Merging insights from artificial and biological neural networks

Does neuromorphic edge intelligence need spikes?

Charlotte Frenkel
(c.frenkel@tudelft.nl)

Assistant Professor
Dept. of Microelectronics, Delft University of Technology

NICE 2025, March 27th

Outline

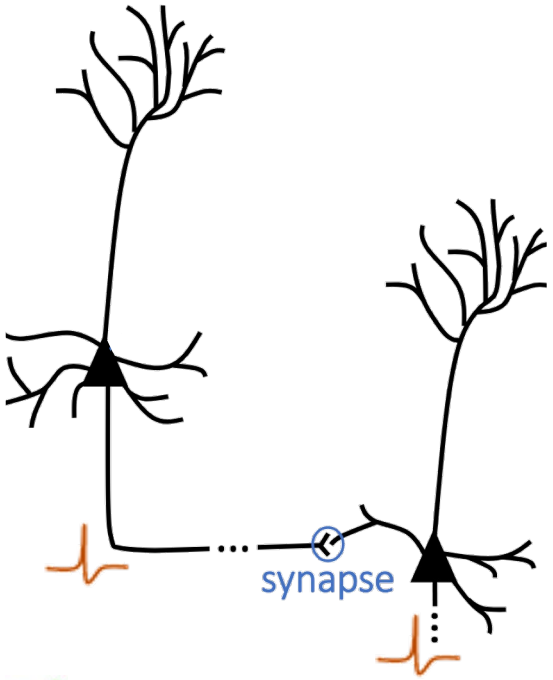
① What are the key synergies between the bottom-up and top-down approaches?

② How can we exploit these synergies for novel spike-based engineering solutions?

Synaptic plasticity rules – Neuroscience as the starting point

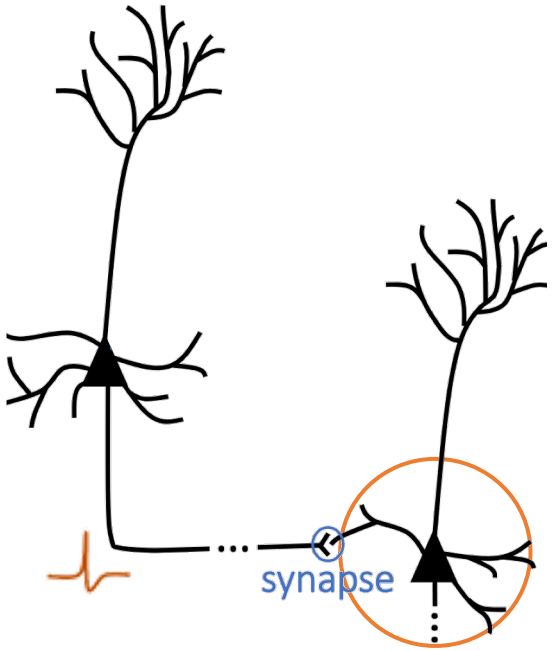
AI algorithms ↑ Neuroscience

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]

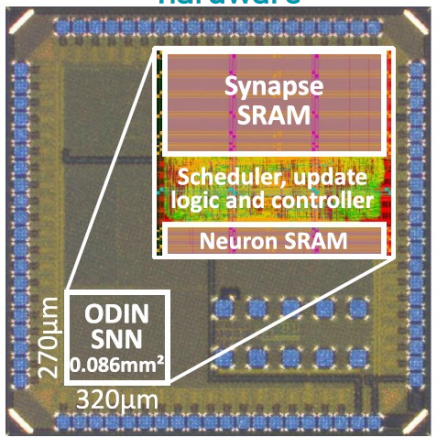


✔ **Local** in space
✔ **Local** in time

Huge savings in silicon
…but hard to exploit.

ODIN

open source hardware

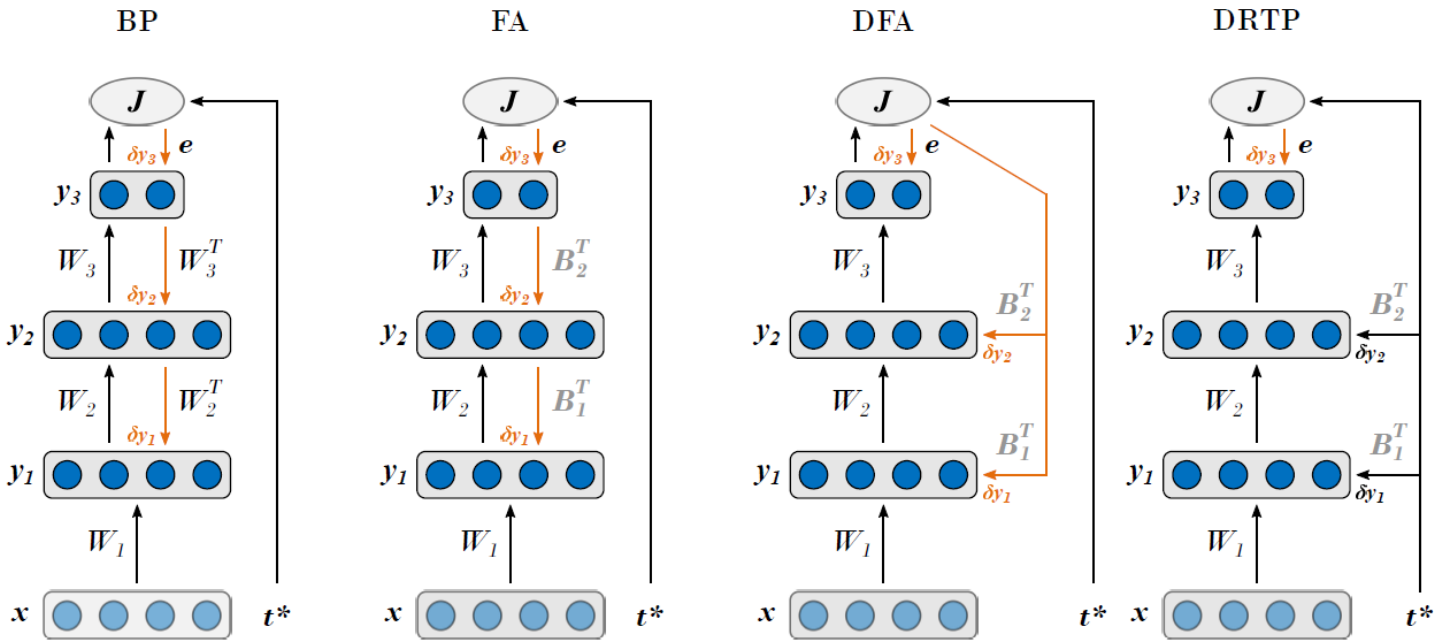


[Frenkel, TBioCAS, 2019]

Neural network training – Bio-plausibility as the end goal

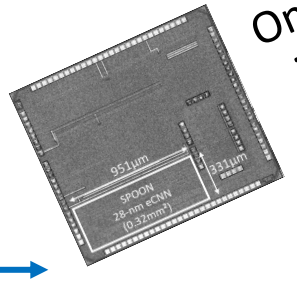
Synergy with hardware: latency, memory access patterns

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	×	×	×	✓

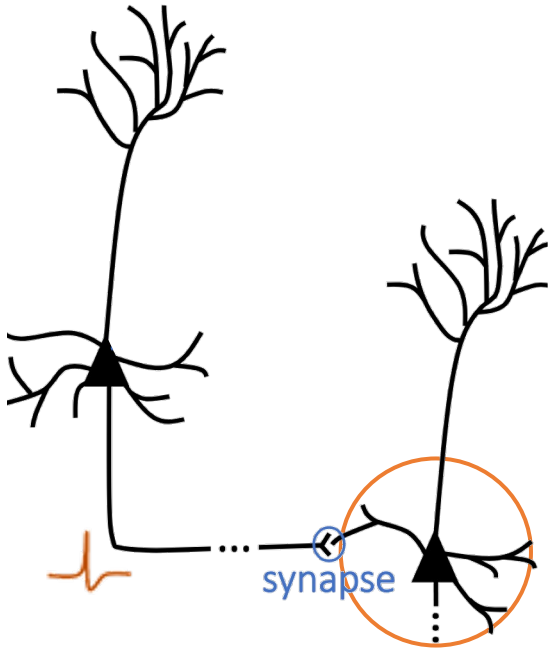


Only ~15% overhead in power and area [Frenkel, *ISCAS'20*]

↘ Computational and memory cost ↘

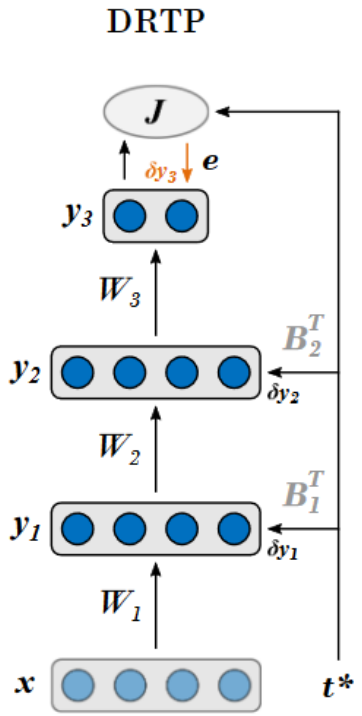
[Lillicrap, *Nat. Comms.*, 2016] [Nokland, *NeurIPS*, 2016] [Frenkel & Lefebvre, *Front. Neur.*, 2020]

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



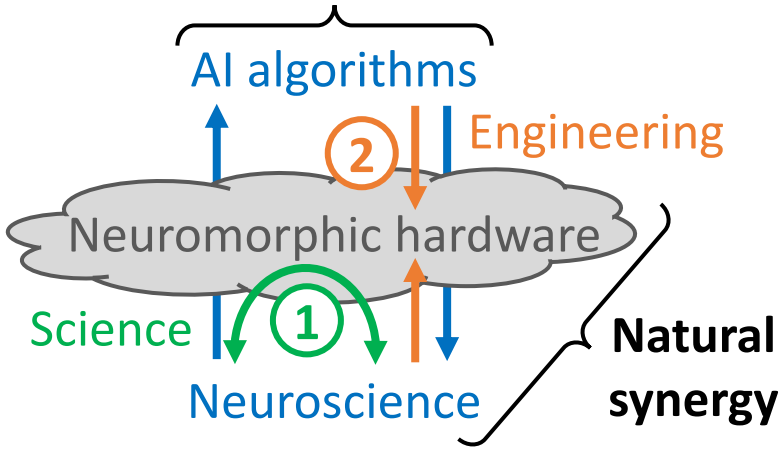
Designing efficient hardware hints toward bio-plausible mechanisms

Bringing AI closer to neuroscience leads to hardware efficiency



On our way to neuromorphic intelligence – Bottom-up or top-down?

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 What are the key synergies between the bottom-up and top-down approaches?

② How can we exploit these synergies for novel spike-based engineering solutions?

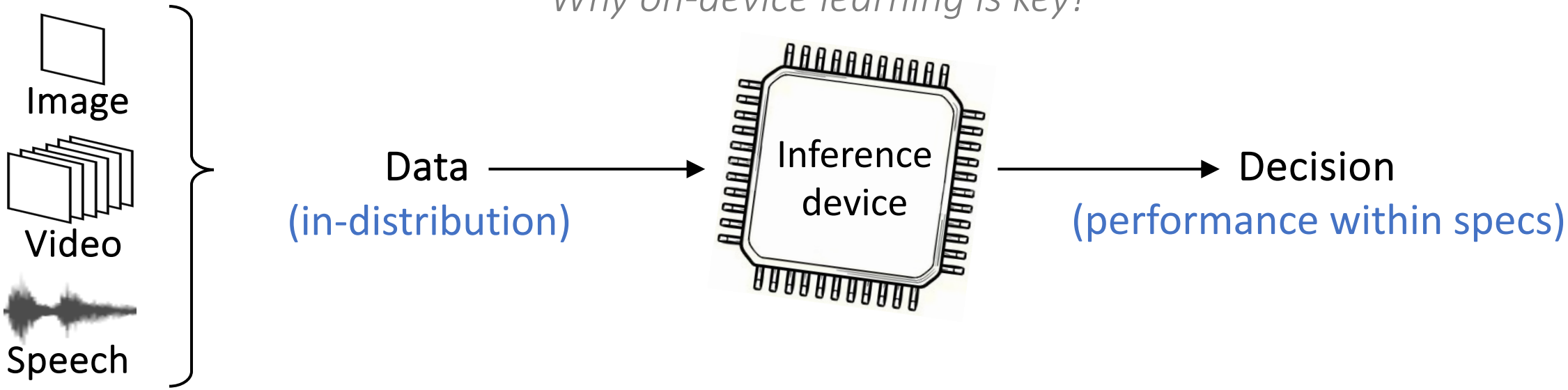
Let's use a 4-step recipe!

Neuromorphic intelligence:

② should be fed by ①

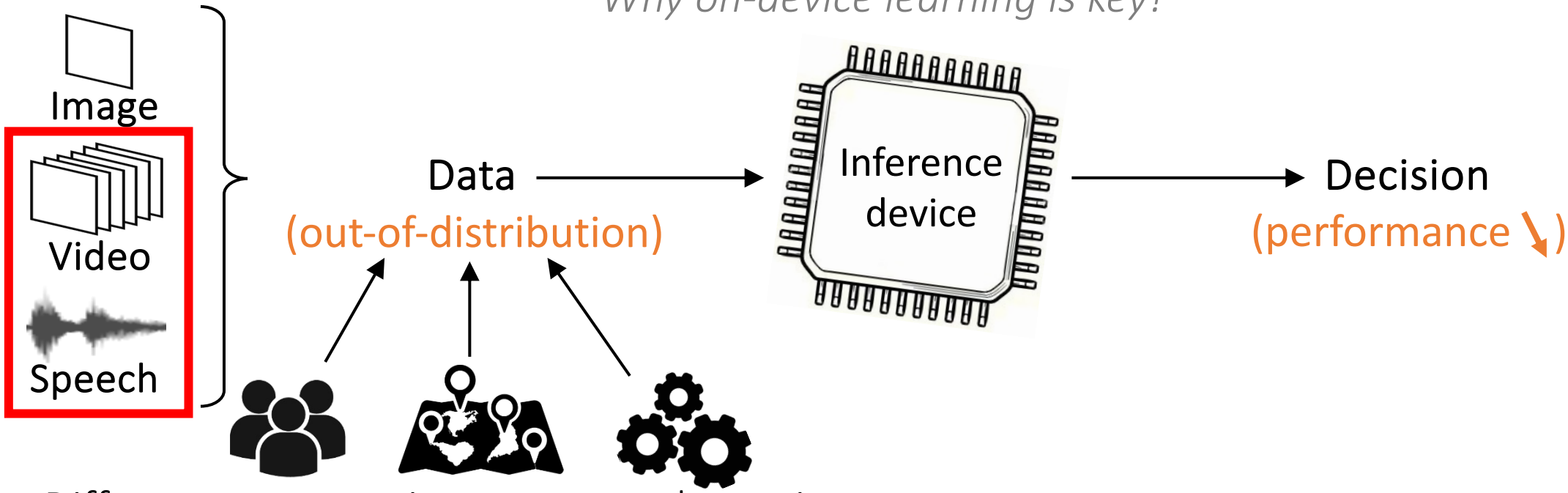
1) Pick the use case

Why on-device learning is key!



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

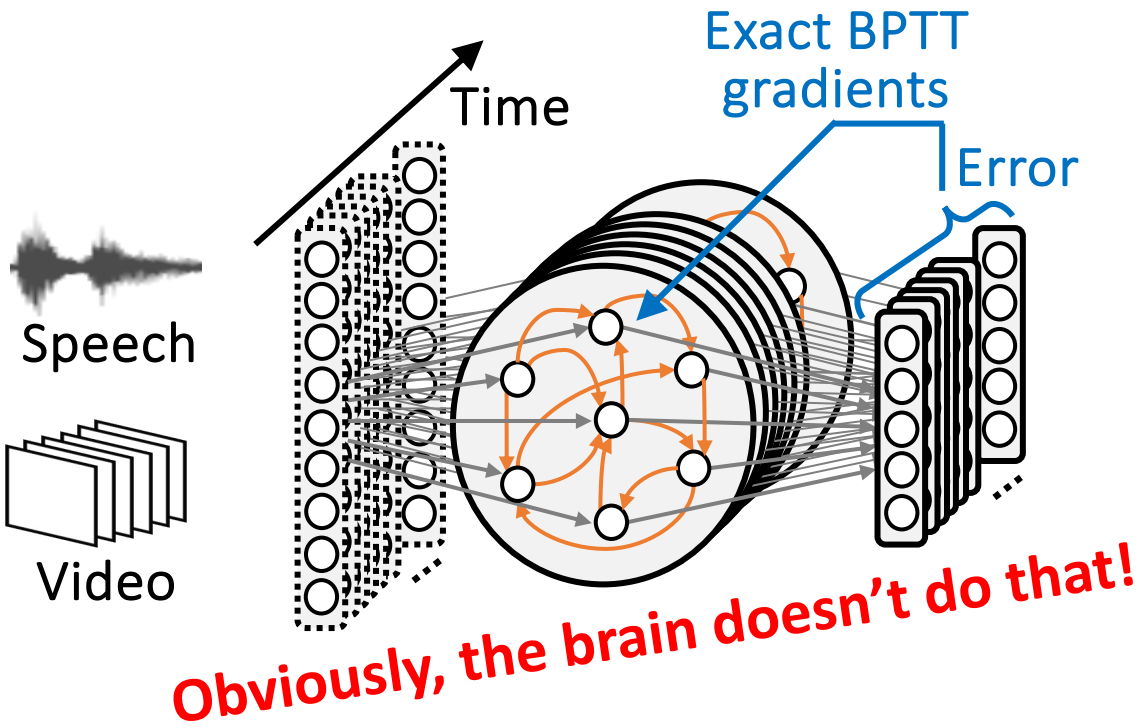
Data exchange with the cloud?

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!



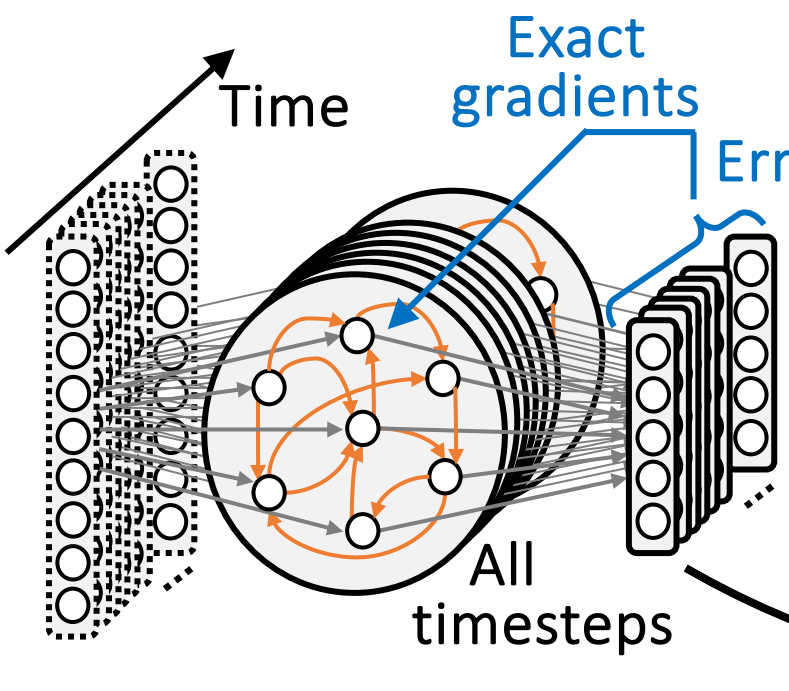
- Unrolling in time: very deep network
  (current learning ICs for static stimuli: ≤3 layers)
- Intractable memory/latency requirements
- No end-to-end on-chip solution to date

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)

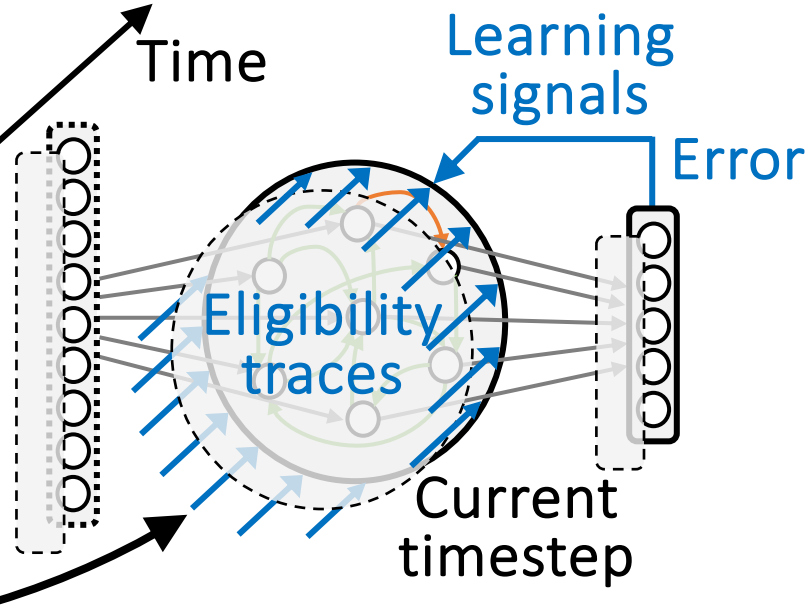


$$\frac{dE}{dW_{ij}} \simeq \sum_t L_j^t \, e_{ji}^t$$

e-prop

Eligibility propagation (e-prop, forward)

[Bellec, *Nat. Comms*'20]

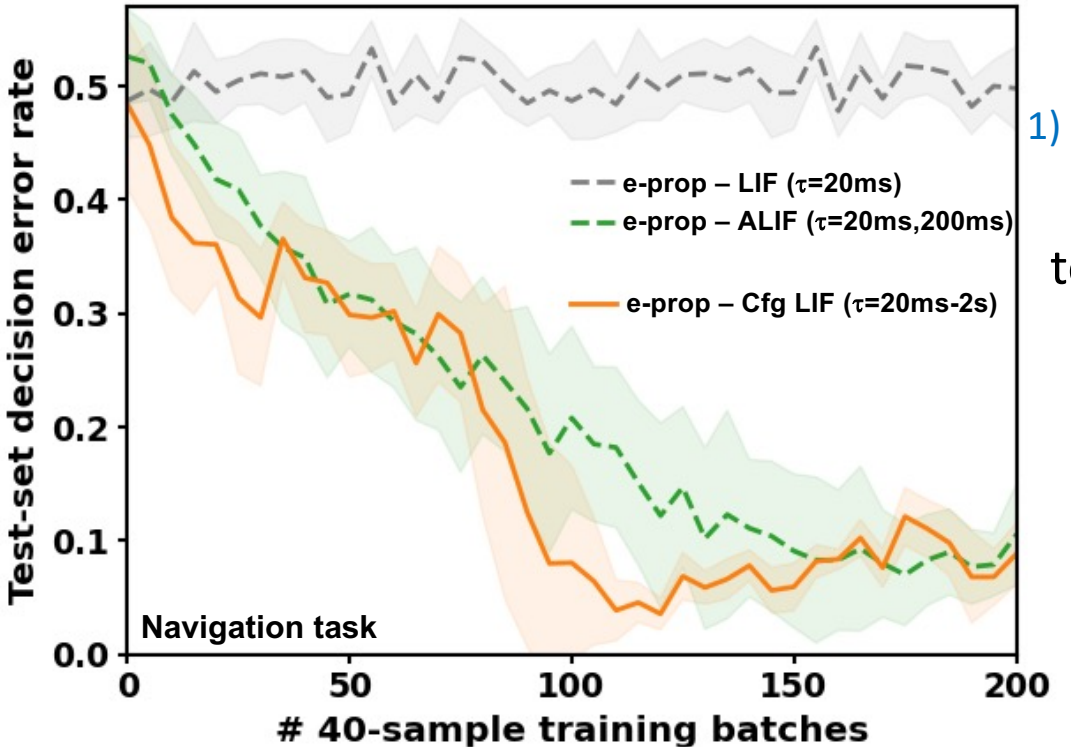


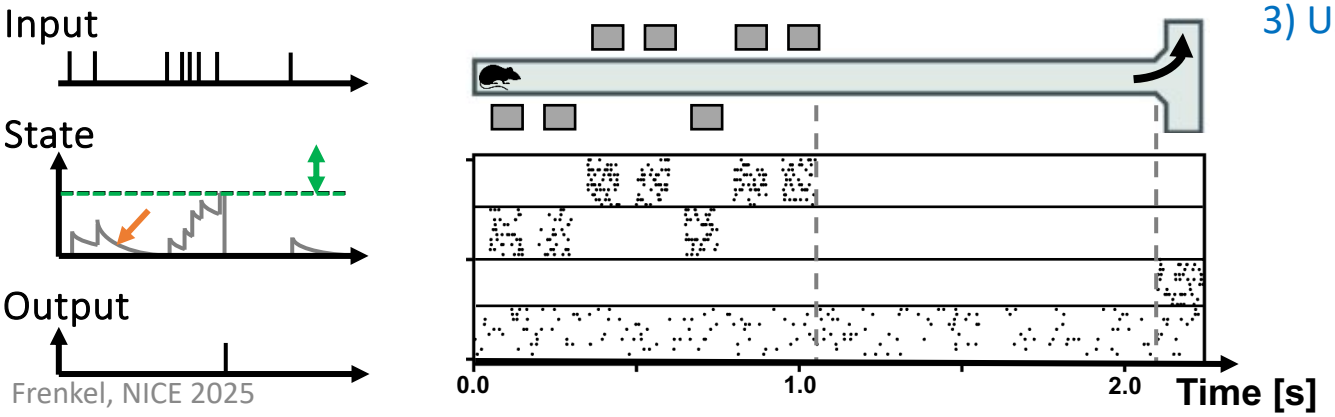
Key concept: space and time locality (again!)

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 → BPTT

HW tractability?
Bio plausibility?

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

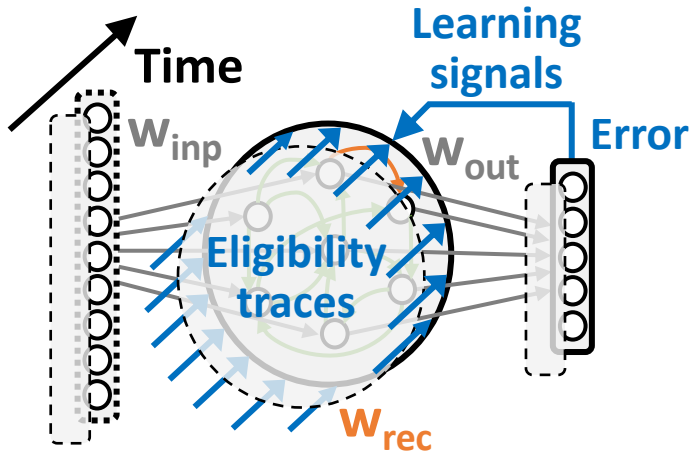
Simplified e-prop

4) Enforce space and time locality

Key steps to minimize memory requirements

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

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



Local decoupling of space and time:

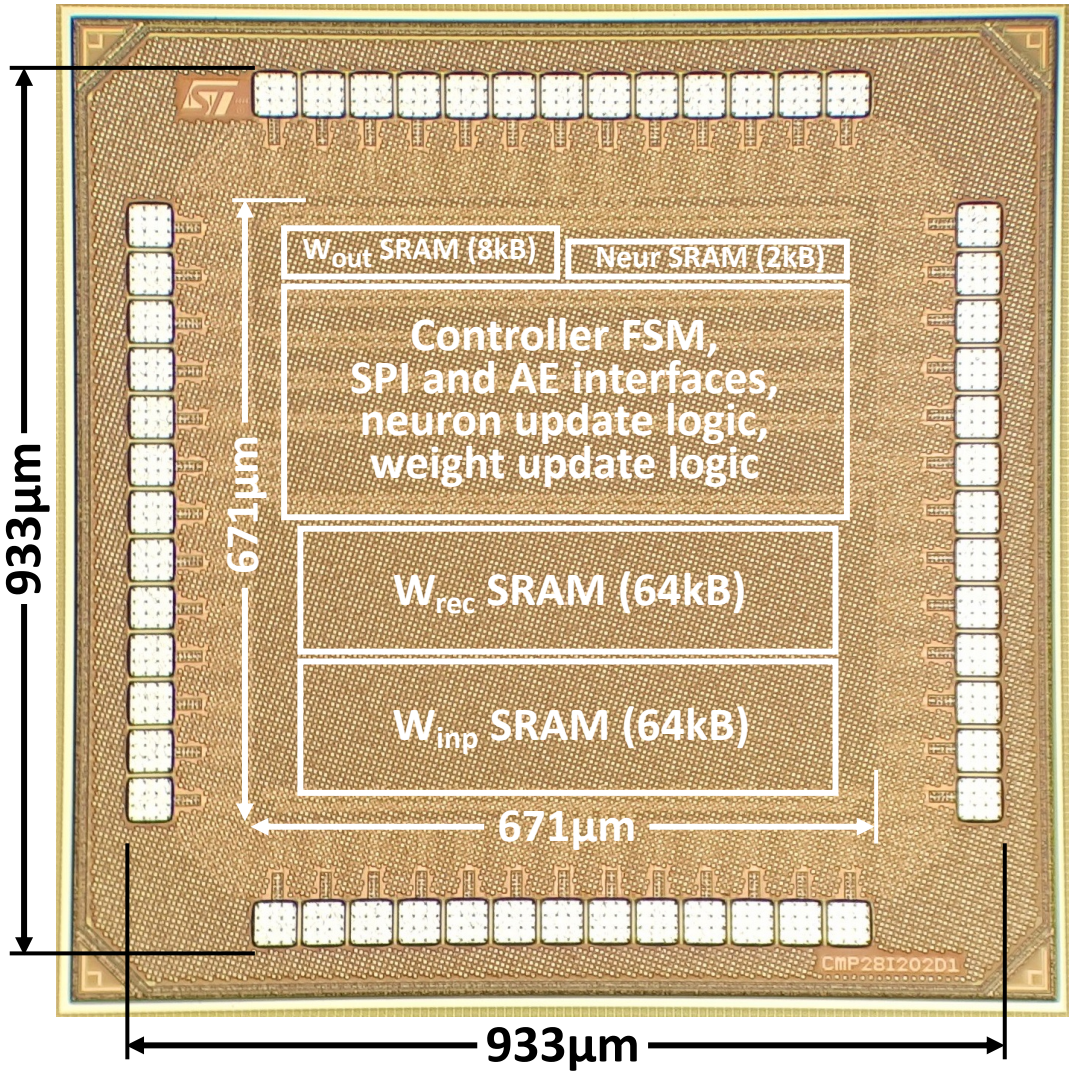
- **Pre-synaptic term:** activity low-pass filtering
- **Post-synaptic term:** surrogate derivative of the spiking activation function

Scales with #neurons in O(N)

Memory overhead of only 1%!

Stochastic weight updates allow reducing weight resolution to 8 bits

The ReckOn neuromorphic chip – Microphotograph and summary

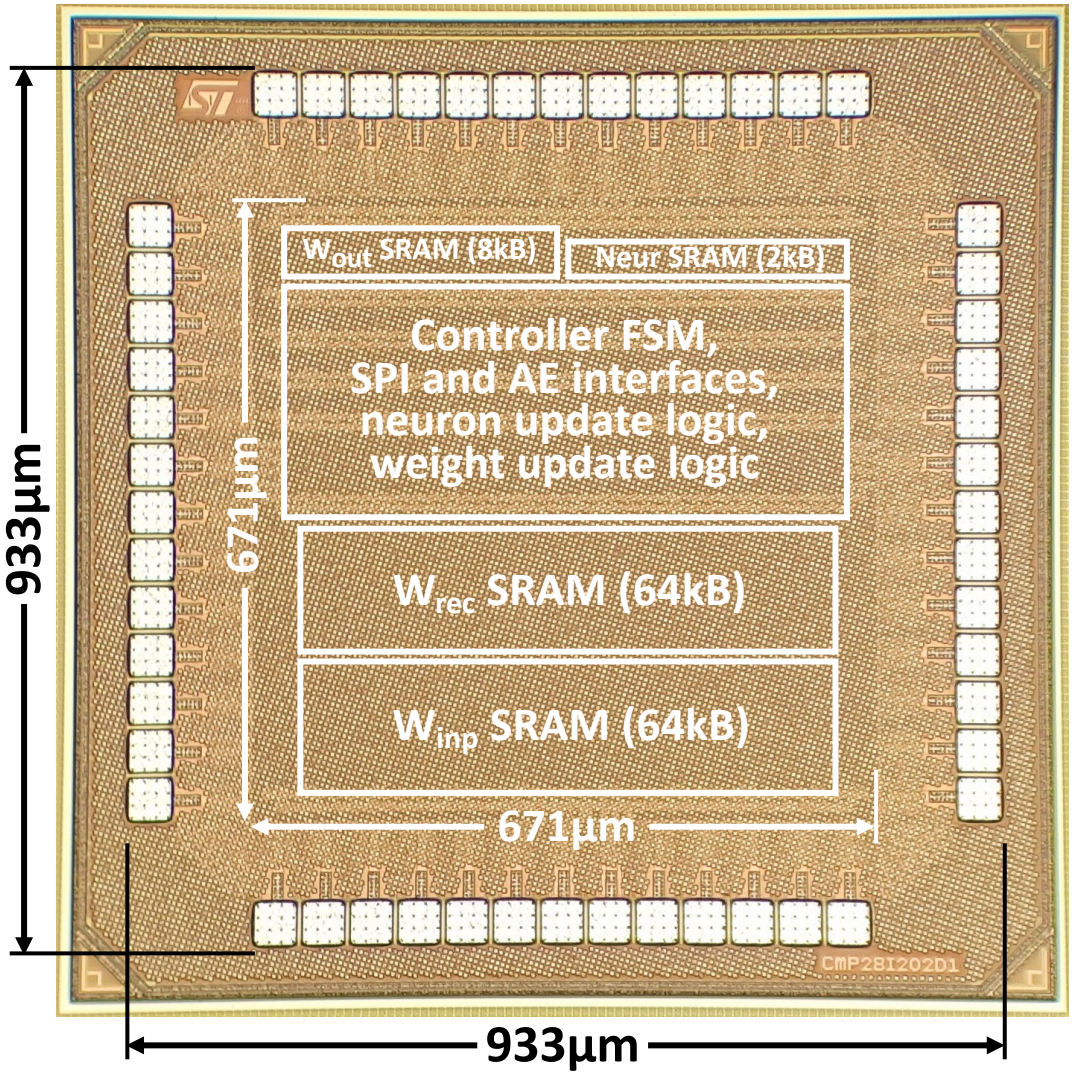


Technology	28nm FDSOI CMOS
Core size	0.67 x 0.67 mm² **0.45mm²**
Die size	0.93 x 0.93 mm²
SRAM	138kB **+ 0kB ext. DRAM!**
Network	Spiking RNN
Training timespan	Max. 32k steps

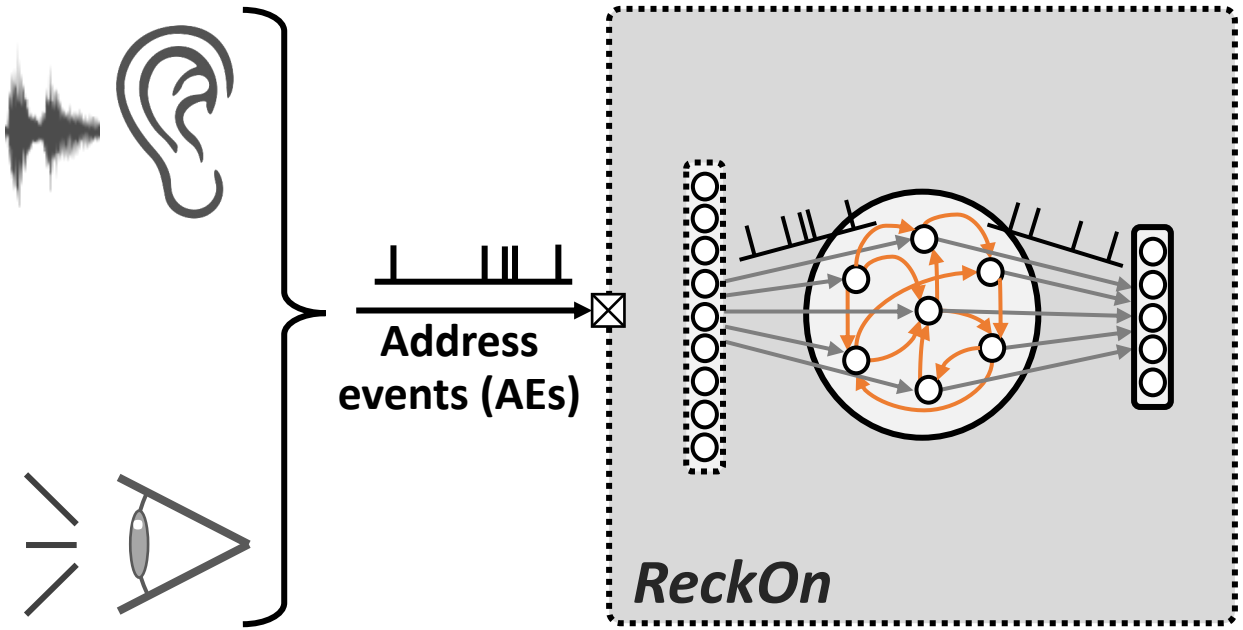
open source hardware

[Frenkel, *ISSCC*, 2022]

The ReckOn neuromorphic chip – Key advantage of using spikes



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

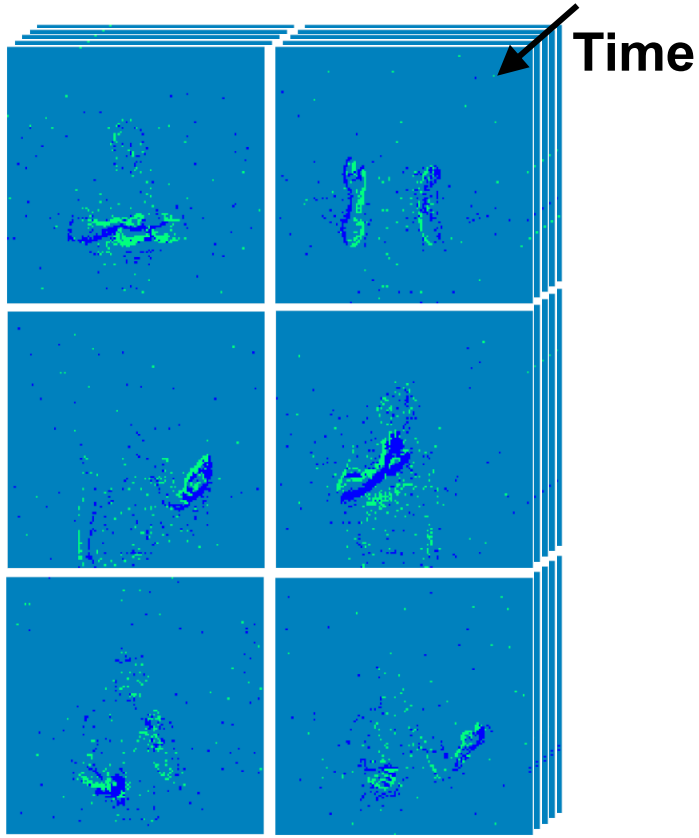


[Frenkel, *ISSCC*, 2022]

The ReckOn neuromorphic chip – Benchmarking

Vision

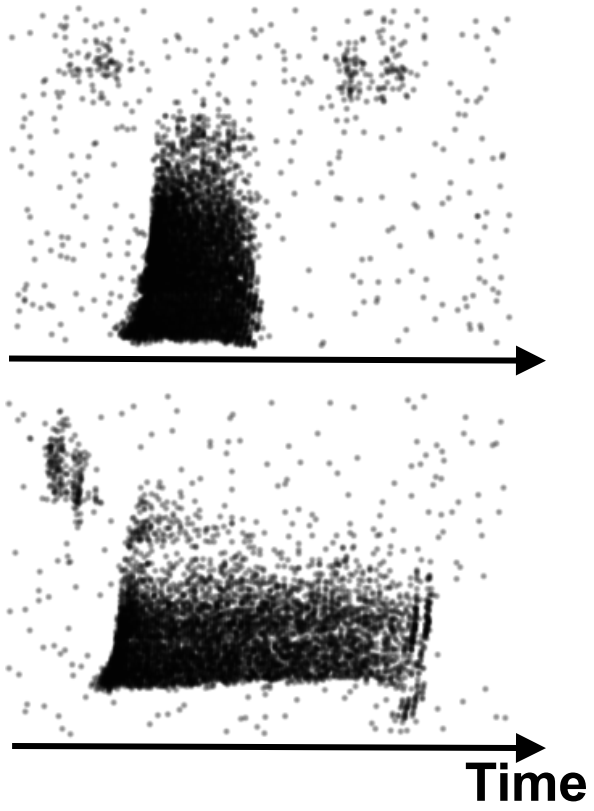
Gesture recognition
(DVS Gestures dataset)



Accuracy: 87.3% (28µW @0.5V)

Audition

Keyword spotting
(Spiking Heidelberg Digits dataset)

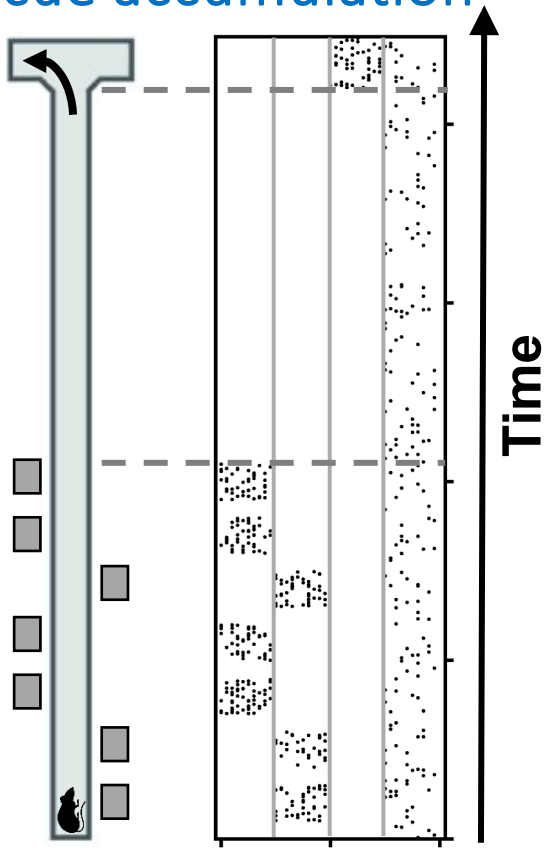


Accuracy: 90.7% (46µW @0.5V)

[Frenkel, *ISSCC*, 2022]

Navigation

Delayed-supervision
cue accumulation



Accuracy: 96.4% (14µW @0.5V)

Outline

① What are the key synergies between the bottom-up and top-down approaches?

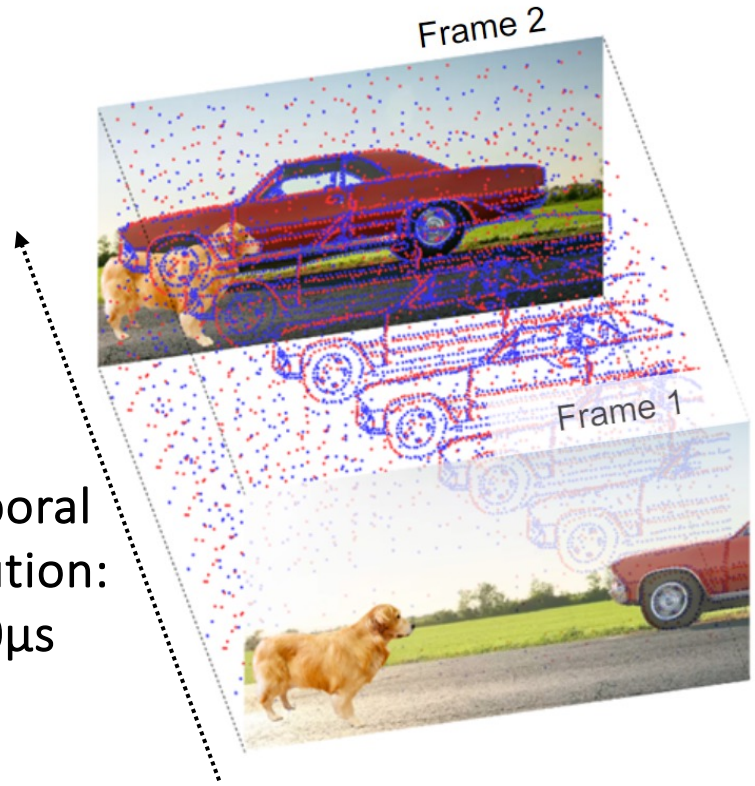
② How can we exploit these synergies for novel spike-based engineering solutions?

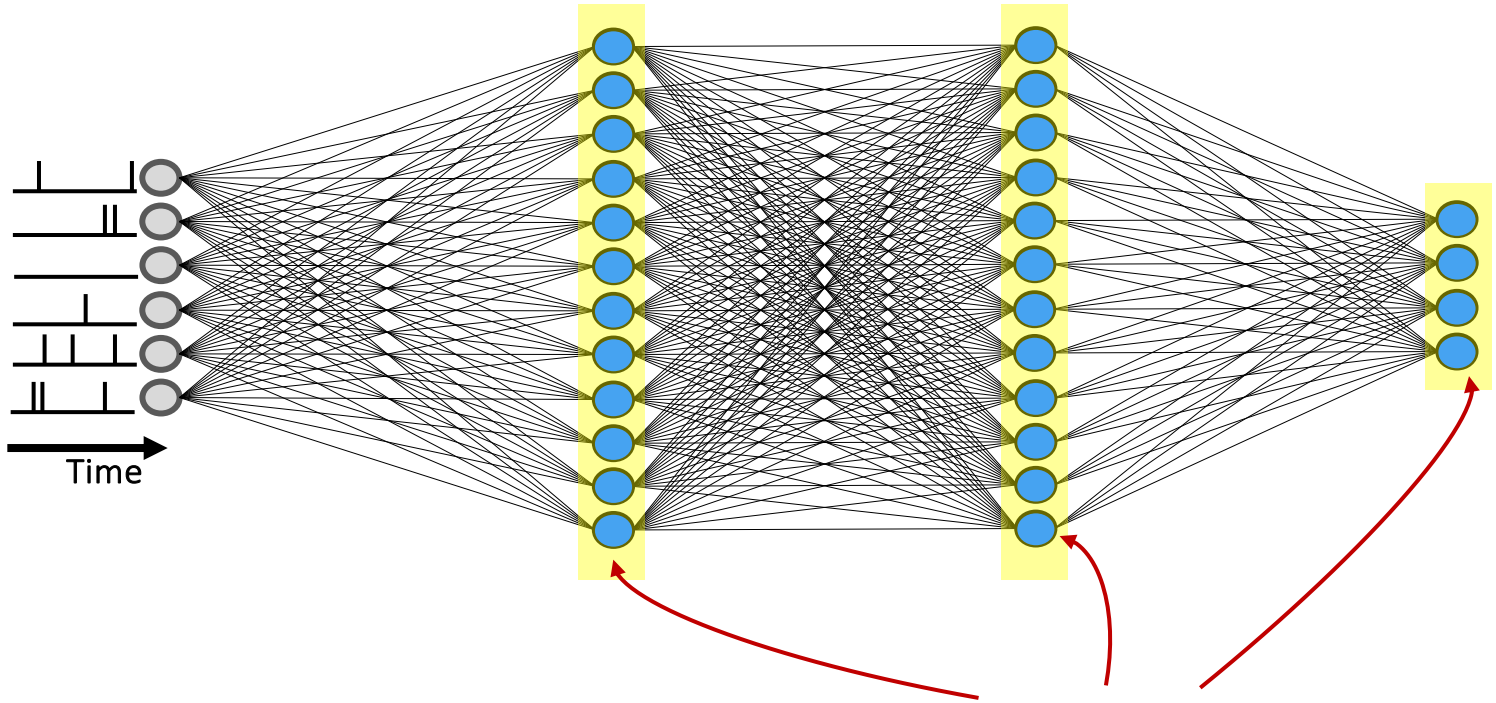
Let’s now look into spikes for low-latency applications!

Nicolas Chauvaux

[N. Chauvaux et al., ISCAS’25 (accepted)]
Extension preprints coming soon.

Spiking neural networks for low-latency event-driven computation



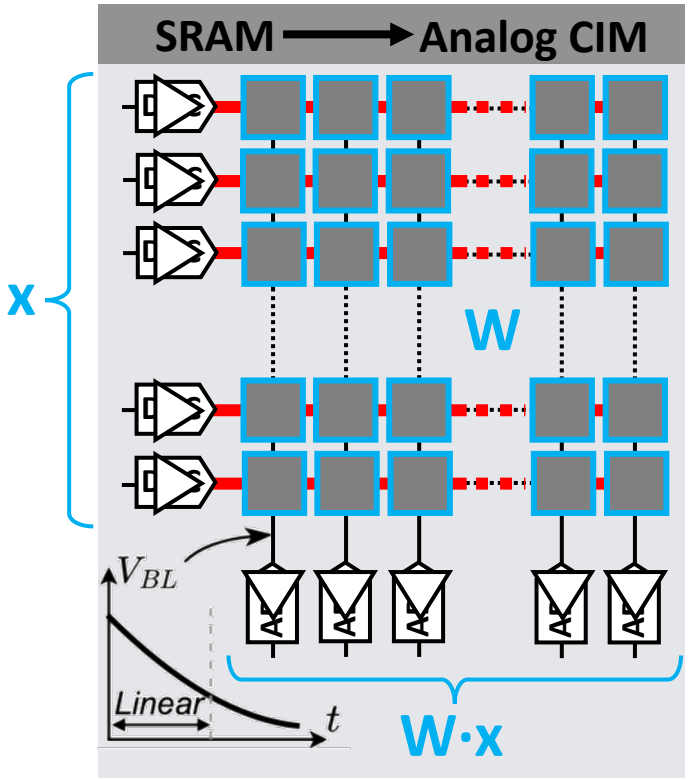


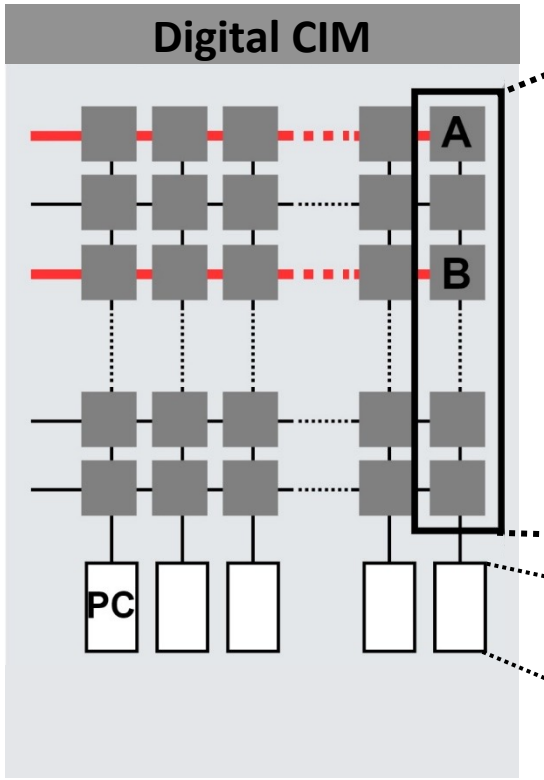
The state of all neurons has to be maintained at all times!

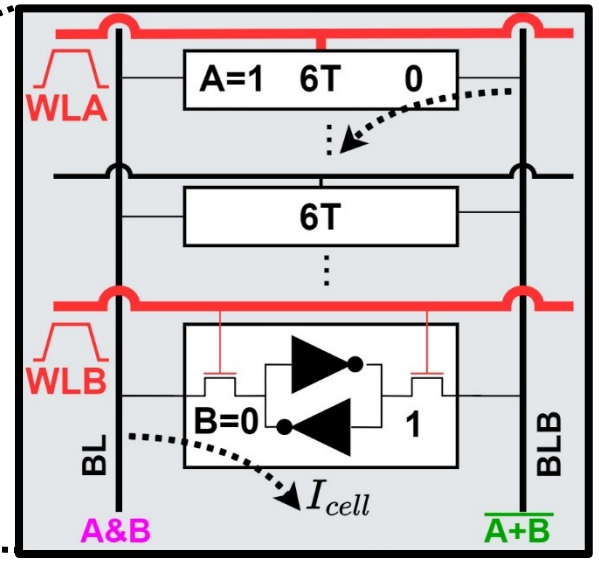
Spiking neural networks
have a memory overhead penalty.

A good case for bringing compute close to memory!

Compute in memory (CIM) to the rescue of spiking neural networks







Peripheral Circuit (PC)

$A \oplus B = \sim (A\&B + (\overline{A+B}))$

$S = A \oplus B \oplus C_{IN}$

$C_{OUT} = A\&B + C_{IN} \,\&\, A \oplus B$

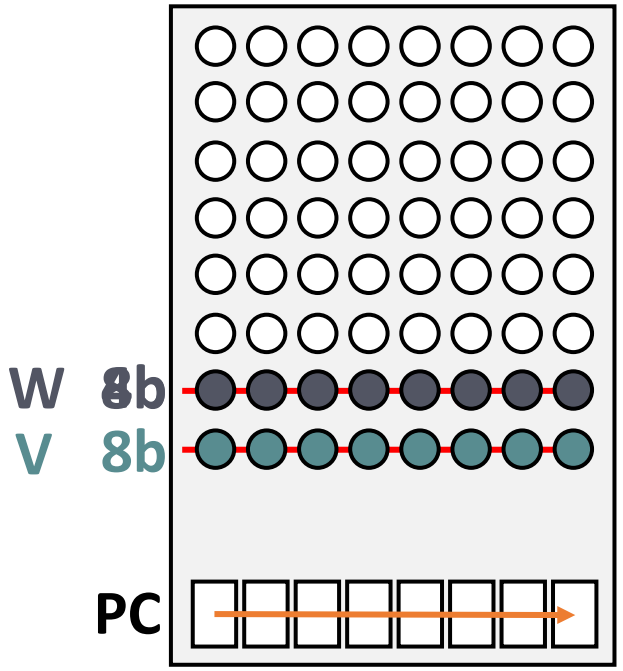
- ✔ Efficient MVM!
- ✖ Sparse inputs?
- ✖ Non-linearity, noise, mismatch

- ✖ Low parallelism…
- ✔ …but efficient event-driven addition and accumulation!
- ✔ Robustness

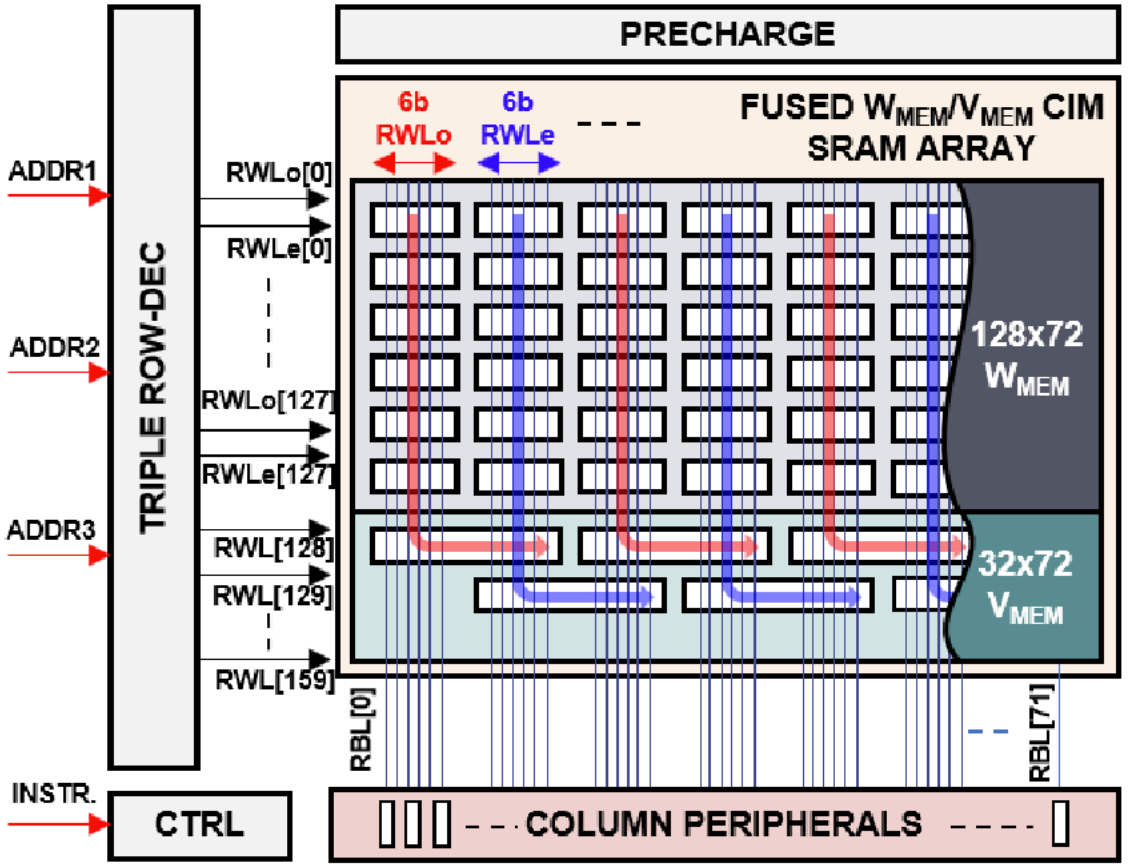
Current digital CIM approaches for SNNs have a flexibility issue

SoA: IMPULSE

[Agrawal, *SSC-L*, 2021]



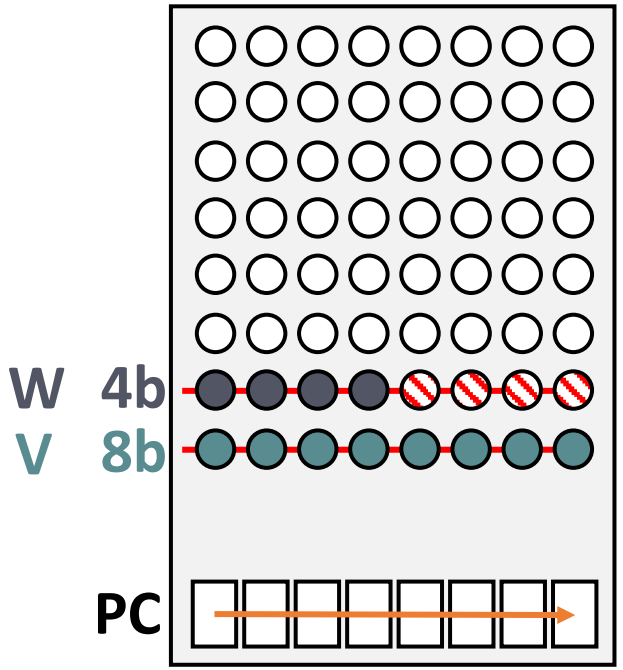
- Inefficient mapping
- Needs hardwiring



Unlocking *dataflow* and *resolution* flexibility in digital CIM for SNNs

SoA: IMPULSE

[Agrawal, *SSC-L*, 2021]

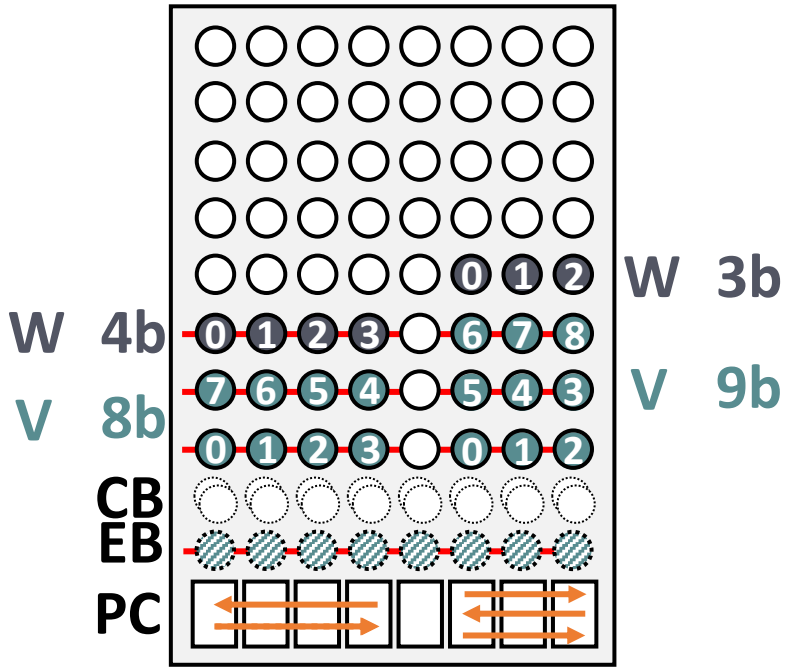


❌ Inefficient mapping

❌ Needs hardwiring

Solution:

FlexSpIM



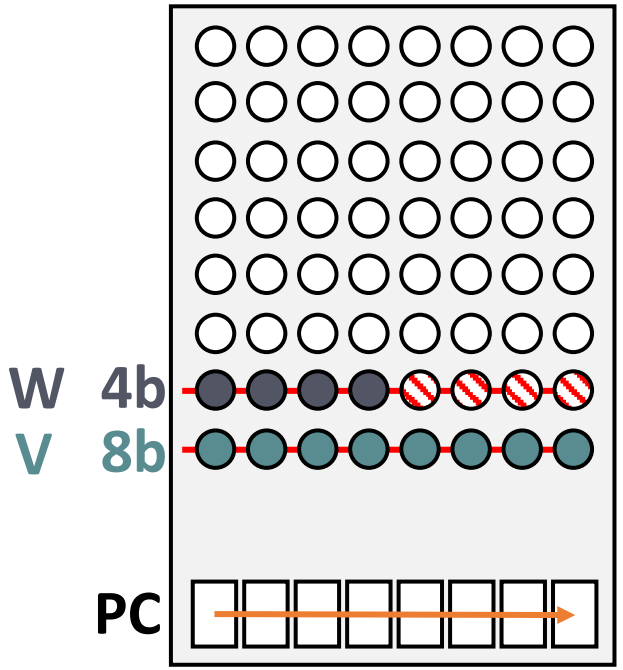
✔ Utilization can be maximized

✔ Flexible dataflow

Unlocking *dataflow* and *resolution* flexibility in digital CIM for SNNs

SoA: IMPULSE

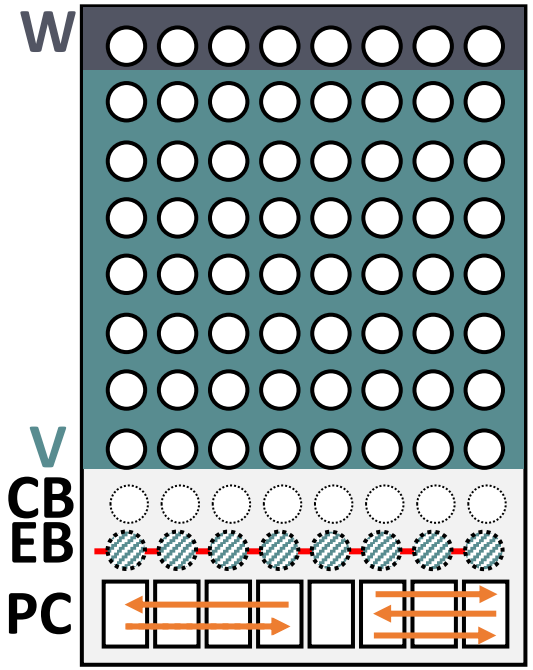
[Agrawal, *SSC-L*, 2021]



- ✗ Inefficient mapping
- ✗ Needs hardwiring

Solution:

FlexSpIM



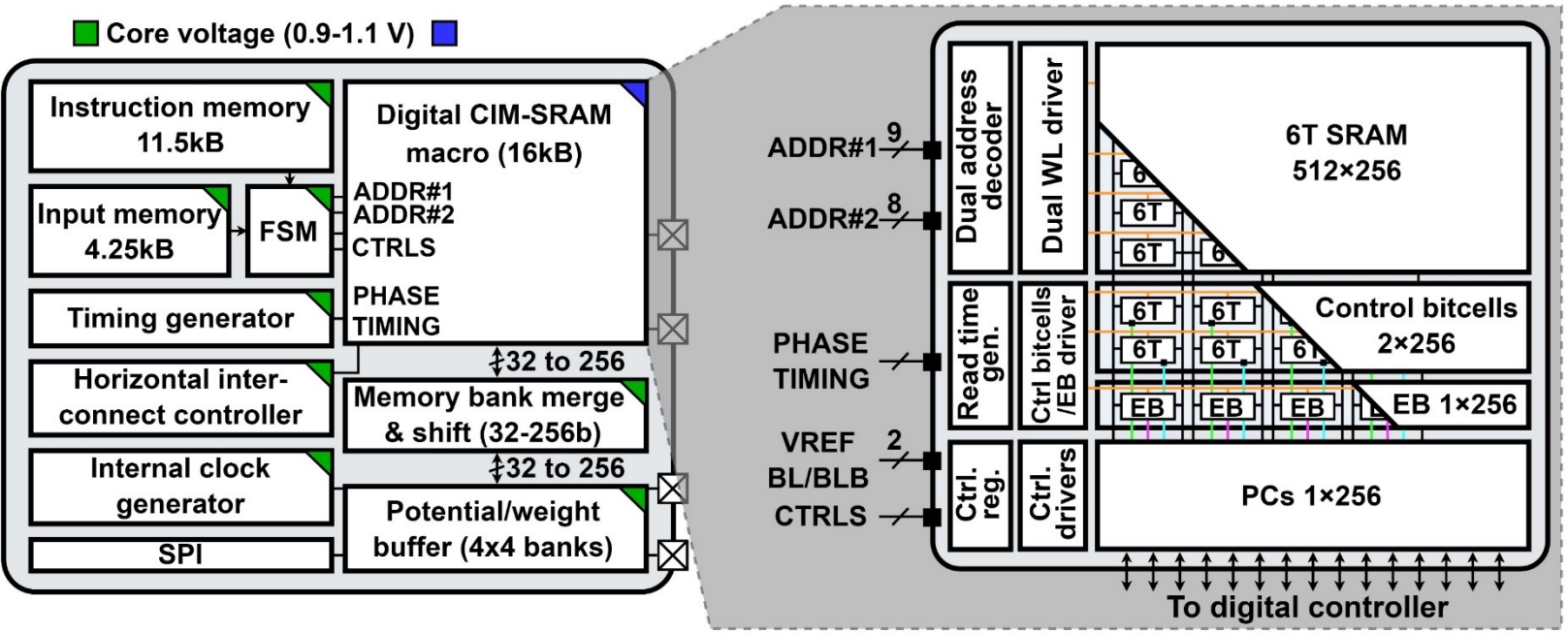
Weight stationarity

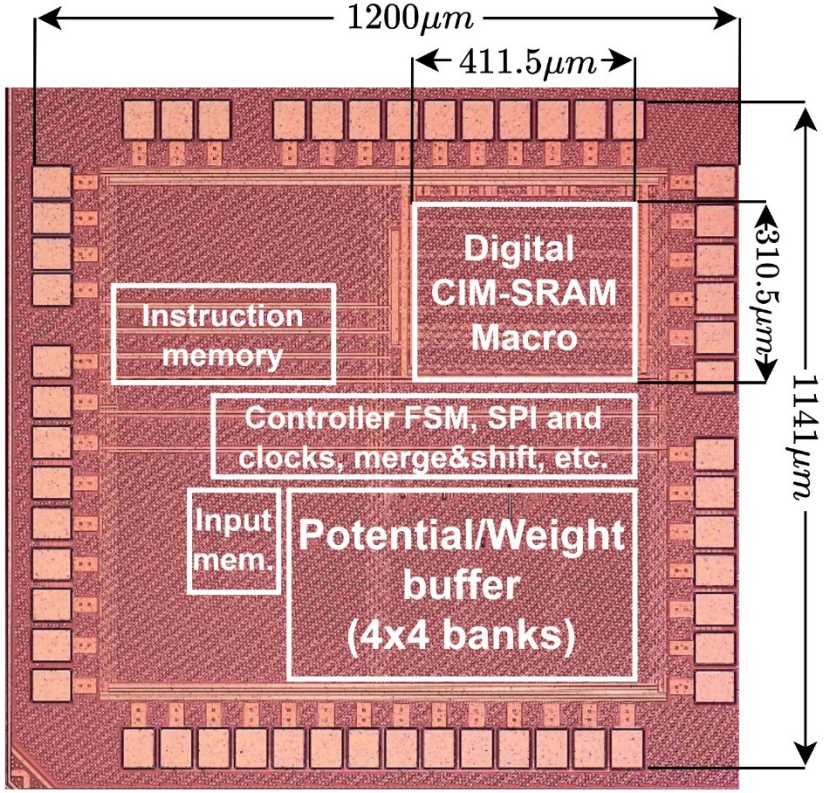
or

Output stationarity

- ✓ Utilization can be maximized
- ✓ Flexible dataflow

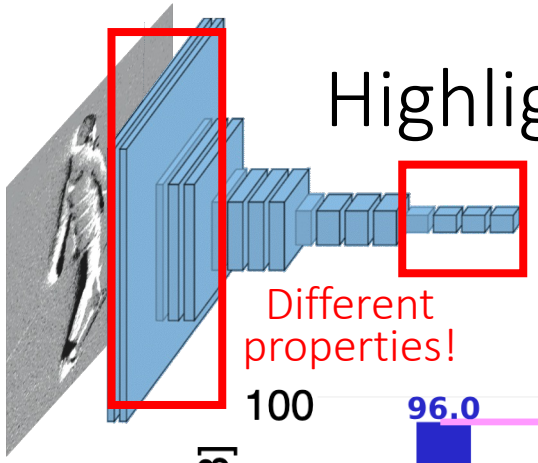
FlexSpIM – A flexible spiking in-memory macro

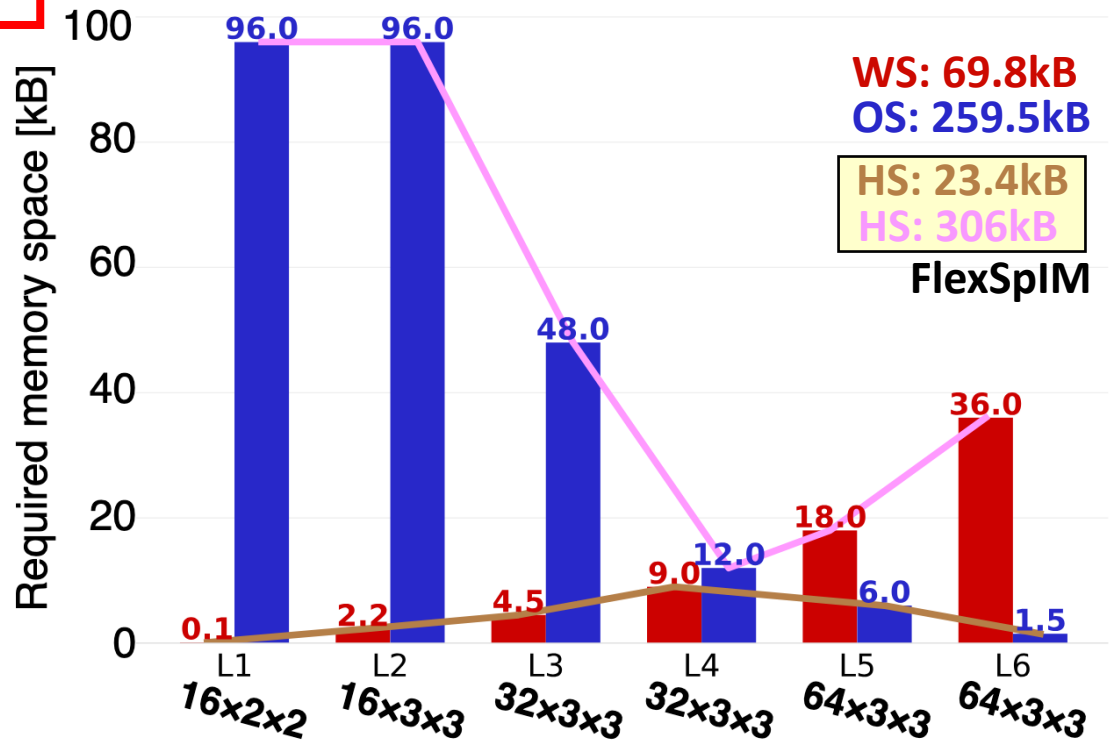


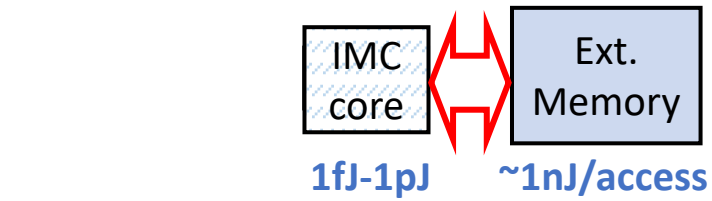


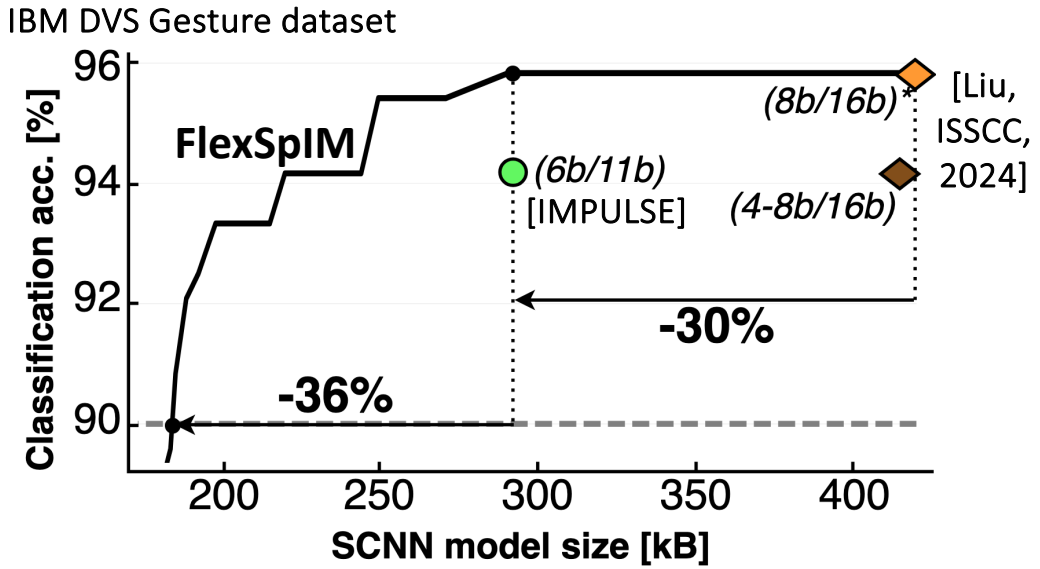
TSMC 40nm, proven *in silico!*

Highlights – CIM flexibility enables system-level savings









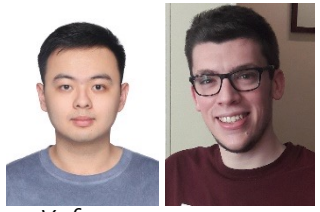
FlexSpIM can be tailored to the use case and target specifications!

Outline

① What are the key synergies between the bottom-up and top-down approaches?

② How can we exploit these synergies for novel spike-based engineering solutions?

Let's now look into spikes for low-latency applications!

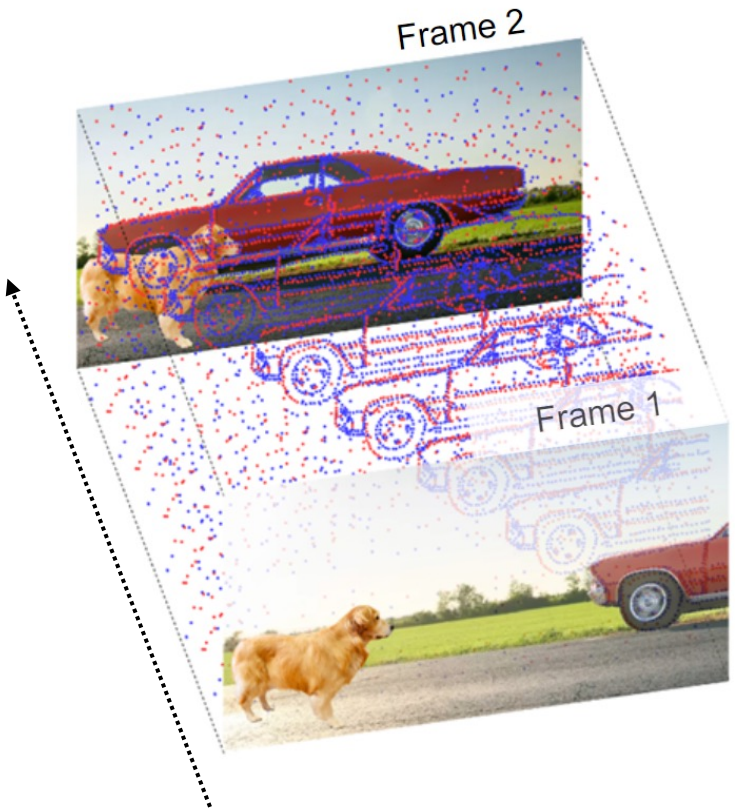


Yufeng Yang

Adrian Kneip

[Y. Yang, A. Kneip*, C. Frenkel, Trans. CAS AI, 2025]*
Open-source: github.com/cogsys-tudelft/evgnn

Graphs – A better representation for event-based data?



Data is only exchanged locally, within neighborhoods!

Graph Neural Network (GNN)

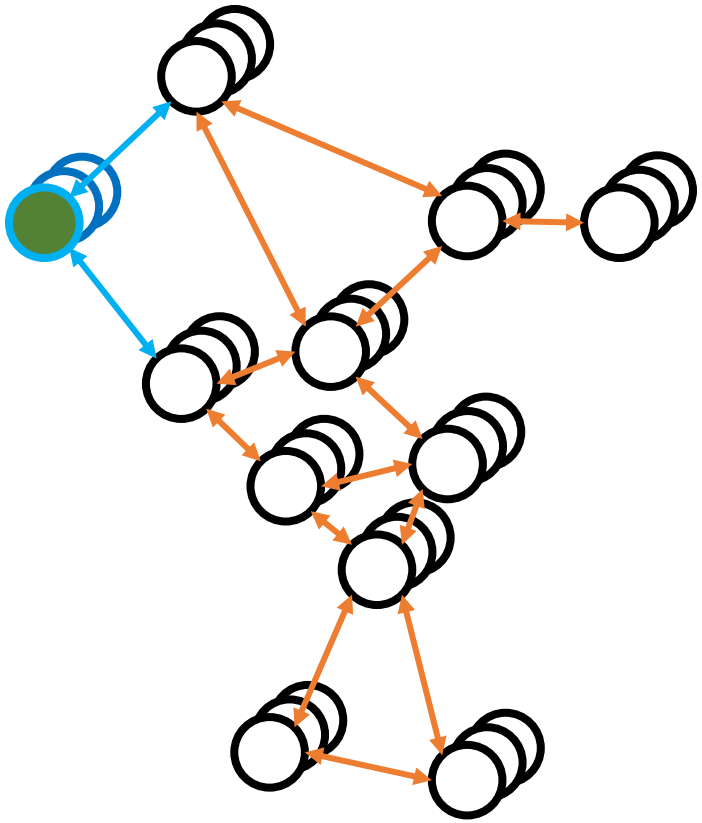
Each layer = message passing between nodes

$$\mathbf{x}_i^{l+1} = \gamma\left(\mathbf{x}_i^l, \bigoplus_{j\in\mathcal{N}(i)} \phi\left(\mathbf{x}_i^l, \mathbf{x}_j^l\right)\right)$$

Edges denote spatiotemporal neighbors

Nodes are events (x,y,polarity,timestamp)

From static graphs to dynamic graphs



<u>Challenge</u>: GNNs focus on static graphs, how we make them work for event-based data?

1) **Graph building**

2) **GNN execution**

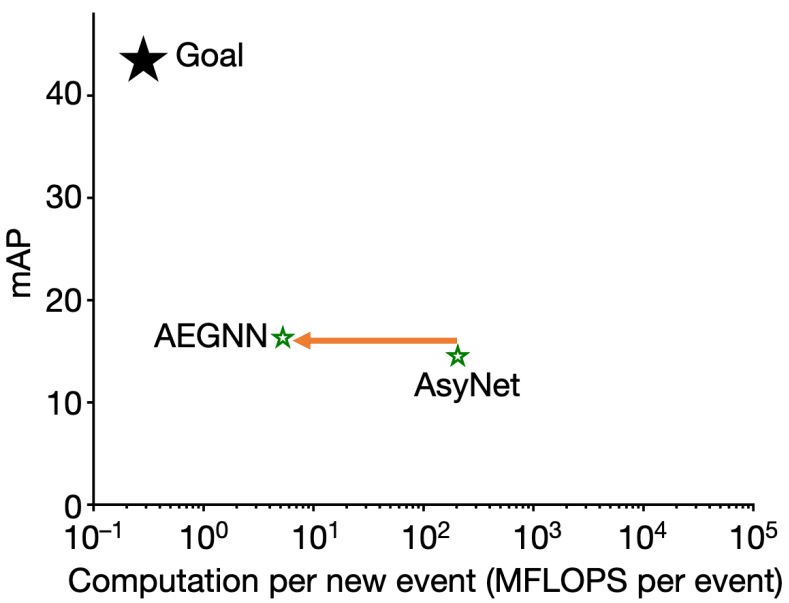
N layers = information can propagate through N hops

For each event, only the N-hop neighborhood needs to be updated

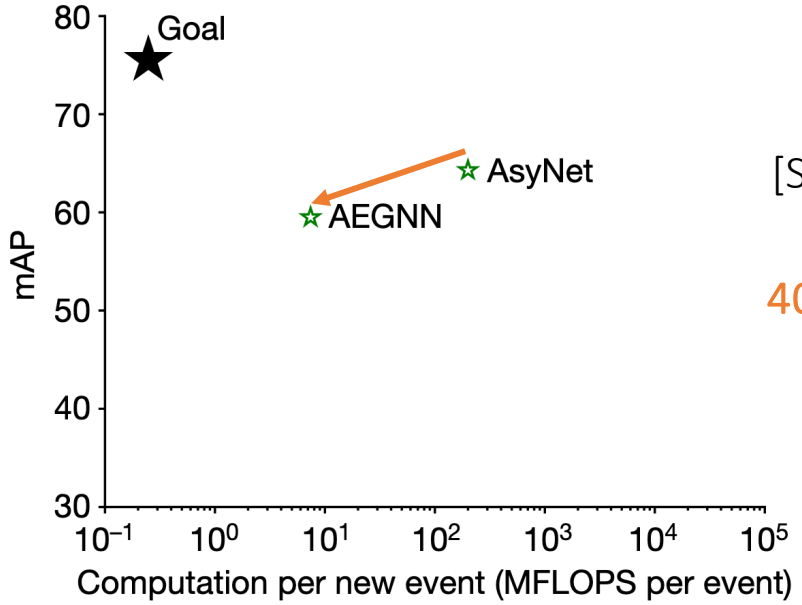
AEGNN [Schaefer, *CVPR*, 2022]

AEGNN promises low-latency execution through local processing!

GEN1 automotive detection dataset



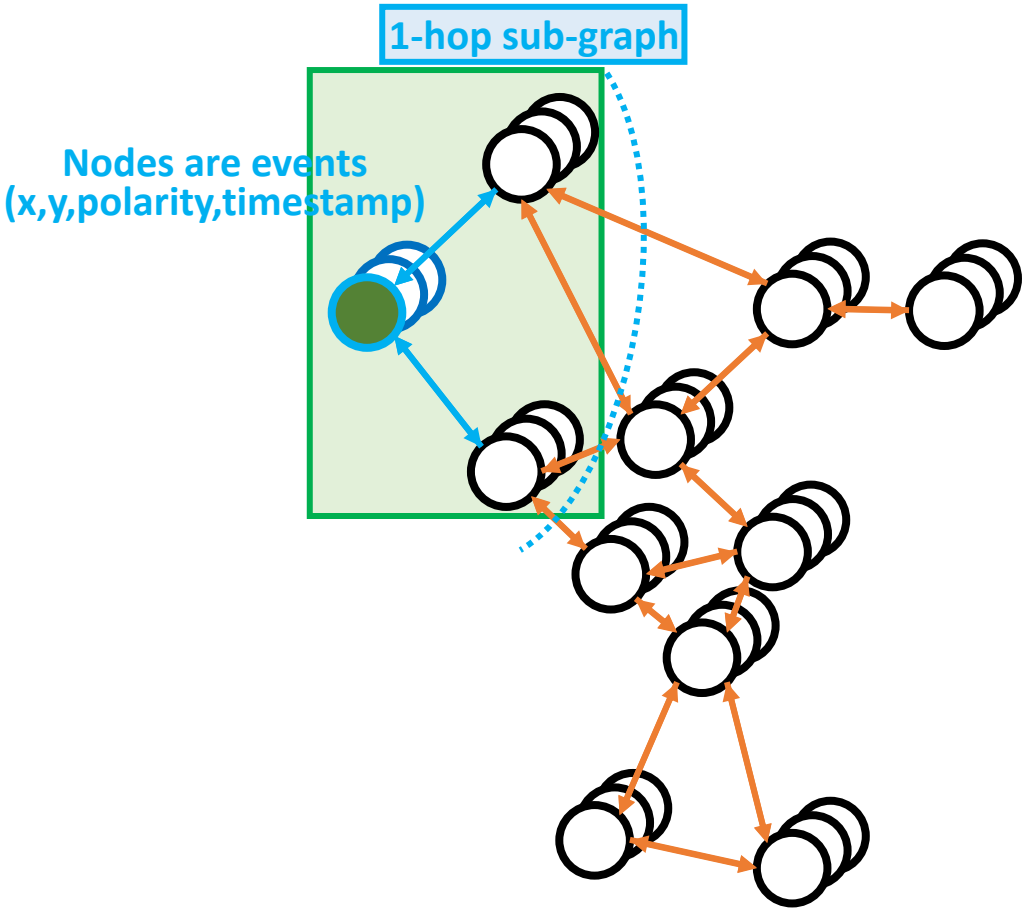
N-Caltech101



[Schaefer, *Nature*, 2024]

40x less MFLOPS/event!

Directed dynamic graphs for low-latency hardware acceleration



<u>Challenge</u>: GNNs focus on static graphs, how we make them work for event-based data?

1) **Graph building**
2) **GNN execution**

AEGNN [Schaefer, *CVPR*, 2022]

But information flows from the future to the past…?

Let's use directed graphs!
HUGNet [Dalgaty, *CVPR-W*, 2023]

EvGNN

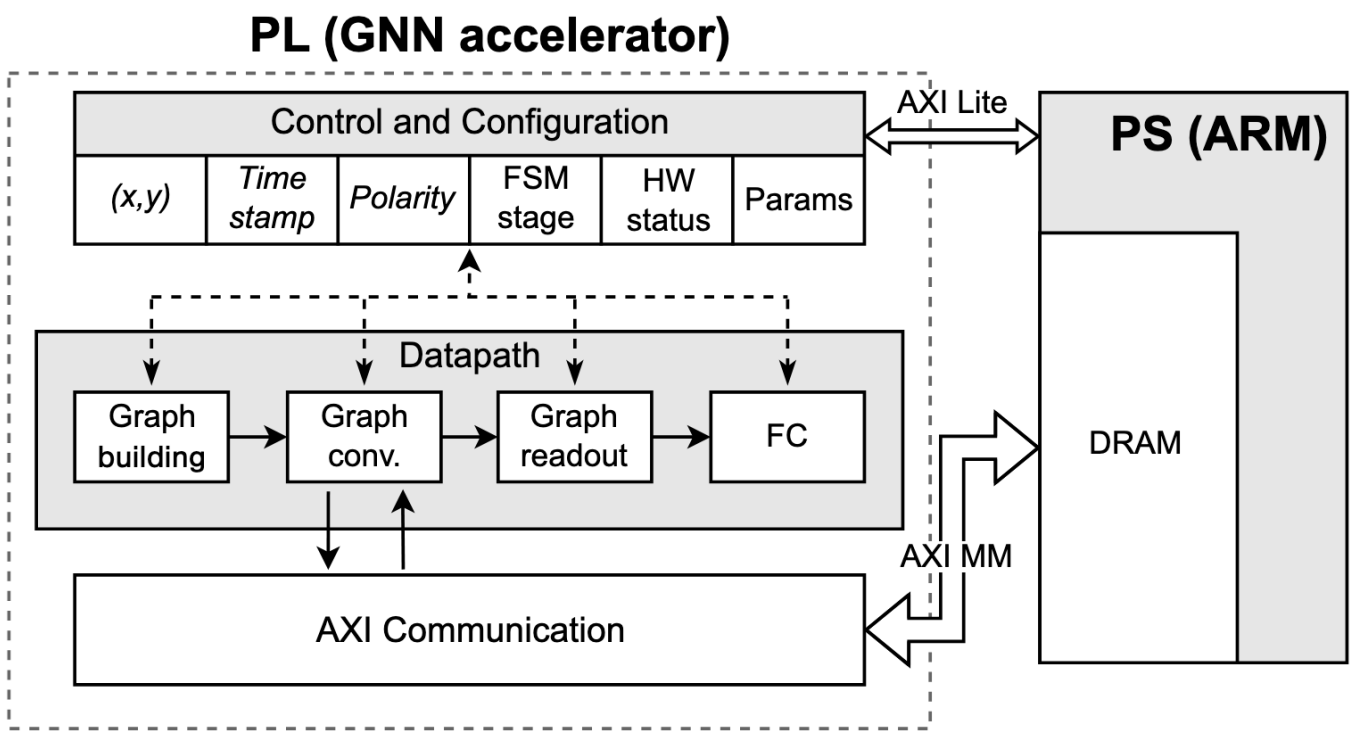
<u>Core insight</u>

Now, data is *really* exchanged locally, in an event-driven fashion!

<u>Core hardware contributions</u>

1) **Edge-free graph storage!**
2) **Neighborhood search decoupled in space-time**
3) **Parallel layer execution**
4) **No HW yet – let's harvest OoMs with a simple design!**

EvGNN – The first GNN-based hardware accelerator for event-based vision



Latency per event of 16µs (N-CARS, 87.8%) in a KV260 edge FPGA platform!

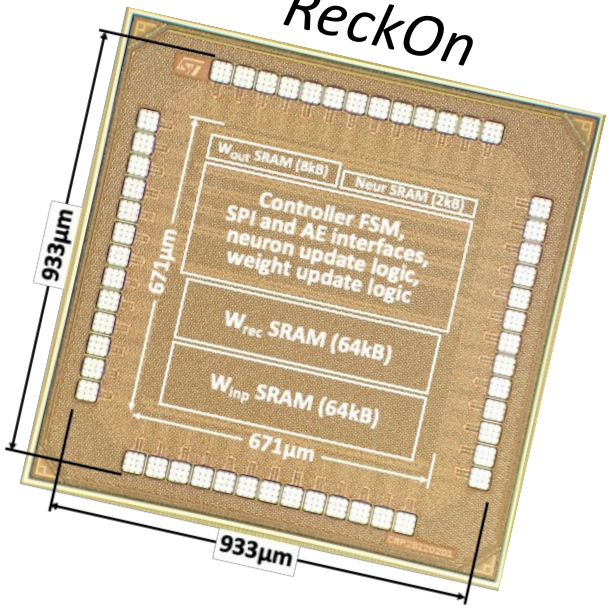
First hardware aligning with the temporal resolution of event-based cameras on a real-world benchmark. No custom silicon needed!

@C_Frenkel
cfrenkel
ChFrenkel

Does neuromorphic edge intelligence need spikes?

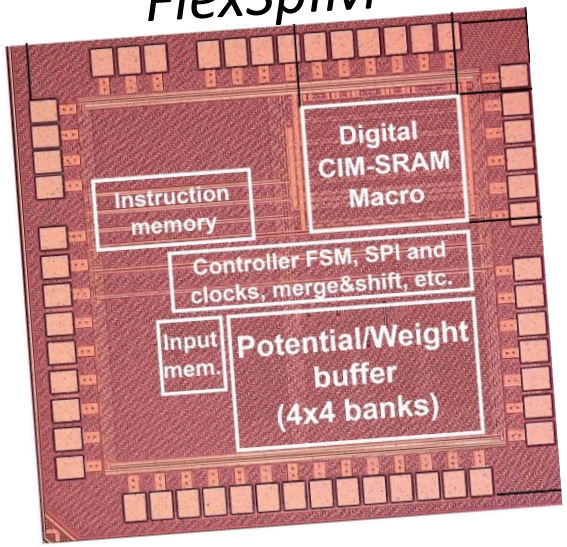
c.frenkel@tudelft.nl
cogsys.tudelft.nl

ReckOn



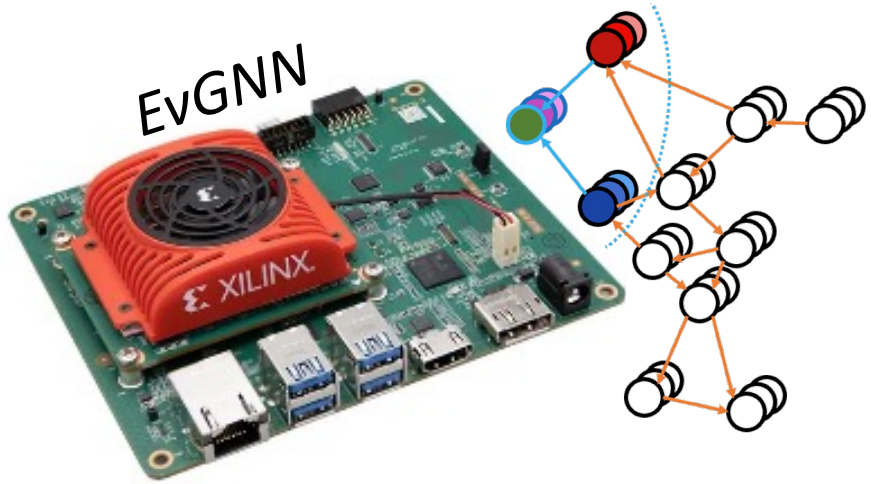
[Frenkel, ISSCC'22]

FlexSpIM



[Chauvaux, ISCAS'25]

EvGNN



[Yang and Kneip, Trans. CASAI'25]

Yes, for sensor- and task-agnostic learning

Yes, for low-latency event-based processing

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

We bridge the bottom-up (bio-inspired) and top-down (engineering-driven) NeuroAI design approaches toward multi-scale, decentralized intelligence.

Funding acknowledgements

NWO TU Delft AiNed PROPHESEE METAVISION FOR MACHINES