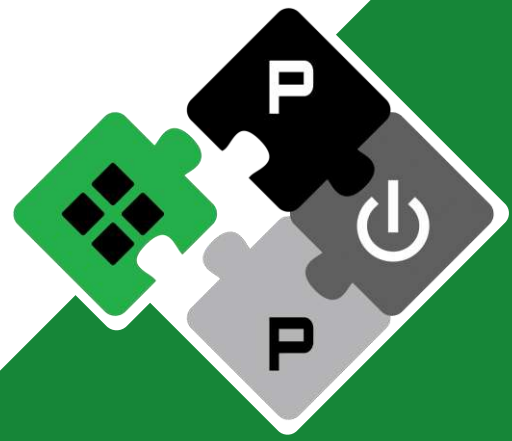
ETH zürich ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Neuromorphic Computing on PULP

Accelerating Spiking Neural Networks with digital Hardware

Alfio Di Mauro adimauro@iis.ee.ethz.ch



@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

PULP Platform

Open Source Hardware, the way it should be!

What is Neuromorphic Computing?

Artificial Neural Network

- Dimensions
  - Height
  - Width
  - Channels
- Non-Linear Activation
  - ReLu
  - Leaky-ReLu
  - Tanh
  - …

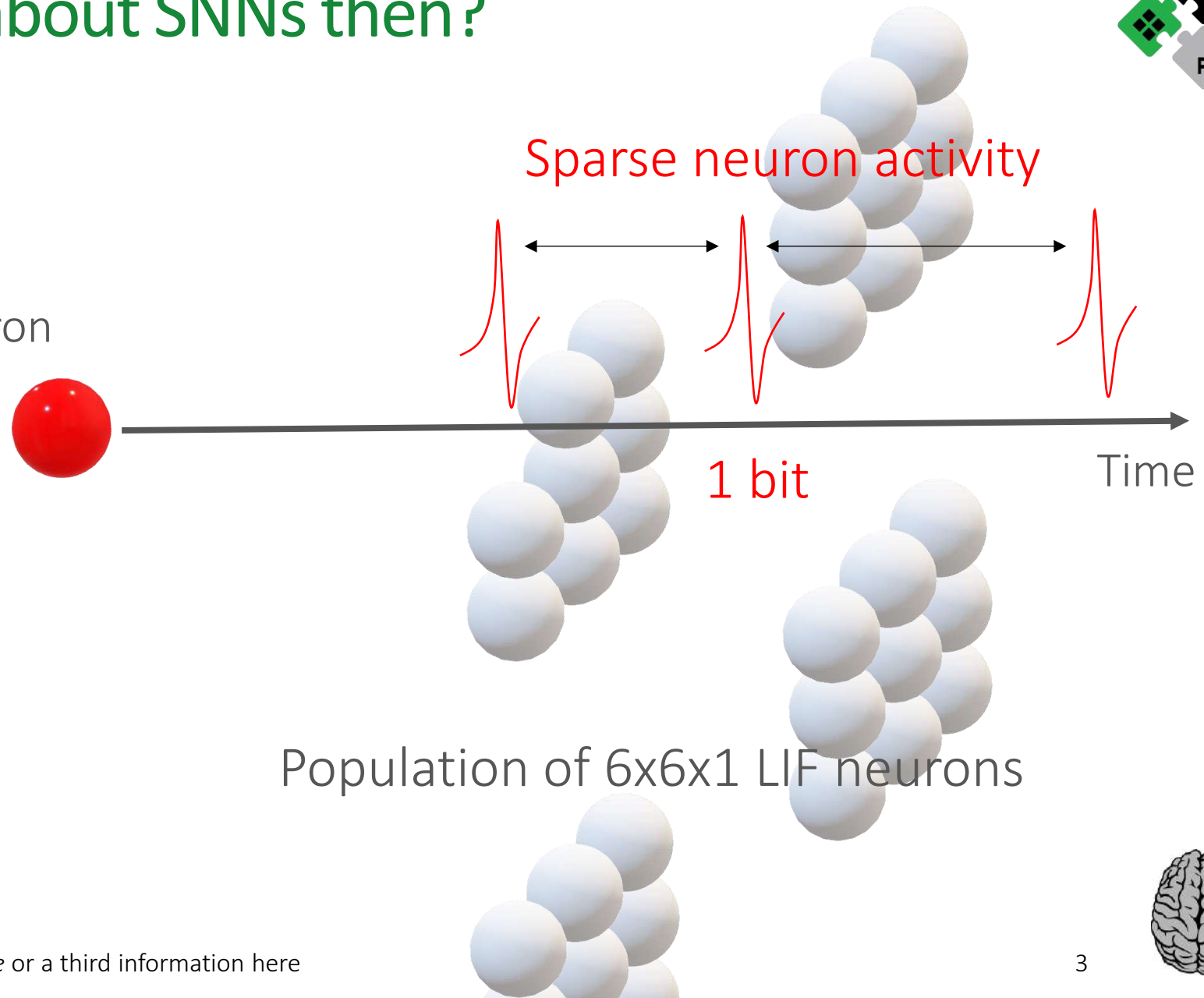
Spiking Neural Network

- Dimensions:
  - Height
  - Width
  - Channels
  - **Time**
- **Neuromorphic Activation**
  - **Leaky Integrate & Fire**
  - **Hodgkin–Huxley**
  - …

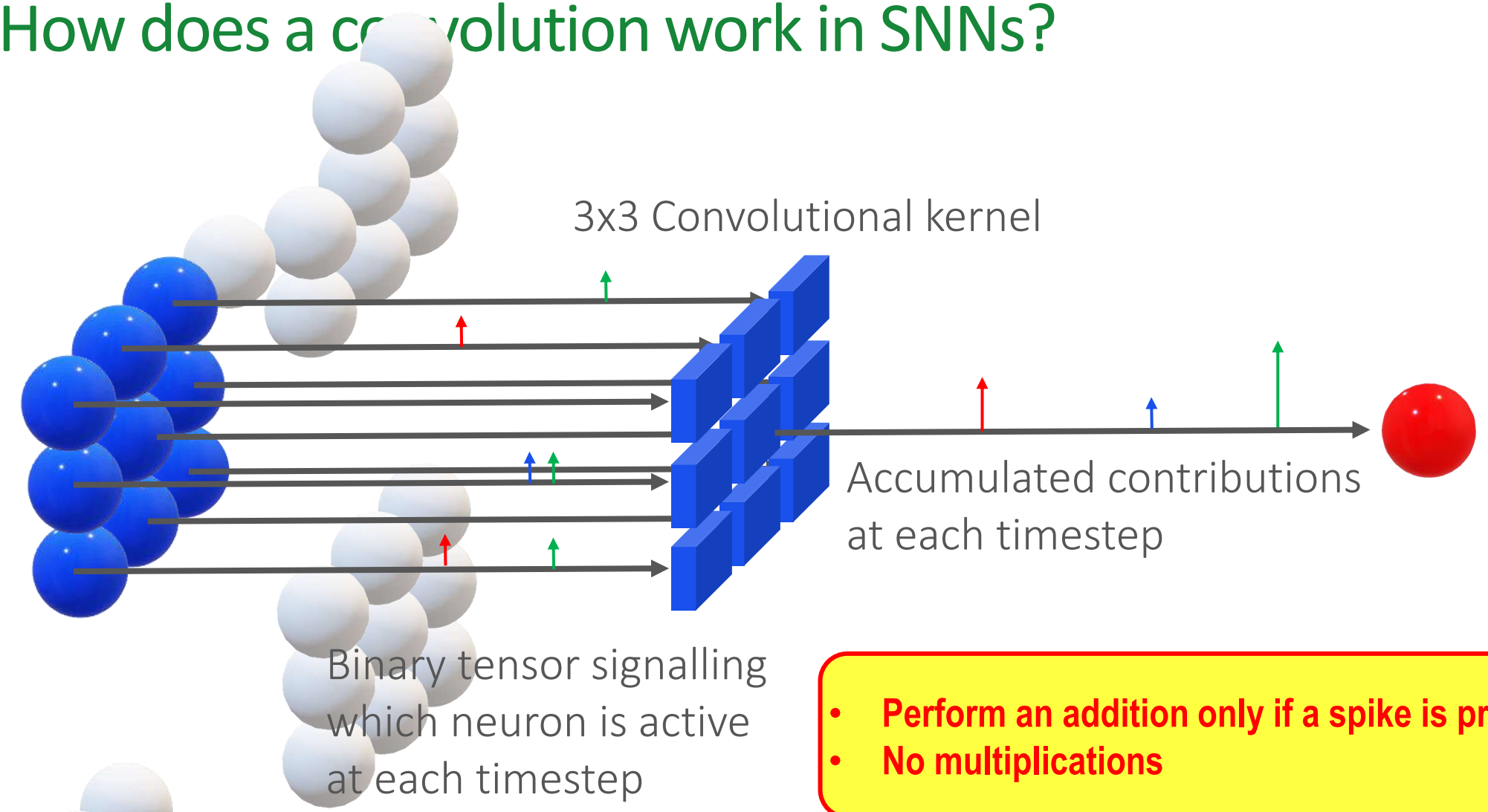
What is so special about SNNs then?

Leaky Integrate & Fire neuron

- Low complexity
- Temporal dynamics



How does a convolution work in SNNs?



- **Perform an addition only if a spike is present**
- **No multiplications**

Sparse data representation & Execution model

Event memory layout as a stream of
List of coordinate (COO) + time

OP	Time	Ch	Y	X

```
For k(c) in range(0,C_o):
        load_sne(k(c), W0)                    # pre-load the weights
```

Output Channels (SW managed)

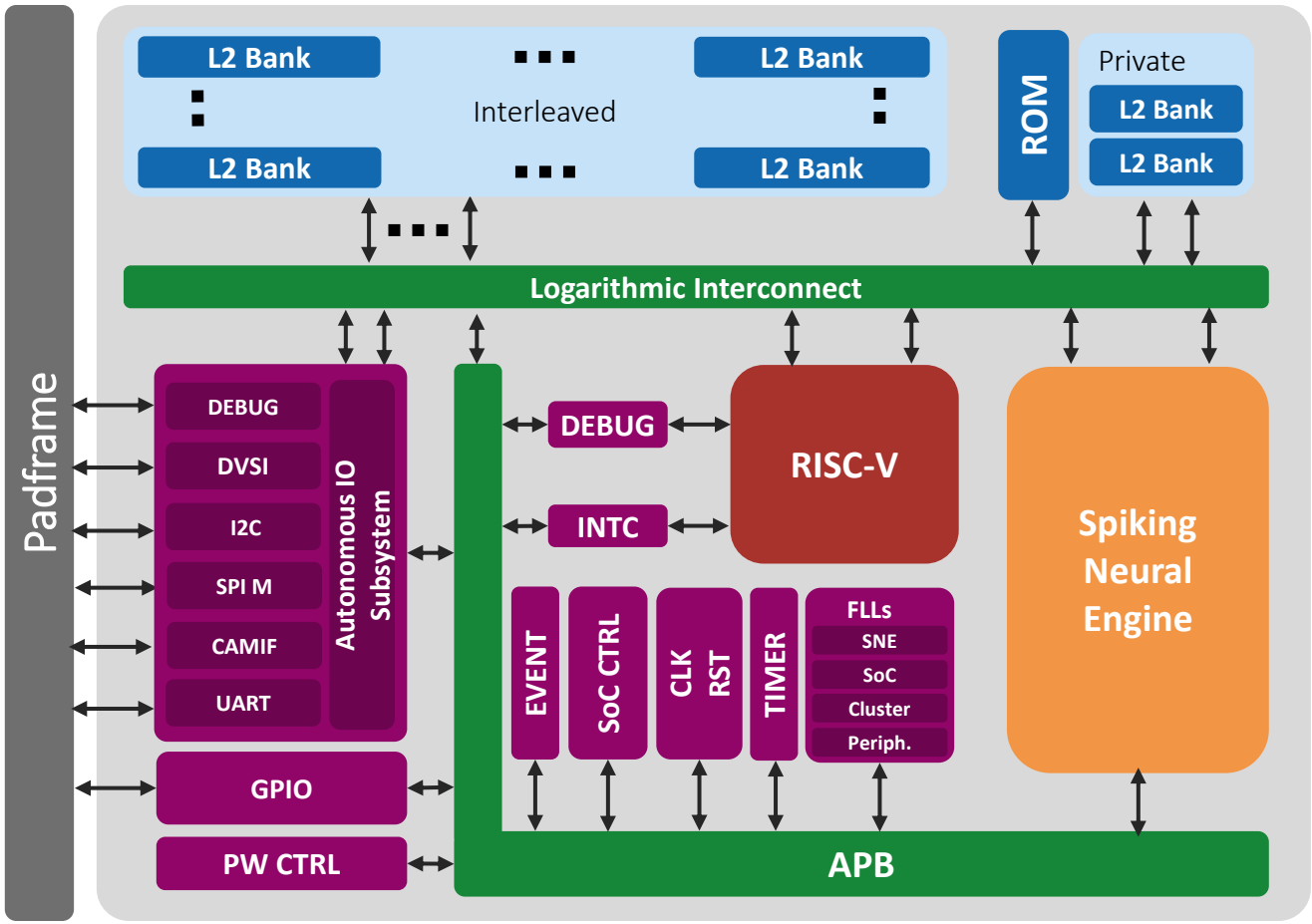
```
For evt_i in events_in:  # fetch a new event from the memory
    if evt_i is TIME_EVT:                                   # check if it's a time event
        t = evt_i                                           # store the time
        Events_out.push_time_evt(t)                         # propagate time on the output stream
        Break                                               # return from this iteration
    Else:
        c_i, e_x, e_y = get_spatial_coordinates(evt_i)      # pre-process the event data for output update

    # perform the output neuron update
    For i in [0,7]:  # span horizontal input event neighborhood
        For j in [0,7]: # span vertical input event neighborhood
            W0_ij = extract_weight(i,j,c_i,e_x,e_y,W0)      # extract the respective weight for output ch 0
            Events_out.push_evt(i,j, LIF_dynamic(W0_ij,t)) # write output events from ch 0 to the memory
```

Output neuron update

SNE execution

Can we accelerate SNNs on PULP?



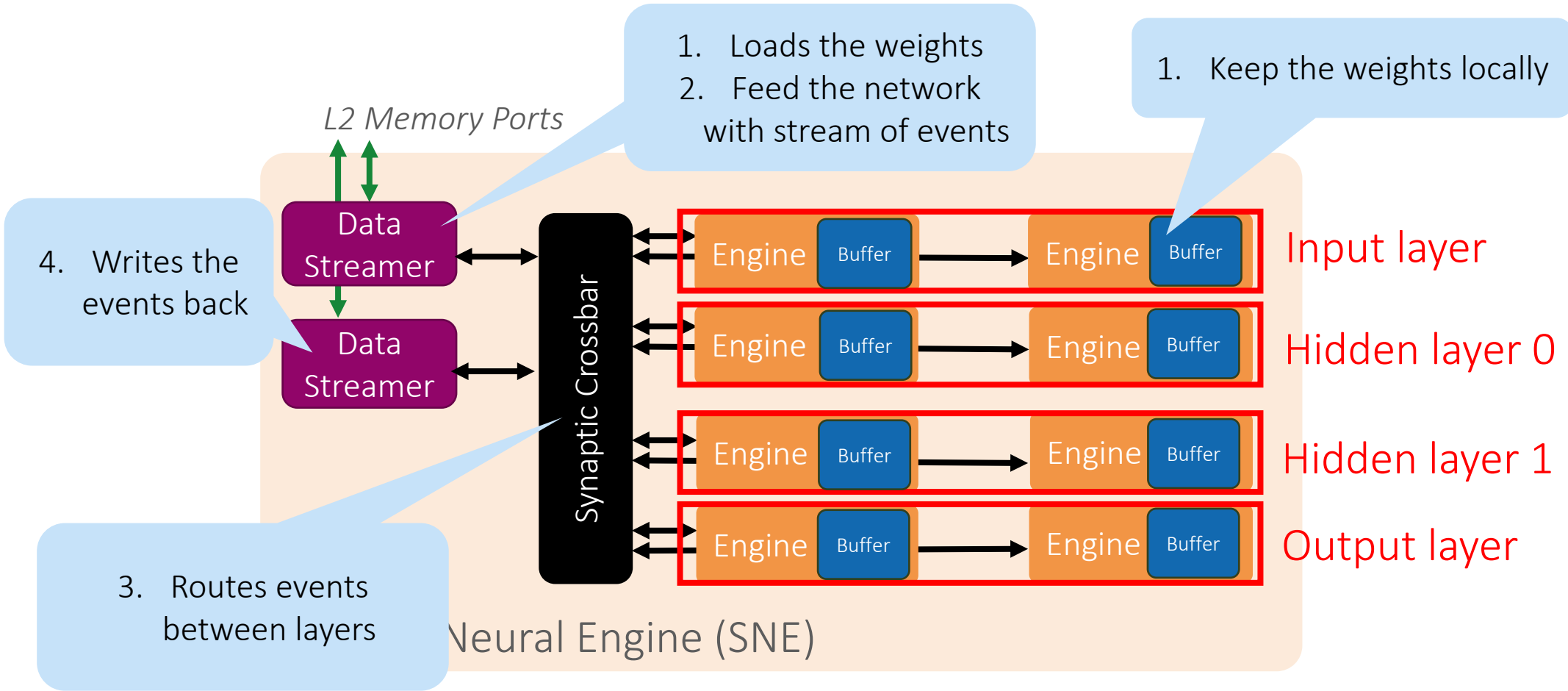
HWPE Architecture

- ❑ High bandwidth to the L2 Memory/Peripherals
- ❑ Close interaction with a GP core

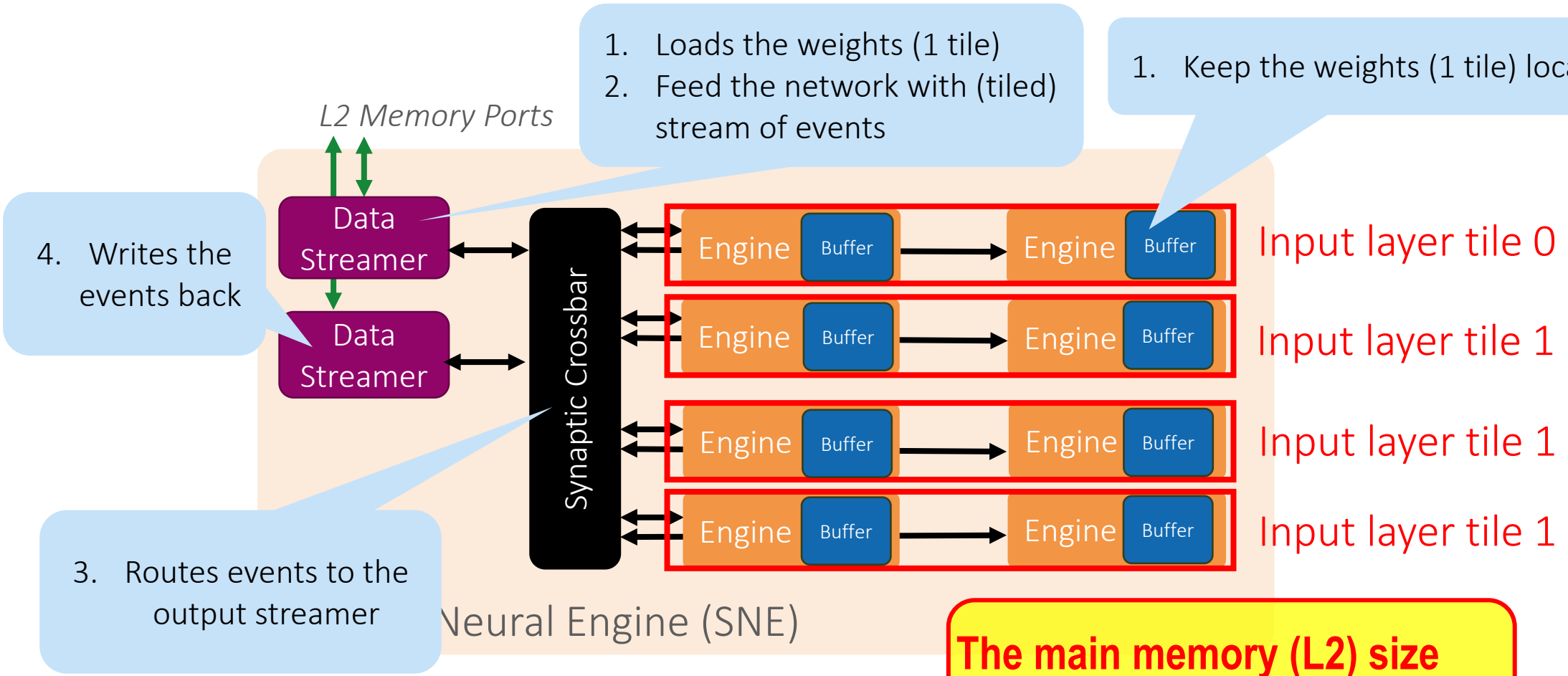
Challenges:

- ❑ Deal with data streams
- ❑ Exploit data sparsity
- ❑ Preserve energy-proportionality

SNE is a data flow architecture



What if the network is too big?

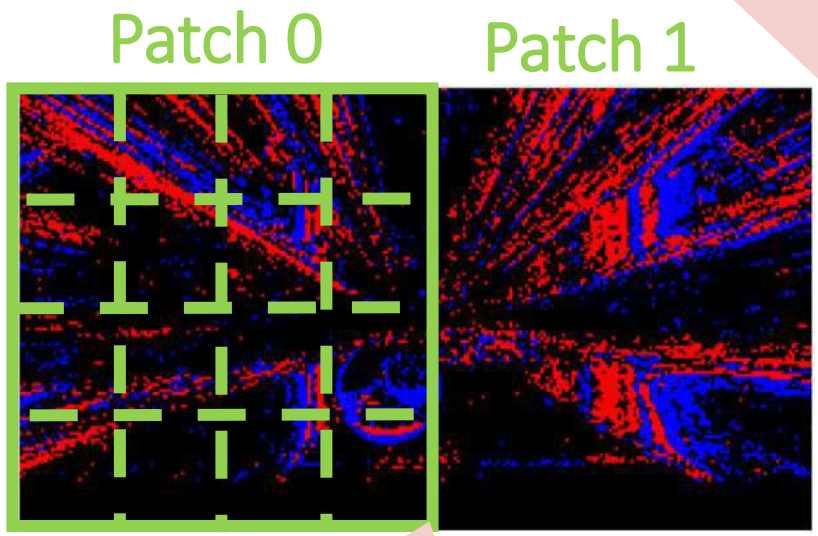


The main memory (L2) size dictates the max network size

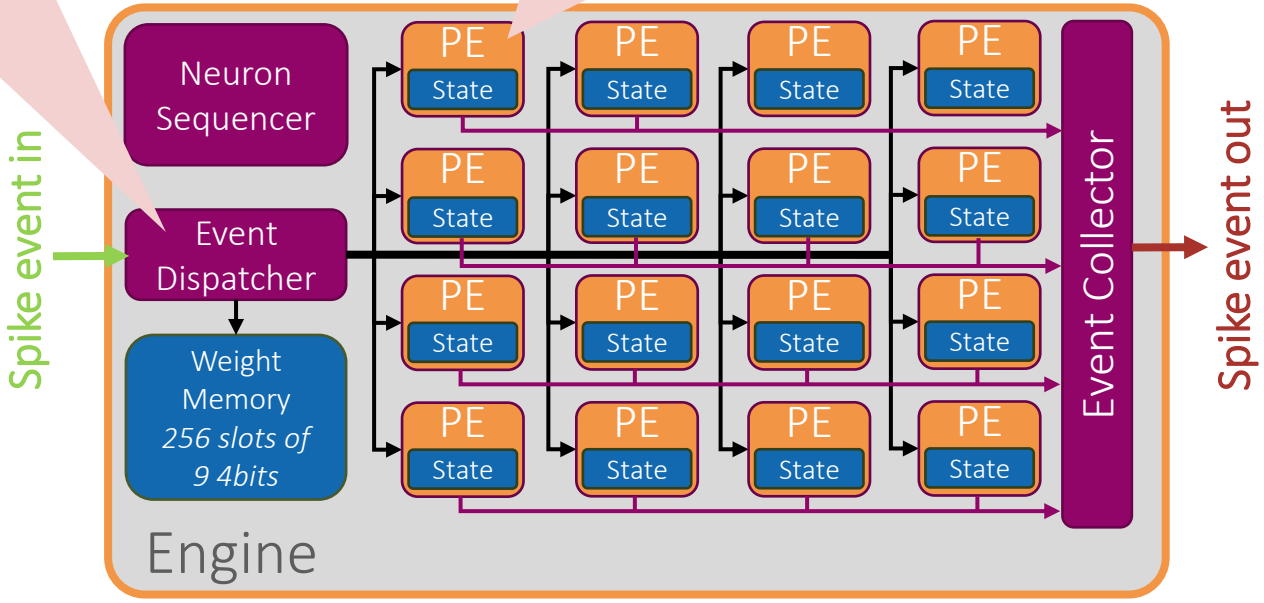
Engine's Architecture

Broadcast events to all PEs

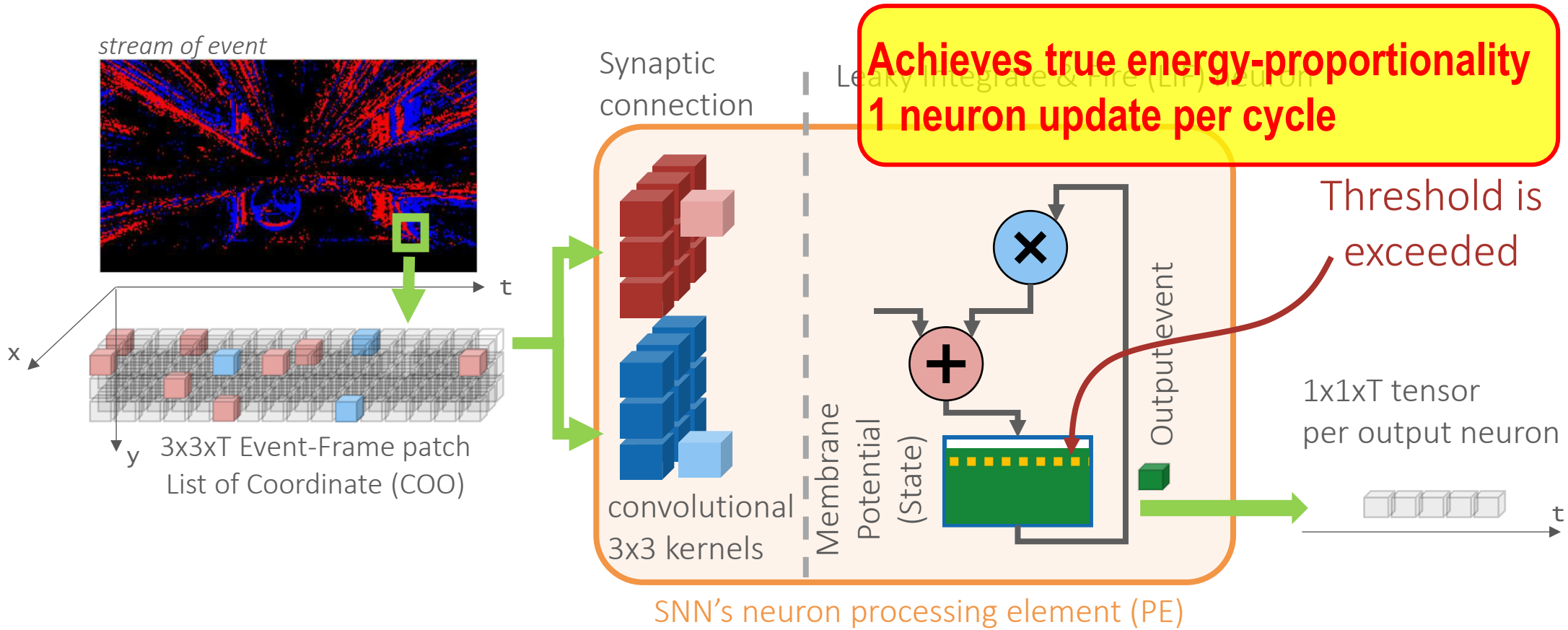
Each PE filters only the events in its spatial region of interest



Tiled execution on an input patch



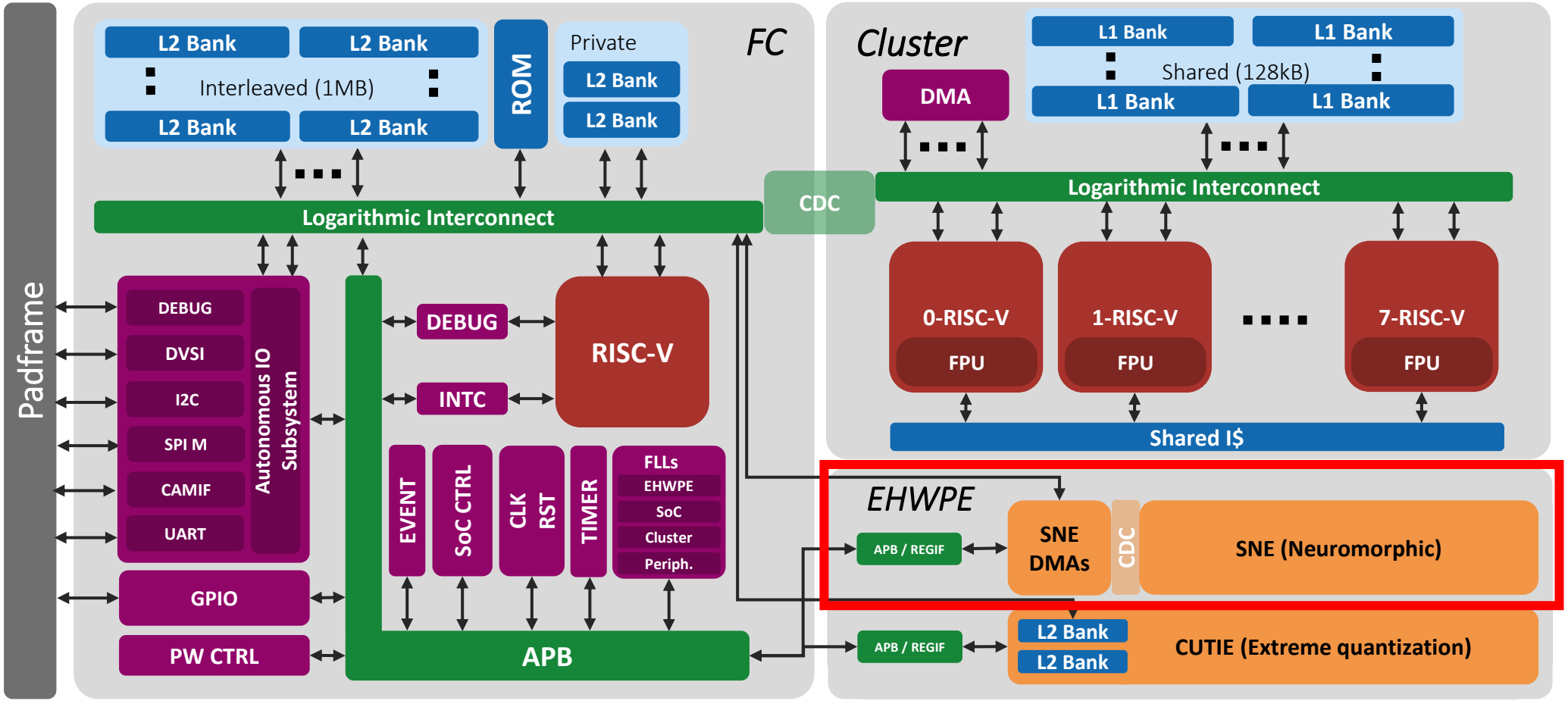
Event consumption, and output spikes generation



A more complex dynamic than conventional DNNs neurons:

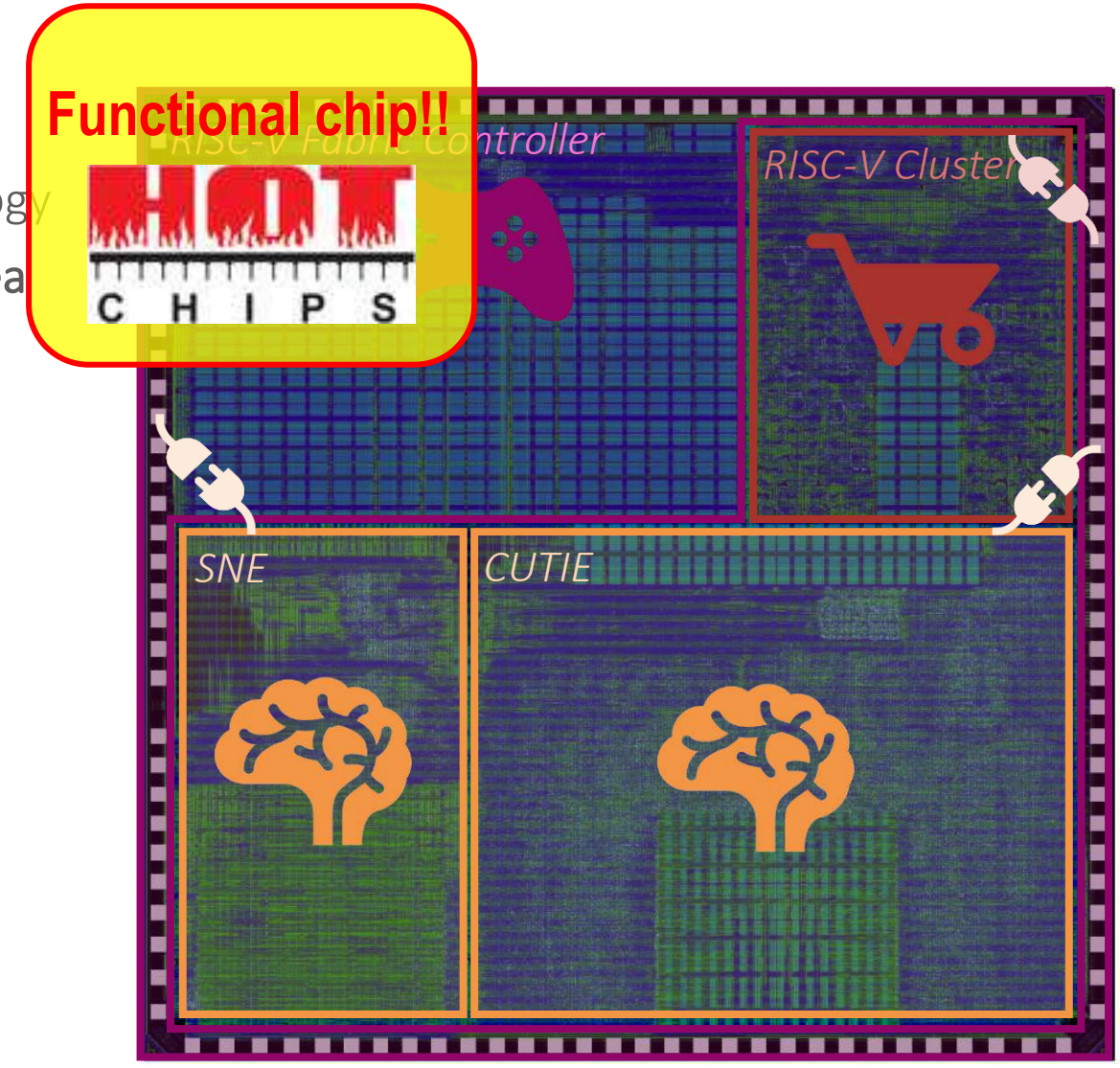
- Membrane Potential Accumulation/Activation 1 SynAcc = 1 4b-ADD + 1 8b-COMPARE
- Membrane Potential decay 1 SynDec = (1 8b-MUL) + (1 8b-MUL + 1 8b-ADD)

Alright we go to silicon: The Kraken Chip

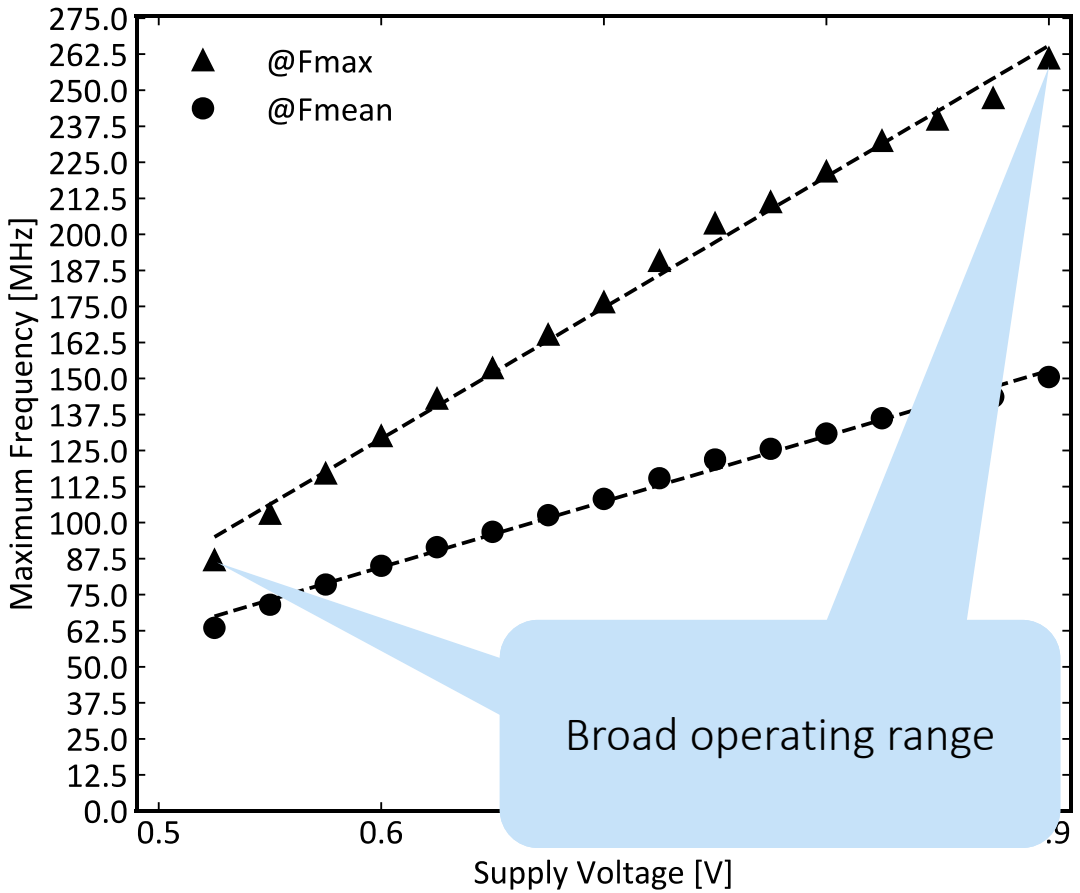


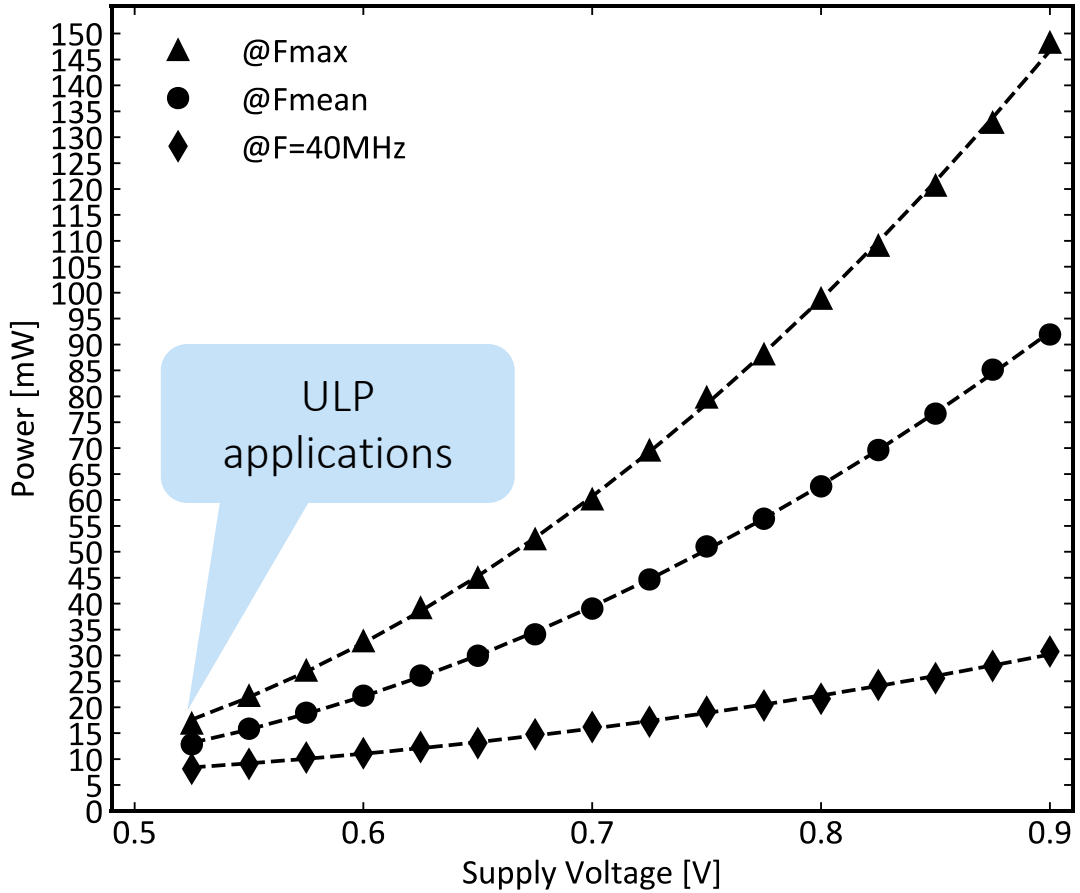
Silicon implementation

- GlobalFoundries 22nm FDX technology
- QFN88 chip package, **9mm2 chip area**
- **0.5V to 0.9V** operating voltage
- Cluster Max Freq: **370MHz**
- CUTIE Max Freq: **140MHz**
- SNE Max Freq: **220MHz**
- Independent clock/power domain:
  - RISC-V Cluster
  - SNE
  - CUTIE



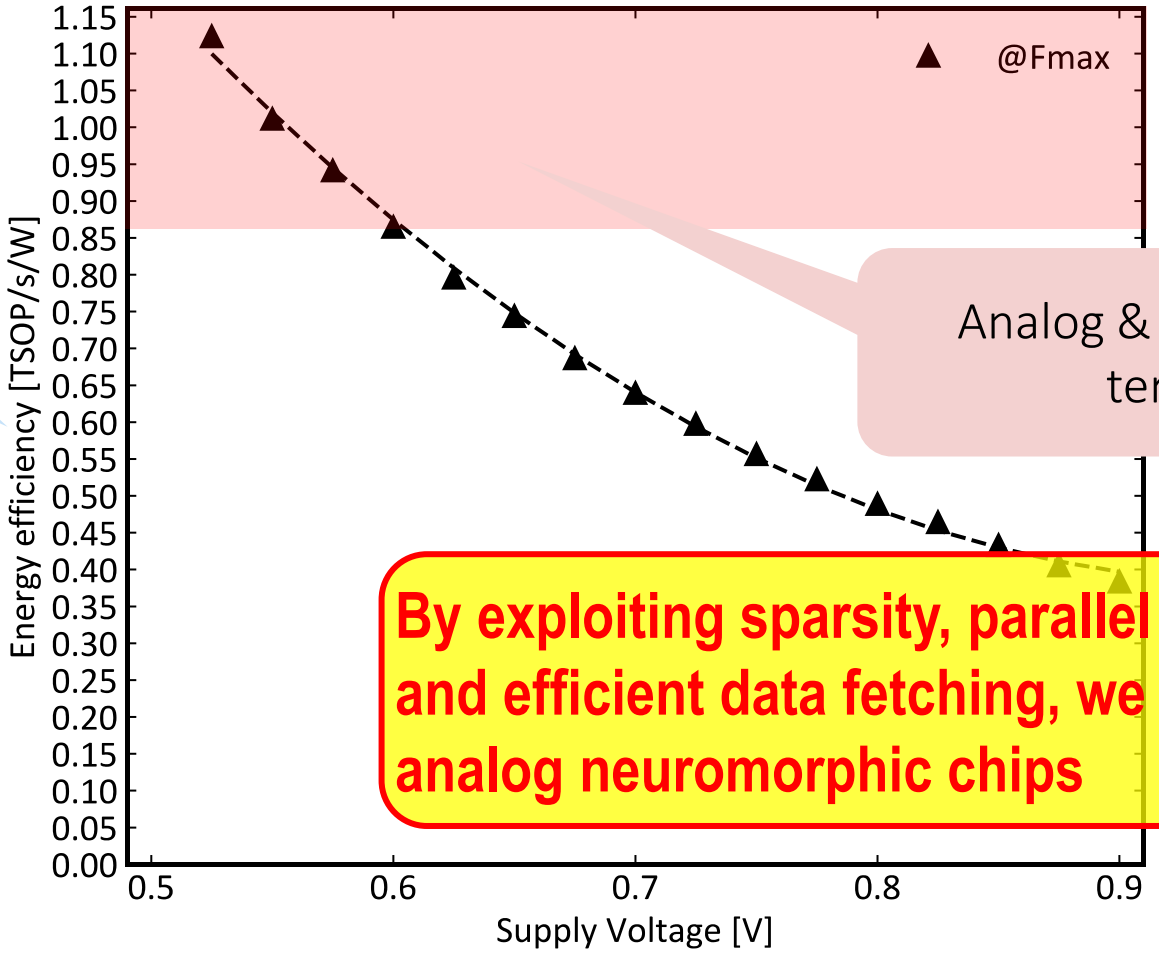
Power and Performance





Yes, we can accelerate SNNs efficiently on PULP!

It's constant in SNE
Regardless of the workload



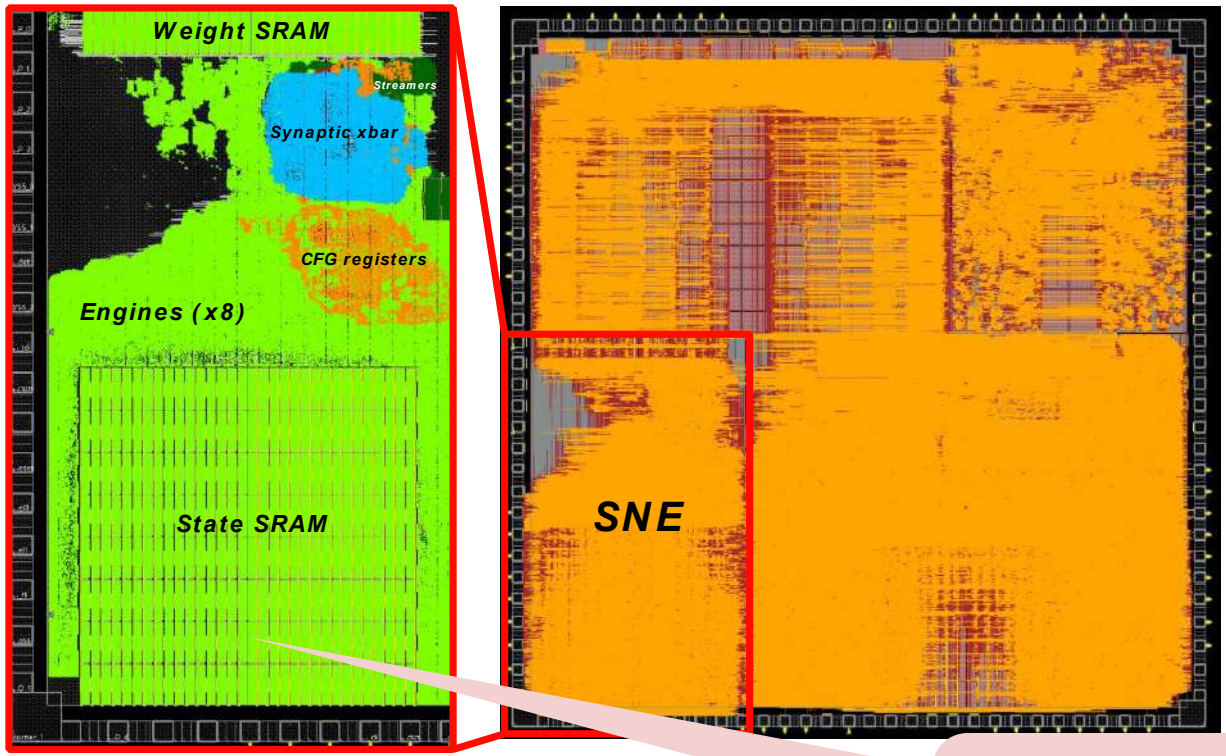
Analog & Mixed-signal territory

By exploiting sparsity, parallel computing and efficient data fetching, we compete with analog neuromorphic chips

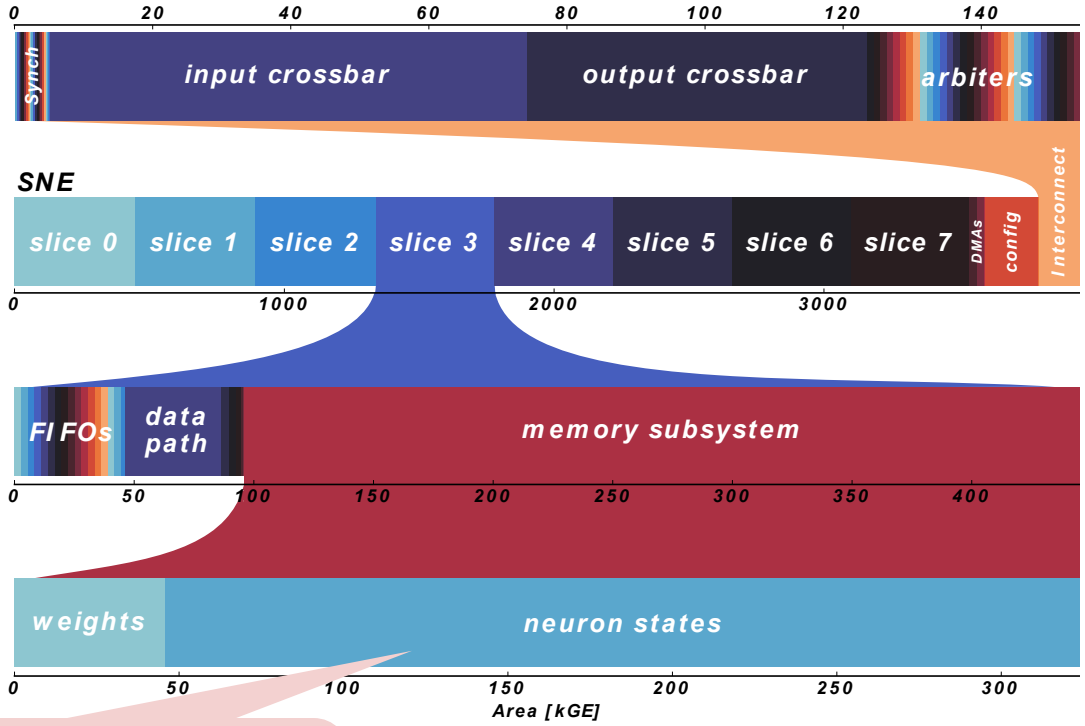
24 Synaptic Acc/cycle
522 Synaptic Dec/cycle

Area Breakdown

Kraken Layout

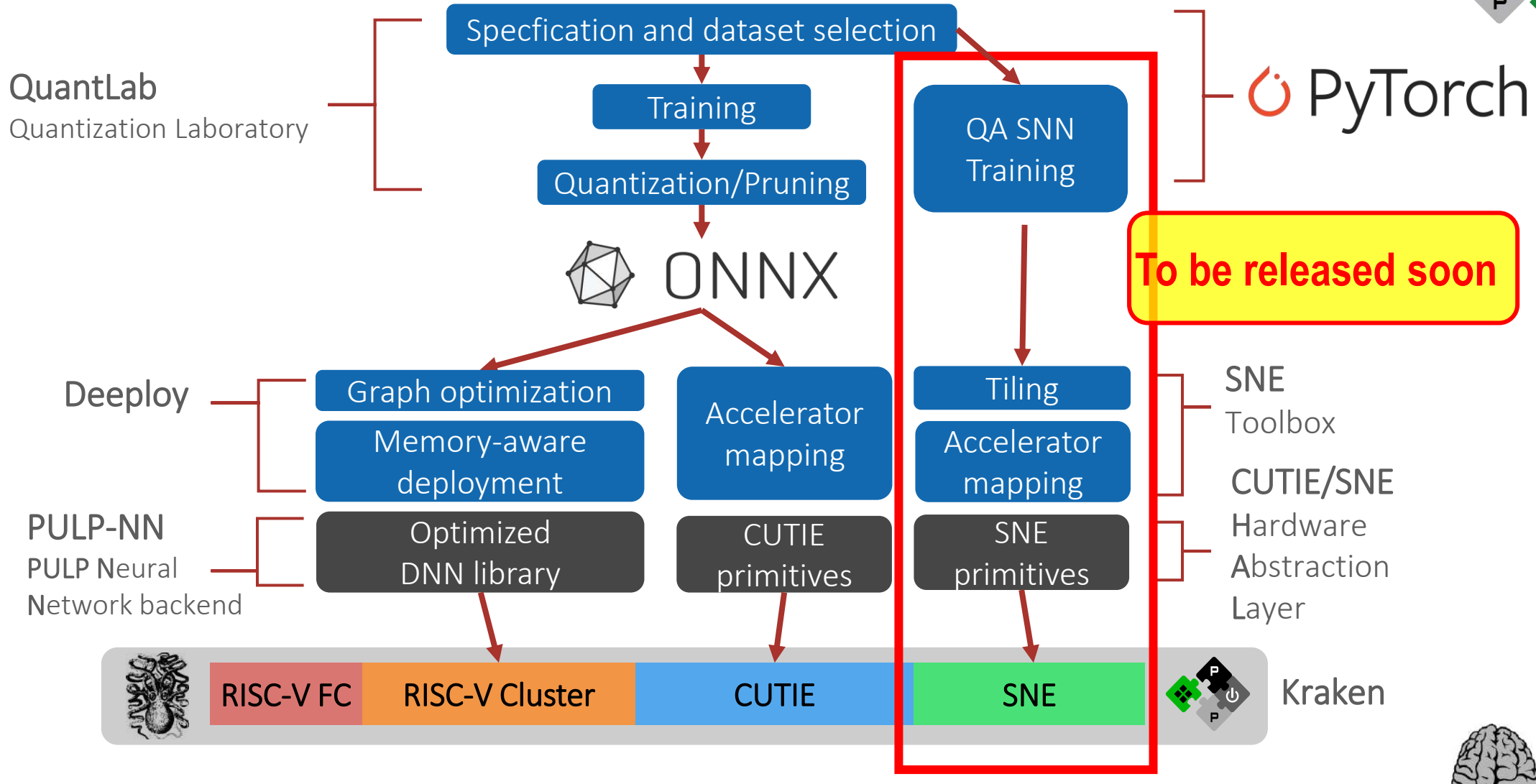


SNE Area (Kraken implem.)

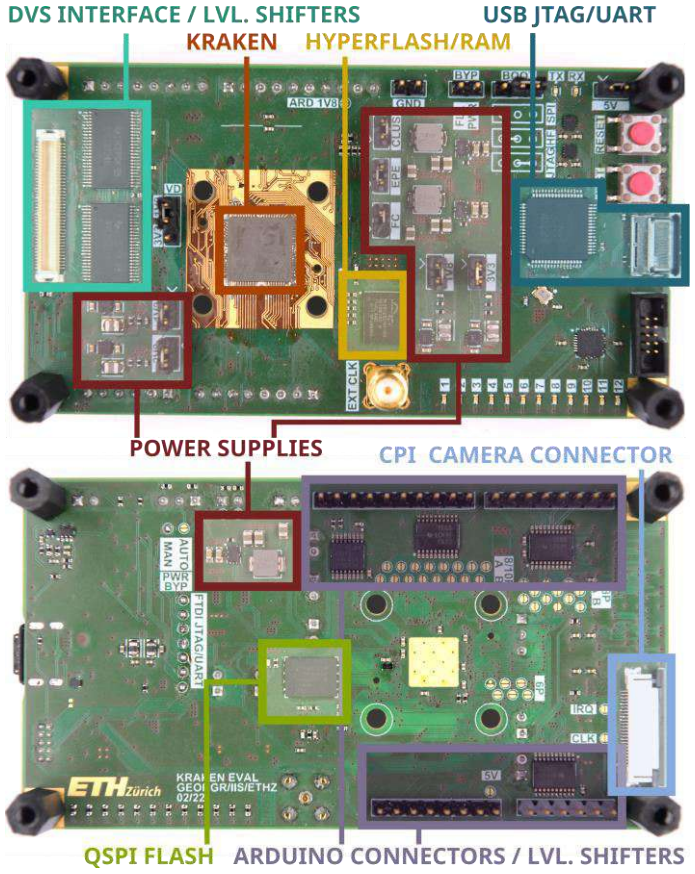


Yes, a lot of memories

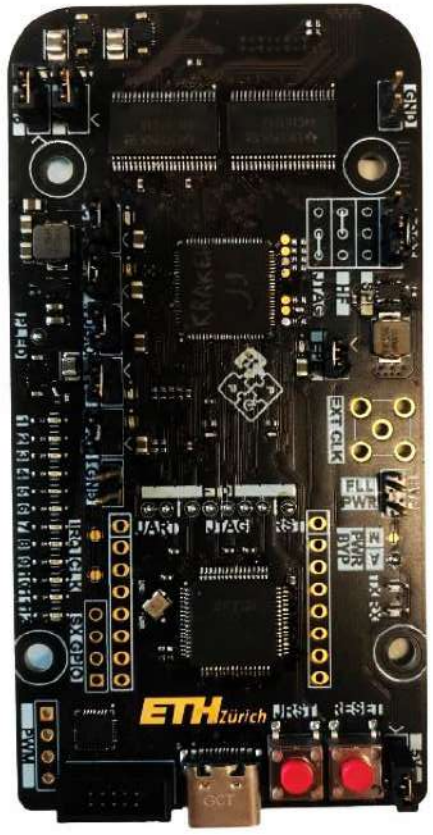
How do I train an SNNs for Kraken?



Release the Kraken



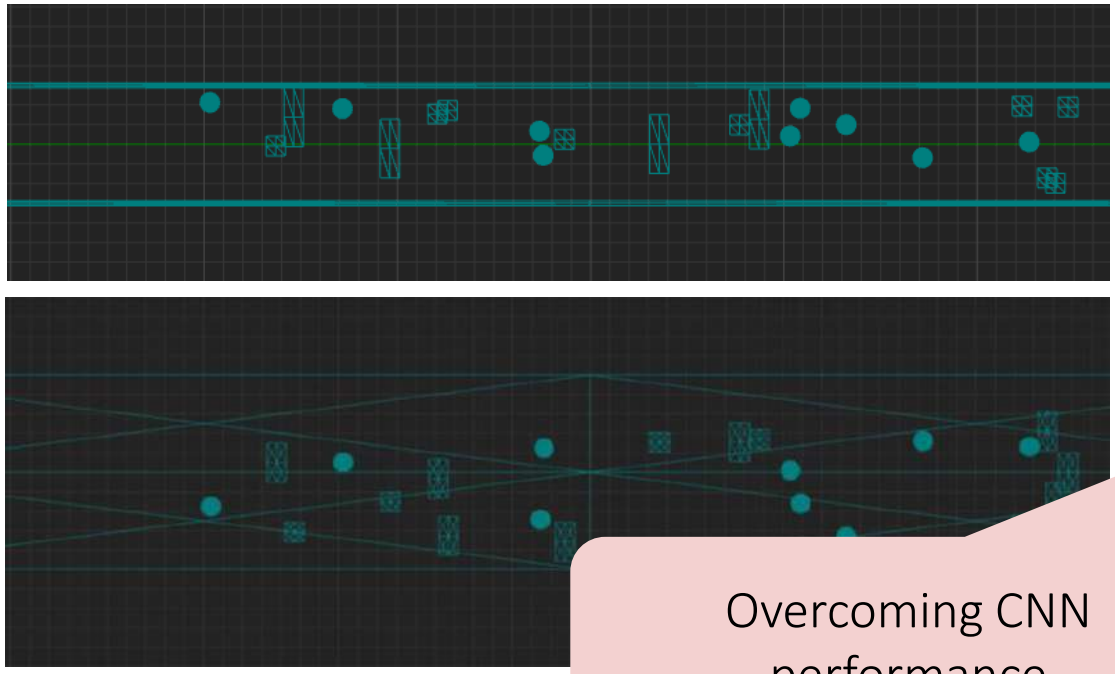
Kraken Eval Board v0



Kraken Eval Board v1

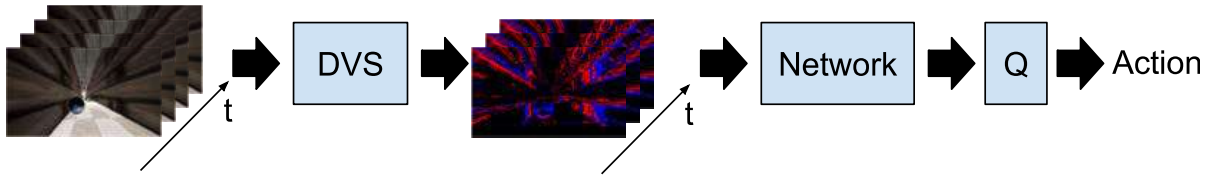
Applications: Reinforcement learning, Drone Navigation

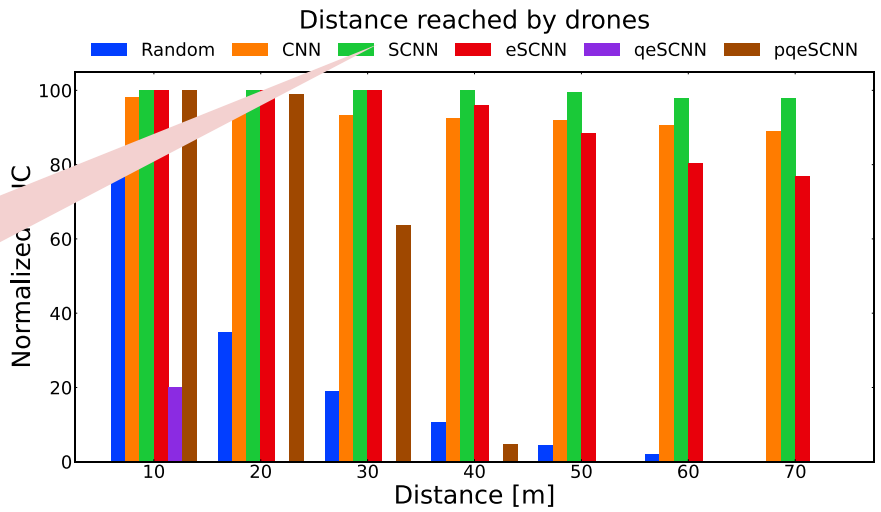
Corridor with obstacles



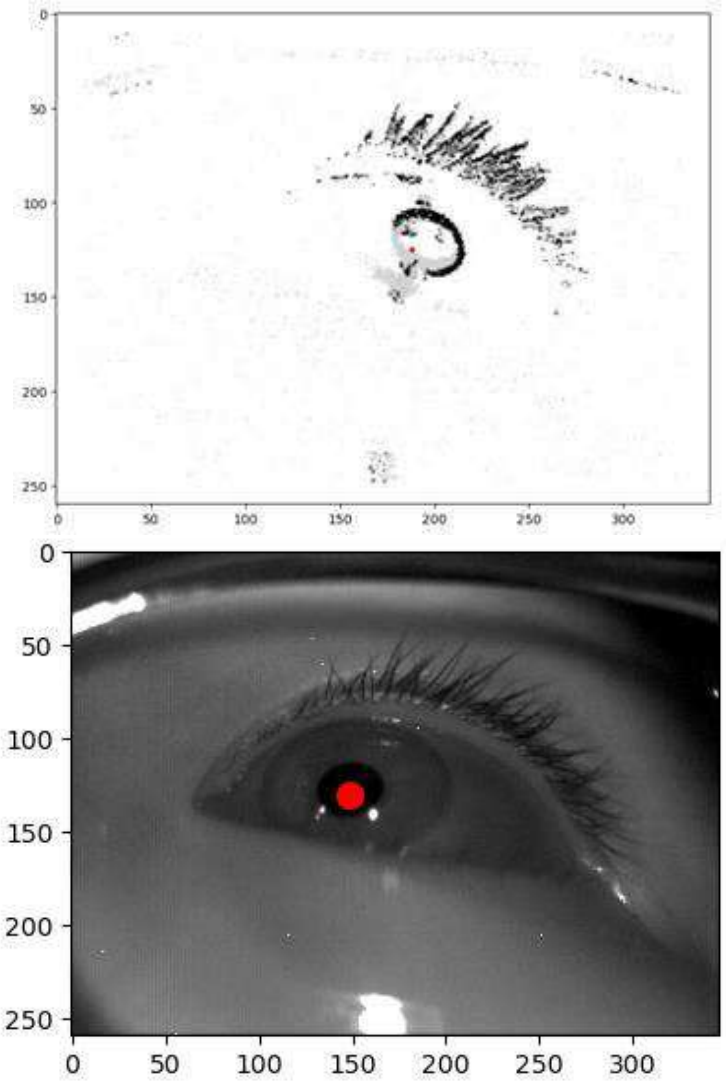
Overcoming CNN performance

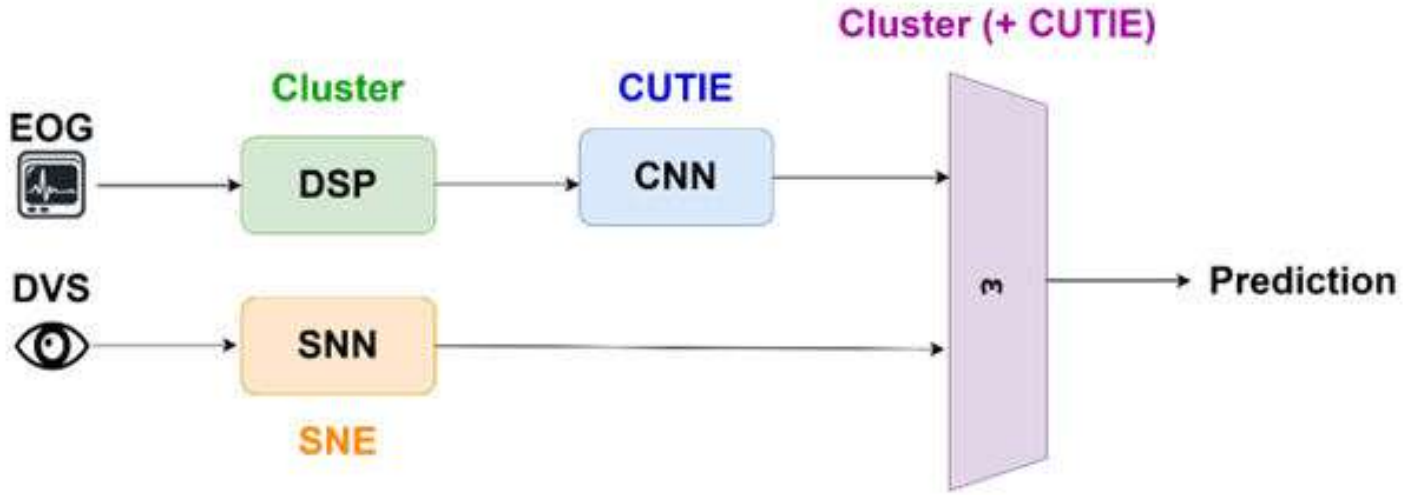
Video processing pipeline (Kraken)





WIP Application: Eye tracking with fused modalities





Robust eye gaze prediction:

- Selective field of view Rendering in AR applications
- Human-Machine Interfaces

Conclusion

SNNs acceleration challenges:

- Exploit the sparse nature of the networks
- Perform conputation in an energy-proportional way

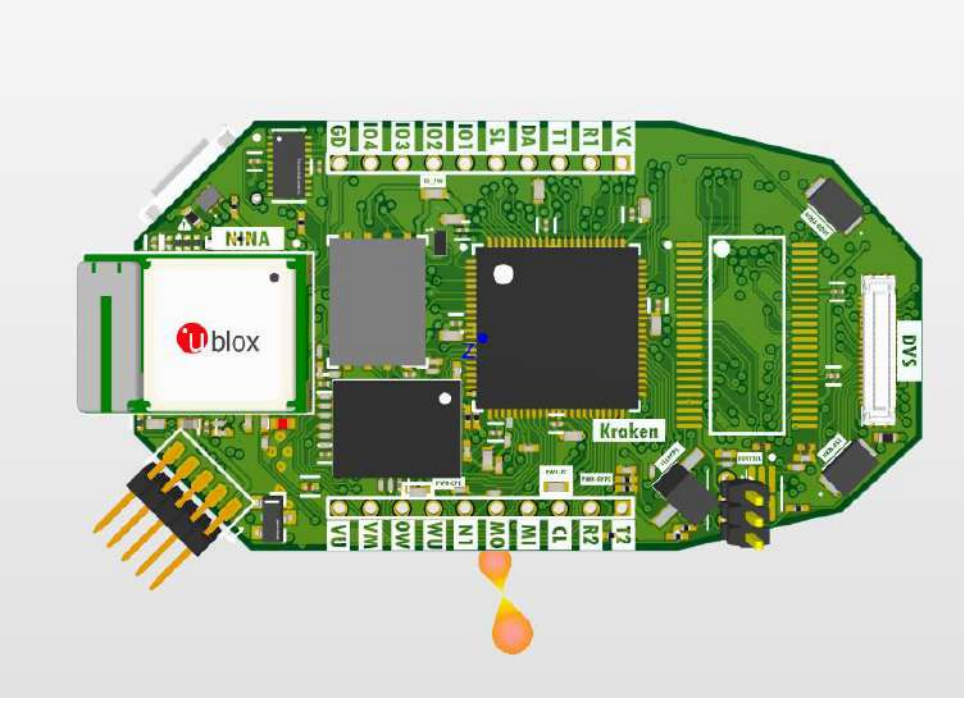
Key take-away message:

- SNNs are not very different from Conventional ANNs
- Analog and Mixed-signal is not the only way for neuromorphic computing to be efficient

Kraken: A Direct Event/Frame-Based Multi-sensor Fusion SoC for Ultra-Efficient Visual Processing in Nano-UAVs
https://www.research-collection.ethz.ch/handle/20.500.11850/565105

pulp-platform/sne (github.com)

What's next? Kraken goes on a drone



Use Kraken to pilot a nano-drone in an autonomous drone race!!

- Event-based Camera
- Frame-based Camera
- SNNs (SNE)
- DNNs (CUTIE + Cluster)