Braindrop: A Mixed-Signal Neuromorphic System that Presents Clean Abstractions

Kwabena Boahen*
Bioengineering & Electrical Engineering
Stanford University

Chris Eliasmith

**Cofounder & Chief Scientific Adviser, Femtosense Inc.*

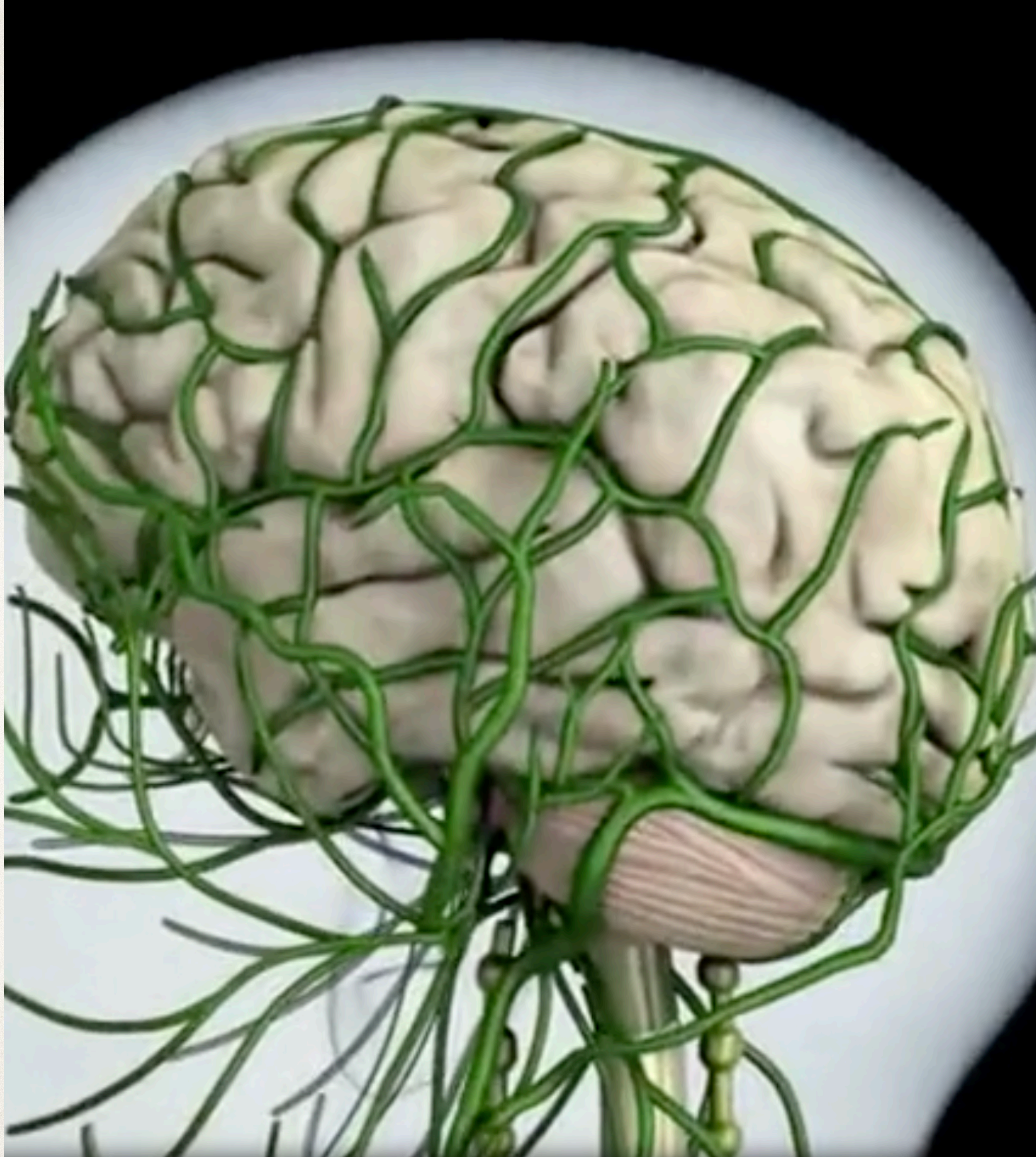
26 March 2019

Deep learning is huge —in the cloud

- Backprop learning is powerful
  - Networks deep in space or time
    - Space is discretized into layers
    - Time is discretized into steps
  - Unit's output must be differentiable (with respect to outputs of units feeding it)

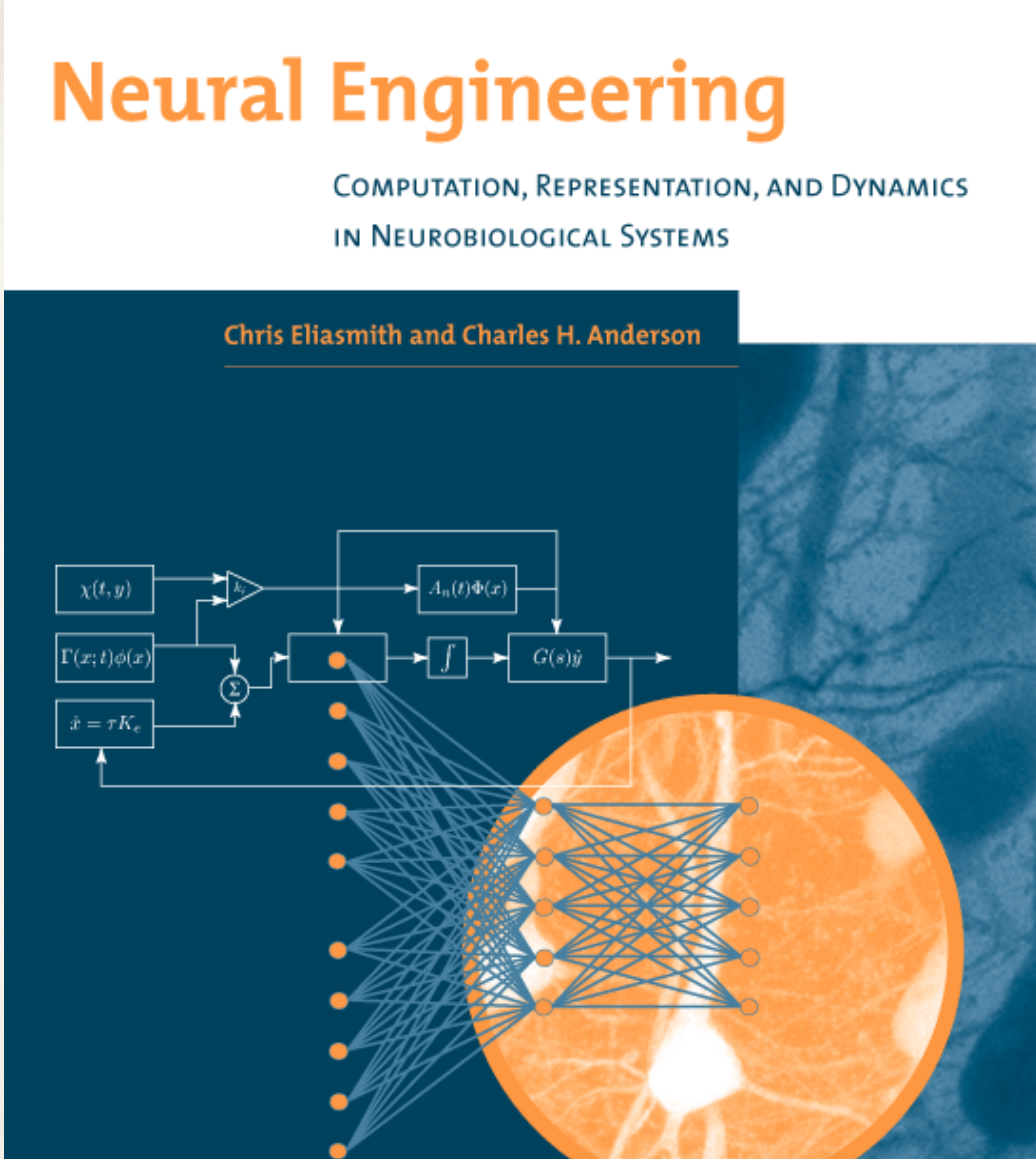
Backprop's constraints limit design-space

- Cannot take advantage of:
  - Physical <u>space</u> (its continuous)
  - Real <u>time</u> (its also continuous)
  - Non-differentiable signals (e.g., <u>spikes</u>)



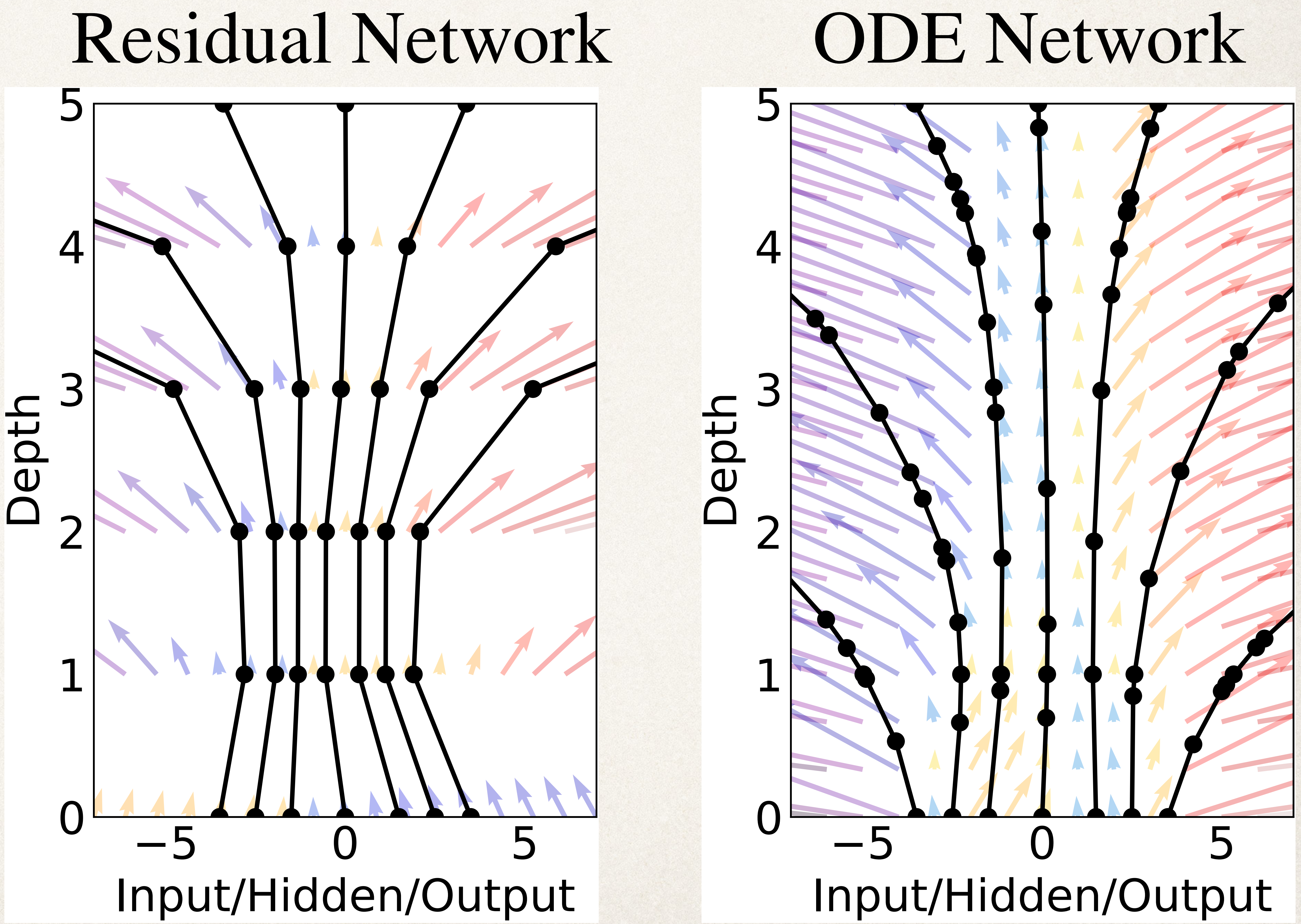
How do we relax its constraints? (Part I)

- Map functional abstractions onto physical ones
- Two existing examples:
  - Neural Engineering Framework (Eliasmith & Anderson 2003)
  - Predictive Coding Framework (Deneve et al. 2014)

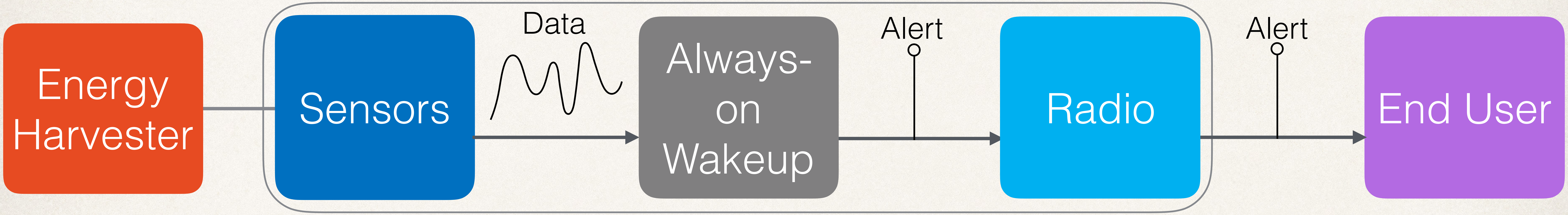


How do we relax its constraints? (Part II)

- Train networks continuous in time and space
  - Known as dynamical systems
- An existing example:
  - Neural Ordinary Differential Equations (Duvenaud et al. 2018)



What's the payoff? Learning at the edge



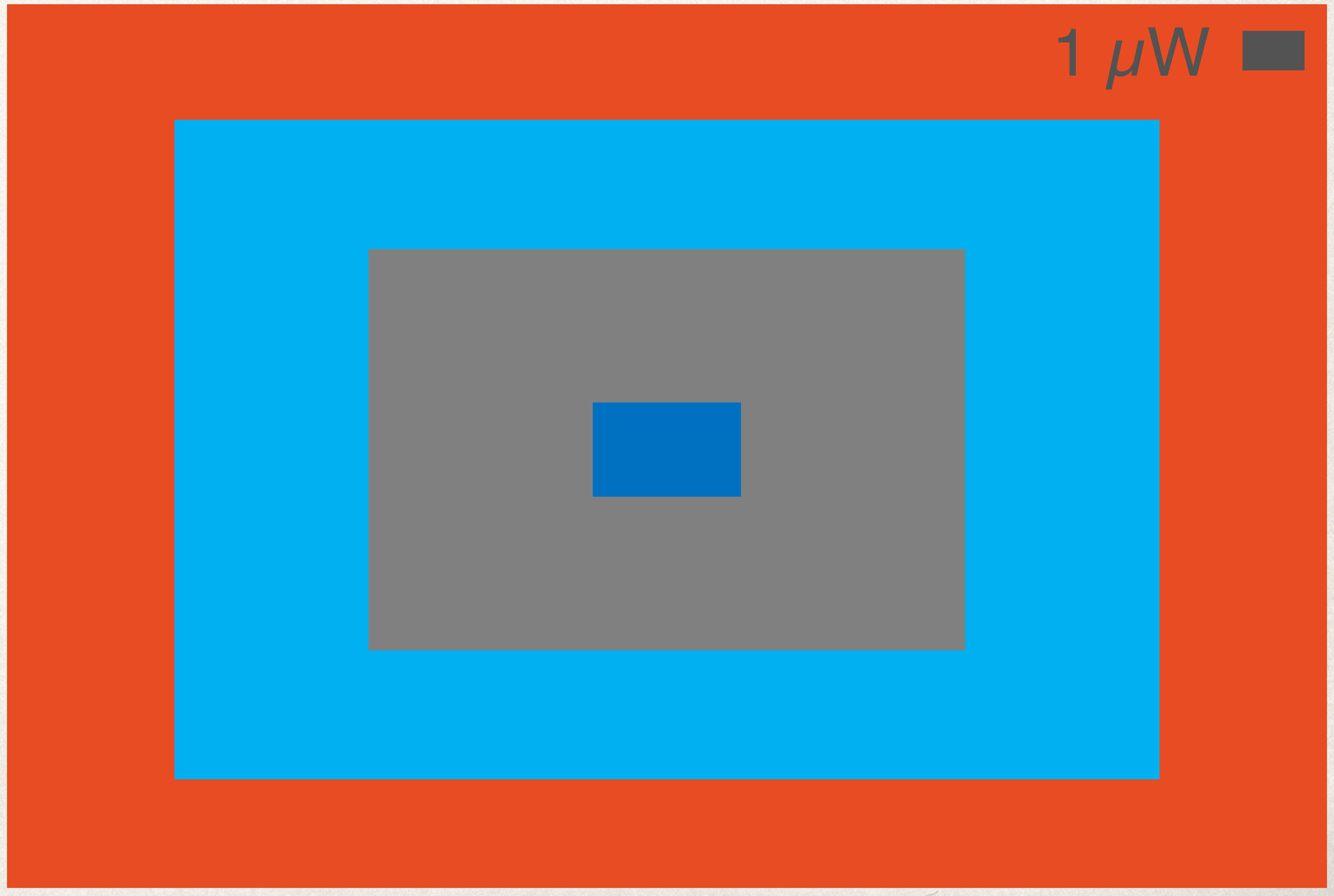
- Exploit <u>physical primitives</u> to implement physical abstractions
- Reap dramatic gains in <u>energy-efficiency</u>

Harvest
Vibration: 500 μW

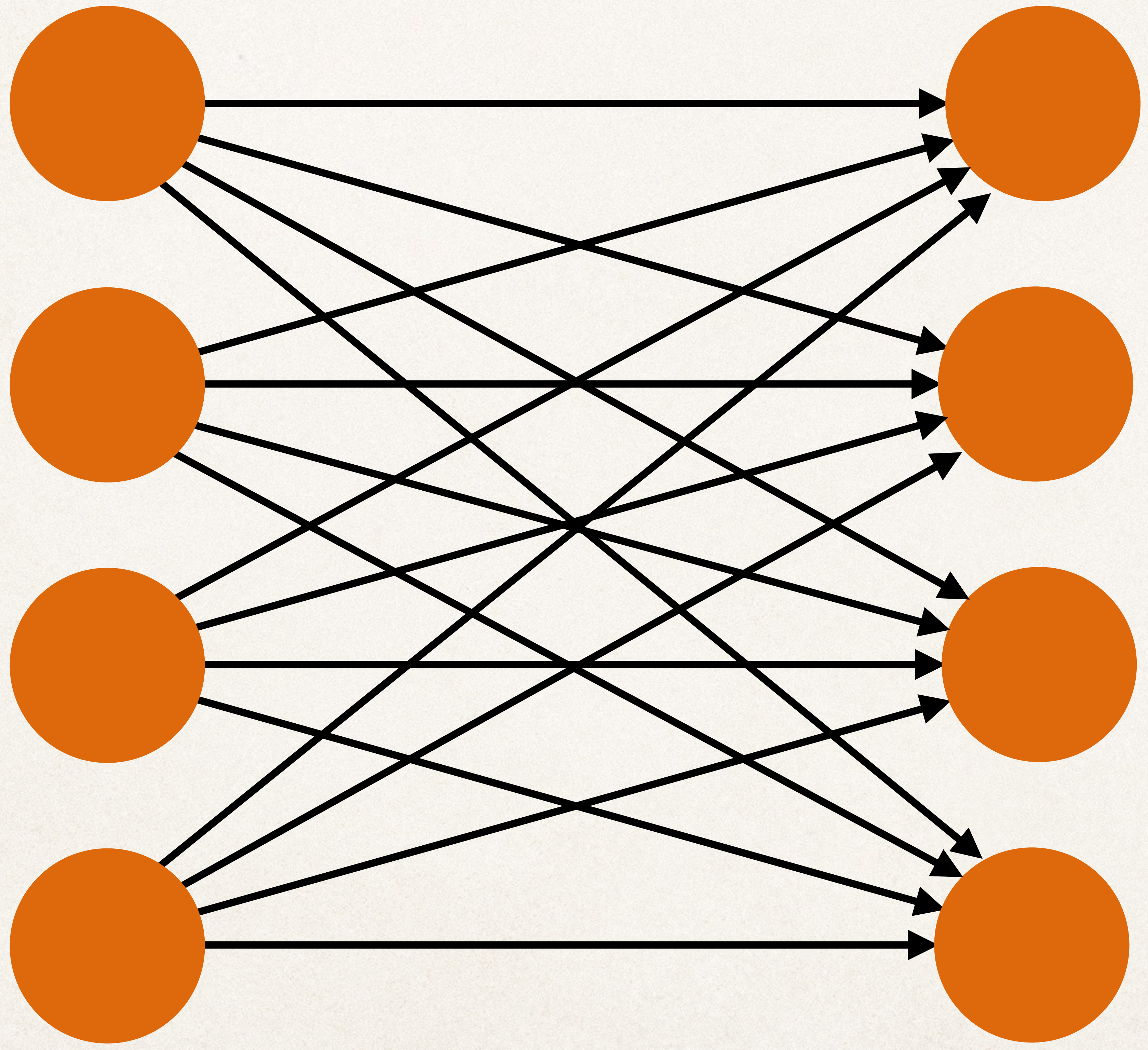
Sense
Accelerometer: 6 μW

Compute
Neuroprocessor: 100 μW

Communicate
BTLE (1% on-time): 280 μW

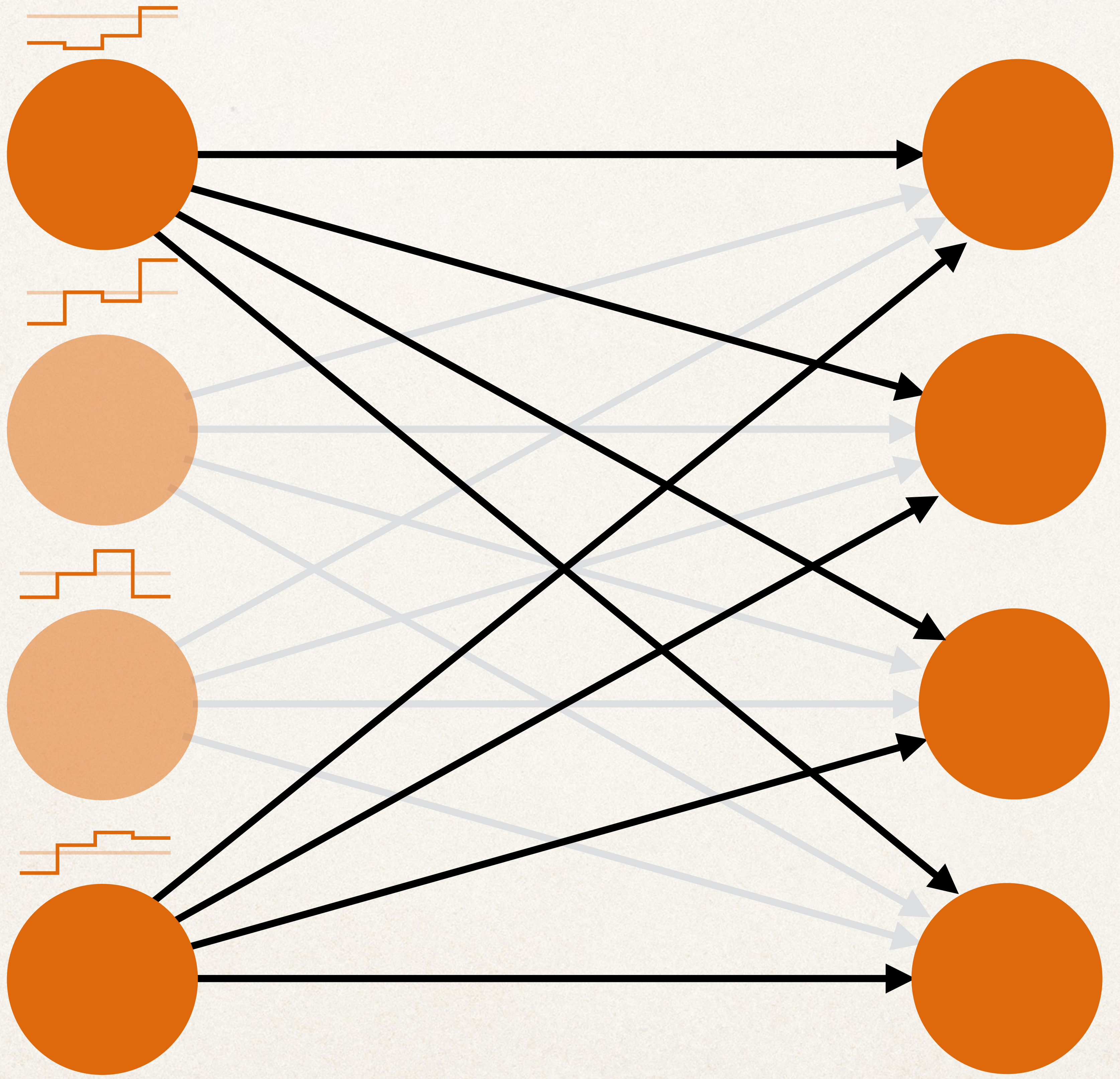


Minimizing energy



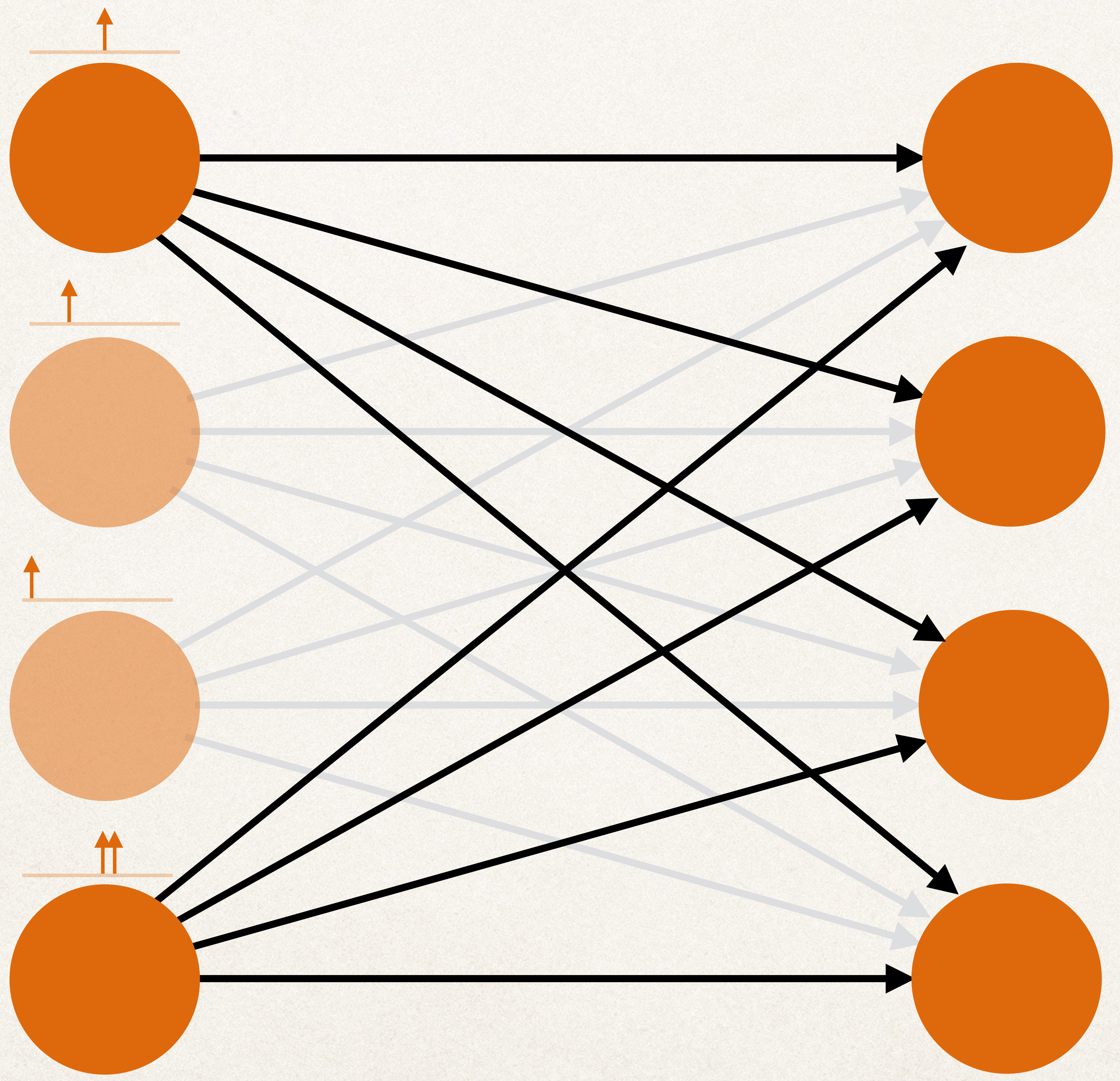
$$E_{op} = N_{active} N_{conn}$$
$$= (\rho_{active} N)(\rho_{conn} N)$$

Minimizing energy: Temporal sparsity



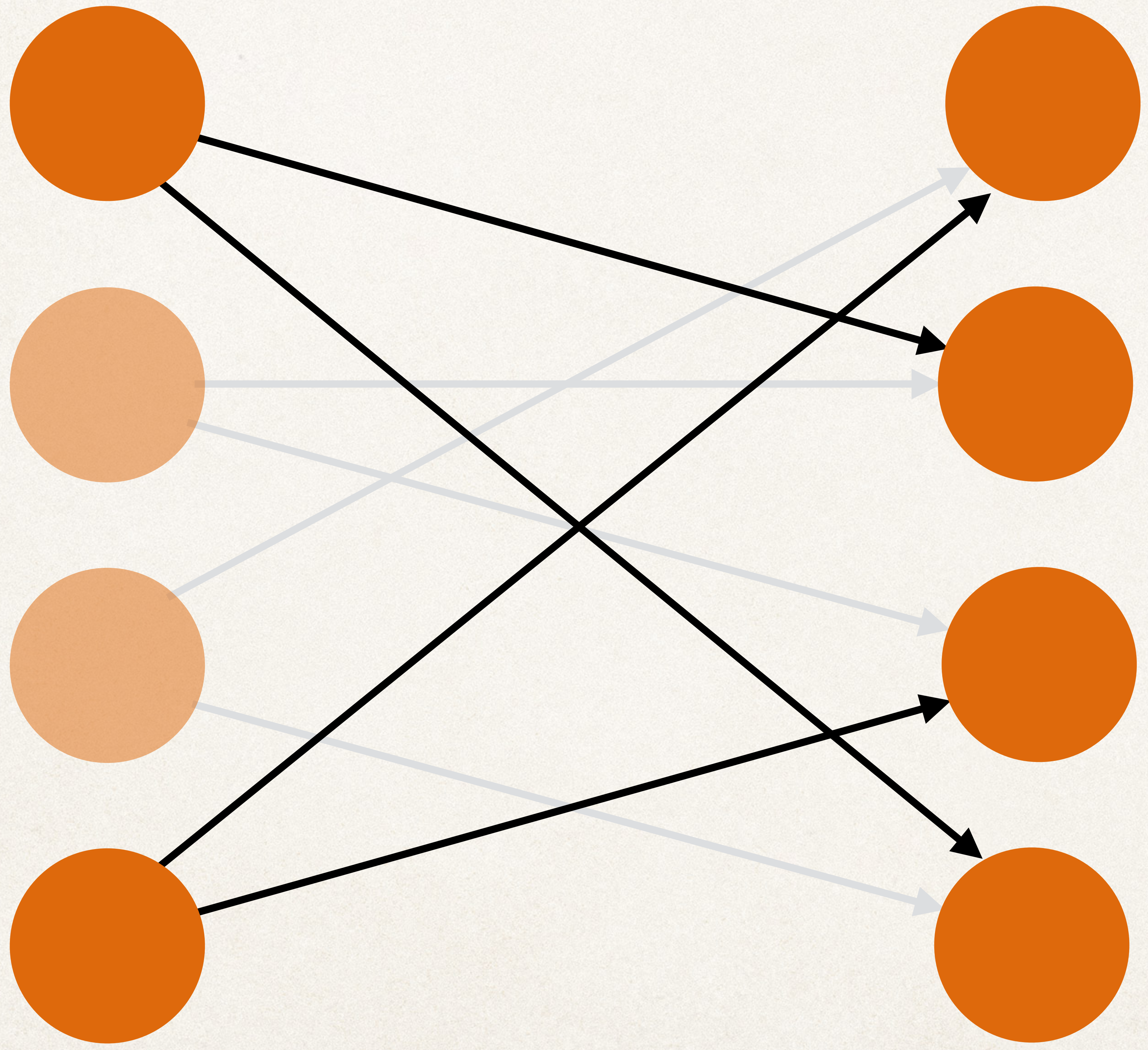
$$E_{op} = N_{active} N_{conn}$$
$$= (\rho_{active} N)(\rho_{conn} N)$$

Temporal sparsity: Spikes



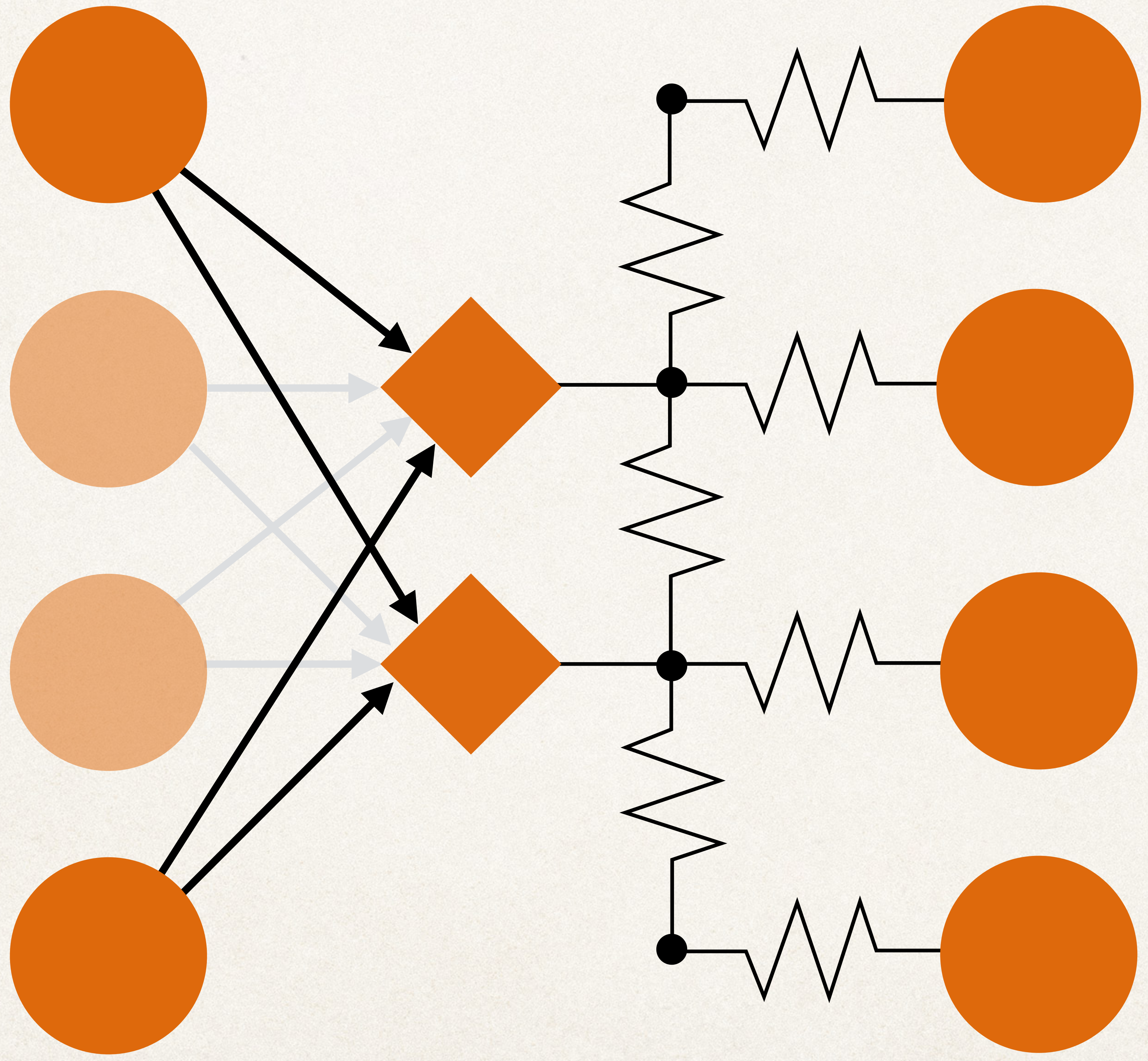
$$E_{op} = N_{active} N_{conn}$$
$$= (\rho_{active} N)(\rho_{conn} N)$$

Minimizing energy: Spatial sparsity

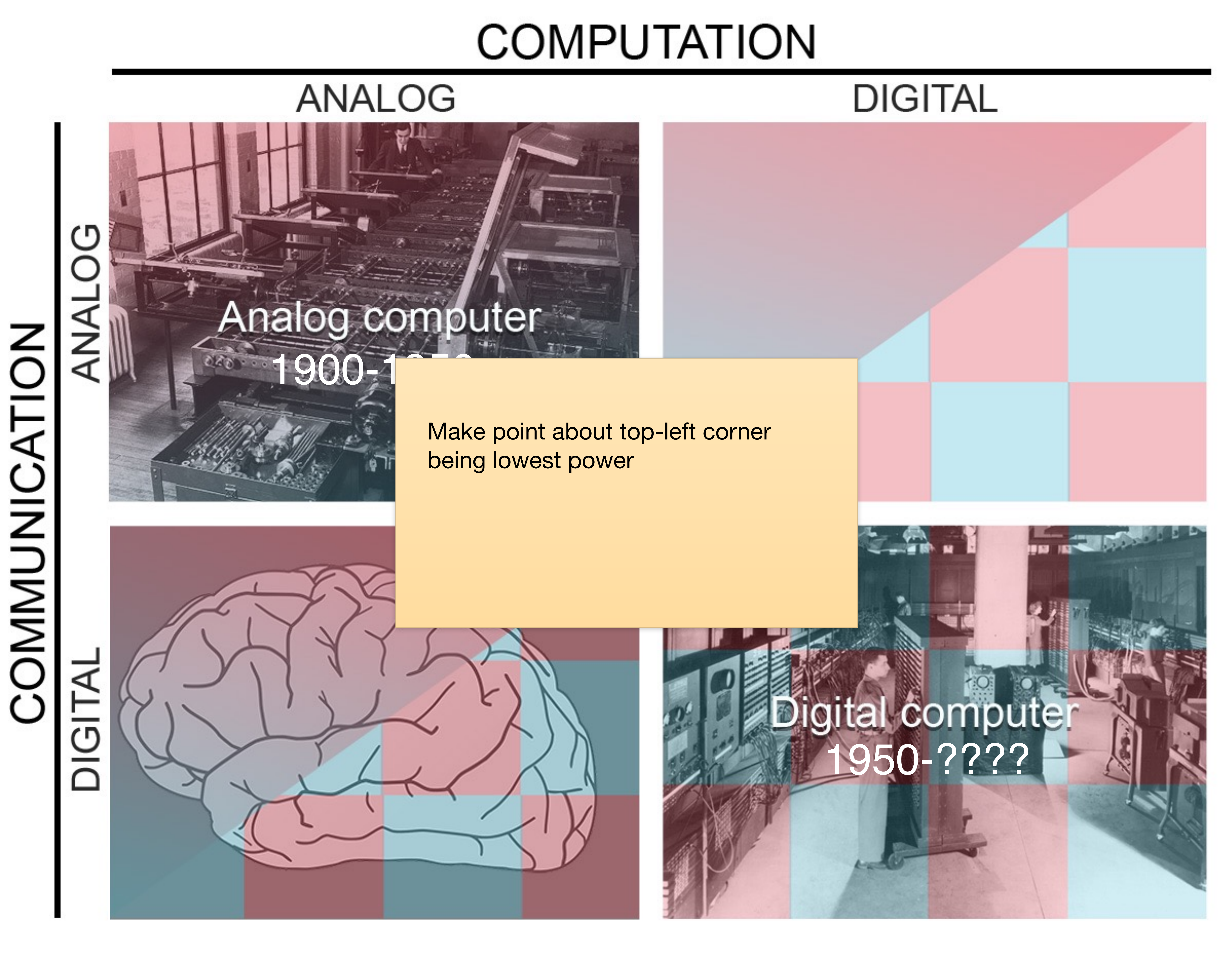


$$E_{op} = N_{active} N_{conn}$$
$$= (\rho_{active} N)(\rho_{conn} N)$$

Spatial sparsity: Analog convolving

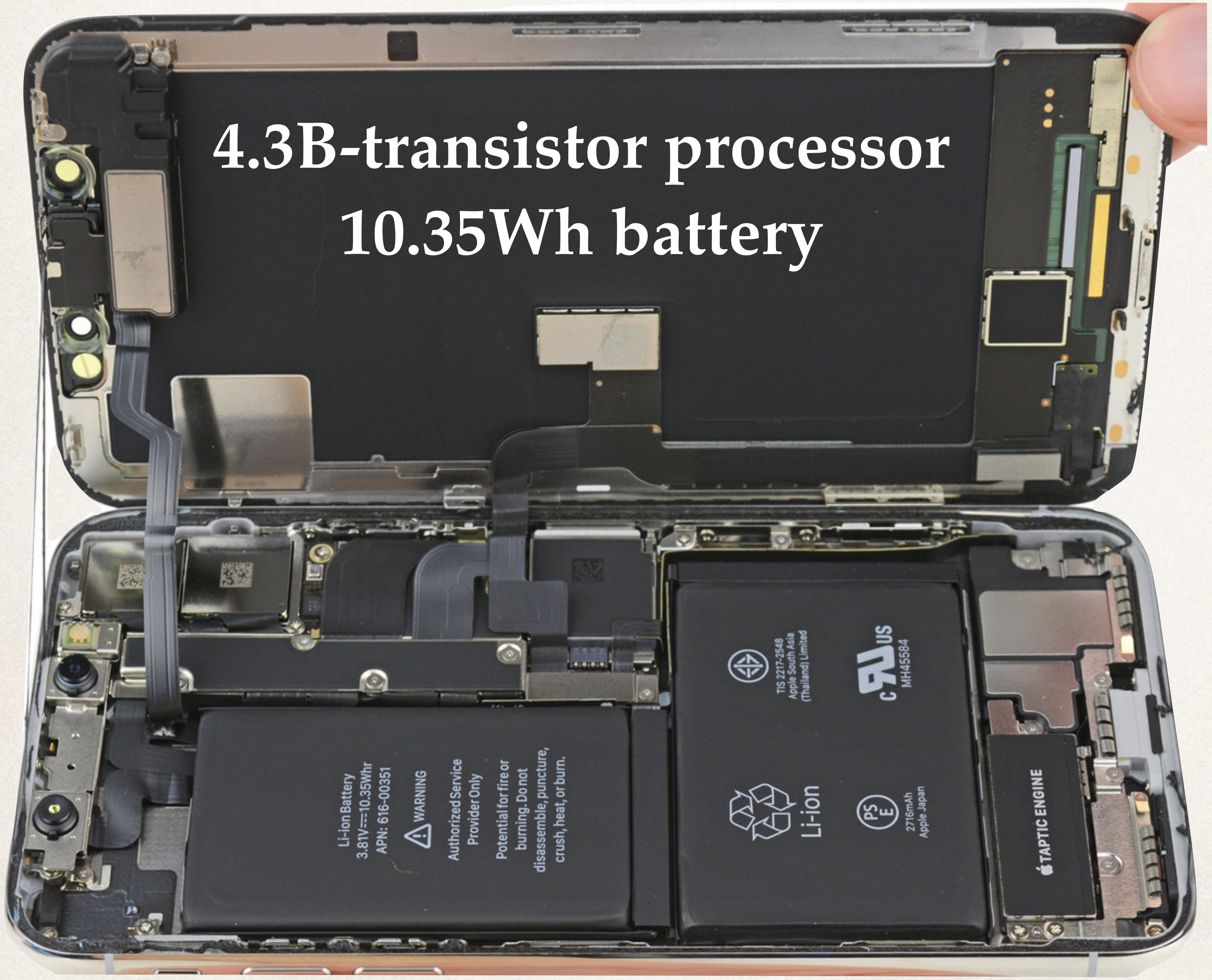


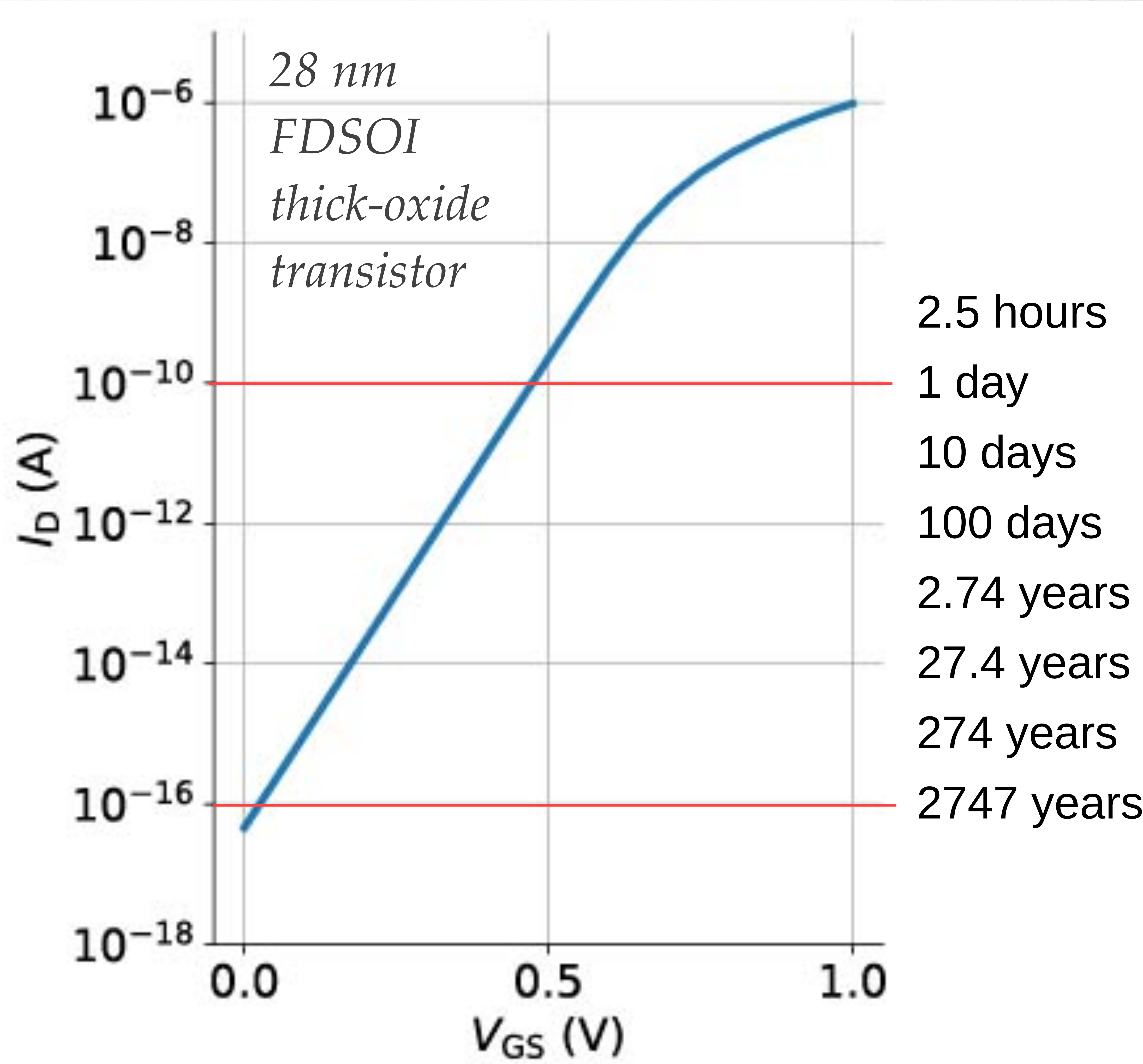
$$E_{op} = N_{active} N_{conn}$$
$$= (\rho_{active} N)(\rho_{conn} N)$$



Make point about top-left corner being lowest power

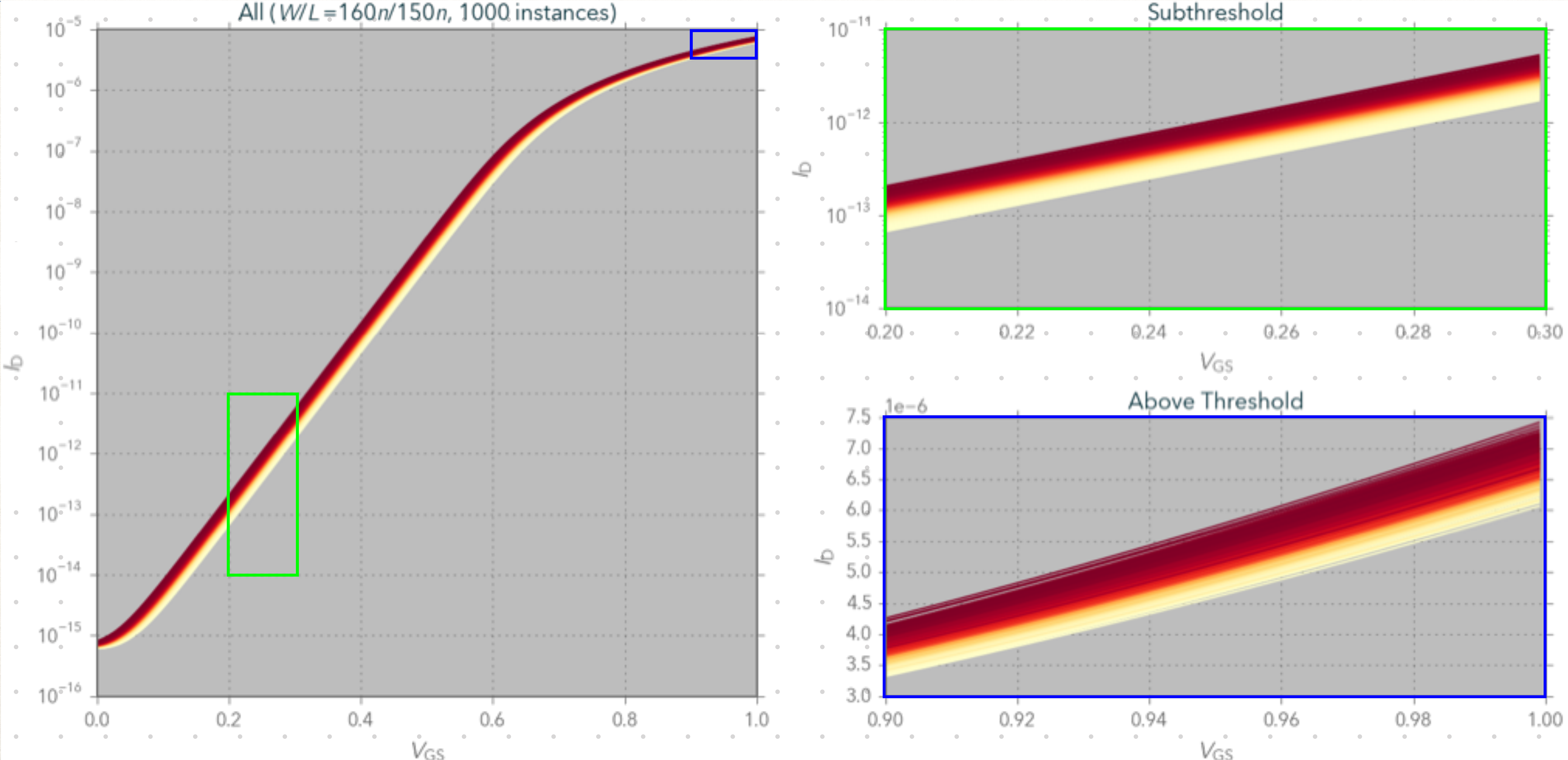
Digital versus Analog: 1 day versus 1000 yrs





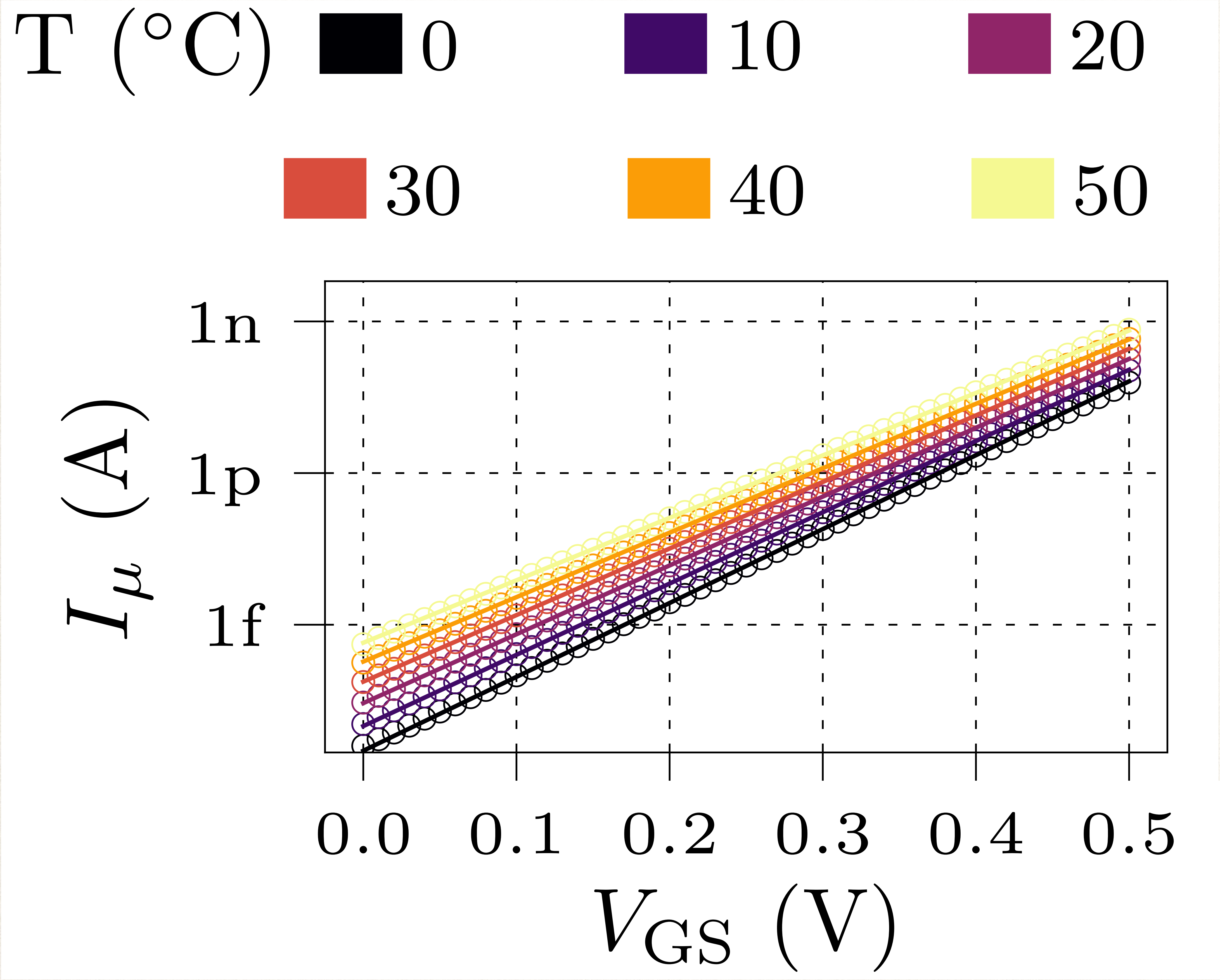
2013 NIPS Demo

Analog Challenge I: Heterogeneity



Silicon neurons' tuning-curves (Braindrop)

Analog Challenge I: Thermal Sensitivity



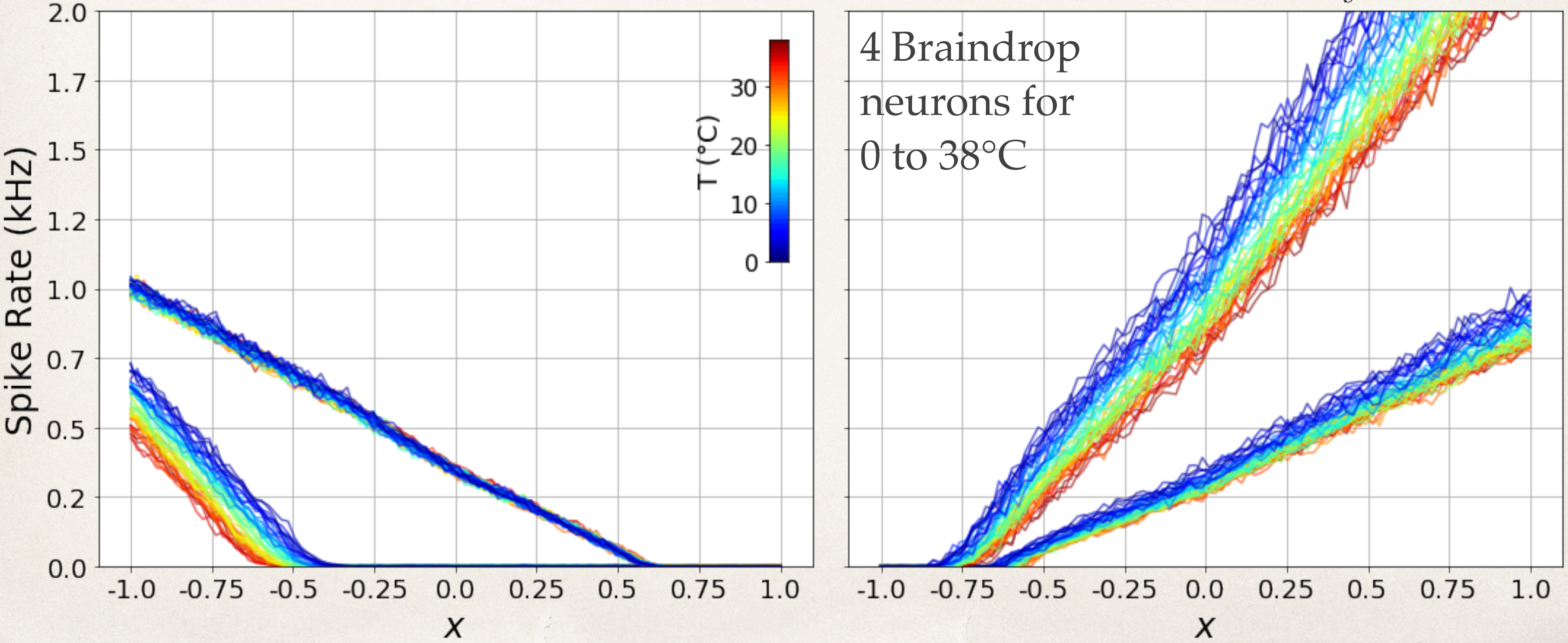
$$I_\mu(T) = I_{0_{\mathrm{nom}}} e^{\langle\gamma_1\rangle\left(1-\frac{T_{\mathrm{nom}}}{T}\right)} e^{\frac{(1-\kappa)V_{\mathrm{BS}}}{U_T}} \times e^{\frac{\kappa V_{\mathrm{GS}}}{U_T}} e^{\left(\lambda_1 \frac{T_{\mathrm{nom}}}{T}+\lambda_2\right)\Delta V_{\mathrm{DS}}} \left(1 - e^{-\frac{V_{\mathrm{DS}}}{U_T}}\right)$$

- A subthreshold transistor's current (I_μ) is exponentially sensitive to temperature
  - T is the absolute temperature
  - $U_T = kT/q$ is the thermal voltage
- Across a 50°C range, the current changes by 1.5 to 3 decades

Benjamin & Boahen 2019

Tuning-curves' thermal sensitivity (Braindrop)

Reid, Montoya, & Boahen 2019



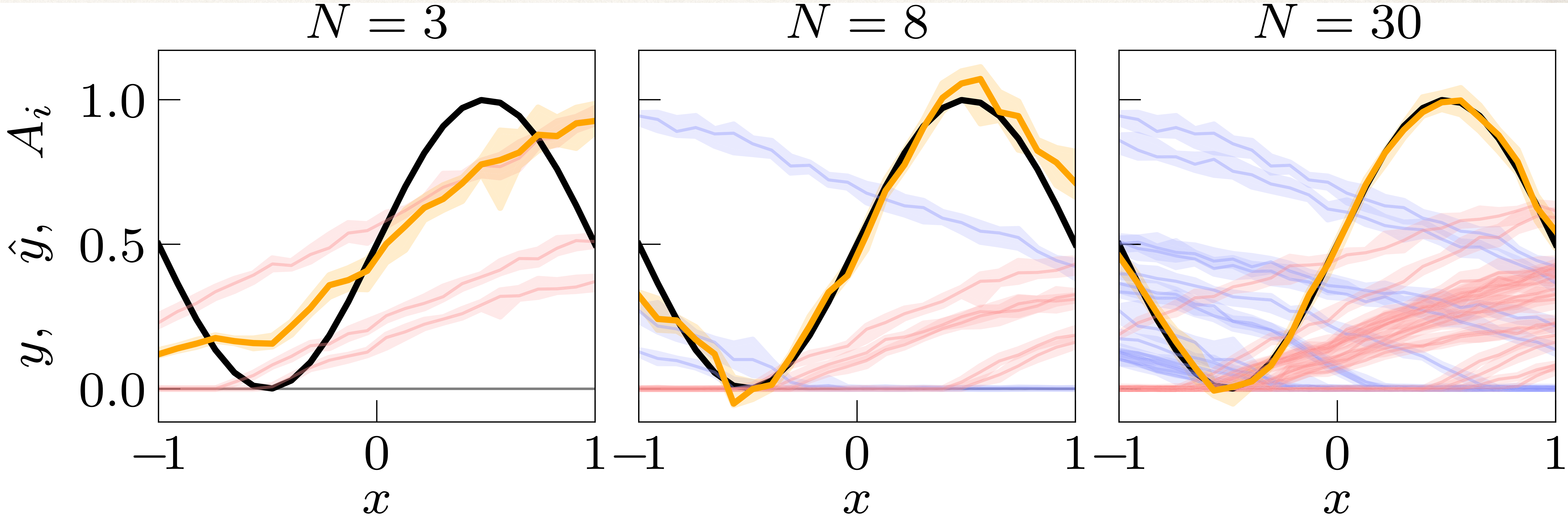
Approximating functions

- The desired function $f(x)$ is expressed as a weighted sum of the neural *tuning curves* $a_i(x)$
- The weights—called decoders—are labeled d_i

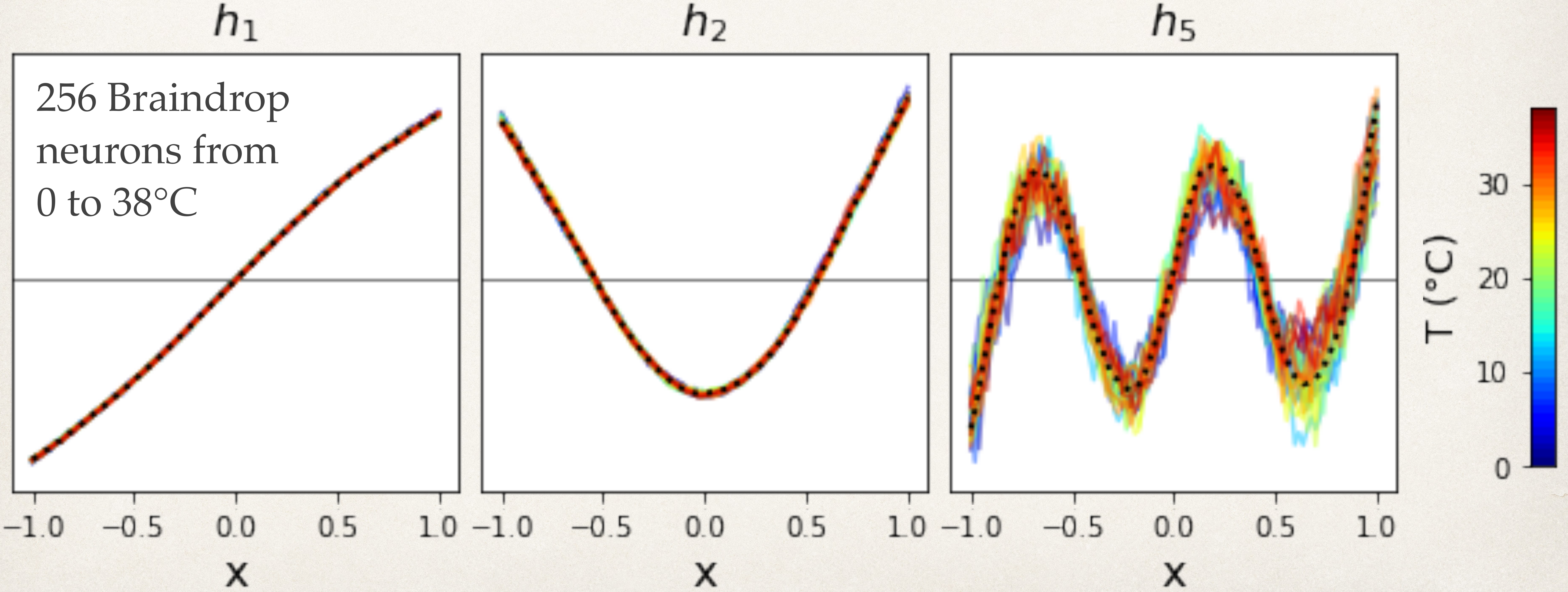
$$\mathbf{f} = \mathbf{A}\mathbf{d} \Rightarrow \mathbf{d} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{f}$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_Q) \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} a_1(x_1) \\ a_1(x_2) \\ \vdots \\ a_1(x_Q) \end{matrix}} & \boxed{\begin{matrix} a_2(x_1) \\ a_2(x_2) \\ \vdots \\ a_2(x_Q) \end{matrix}} & \begin{matrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} & \boxed{\begin{matrix} a_N(x_1) \\ a_N(x_2) \\ \vdots \\ a_N(x_Q) \end{matrix}} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}$$

3 to 30 Braindrop neurons at 26°C



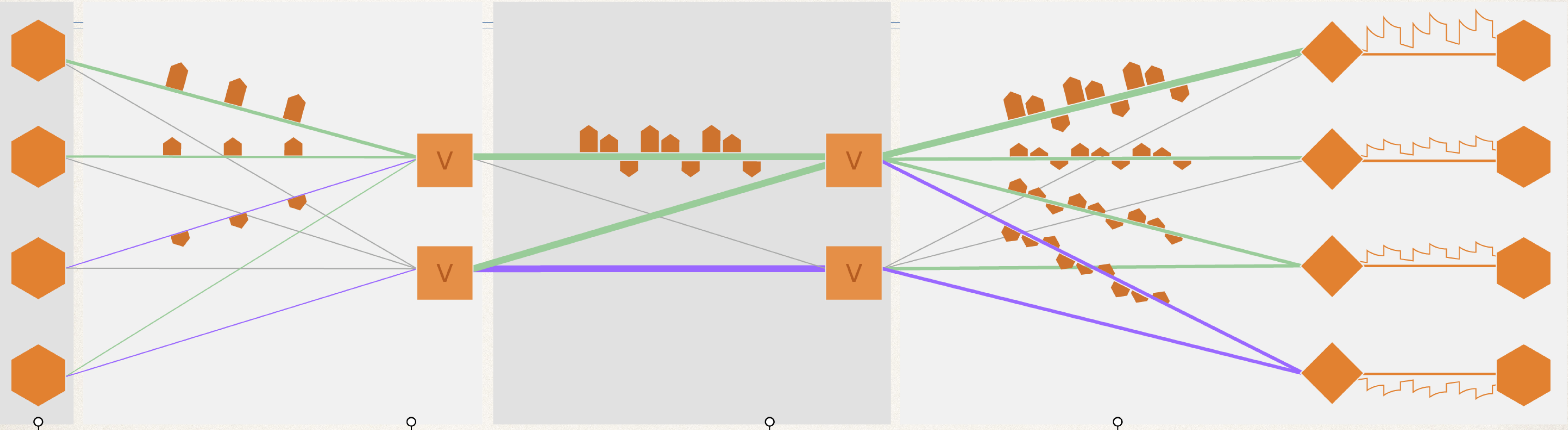
Thermally robust computation (Braindrop)



Reid, Montoya, & Boahen 2019

NEF: Decode-Transform-Encode

Eliasmith & Anderson 2003



❶ Spike

$\langle \delta_{x_i} \rangle_t = \lceil 0, \alpha_i(I_i + \beta_i) \rceil$

Somas emit unit-area deltas δ_{x_i} at rates $\langle \delta_{x_i} \rangle_t$ dictated by their input current I_i.

❷ Decode

$\delta_{y_j} = \sum_i D_{ji} \delta_{x_i}, D \in \mathbb{R}^{D \times N}$

Deltas are then scaled by their decode weight and merged together.

❸ Transform

$\delta_{z_k} = \sum_j T_{kj} \delta_y, T \in \mathbb{R}^{D \times D}$

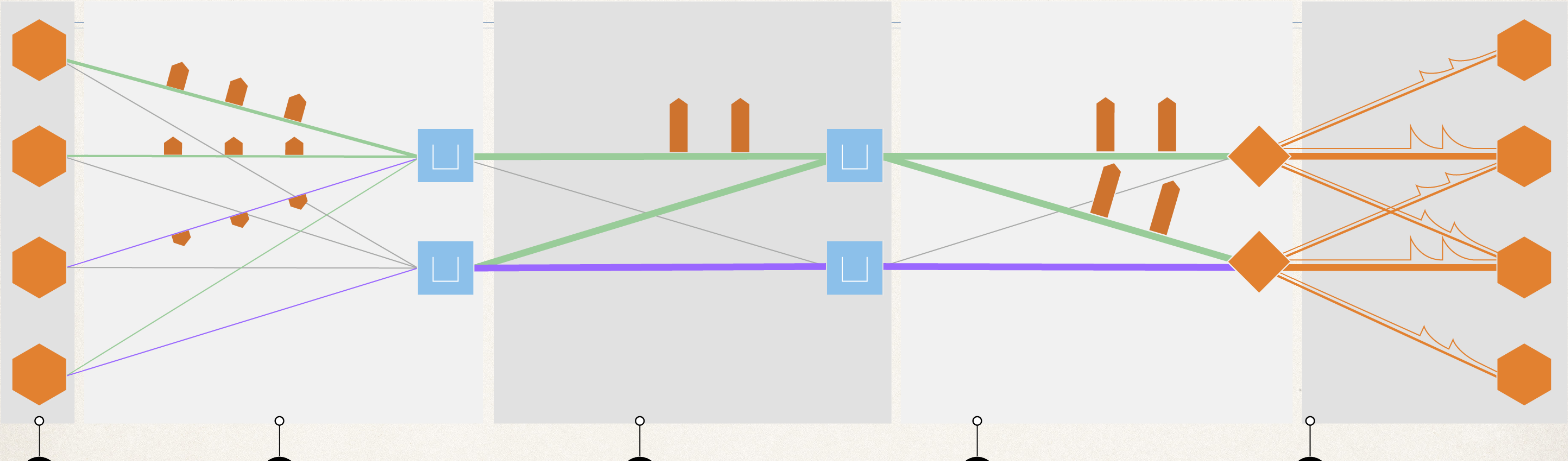
Transform works the same way as Decode.

❹ Encode

$\tau \dot{I}_l = -I_l + \sum_k E_{lk} \delta_{z_k}, E \in \mathbb{R}^{N \times D}$

Synaptic filters superpose and low-pass filter weighted deltas to produce output currents (I_l) that feed the next soma layer.

Digital thinning and analog convolving



❶ Spike

$\langle \delta_{x_i} \rangle_t =$
$\lceil 0, \alpha_i(I_i + \beta_i) \rceil$

Somas emit delta trains (as in Figure 2).

❷ Decode

$\langle \delta_{y_j} \rangle_t = \sum_i D_{ji} \langle \delta_{x_i} \rangle_t$
$D \in [-1, 1]^{D \times N}$

Weighted deltas are accumulated to produce a stream of unit-area deltas.

❸ Transform

$\langle \delta_{z_k} \rangle_t = \sum_j T_{kj} \langle \delta_{y_j} \rangle_t$
$T \in [-1, 1]^{D \times D}$

Transform still works the same as Decode ($T_{kj} = 1$ in this example).

❹ Sparse Encode

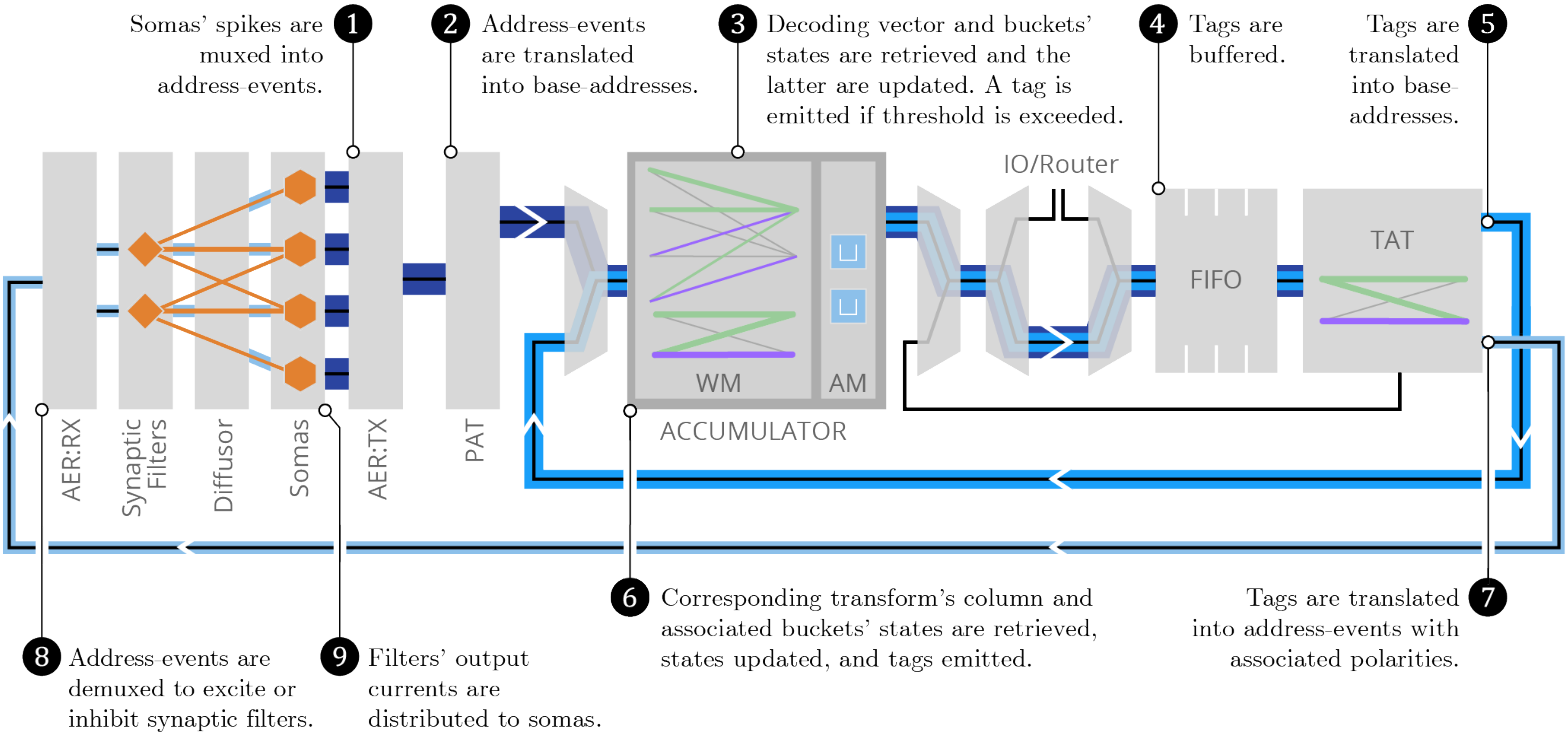
$\tau I_l = -I_l + \sum_k S_{lk} \delta_{z_k}$
$S \in \{-1, 0, 1\}^{N \times D}$

Each accumulator's deltas are sent to a subset of synaptic filters (tap-points).

❺ Convolve

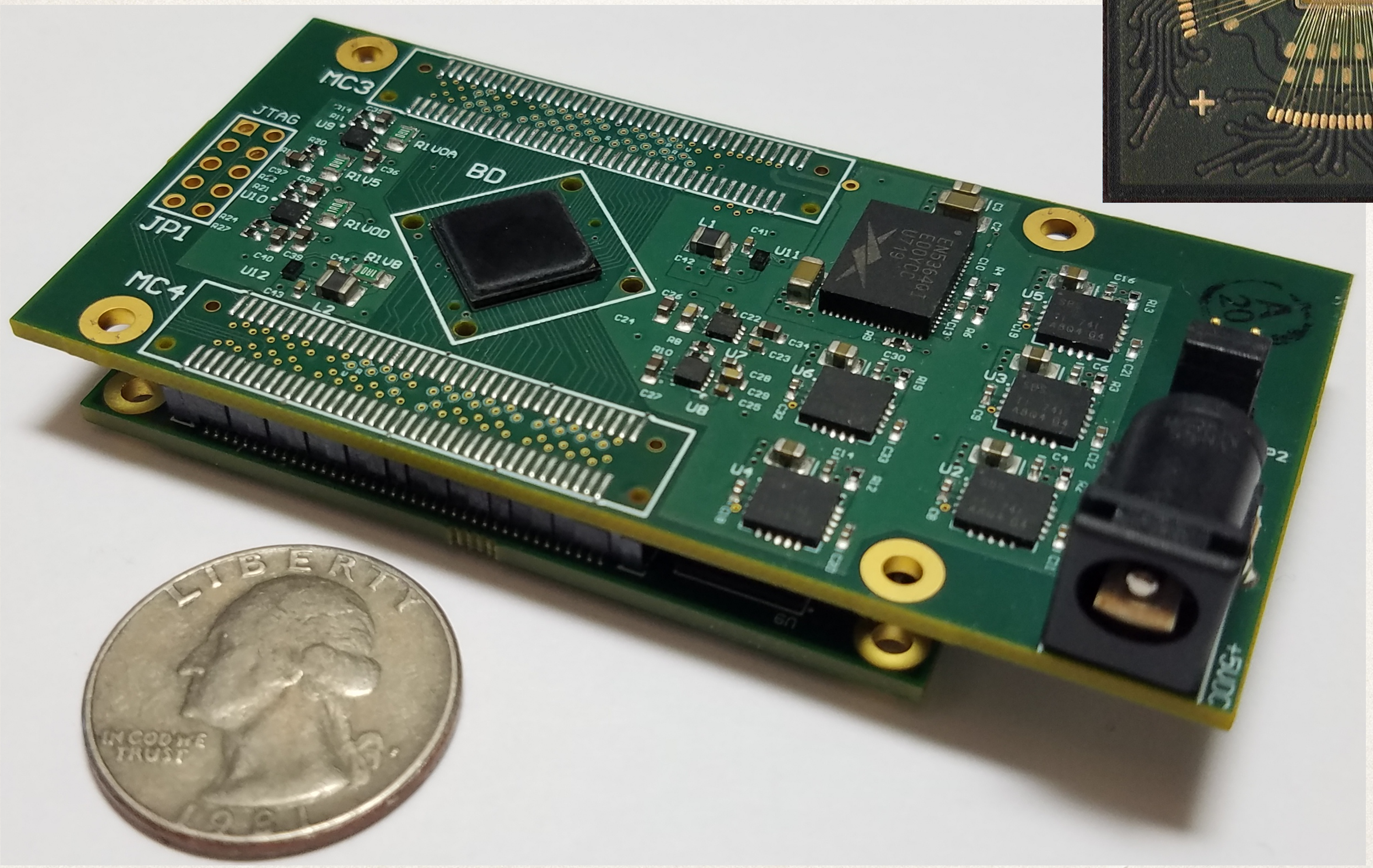
$I_m = \sum_l \hat{d}_\gamma(m - l) I_l$

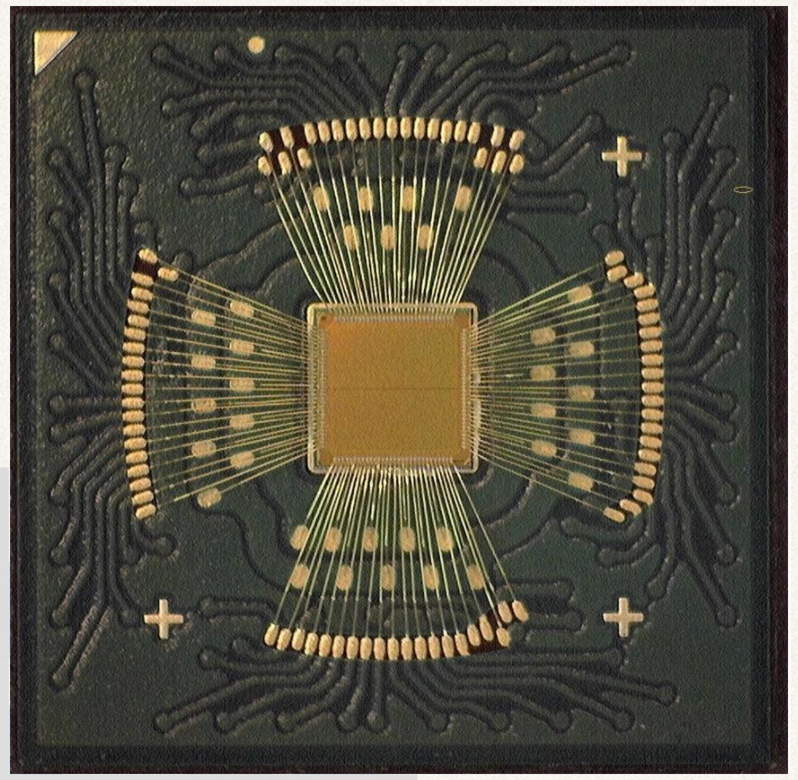
Filter outputs (I_l) are convolved (I_m) and sent to the next soma layer.

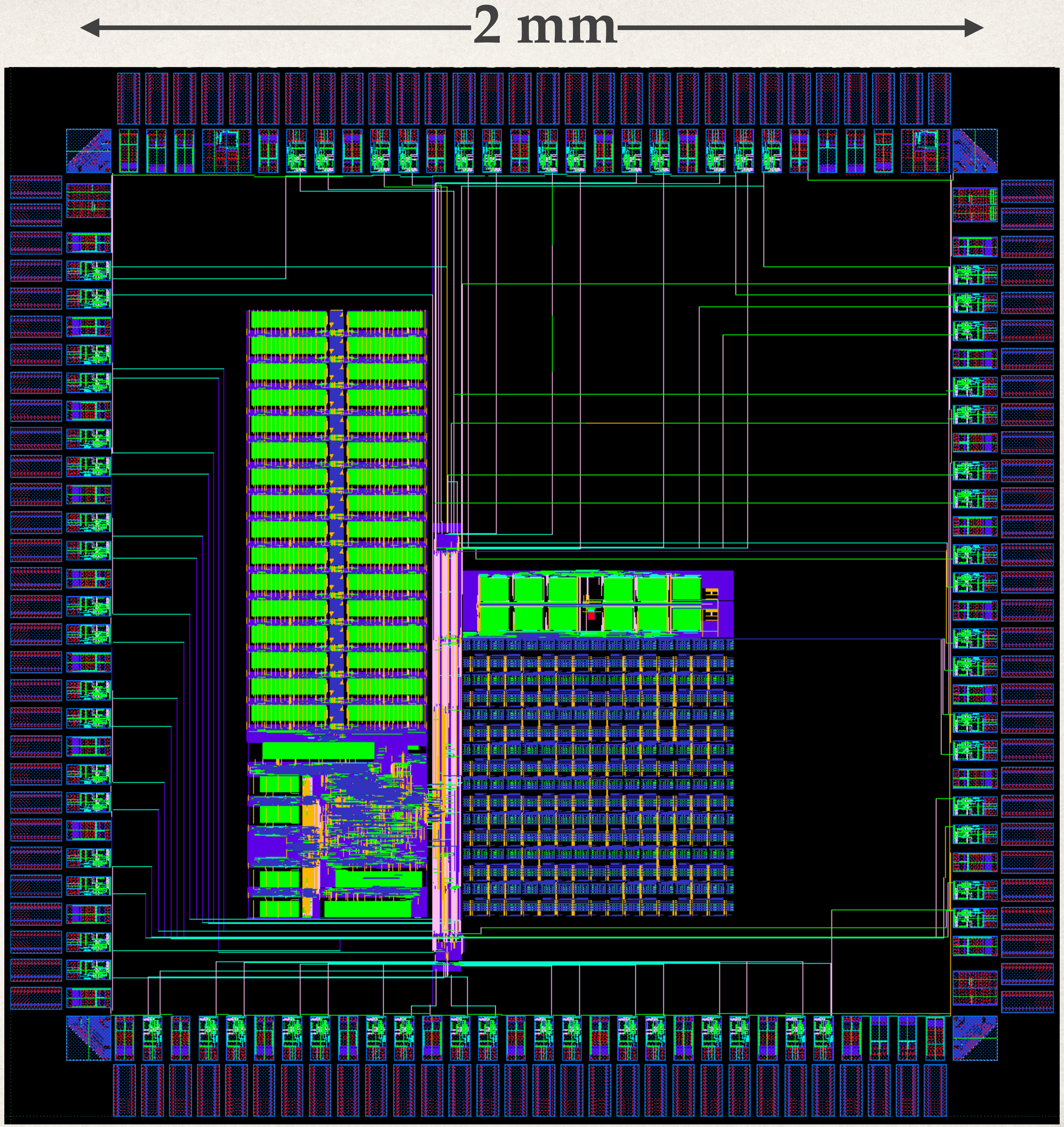


Somas' spikes are muxed into address-events.
1
2
Address-events are translated into base-addresses.
3
Decoding vector and buckets' states are retrieved and the latter are updated. A tag is emitted if threshold is exceeded.
4
Tags are buffered.
Tags are translated into base-addresses.
5
IO/Router
WM
AM
ACCUMULATOR
FIFO
TAT
AER:RX
Synaptic Filters
Diffusor
Somas
AER:TX
PAT
6
Corresponding transform's column and associated buckets' states are retrieved, states updated, and tags emitted.
Tags are translated into address-events with associated polarities.
7
8
Address-events are demuxed to excite or inhibit synaptic filters.
9
Filters' output currents are distributed to somas.

Braindrop: 4096 neurons in 28nm FDSOI CMOS

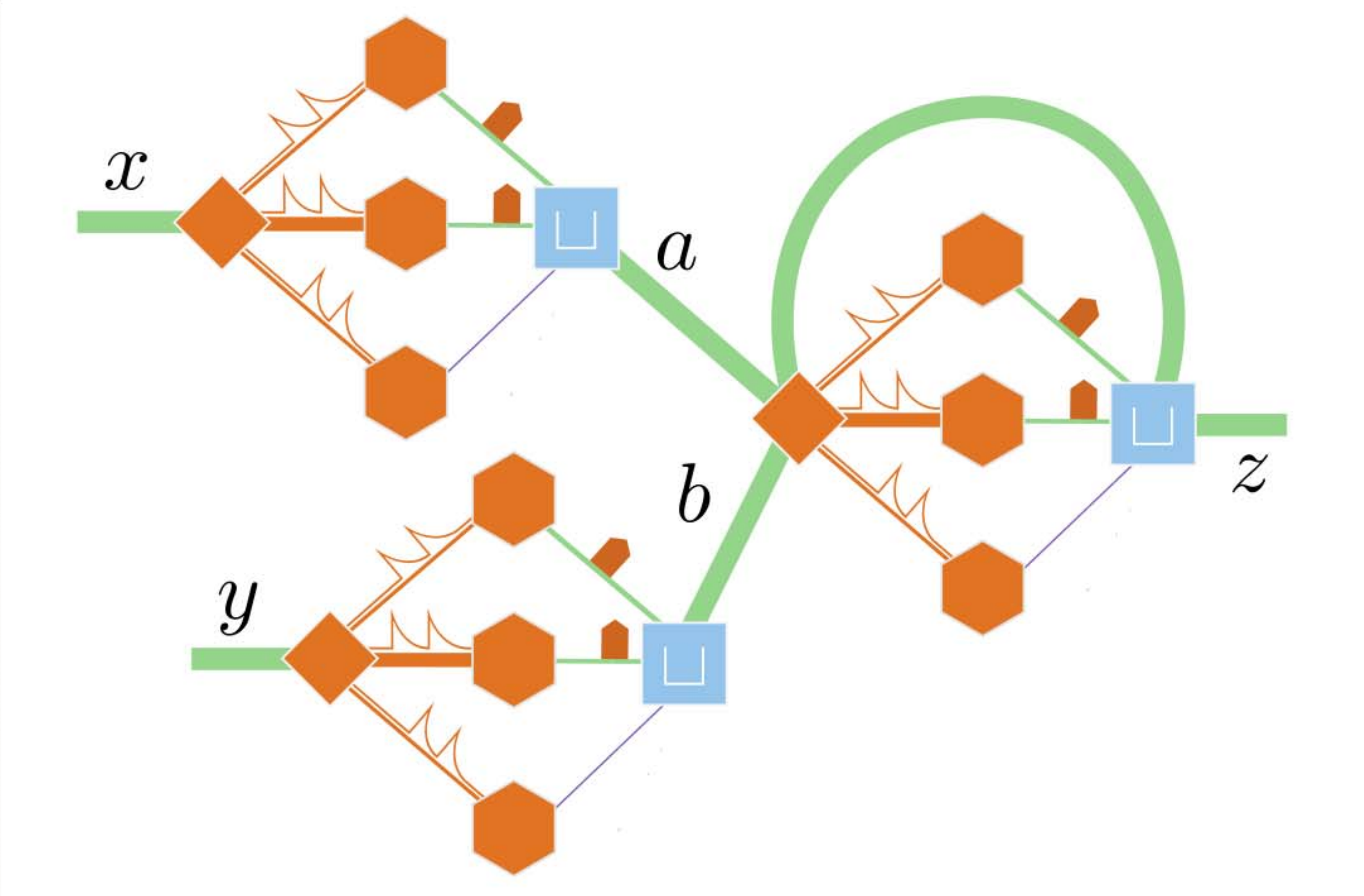


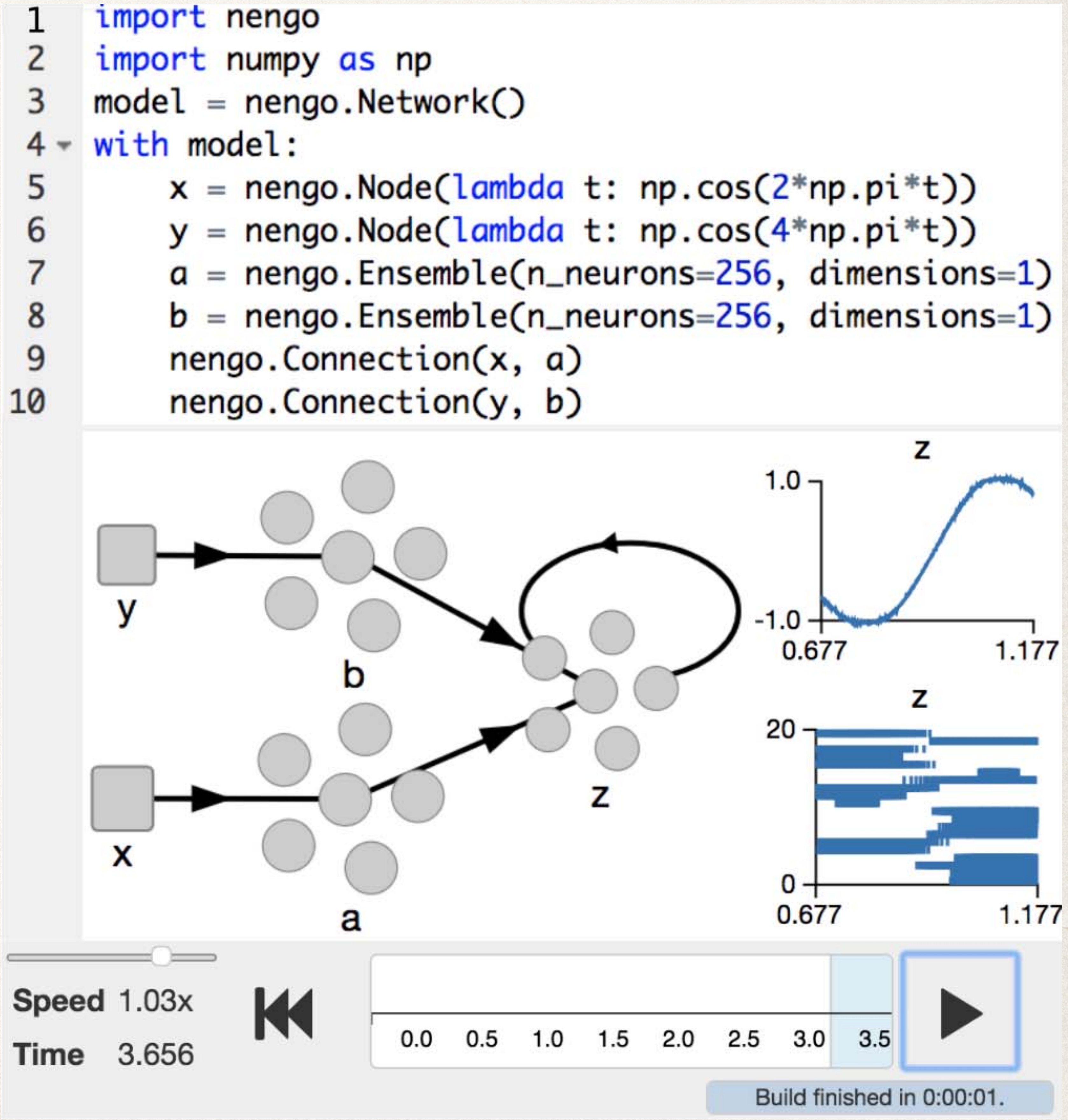


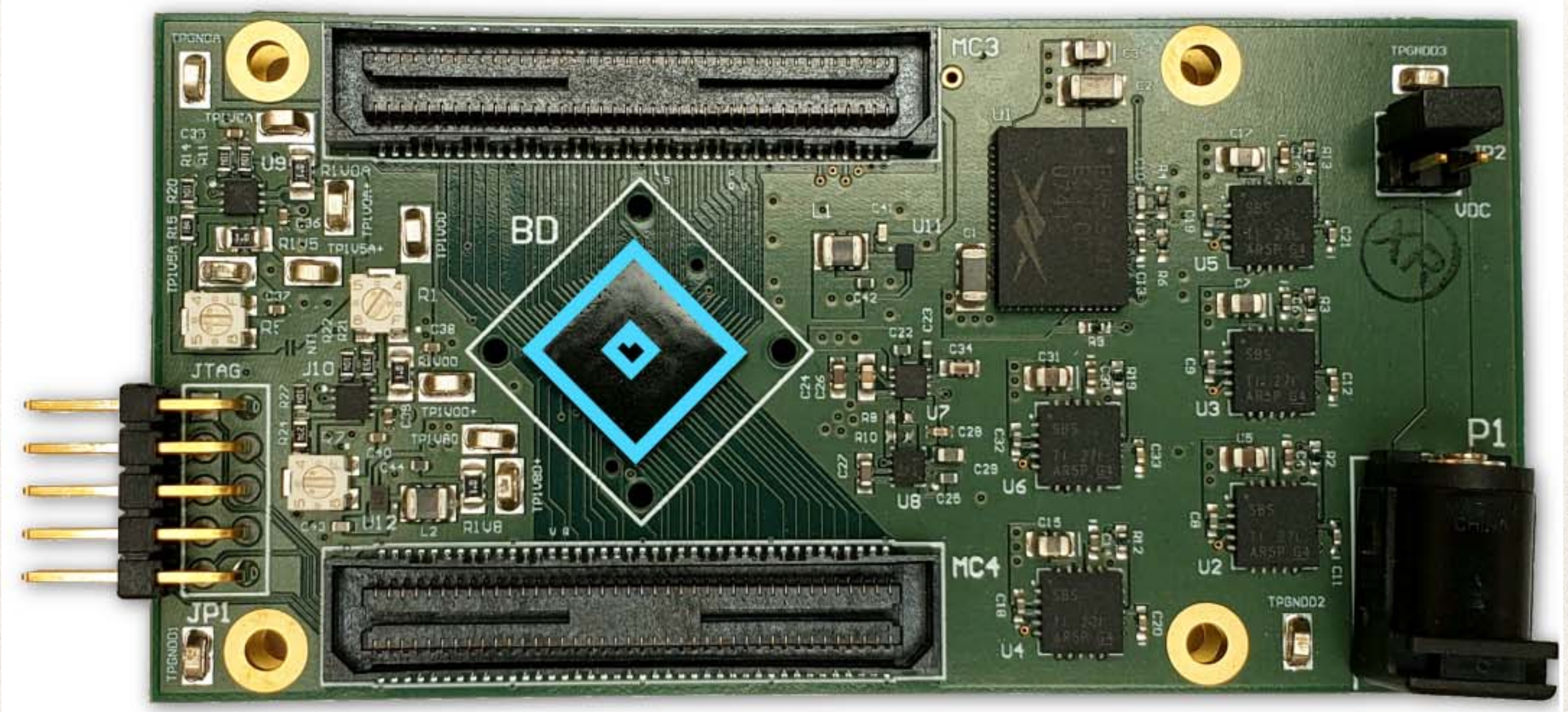


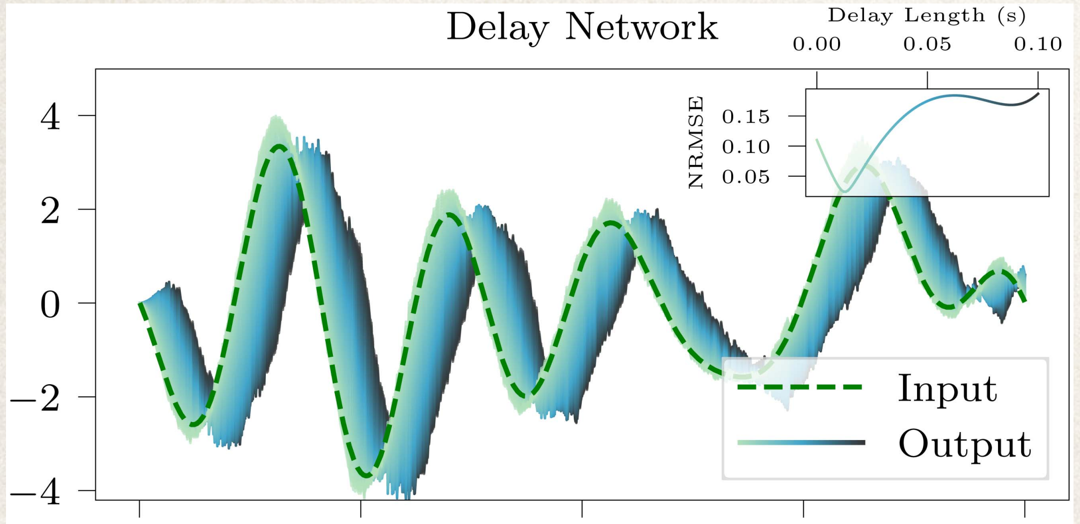
Programming Environment (Nengo)

$$
\begin{aligned}
a &= f(x) \\
b &= g(y) \\
\dot{z} &= h(z) + a + b
\end{aligned}
$$



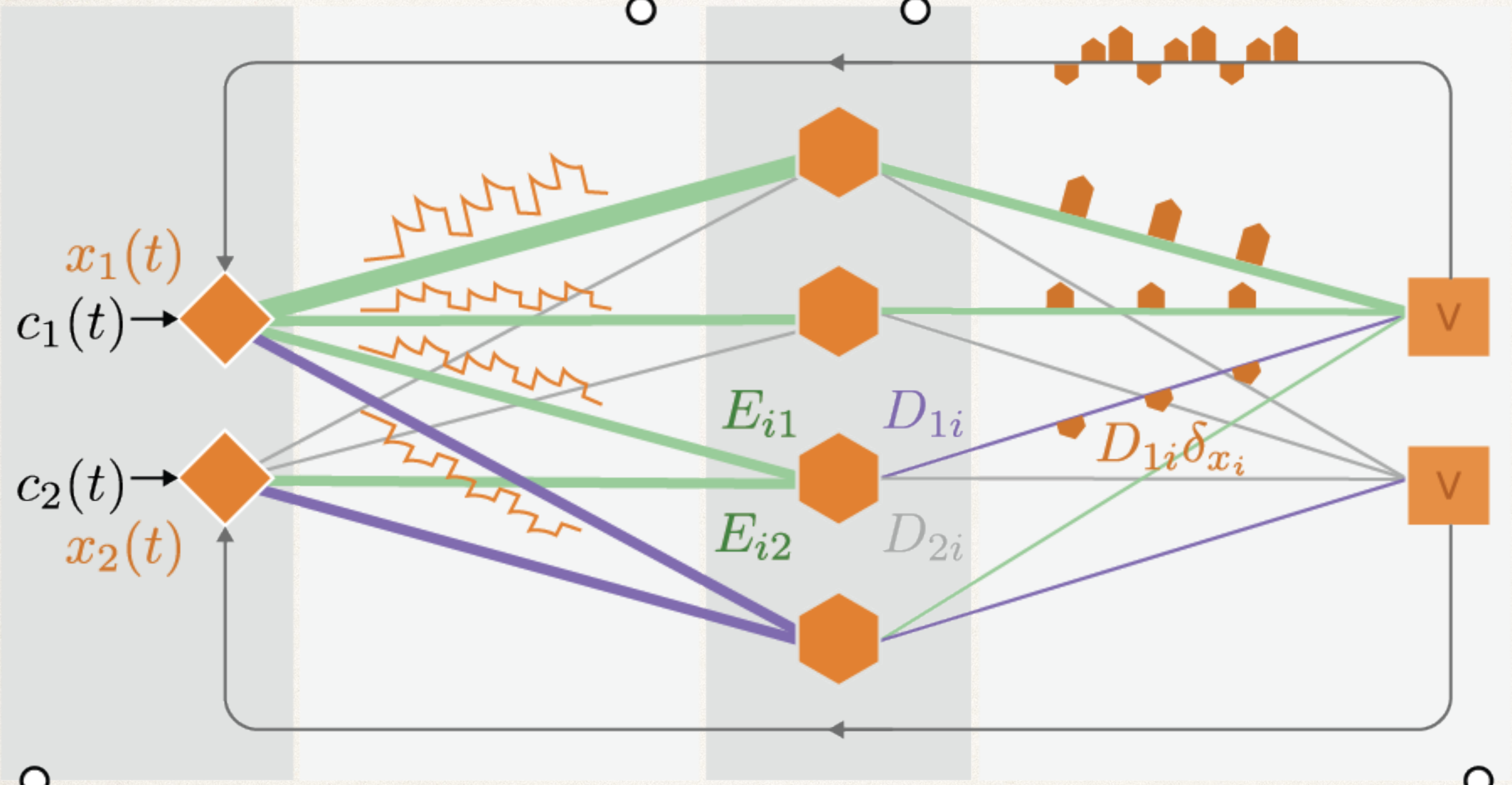






Encode 1
Synaptic filters' states, $x(t)$, are projected onto encoding vectors, E_i, to drive somas.

2 **Somas**
The i^{th} soma's input current, $J_i = E_i x(t)$, drives it to spike at rate $\langle \delta_{x_i} \rangle_t$.



Decode 3
Deltas with area equal to components of the neuron's decoding vector, D_i^{T}, replace spikes.

4 **Synaptic Filters**
Merged delta trains are lowpass filtered.
As a result, $\tau_{\text{syn}}\dot{x}(t) + x(t) = \sum_i D_i^{\mathrm{T}} \delta_{x_i} + c(t)$, where $c(t)$ is a vector of injected currents.

- To emulate the dynamical system

$$\tau_{\text{dyn}}\dot{x}(t) = f(x) + u(t)$$

- Choose decoding weights such that, after synaptic filtering,

$$\textstyle\sum_i D_i^{\mathrm{T}} \delta_{x_i} \approx \tau_{\text{syn}}/\tau_{\text{dyn}} f(x) + x$$

- And set

$$c(t) = \tau_{\text{syn}}/\tau_{\text{dyn}} u(t)$$

Eliasmith & Anderson 2003

Tapped delay-line (Braindrop)

384 Braindrop neurons

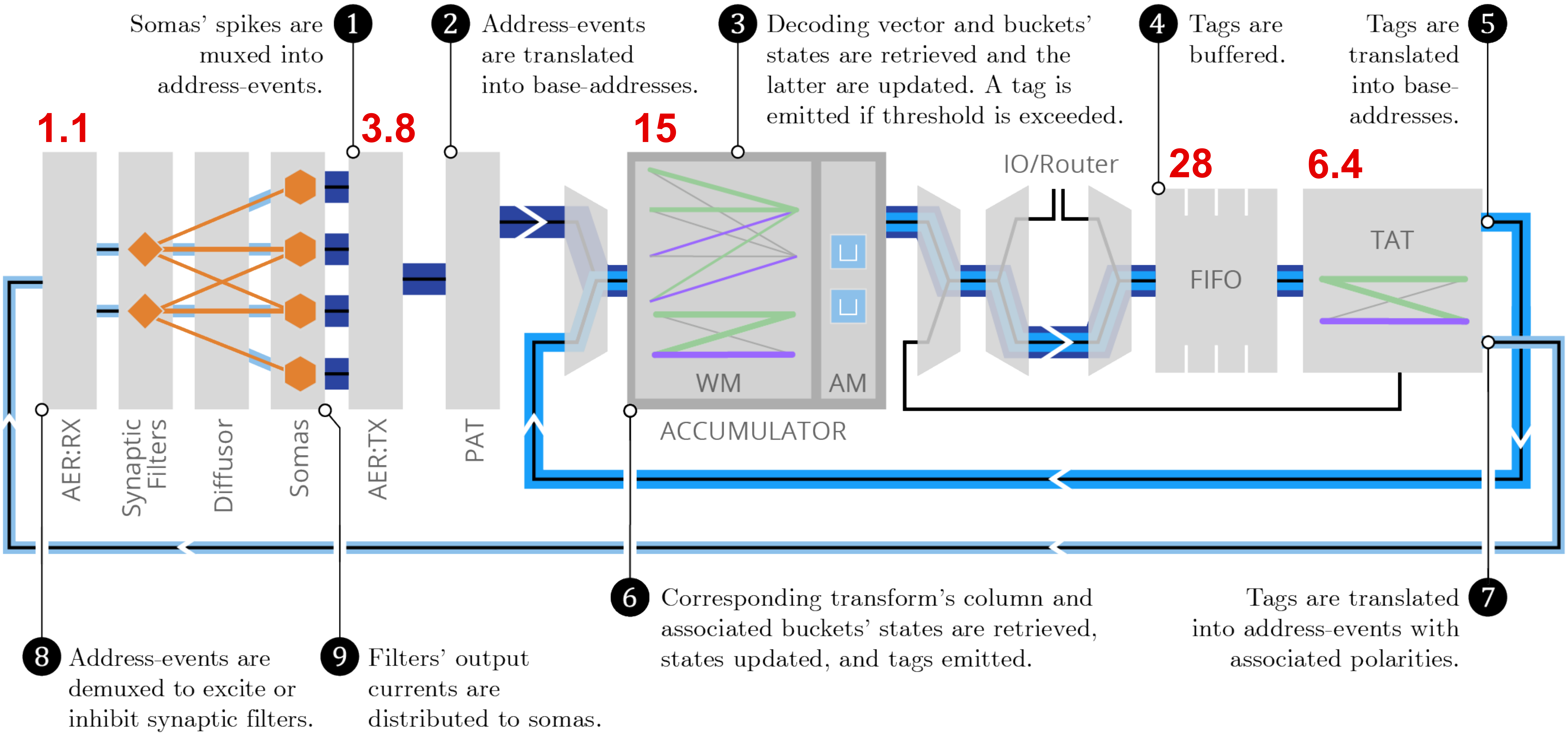
$$\theta \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bc(t)$$

$$c(t - \theta') \approx C_{\theta'/\theta}\mathbf{x}(t), \quad 0 \leq \theta' \leq \theta$$

Neckar et al. 2019

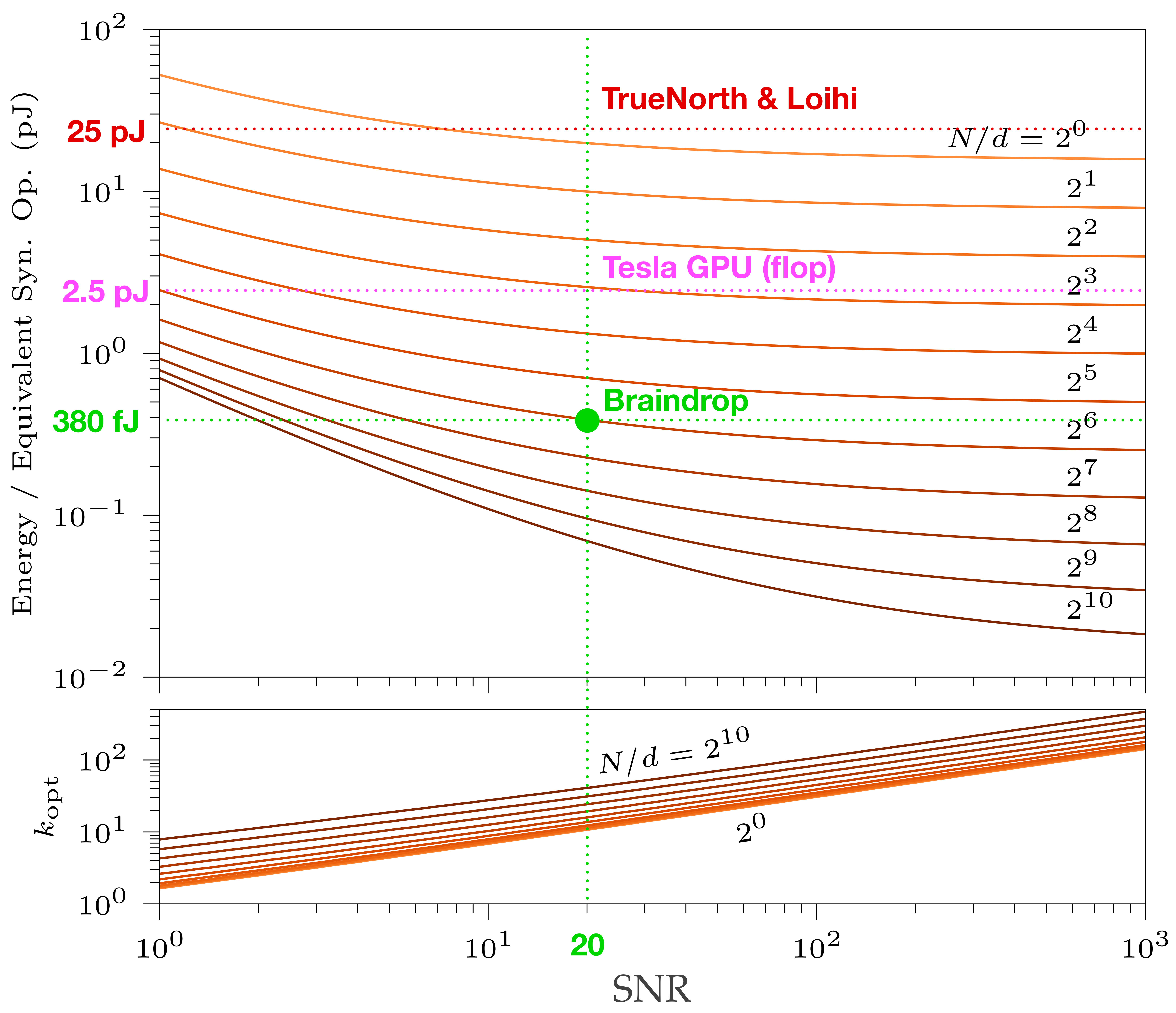
Voelker & Eliasmith 2017

Measured Energy/op (pJ)



- <u>Analog convolving</u> fans out d spike-trains to N neurons; <u>sparsifies spatially by d/N</u>

- <u>Digital thinning</u> lets one per SNR spikes through; <u>sparsifies temporally by 1/SNR</u>

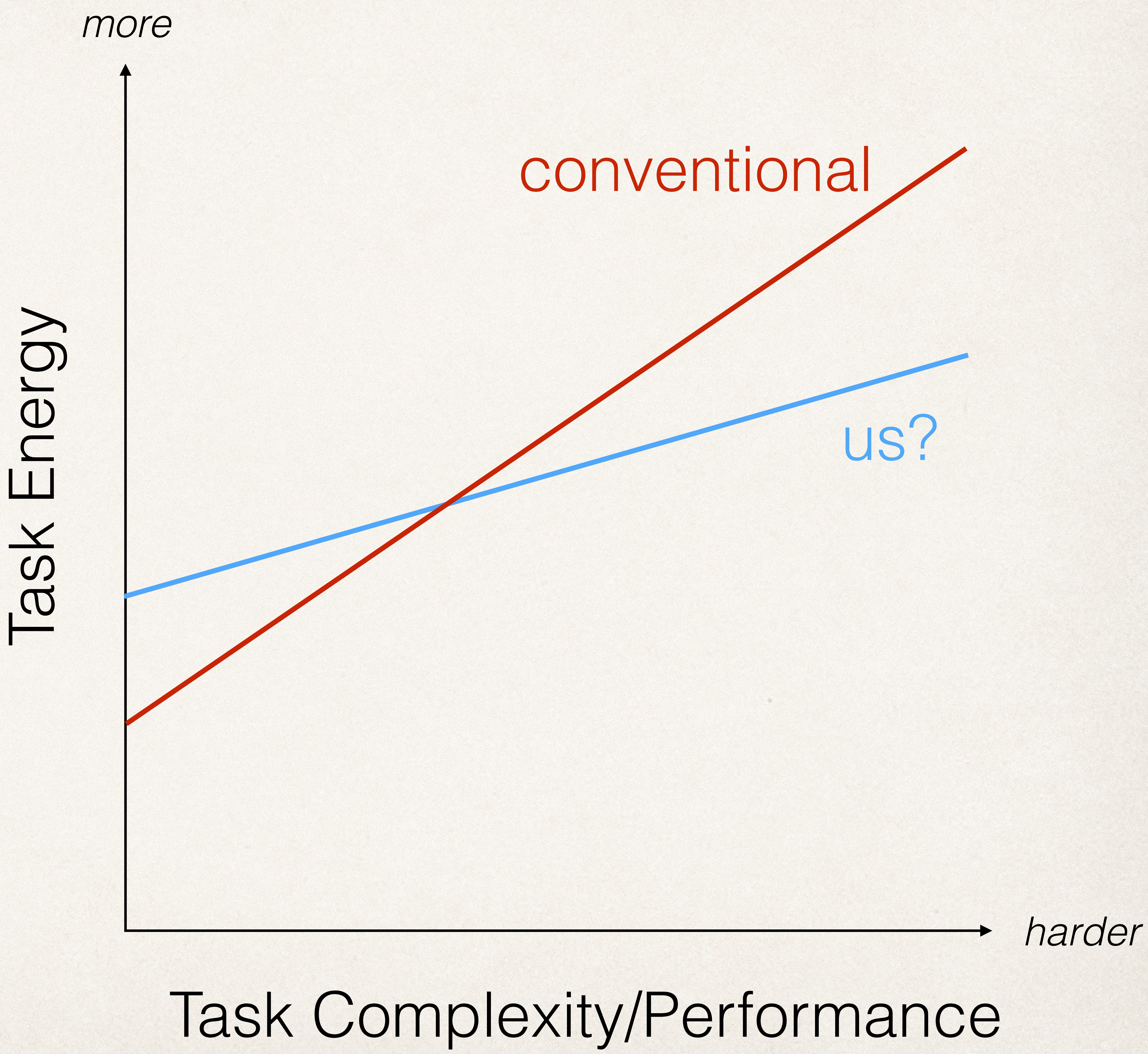


Task performance

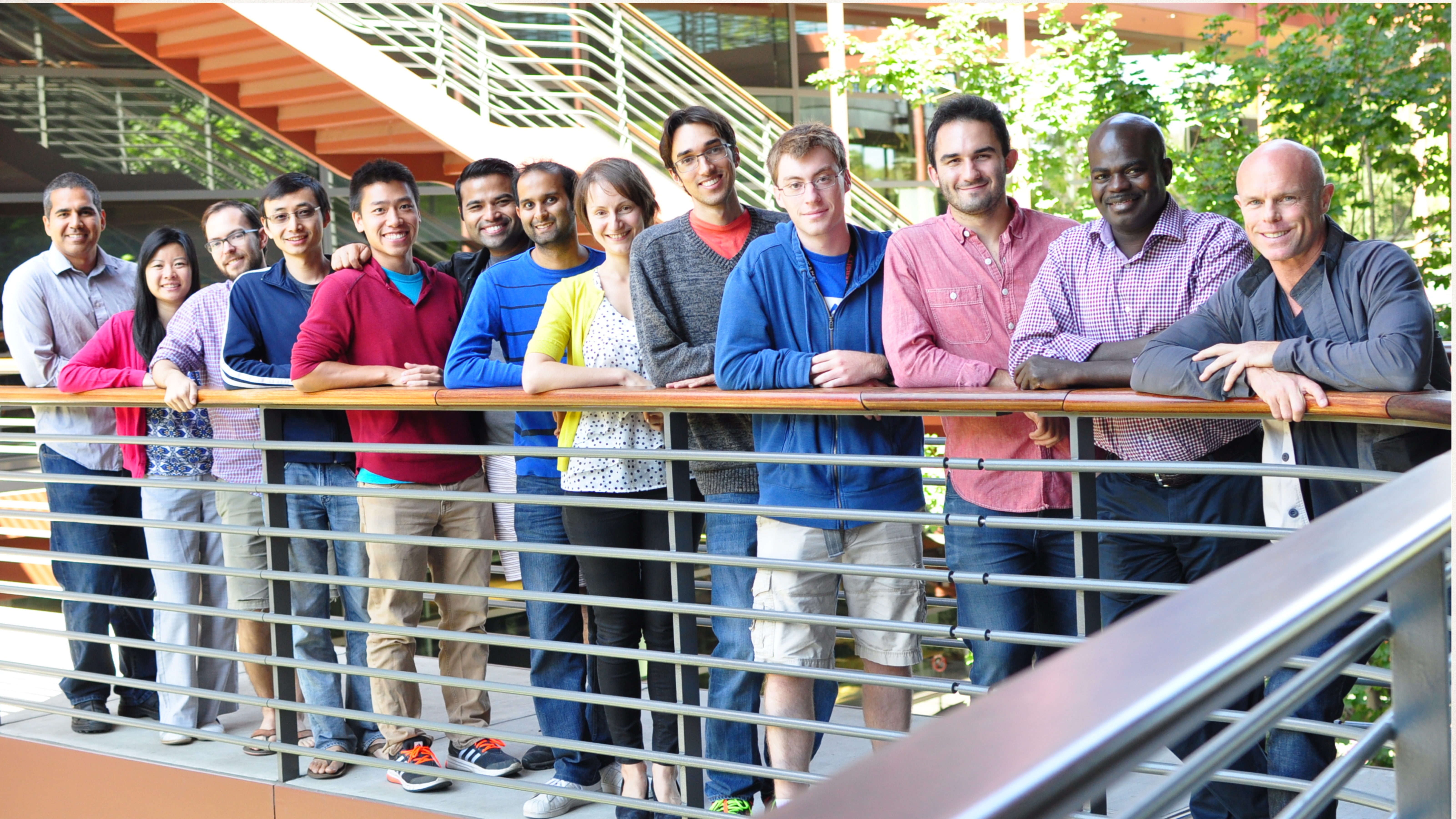
- Two components:
  - Network design
  - Hardware design

$$E_{sys}(task) = E_{HW}(R_{net}(task))$$

$$\begin{aligned} E_{op} &= N_{active} N_{conn} \\ &= (\rho_{active} N)(\rho_{conn} N) \end{aligned}$$



Acknowledgments



Students & Staff

Max Kanwal
Scott Reid
E. Kauderer-Abrams
Jonathan Timcheck

Recent Alumni

Ben Benjamin
Alex Neckar
Sam Fok
Nick Oza
Ashok Cutkosky
John Aguayo
Tatiana Engel
Samir Menon
Peiran Gao
Nick Steinmetz
Rodrigo Alvarez

Collaborators

Stanford

Krishna Shenoy
Tirin Moore
Oussama Khatib

Waterloo

Chris Eliasmith
Terry Stewart
Aaron Voelker

Cornell & Yale

Rajit Manohar
Ned Bingham
Silvia Ferarri
Taylor Clawson

Funding

NIH
Pioneer,TR01
ONR: C. Baatar

To learn more …

J Dethier, P Nuyujukian, C Eliasmith, T Stewart, S A Elassaad, K V Shenoy, and K Boahen, **A Brain-Machine Interface Operating with a Real-Time Spiking Neural Network Control Algorithm**, *Advances in Neural Information Processing Systems 24*, Curran Associates, Inc., pp 2213-21, 2011.

S Choudhary, S Sloan, S Fok, A Necker, E Trautmann, P Gao, T Stewart, C Eliasmith, and K Boahen, **Silicon Neurons that Compute**, *International Conference on Artificial Neural Networks, LNCS* vol VV, pp 121-128, Springer, Heidelberg, 2012.

S Menon, S Fok, A Neckar, O Khatib, and K Boahen, **Controlling Articulated Robots in Task-Space with Spiking Silicon Neurons**, *IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, IEEE Press, pp 181-186, 2014.

K Boahen, **A Neuromorph's Prospectus**, *Computing in Science & Engineering*, vol 19, no 2, pp 14-28, IEEE Computer Society, Los Alamitos CA, USA, 2017.

E Kauderer-Abrams, A Gilbert, A Voelker, B Benjamin, and T C Stewart, and K Boahen, **A Population-Level Approach to Temperature Robustness in Neuromorphic Systems**, *IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore MD, 2017.

A R Voelker, B V Benjamin, T C Stewart, K Boahen, and C Eliasmith, **Extending the Neural Engineering Framework for Nonideal Silicon Synapses**, *IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore MD, 2017.

E Kauderer-Abrams and K Boahen, **Calibrating Silicon-Synapse Dynamics using Time-Encoding and Decoding Machines**, *IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore MD, 2017.

Proceedings of the IEEE, Jan 2019

INVITED PAPER

Braindrop: A Mixed-Signal Neuromorphic Architecture With a Dynamical Systems-Based Programming Model

This paper provides an overview of a current approach for the construction of a programmable computing machine inspired by the human brain.

By Alexander Neckar, Sam Fok, Ben V. Benjamin, Terrence C. Stewart, Nick N. Oza, Aaron R. Voelker, Chris Eliasmith, Rajit Manohar, *Senior Member IEEE*, and Kwabena Boahen, *Fellow IEEE*