

Merging insights from artificial and biological neural networks

A competitive advantage for neuromorphic edge intelligence?

Charlotte Frenkel
(c.frenkel@tudelft.nl)

Assistant Professor
Dept. of Microelectronics, Delft University of Technology

*ETH Zurich
March 20th, 2024*

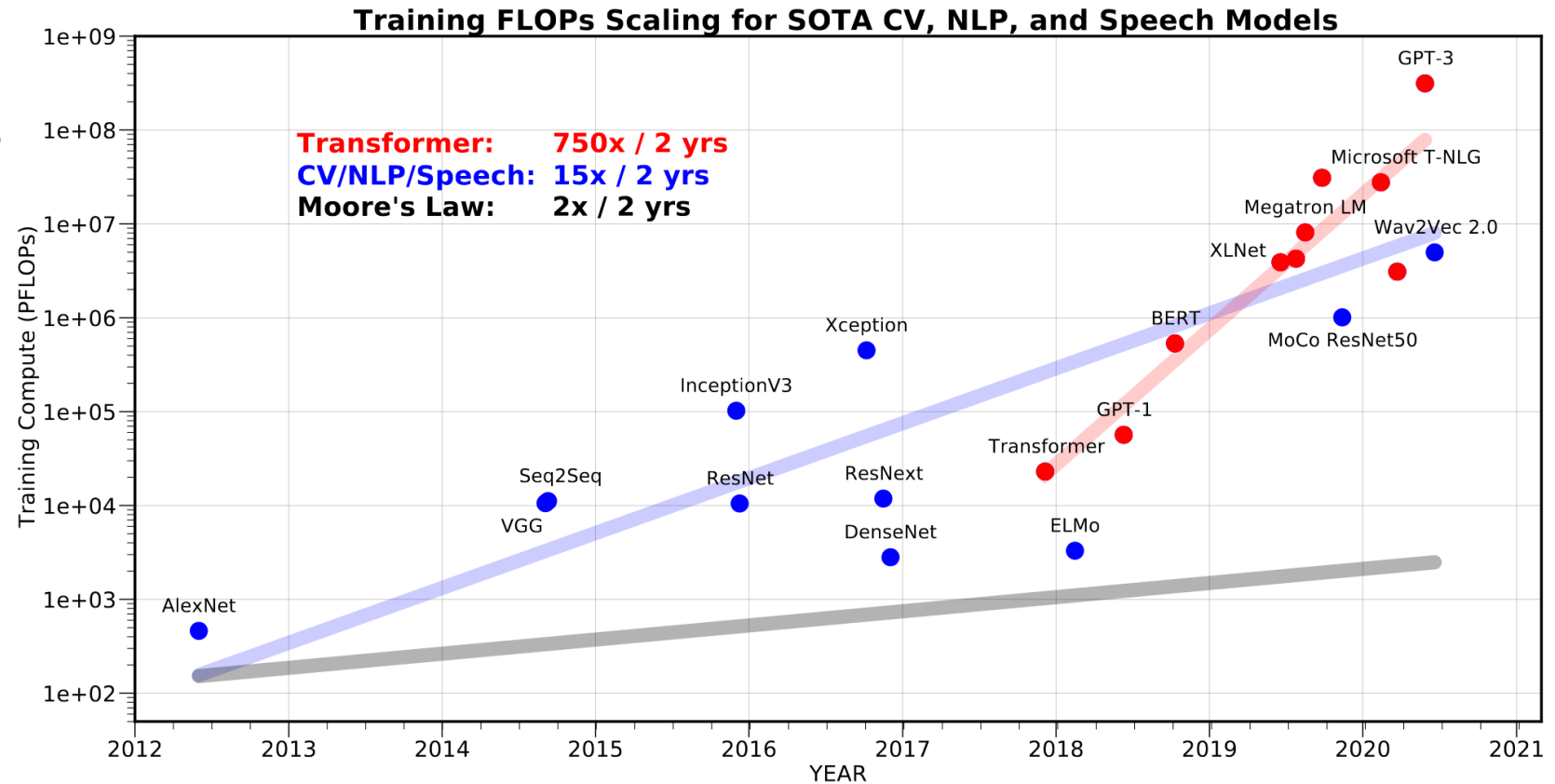
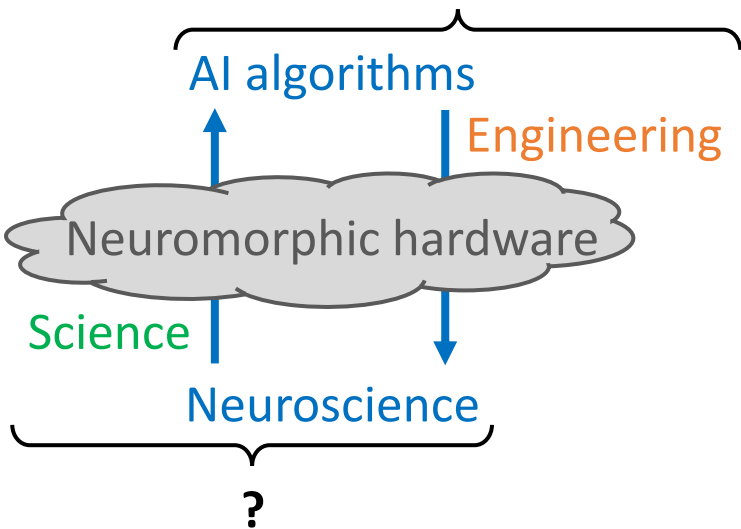
Outline

- ① From neuroscience to AI and back again...
...which perspective?
...which starting point?
- ② Why should we bother with neuroscience?
- ③ How can we morph these questions into interesting engineering solutions?

From neuroscience to AI and back again

Which starting point? Which perspective?

AI without hardware is unsustainable



[A. Gholami, *RiseLab Medium Post*, 2021]

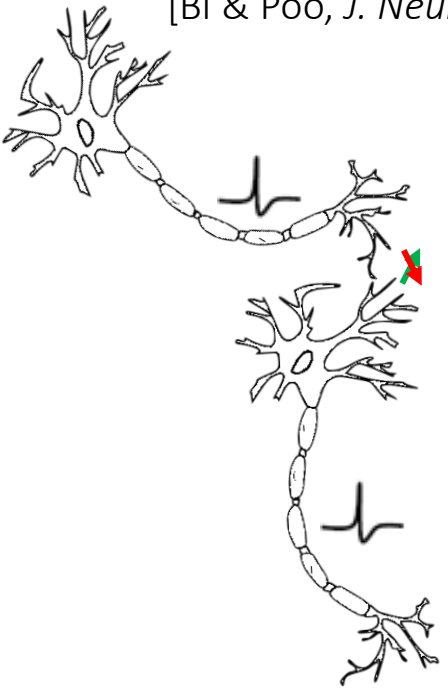
Outline

- ① From neuroscience to AI and back again...
...which perspective?
...which starting point?
- ② Why should we bother with neuroscience?
- ③ How can we morph these questions into interesting engineering solutions?

Synaptic plasticity rules – Neuroscience as the starting point

Spike-timing-dependent plasticity (STDP)

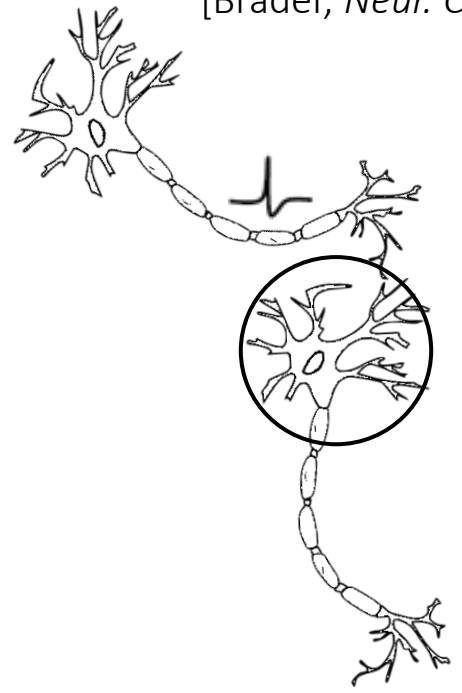
[Bi & Poo, *J. Neurosci.*, 1998]



✓ Local

Spike-dependent synaptic plasticity (SDSP)

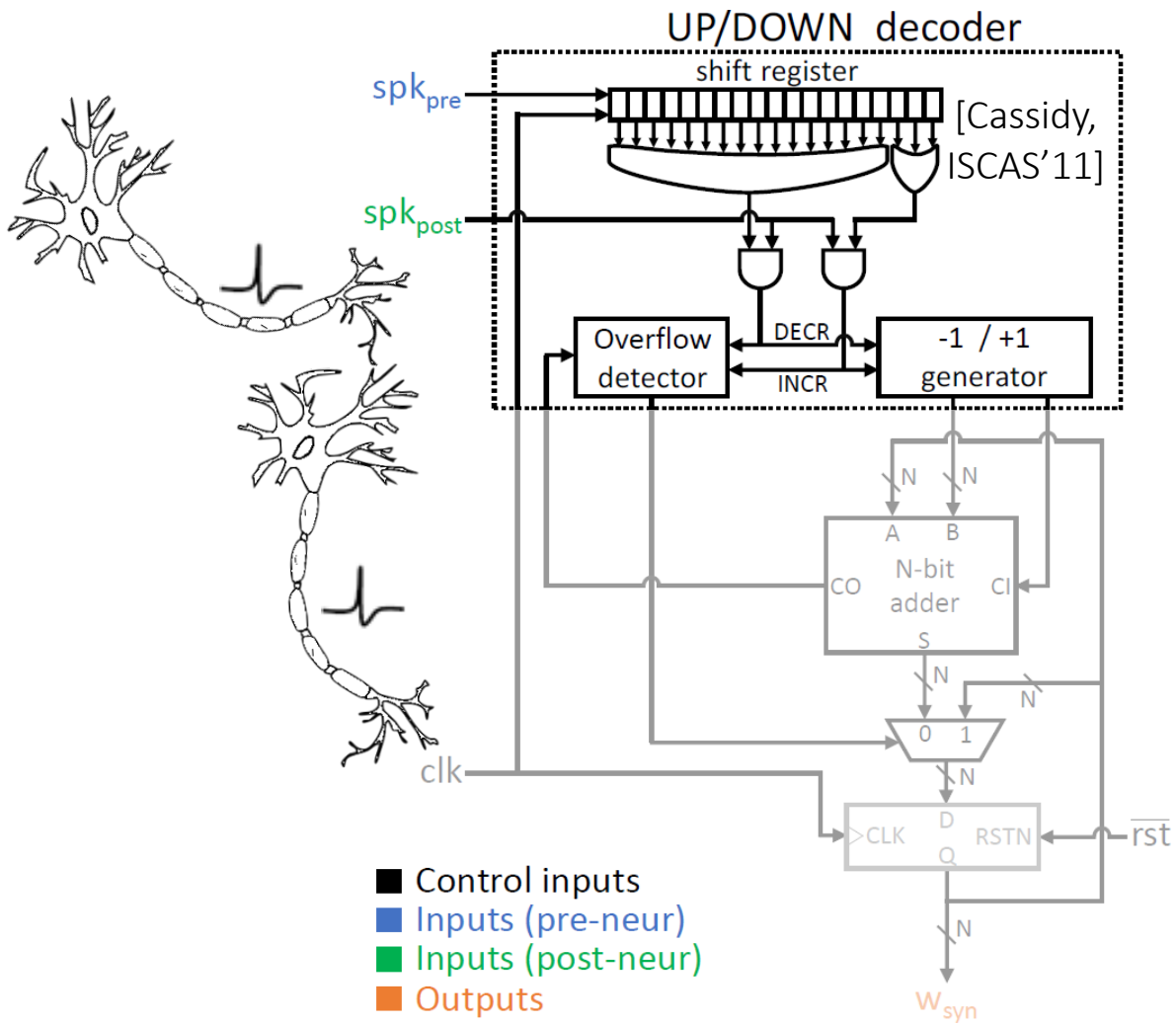
[Brader, *Neur. Comp.*, 2007]



✓ Local

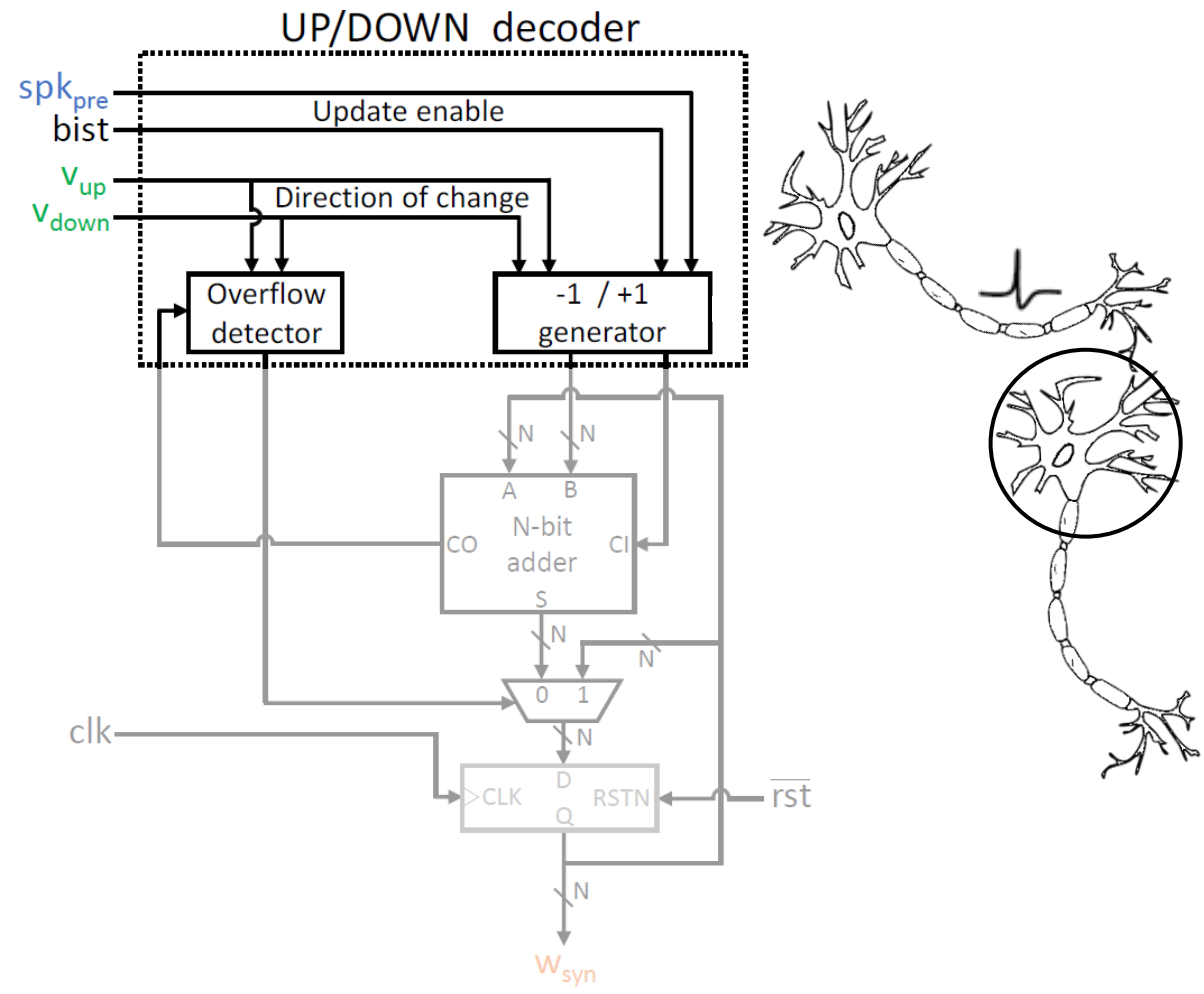
Synaptic plasticity rules – Neuroscience as the starting point

Digital synapse implementation



STDP

[Frenkel, *Trans. BioCAS*, 2019]



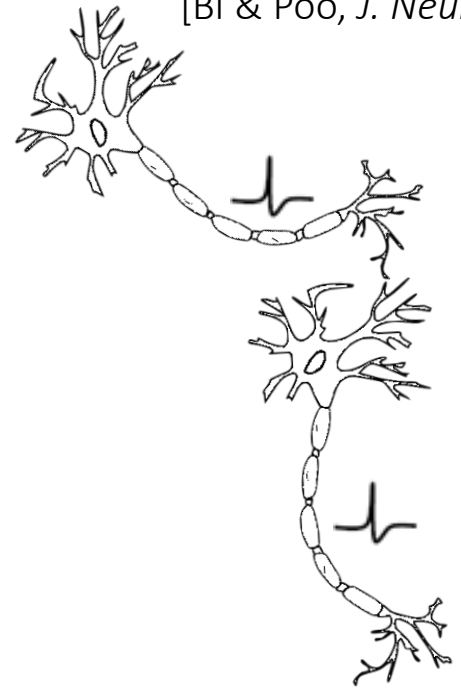
SDSP

Synaptic plasticity rules – Neuroscience as the starting point

The key perspective of data locality

Spike-timing-dependent plasticity (STDP)

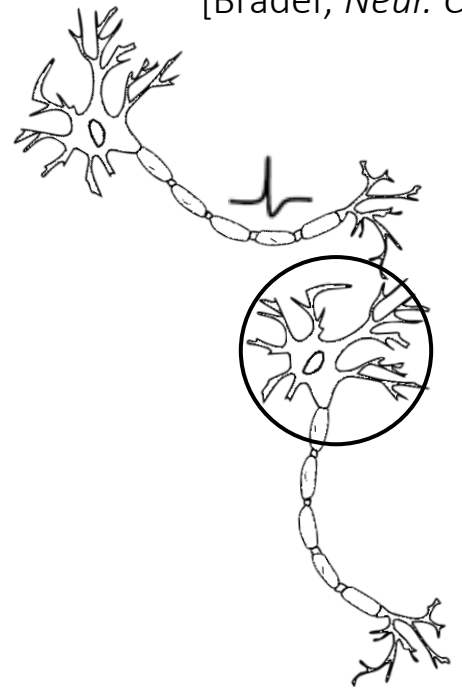
[Bi & Poo, *J. Neurosci.*, 1998]



- ✓ **Local** in space
- ✗ **Non-local** in time

Spike-dependent synaptic plasticity (SDSP)

[Brader, *Neur. Comp.*, 2007]



- ✓ **Local** in space
- ✓ **Local** in time

Huge savings in silicon

[Clopath and Gerstner, *Front. Syn. Neuro.*, 2010]

[Frenkel, *TBioCAS*, 2019]

Synaptic plasticity rules – Neuroscience as the starting point

The ODIN neuromorphic chip – Architecture

ODIN is a 256x256 SNN crossbar!

Features (aka “salt and pepper”)

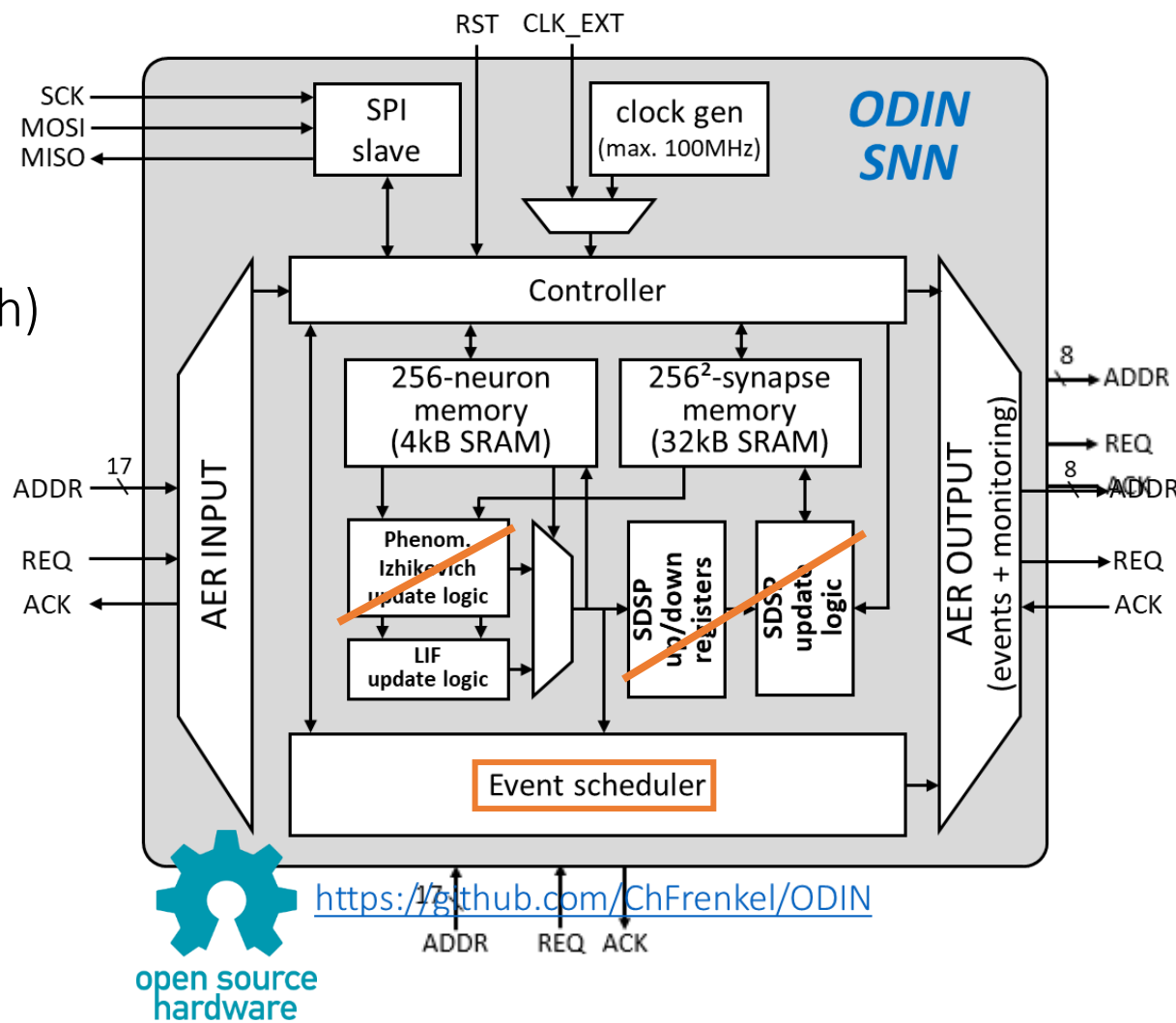
- Synaptic plasticity (SDSP)
- Large neuron behavior repertoire (LIF + Izhikevich)

<https://github.com/ChFrenkel/tinyODIN>

Design decisions

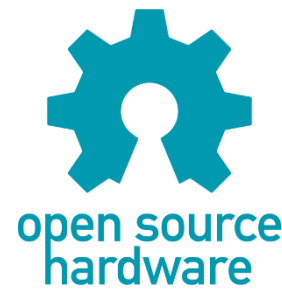
- Simple
- Low-cost
- Flexible and portable
- Full space and time locality
- No PDE solvers (= phenomenological modelling)

} Digital and time-multiplexed



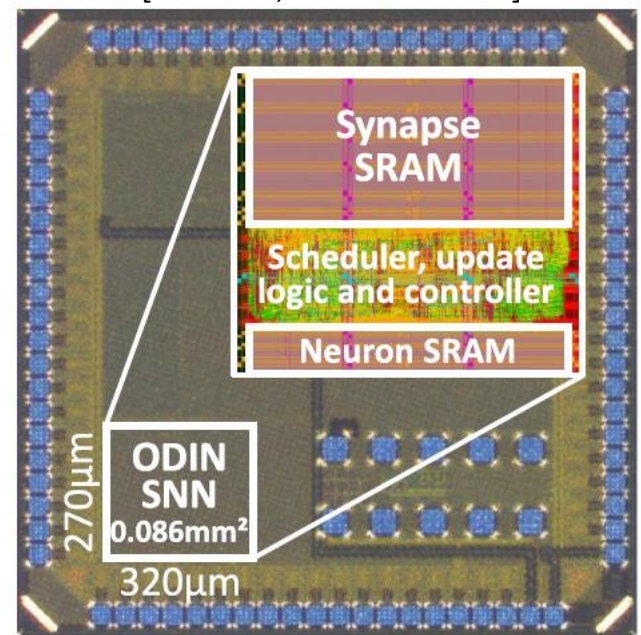
Synaptic plasticity rules – Neuroscience as the starting point

The ODIN and MorphIC neuromorphic chips – Silicon



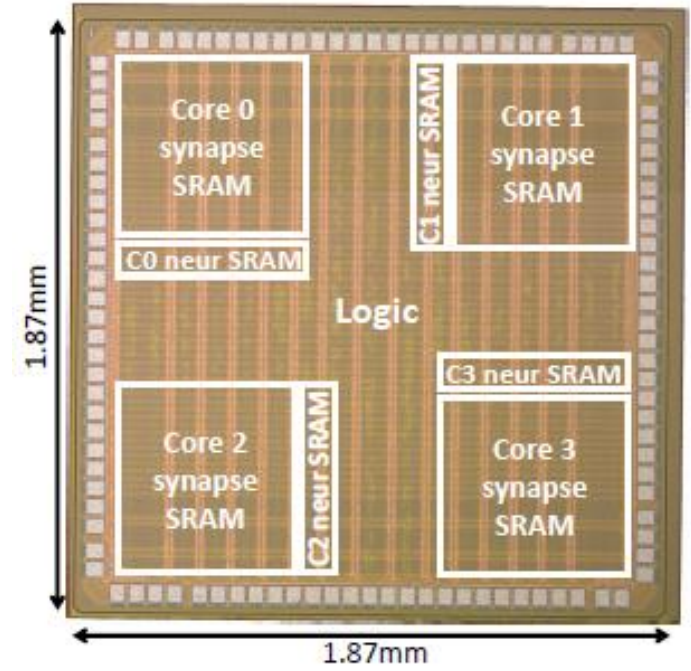
ODIN (single-core)

[Frenkel, TBioCAS'19a]



MorphIC (quad-core)

[Frenkel, TBioCAS'19b]

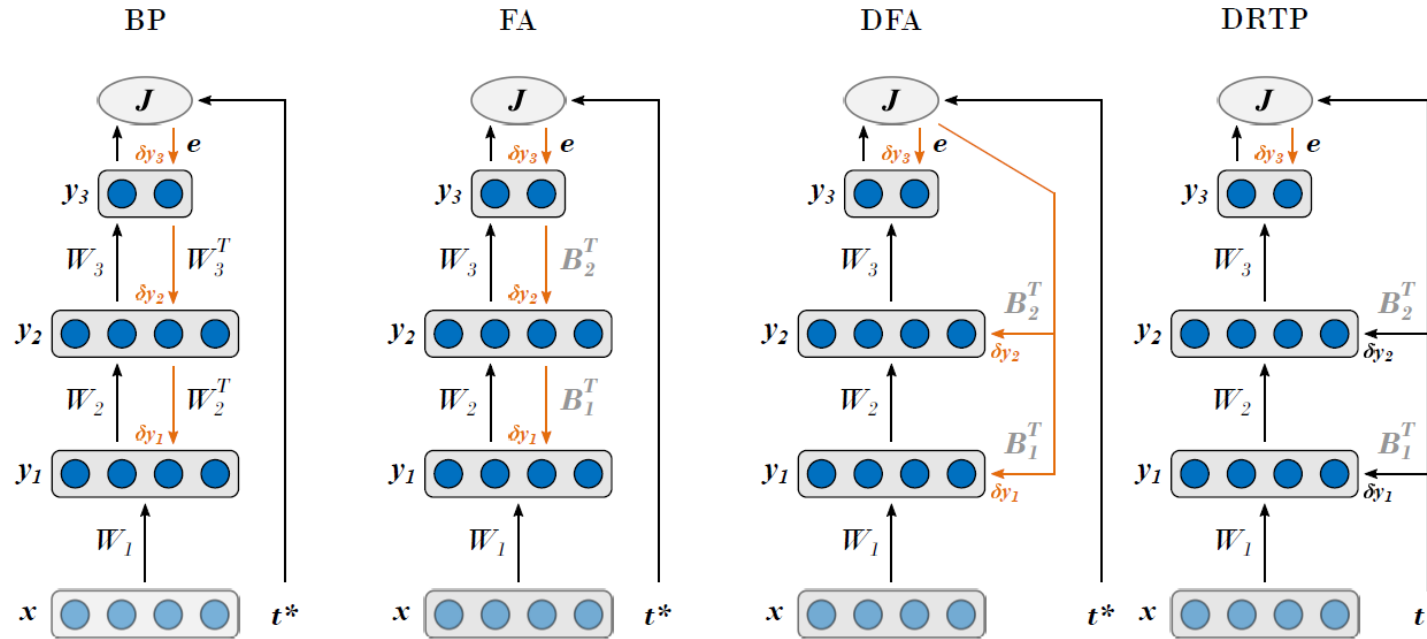


- ✔ Record synaptic density
- ✔ Energy efficiency competitive with mixed-signal designs
- ✔ Large feature set (incl. 20 Izhikevich behaviors, synaptic plasticity)
...but quite painful to exploit!

Neural network training – Bio-plausibility as the end goal

Synergy with hardware: latency, memory access patterns

AI algorithms
↓
Neuroscience



Output-independent target signals are also found in the brain!
[Magee & Grienberger, Ann. Rev. Neuro., 2020]

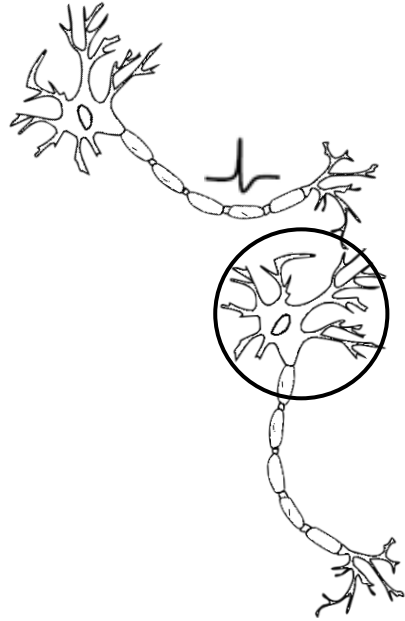
| δy_k | $\frac{\partial J}{\partial y_k} = W_{k+1}^T \delta z_{k+1}$ | $B_k^T \delta z_{k+1}$ | $B_k^T e$ | $B_k^T t^*$ |
|-----------------------|--|------------------------|-----------|-------------|
| Weight-transport-free | × | ✓ | ✓ | ✓ |
| Update-unlocked | × | × | × | ✓ |

↓ Computational and memory cost ↓

Only ~15% overhead in power and area [Frenkel, ISCAS'20] (🏆 Best paper award)

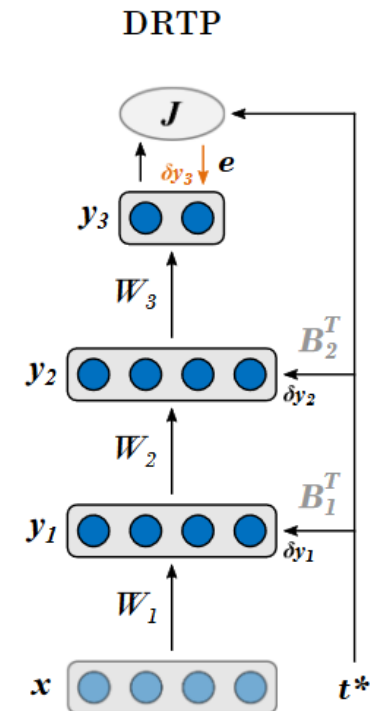
HW efficiency and bio-plausibility are often two sides of the same coin!

Many more examples: quantization, stochastic computing, event-driven computation,...



Designing efficient hardware hints toward bio-plausible mechanisms

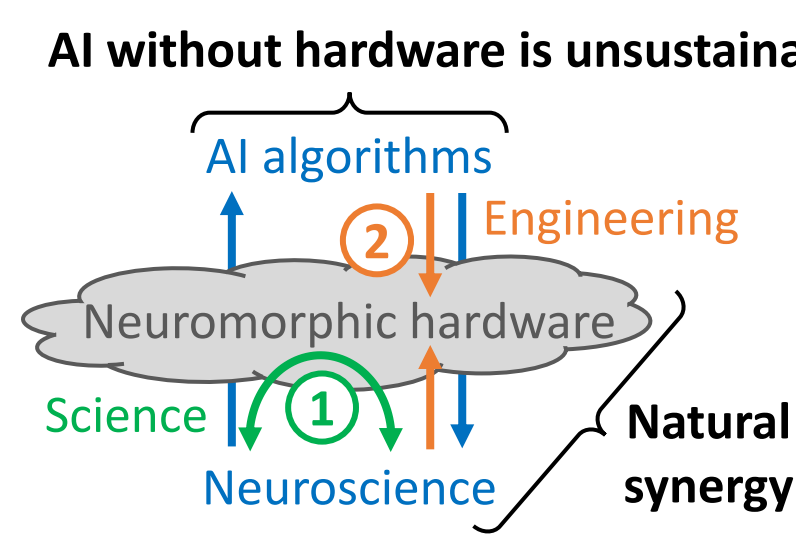
Bringing AI closer to neuroscience leads to hardware efficiency



From neuroscience to AI and back again

Which starting point? Which perspective?

AI without hardware is unsustainable



① Bottom-up science-driven approach

- ✓ Analysis-by-synthesis
- ✗ Difficult to scale efficiently to real-world problems

② Top-down engineering-driven approach

- ✓ Starts from working solutions to real-world problems
- ✗ Which “salt & pepper” from neuroscience?

Neuromorphic intelligence:

② should be fed by ①

Outline

- ① From neuroscience to AI and back again...
...which perspective?
...which starting point?
- ② Why should we bother with neuroscience?
- ③ How can we morph these questions into interesting engineering solution?

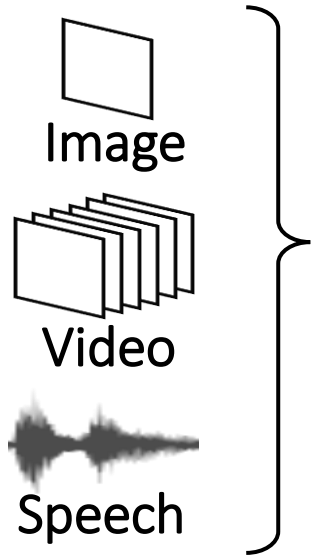
Let's use a 4-step recipe!

Neuromorphic intelligence:

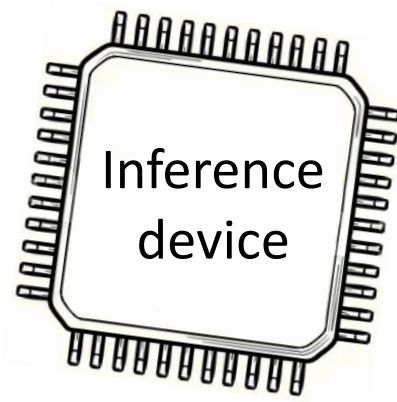
② should be fed by ①

1) Pick the use case

Why on-device learning is key!



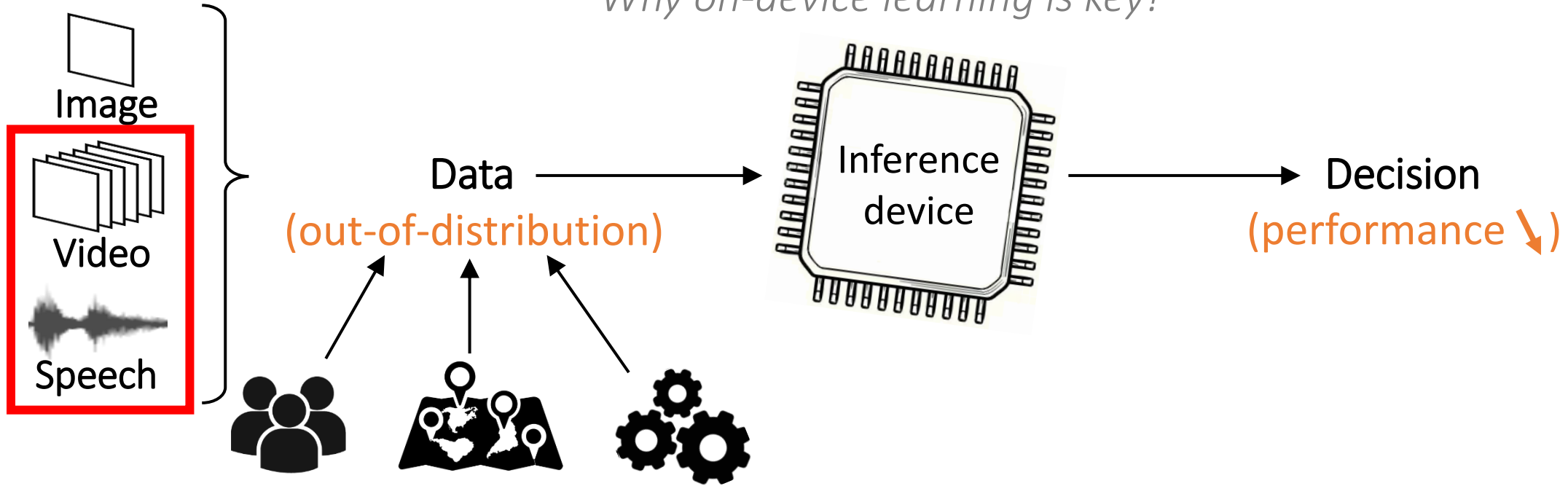
Data
(in-distribution)



Decision
(performance within specs)

1) Pick the use case

Why on-device learning is key!



Different users, environments, task requirements

More training data before deployment?

Issues: cost, robustness, flexibility

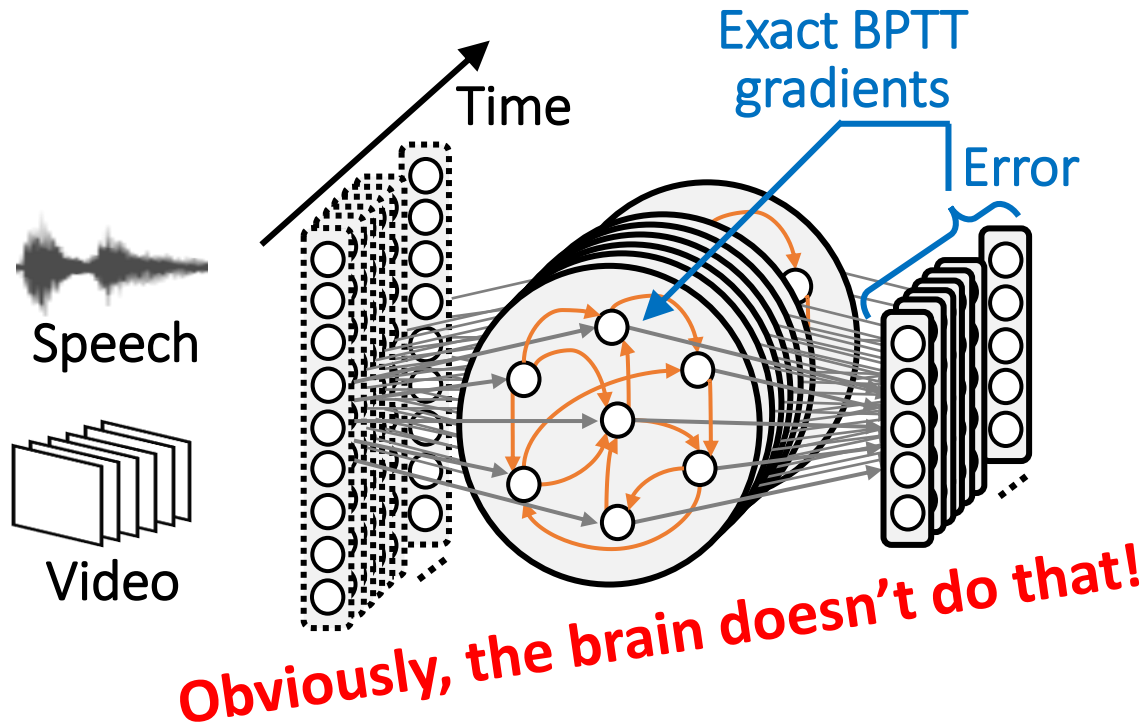
Data exchange with the cloud?

Issues: power budget, privacy

On-chip training
(end-to-end)

Why is on-chip learning over second-long timescales difficult?

Let's solve a yet unsolved engineering challenge!



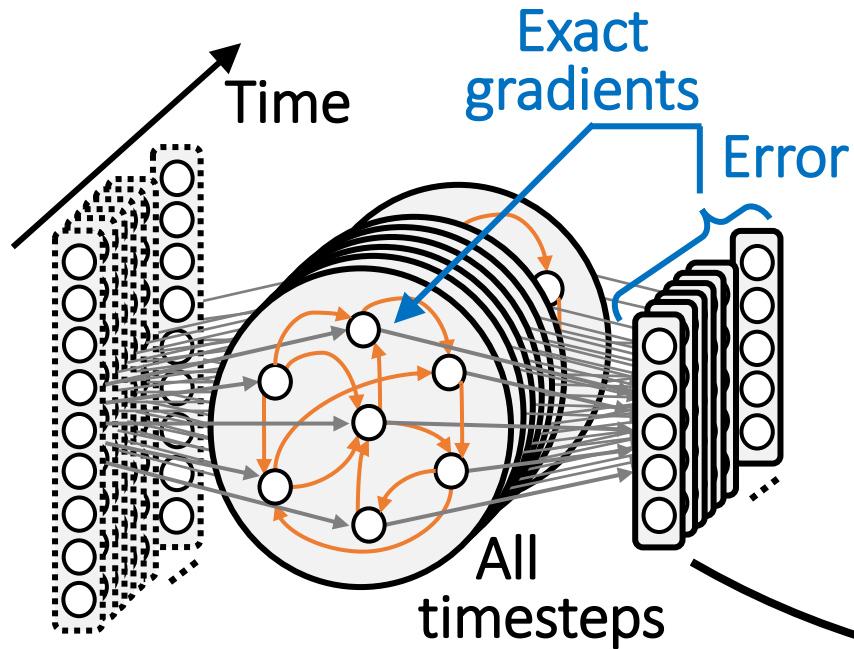
- Unrolling in time: very deep network (current learning ICs for static stimuli: ≤ 3 layers)
- Intractable memory/latency requirements
- No end-to-end on-chip solution to date

Key challenge: On-chip learning over long timescales while keeping a fine-grained temporal resolution

2) Select the (ML-informed) starting point

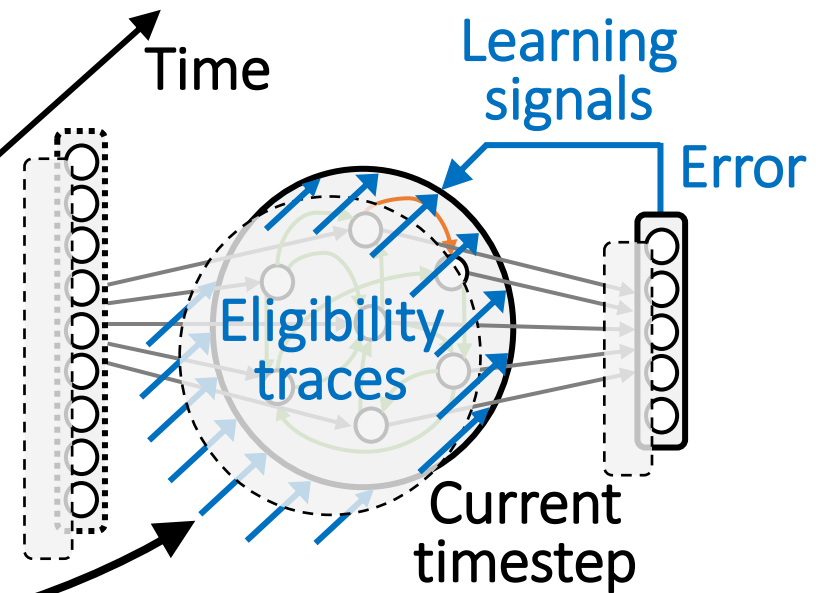
From BPTT to biologically plausible training

Backprop through time (BPTT, backward)



Eligibility propagation (e-prop, forward)

[Bellec, *Nat. Comms*'20]

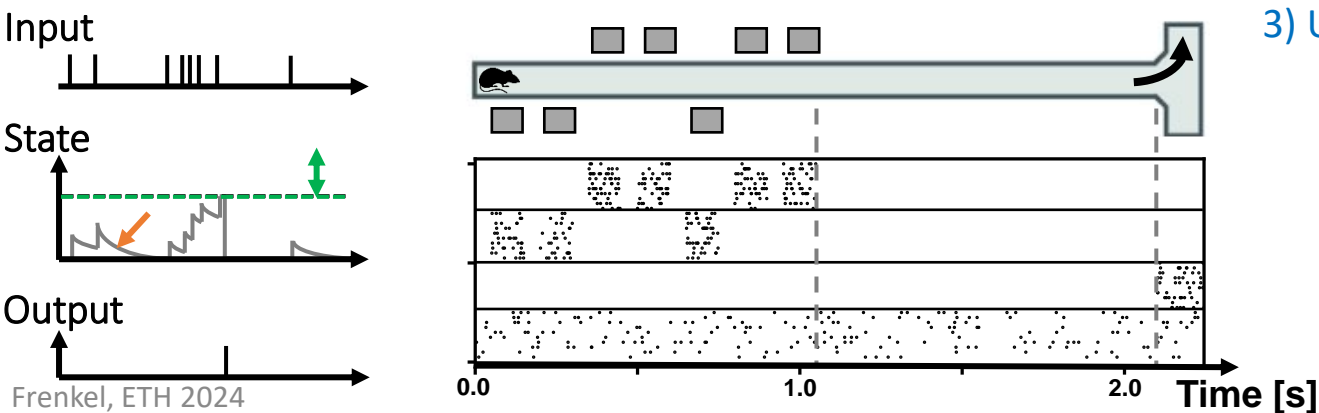
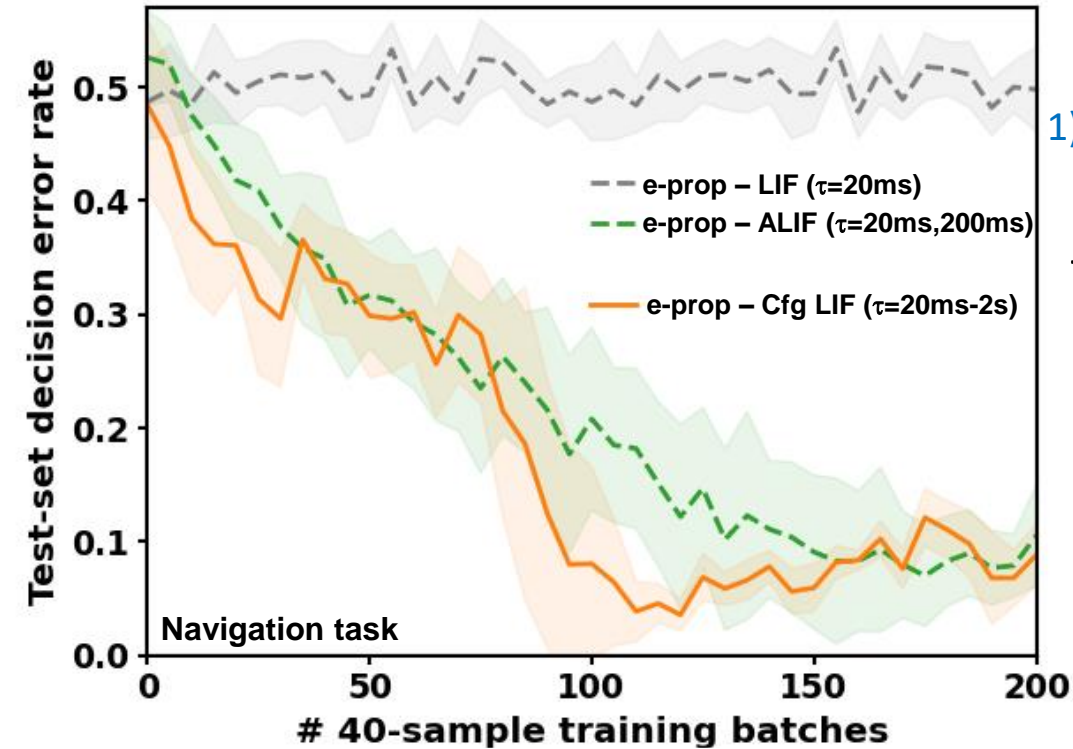


Key concept: space and time locality (again!)

And the brain is a good source of inspiration for this!

3) Use-case-driven feature set selection

Neuron model selection... driven by the application requirements!



1) Pick the use case

On-chip learning of temporal data

HW tractability?
Bio plausibility?

BPTT

2) Select the (ML-informed) starting point

Eligibility traces

e-prop

4) Space & time locality

Long timescales?

3) Use-case-driven feature set

Config. LIF

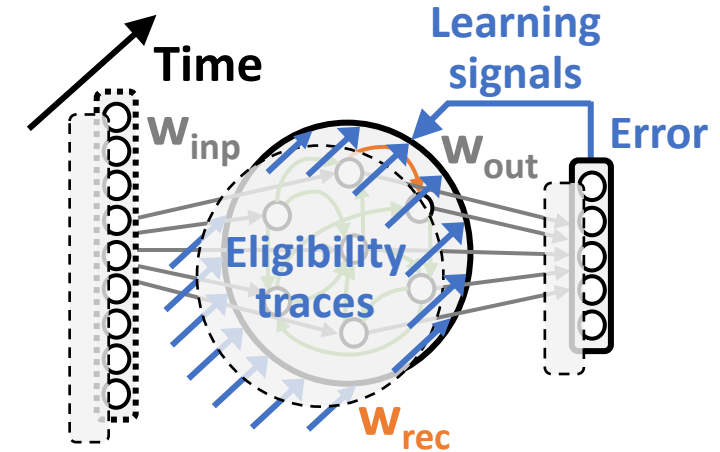
Threshold adaptation

Simplified e-prop

4) Enforce space and time locality

Key steps to minimize memory requirements

| | Learning signals (LS) | Eligibility traces (ET) |
|------------------------------------|---|---|
| $\Delta W_{out,kj} \propto \sum_t$ | $(y_k^t - y_j^t)$ | $\sum_{t'} (x_j^{t'} y_k^t)$ |
| $\Delta W_{rec,ji} \propto \sum_t$ | $\left(\sum_{k'} x_{k'} y_k^t (y_k^t - y_j^t) \right)$ | $\sum_{t'} \left(x_j^{t'} y_k^t \sum_{k'} (x_{k'}^{t'} y_k^t) \right)$ |
| $\Delta W_{inp,ji} \propto \sum_t$ | $\left(\sum_{k'} x_{k'} y_k^t (y_k^t - y_j^t) \right)$ | $\sum_{t'} \left(x_j^{t'} y_k^t \sum_{k'} (x_{k'}^{t'} y_k^t) \right)$ |



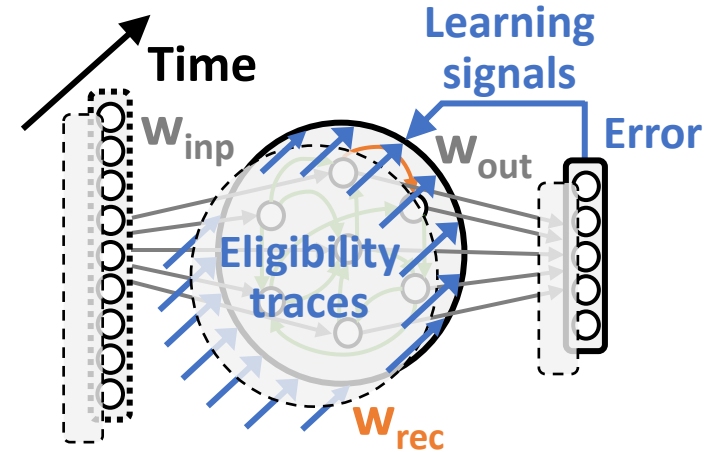
4) Enforce space and time locality

Key steps to minimize memory requirements

$$\begin{aligned}
 \Delta W_{\text{out},kj} &\propto \sum_t \left(y_k^{*,t} - y_k^t \right) \\
 \Delta W_{\text{rec},ji} &\propto \sum_t \left(\sum_k W_{\text{out},kj} \left(y_k^{*,t} - y_k^t \right) \right) \\
 \Delta W_{\text{inp},ji} &\propto \sum_t \left(\sum_k W_{\text{out},kj} \left(y_k^{*,t} - y_k^t \right) \right)
 \end{aligned}$$

Learning signals (LS)
Eligibility traces (ET)

Error

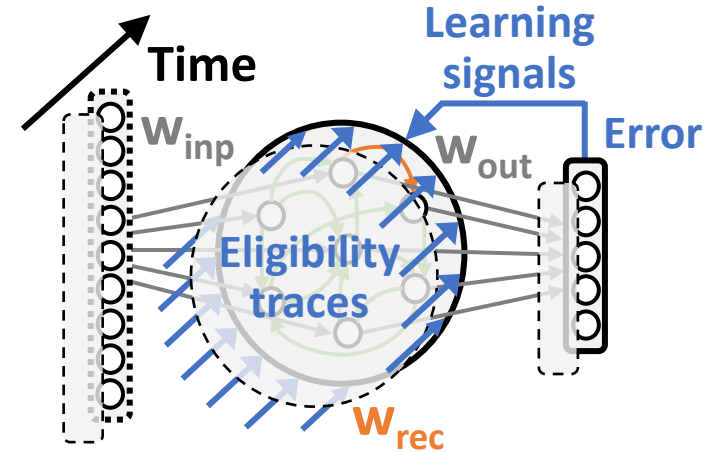


4) Enforce space and time locality

Key steps to minimize memory requirements

$$\begin{aligned}
 \Delta W_{out,kj} &\propto \sum_t (y_k^{*,t} - y_k^t) \\
 \Delta W_{rec,ji} &\propto \sum_t \left(\sum_k W_{out,kj} (y_k^{*,t} - y_k^t) \right) \\
 \Delta W_{inp,ji} &\propto \sum_t \left(\sum_k W_{out,kj} (y_k^{*,t} - y_k^t) \right)
 \end{aligned}$$

Learning signals (LS)
Eligibility traces (ET)

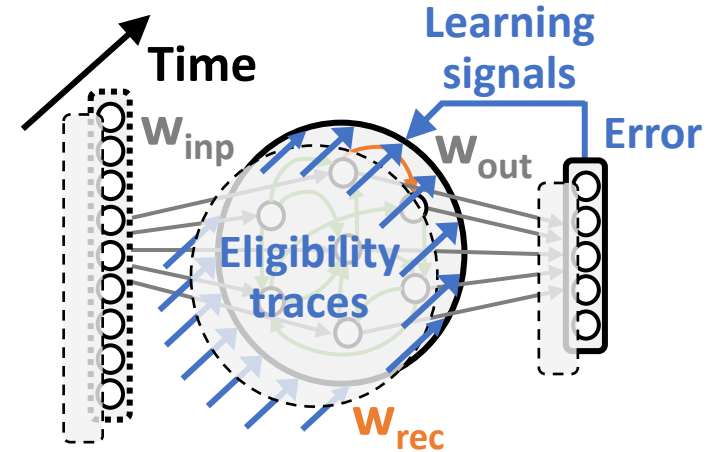


1 Requires a dedicated gradient memory \longrightarrow Per-timestep updates

4) Enforce space and time locality

Key steps to minimize memory requirements

| | Learning signals (LS) | Eligibility traces (ET) |
|-------------------------------|--|--|
| $\Delta w_{out,kj}^t \propto$ | $(y_k^{*,t} - y_k^t)$ | $\sum_{k'} (w_{out,kj}^{t-1} x_{k'}^t)$ |
| $\Delta w_{rec,ji}^t \propto$ | $\left(\sum_k w_{out,kj}^t (y_k^{*,t} - y_k^t) \right)$ | $\sum_{k'} (w_{rec,ji}^{t-1} \sum_{k''} (w_{out,kj}^{t-1} x_{k''}^t))$ |
| $\Delta w_{inp,ji}^t \propto$ | $\left(\sum_k w_{out,kj}^t (y_k^{*,t} - y_k^t) \right)$ | $\sum_{k'} (w_{inp,ji}^{t-1} \sum_{k''} (w_{out,kj}^{t-1} x_{k''}^t))$ |

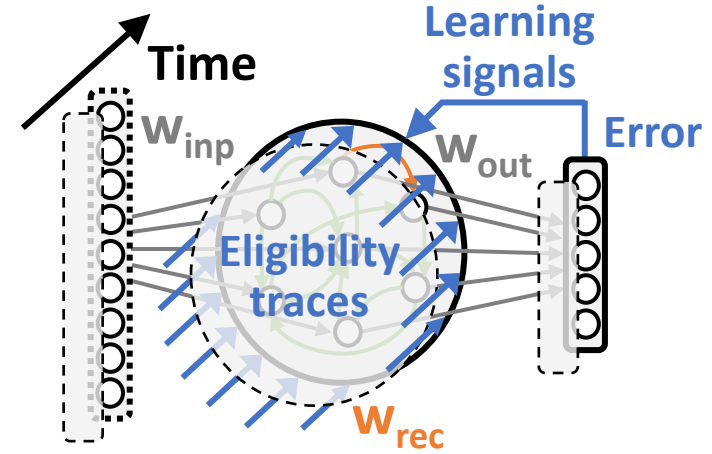


4) Enforce space and time locality

Key steps to minimize memory requirements

| | Learning signals (LS) | Eligibility traces (ET) |
|-------------------------------|--|---|
| $\Delta W_{out,kj}^t \propto$ | $(y_k^{*,t} - y_k^t)$ | $\sum_{t' \leq t} (\kappa^{t-t'} z_j^{t'})$ |
| $\Delta W_{rec,ji}^t \propto$ | $\left(\sum_k W_{out,kj}^t (y_k^{*,t} - y_k^t) \right)$ | $\sum_{t' \leq t} \left(\kappa^{t-t'} h_j^{t'} \sum_{t'' \leq t'} (\alpha^{t'-t''} z_i^{t''}) \right)$ |
| $\Delta W_{inp,ji}^t \propto$ | $\left(\sum_k W_{out,kj}^t (y_k^{*,t} - y_k^t) \right)$ | $\sum_{t' \leq t} \left(\kappa^{t-t'} h_j^{t'} \sum_{t'' \leq t'} (\alpha^{t'-t''} x_i^{t''}) \right)$ |

Post-synaptic straight-through estimator (STE) Pre-synaptic activity LPF



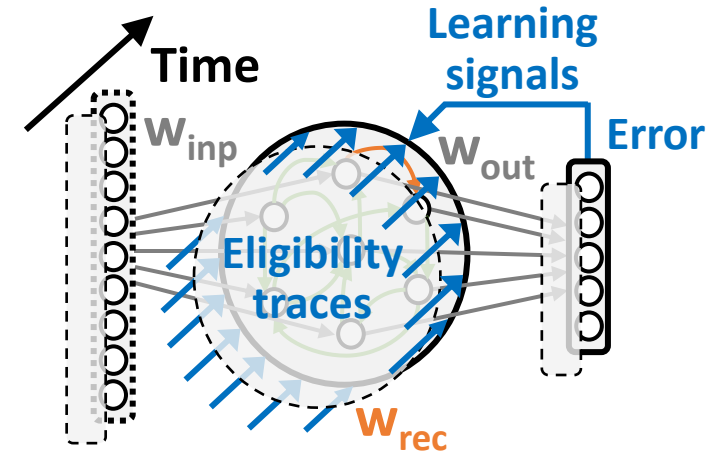
2 Temporal coupling of pre- and post-synaptic terms \longrightarrow Can be neglected

4) Enforce space and time locality

Key steps to minimize memory requirements

$$\frac{dE}{dW_{ij}} \approx \sum_t L_j^t e_{ji}^t$$

Memory overhead scales with #synapses in $O(N^2)$



Local decoupling of space and time:

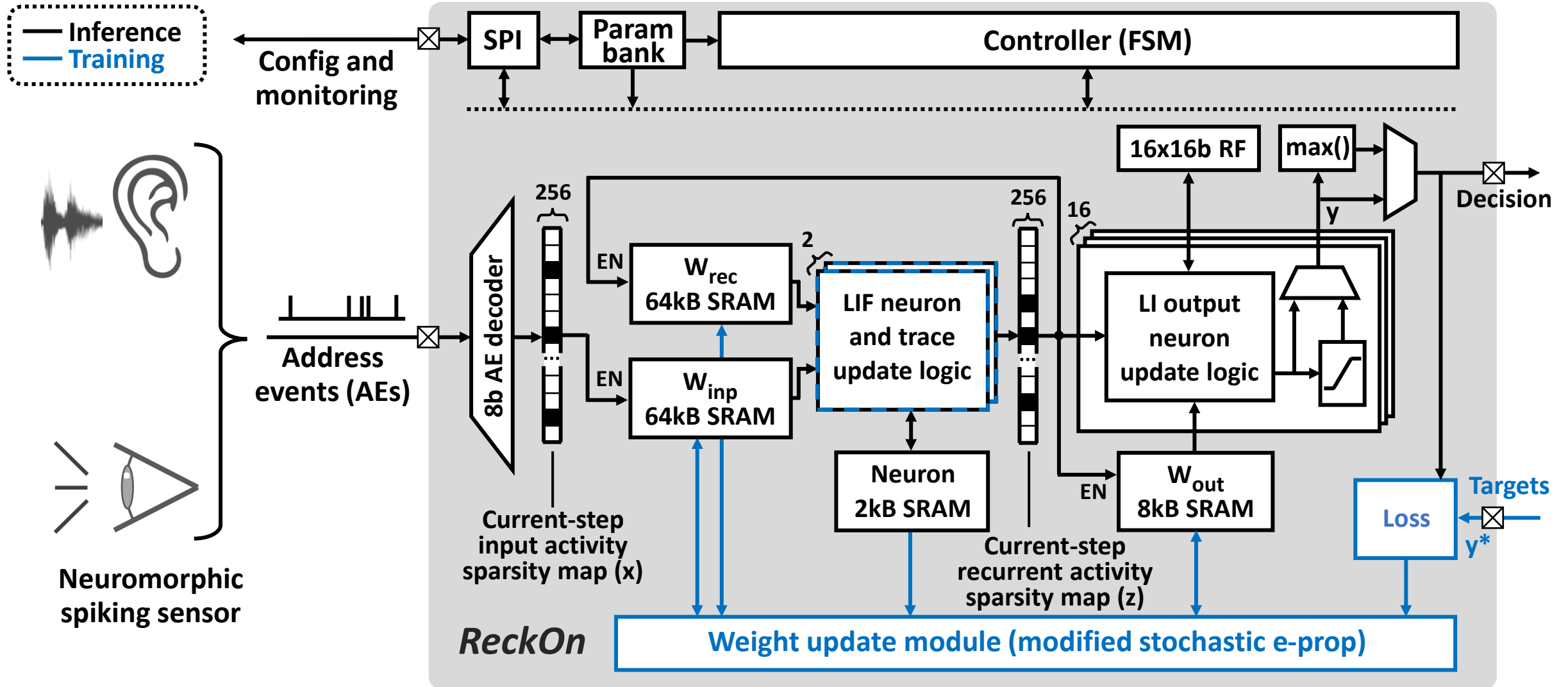
- **Pre-synaptic term:** activity low-pass filtering
- **Post-synaptic term:** surrogate derivative of the spiking activation function

Scales with #neurons in $O(N)$

Stochastic weight updates allow reducing weight resolution to 8 bits

The ReckOn neuromorphic chip – Architecture

Same recipe as for ODIN: time multiplexing, no PDE solver, space and time locality



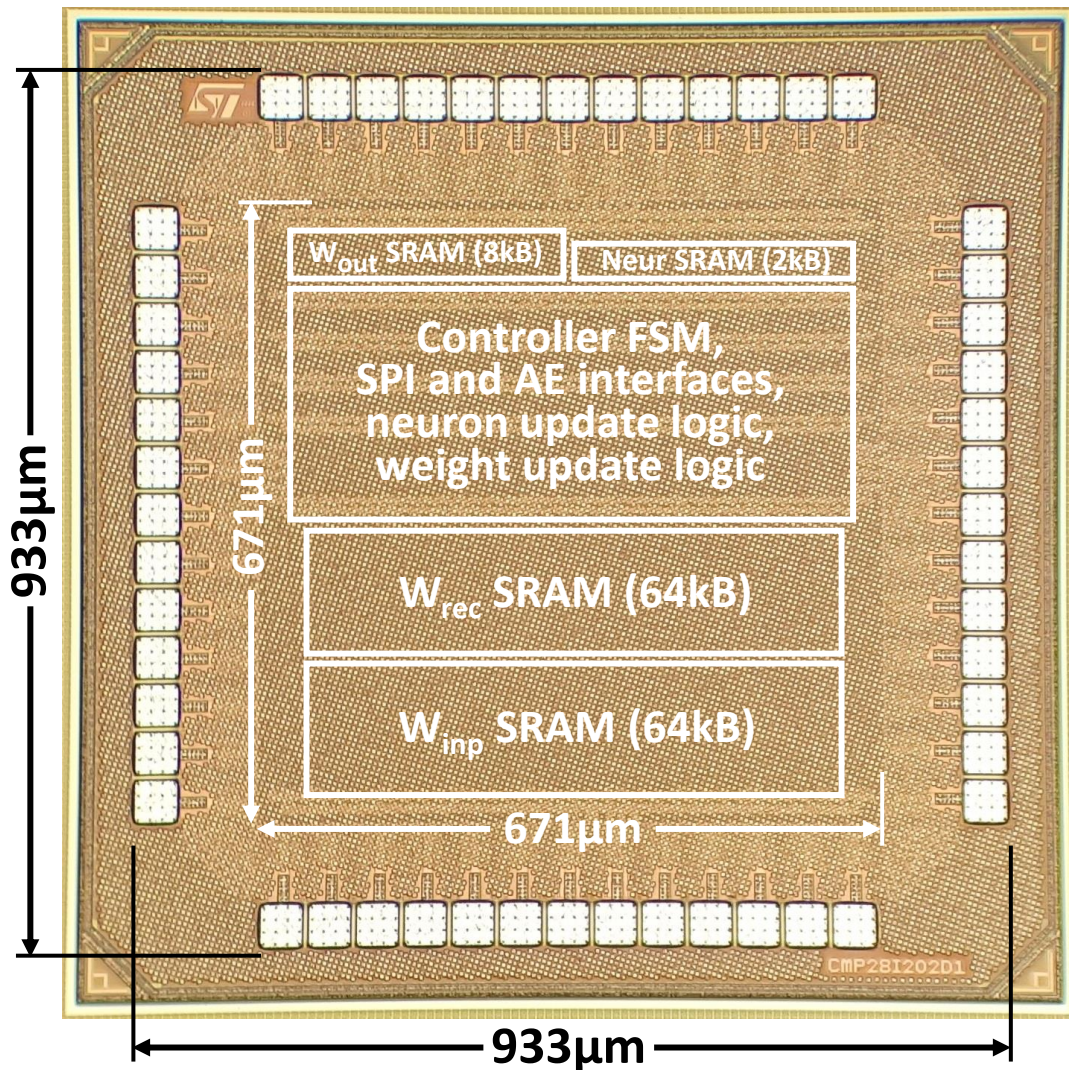
The ReckOn neuromorphic chip – Microphotograph and summary



University of Zurich^{UZH}



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



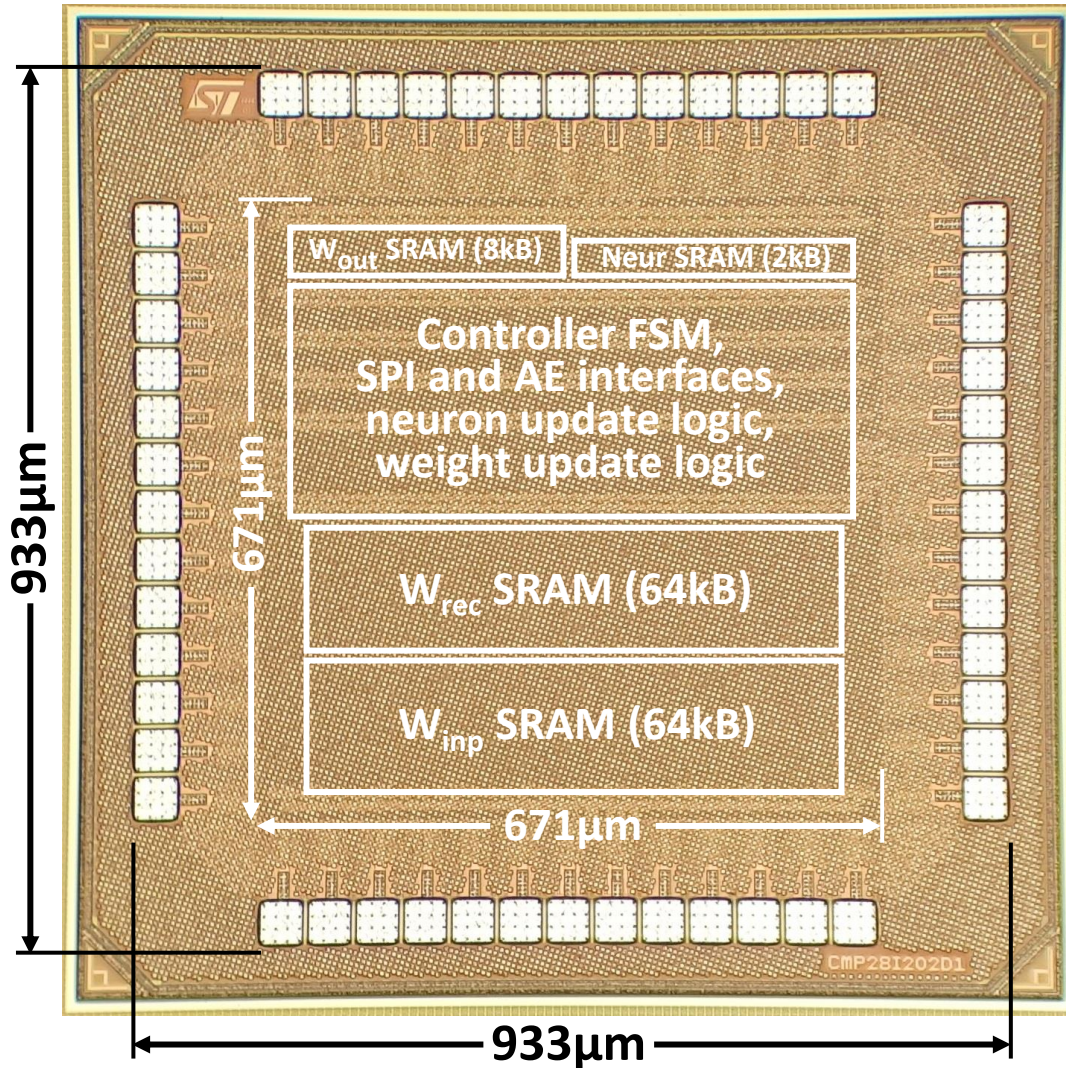
| | | |
|-------------------|-----------------------------|---------------------|
| Technology | 28nm FDSOI CMOS | |
| Core size | 0.67 x 0.67 mm ² | 0.45mm ² |
| Die size | 0.93 x 0.93 mm ² | |
| SRAM | 138kB | + 0kB ext. DRAM! |
| Network | Spiking RNN | |
| Training timespan | Max. 32k steps | |



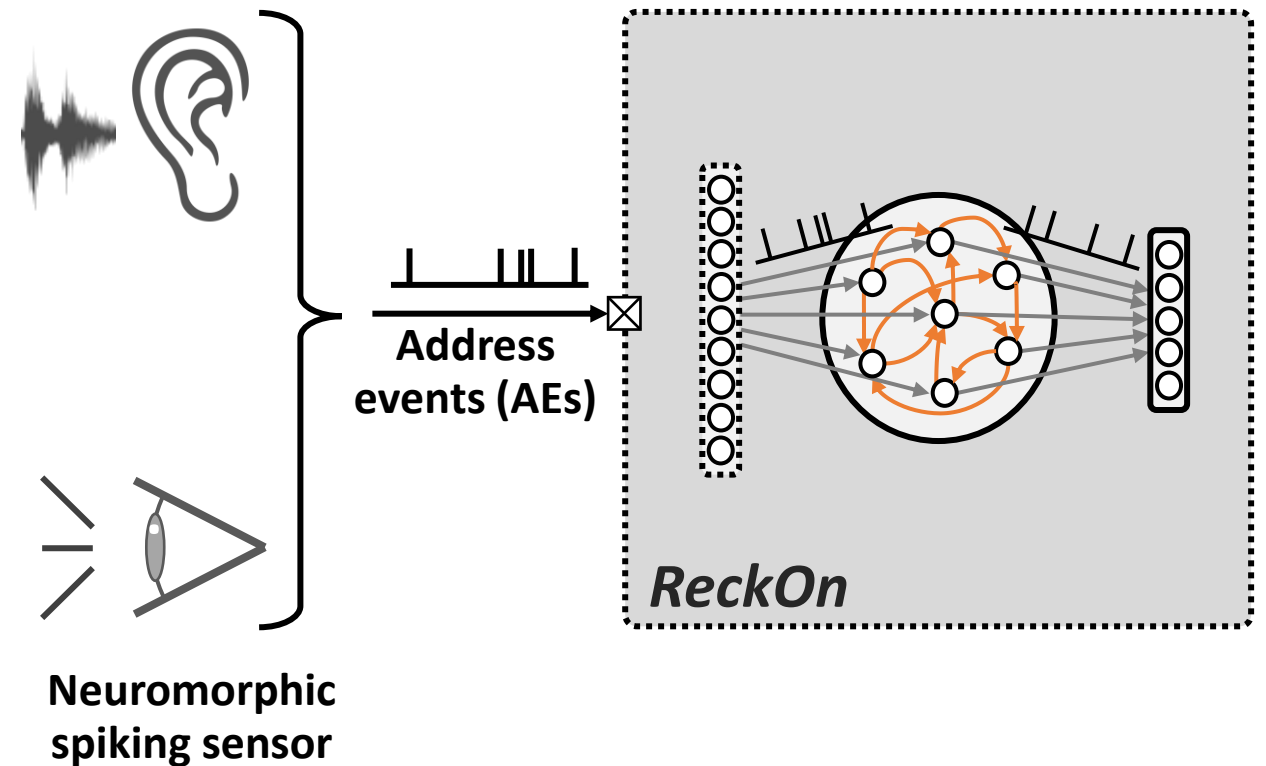
open source hardware



The ReckOn neuromorphic chip – Key advantage of using spikes

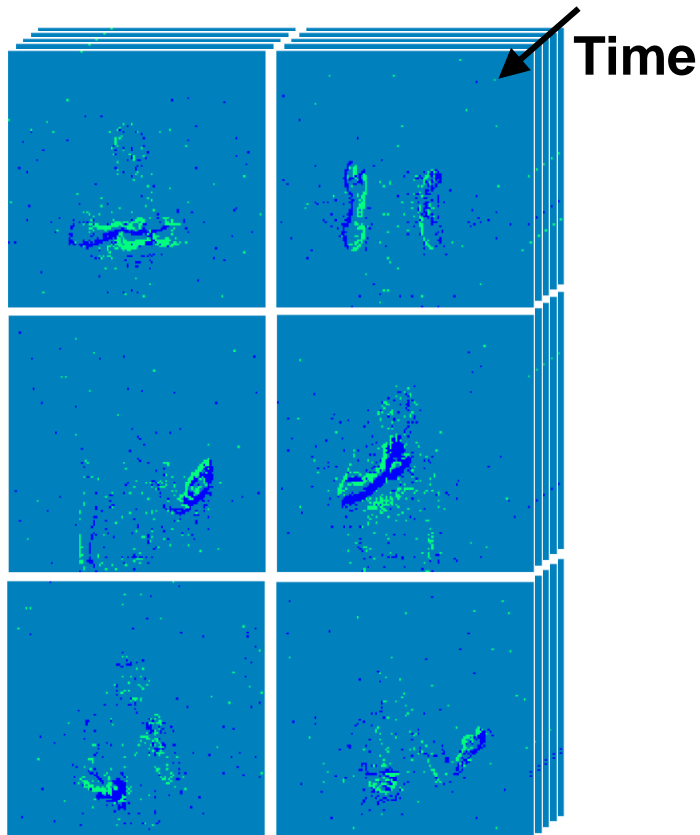


- Event-driven / sparsity-aware computation
- Sensor-agnostic raw-data processing
- **Task-agnostic processing and learning**



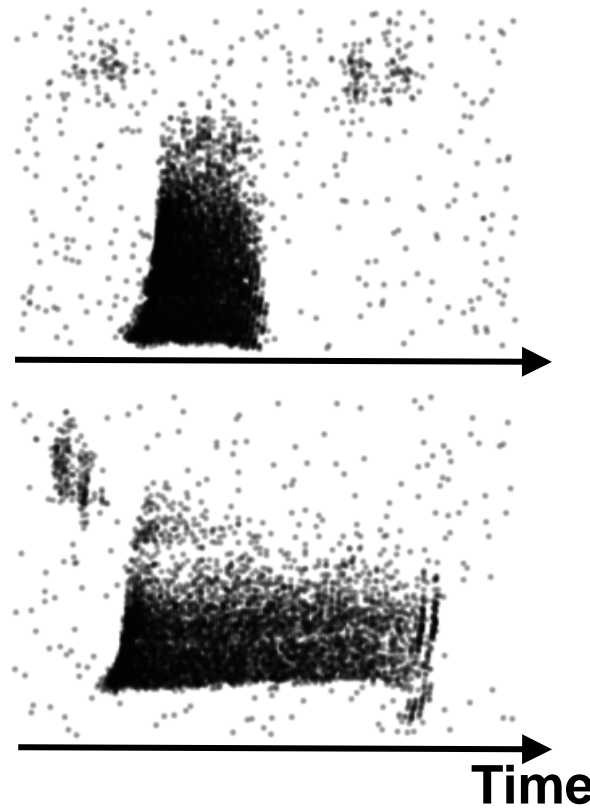
The ReckOn neuromorphic chip – Benchmarking

 **Vision**
Gesture recognition
(DVS Gestures dataset)




Accuracy: 87.3% (28 μ W @0.5V)

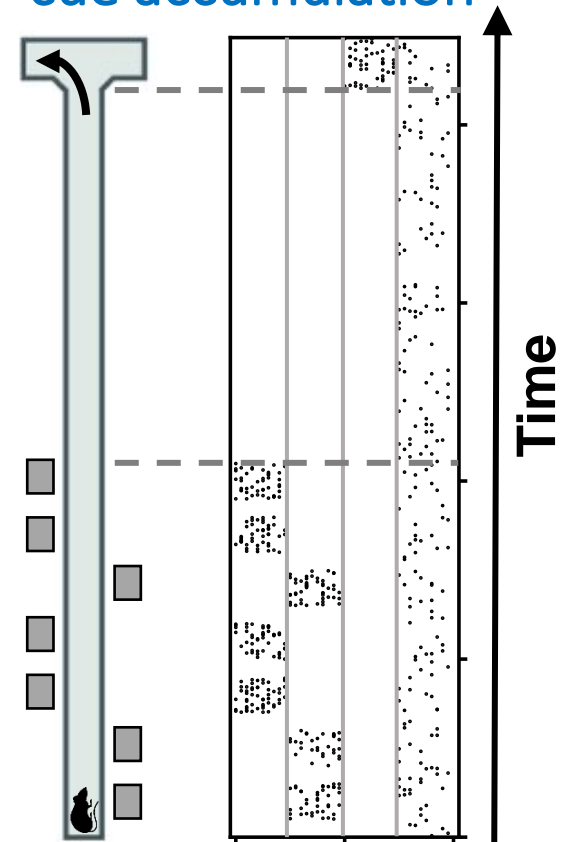
 **Audition**
Keyword spotting
(Spiking Heidelberg Digits dataset)



Accuracy: 90.7% (46 μ W @0.5V)

[Frenkel, *ISSCC*, 2022]

 **Navigation**
Delayed-supervision
cue accumulation



Accuracy: 96.4% (14 μ W @0.5V)

What you should remember

Key elements for a competitive advantage with neuromorphic edge intelligence

Merging AI, neuroscience and hardware is key to

achieve **end-to-end on-chip learning over second-long timescales** while keeping a **milli-second temporal resolution**, a **yet unsolved challenge**,

provide a low-cost solution: **0.45-mm²** core area,

<50μW for real-time training **@0.5V**,

demonstrate **task-agnostic learning** with a spike-based encoding toward user customization and chip repurposing at the edge.

This outlines an exciting future for neuromorphic edge intelligence!

...But wait, it's not over!

**Time for
yet-unpublished stuff!**

The *Cognitive Sensor Nodes and Systems* (CogSys) Team

We bridge the bottom-up (bio-inspired) and top-down (engineering-driven) design approaches toward neuromorphic intelligence.



Questions?



@C_Frenkel



cfrenkel



ChFrenkel



Charlotte-Frenkel



c.frenkel@tudelft.nl



chfrenkel.github.io

Main references:

- ODIN: [C. Frenkel et al., “A 0.086-mm² 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28nm CMOS,” *IEEE Trans. BioCAS*, 2019]
- MorphIC: [C. Frenkel et al. “MorphIC: A 65-nm 738k-synapse/mm² quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning,” *IEEE Trans. BioCAS*, 2019]
- DRTP: [C. Frenkel, M. Lefebvre et al., “Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks,” *Frontiers in Neuroscience*, 2021]
- SPOON: [C. Frenkel et al., “A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas,” *IEEE ISCAS*, 2020]
- **Review:** [C. Frenkel, D. Bol and G. Indiveri, “Bottom-up and top-down approaches for the design of neuromorphic processing systems: Tradeoffs and synergies between natural and artificial intelligence,” *Proceedings of the IEEE*, 2023]
- **ReckOn:** [C. Frenkel and G. Indiveri, “ReckOn: A 28-nm Sub-mm² Task-Agnostic Spiking Recurrent Neural Network Processor Enabling On-Chip Learning over Second-Long Timescales,” *IEEE International Solid-State Circuits Conference (ISSCC)*, 2022]

Open-sourced!

github.com/ChFrenkel/ODIN

Open-sourced!

github.com/ChFrenkel/DirectRandomTargetProjection

Open-sourced!

github.com/ChFrenkel/ReckOn