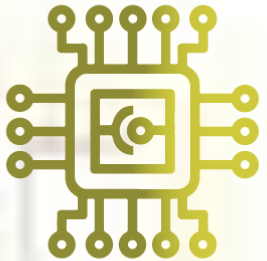
AI at the Edge: hype or hope?



Keynote at FIRE HWACC FPGA workshop

Henk Corporaal, Manil Dev Gomony

Eindhoven Univ. of Technology (TUE)

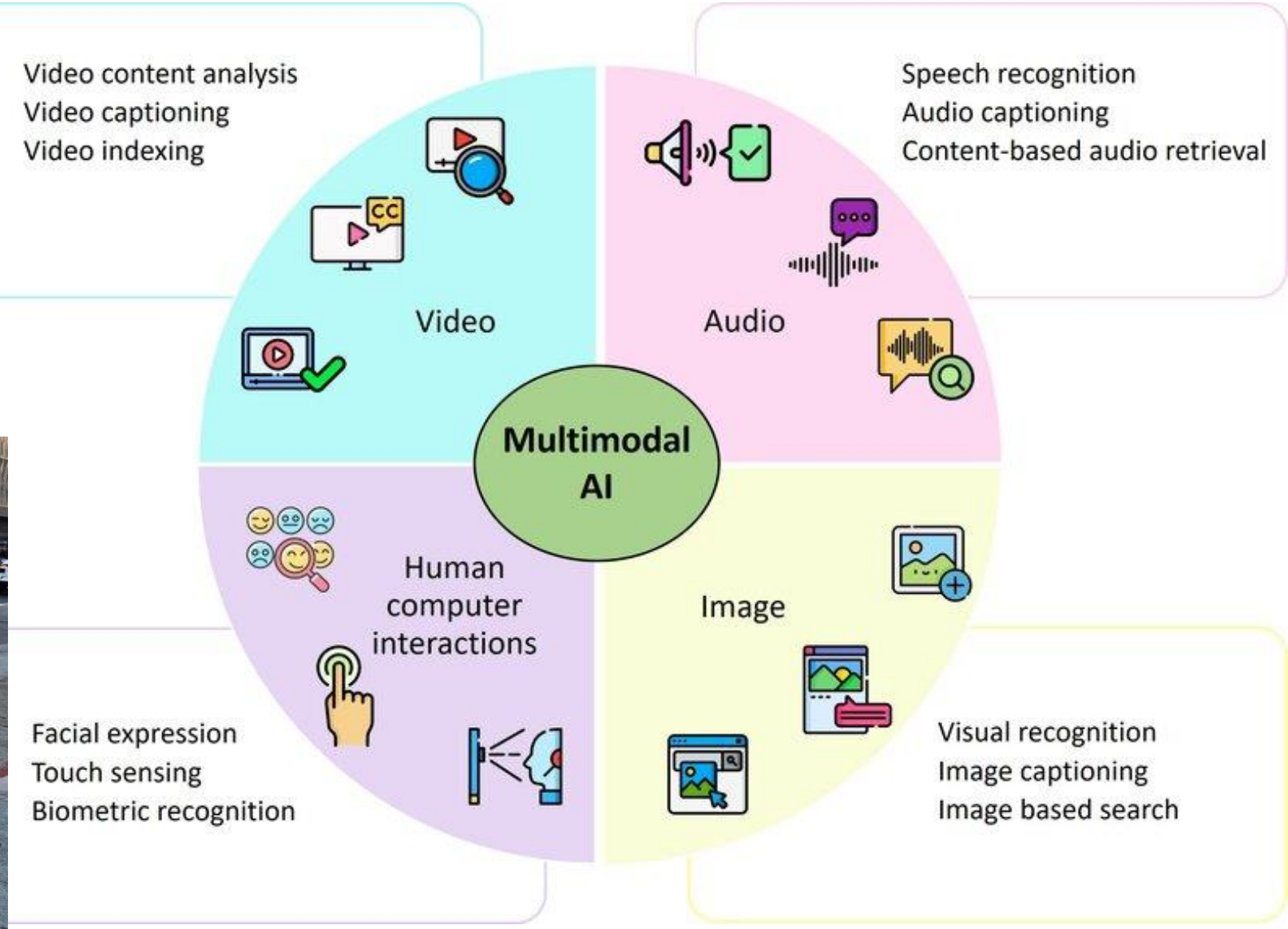
December 6, 2024

SURF

ES ELECTRONIC SYSTEMS TU/e

Booming AI

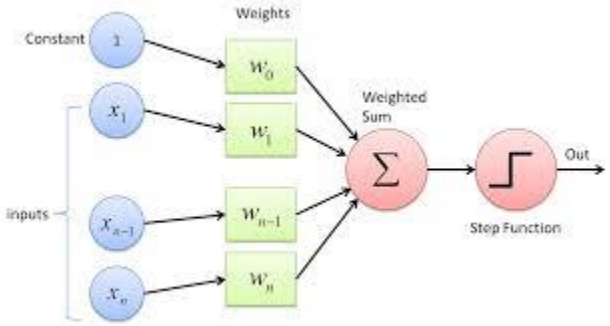
- Waymo car
  - Lidar for 3D map
  - Cameras scene understanding
  - Radar in adverse wheather
  - Ultrasonic closeby
  - GPS positioning

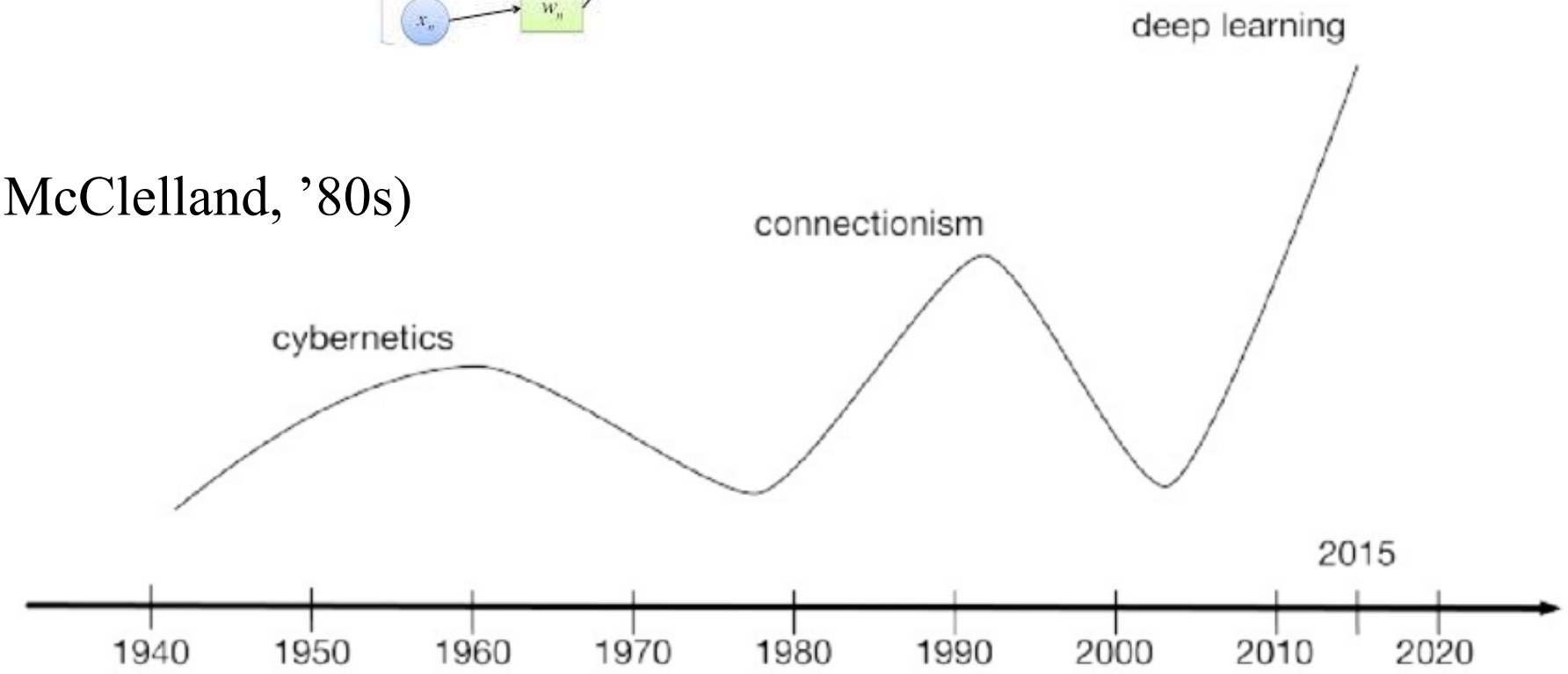


Sohail, Shahab, et.al. (2023). The Future of GPT: A Taxonomy of Existing ChatGPT Research, Current Challenges, and Possible Future Directions. 10.48550/arXiv.2307.14107

3 Historical Neural Network Waves

- ~1960
  - Perceptron
    - McCulloch-Pitts ’43; Rozenblatt ‘58
  - 1 layer
- ~1990
  - PDP (Rumelhart & McClelland, ’80s)
  - Backpropagation,
  - 2 layer perceptrons
- ~2010
  - Deep Learning
  - CNNs & RNNs

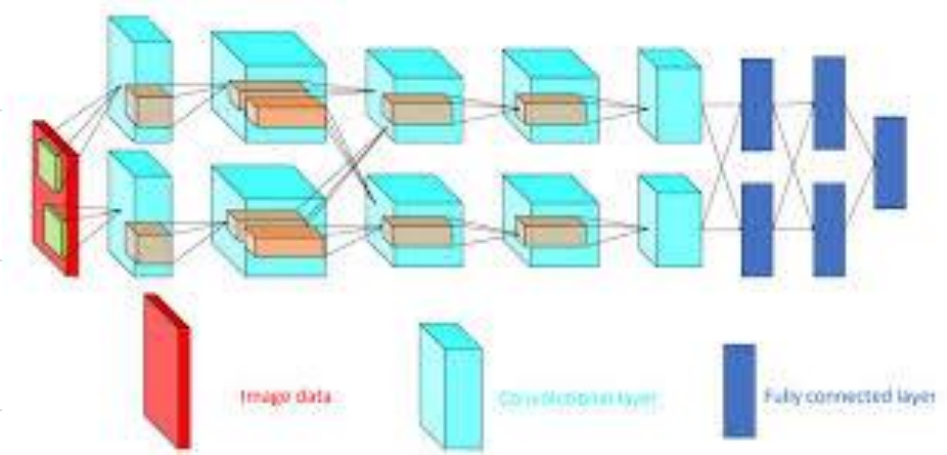




The three historical waves of artificial neural networks research (GOODFELLOW; BENGIO; COURVILLE, 2016)


saturn
school-bus
scorpion-101
screwdriver
segway
self-propelled-lawr
sextant
sheet-music
skateboard
skunk
skyscraper
smokestack
snail
snake
sneaker
snowmobile
soccer-ball
socks
soda-can
spaghetti
speed-boat
spider
spoon
stained-glas
starfish-101
steering-wheel
stirrups
sunflower-101
ImageNet challenge: 10M images, 10000 classes

ImageNet Winners (top-5 classification error)



What to expect?

- **AI Deep Learning Models**
  - **What is learning?**
  - **From CNN to Transformer**
- Edge Mismatch: Cloud vs Edge
- Optimizations
- Learn from the Brain
- SOTA in Edge AI computing
- Future
- Conclusions

Learning

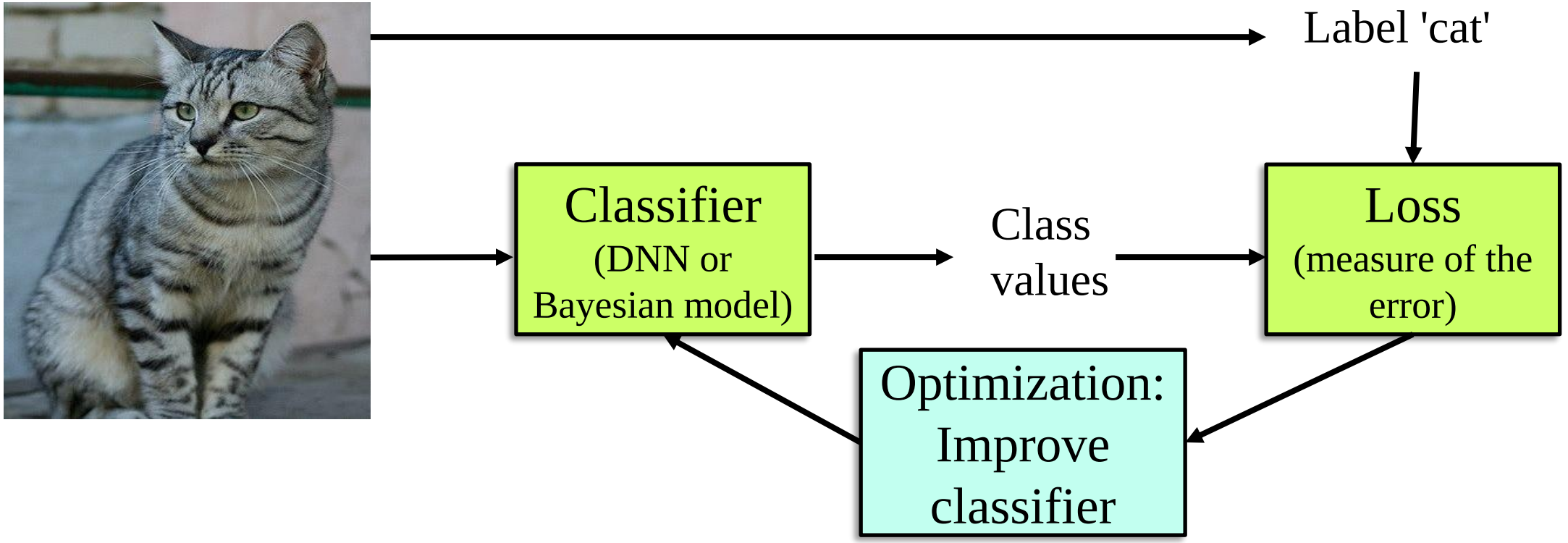
Traditional CS

Machine Learning

Concl: We learn 'by example'

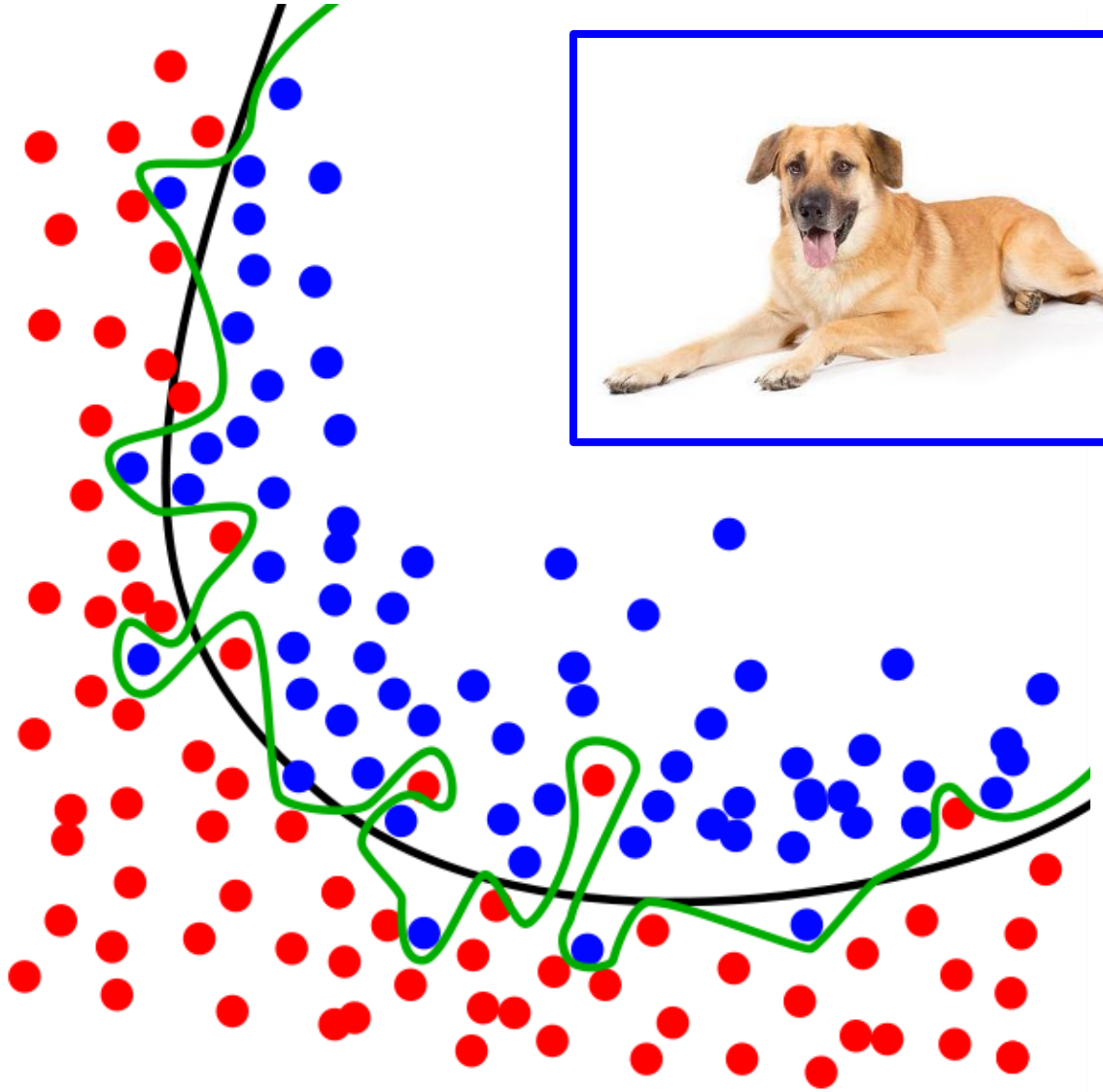
3 key functions of a learning system

- Score function (**Classifier**) : Function to map input to output
- **Loss** Function : Evaluate quality of mapping
- **Optimization** Function : Update classifier (minimizing Loss)



Overfitting vs. Generalization

- add Regularization loss term (L1, L2), penalizing large w_i
- later other regularization techniques were added, e.g.:
  - dropout
  - data augmentation



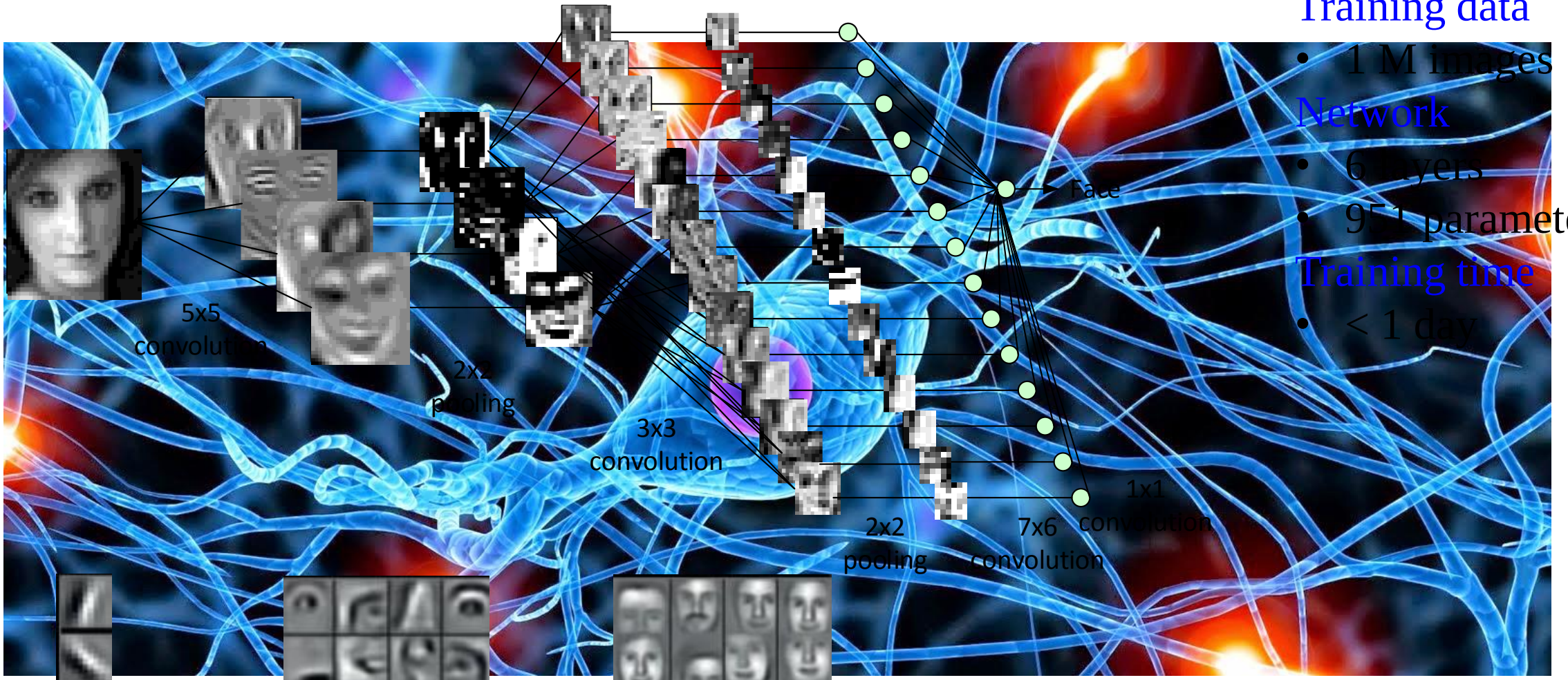
Deep Learning, a quick tour

A Simple Task
- Detect face

Training data
- 1 M images

Network
- 6 layers
- 951 parameters

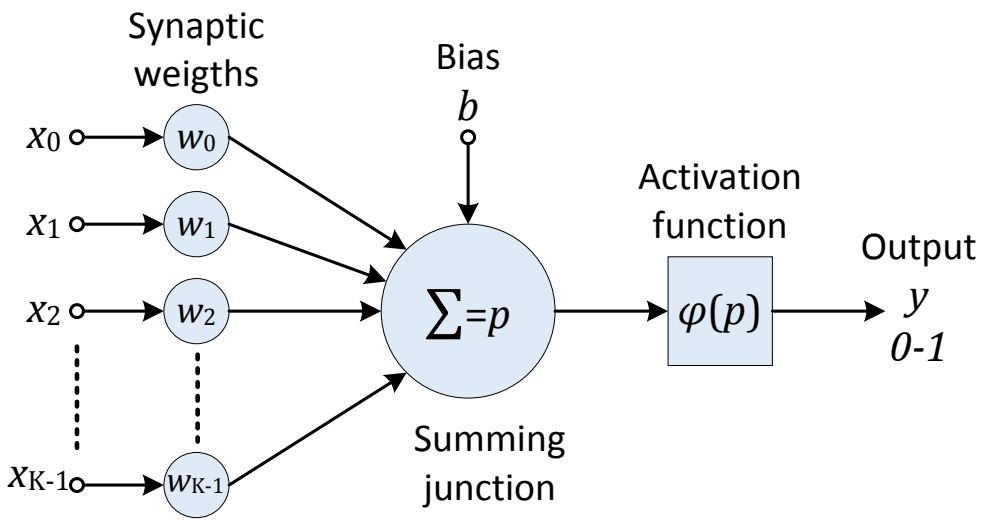
Training time
- < 1 day

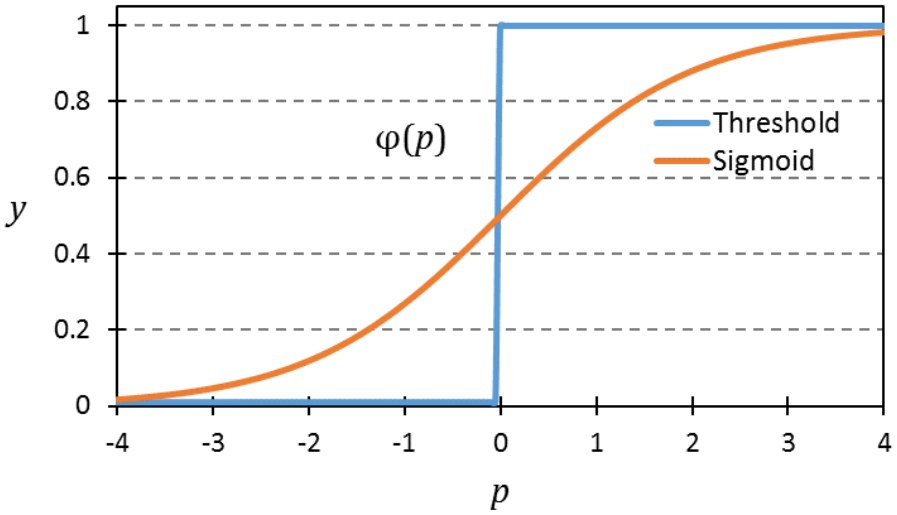


Convolutional network as a deep loop-nest

Example input vector

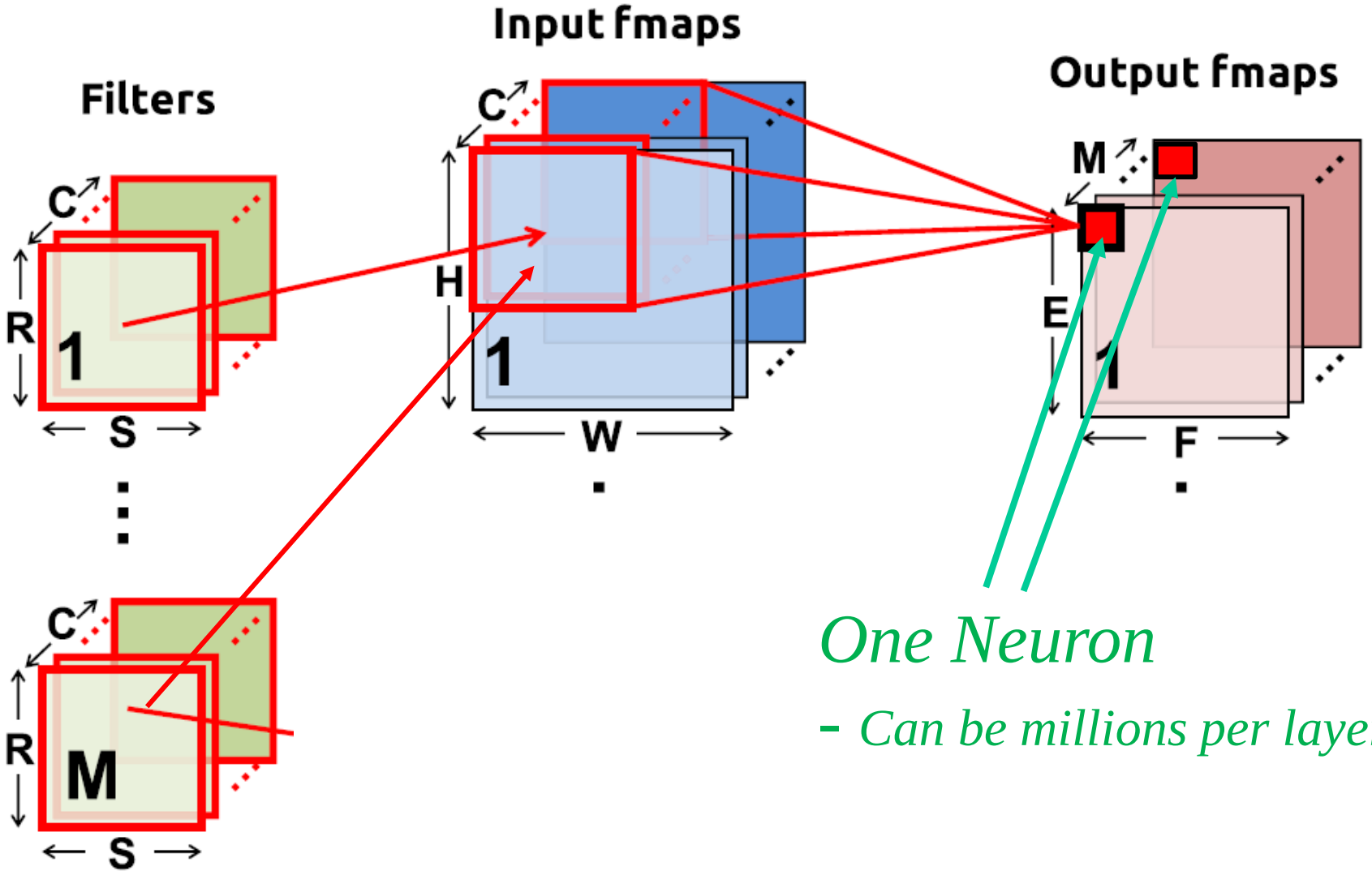
0	0	0	0	0
0	0	1	0	0
0	1	1	0	0
0	0	1	0	0
0	0	1	0	0
0	1	1	1	0
0	0	0	0	0





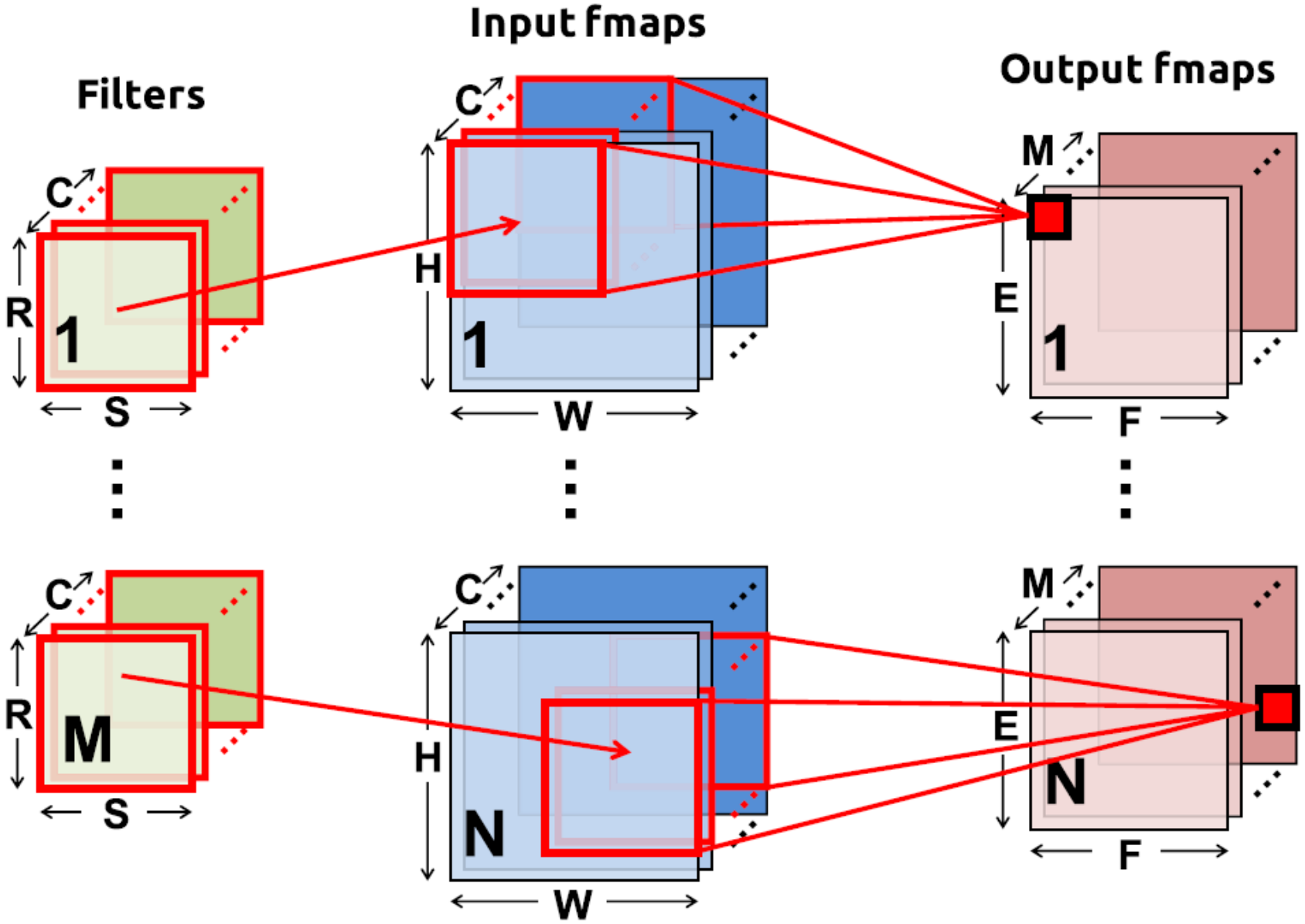

```
for l in layers:
  for o in output_maps[l]:
    for i in input_maps[l]:
      for x in columns[l]:
        for y in rows[l]:
          for kx in kernel_widht[l]:
            for ky in kernel_height[l]:
              out[l][o][x][y] += in[l][i][x+kx][y+ky] * w[l][i][o][kx][ky]
          fout[l][o][x][y]= f_act(out[l][o][x][y])
```

Convolution in CNNs: **1 layer**



- C input feature maps of size HxW
- M output feature maps of size ExF
- M filters of size RxS

Convolution layer: **1 layer with input batch**

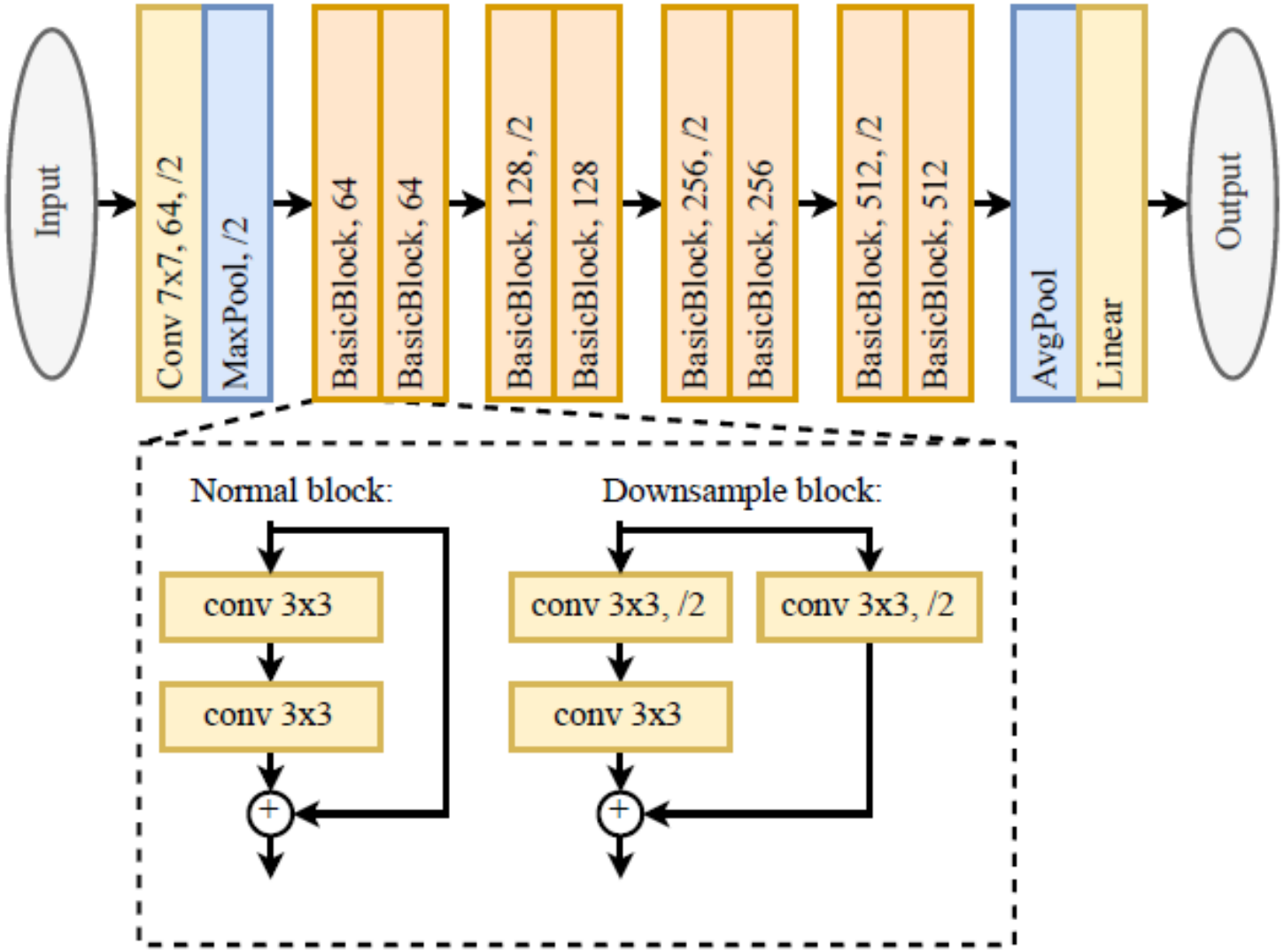


- N = batch size
- C input feature maps of size HxW
- M output feature maps of size ExF
- M filters of size RxS

- Note: for a fully connected layer: filter size = input size, RS = HW

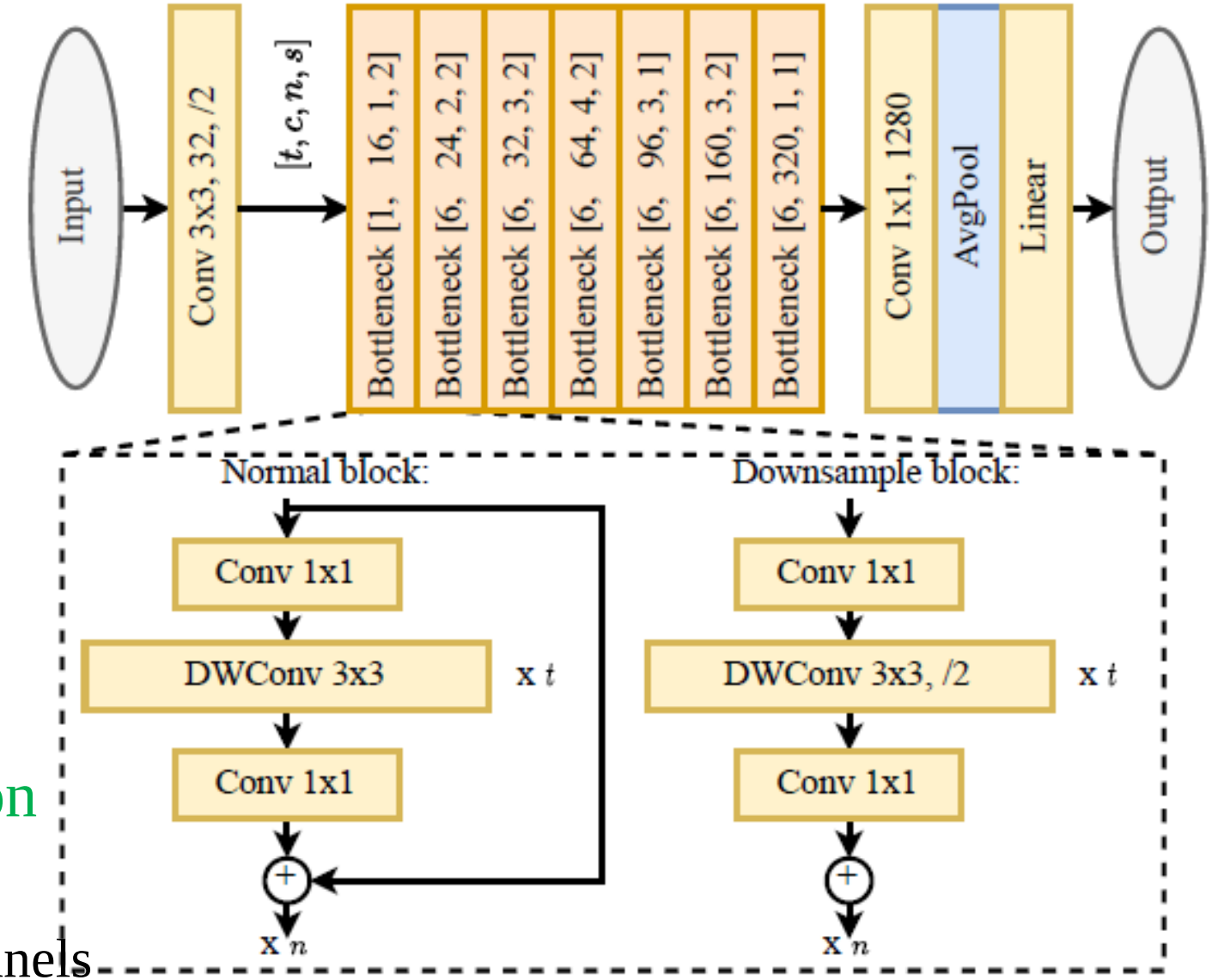
ResNet18

- Legend:
  - 7x7: filtersize
  - 64: output channels
  - /2 : stride 2
- 11.7 M weights
- Residual
  - makes training easier for deep networks
- ResNets often used as backbone network



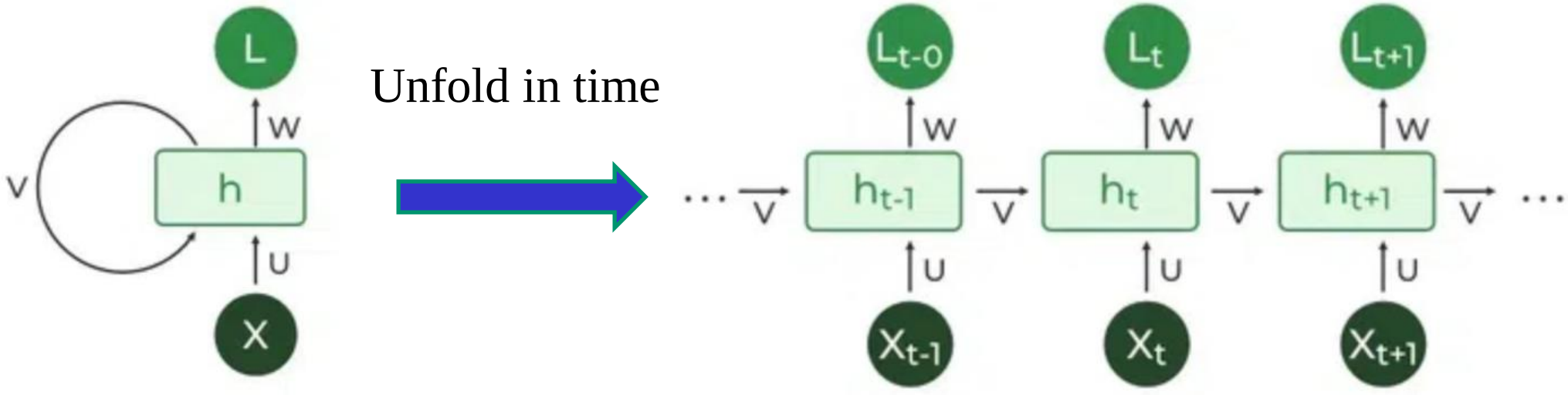
MobileNetv2

- Legend, e.g. 6, 64, 4, 2 :
  - 6: channel expansion before depth-wise conv.
  - 64: output channels
  - 4: repetitions of this block
  - /2: stride
- 3.4 M weights
  - cheaper alternative to ResNet18
- Depth-Wise separable convolution
  - 1x1 conv combines input channels
  - allows easy scaling of #output channels



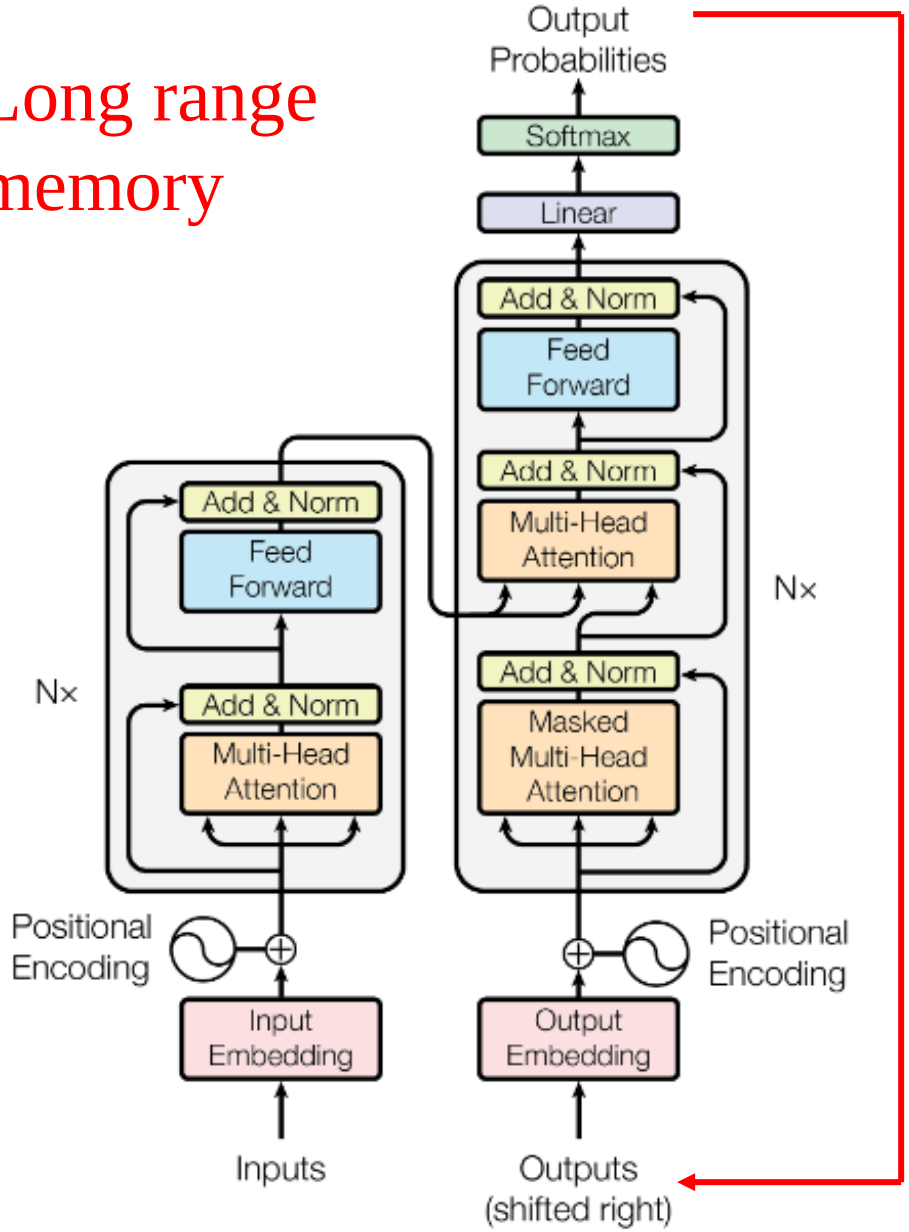
Stateful Networks

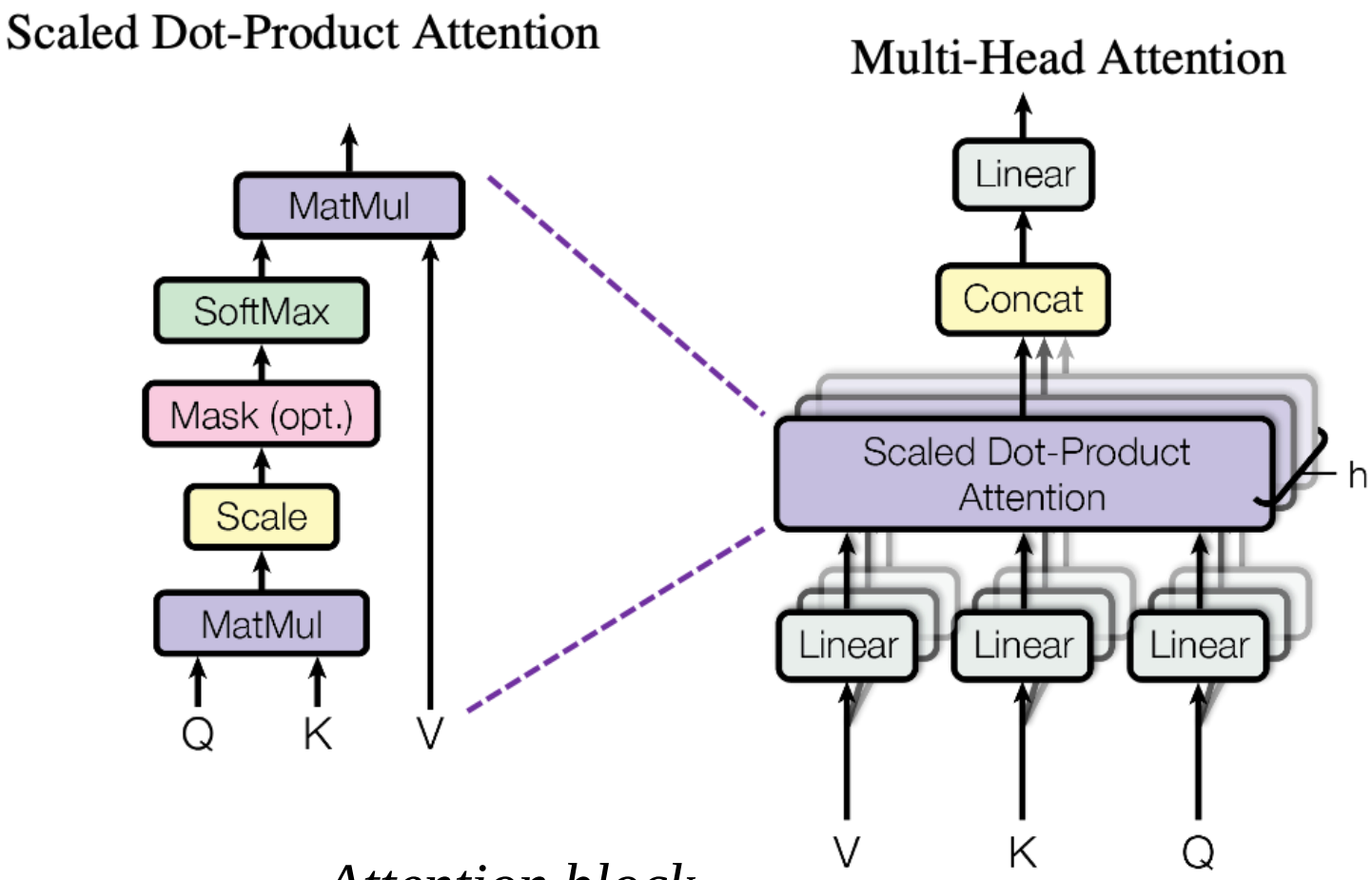
- Neurons have state
  - inside neuron: Spiking Neural Network: SNN
  - outside neuron: using Recurrent Connections: RNN, LSTM, GRU
- Creates a **short** term memory
  - … and thereby the notion of time ..



Transformer architecture: using self-attention

Long range memory





Attention block

- *inputs: Value V, Key K, Query Q*
- *h Heads*

What to expect?

- AI Deep Learning Models
- **Edge Mismatch:**
  - **Cloud vs Edge**
  - **Energy as key driver**
- Optimizations
- Learn from the Brain
- SOTA in Edge AI computing
- Future
- Conclusions

AI mostly in the cloud

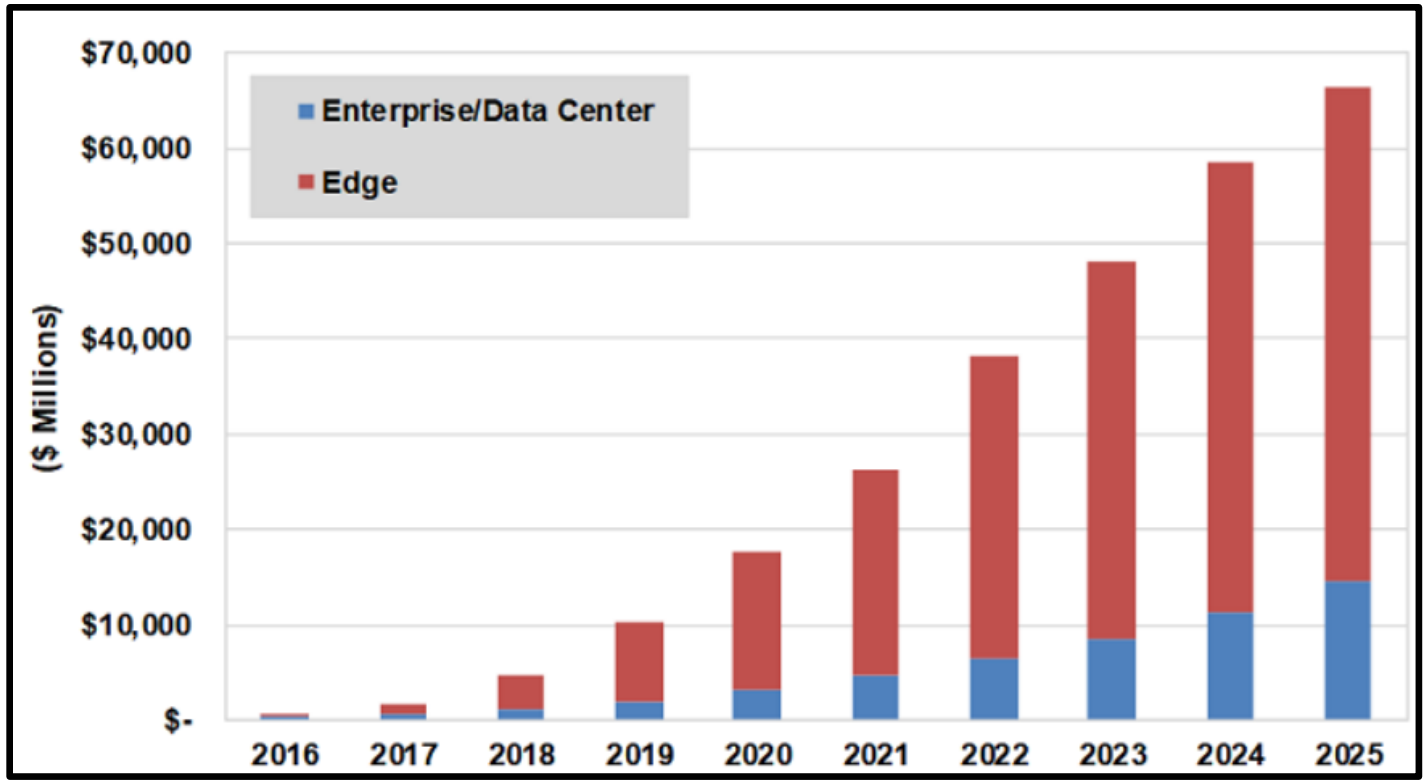
- Pre-Cloud: 1980 – 2005
  - data centers of limited scale
- Cloud transition 2006 – 2020
  - AWS (Amazon Web Services), Google Cloud, etc.
- Cloud only 2021 –
  - cloud optimized for many business cases, including AI
- Why AI @ Cloud
  - data, memory, compute power, (electric power), …
  - cost of training GPT4: ~5 months on 10,000 V100 = 7200 MWh
  - trained on 10 trillion words

AWS cluster

https://base10.vc/post/generative-ai-mission-critical/

Smart Edge?

- Smart applications powered by AI in almost every edge device
- Edge-AI μProcessor market expected to **grow beyond 70 Billion USD by 2026**
- However, currently AI mostly in the cloud:
  - 95% / 5% (2024)
  - 50% / 50% (2028)
    - *Alan Lee, DAC61, 2024*
- Urgent need for ultra low-power edge-AI processing !!



SemiConductor Engineering: AI Chip Architectures Race To The Edge

Cloud vs Edge

- Distributed Training
- Compression/Encryption
- Scientific Computing

- Data pre-processing
- On-device DNN optimization
- Federated Learning

- Google **TPU v5p**
  - 459 TFLOPs (bf16)
  - 95GB HBM2e at 2765 GBps
- NVIDIA Grace **Blackwell**
  E.g. B100
  - 700 W
  - 7/3.5/1.8 PFlops (16,8,4 bit operations)
  - *2 for sparse execution
  - 20/10/5 TOPs/W or 50/100/200 fJ/op

- Google **Edge TPUs**
  - 8 TOPs (int8)
  - 2 TOPs/W or 500 fJ/Op
- NVIDIA **Jetson** Series
  - 0.5 – 275 TOPs
  - 5 – 60 Watts
  - ~ 4 TOPs/W or 250 fJ/Op

- https://cloud.google.com/tpu/docs/v5p
- https://www.nvidia.com/en-us/data-center/gb200-nvl72/
- https://coral.ai/docs/m2-dual-edgetpu/datasheet/ =
- https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

What is the Edge Budget : Smart glasses

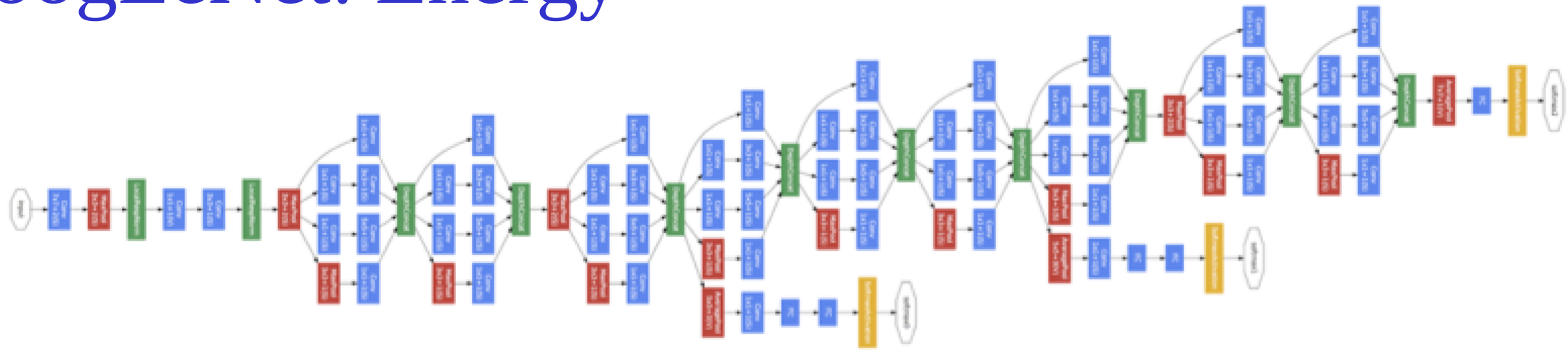
Assumptions:

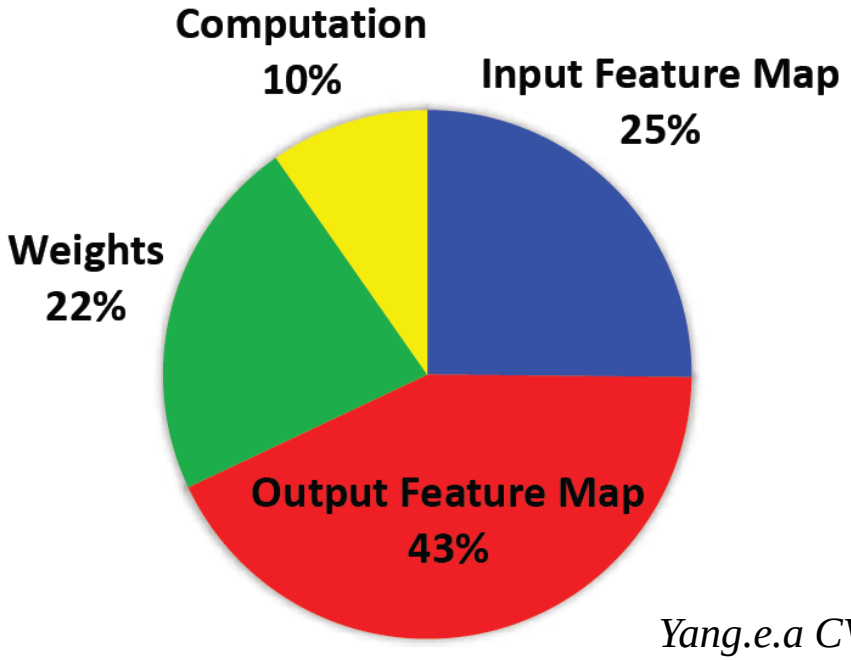
- 2 stereo cameras @ 30 fps, 1280x720x3 (RGB)
- Resnet-50/frame (200 GOPs/frame - 32-bit batch size of 1, underestimation!)
- I only have 10mW!
- Note: coin cell CR2032 225 mAh x 3V => runs less than 3 days

Required:

- Need 60 inferences/second (2 cameras x 30frames/second)
- (200 GOPS/frame x 60 inferences/sec.) / 0.01W = 1200 TOPS/W or < 1 fJ/op

GoogLeNet: Energy





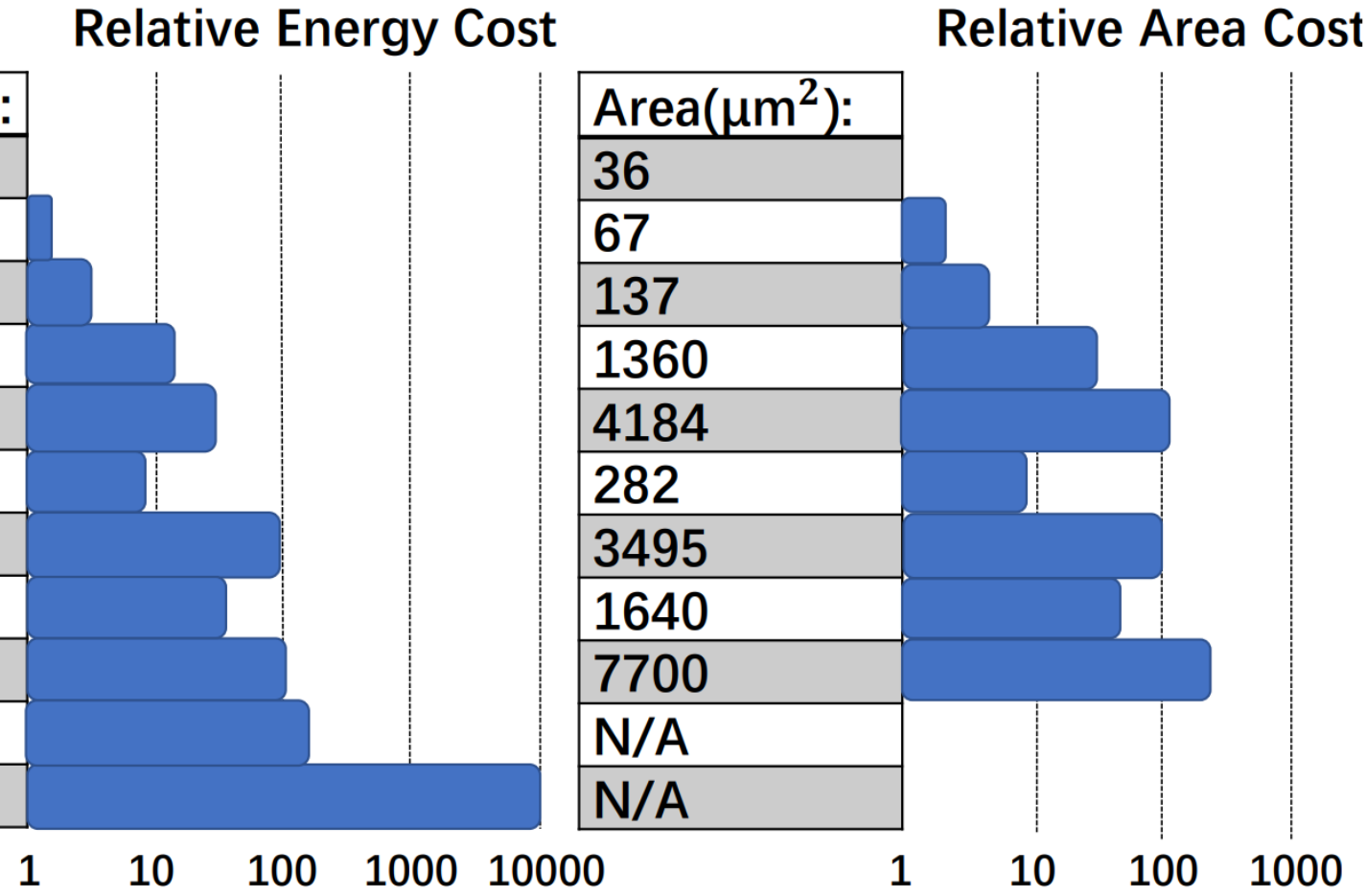
Yang.e.a CVPR'17

Convolution
Pooling
Softmax
Other

- CNN layers: 21 depth, 57 in total
- FC layer: 1
- Weights: 7.0 M
- MACs: 1.43 G per input

Where is the energy consumed?

Operation:	Energy(pJ):	Area(μm²):
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8kb)	5.0	N/A
32b DRAM Read	640	N/A



45nm, 0.9V

Image source: A Survey of Quantization Methods for Efficient Neural Network Inference – A. Gholami et al. 2021 / adapted from Horowitz, ISSCC 2014

What to expect?

- AI Deep Learning Models
- Edge Mismatch: Cloud vs Edge
- **Optimizations**
  - **Pruning, Quantization, Data reuse**
- Learn from the Brain
- SOTA in Edge AI computing
- Future
- Conclusions

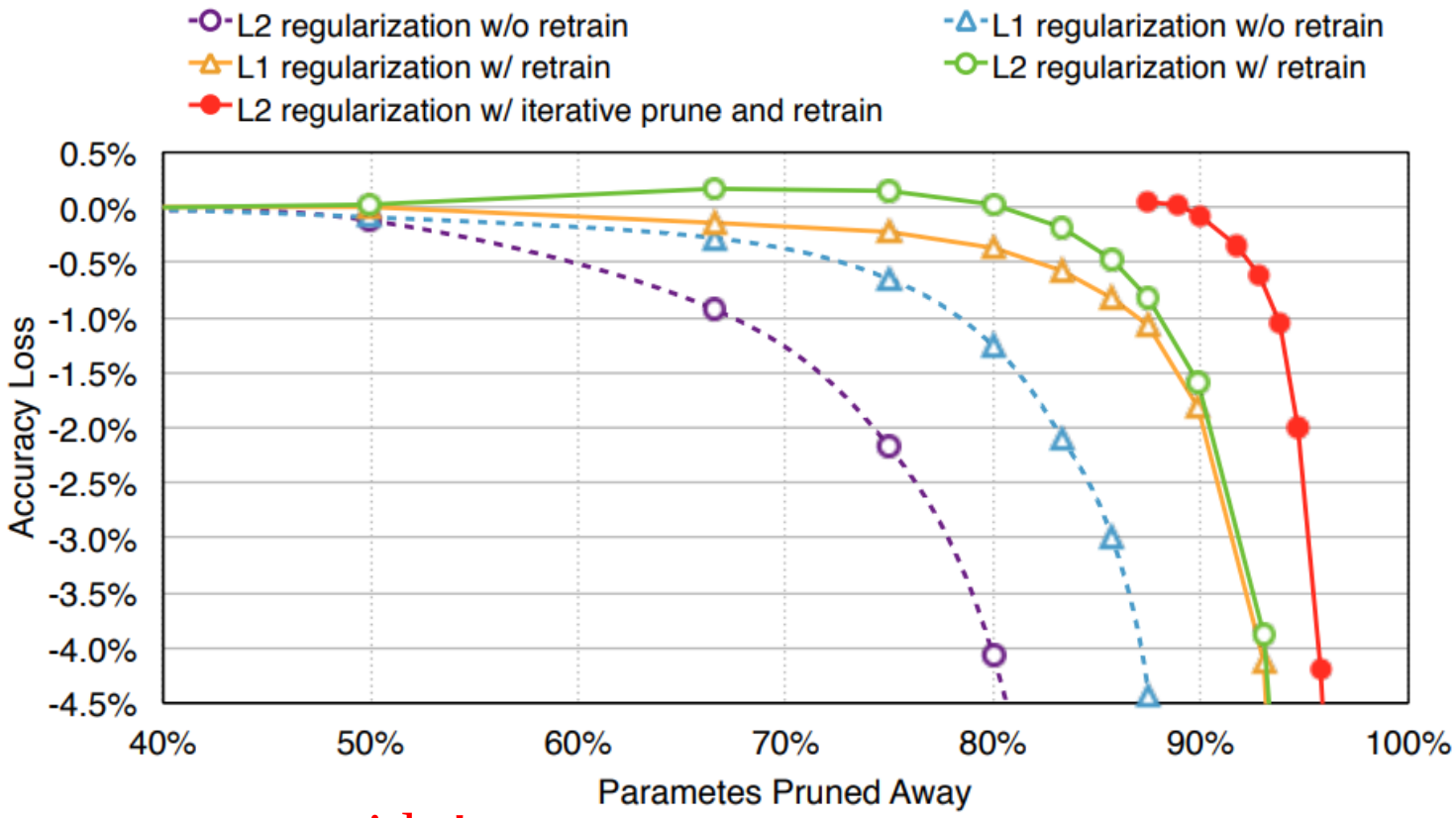
Optimizations for high energy-efficiency

- Model transformations
- Pruning & Sparsity
- Quantization
- Data reuse (activations and weights)

- Model exploration: NAS, HW-aware NAS
- Mapping exploration: ZigZag, Stream, Timeloops

Iterative Pruning and Re-training: Alexnet

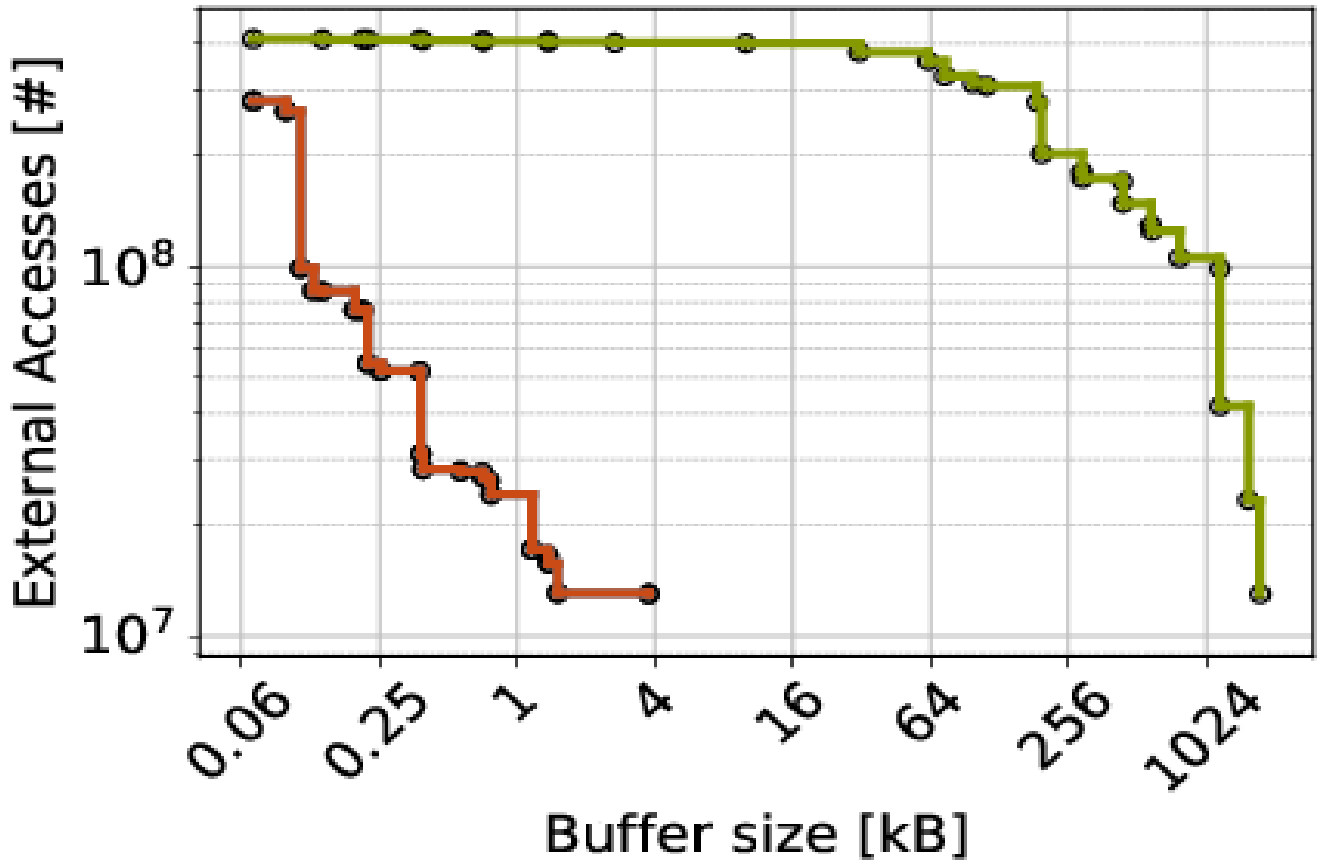
- Re-training once to recover from pruning damage
  - L2 regularization performs better
  - 85% pruning of parameters
  - **6.7x model reduction**
    - From 61M to 9.1M Parameters
    - Or 233MB to 34.7MB
- Iterative pruning even better
  - Without loss of quality prune 90%
  - **10x model reduction**
    - From 61M to 6.1M
    - Or 233MB to 23.3MB



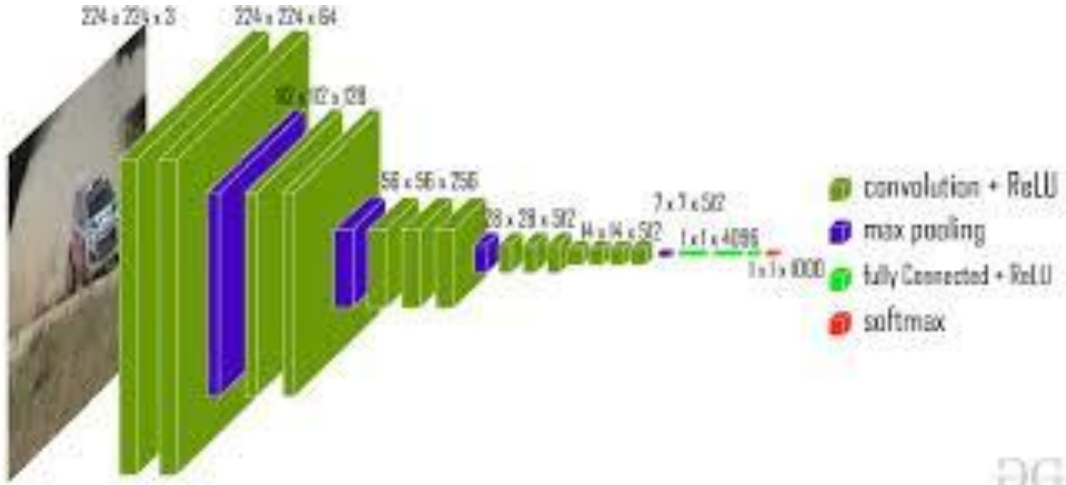
- **Catch**:
  - Alexnet had far too many parameters to start with !
  - 10x does not translaten necessarily into energy savings!

Impact of data reuse: Reducing ext. memory accesses

Original code
Rescheduled code

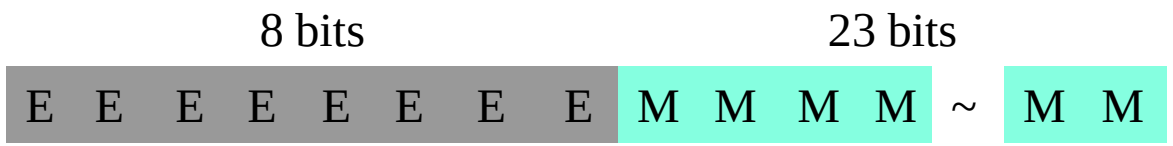
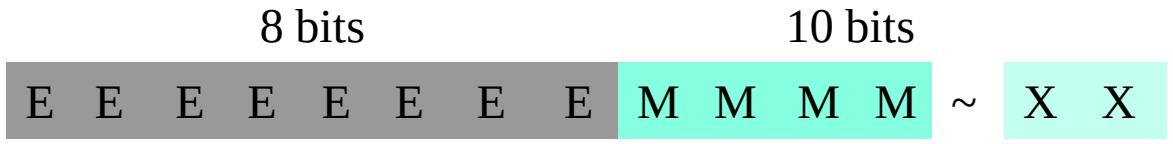
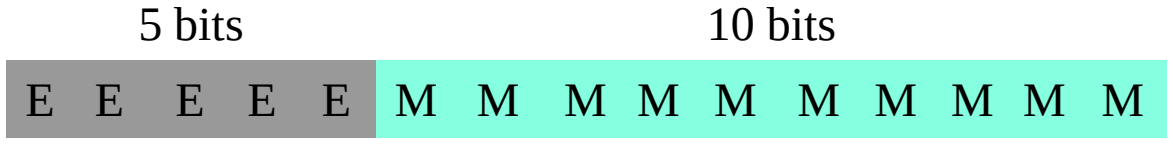
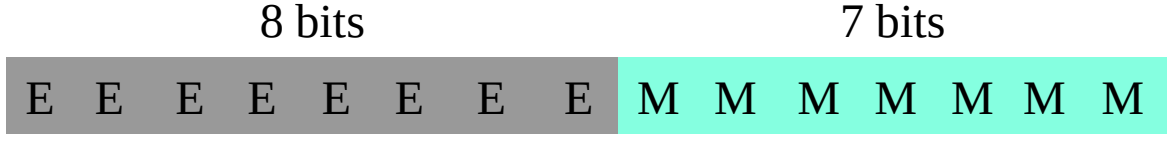
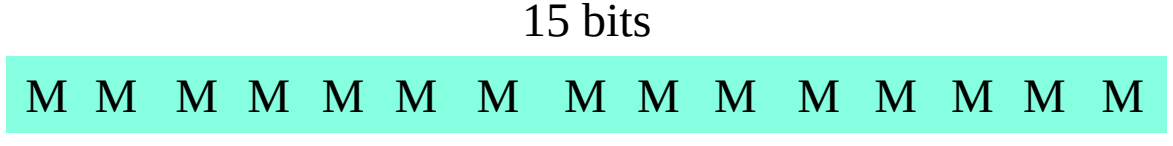
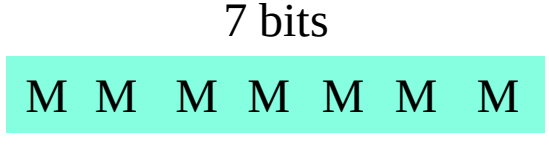


VGG16 example:



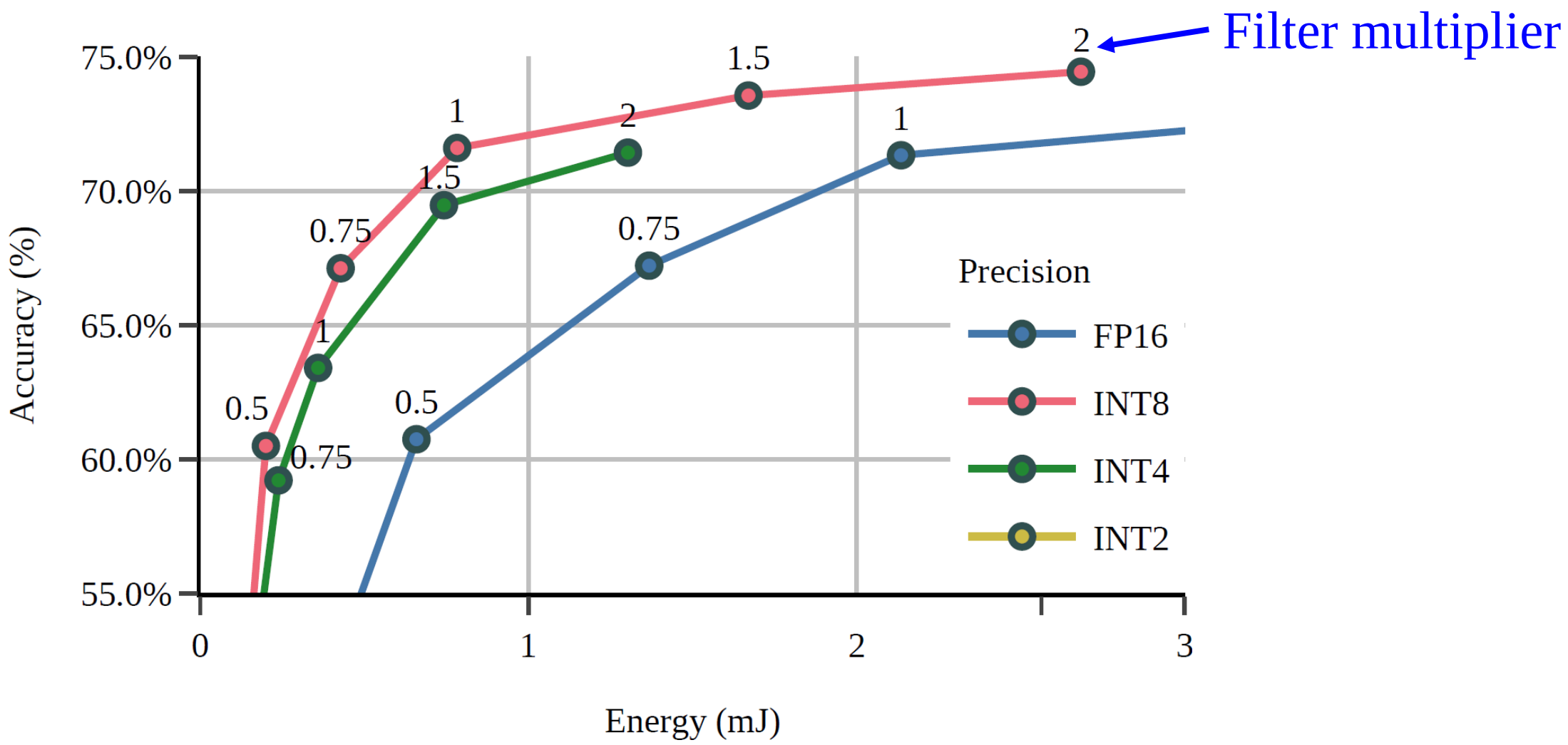
Conclusion: we need advanced loop transformation to exploit data locality (in local buffers), reducing external accesses

Quantization: Data formats used in NNs

Number formats		Dynamic Range	Relative Precision
float32		$1e^{-38}$ to $3e^{38}$	$6e^{-6}$%
tensorfloat32		$1e^{-38}$ to $3e^{38}$	0.05%
float16		$6e^{-5}$ to $6e^{5}$	0.05%
bfloat16		$1e^{-38}$ to $3e^{38}$	0.4%
integer16		1 to $3e^{4}$	1 (abs.)
integer8		1 to 127	1 (abs.)

What's next: see ***OCP Microscaling Formats (MX) Specification Version 1.0***

How far to quantize: Accuracy vs Energy?



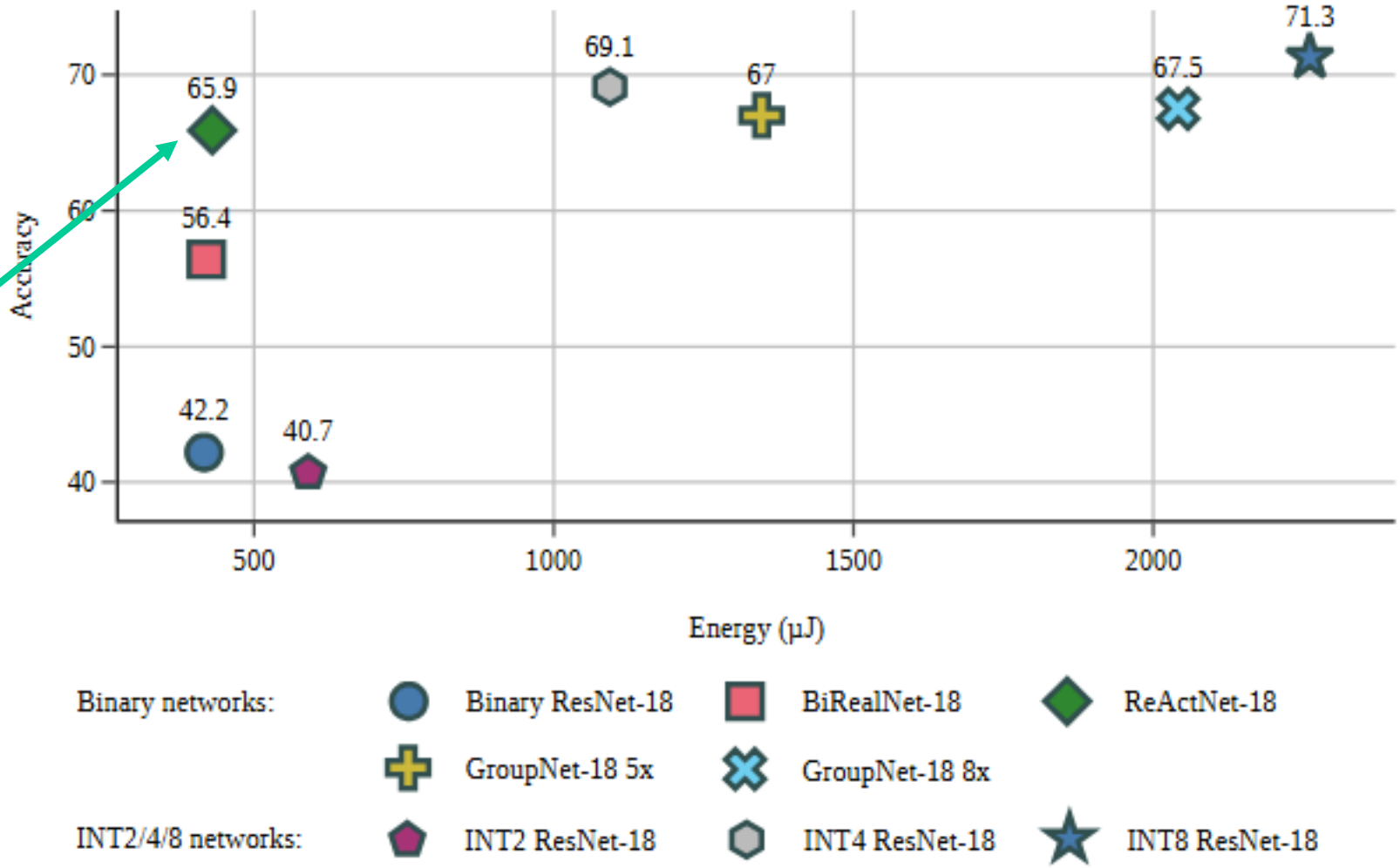
Accuracy-Energy trade-off of MobileNetV2 on Meta HW. The color represents the precision level, while the filter multiplier is specified above each data point. Note that INT2 data points fall outside the range of this figure.

How far to quantize? Repair BNNs

Another study:

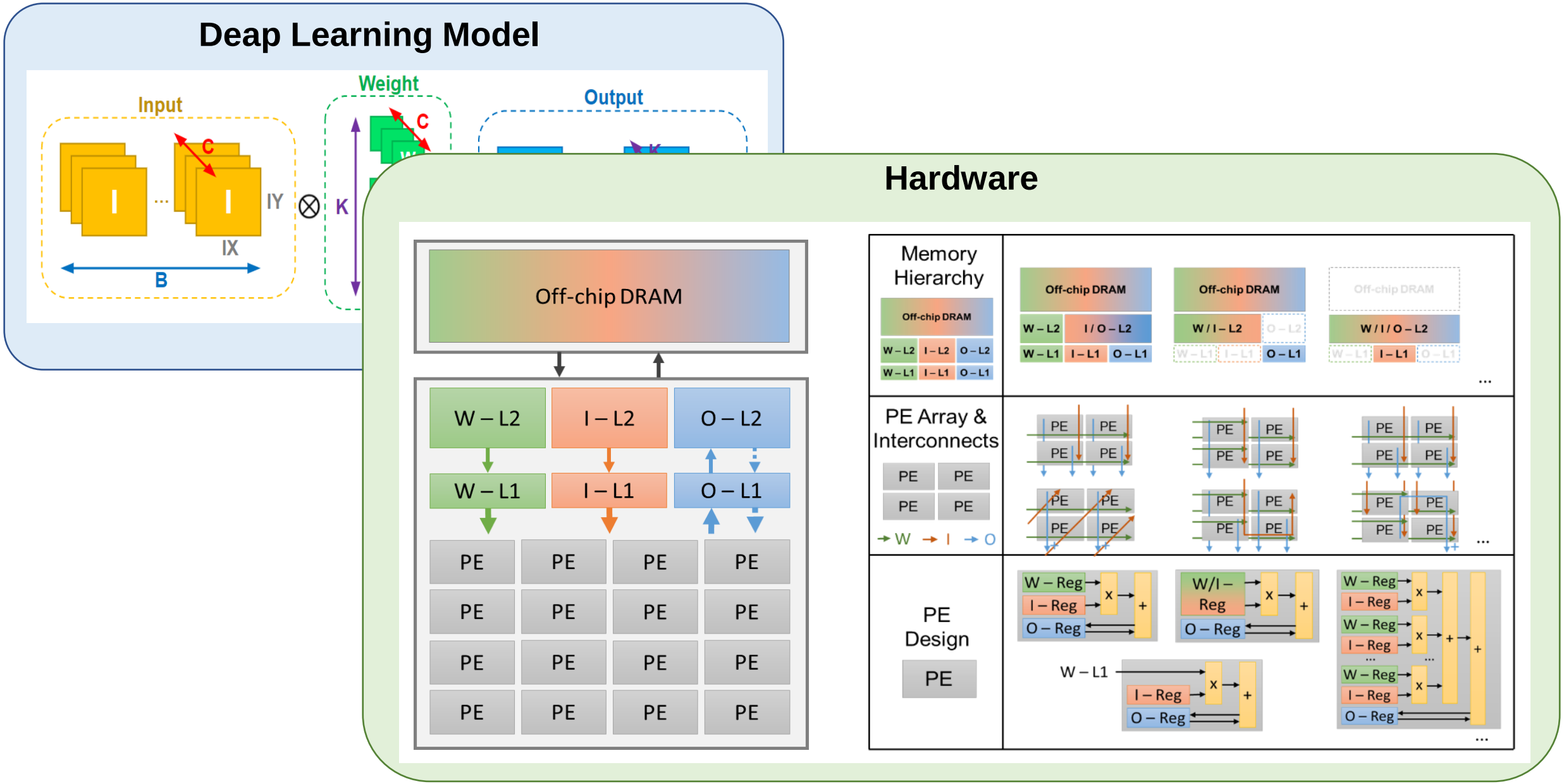
- Comparing 1,2,4 and 8-bit on ResNet-18 variants

- With repair mechanisms BNNs get much closer to 8-bit
  - e.g. ReActNet
  - https://arxiv.org/abs/2003.03488

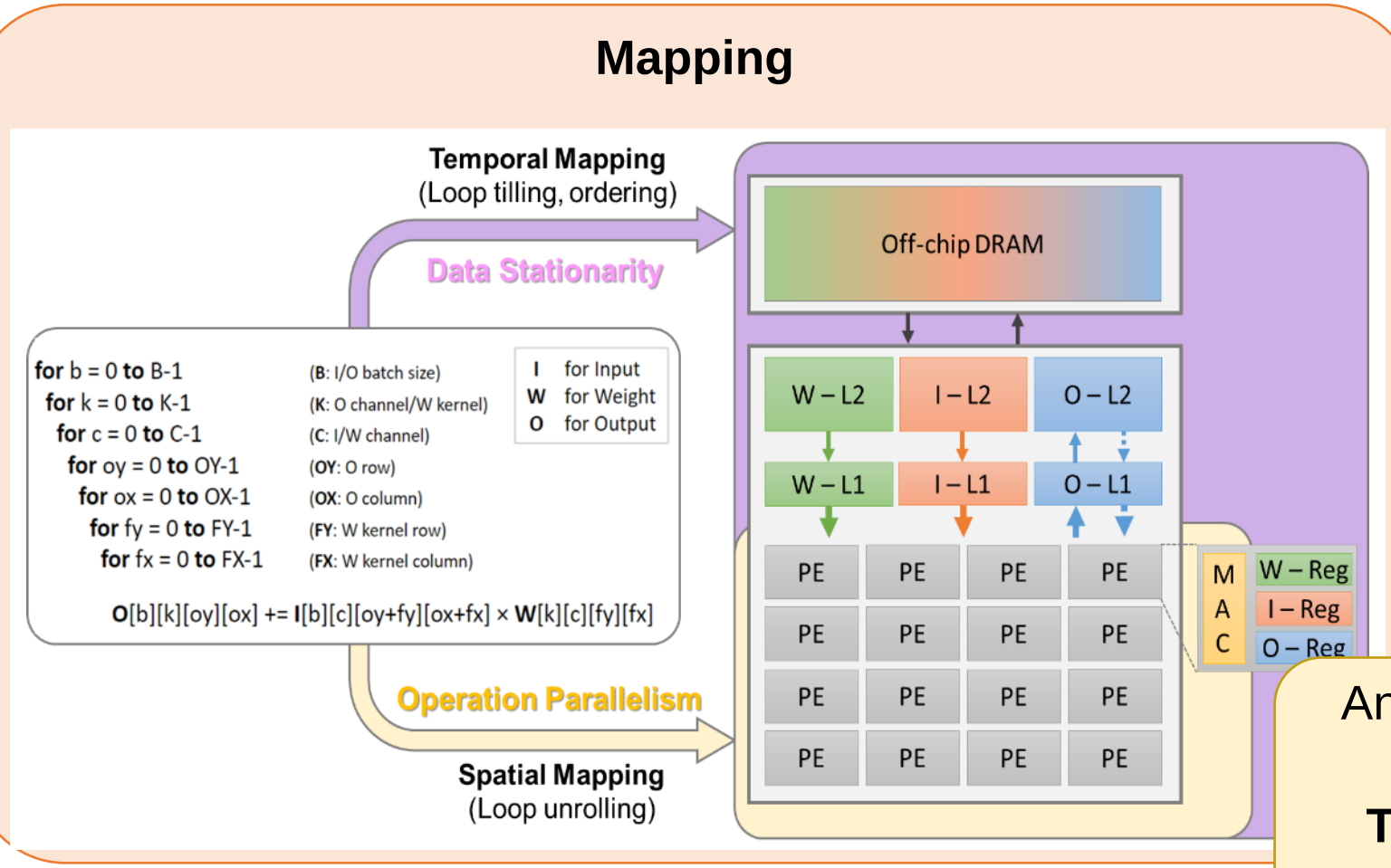


Floran de Putter e.a. Quantization, how far should we go?, DSD 2022

DSE of mappings: ZigZag *(KULeuven)*



DSE of mappings: ZigZag *(KULeuven)*



Analytic model - Huge Design Space

Technology: 65nm/40nm/28nm/…, NVM, CIM, 3D IC, etc.

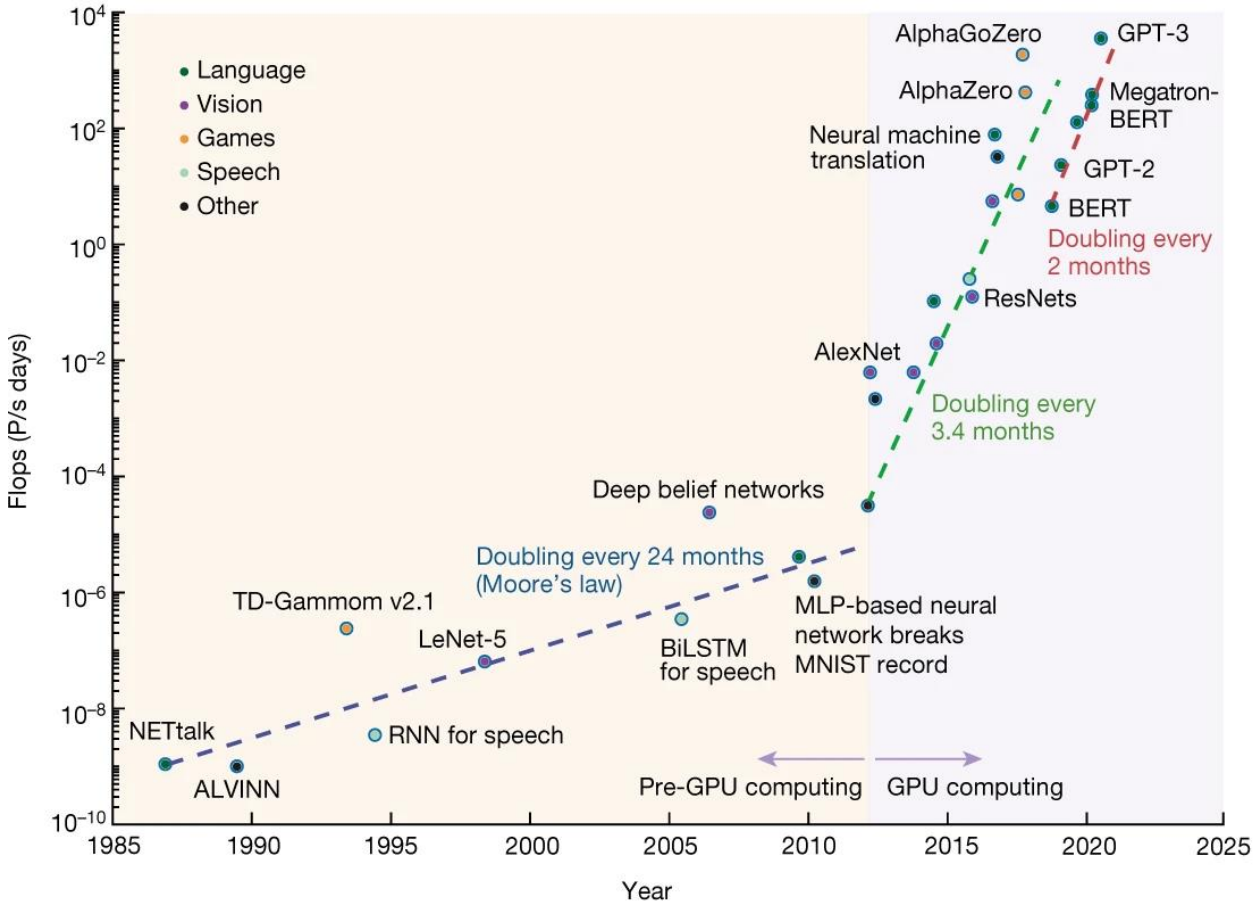
Others: Sparsity, Quantization, Layer Fusion, etc.

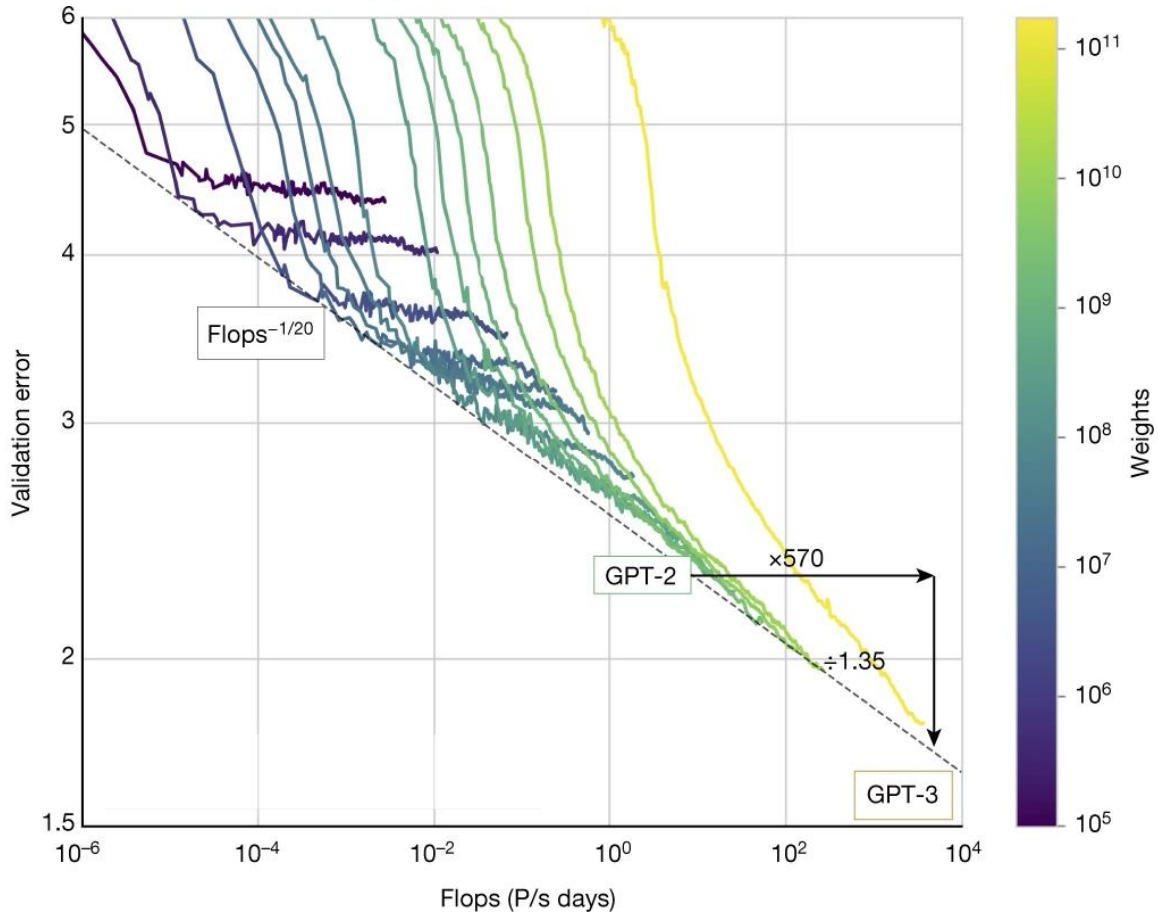
What to expect?

- AI Deep Learning Models
- Edge Mismatch: Cloud vs Edge
- Optimizations
- **Learn from the Brain**
  - **3 lessons**
- SOTA in Edge AI computing
- Future
- Conclusions

Deep Learning seems to have no limits

K. Boahen, Nature 2022





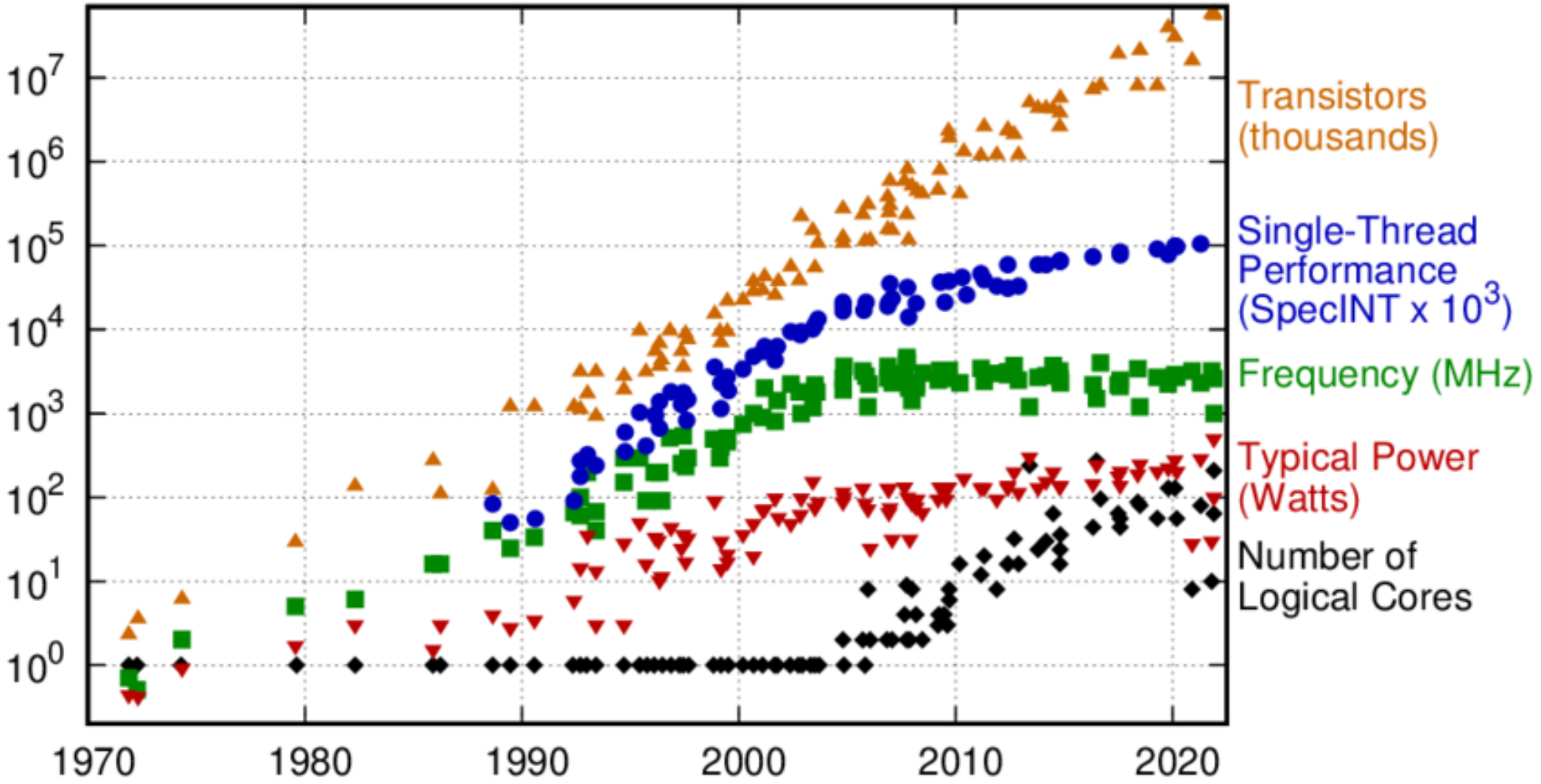
Energy expensive.

Training - GPT-3: 1.29 GWh

- GPT4: 50 GWh => 40x

Diminishing returns in deep learning (i.e., scaling follows a power law)

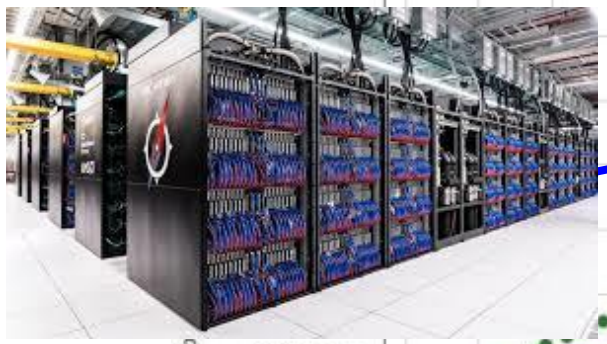
HW has its limits: 50 years of Microprocessors



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

Brain, any limits?

- Speed = 1000 Exa Op/s*
- Power = 20 Watt
- Energy/operation = Power / Speed = $2x10^{-5}$ fJ/operation

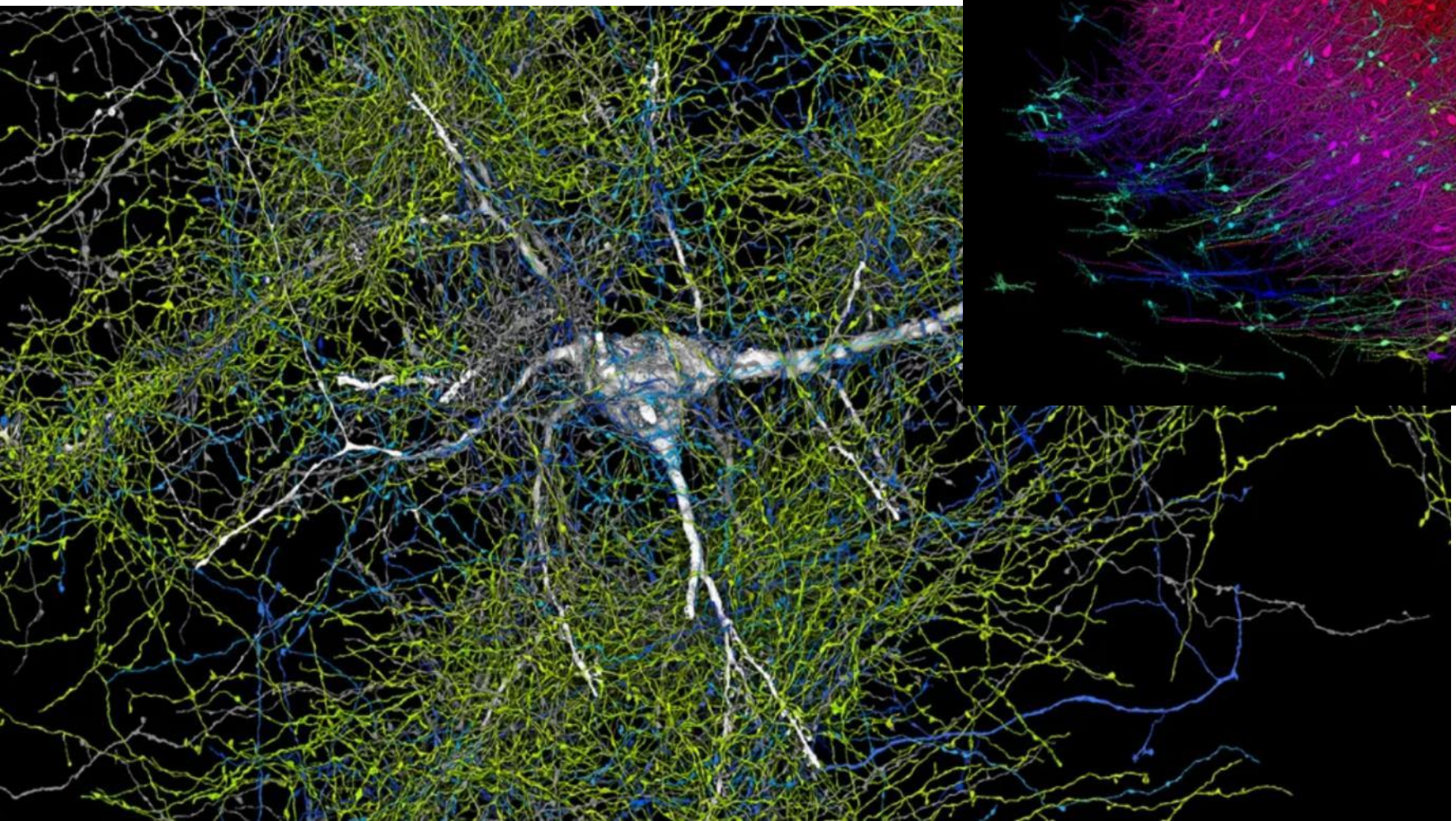


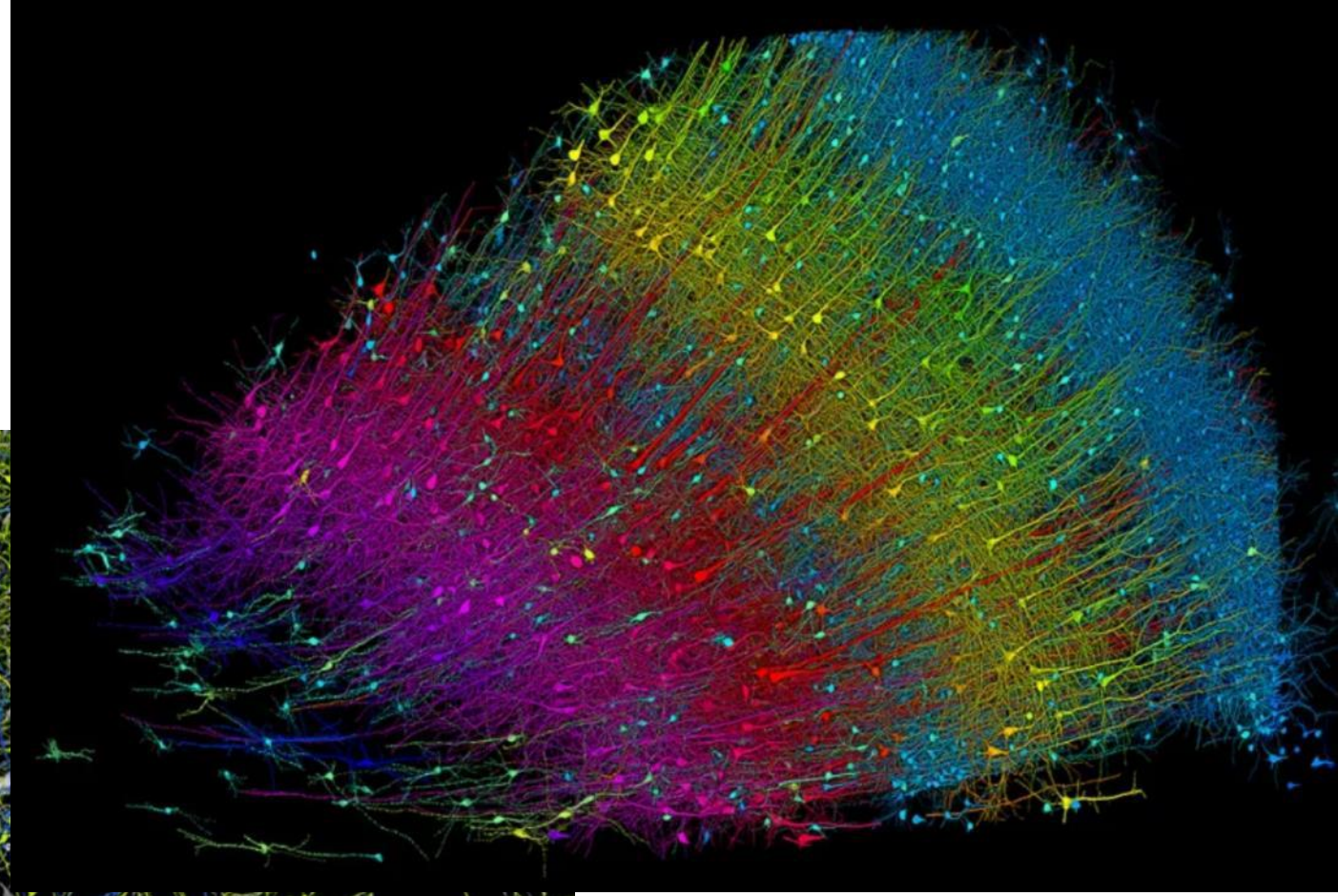
- Compare to **Frontier**
  - Power = 22.7 MW
  - Peak = 1.206 ExaFlop/s
  - 8.7 Mcores, 680 m^2
  - Energy/op. = 19 pJ/op

**Tim Dettmers:*
"making deep learning accessible"
2015

Inside our brain

Google Research & Lichtman Lab
Harvard University





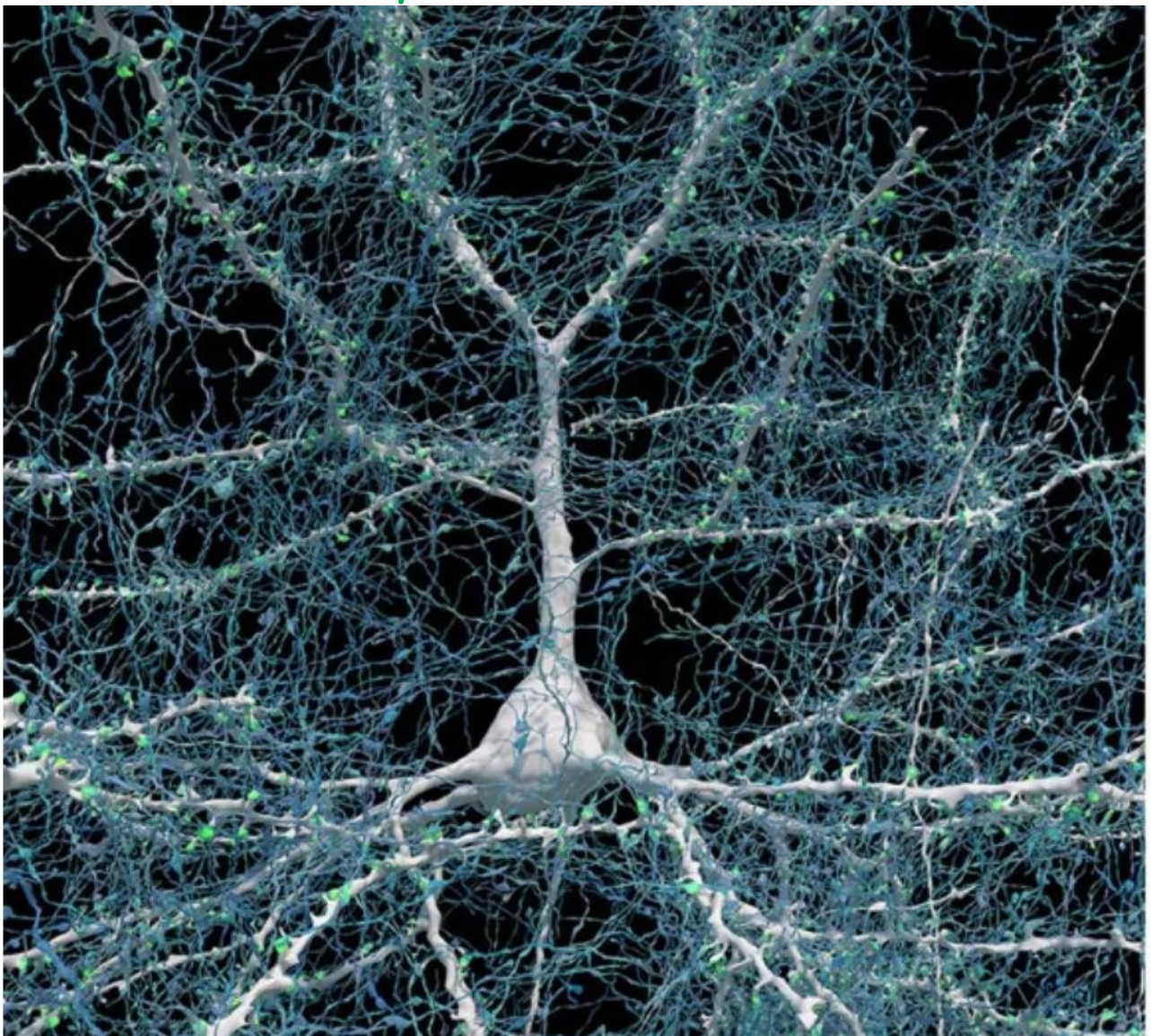
Excitatory neurons
- colored by depth
- blue=outside surface

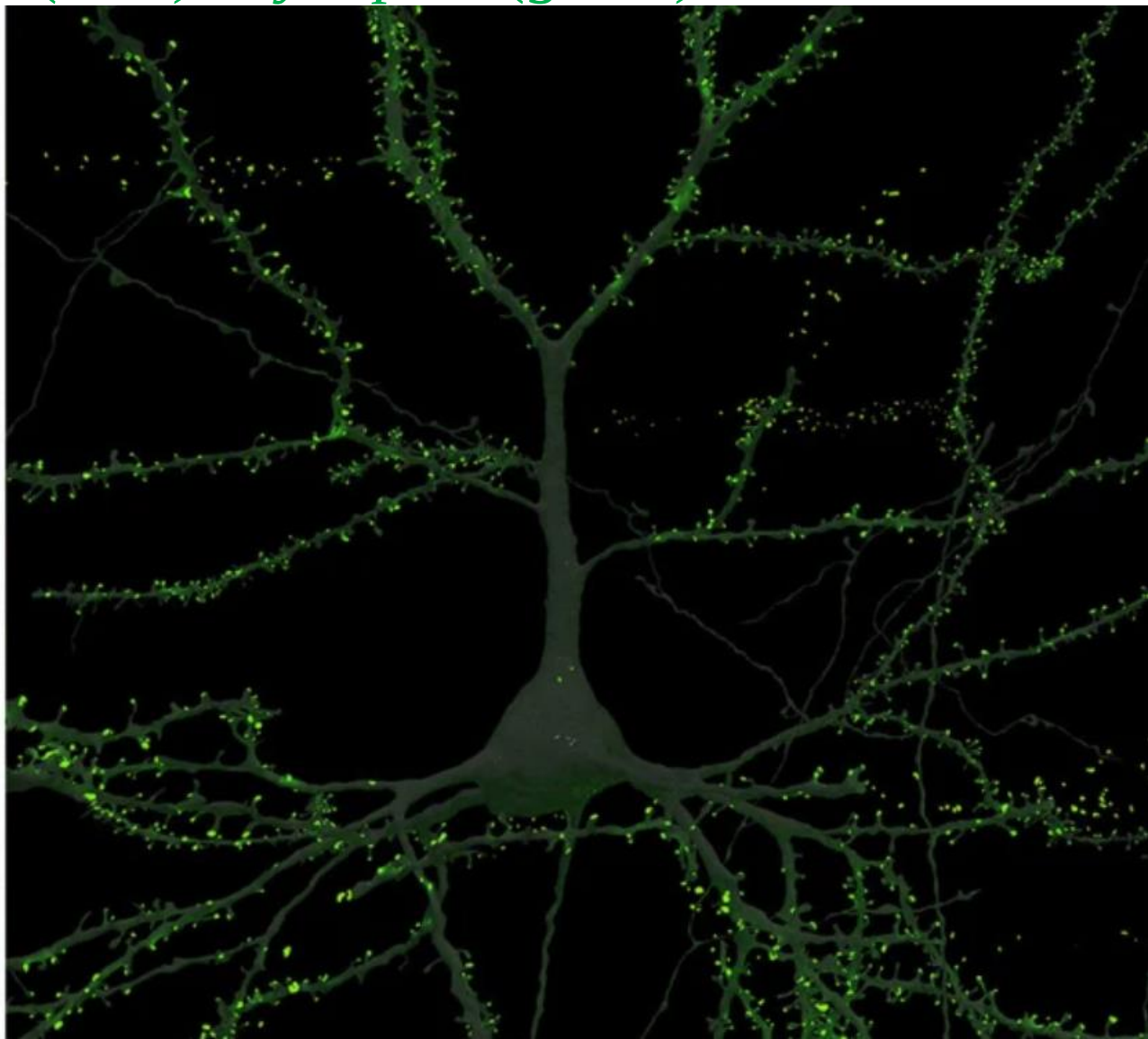
1 neuron:
- *green excitatory*
- *blue inhibatory*

Inside our brain

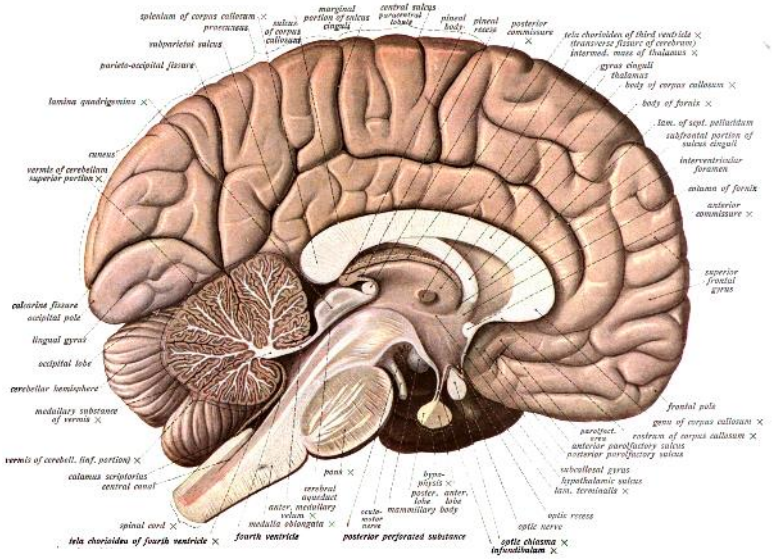
Google Research & Lichtman Lab
Harvard University

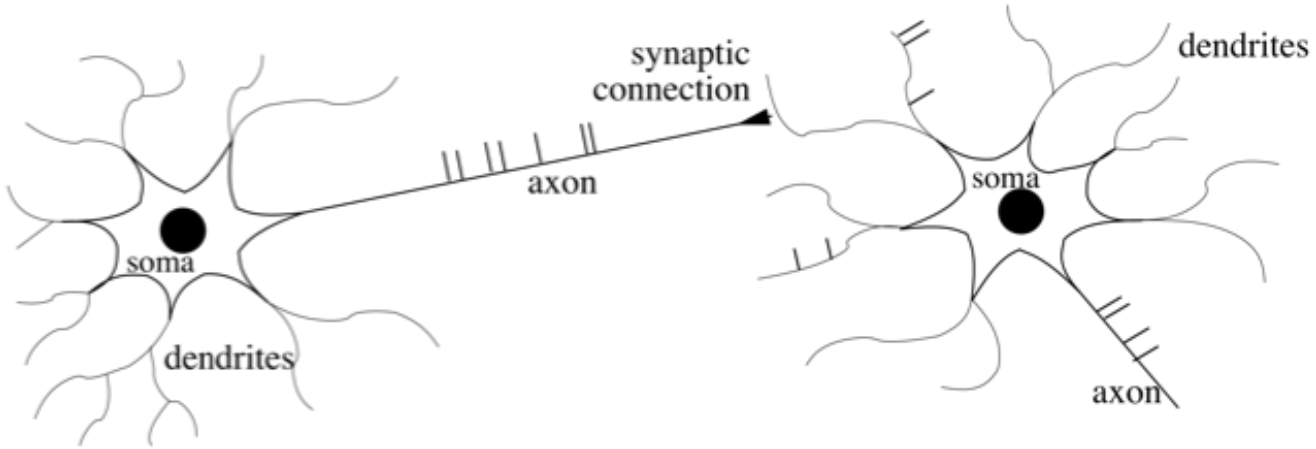
Neuron 14 μm / 5000 axons connect to I (blue) / synapses (green)

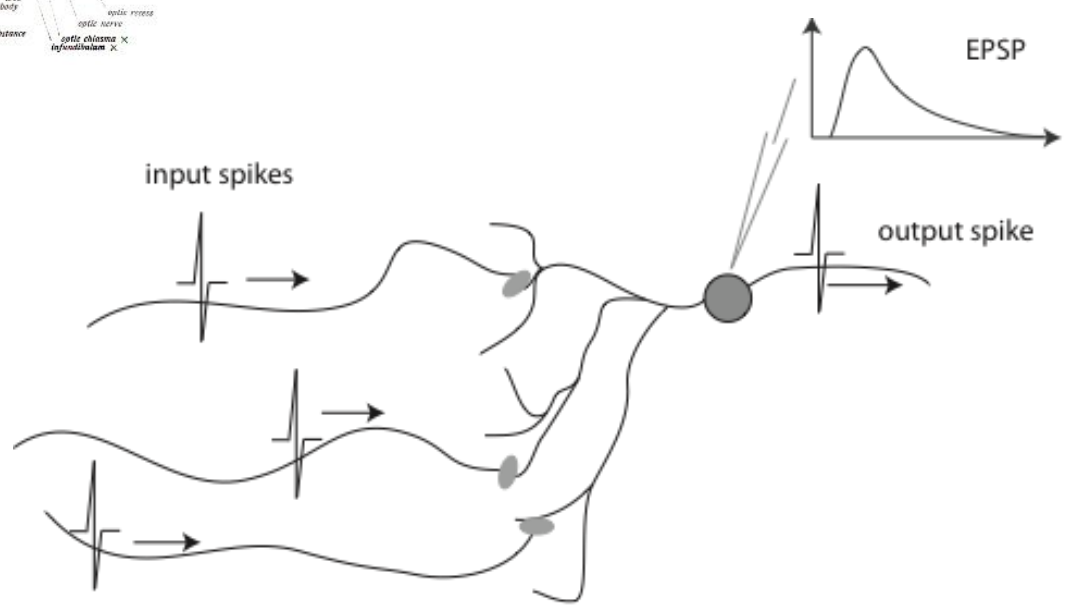


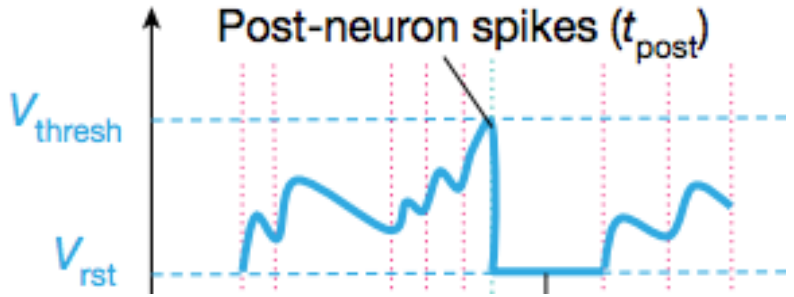


Learn from our brain (1): Spiking Neurons









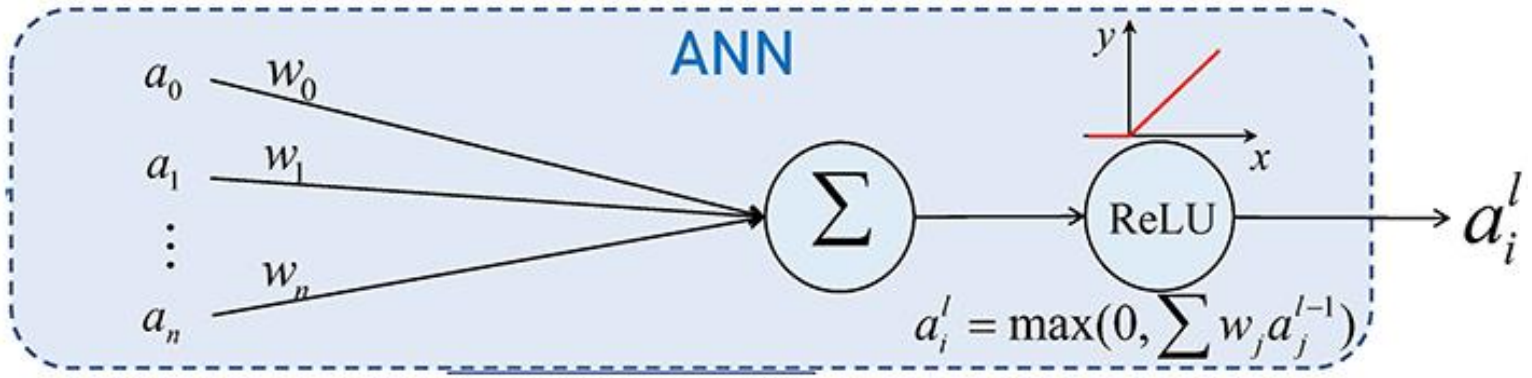
Spiking Neural Network: **SNN vs ANN neuron**

SNNs have 2 key properties:

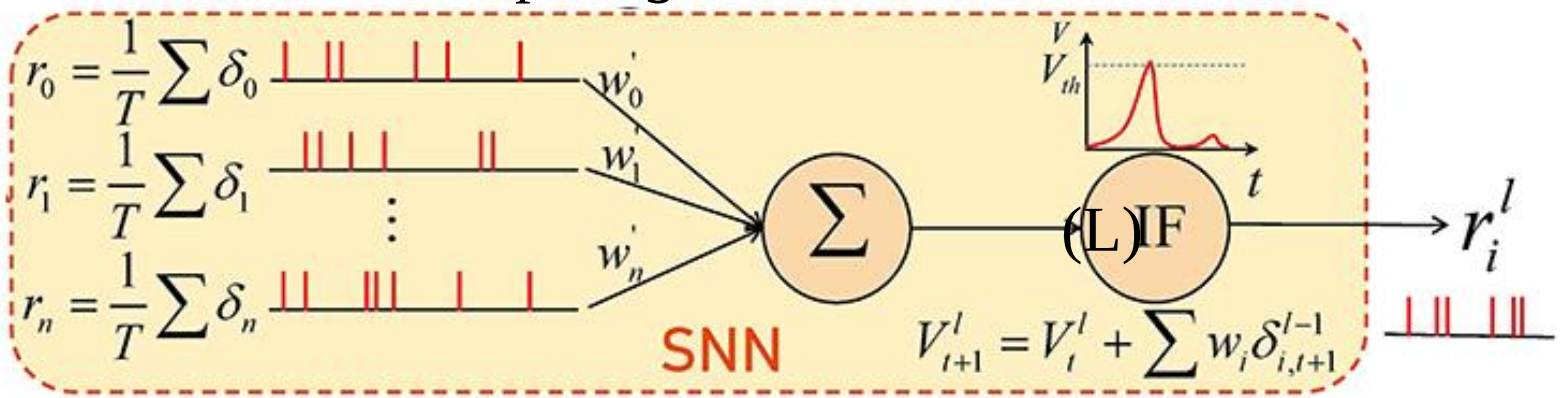
1. Neurons exchange spikes
   - Sparse spike input & output

2. Neurons have state (V)
   - State evolves over time
     - Increases when spikes enter
     - Decreases otherwise (exponential decay in time)
   - Execution is time dependent

Traditional Neuron



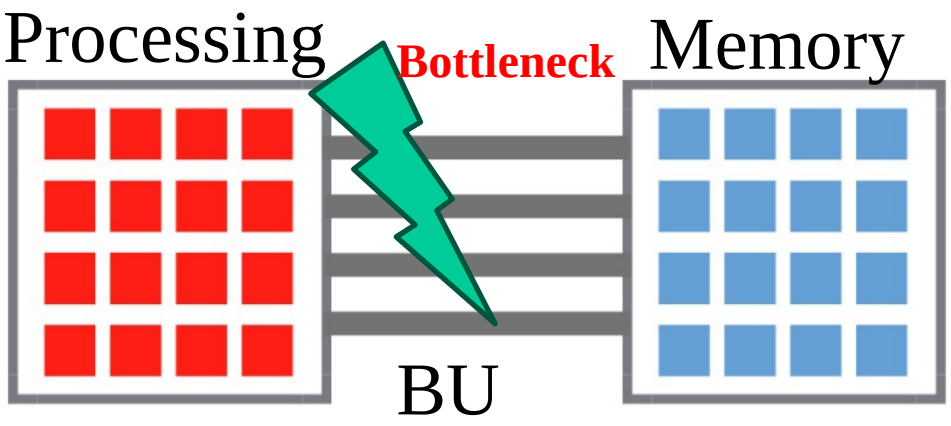
Spiking Neuron



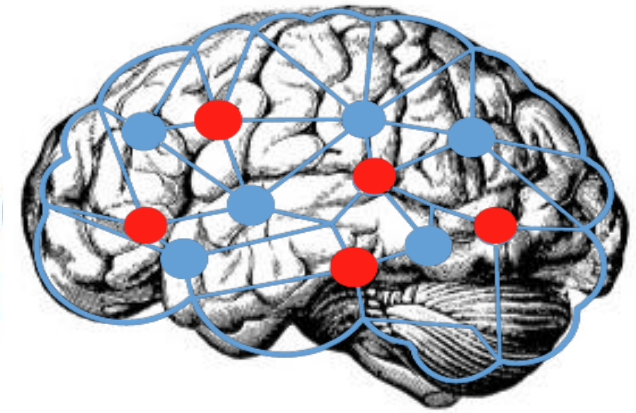
(L)IF = Leaky Integrate & Fire

Learn from our brain (2)

Von Neumann Architecture: energy inefficient



Brain architecture: highly energy efficient



Processing Units/Neurons

Memory/Synapses

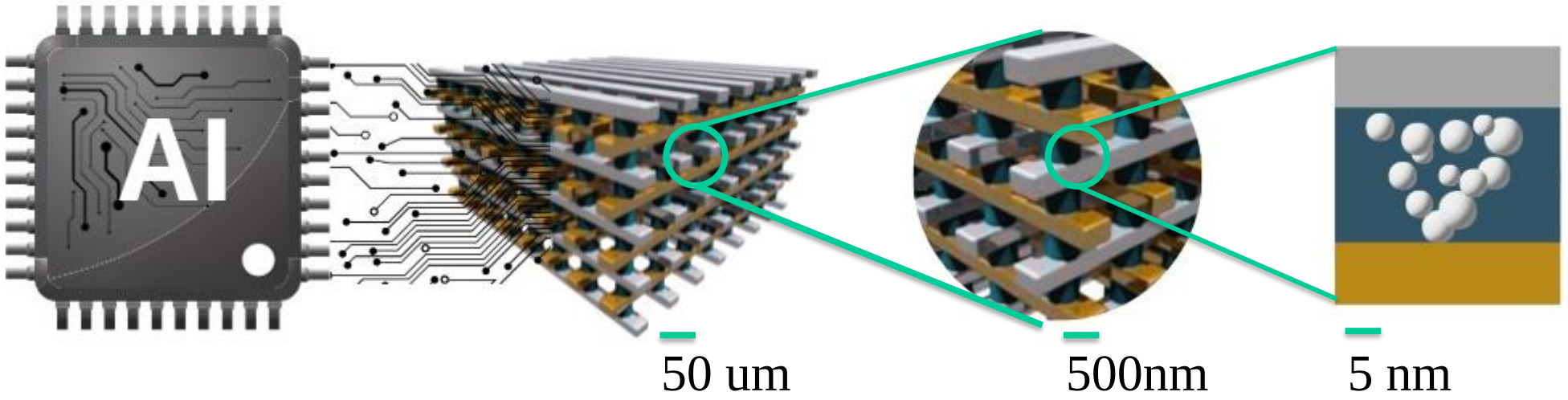
Simulating the brain ~20 MW

Valle Solar Power Station (Spain)

Less than 20 W

Solution: Compute in Memory (CIM)

Compute in Memory architecture with memristors (ReRAM)



Solving the energy efficiency **bottleneck**:

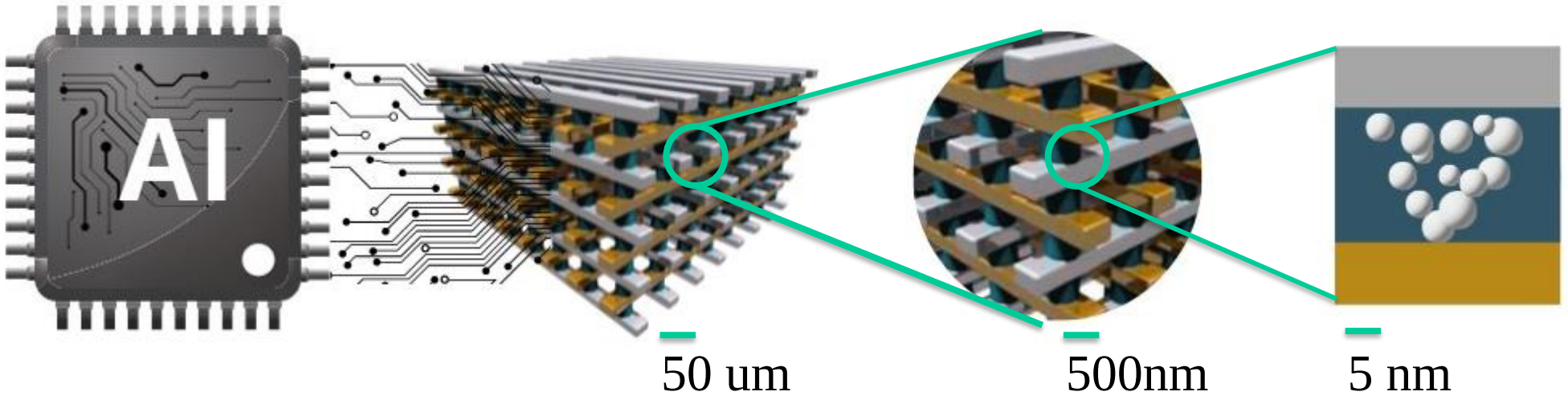
- Compute In-memory (**CiM**)
- Enabled by emerging device technologies (**ReRAM/FeFET**)

Potential Energy Gains: up to **100X**

- 5X don’t move data CiM/no cloud
- 5X using ReRAM/FeFET
- 2X event-driven computation
- 2X near threshold computing

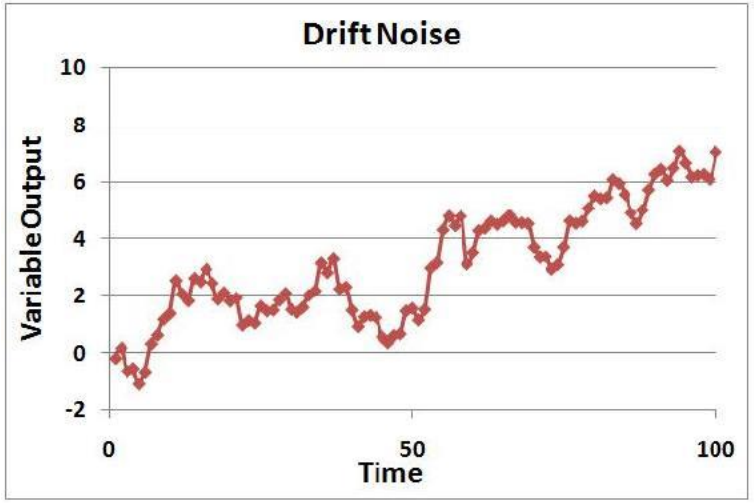
Issues of new technologies

Compute in memory architecture with memristors (ReRAM)



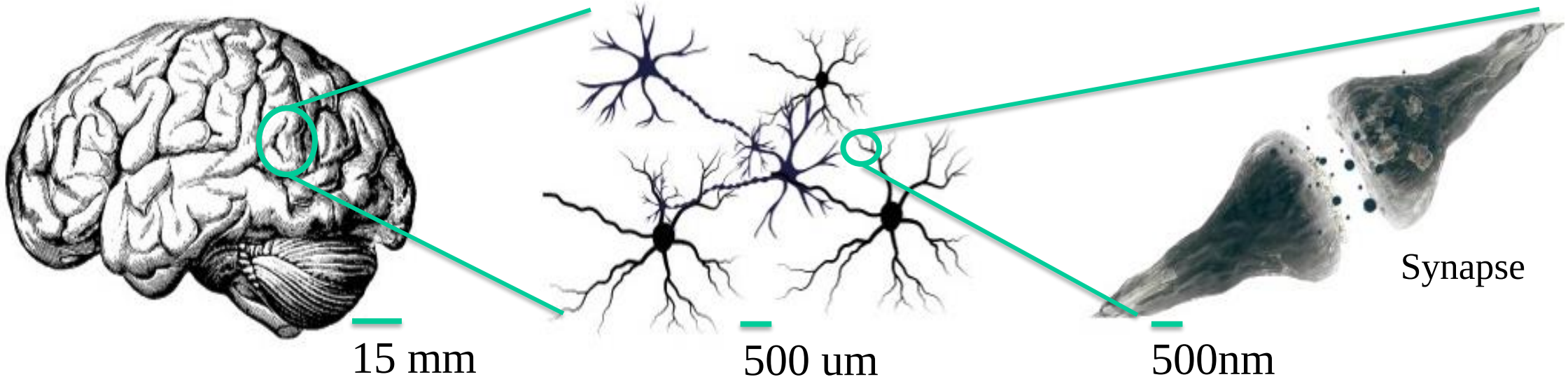
ReRAM/FeFET issues:

- Accuracy of computing
- Noise, Drift
- Device's lifetime



Learn from our brain (3)

Brain-inspired compute in memory architecture with self-healing at different levels

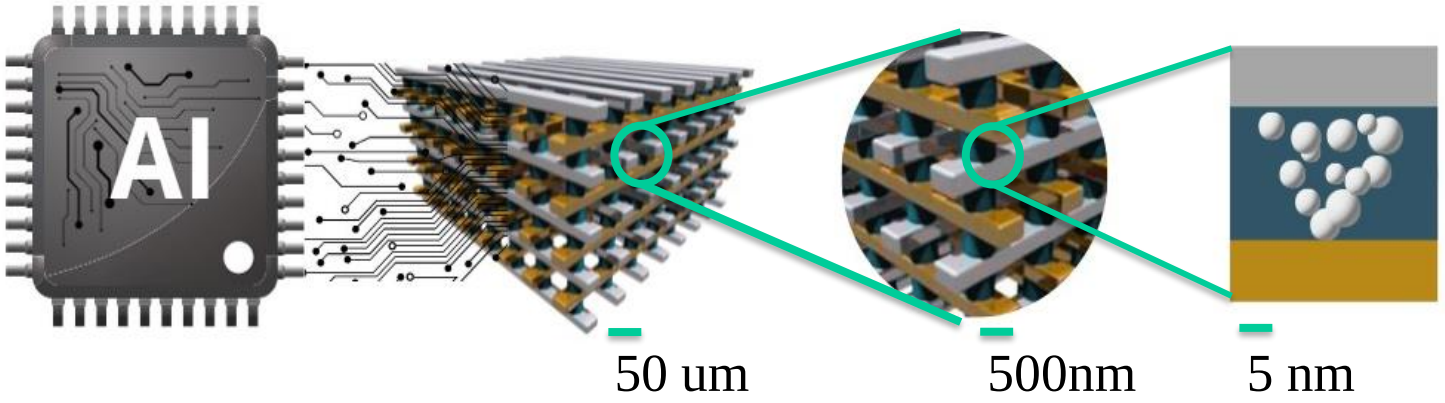


Use Self-Healing:

- Re-learn / Repair Synapses
- Remap functions
- Redundancy

Solution: Self-Healing

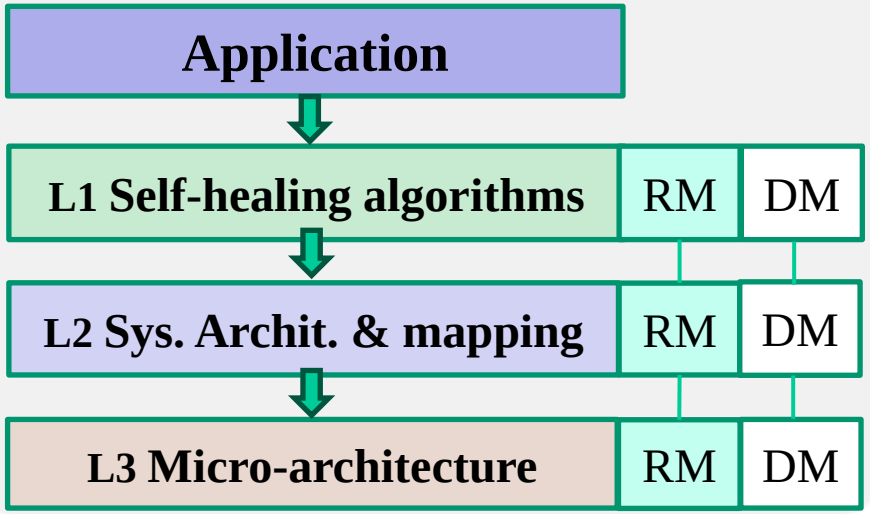
Compute in-memory (CiM) architecture with memristors (ReRAM/FeFET) & self-healing



Self-Healing is the Key; @ all design levels

- Adapt neural network (L1)
- Remap functionality (L2)
- Exploit Redundancy (L3)
- to compensate for
  - ReRAM/FeFET issues
  - Changing environments

DMs: Detection Mechanisms
RMs: Recovery Mechanisms

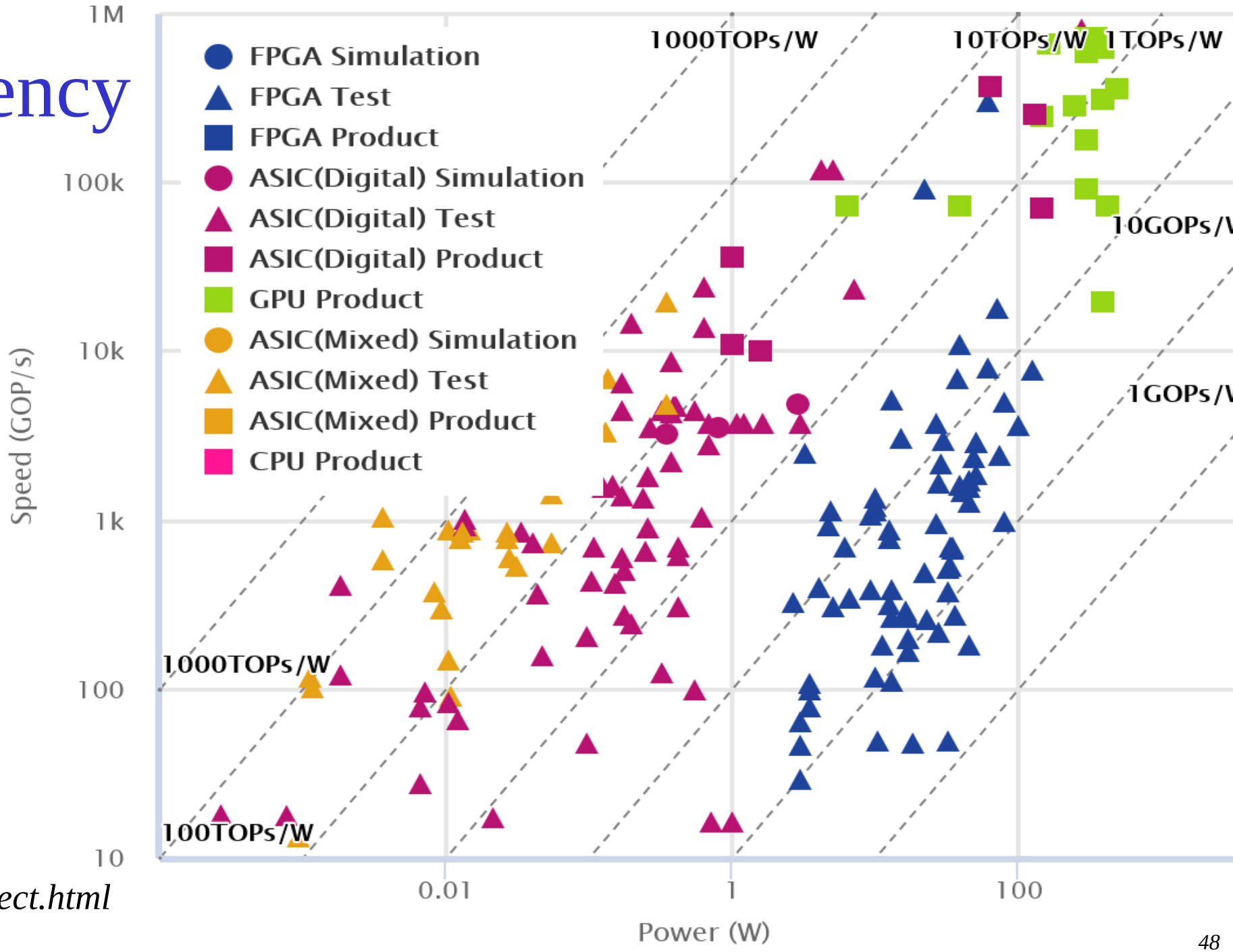


What to expect?

- AI Deep Learning Models
- Edge Mismatch: Cloud vs Edge
- Optimizations
- Learn from the Brain
- **SOTA in Edge AI computing**
  - **Accelerator Examples**
- Future
- Conclusions

Energy efficiency

- 2000-now

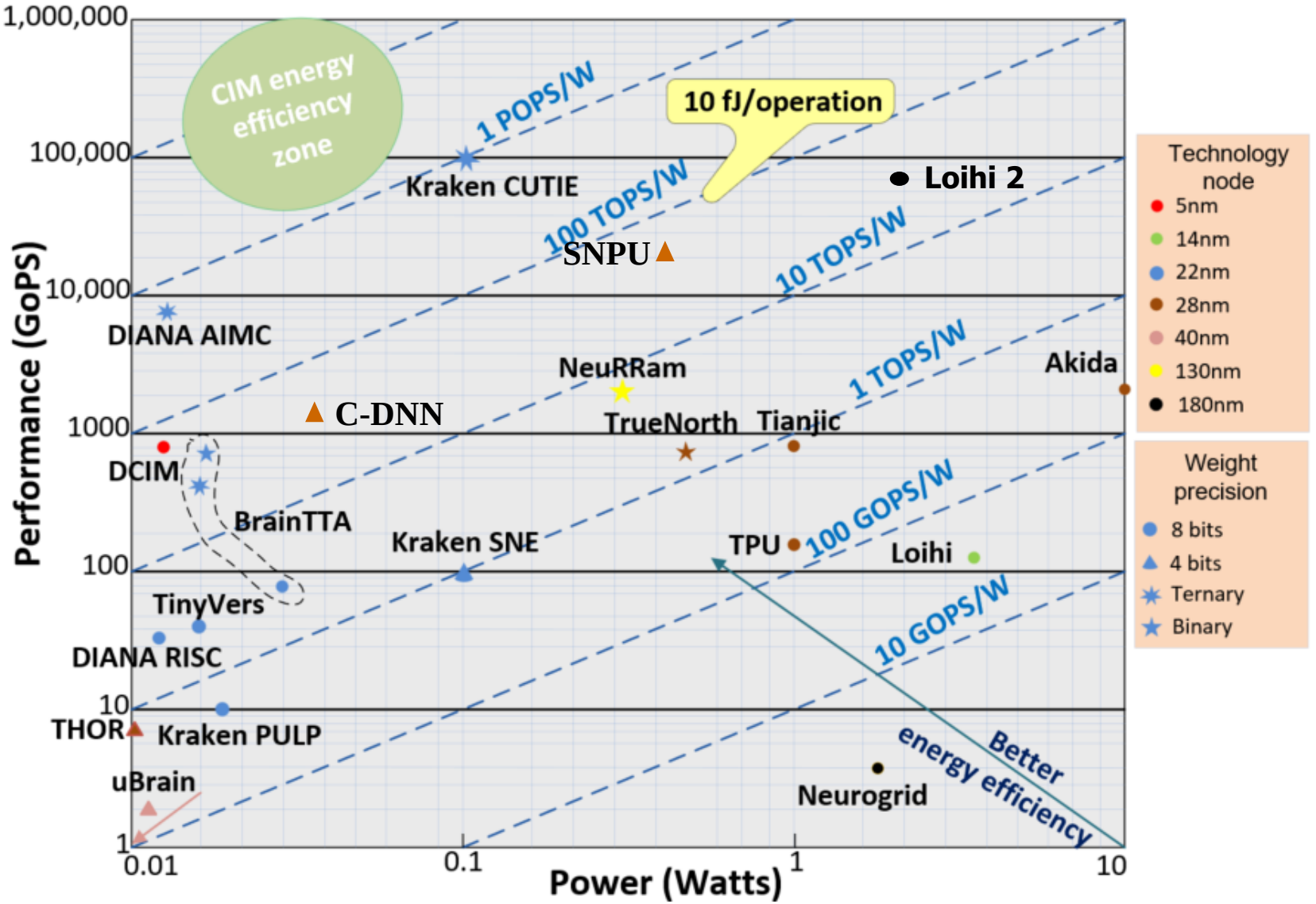


From:
nicsefc.ee.tsinghua.edu.cn/project.html

SOTA in Edge-AI Processor/Accelerator HW

- Models
  - Mostly Artificial Neural Network (ANN)
  - Spiking Neural Network (SNN)
- Weight Precision
  - 8/4/Ternary/1
- Technology nodes
  - from 5 – 180nm

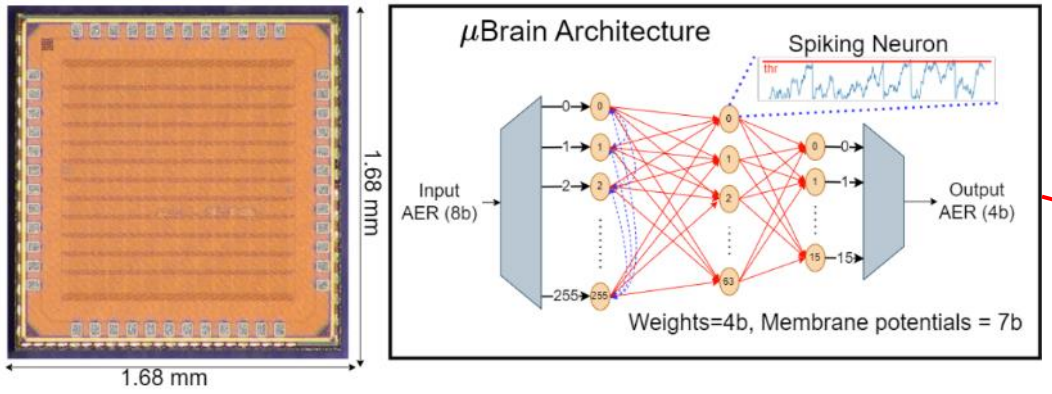
- Note, **real comparison should include accuracy !!**

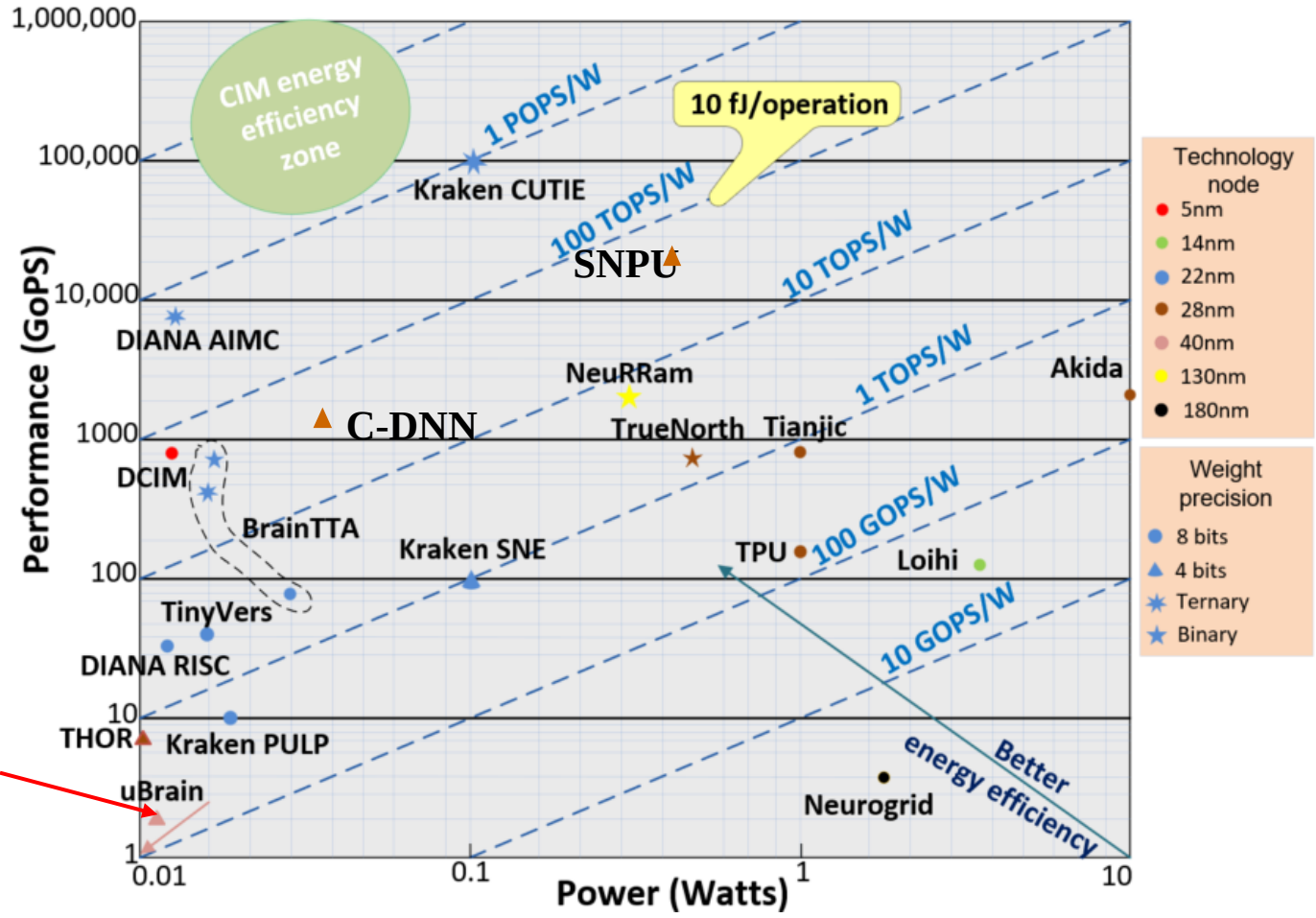


Adapted from Gomony e.a., DATE 2023, PetaOps/W edge-AI μ Processors: Myth or reality?

µBrain- Digital SNN (IMEC)

- 40 nm, 1.4mm^2
- Event based architecture
- Asynchronous design (no clock)
  - Without schedules, clocks, state machines
- Extreme low power, not en-eff.

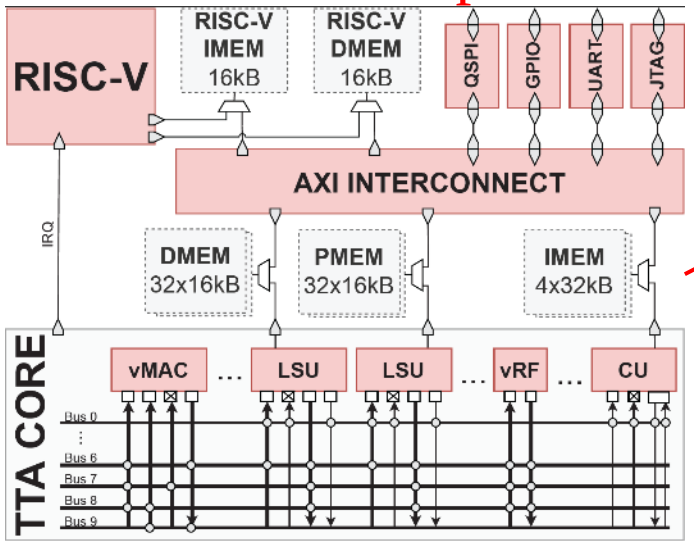


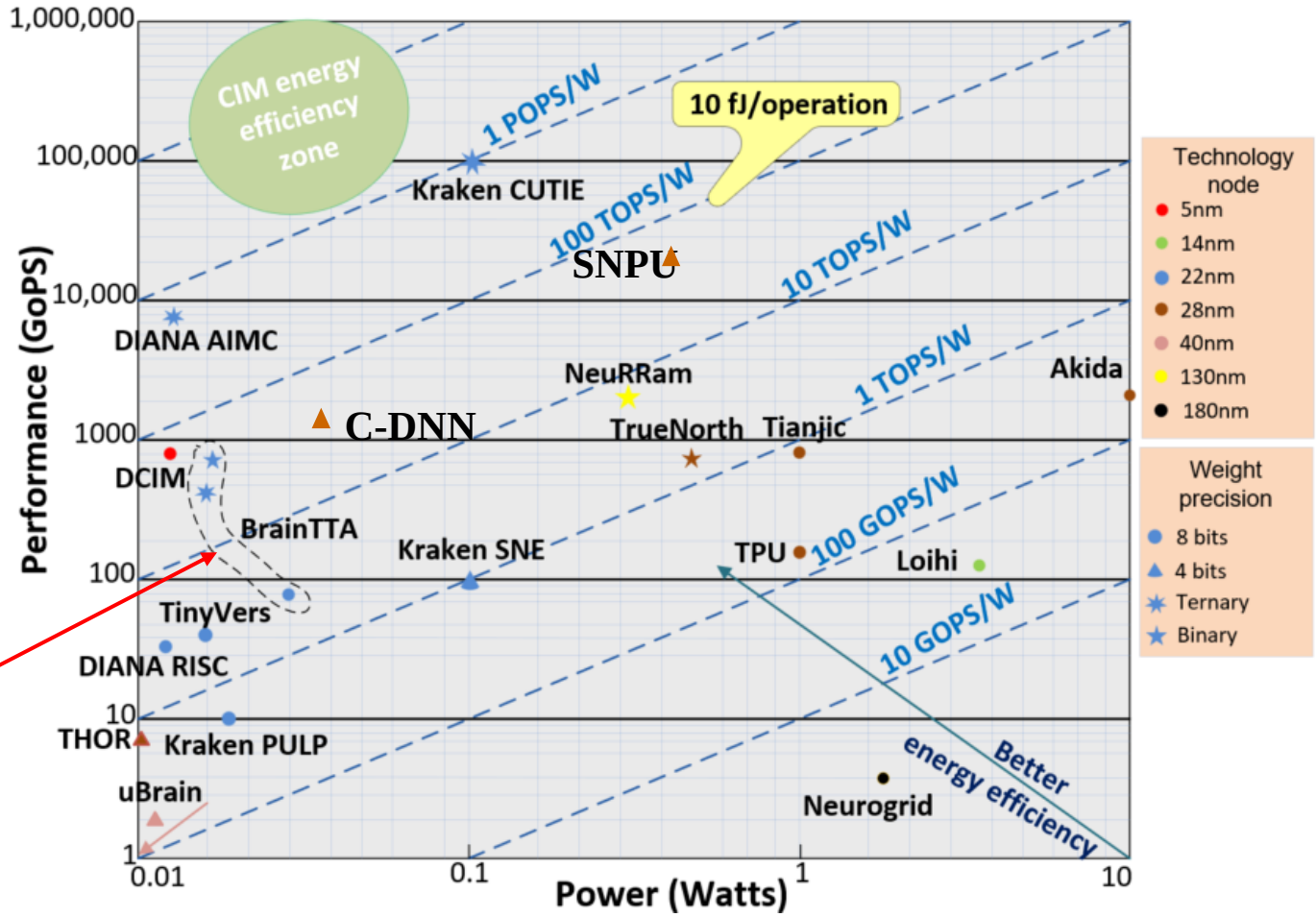


Jan Stuijt e.a., Frontiers, May 2021

BrainTTA - Flexible & Mixed-precision (TUE)

- TTA-based accelerator, 22nm
- Fully-programmable
  - C-compiler
- Flexible precision
  - INT8, ternary, binary
  - 405 / 67 / 35 fJ/op

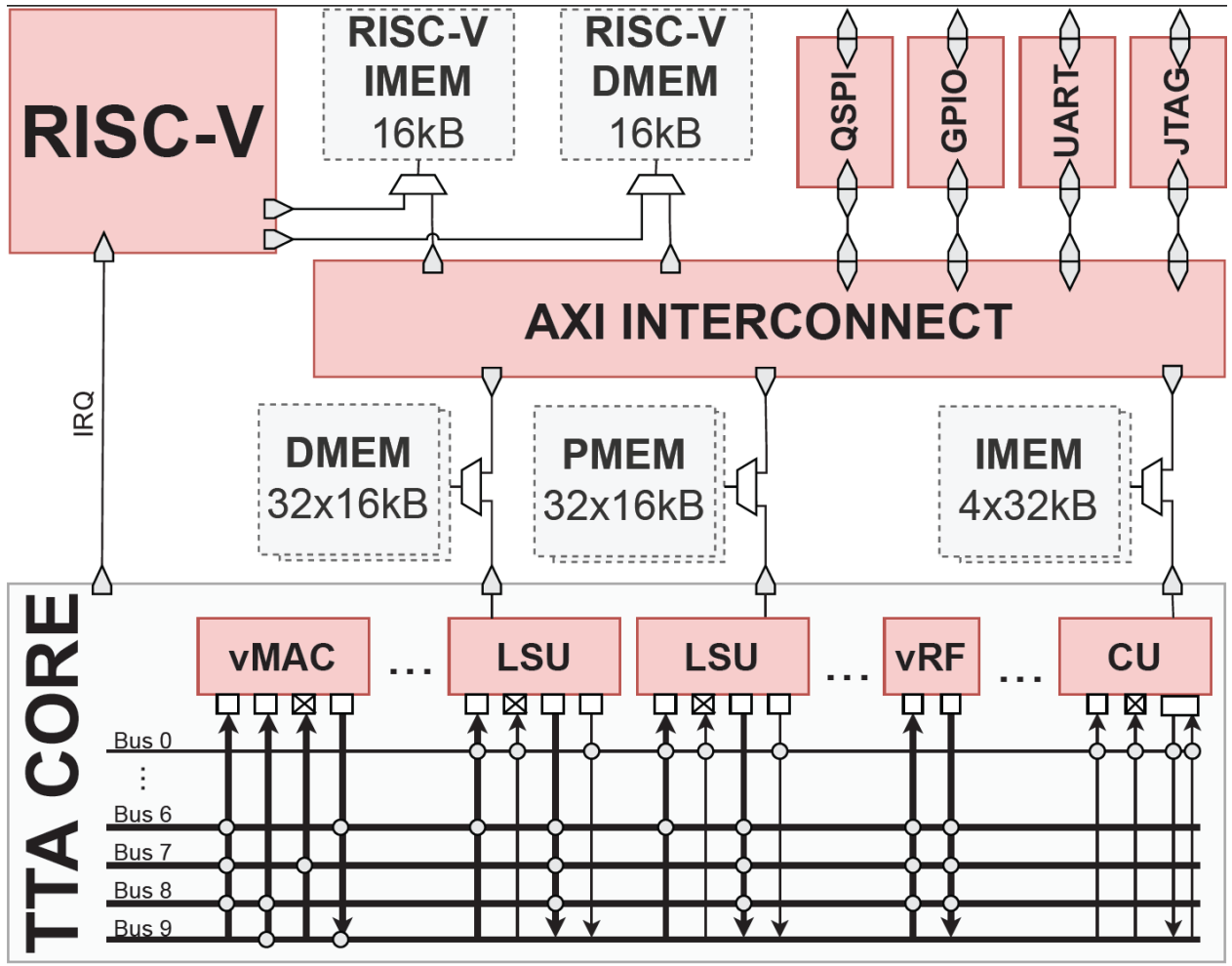




Molendijk e.a. ICCD 2023

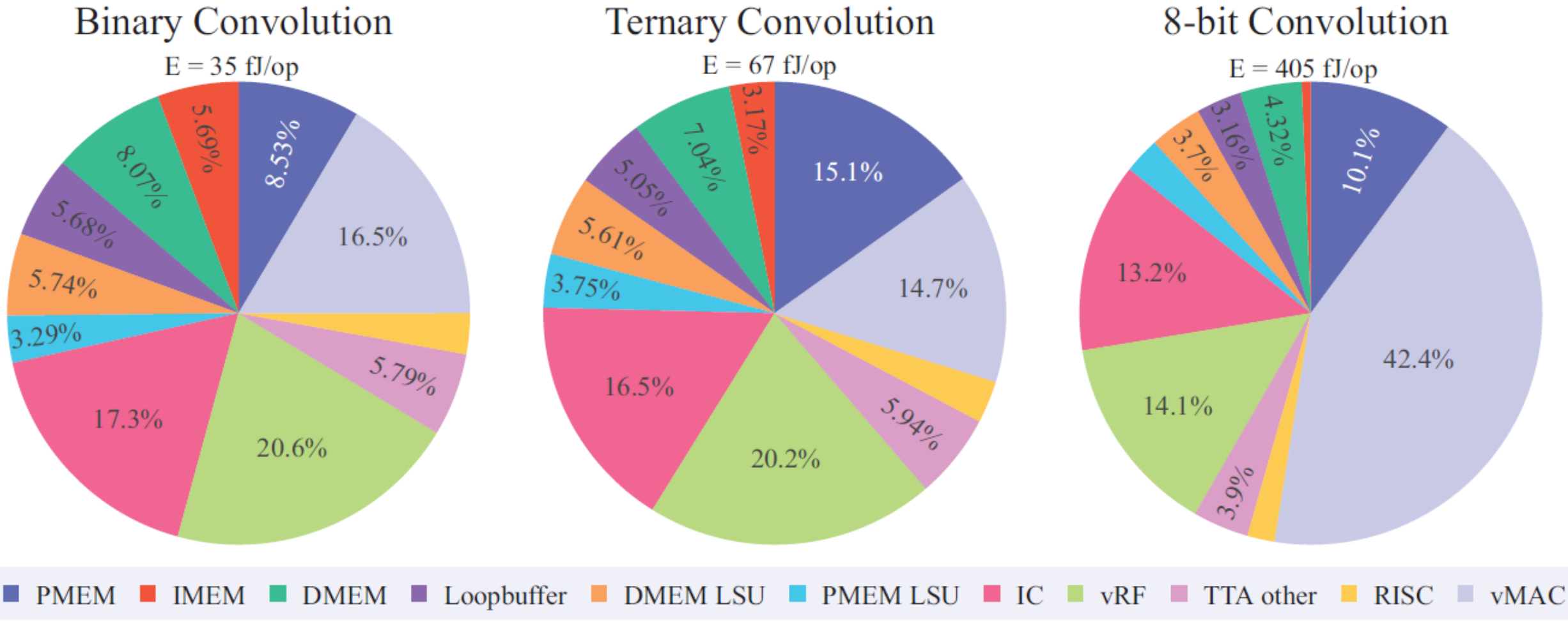
Example Edge-AI accelerator: BrainTTA

- TTA-based accelerator, 22nm
  - Fully-programmable (C-compiler)
  - Huge **vector** units, 1024 bit:
    - MAC: multiply accumulate
    - Other operations, like ReLU
    - Vector register files
    - Load/Store vector units
- Large on-chip memories for
  - Inputs, Output, Weights
- Flexible precision
  - INT8, ternary, binary:
  - 405 / 67 / 35 fJ/op



Molendijk e.a. ICCD 2023

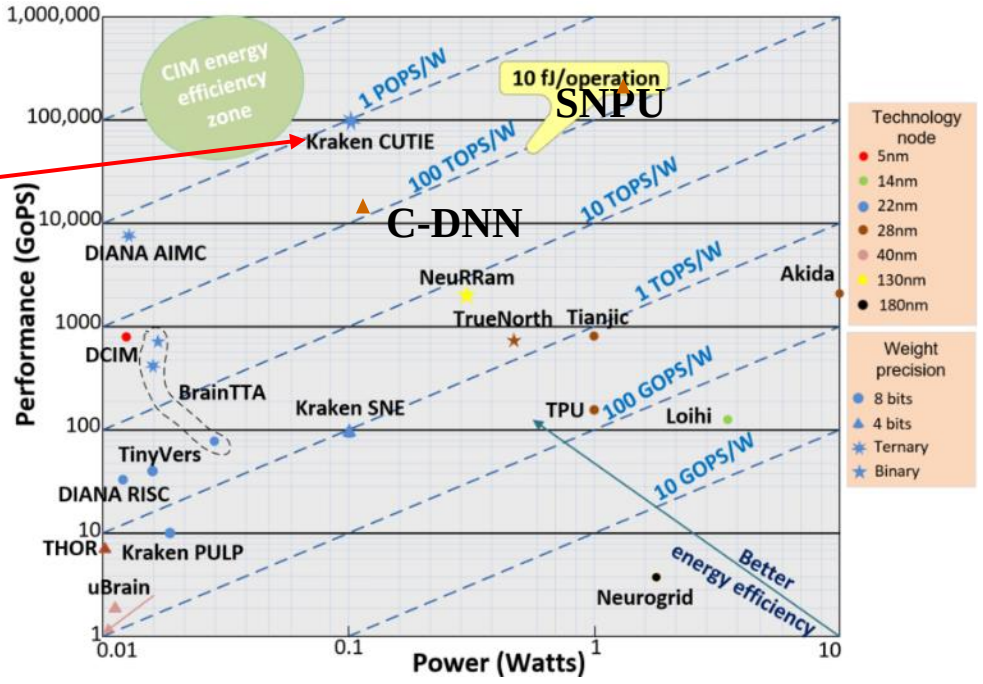
Where is the energy going: BrainTTA

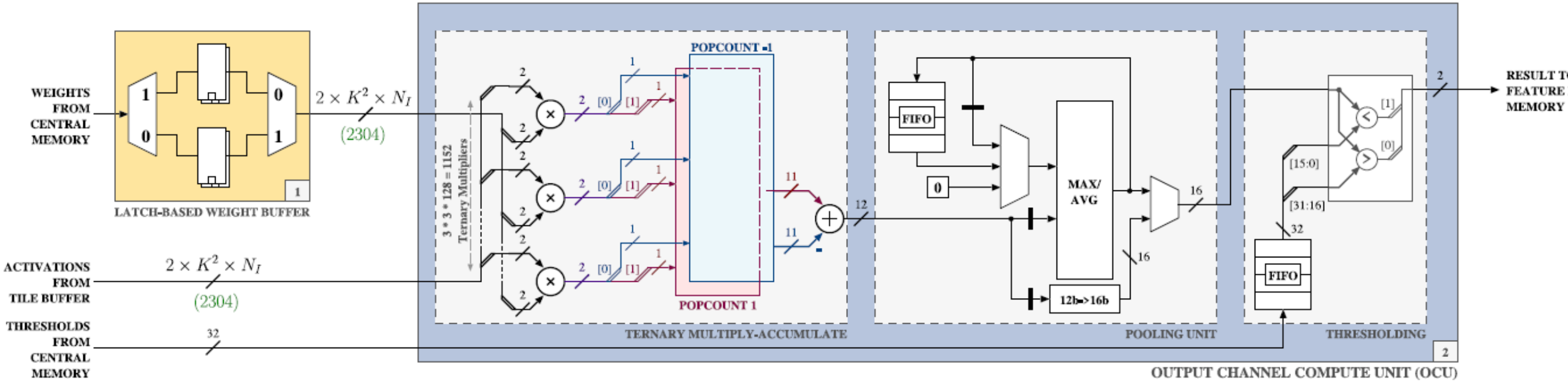


Energy: 14.7 – 42.4 % in vMACs

CUTIE –Ternary DNN (ETH)

- ANN: **Ternary** (-1,01) Inference Engine
  - exploit zero surpression
  - ternary compression
    - 1.6 bit / symbol
  - complete unrolled loops in HW
    - 3x3*128 = 1152 kernel size convolution
  - 22 nm, 2.5-4 fJ/op (%Sparsity dependent)

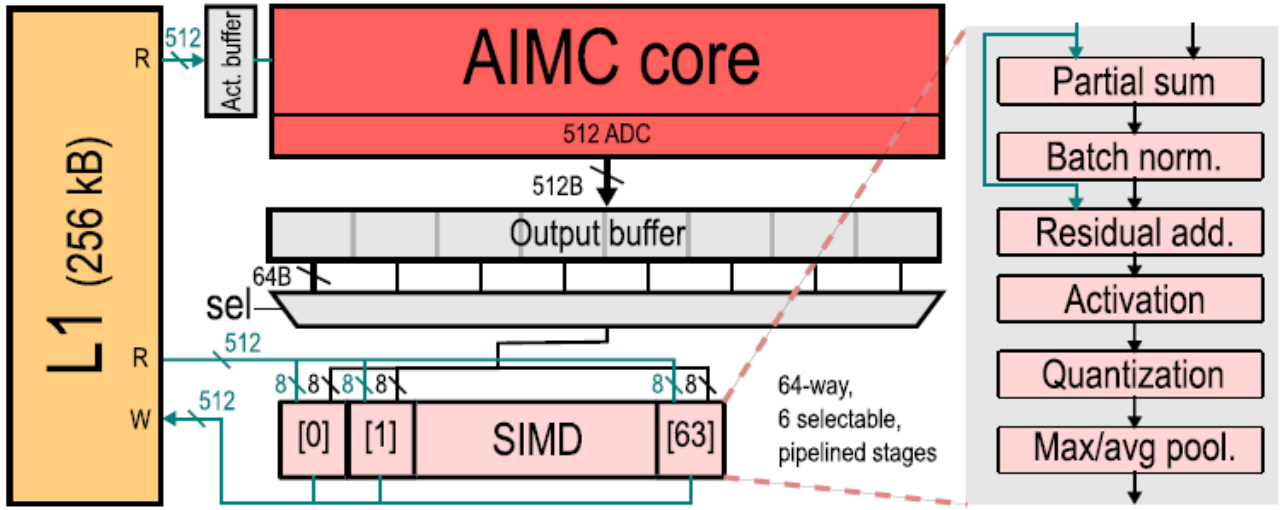


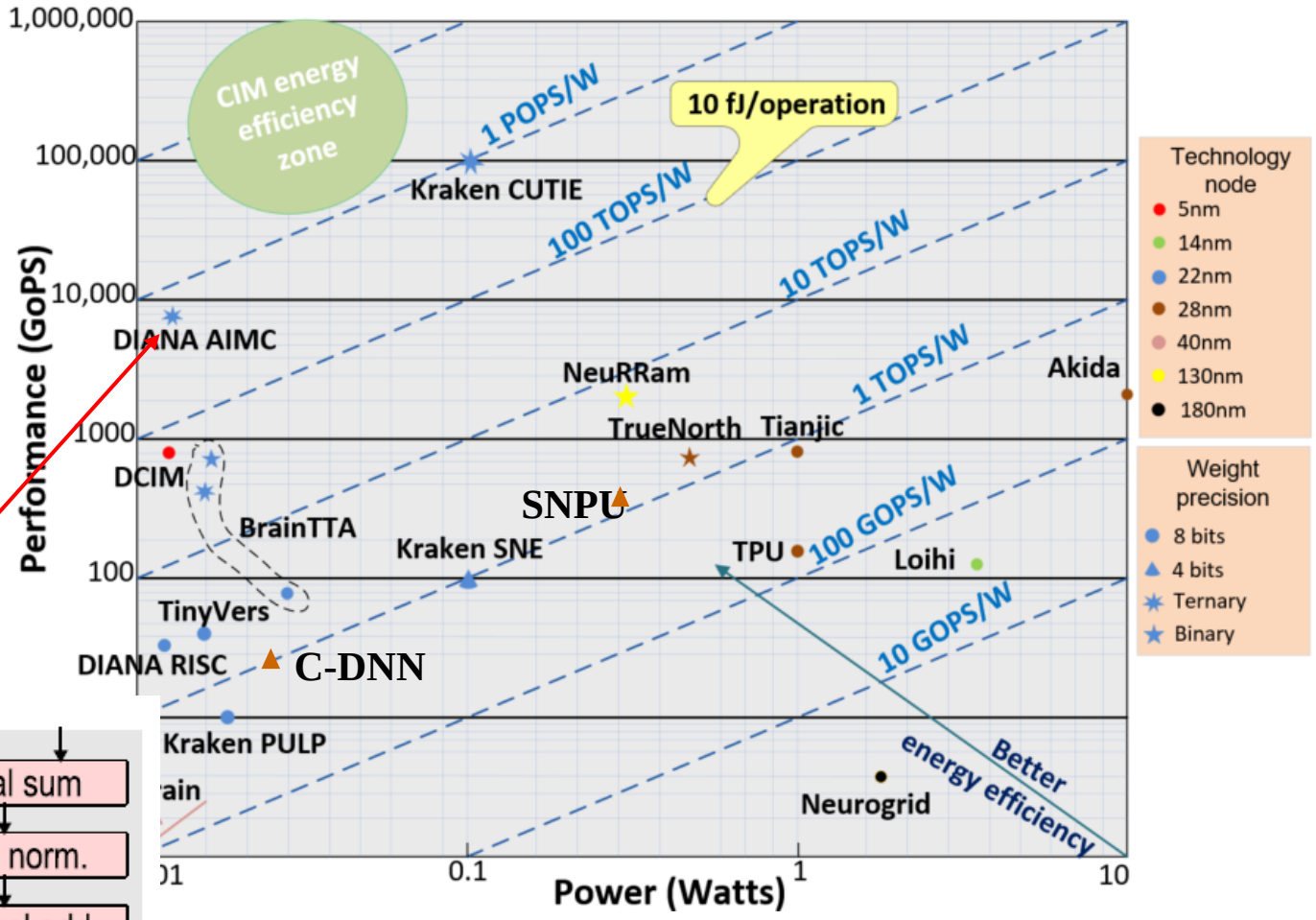


Scherer e.a. TCAD 2021, Cutie , Beyond PetaOp/s/W Ternary DNN

DIANA - Mixed-signal

- Precision-scalable **digital** core
  - INT 8/4/2
- **Analog** In-Memory Core (AIMC)
  - Ternary weights, 7-b activation
  - Programmable SIMD
  - 22nm 1.7 fJ/Op (I/W/O= 7/1.5/6-bit)

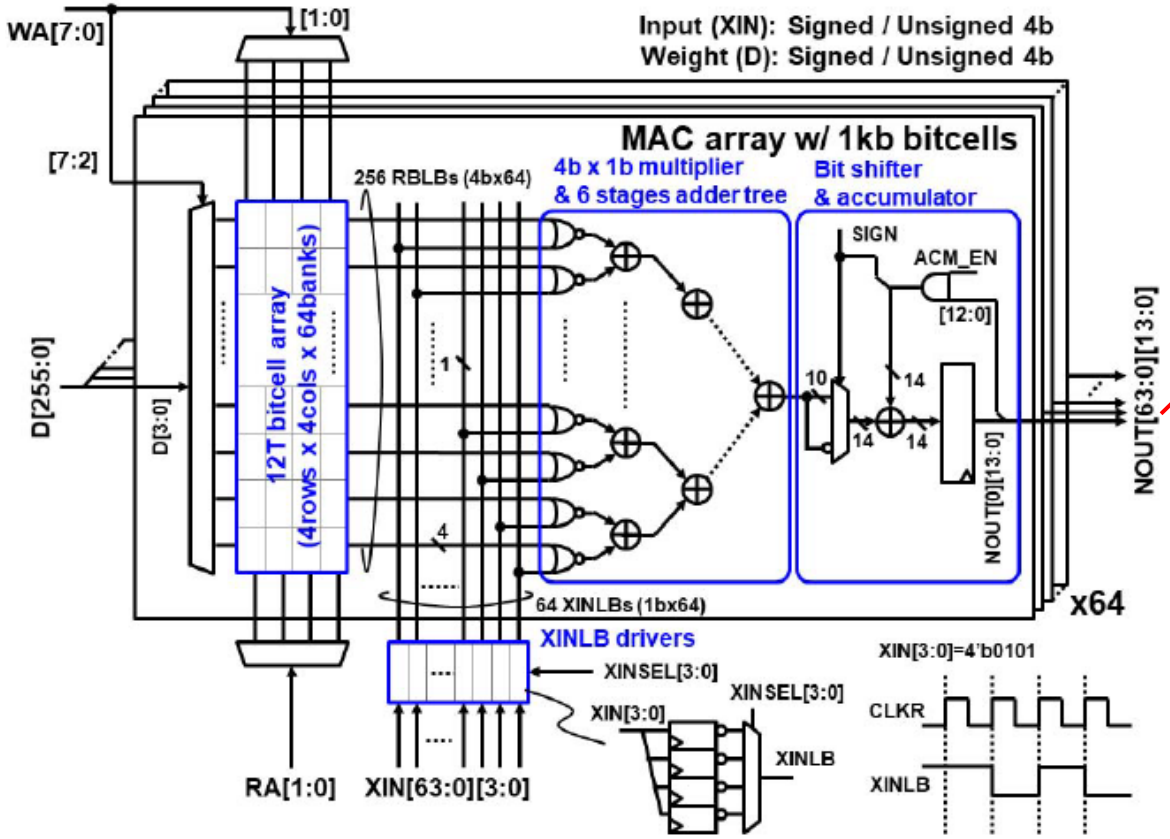


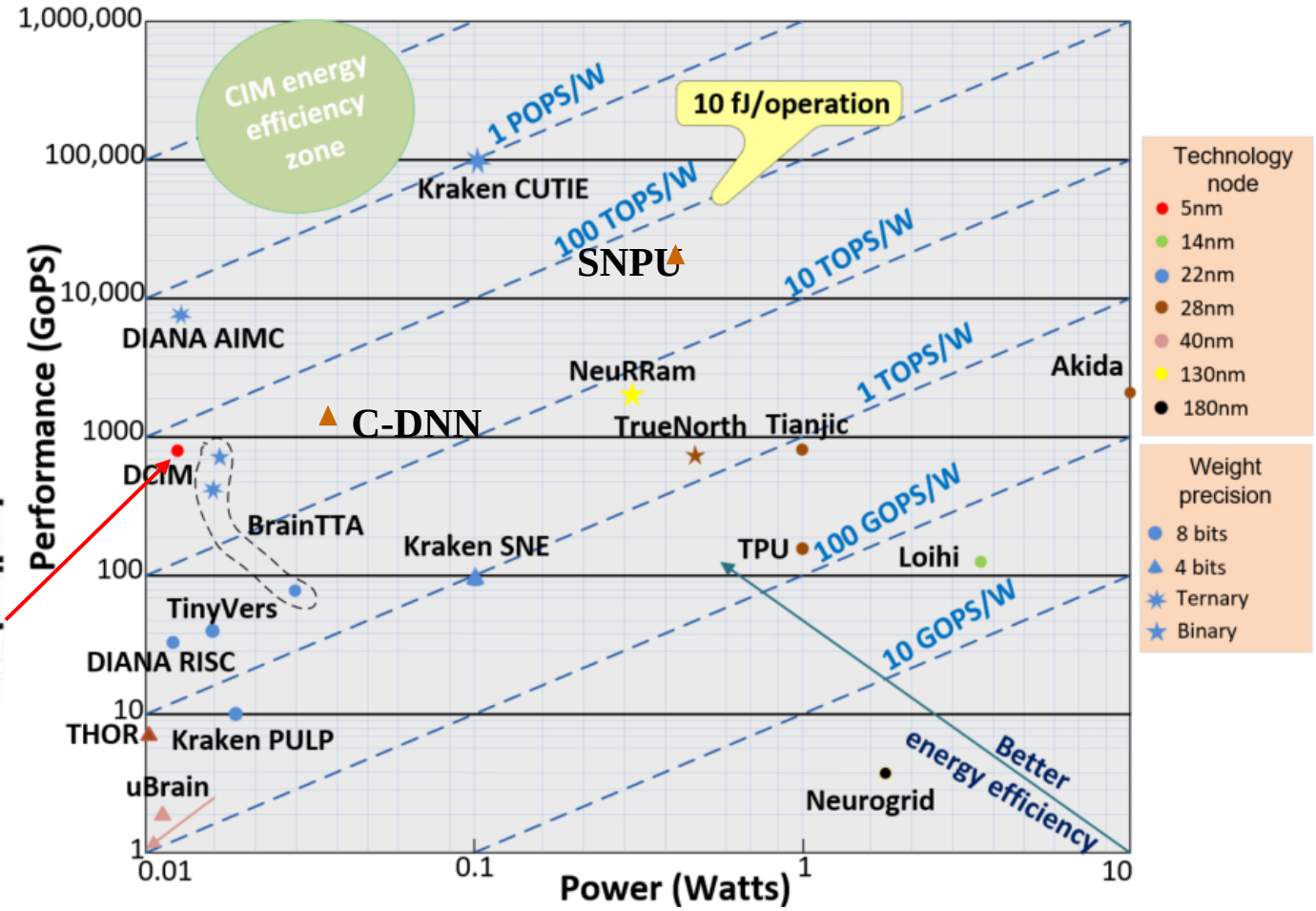


Houshmand e.a., JSSC 2022

Digital CIM - SRAM-based

- 12T bitcell based architecture
- Flexible precision
  - INT 8/4, 5nm, 15.8 / 3.9 fJ/Op
- MAC using 1-b at a time

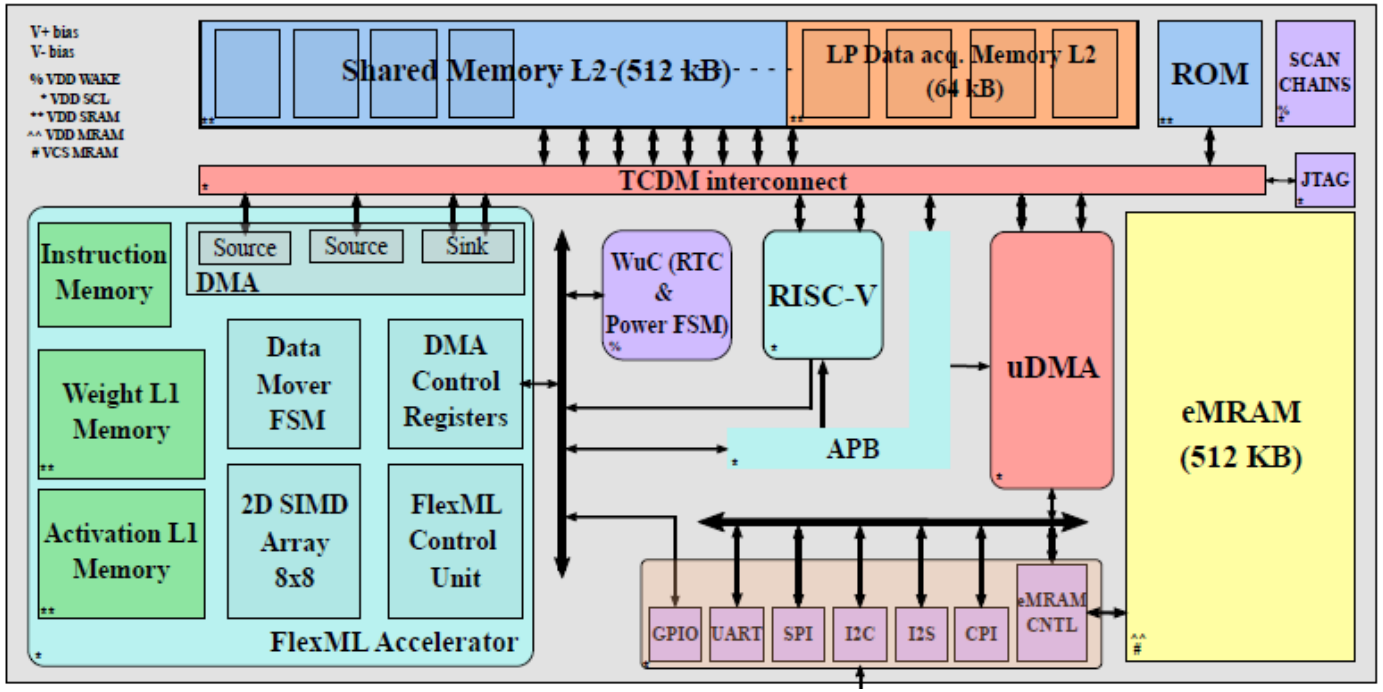


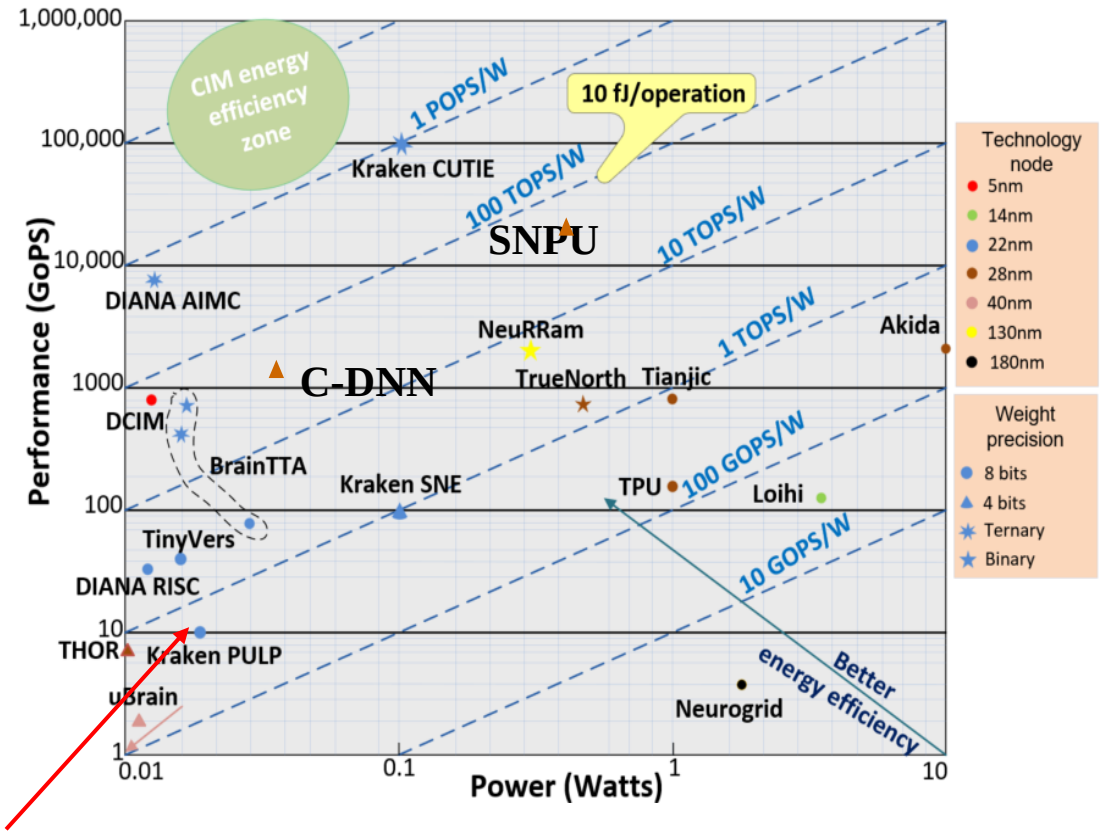


Fujiwara e.a., ISSCC 2022

TinyVers - Embedding MRAM

- Supports various DNN layers, to traditional ML models like SVM
- RISC-V + ML Acc.
- Flexible-precision scalable digital accelerator, max 17 TOPS/W ~ 59 fJ/Op (Int2)
  - INT2/4/8 precision

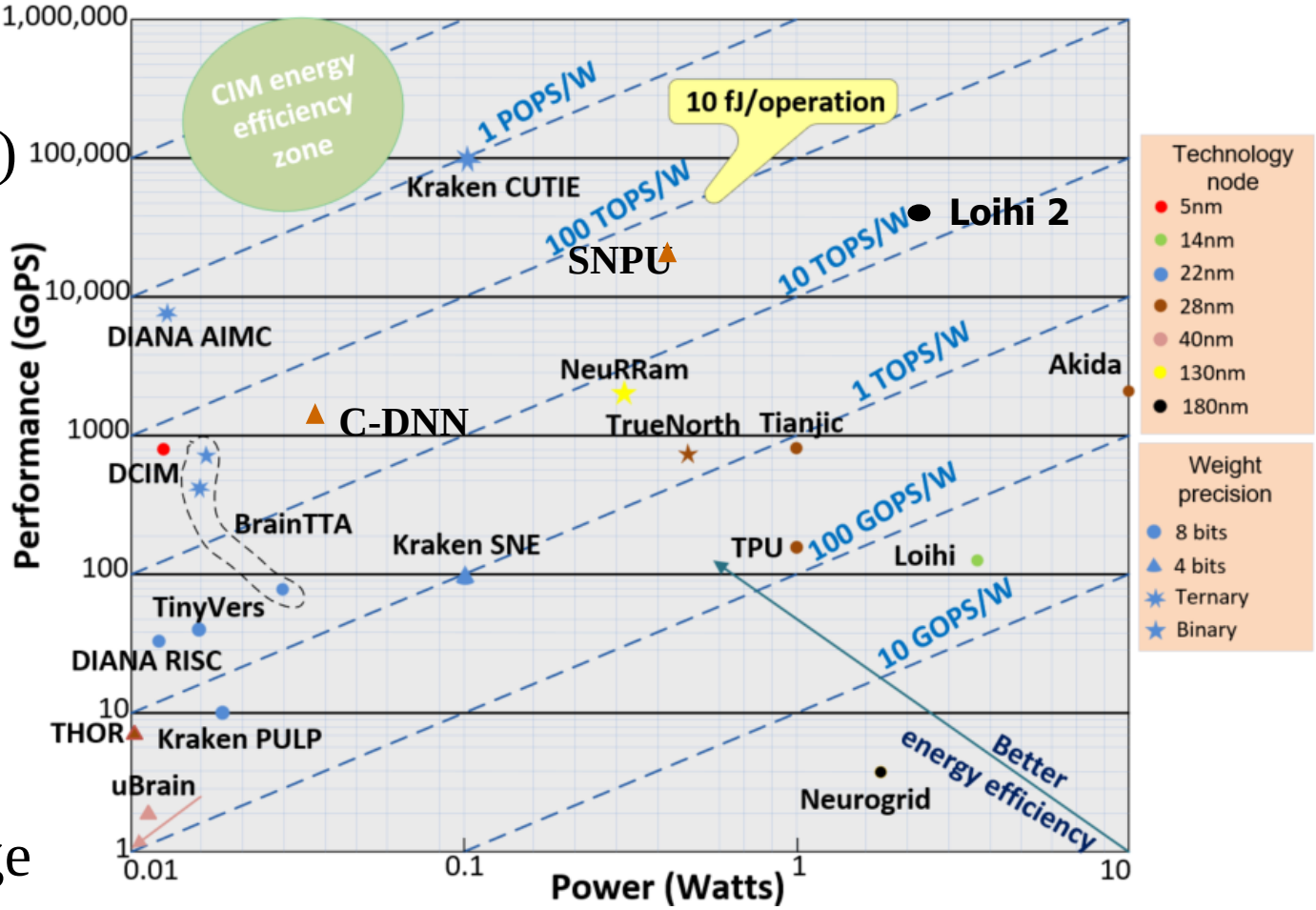




Jain e.a., VLSI Symp 2022

State-of-the-Art Summary

- **ANN** chips approaching 1 fJ/Op
- However:
  - Requires complete unrolling (CUTIE) and/or AIMC (DIANA)
  - System overhead often neglected
- Flexibility has its price:
  - E.g. BrainTTA suffers at least one-order in energy efficiency
  - DRAM overhead not included
- **SNNs** have high potential,
  - However: at best in the 35 fJ/Op range



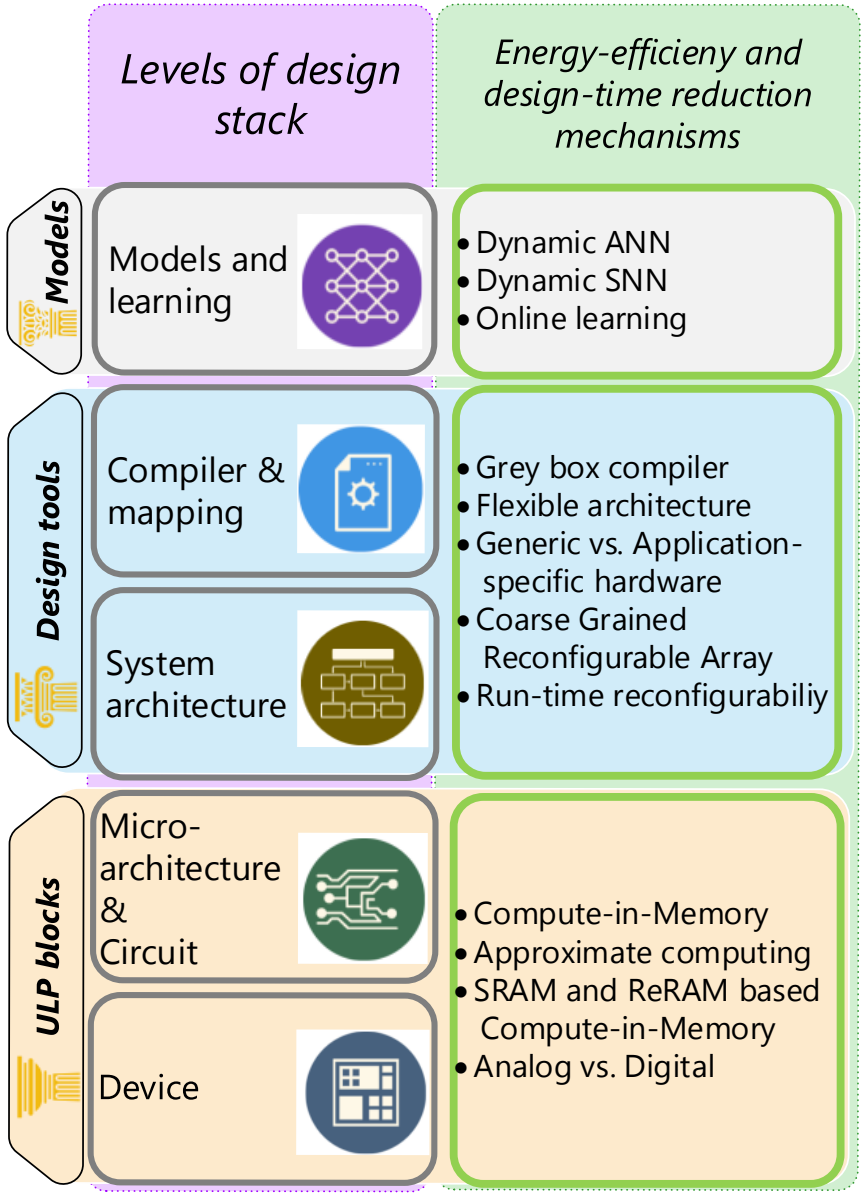
What to expect?

- AI Deep Learning Models
- Edge Mismatch: Cloud vs Edge
- Optimizations
- Learn from the Brain
- SOTA in Edge AI computing
- **Future**
  - **CONVOLVE Methodology**
  - **Accelerator research**
- Conclusions

CONVOLVE whole stack methodology

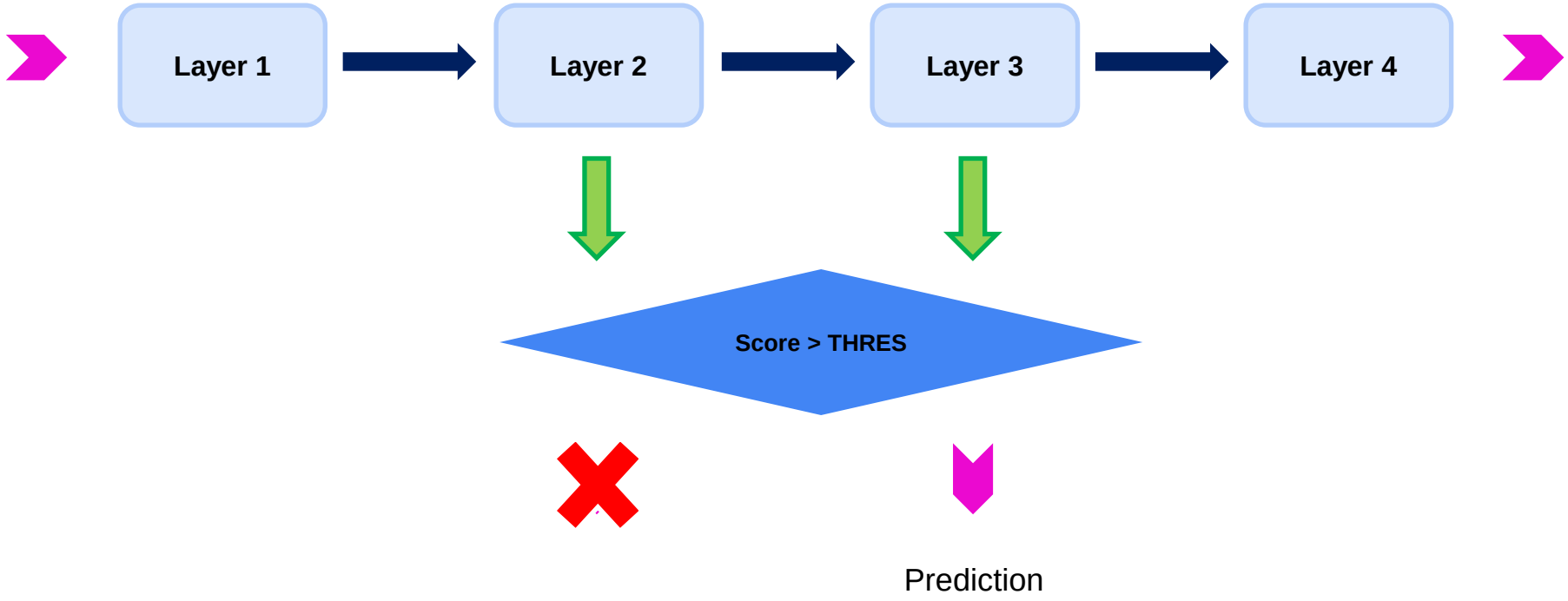
Improvements from all design levels, e.g.:

- Models:
  - Exploiting dynamism
  - Online learning: adapt and deal with errors
- Tools:
  - DSE searching huge mapping space
  - Compiler
  - Dynamic reconfiguration
- ULP blocks:
  - Various accelerators
  - CIM, ReRAM based



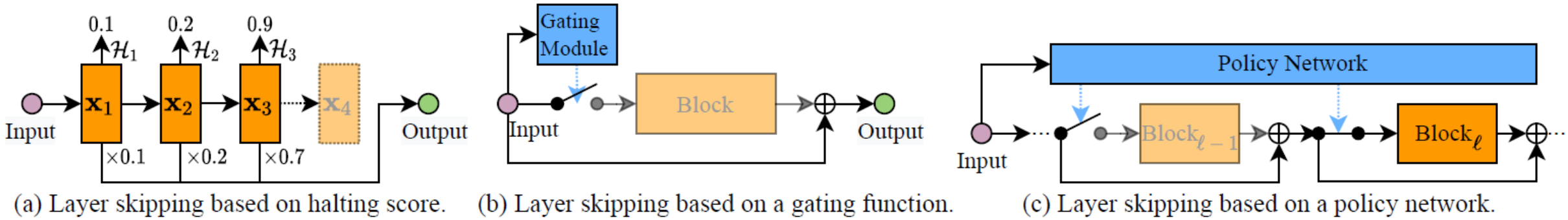
Dynamic NNs

- Exploit the 80-20 rule of life
- Multi-exiting allows confident early termination of computation via decision blocks



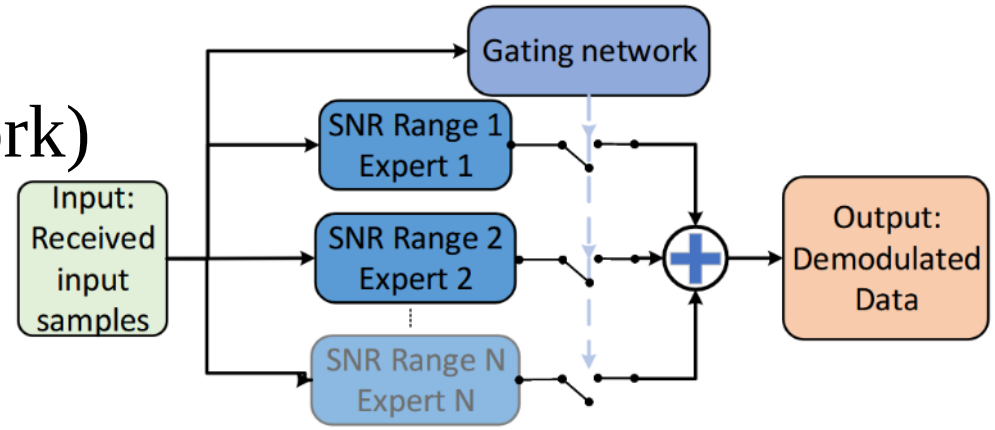
Dynamic NNs; other approaches

wireless comm./Van Bolderik TUE



(a) Layer skipping based on halting score. (b) Layer skipping based on a gating function. (c) Layer skipping based on a policy network.

- Layer skipping
  - skipping upto 60%, overhead 17%
- Mixture of experts
  - 50% dynamic reduction (vsone big network)
  - 37% power saving (incl. overhead)

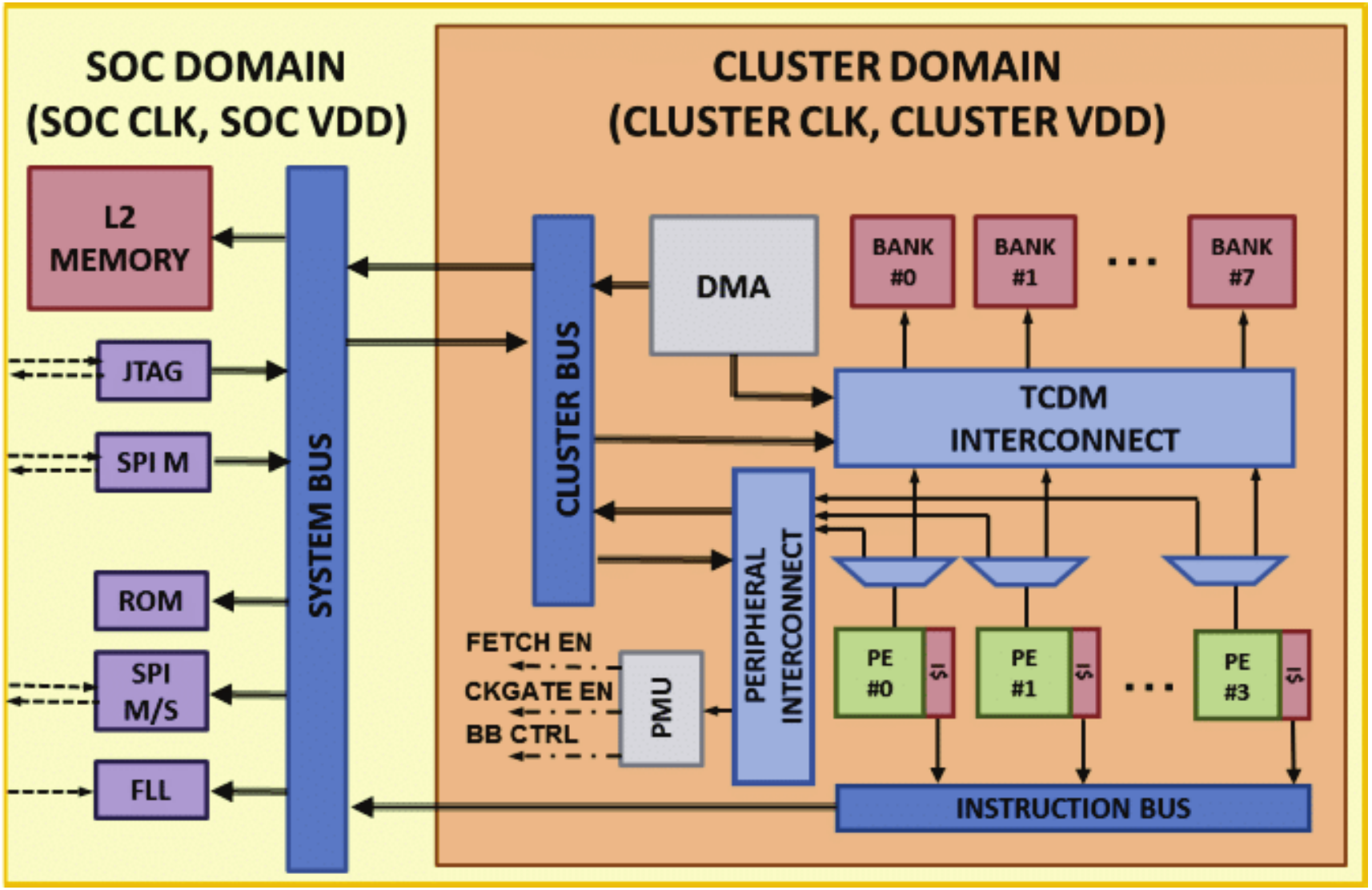


Bram van Bolderik, TUE, e.g. VLSI-SoC 2024
Yizeng Han e.g., Dynamic NN: A Survey, https://arxiv.org/abs/2102.04906

Integration: SoC architecture

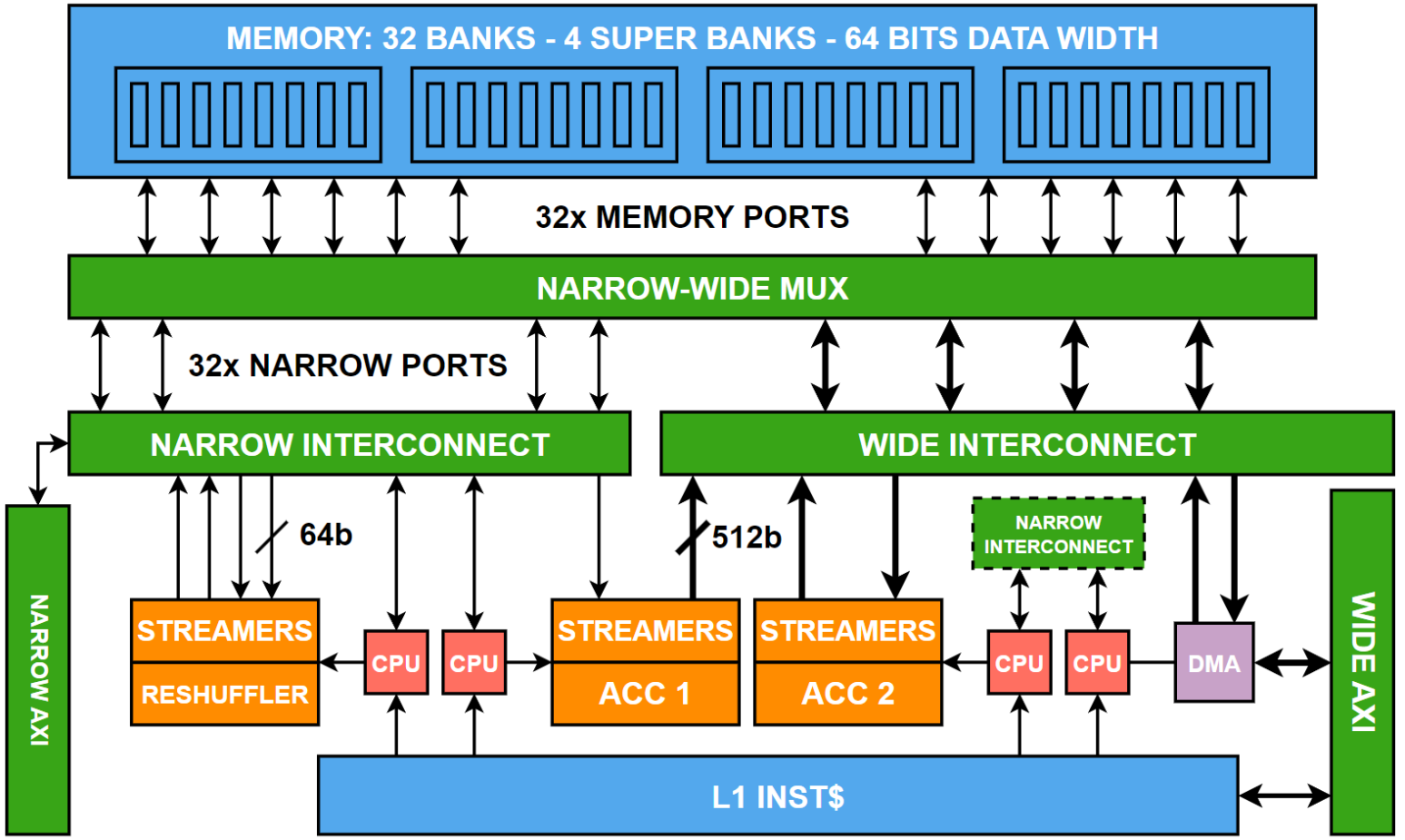
- PULP SoC (ETH)
- GF22 nm
- Many tapeouts

- Banked L1 memory
- Flexible accelator integration support



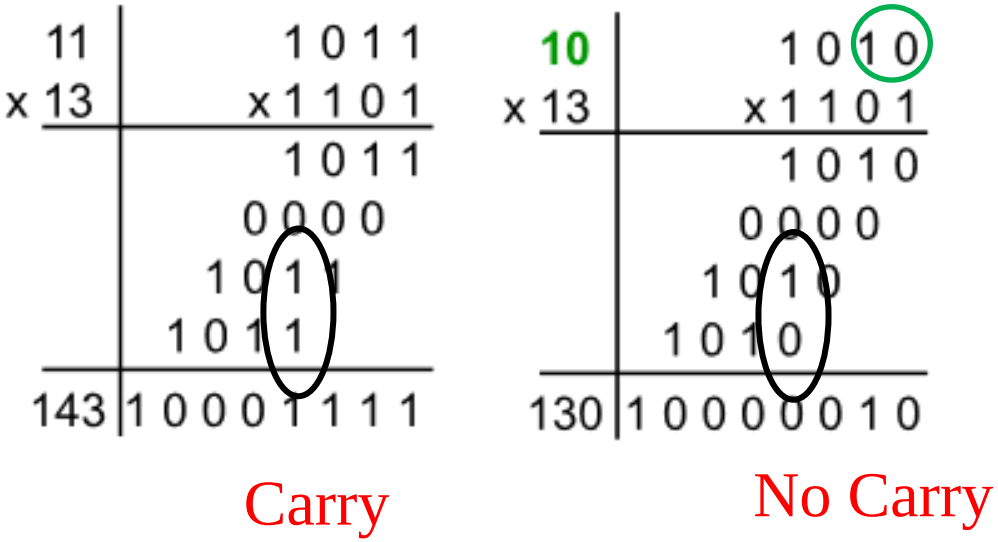
SNAX: Heterogeneous Cluster *(KULeuven)*

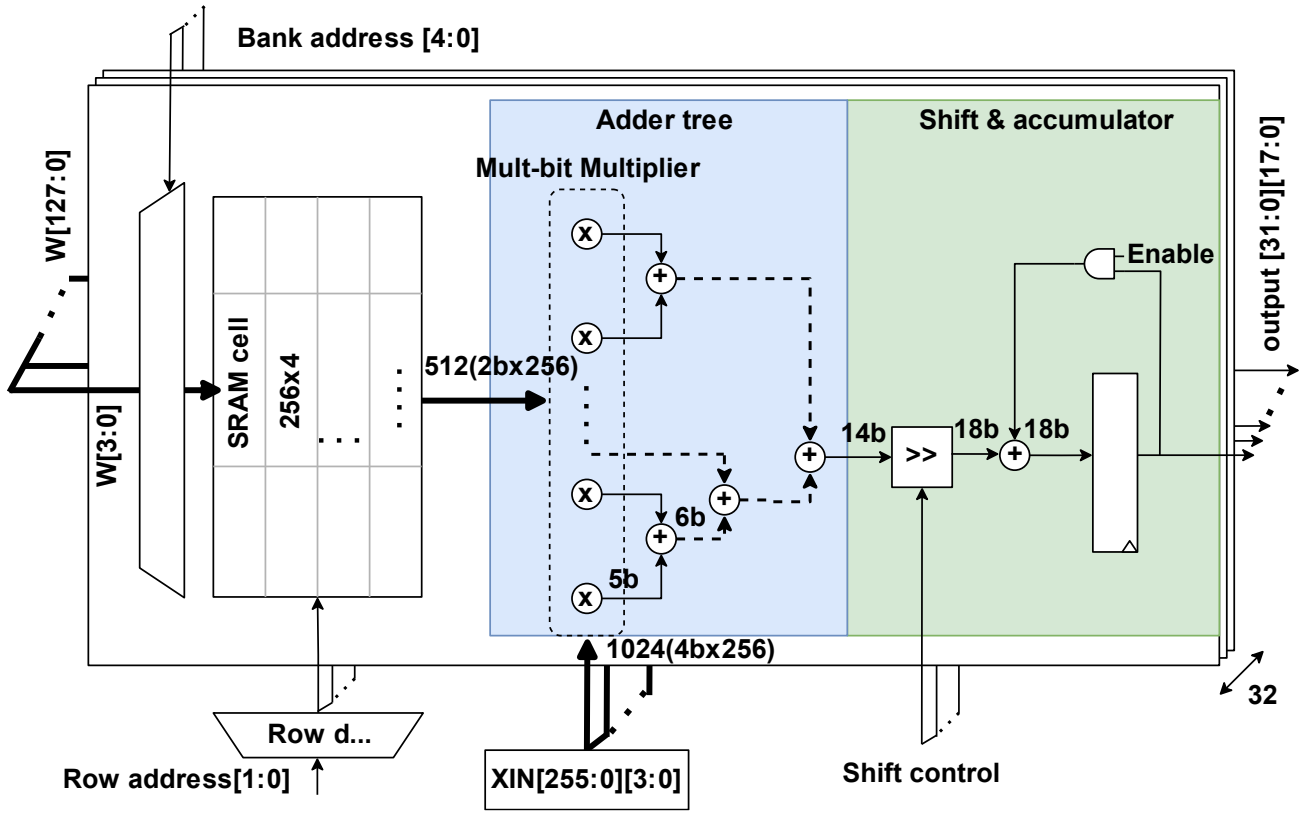
- Seamless accelerator insertion
- Design-time parameters
  - Memory sizes and interconnects
  - Accelerator connections towards memory (narrow or wide)
  - Number of accelerators
  - Number of management cores
- Modules that support run-time customization
  - Data streamers for managing data access patterns
  - Data reshufflers for re-arranging data layouts in memory



Digital CIM (TUE)

- Custom foundry-based SRAM
- Approximate Mul-Adder Tree
- Support 4b-8b weights/activation
- Fibbinary encoding (no ‘11’)
- 157 TOPS/W, 6.37 fJ/Op (2b*4b)
- GF22nm

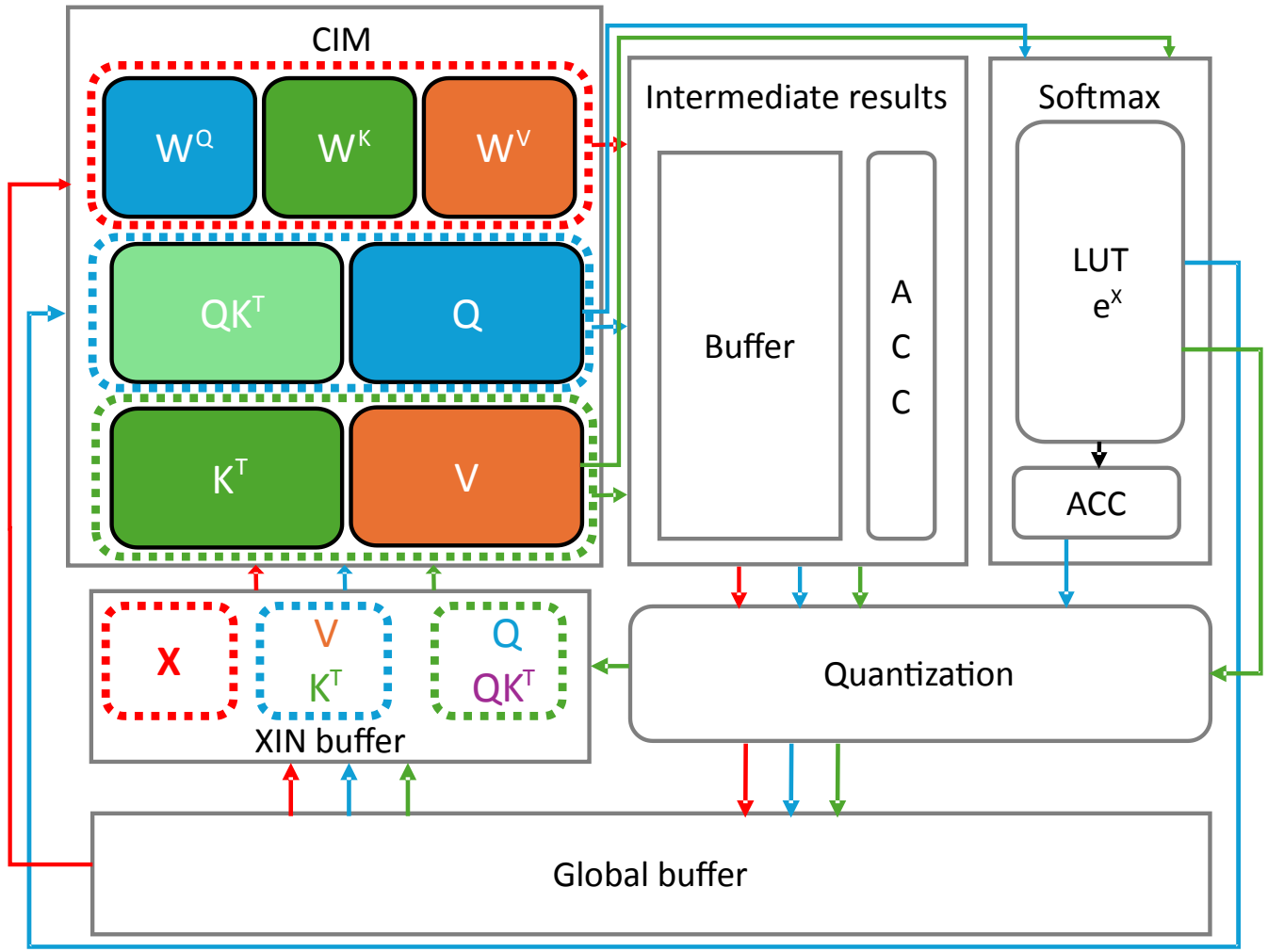




Example of a 32Kb SRAM DCIM architecture

CIM based Transformer accelerator: CIMple

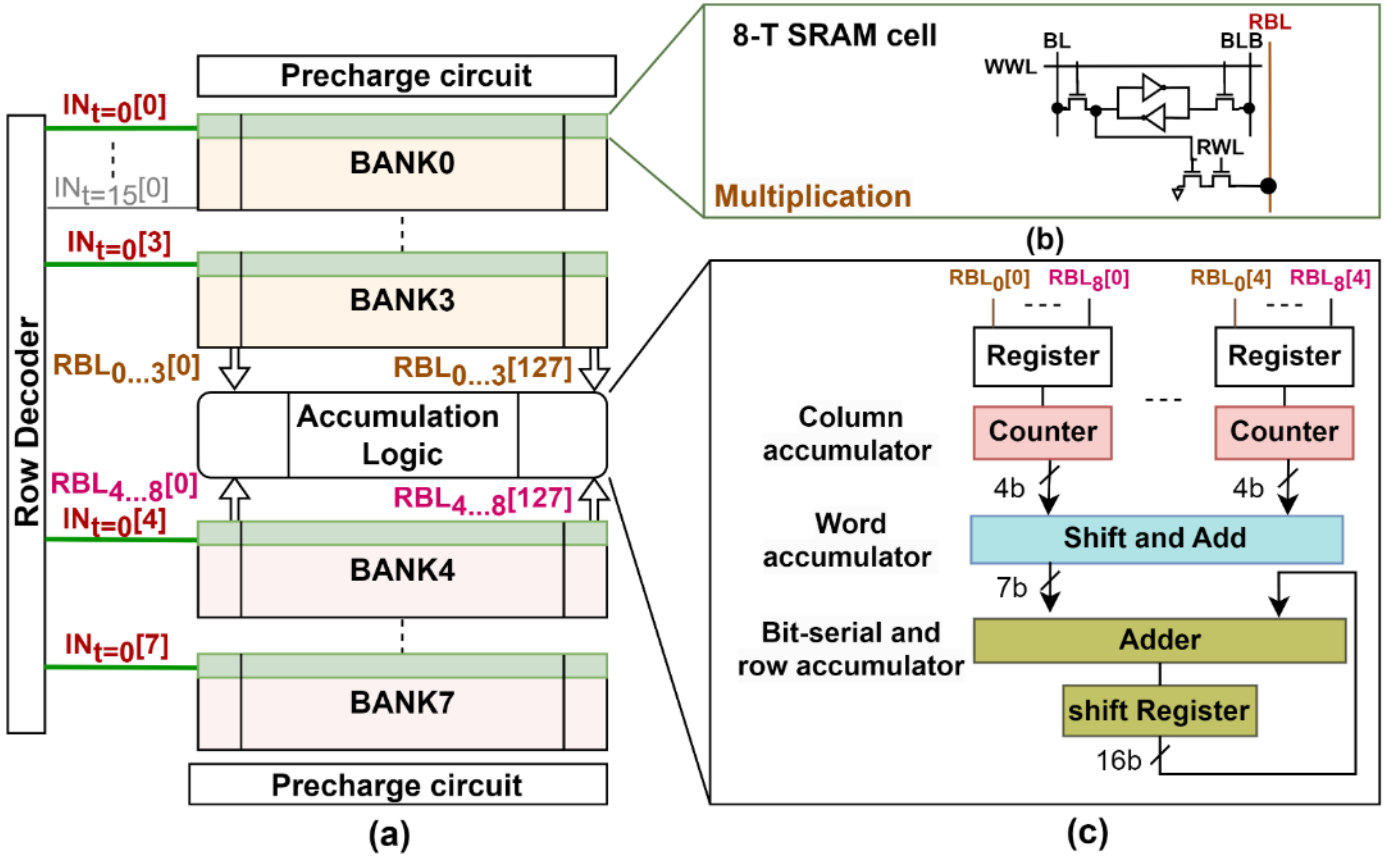
- 8T bitcell based architecture
- MAC using 1-b at a time
- Standard-cell approach
- Flexible architecture
  - Encoder only
  - Decoder only
  - Encoder-Decoder
  - Softmax support in HW
- INT8 precision
- 26.1 TOPS/W at 0.85 V or 38 fJ/op
- 2.31 TOPS/mm^2 in 28nm



Adder-free SRAM-based-CIM architecture(TUD)

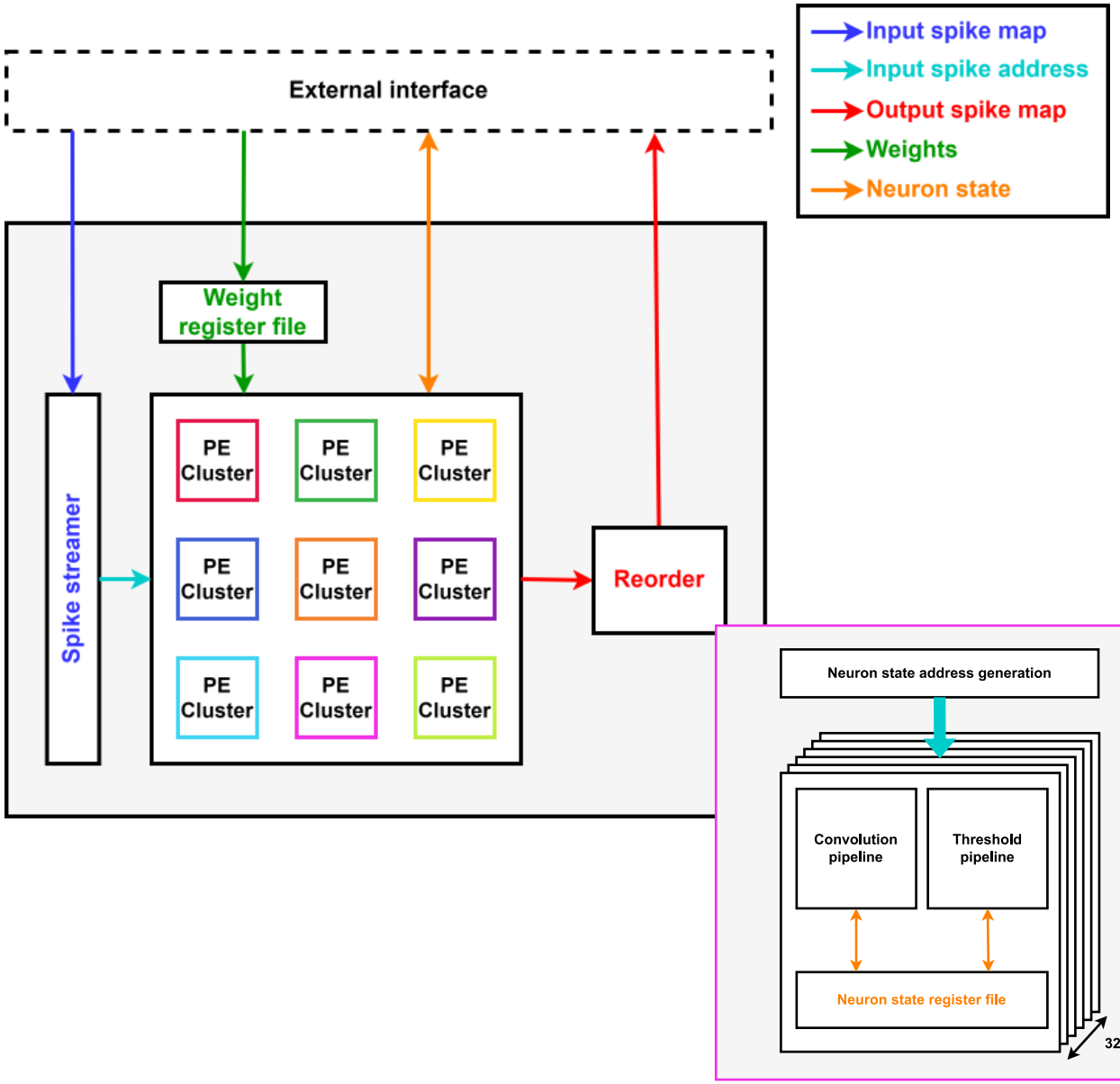
Energy and Area Efficient SRAM-Based Digital CIM Accelerator for Edge AI

- Adder-Tree-free accumulation for energy efficiency
- Embedded bit wise multiplication for energy/area efficiency
- Memory banks partitioning for enhanced throughput



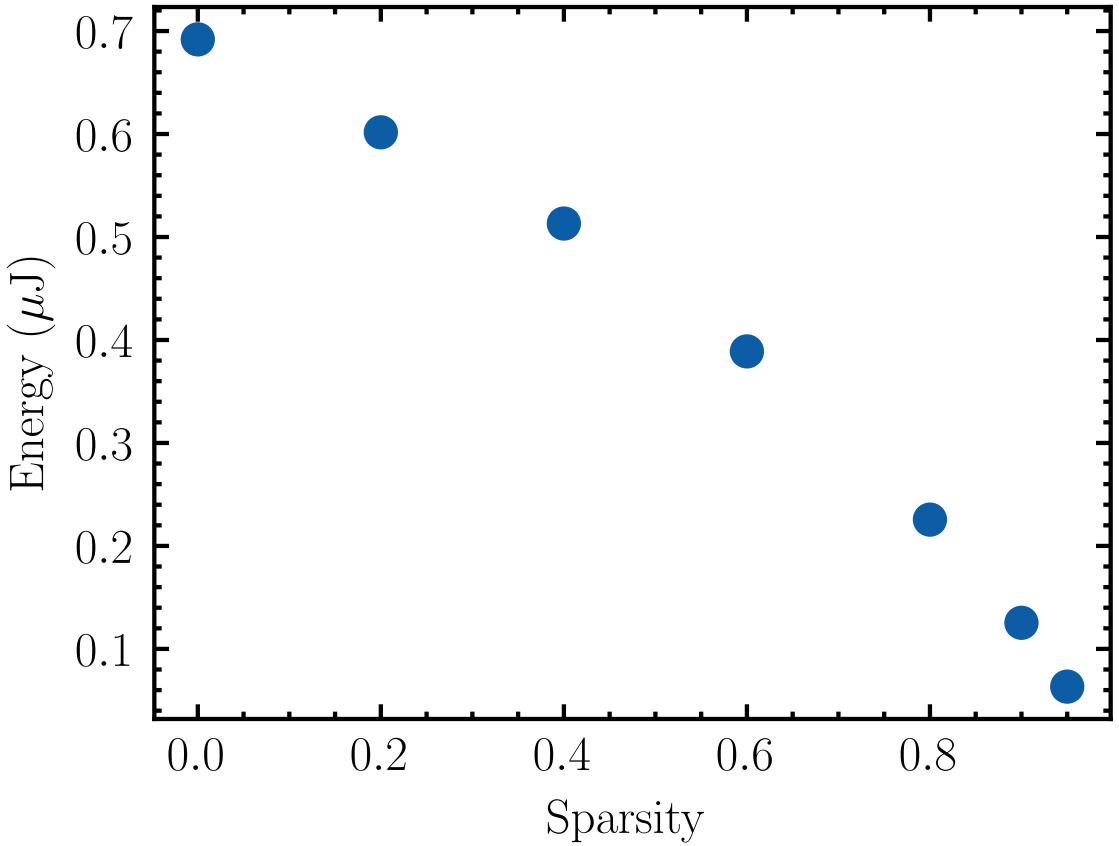
Mega: SNN accelerator

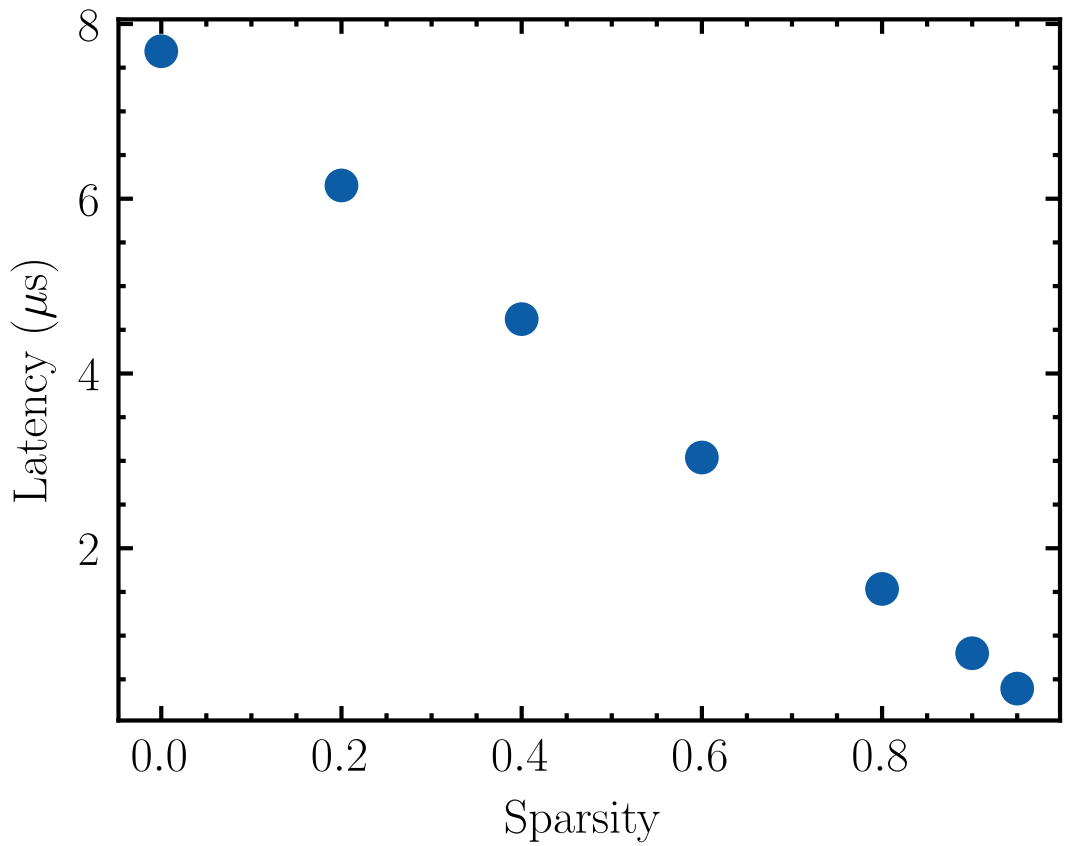
- 3x3 spiking convolution
  - Event-based: operations happen when there is a spike
  - Each PE cluster updates 32 neurons in parallel
- Memory hierarchy
  - Small, fast memories near the PEs
  - Larger memory for flexibility
- 0.501 pJ/SOP
  - Post-synthesis in 22 nm
  - 625 MHz @ 0.8V



Mega: What happens if spiking is sparse?

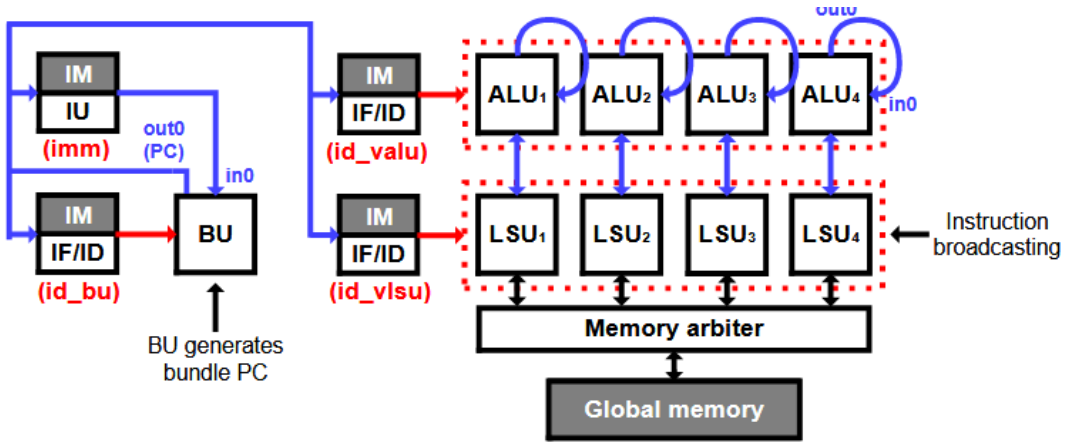
Energy and latency for a 96x48 spike map

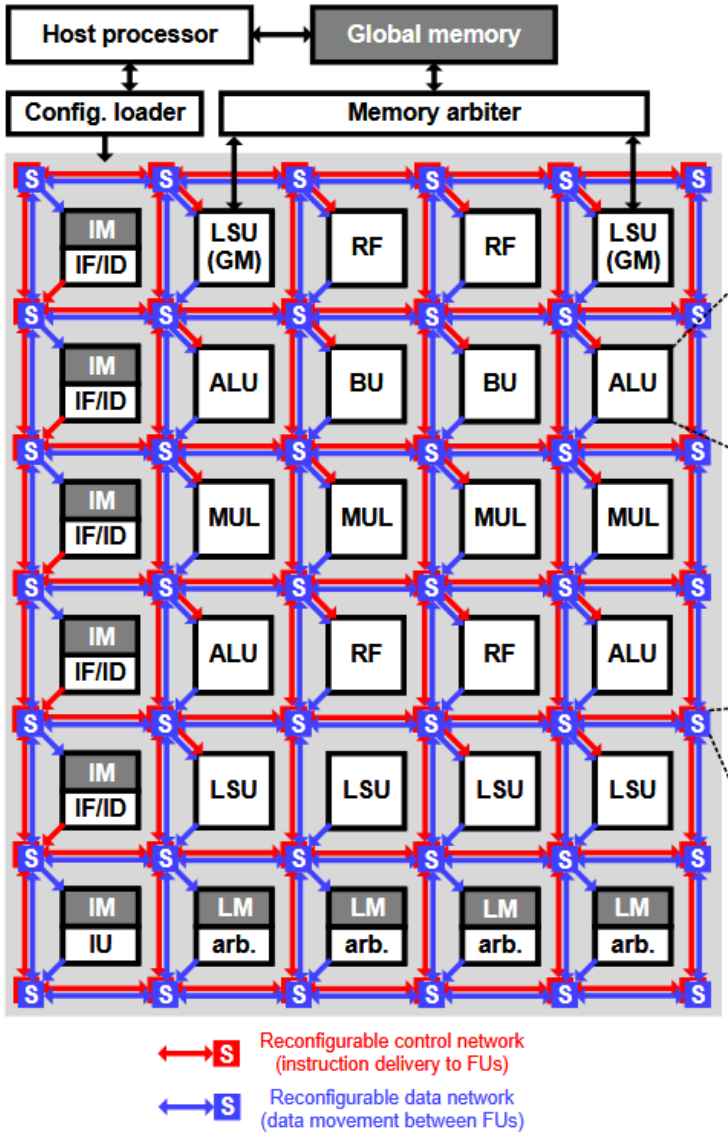


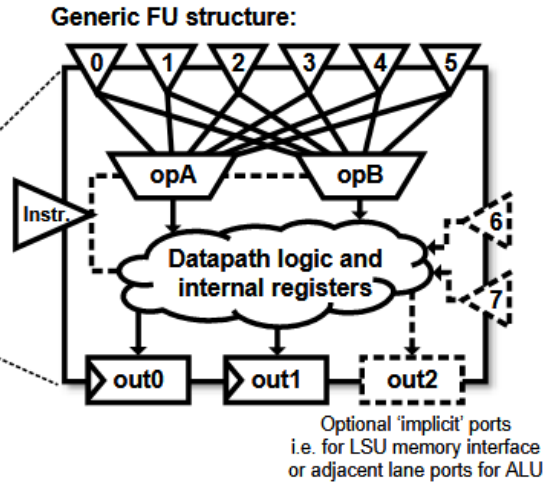


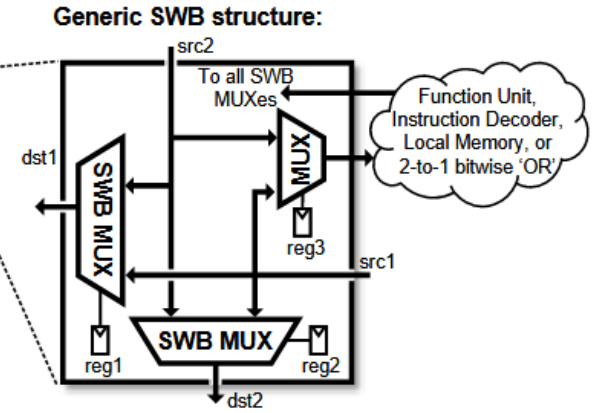
R-Blocks CGRA: Coarse Grain Reconfigurable Array

- Separate NOC for data & Ctrl
- Flexible SIMD support by configuring the control NoC
- Includes approximate functional units







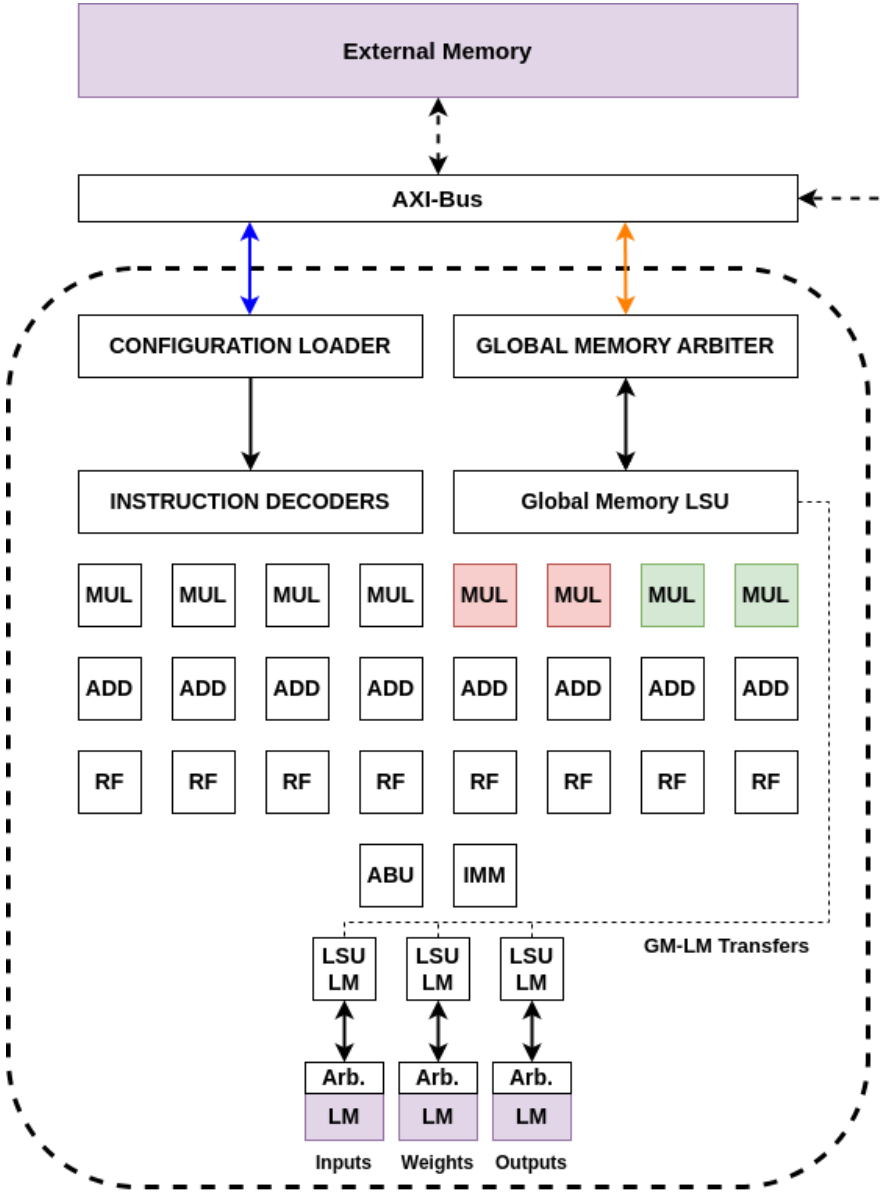


Switch Box (SWB)

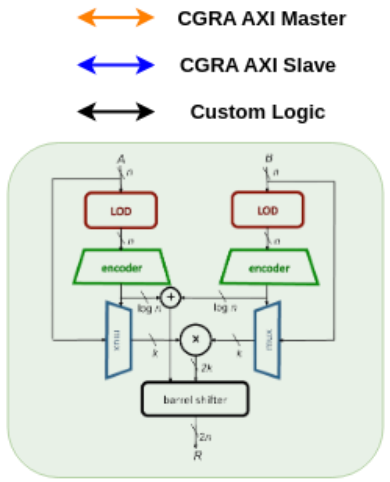
Approximate CGRA: AxC Exploration for R-Blocks

- Modular architecture
- 3 x 4KB Local Mem.
- 8 MAC Units (MUL+ALU)
  - 4 MUL Tiles Utilising 2 Approx. Techniques
  - DRUM[1]
  - ROUP [2]

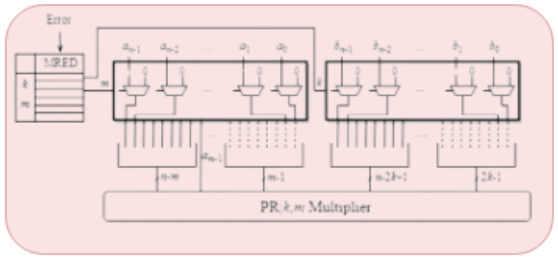
- 13% Area Gain
- 21% Power Gain (67 % for a unit)
- MSE down to 0.15 in YOLOv6



R-Blocks CGRA Architecture



DRUM MUL Tile



ROUP MUL Tile

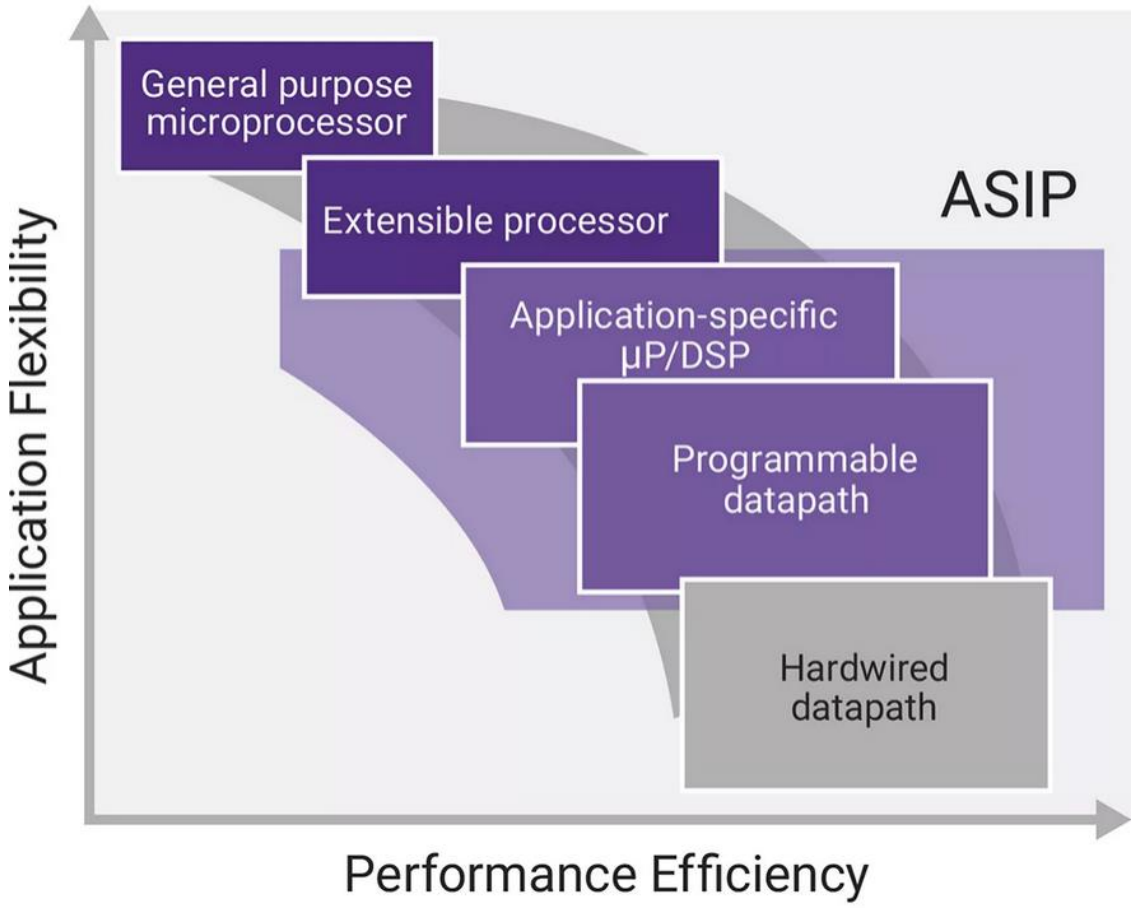
[1] S. Hashemi, R. I. Bahar and S. Reda, "DRUM: A Dynamic Range Unbiased Multiplier for approximate applications," ICCAD 2015

[2] V. Leon, K. Asimakopoulos, S. Xydis, D. Soudris and K. Pekmestzi, "Cooperative Arithmetic-Aware Approximation Techniques for Energy-Efficient

What to expect?

- AI Deep Learning Models
- Edge Mismatch: Cloud vs Edge
- Optimizations
- Learn from the Brain
- SOTA in Edge AI computing
- Future
- **Conclusions**
  - **Flexibility of accelerators**
  - **LEC**
  - **Summary**

Flexibility: tradeoff with Efficiency (energy,area)

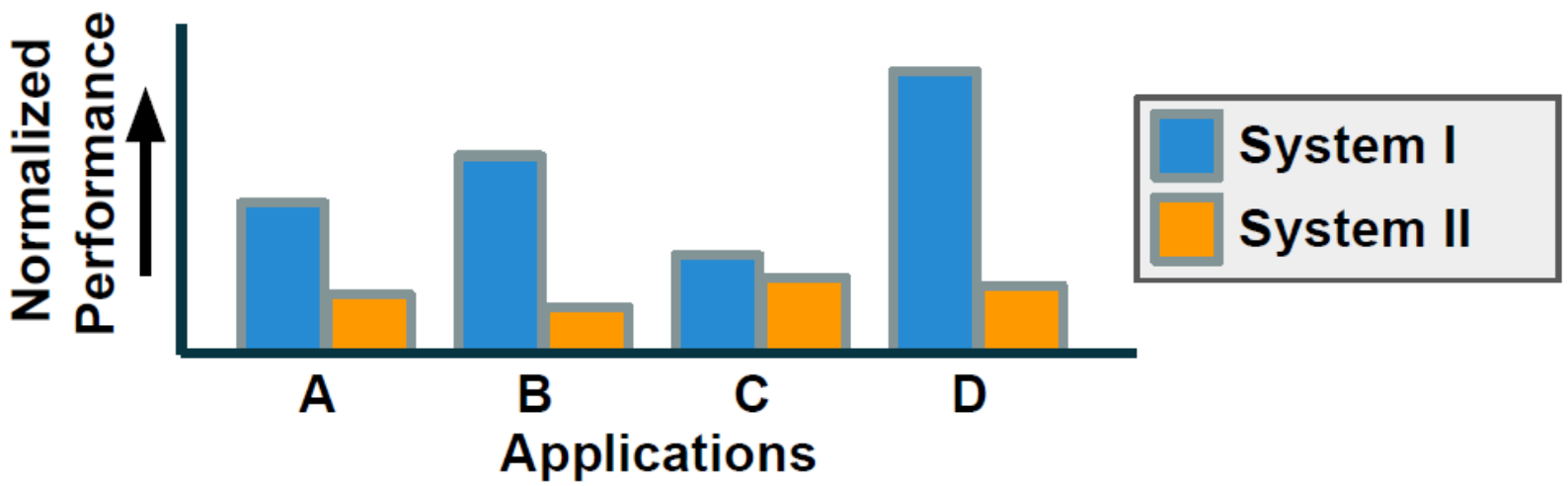


- How to achieve flexibility?
  - Heterogeneous platform
    - multi-core with many dedicated accelerators
    - dark silicon issue
  - CGRA multi-domain platform

Markus Willems. 2019. Application-Specific Processors for High Throughput, Low Latency, and Flexible 5G Communication SoCs. Synopsis. https://www.synopsys.com/designware-ip/technical-bulletin/5gasips- communication-socs.html

Flexibility metric

- Performance of 2 systems compared to a reference, RISC CPU
  - Which one is more flexible?



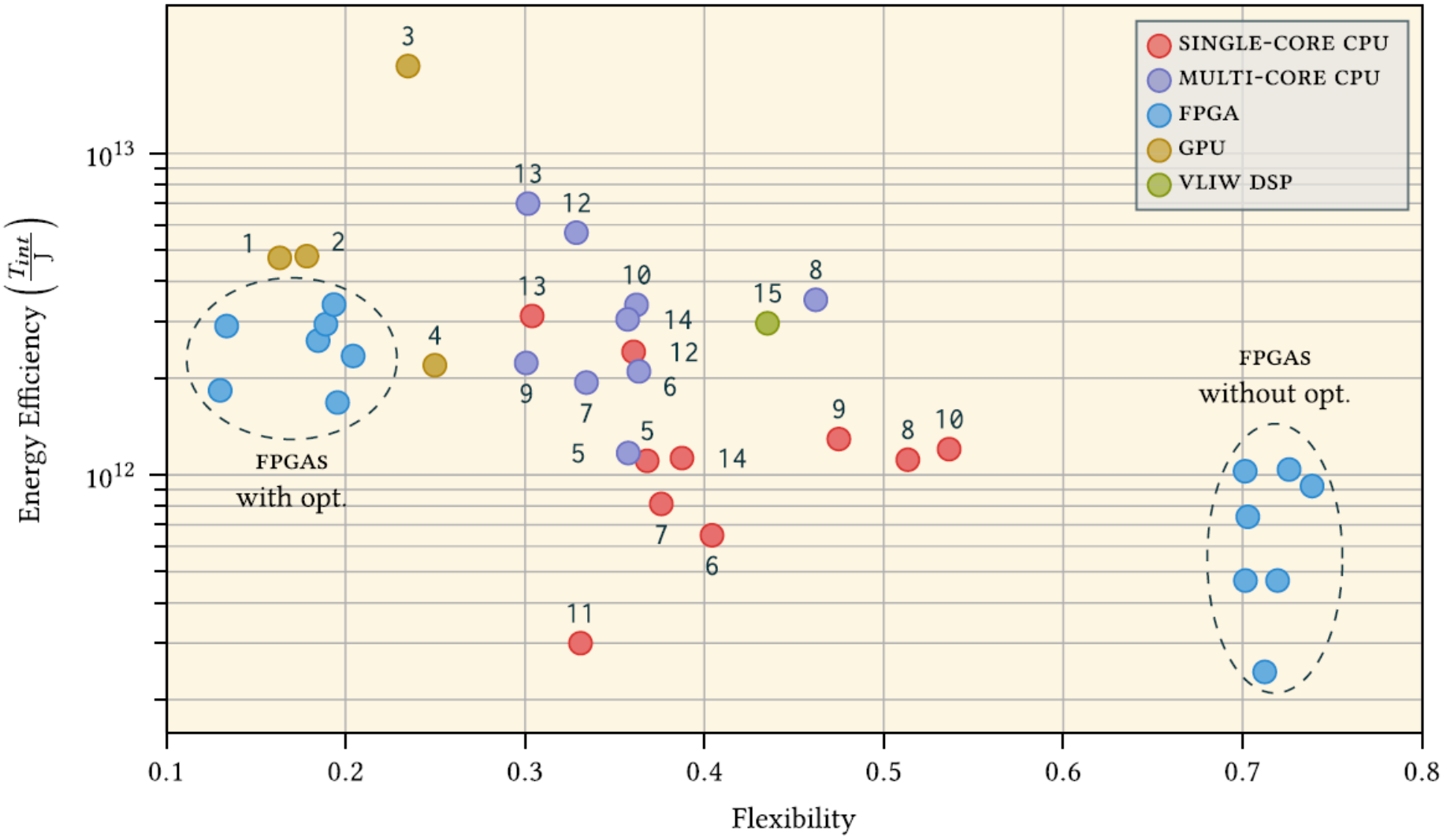
Compute system flexibility = the inverse of the geometric standard deviation of a system's normalized performance, energy efficiency, or other secondary metric, within a benchmark set

How Flexible is Your Computing System?
Shihua Hung, Luc Waeijen, Henk Corporaal
ACM TECS Aug '22

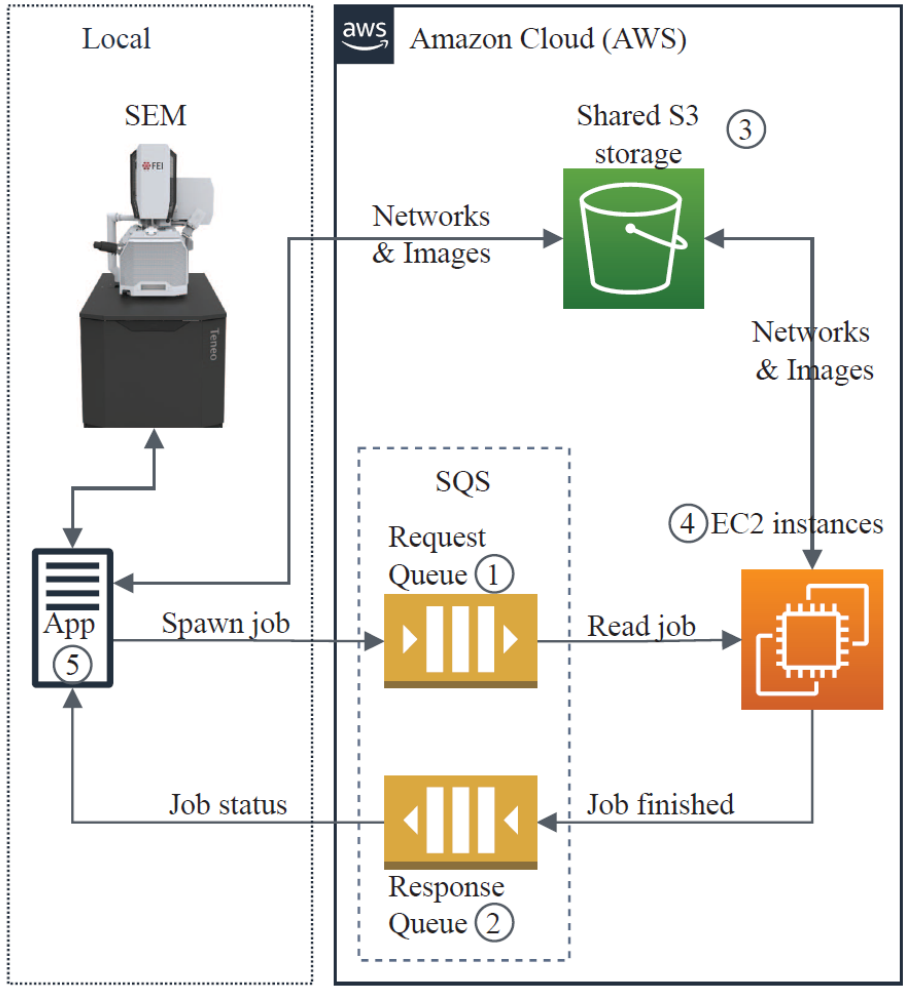
$$Flexibility(\vec{X}) = \left[GSD(\vec{X})\right]^{-1} = \exp\left(-\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\ln\frac{x_i}{GM(\vec{X})}\right)^2}\right),\ where\ GM(\vec{X}) = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

Energy-efficiency vs Flexibility

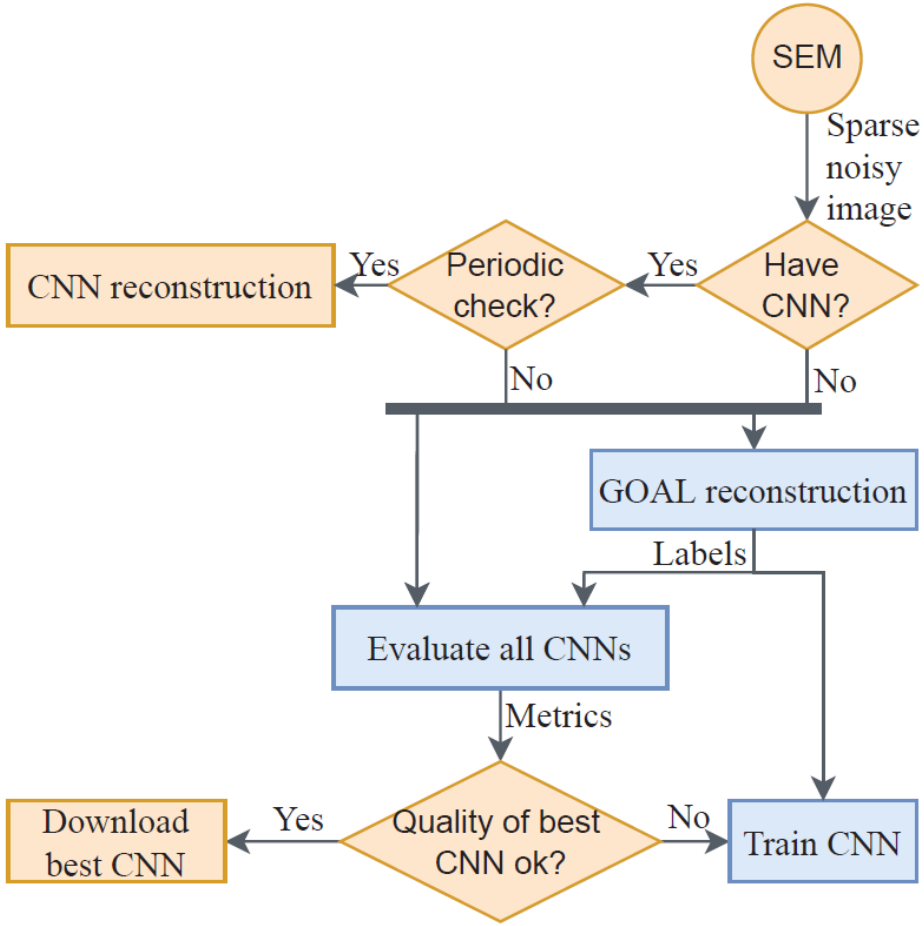
- Look at FPGAs
- and GPUs ?
- Multi-core more energy-eff than single-core



LEC: Liquid Edge-Cloud processing example



Electron Microscope using NN for image enhancement, enabling fast scan

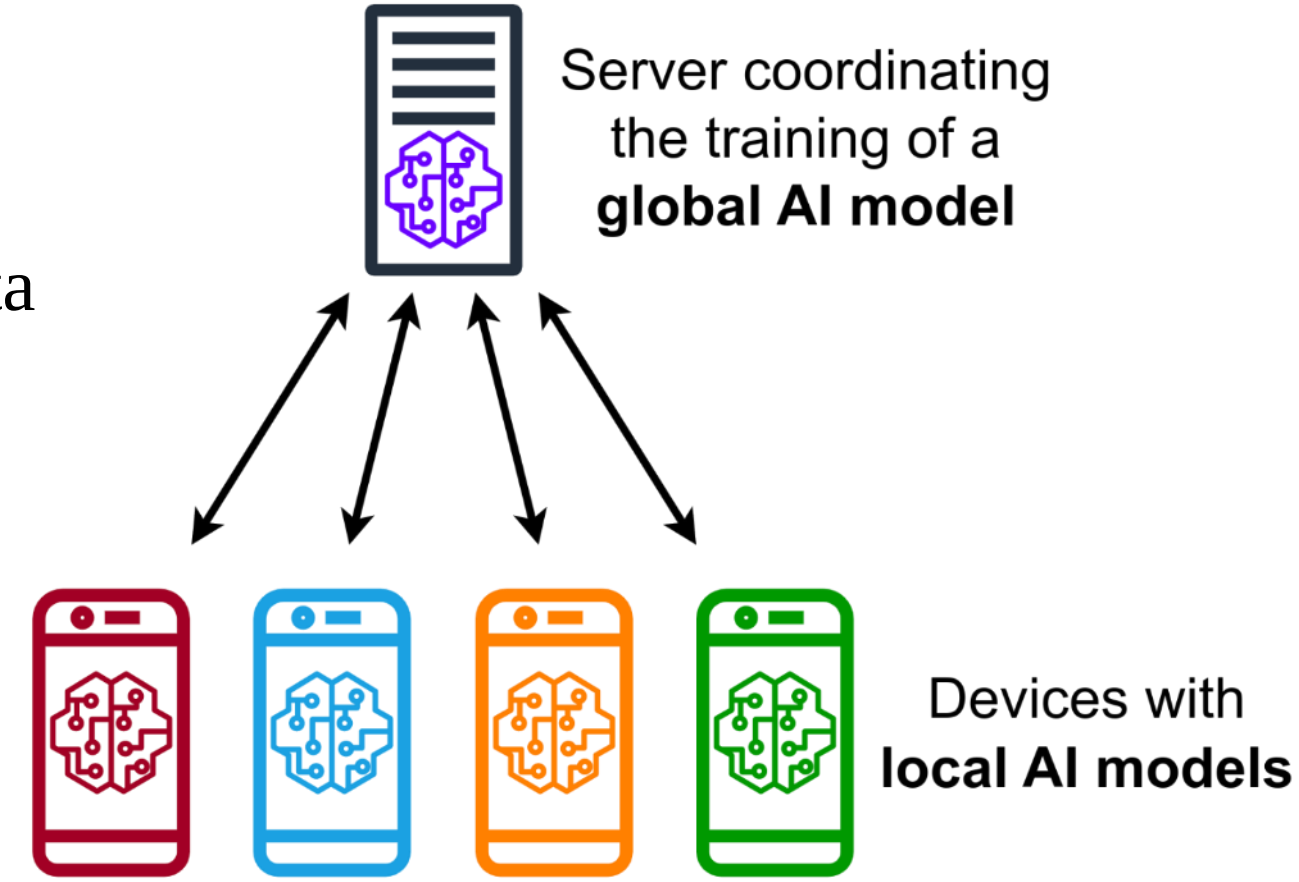


Orange: @Edge, Blue in Cloud

Floran De Putter, "CELR: Cloud Enhanced Local Reconstruction from low-dose sparse Scanning Electron Microscopy images", DSD 2022

One step further: Federated Learning

- Collaborative learning by multiple edge devices
- Devices can
  - learn with different (private) data
  - learn different modalities (like audio vs video)
  - be heterogeneous, etc.



AI at the Edge: hype or hope?

- SOTA: Edge-AI *Processing HW*:
  - ANN accelerators: close to 1 fJ/Op peak efficiency
    - low precision operands
    - often assume idealities
  - SNN (Neuromorphic) has potential, however needs to catch up
    - High Sparsity / No multipliers / Low latency
  - Limited on-chip storage capabilities
    - Required by LLMs
  - Lack of efficient end-to-end compilation flow

AI at the Edge: hype **and** hope! **YES**

- 100x energy↓ => Whole Design Stack approach:
  - Online learning & NN Model level, e.g. Dynamic NNs
  - Architecture and Compiler level, e.g. New optimizations
  - Implementation, e.g. Accelerators
  - SoC level: System-level modeling, SoC generation
  - Device level:
    - e.g. e-DRAM, & MRAM enables large on-chip models
    - Emergent technologies, like Memristors
- Cooperation with Cloud
  - but with all edge advantages