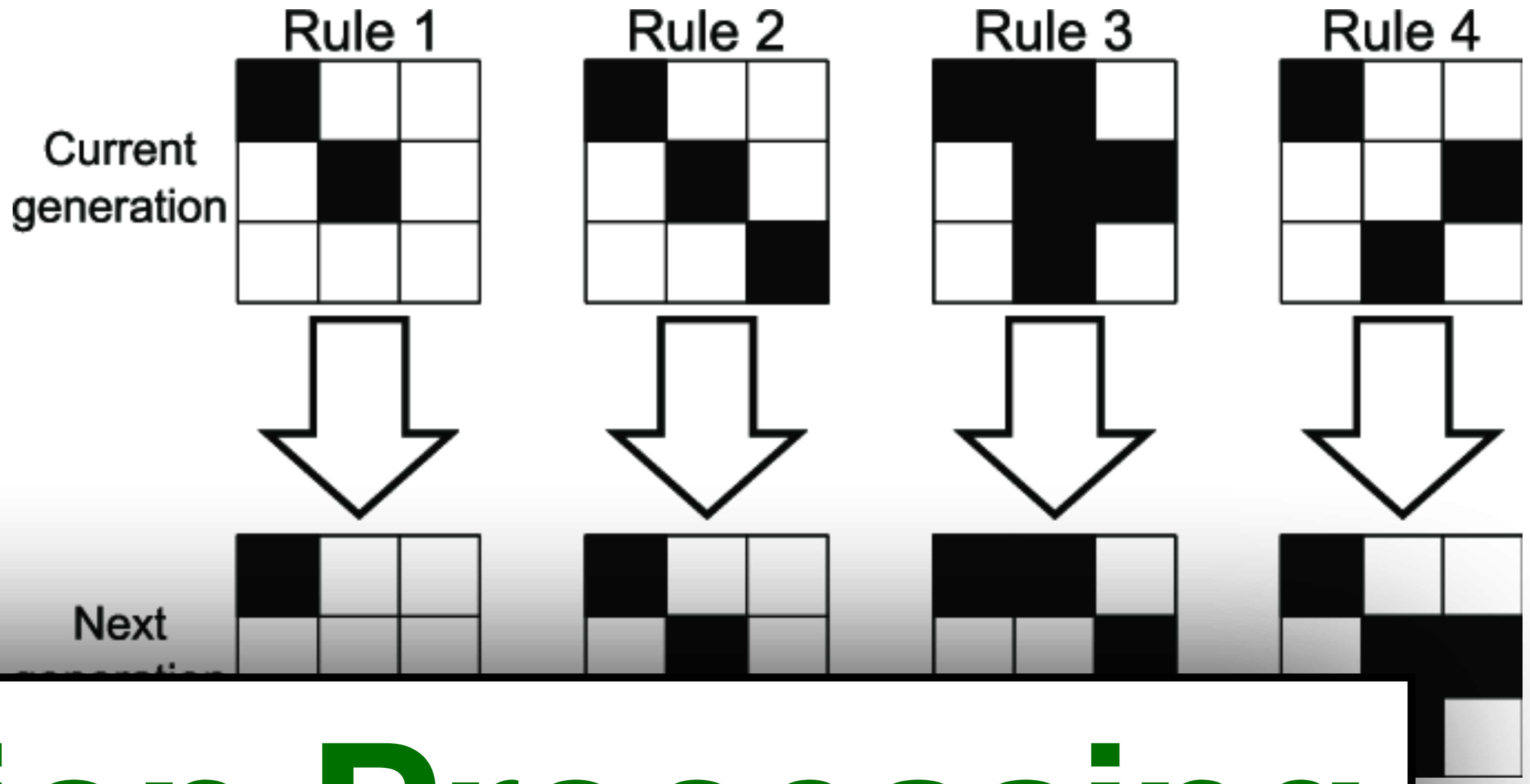




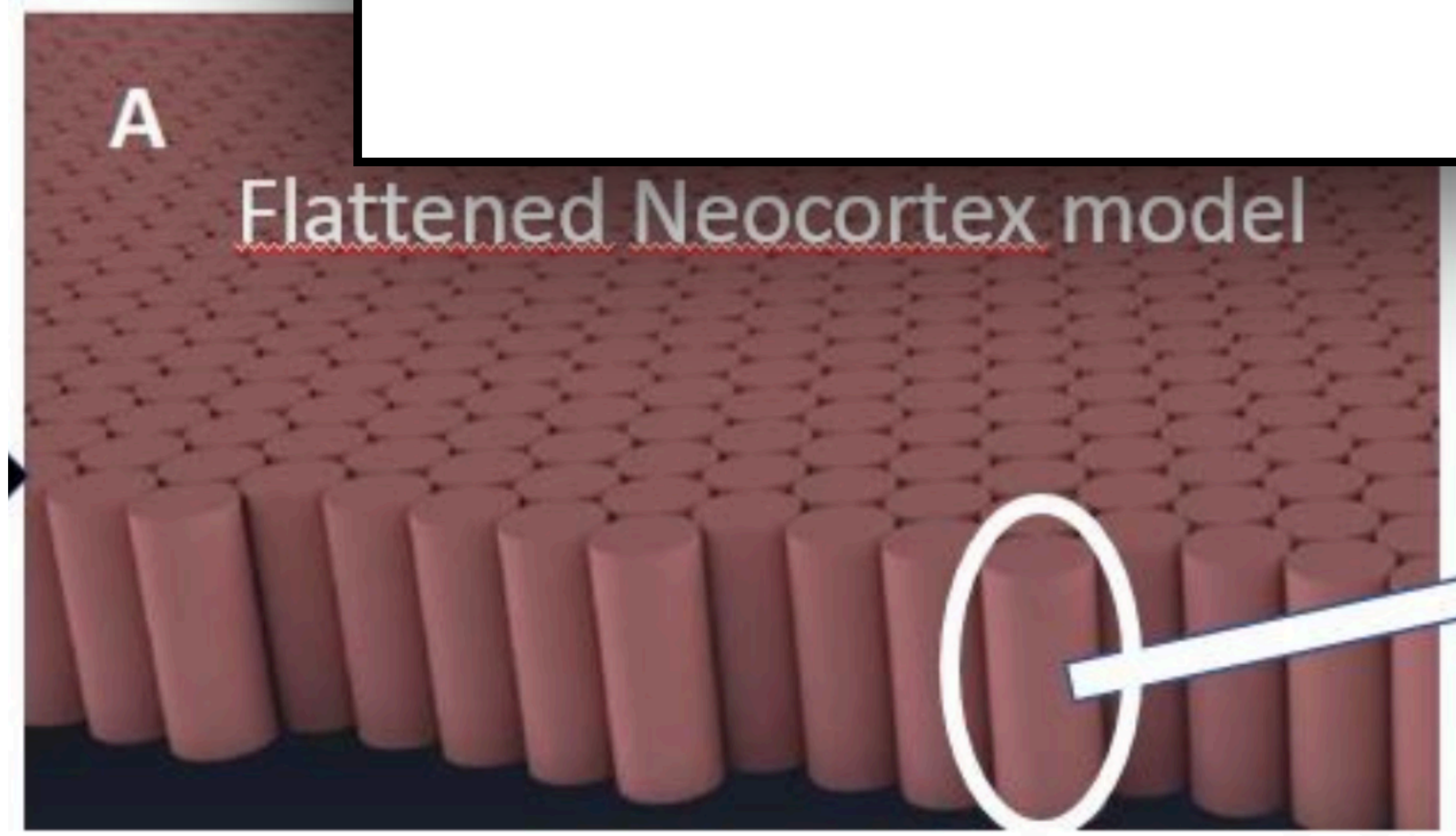
FEP (Friston), 1000 Brains (Hawkins), New Kind of Science (Wolfram), etc.

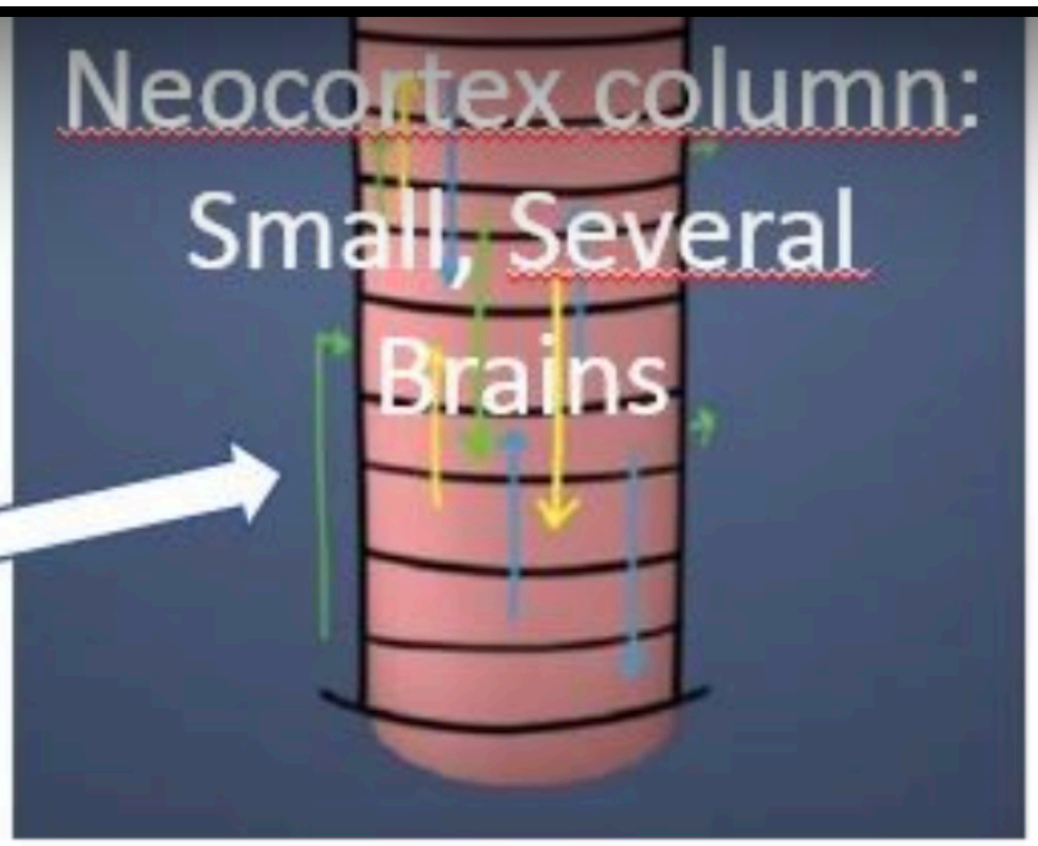
Beyond LLMs

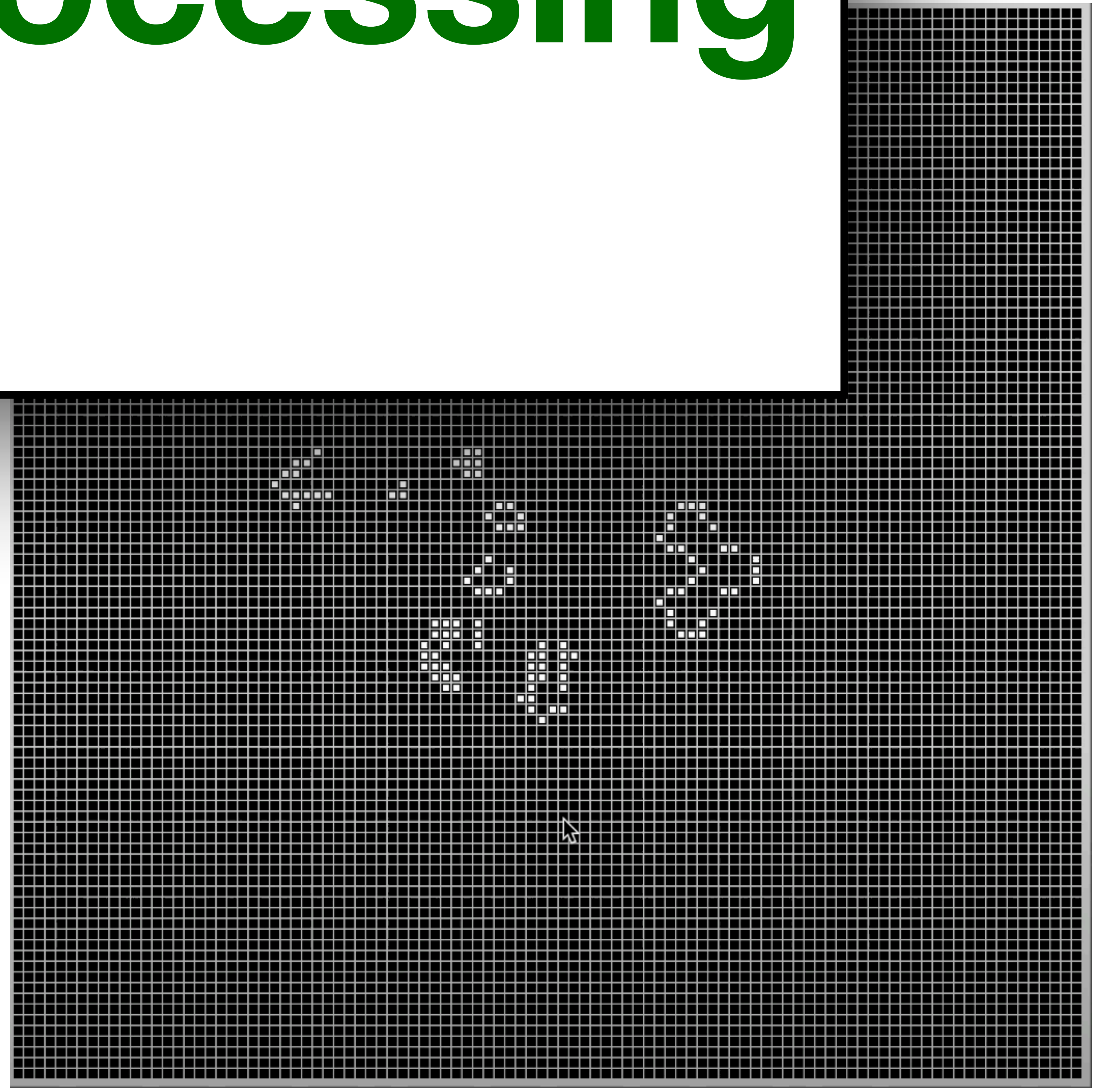
Cortical Universal Update Rule



Local Information Processing is Key!



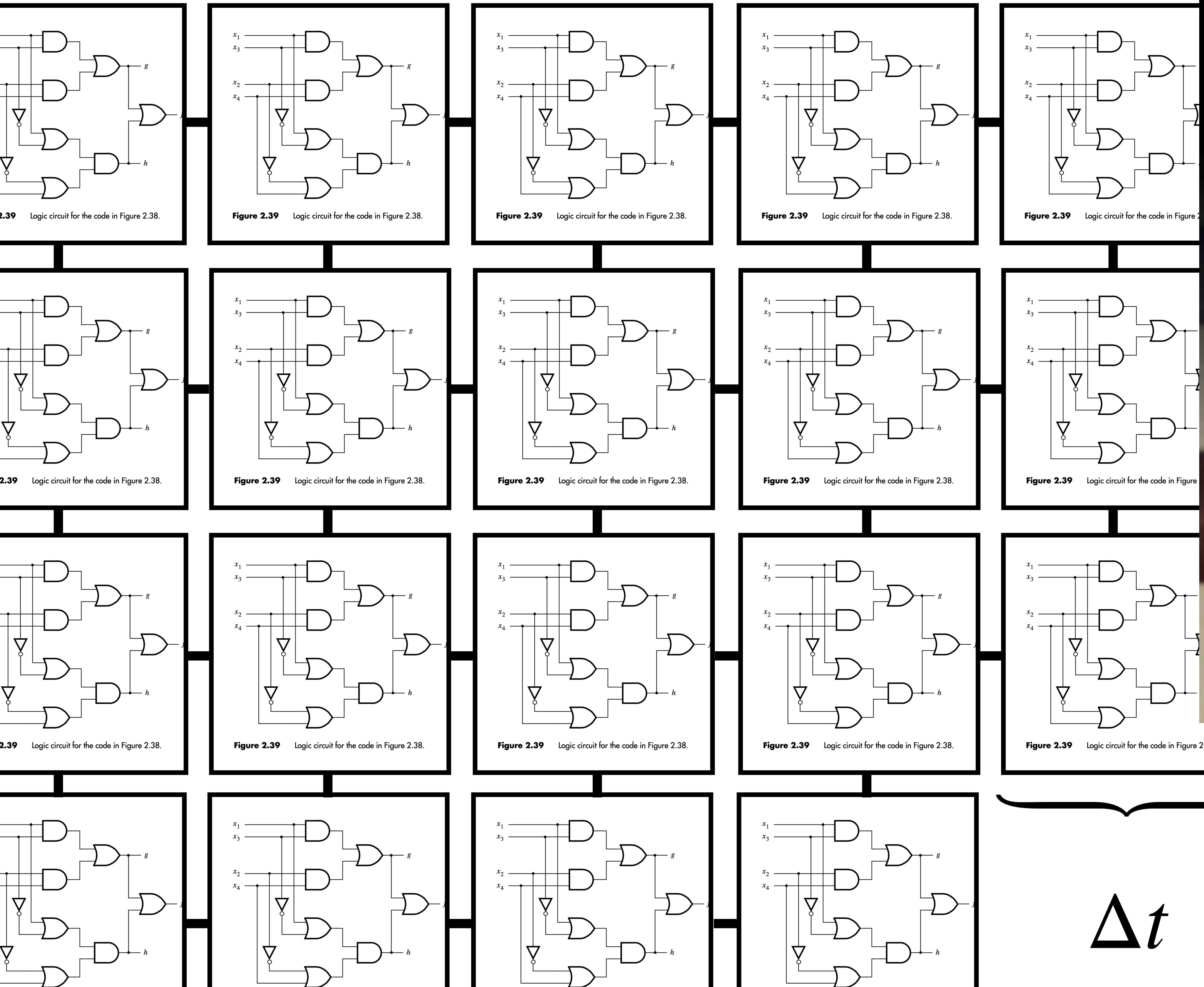


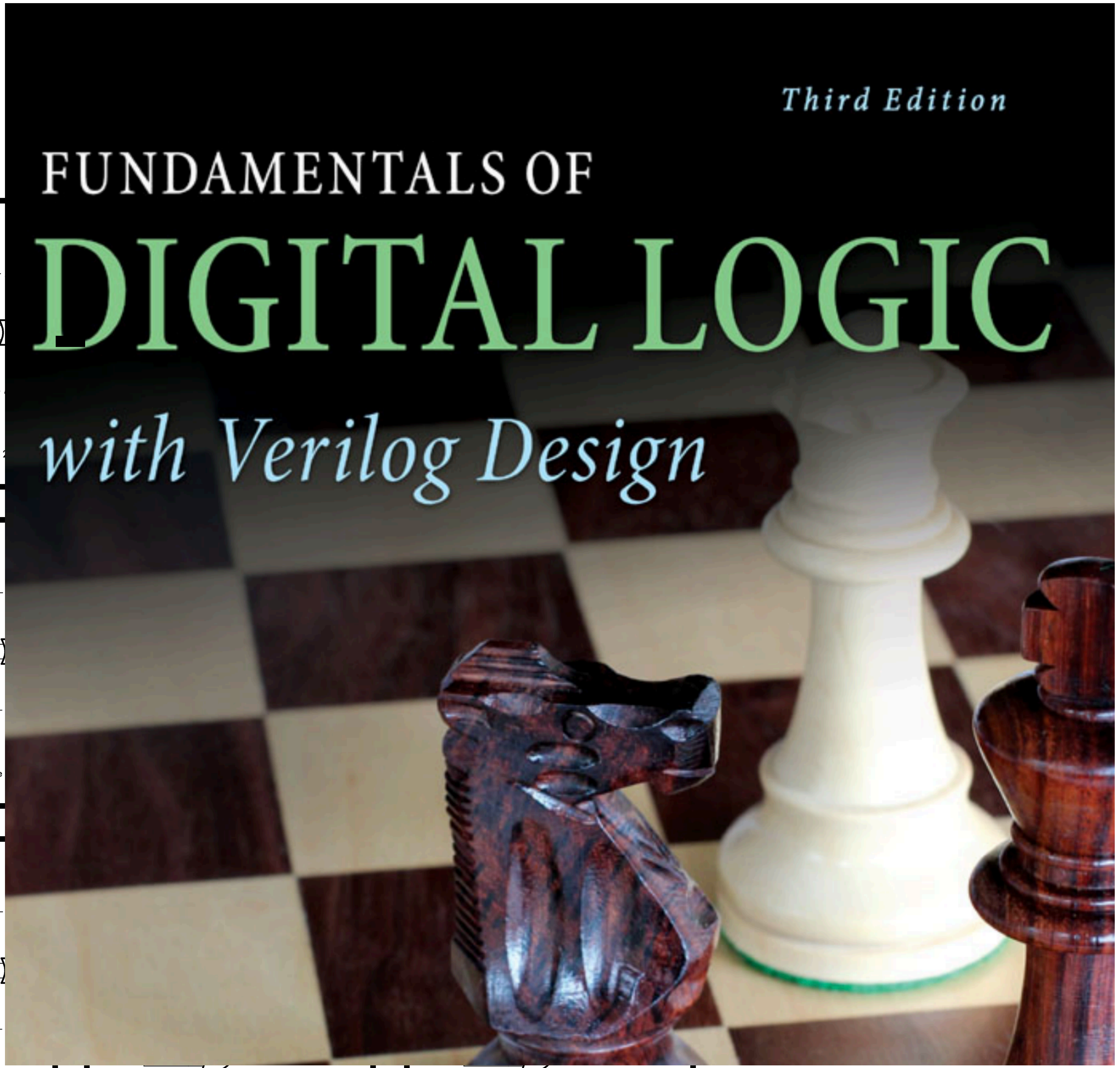


FEP (Friston), 1000 Brains (Hawkins), New Kind of Science (Wolfram), etc.

Beyond LLMs

Cortical Universal Update Rule



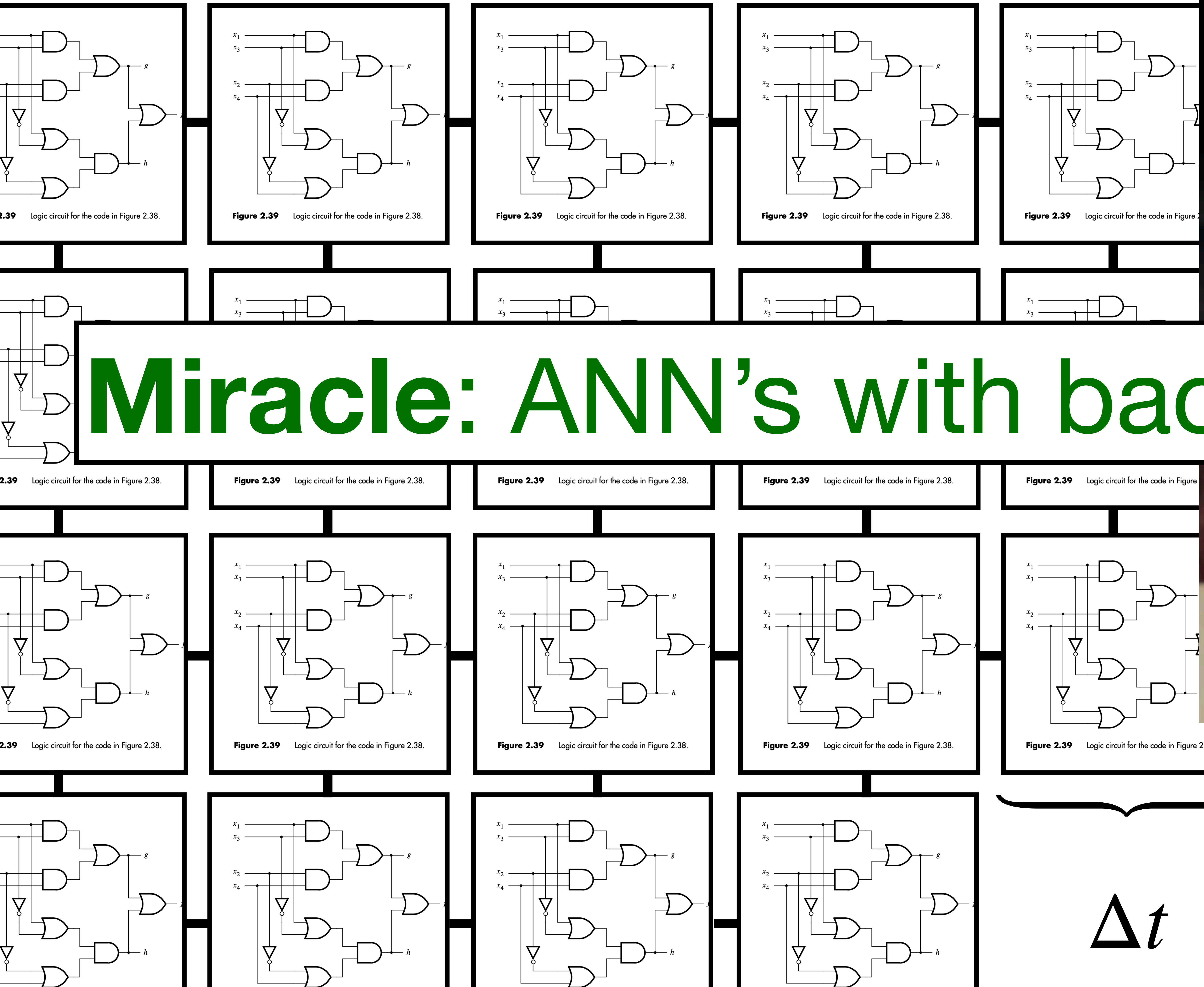


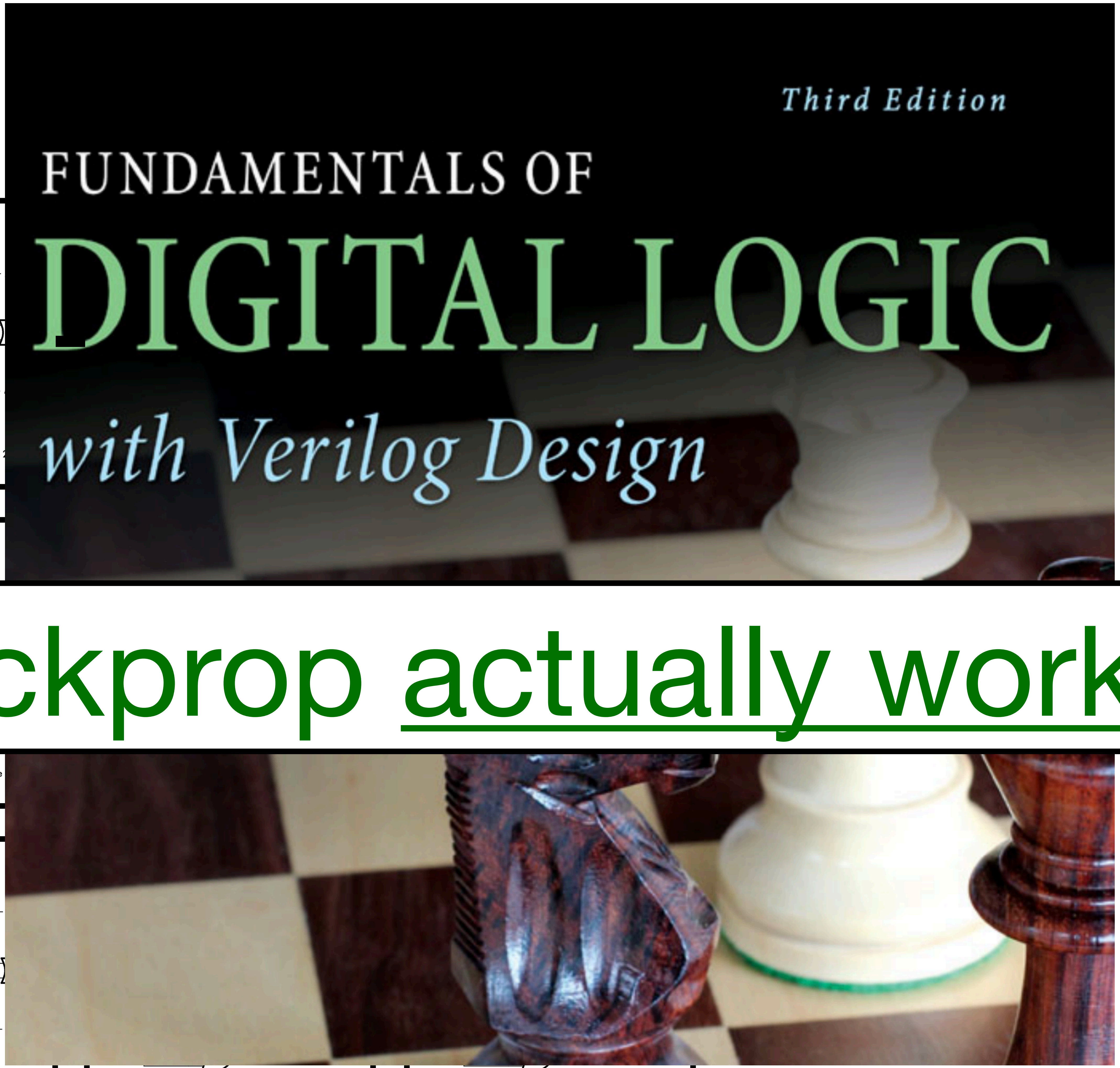
Δt

FEP (Friston), 1000 Brains (Hawkins), New Kind of Science (Wolfram), etc.

Beyond LLMs

Cortical Universal Update Rule





Miracle: ANN’s with backprop <u>actually work!</u>

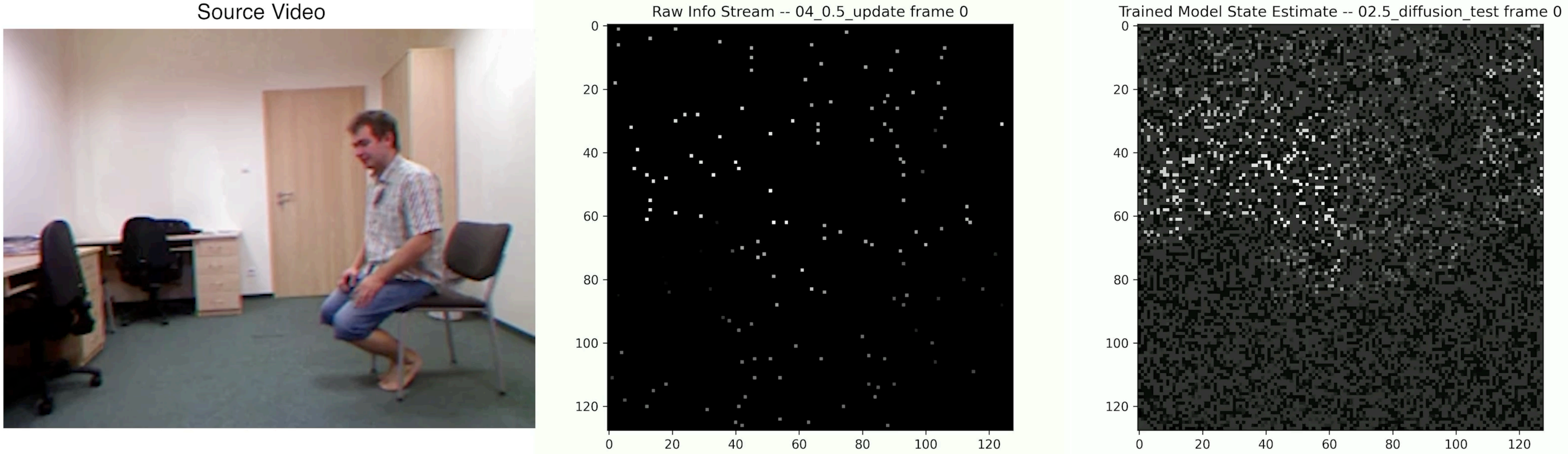
Δt

FEP (Friston), 1000 Brains (Hawkins), New Kind of Science (Wolfram), etc.

Toward Cellular Computing

Limit case of physically-realizable computation

Neural Cellular Automata do Active Inference

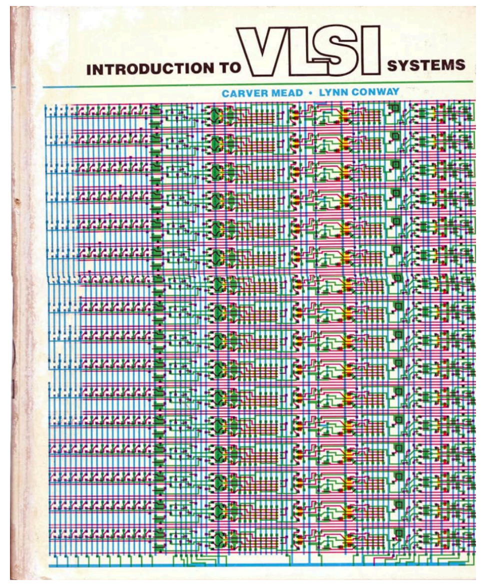


aman-bhargava.com

Toward Cellular Computing

Limit case of physically-realizable computation

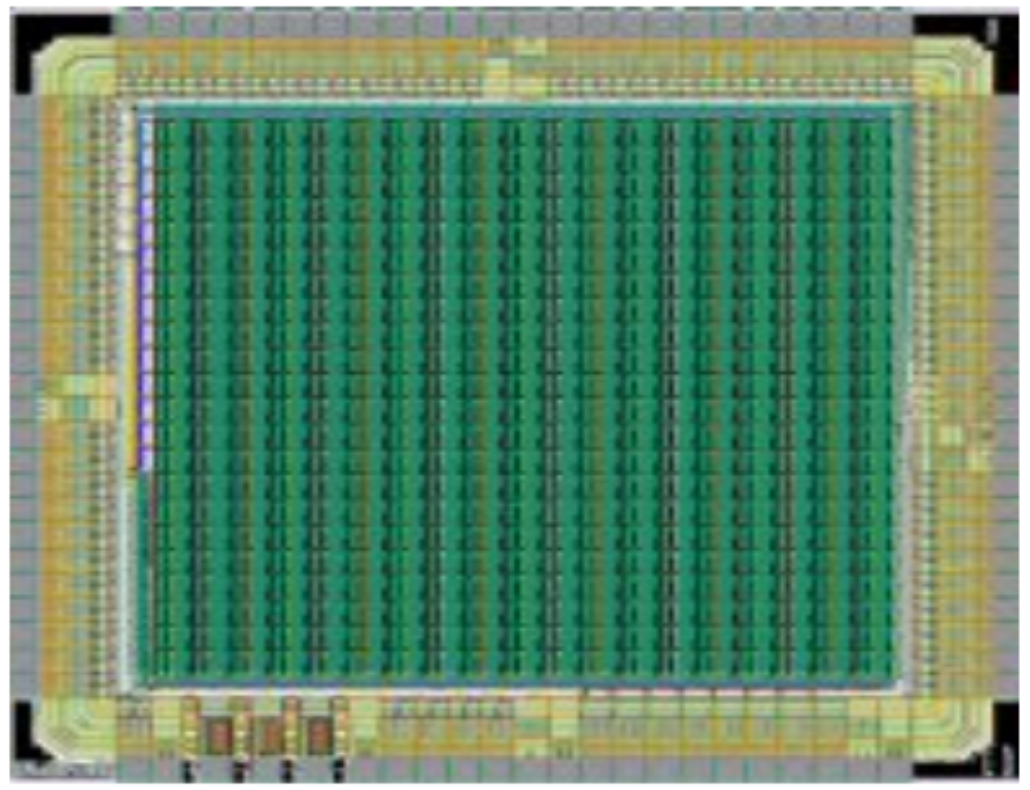




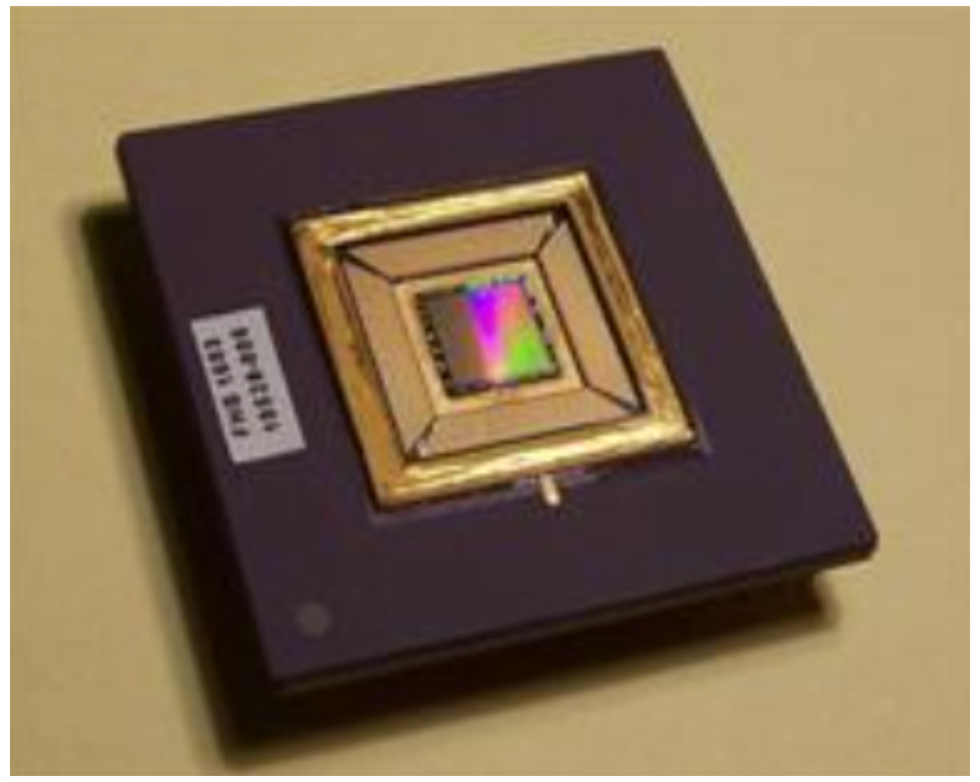
Mead/Conway Neuromorphics

Piotr Dudek

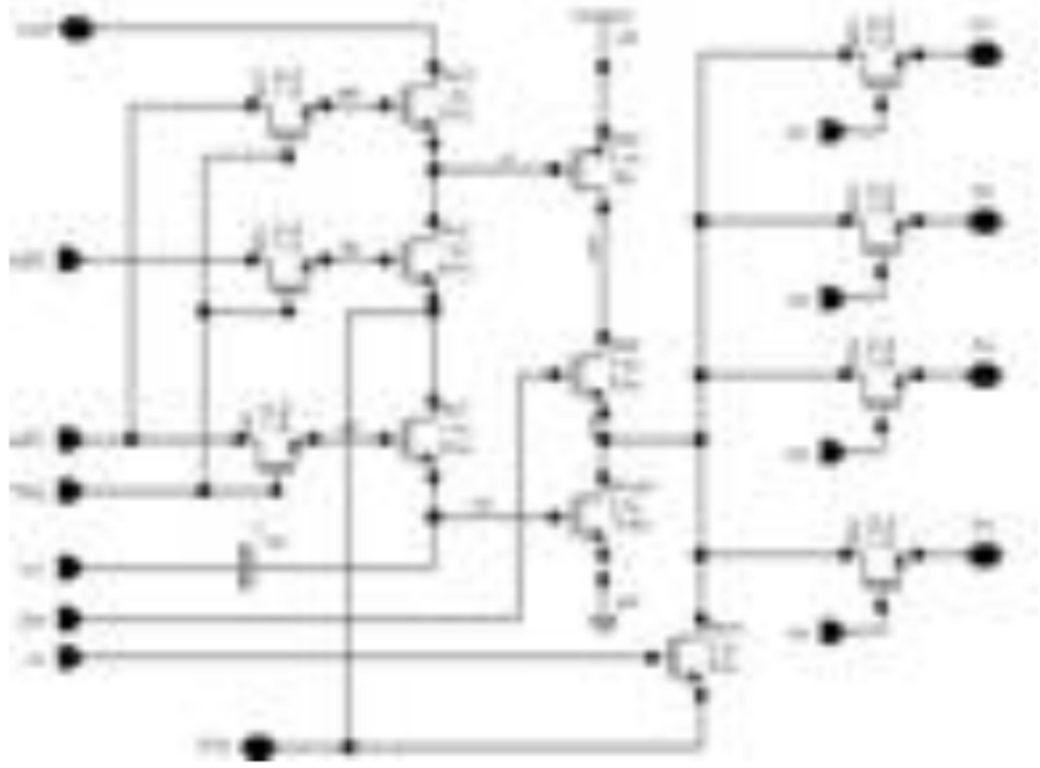
The University of Manchester
Verified email at manchester.ac.uk - Homepage
VLSI Design   Neuromorphic Engineering   Imag



Cellular processors arrays



Vision Chips



Analog Sig Proc

https://personalpages.manchester.ac.uk/staff/p.dudek/

Thank You!

Questions?

Mentors

- Matt Thomson
- Erik Winfree
- Ralph Adolphs
- Frederick Eberhart
- Rob Phillips
- Steve Mann (UToronto)
- Milad Lankarany (UToronto)
- Michael Levin (Tufts)

Friends + Collaborators

- Pantelis Vafidis
- Cameron Witkowski (UToronto)
- Salvador Buse
- Cayden Pierce (MIT)
- James Gornet
- Meera Prasad
- Michael Zellinger
- Hersh Bhargava (UCSF)
- Mango Weng
- Mingshi Chi (UToronto/York)

Neural Cellular Automata do Active Inference