3rd TWIN-RELECT Training School on Fault-Tolerant AI Systems

Safe and Reliable Neuromorphic Computing

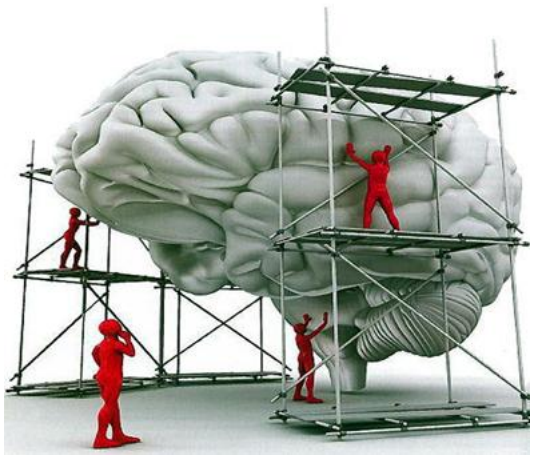
Haralampos-G. STRATIGOPOULOS
Sorbonne Université, CNRS, LIP6
Paris, France

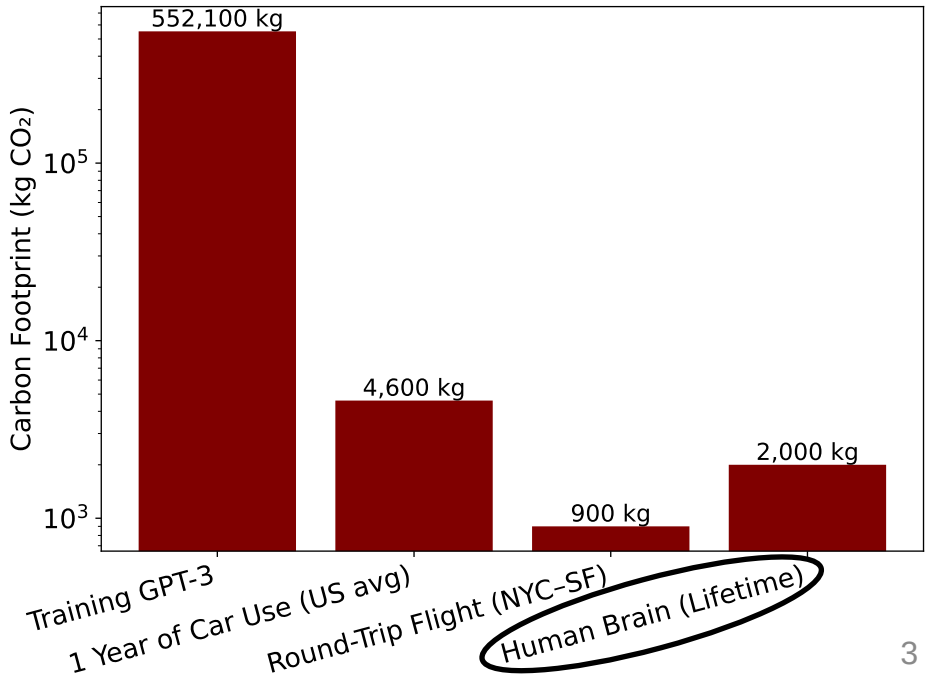
Outline

- Neuromorphic computing and Spiking Neural Networks (SNNs)
- Hardware accelerators
- Reliability assessment
- Testing
- Fault-tolerance
- Conclusions

Brain-inspired neuromorphic computing

- Brain is the most brilliant computing machine
- Very “green”: Computational power efficiency orders of magnitude higher than computers
- Augmented capabilities (learning, produces ideas, error resilience…)
- Brain-inspired neuromorphic computing: replicate the neural architecture, processing mechanisms, and energy efficiency of biological systems using hardware and software models

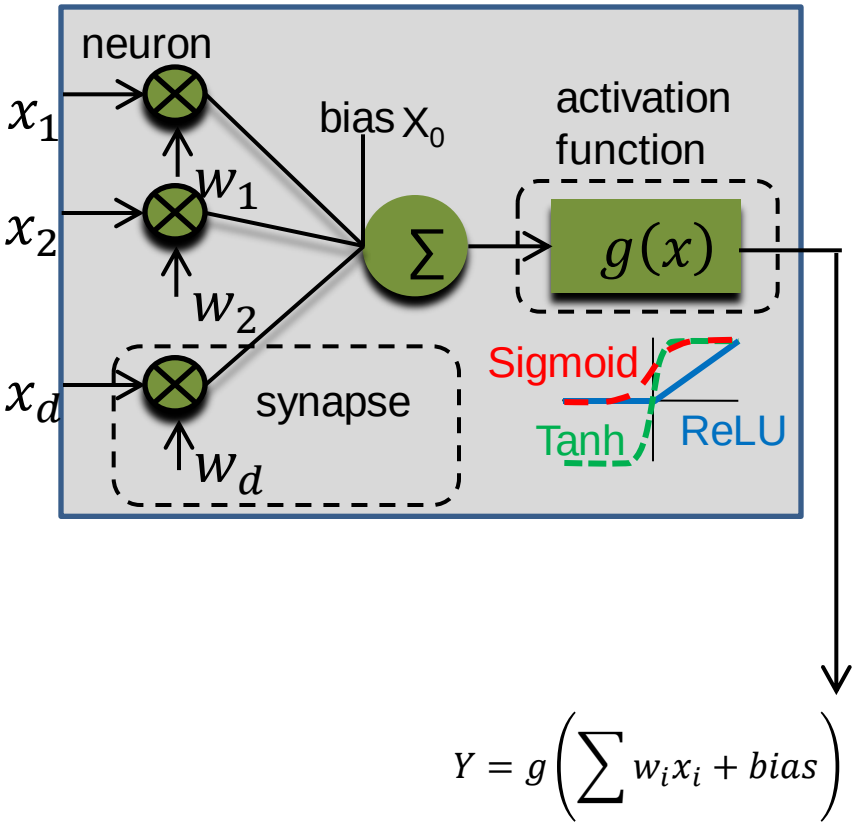




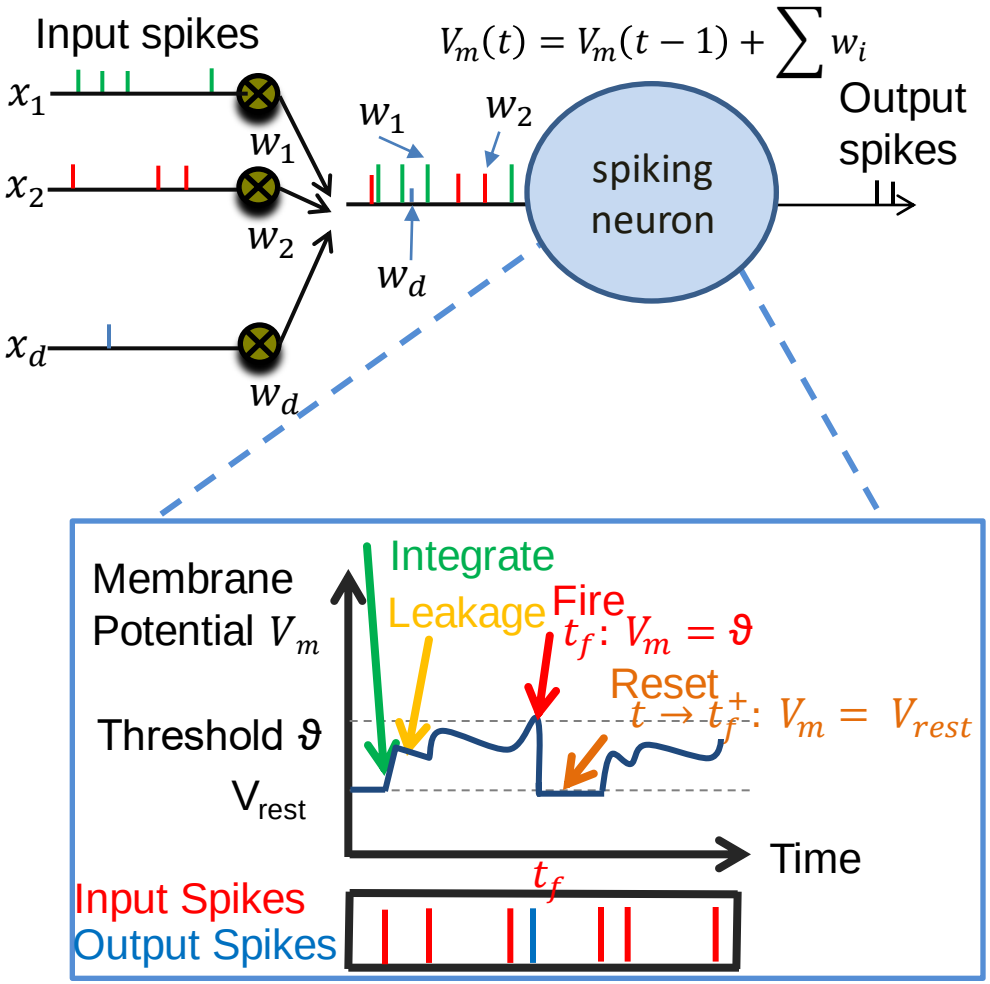
W. Gerstner, *Neuronal dynamics*

Spiking neurons

ANNs



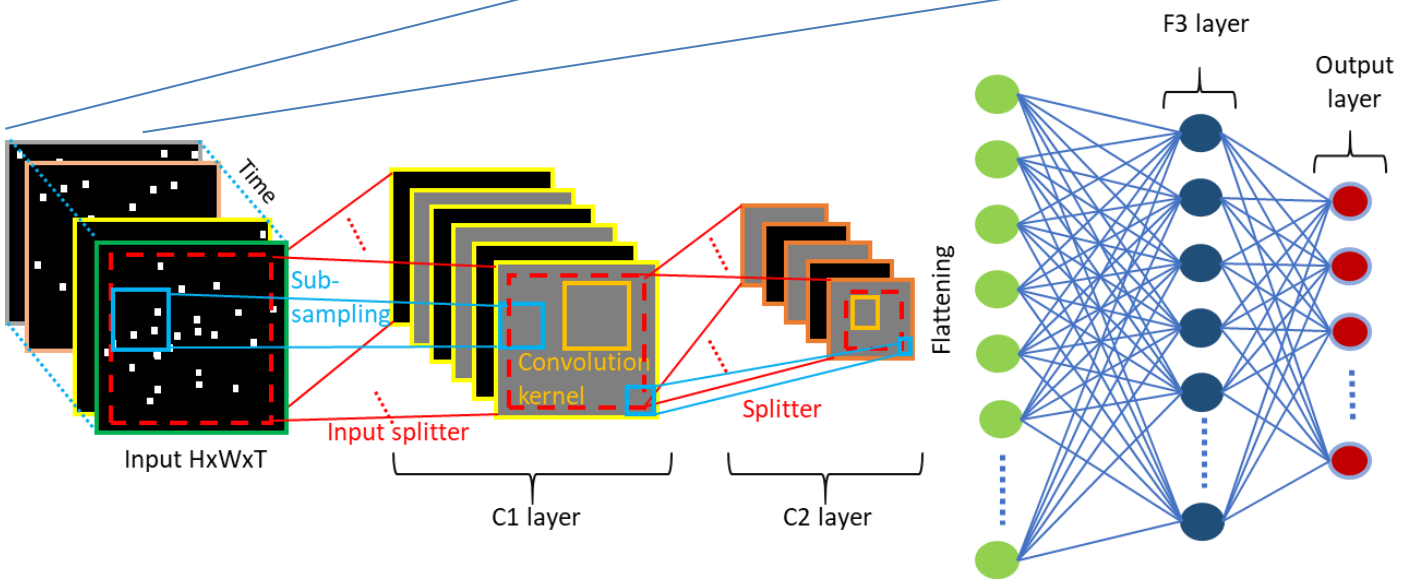
SNNs



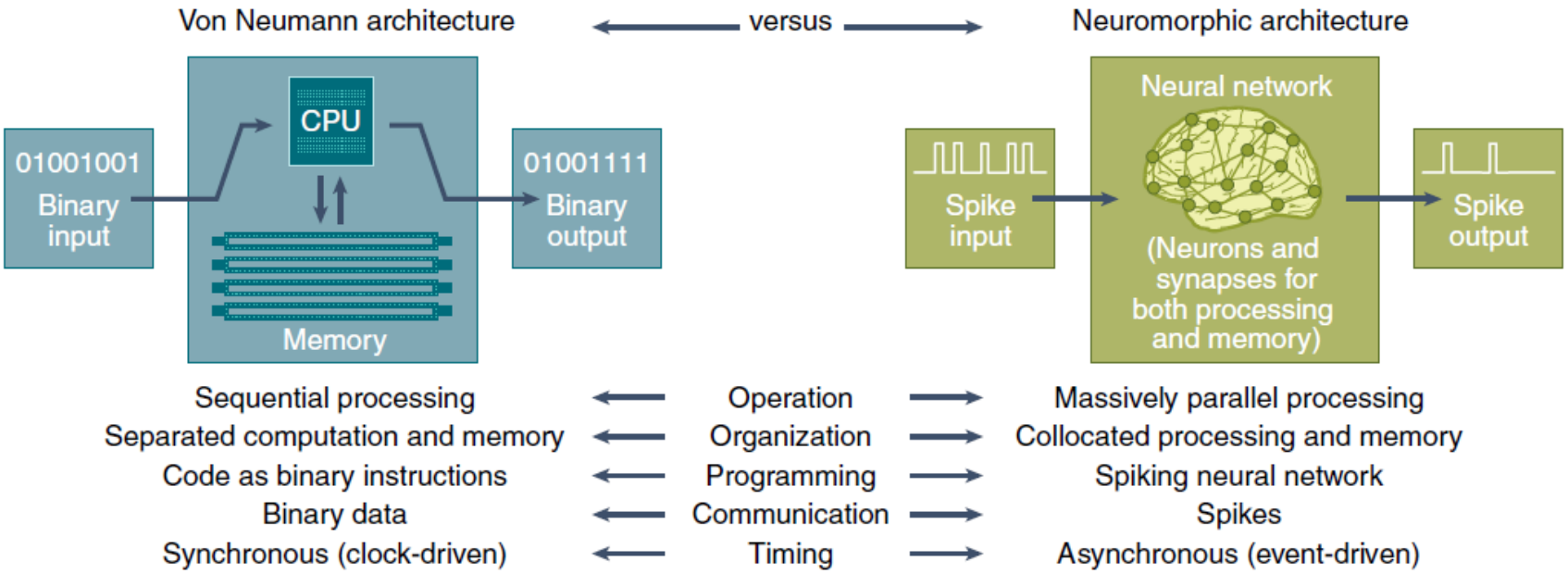
From neuromorphic sensors to Spiking Neural Networks (SNNs)

Dynamic Vision Sensor (DVS):

- Pixels sensitive to scene dynamics
- Produce spikes in response to brightness changes



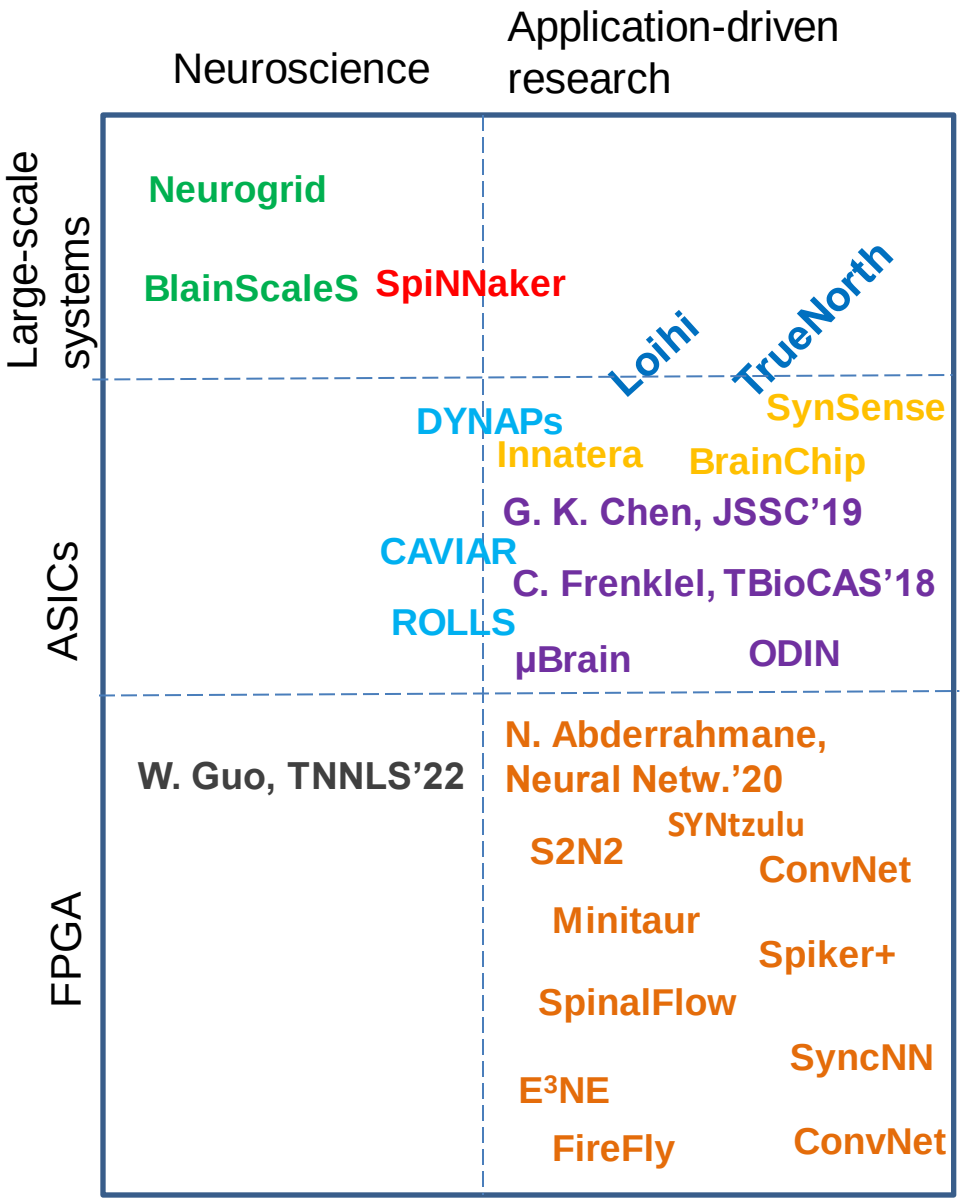
Von Neumann vs. neuromorphic architecture



- Energy efficiency: near-memory computing and event-based operation
- Fast inference: asynchronous event processing
- Challenges in learning
- Challenges in developing hardware

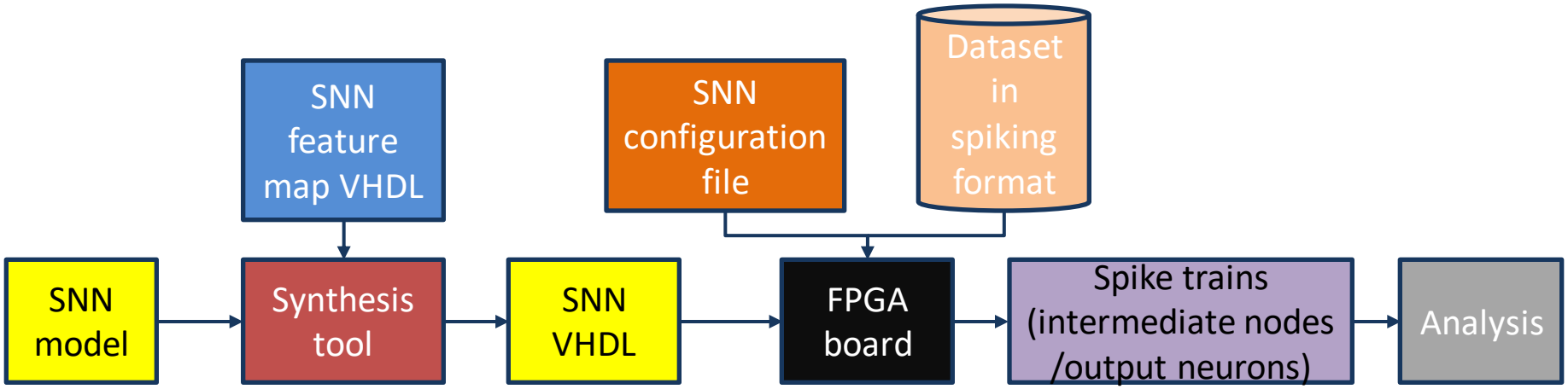
D. C. Schumman, Nature Computational Science, 2022

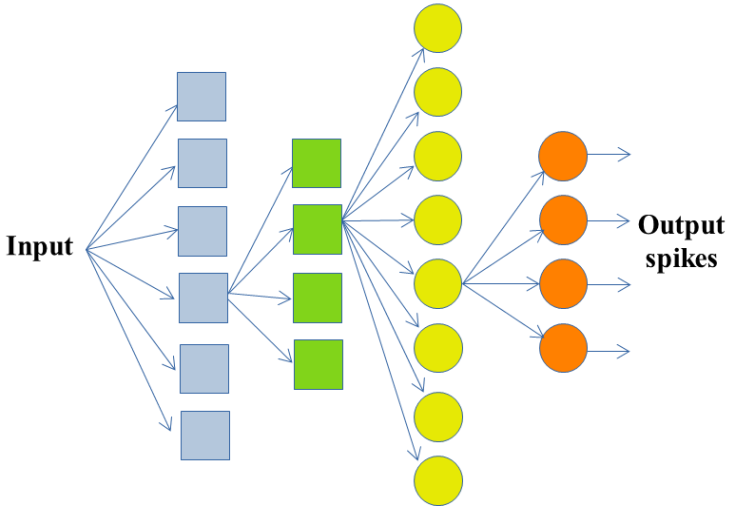
Landscape of neuromorphic hardware

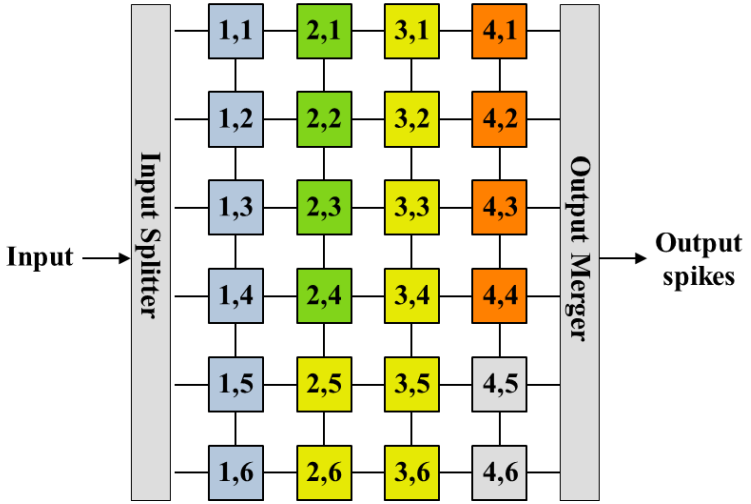


	SpiNNaker	Loihi	TrueNorth
Neurons/ core	36K	130K	1M
Synapses /core	2.8M	130M	256M
Cores/ chip	144	128	4096
Chips/ board	56	768	4096
Neurons	2.5B	100M	4B
Synapses	200B	100B	1T

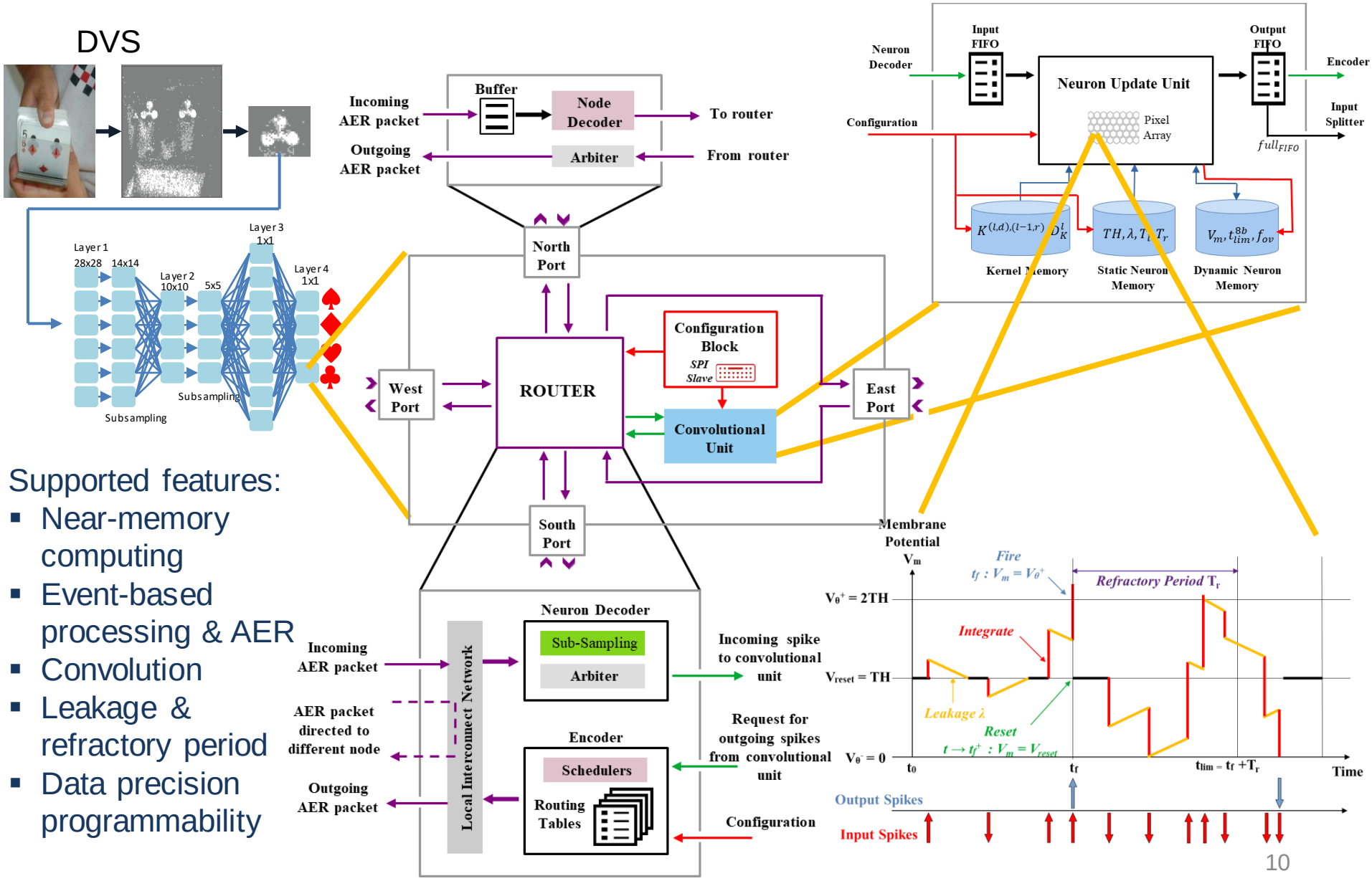
End-to-end model-to-hardware automation flow for SNNs







SNN hardware accelerator architecture



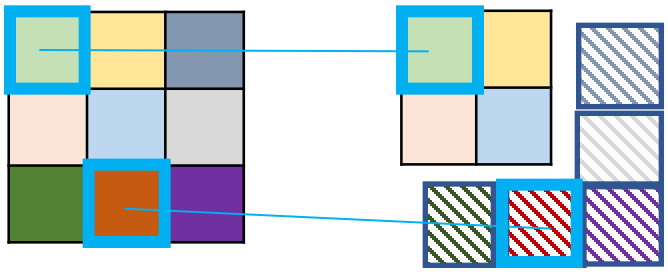
Supported features:

- Near-memory computing
- Event-based processing & AER
- Convolution
- Leakage & refractory period
- Data precision programmability

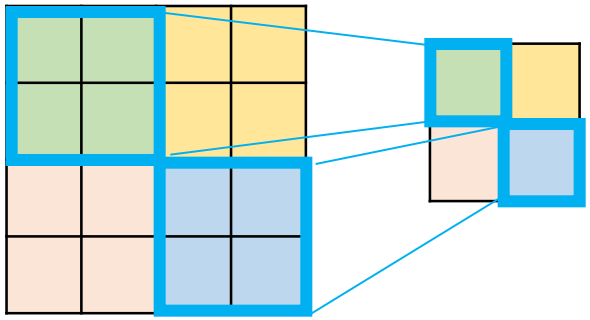
Sub-sampling

$$(i,j) \rightarrow \left(\left\lfloor \frac{i}{2^{\alpha}} \right\rfloor, \left\lfloor \frac{j}{2^{\alpha}} \right\rfloor\right)$$

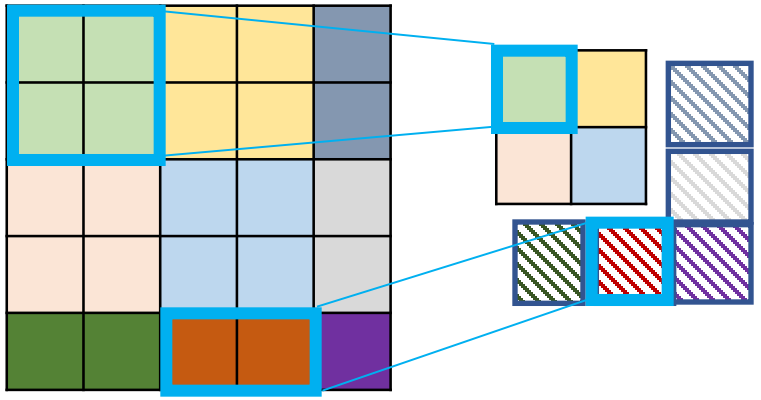
α = 0



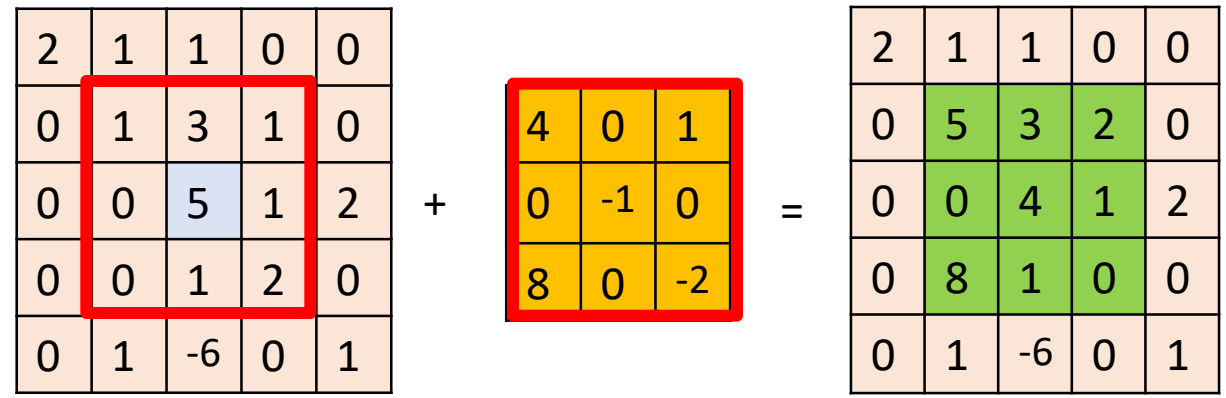
α = 1

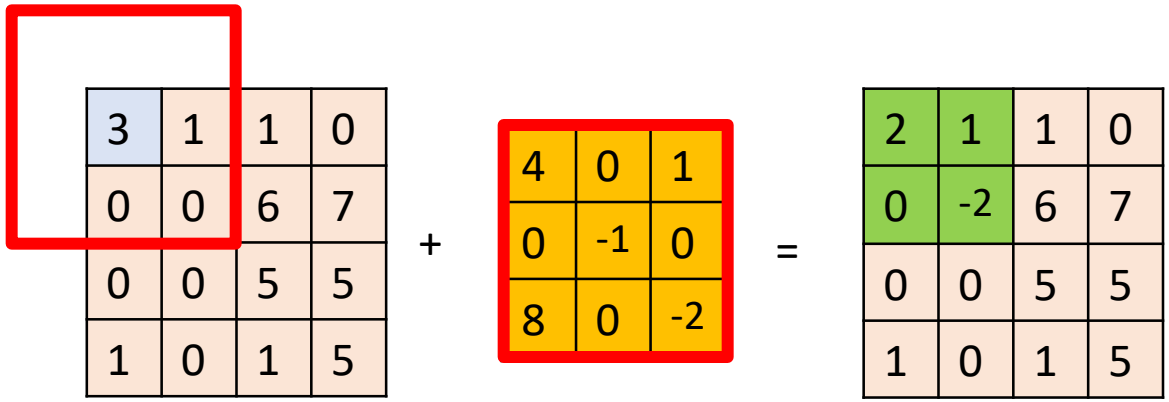


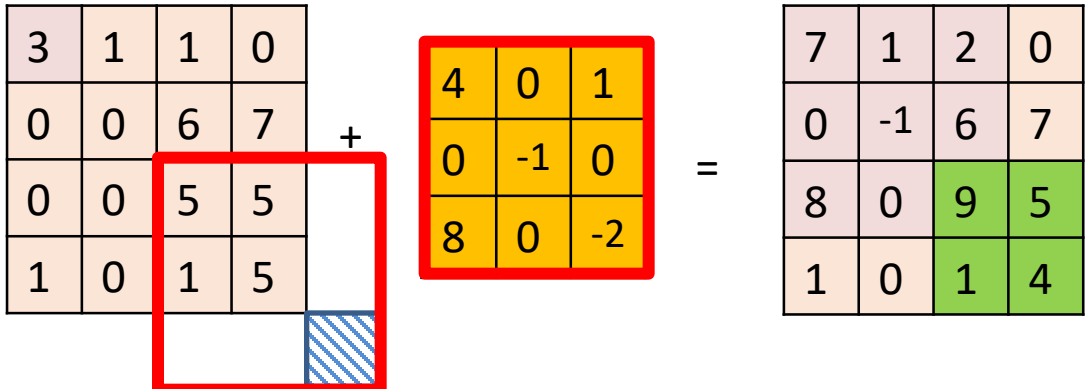
α = 1



Convolutions

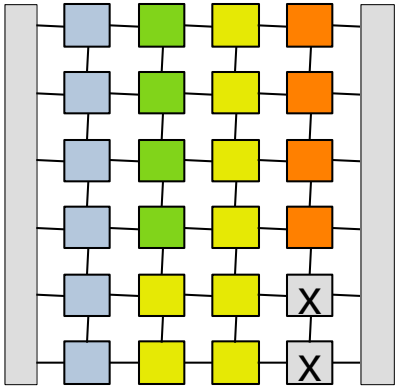




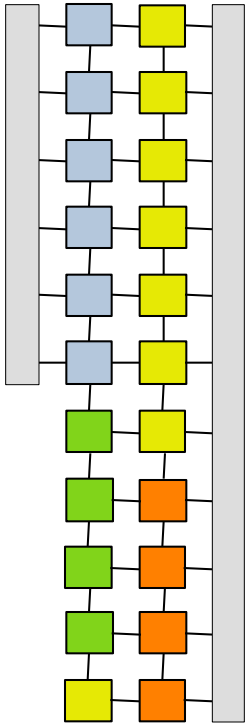


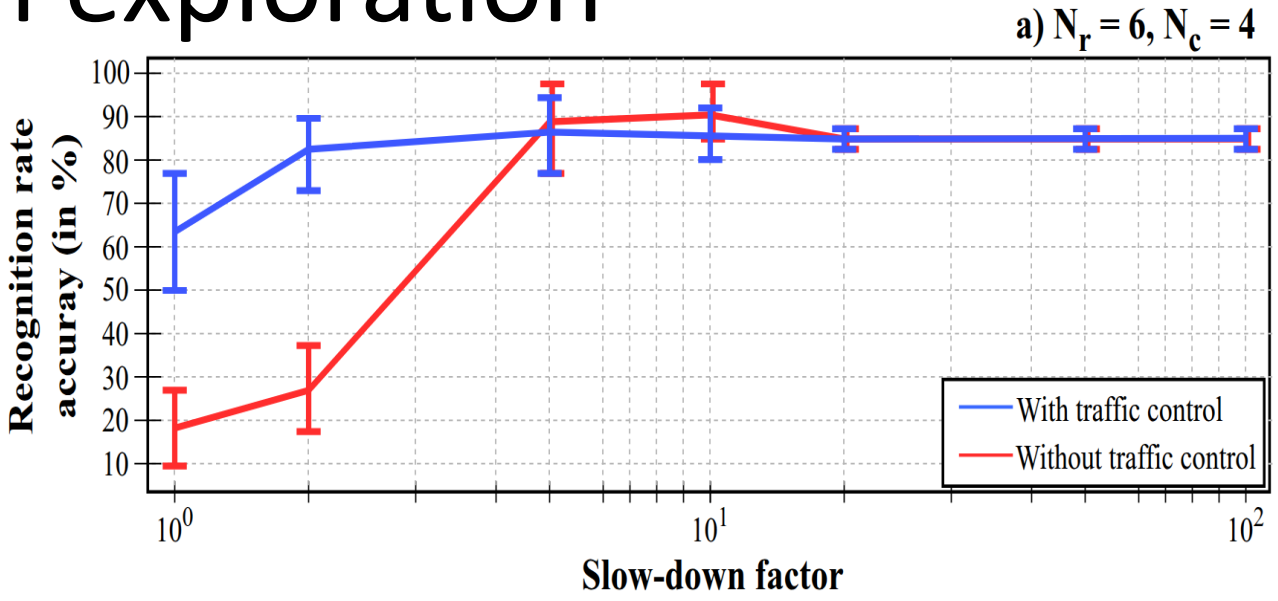
2-D mesh exploration

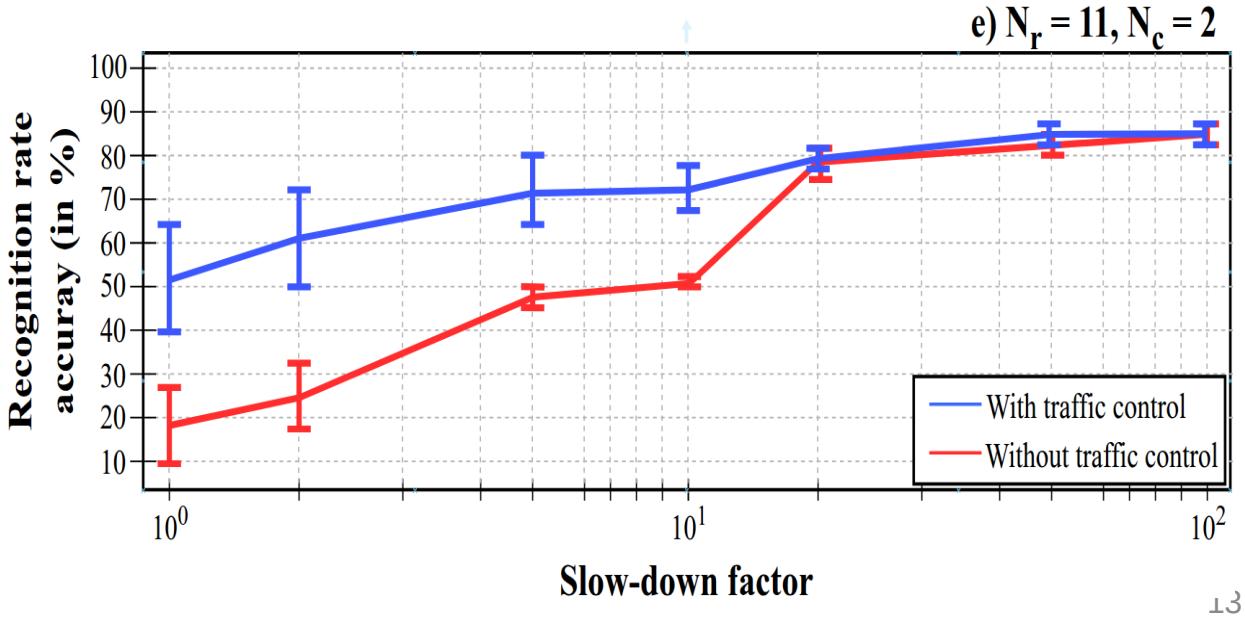
Nr = 6 Nc = 4



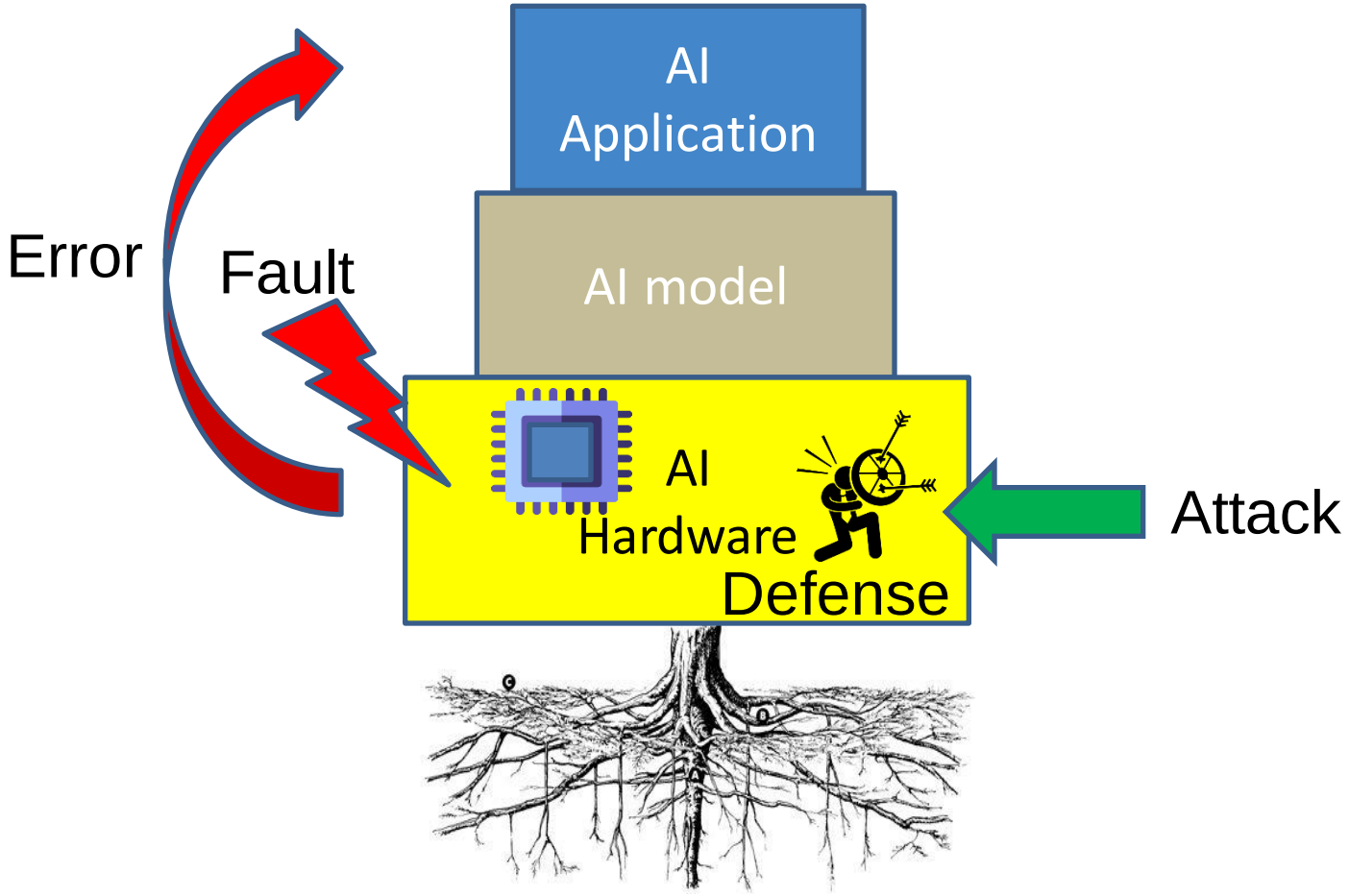
Nr = 11 Nc = 2



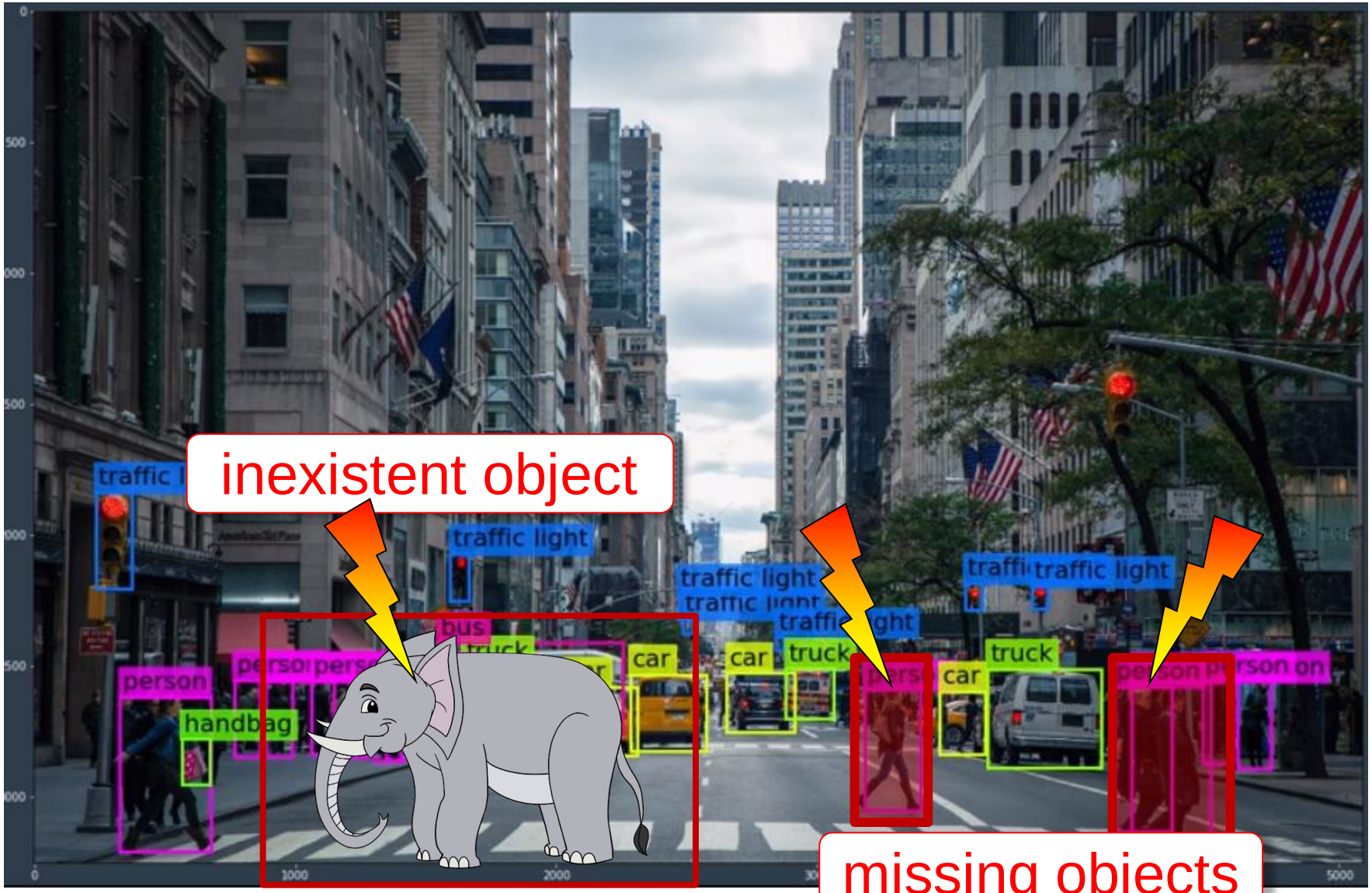




Hardware the Root of Trust?



AI reliability



Source: Paolo Rech, University of Trento

AI hardware in safety-critical applications

- Within the next years, many safety- and mission-critical applications are expected to materialize and exploit the advantages of AI hardware and neuromorphic circuits
- Example: Autonomous-Driving Vehicles
- Safety standard ISO 26262 demands: Errors < 0.2 Failures/10^9 operating hours
  - Low dppm post-manufacturing
  - Self-test during normal operation becomes essential

An automobile may have >400 chips

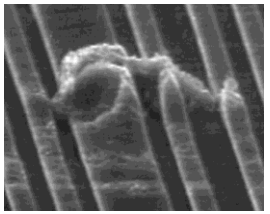
P(Chip fails) = p

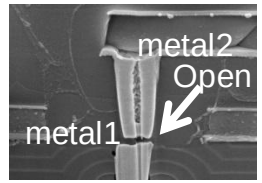
P(System fails) = $1-(1-p)^N$

p=5ppm, N=500

0.25 % failure

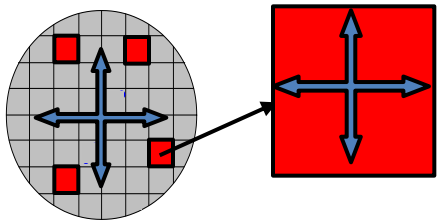
Hardware-level faults

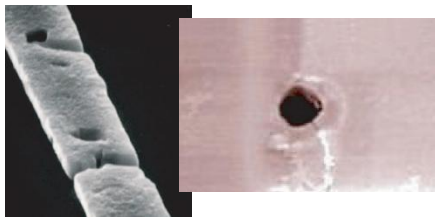


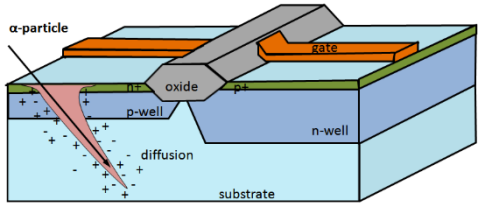


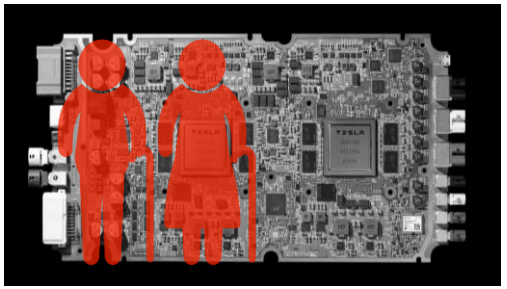
Hardware Faults

- During Manufacturing
  - Process variations
  - Defects
- In the Field
  - Aging
  - Radiation
  - Voltage overstress
  - Temperature fluctuations
  - Low endurance (memristor synapses)
- Before training
- During training
- After Training







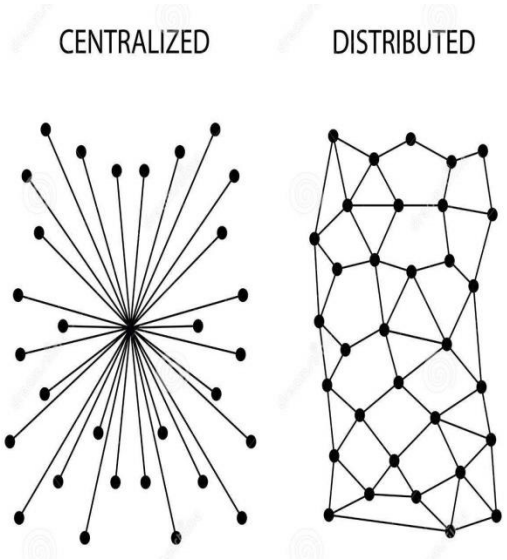


Are AI hardware accelerators fault-tolerant?

They inherit the r… fault-tolerance capabilities of …gical brain

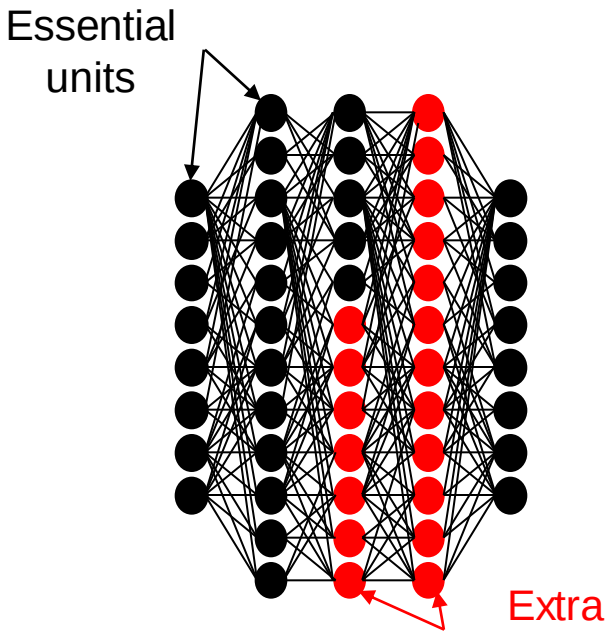
False

Distributed Processing



Biology-Inspired Architectures

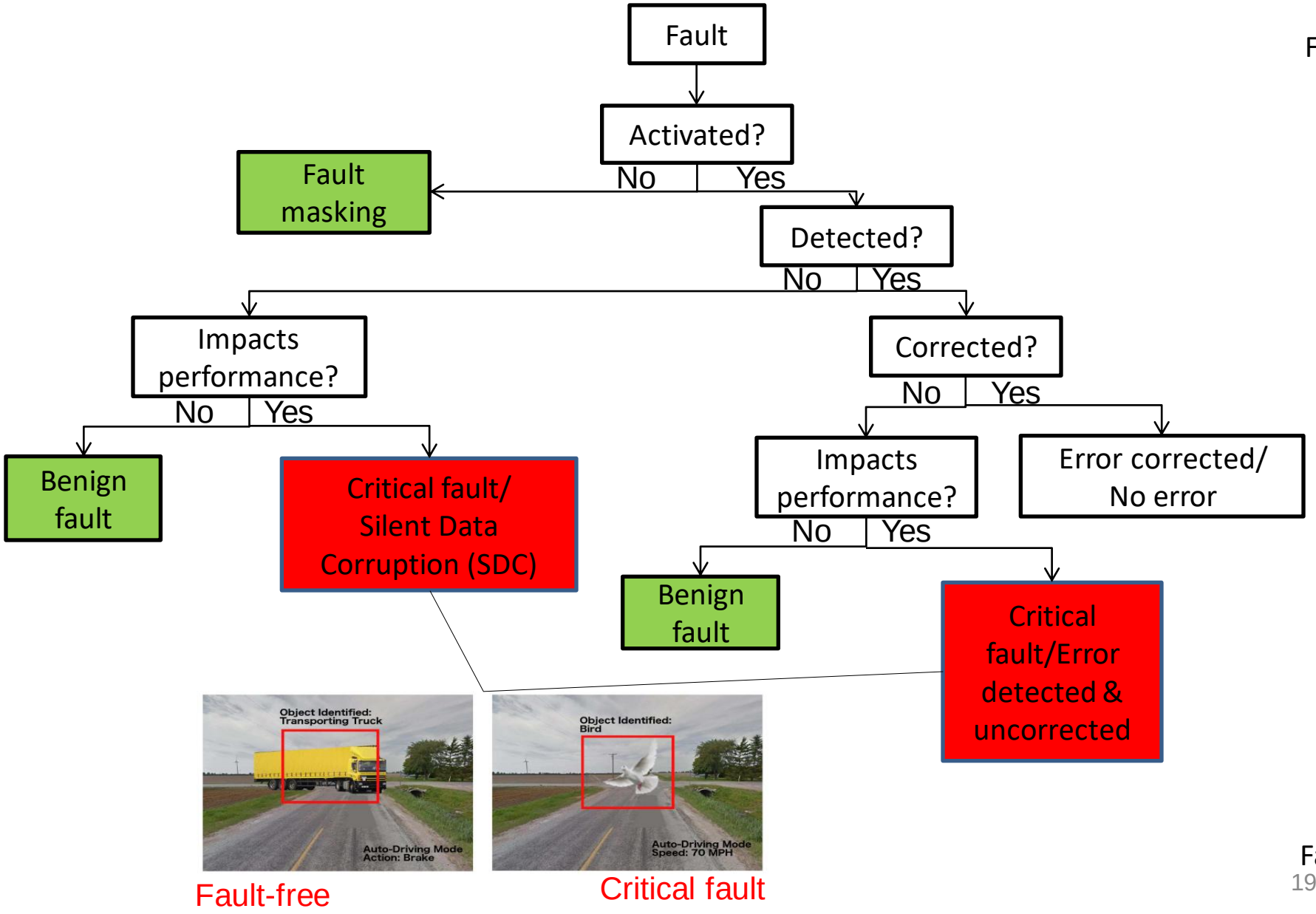
Overprovisioning



- Many fault simulation experiments have shown that this assumption is false

Li, SC'16
Reagan, DAC'18
Zhang, D&T'19
Vatajelu, VTS'19
Neggaz, D&T'20

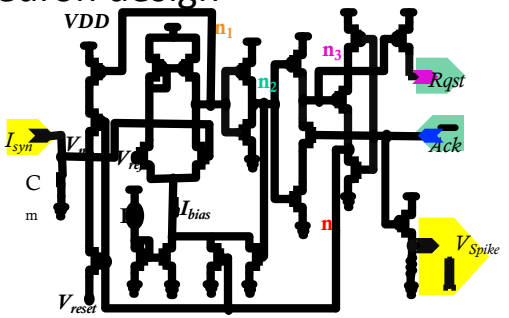
Not all faults are equal!



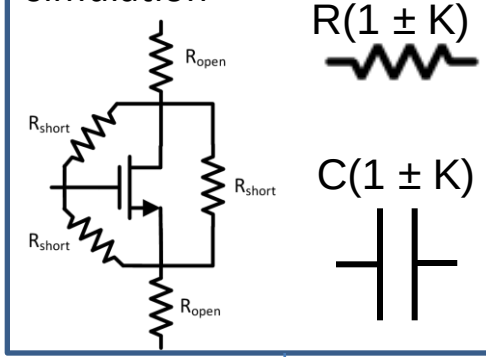
Testability and Reliability Framework

Fault modeling

Neuron design



Transistor-level fault simulation



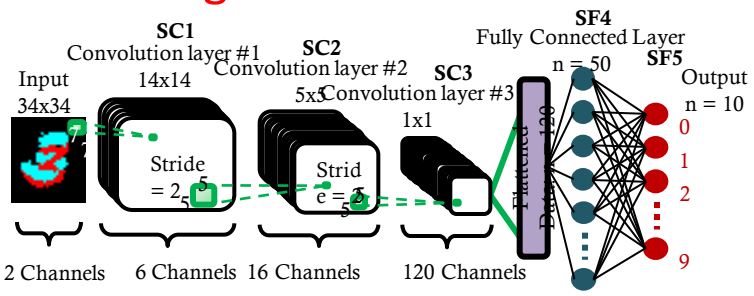
R(1 ± K)

C(1 ± K)

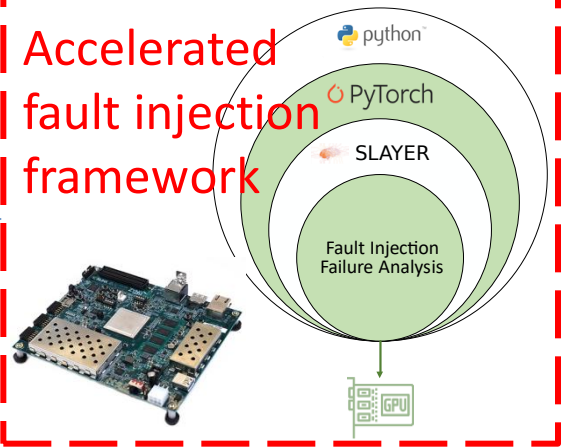
Faulty behaviors

Behavioral –level fault model

SNN designs



Accelerated fault injection framework

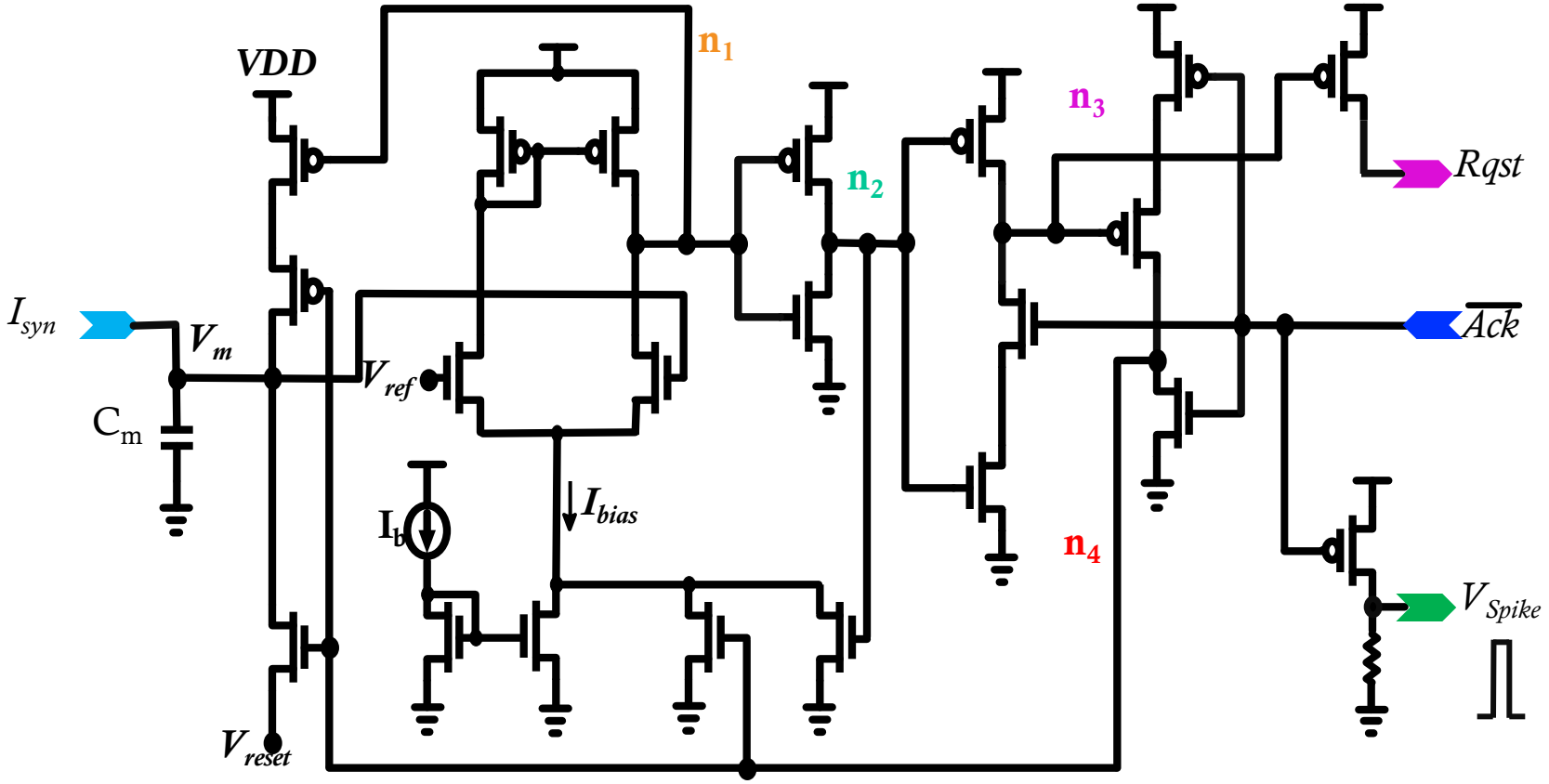


Reliability assessment

Testing & Fault tolerance

- Testing
- Model-based fault tolerance
- Hardware-based fault tolerance

I&F spiking neuron circuit

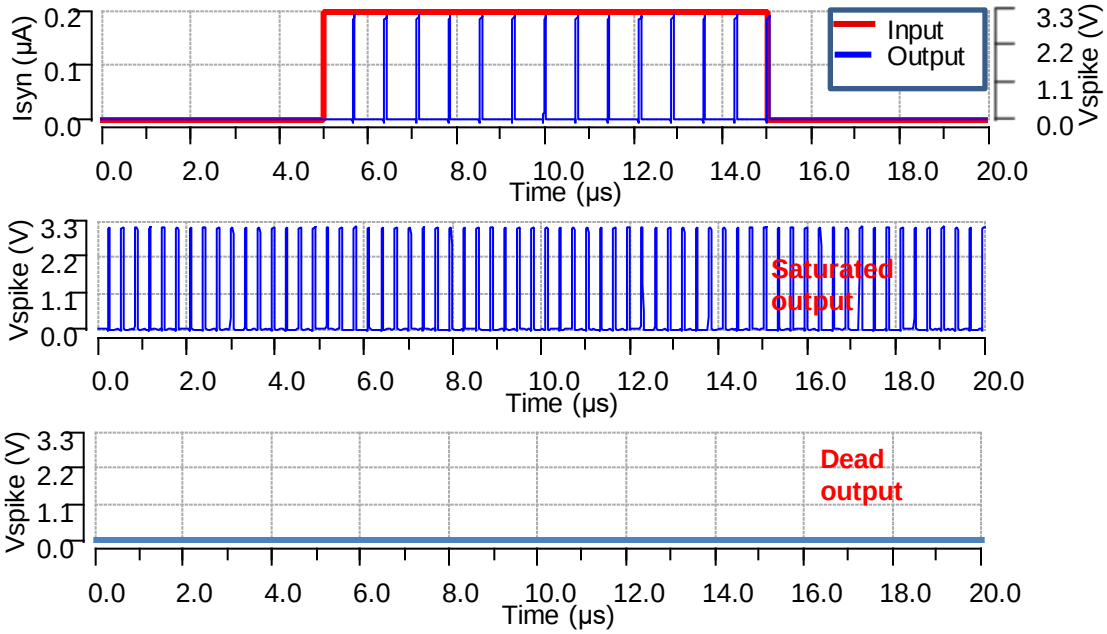


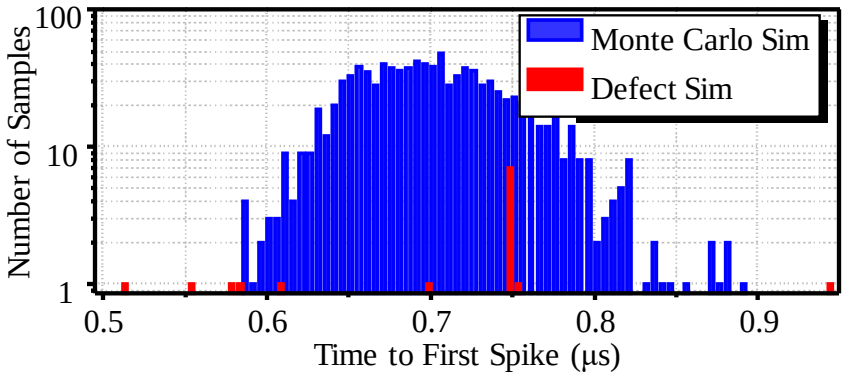
R. Serrano-Gotarredona, *TCAS*'06

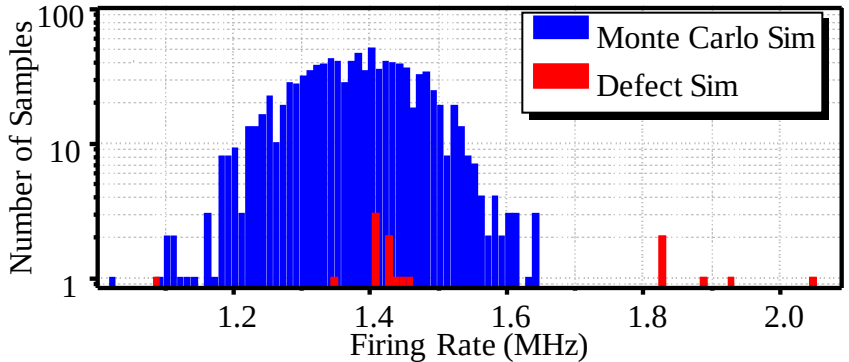
Transistor-level fault simulation

- 1000 MC runs using PDK
- Defect simulation using DefectSim by Siemens S. Sunter, *TCAS-I*'16
- Two types of faulty behaviors:
  - Catastrophic: neuron non-functional (observed for 31 defects)
  - Parametric: output spike train with timing variations (observed for MC and 15 defects)

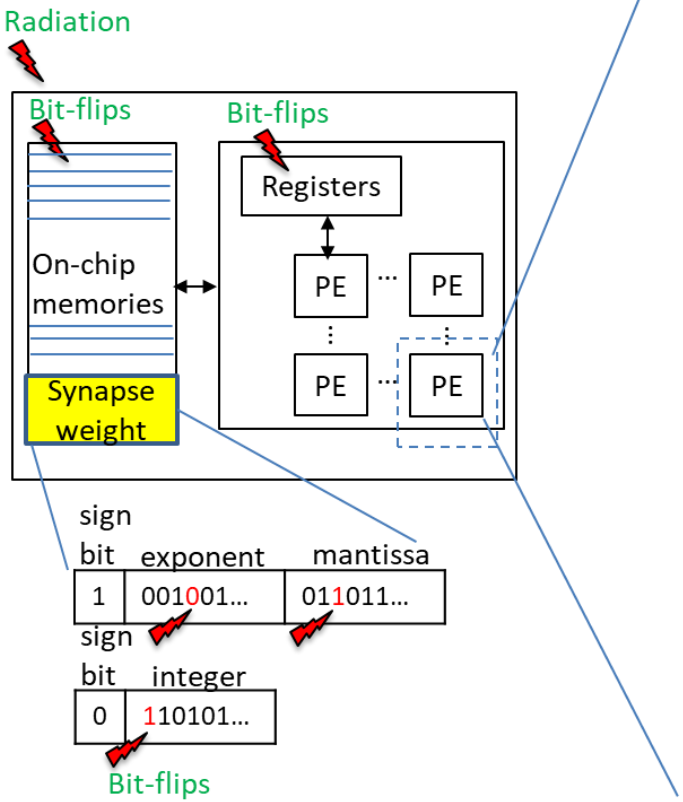
S. Elsayed, *IOLTS*'20

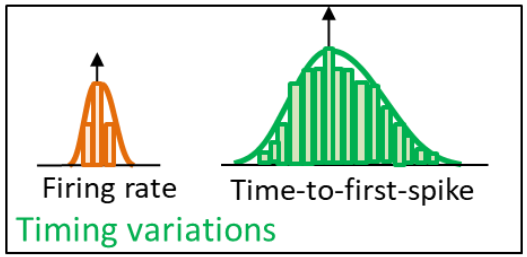


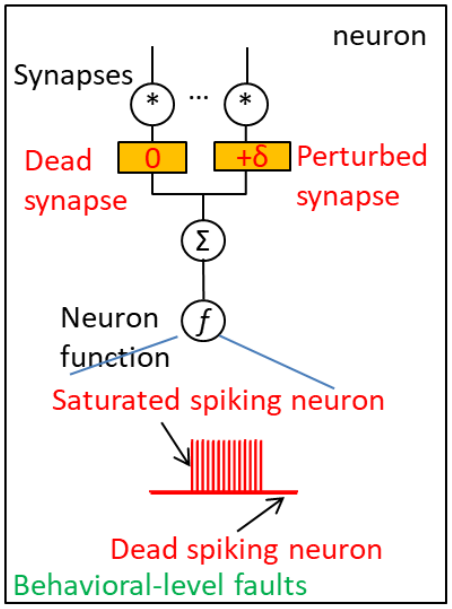


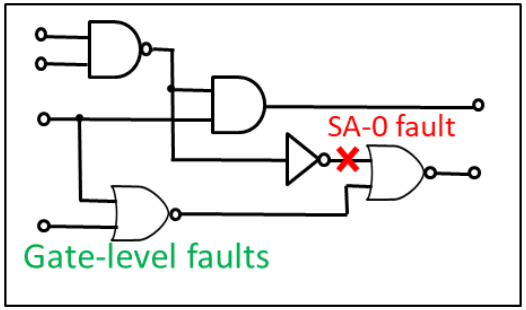


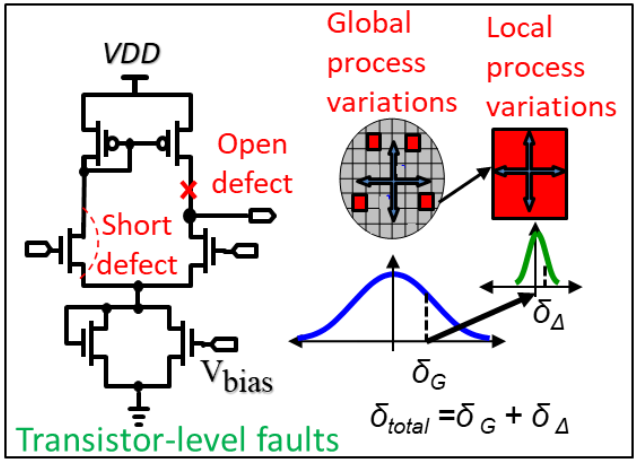
Hardware-level fault modelling





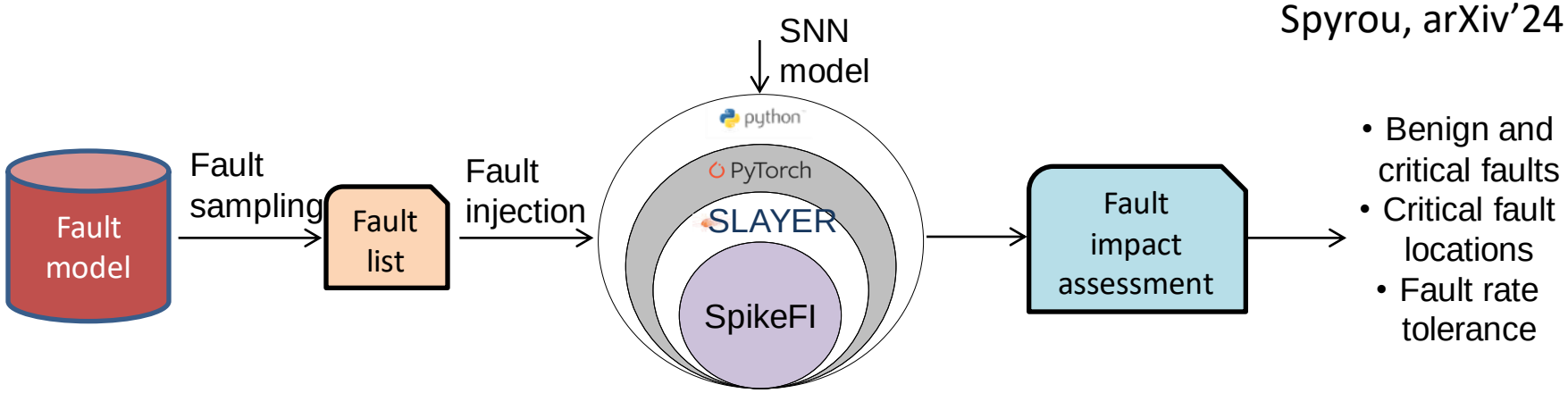






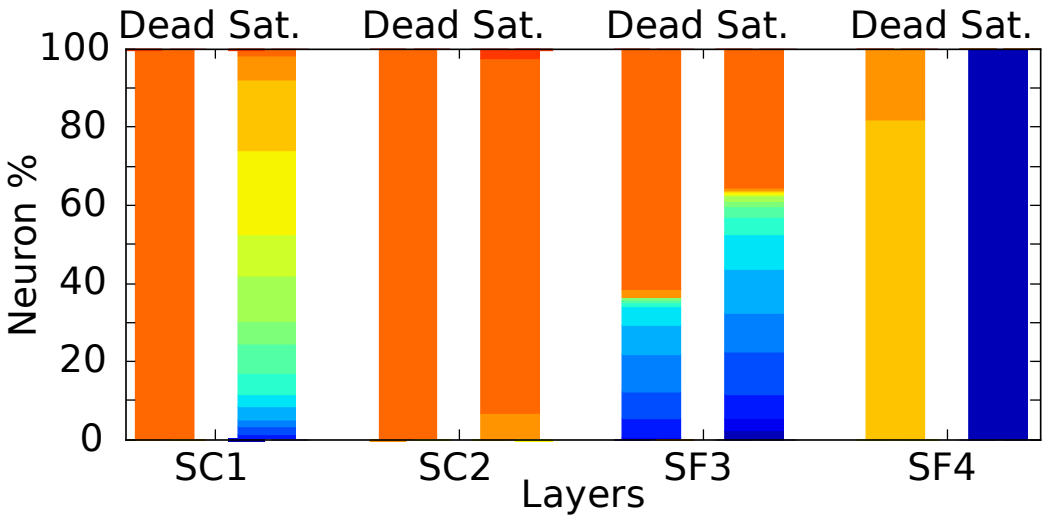
SpikeFI: Fault injection tool for SNNs

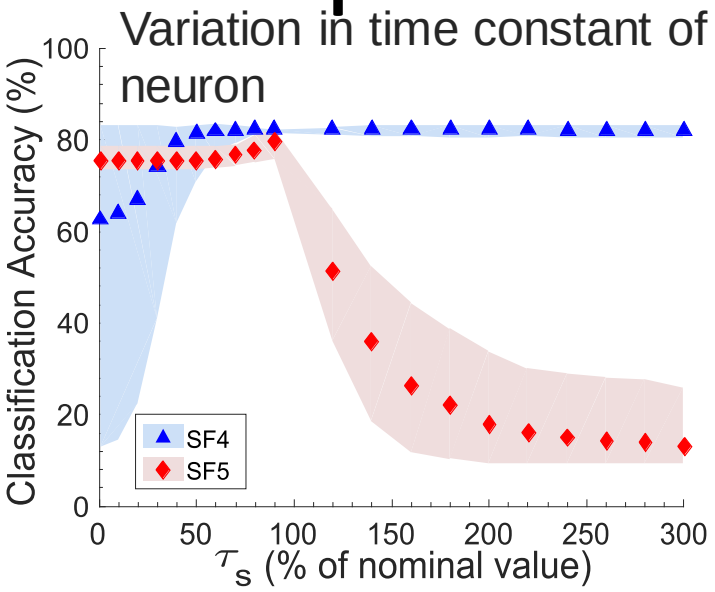
Spyrou, arXiv'24

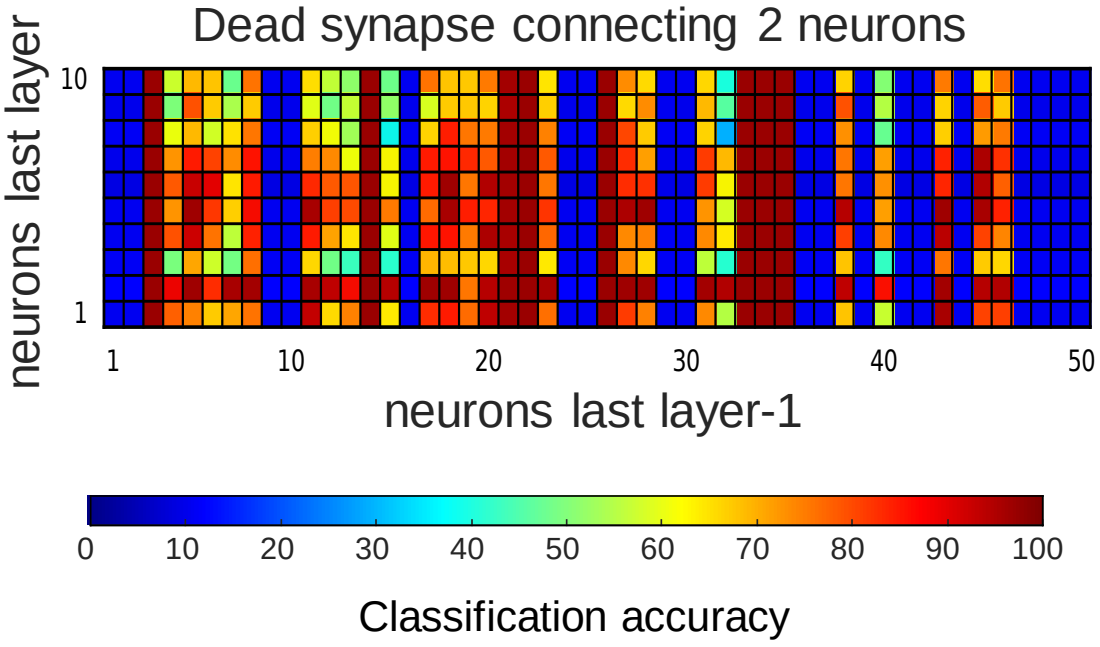


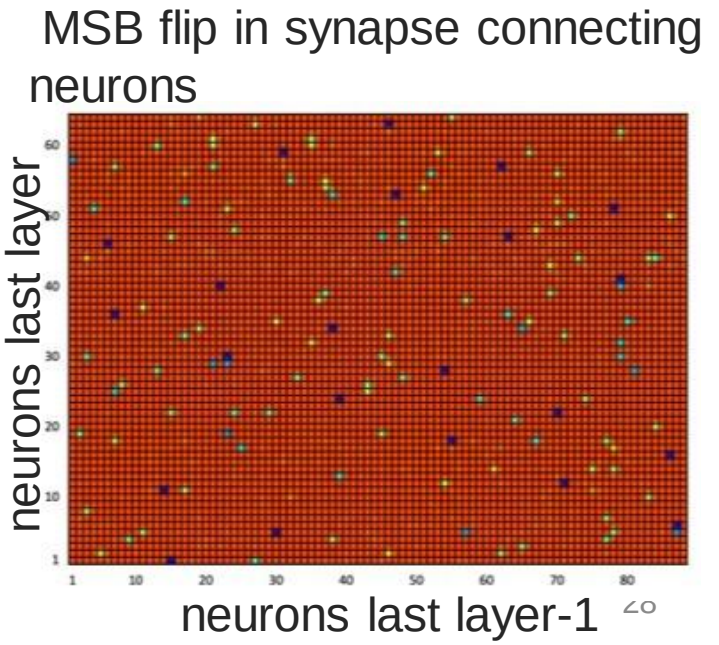
- SpikeFI: PyTorch framework built upon the SNN training framework SLAYER S. B. Shrestha, *NeurIPS*'18
- SpikeFI evaluates the reliability of an SNN model, i.e., if a hardware fault occurs what is the effect on the application?
- Supports: single/multiple and permanent/transient fault injection; fault injection before, during or after training
- Identifies critical faults and gives feedback for fault tolerant design
- Fault simulation acceleration: early stopping, late start, GPU
- Open source: https://github.com/SpikeFI/SpikeFI

Reliability assessment with SpikeFI

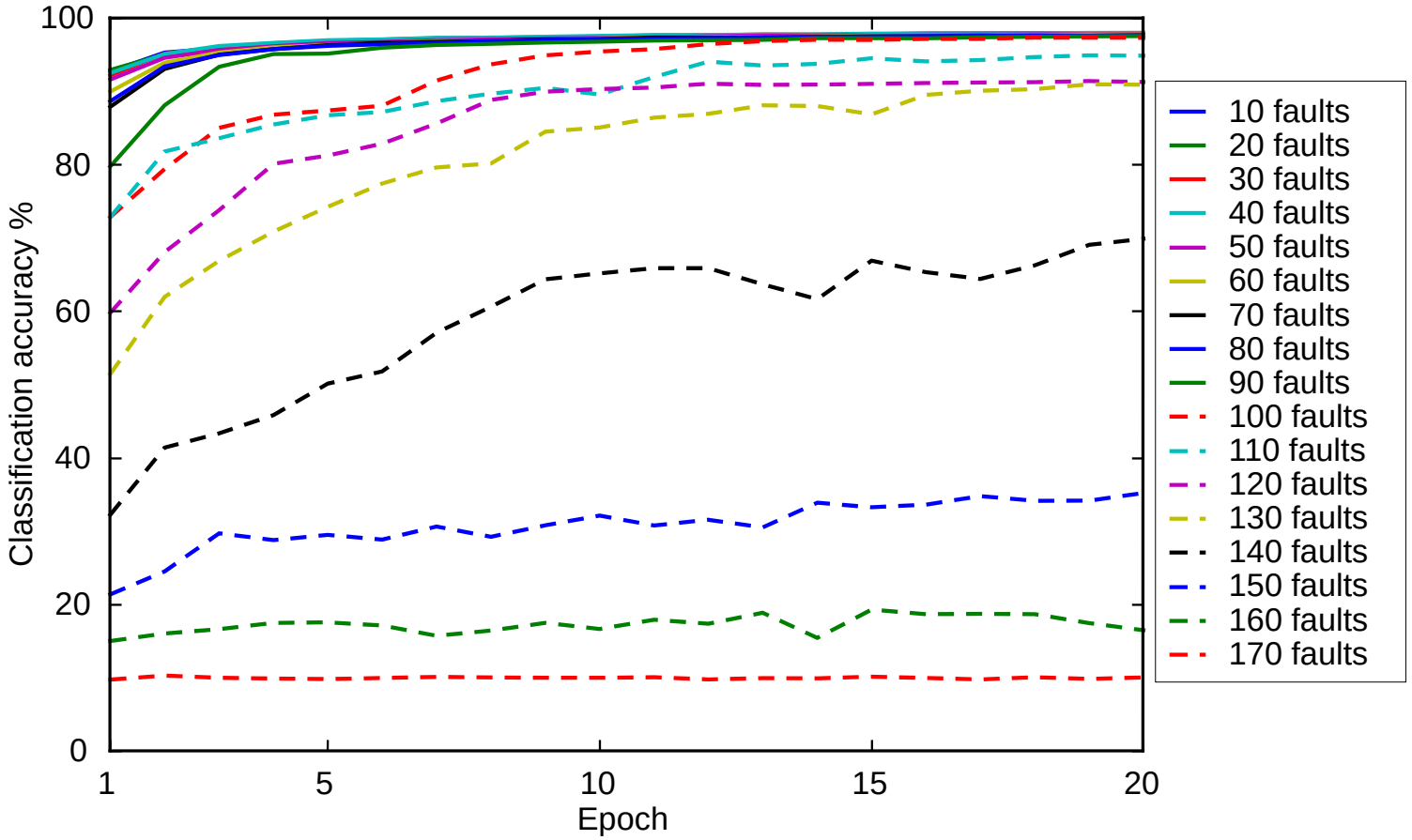








Faults occurring before training



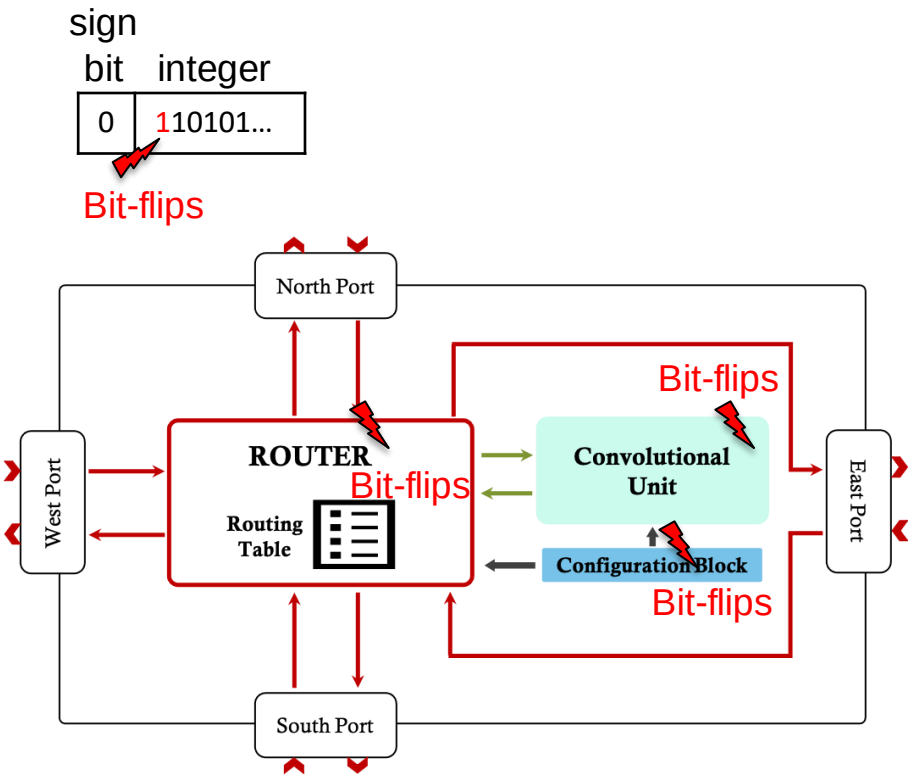
- A high fault rate can be compensated if faults occur before the training

Reliability analysis of our SNN hardware accelerator

T. Spyrou, *DATE'22*

- Each node is configurable through a set of 8-bit parameters
- Parameters are stored in memory blocks inside the node:

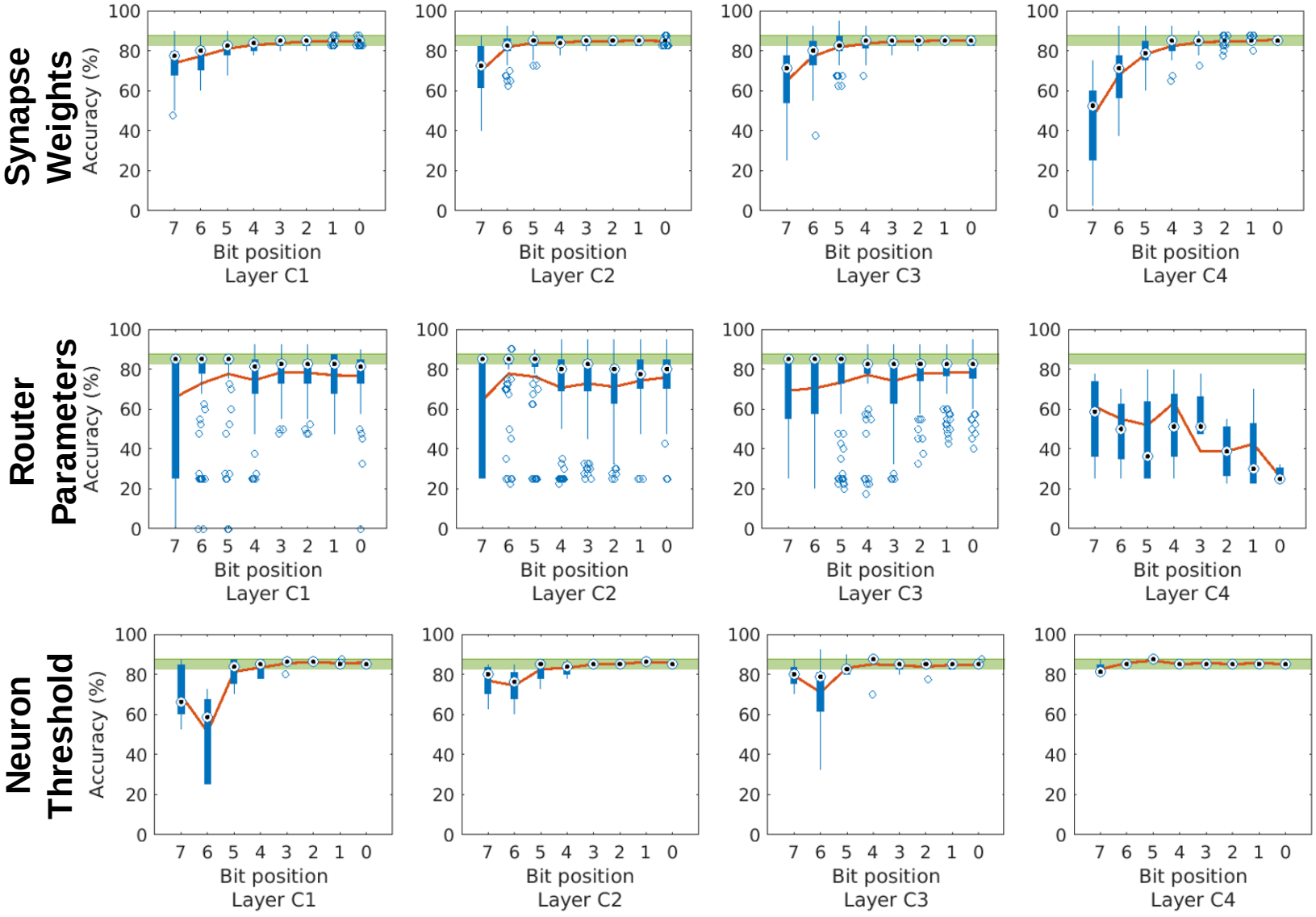
Memory	Purpose
Splitter Parameters	input split information to first layer
Router Parameters	routing information in the nodes' mesh
Neuron Parameters	key features of the neurons within the node
Kernel Parameters	kernels structural characteristics
Synapse Weights	values of the synaptic weights



- Fault model: bit-flips in memories
  - Single bit-flips across different bit positions
  - Multiple bit-flips with a BER probability

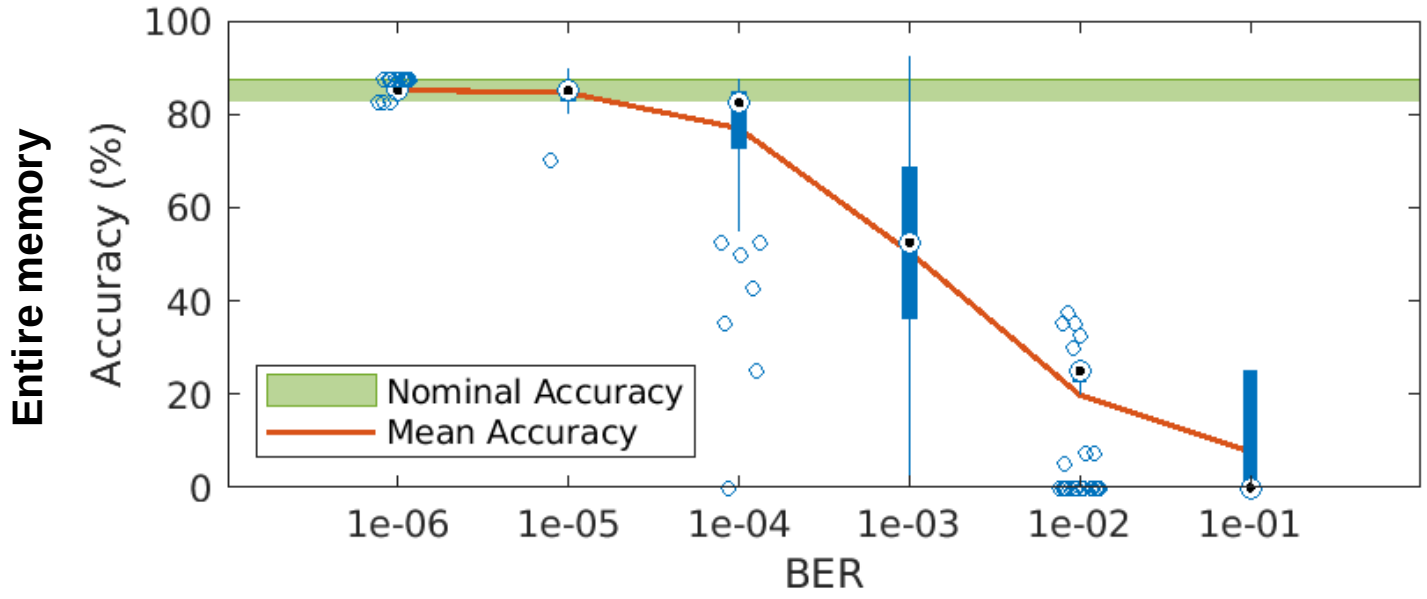
Reliability analysis results

T. Spyrou, *DATE*'22

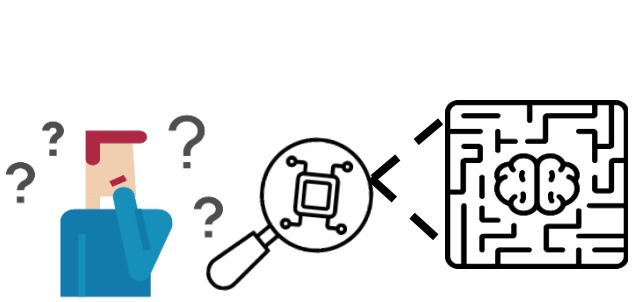


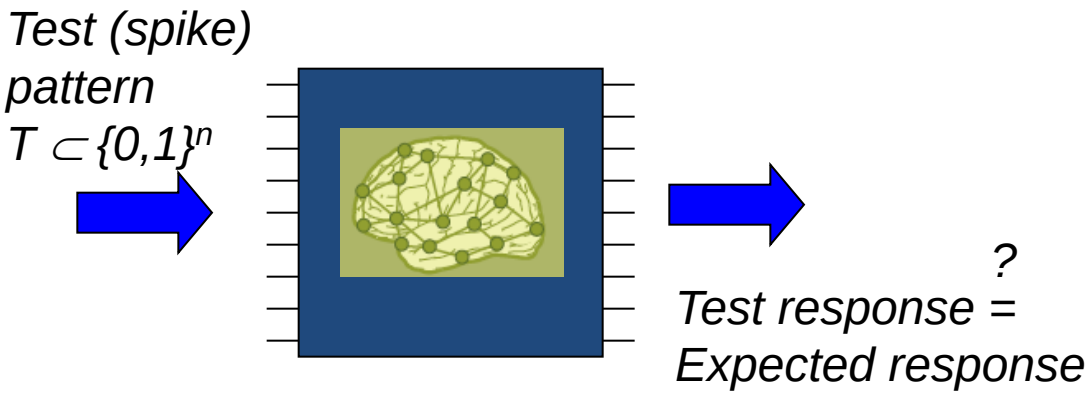
Reliability analysis results (cont'd)

T. Spyrou, *DATE*'22



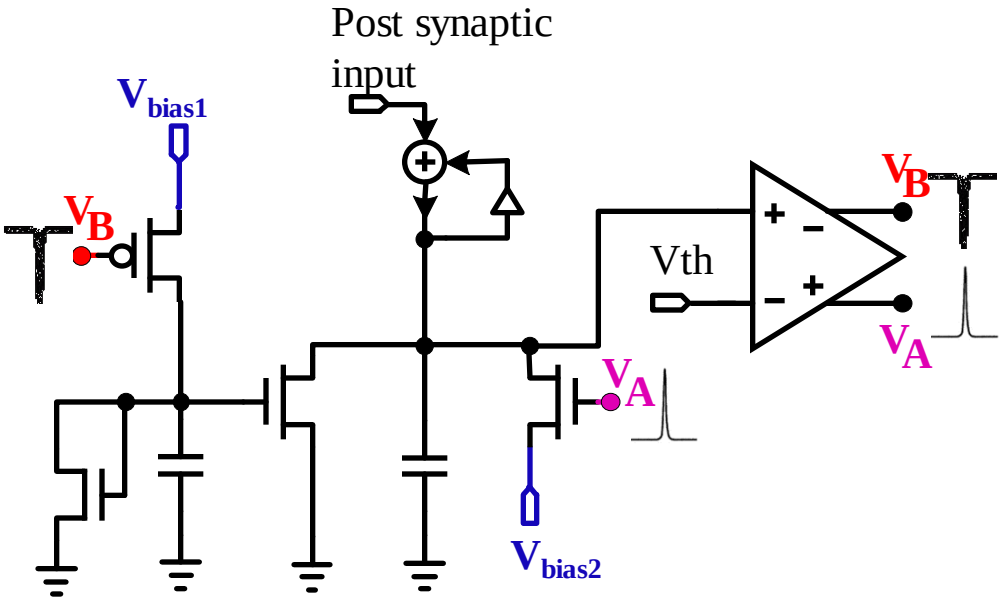
Testing neuromorphic processors

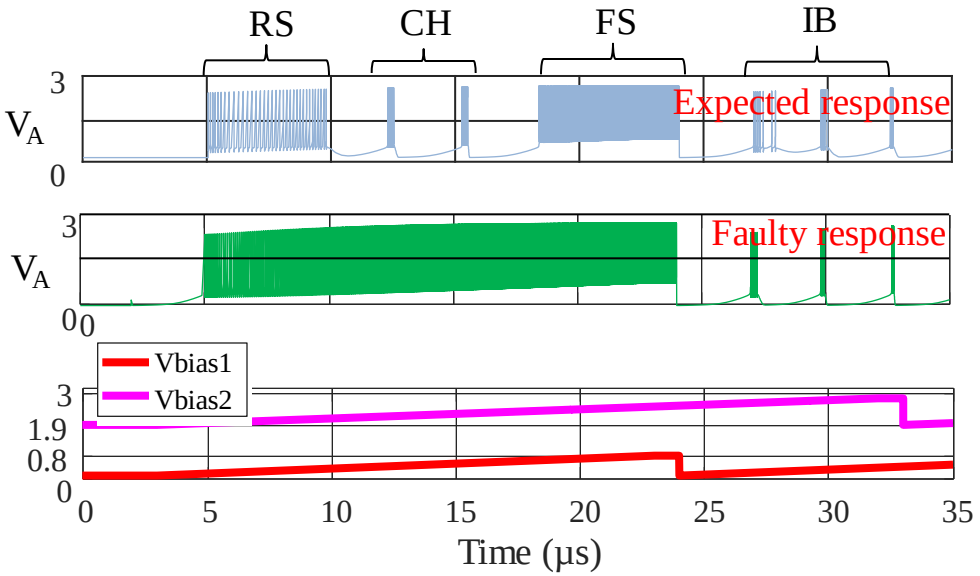




- "Business as usual"?
- Unique architectural features of neuromorphic processors make standard test methods less efficient
  - Small multiple identical cores -> large DfT overheads
  - IMC and memory-hungry - > MBIST shows large PPA penalties
- New and emerging fault models (i.e., IMC)
- Analog, mixed analog-digital
- Not all faults are equal, approximate computing
- Programmability

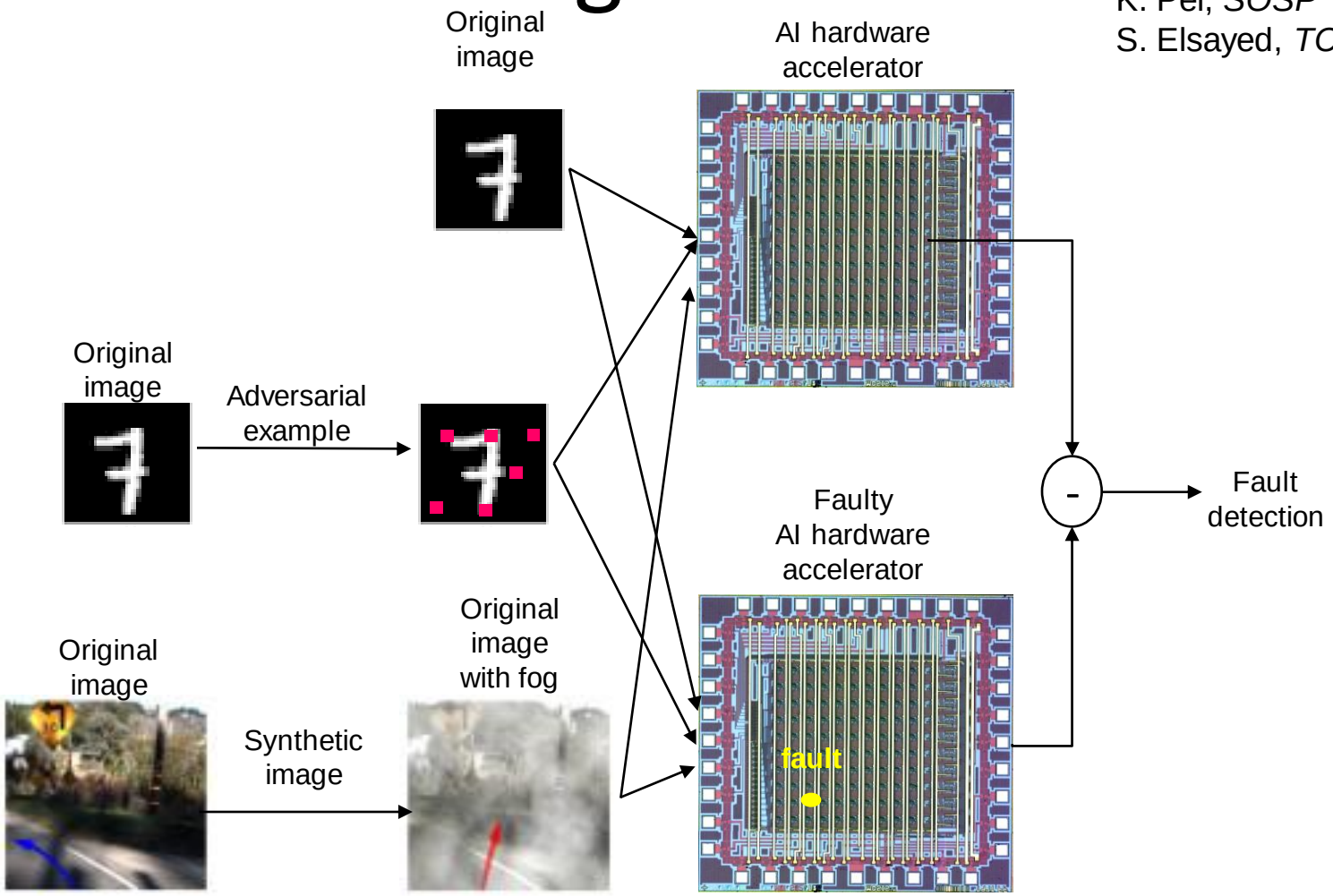
Testing of a biological neuron





Functional testing

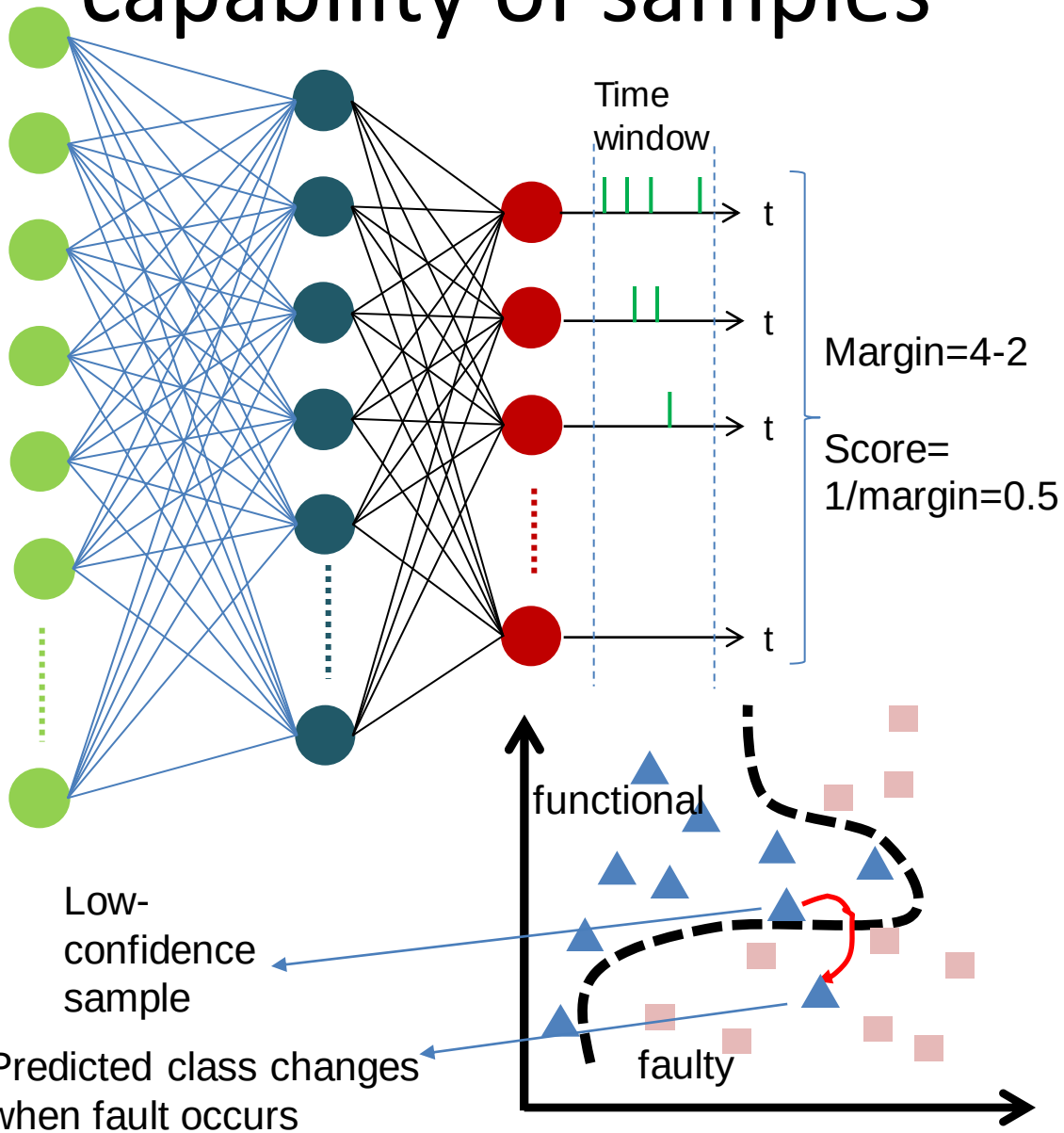
S. Kundu, *TVLSI*'21
H.-Y. Tseng, *ICCAD*'21
Y. Tian, *ICSE*'18
K. Pei, *SOSP*'17
S. Elsayed, *TCAD*'23



- Use existing samples in training/testing sets or craft new samples that can detect faults
- Fault is detected if responses of nominal/faulty chips differ

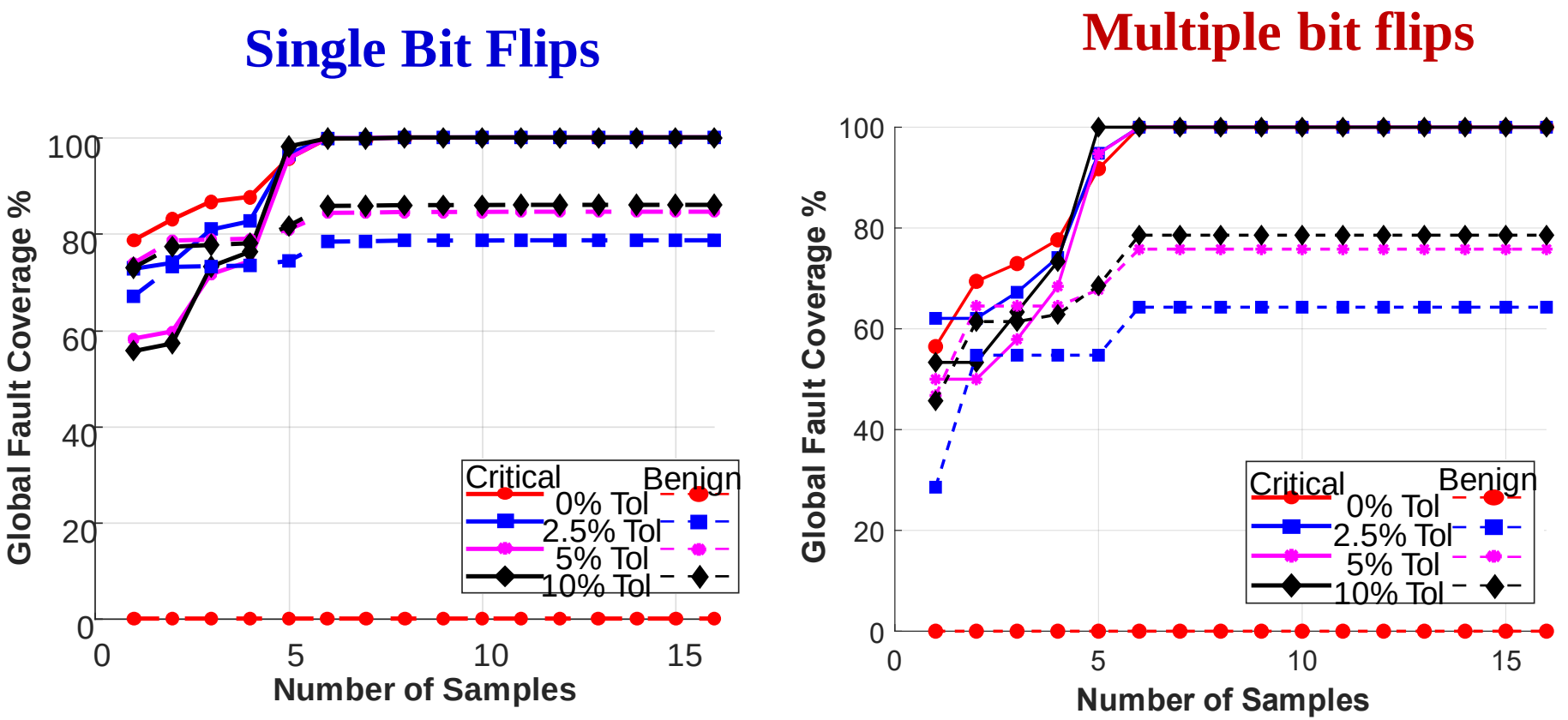
ATPG based on ranking fault detection capability of samples

S. Elsayed, *TCAD*'23



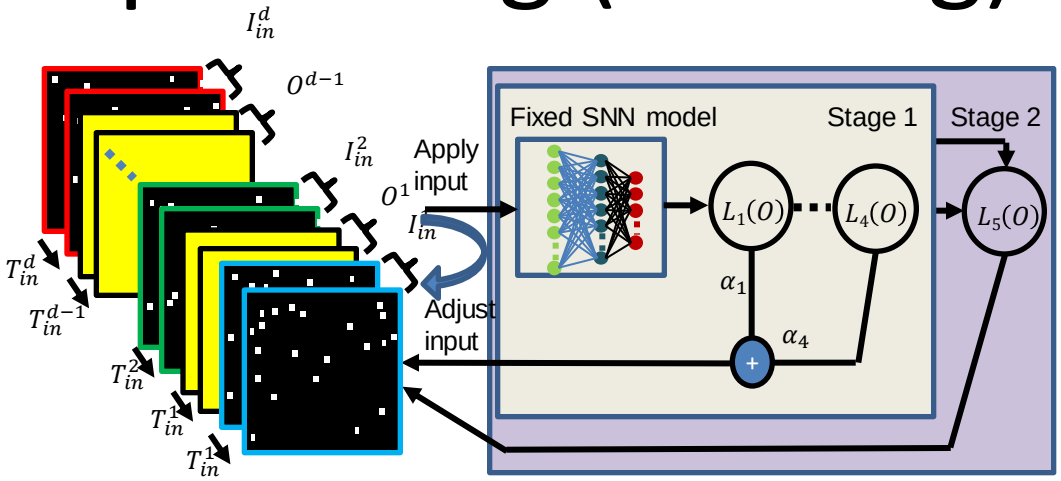
- Assess the fault coverage of an input sample <u>with no fault simulation</u>
- Fault coverage ∝ prediction confidence
- Proposed criterion: difference in output spikes between top-1 and top-2 classes
- Rank samples based on confidence in ascending order
- Add samples in the test-set according to ranking until fault coverage maximizes

Results on SNN hardware accelerator

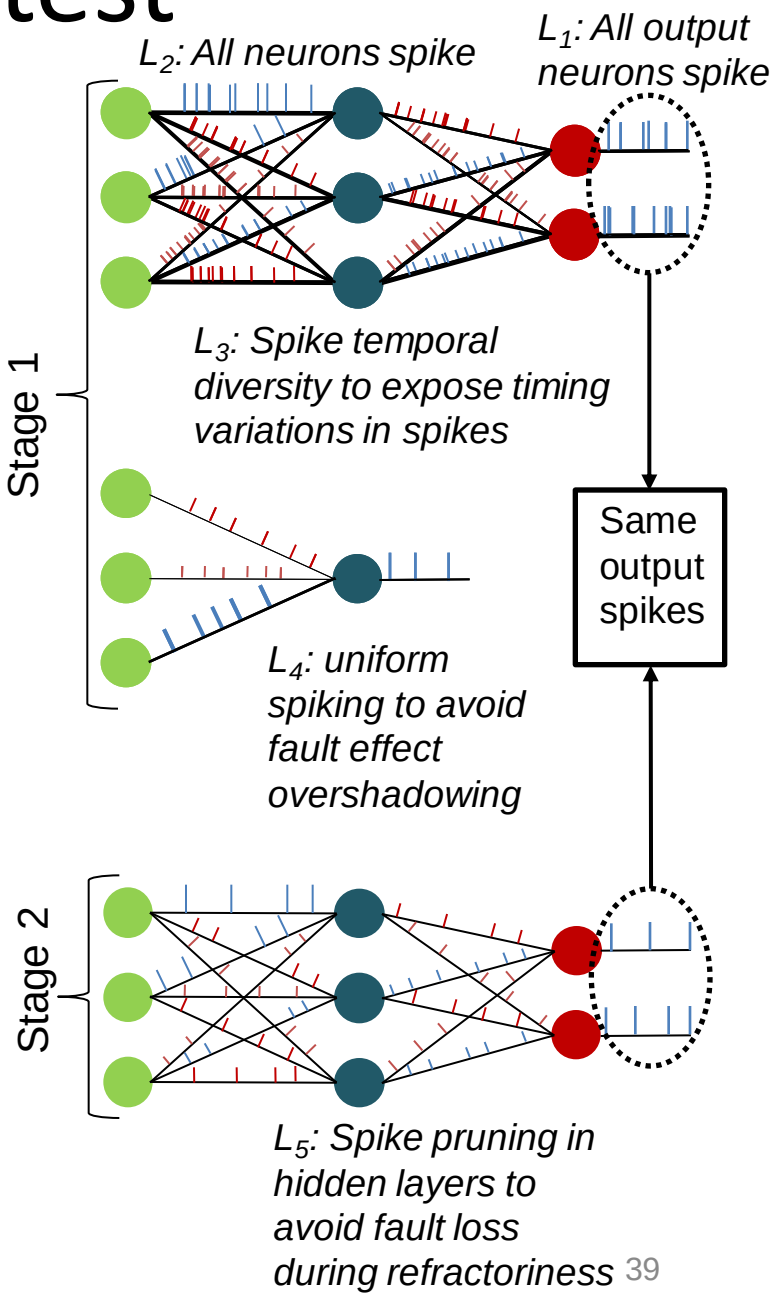


- The global cumulative fault coverage curves quickly reach 100%
- 6 samples suffice to detect all critical faults and a high percentage of benign faults

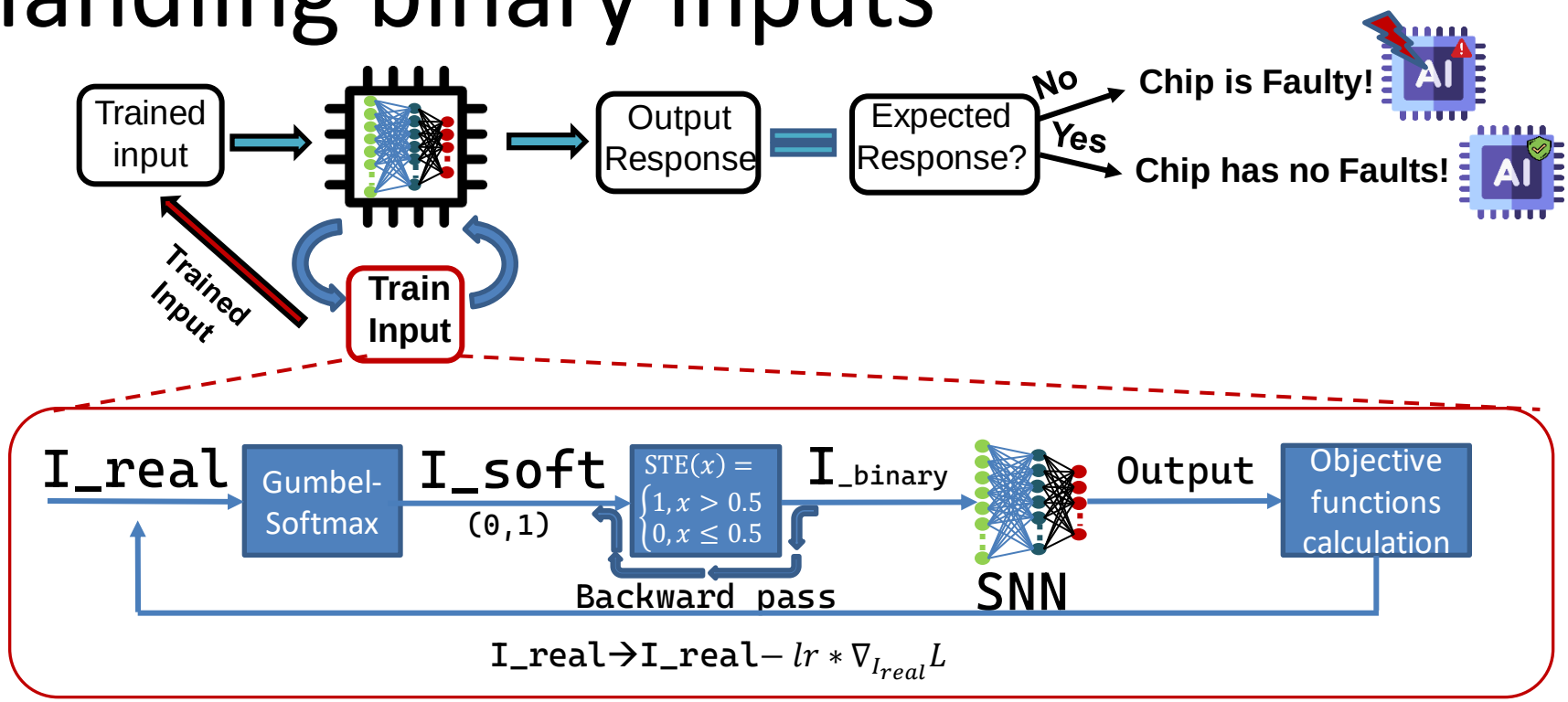
Optimizing (training) a test



- Train test input for maximum fault sensitization and fault effect propagation
- Fault simulation is circumvented using objective functions defined on spike trains
- 2- stage optimization algorithm
- Each iteration generates a short optimized test input I_{in}^{j}
- Test inputs are interleaved with zero inputs 0^{j} to reset the SNN
- Final test input is the concatenation of the short test inputs



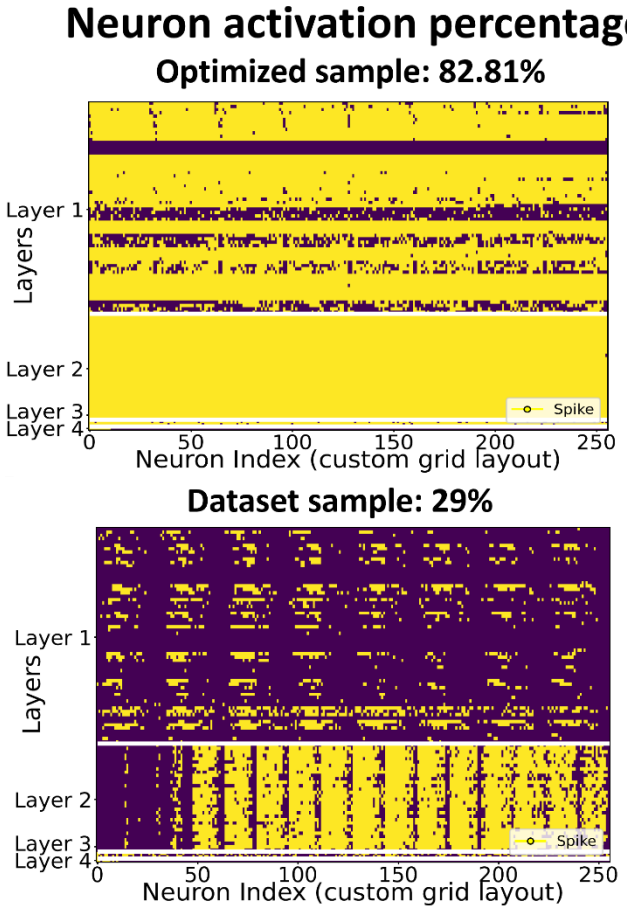
Handling binary inputs



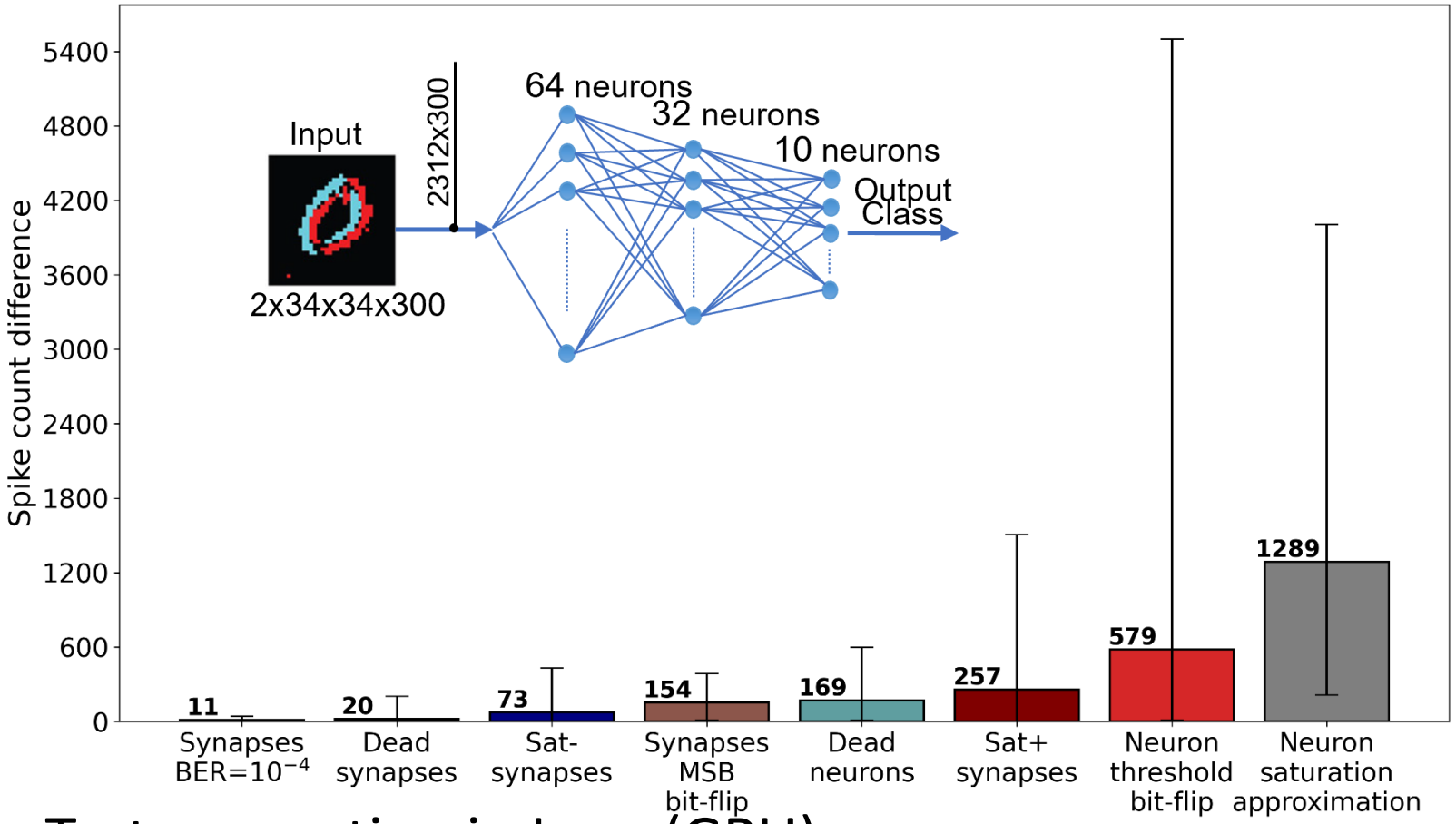
- Real-valued input as a starting point
- **Gumbel-Softmax**: Soft binarization to approximate spikes
- **STE (Straight-Through-Estimator)**: Binarizes input in the forward pass. Works as an identity function during the backward pass, allowing the gradients to flow
- Backpropagation using surrogate gradients w.r.t. membrane potential of neuron

Results on SNN benchmarks

Metric	N-MNIST	IBM	SHD
Test generation runtime	1.5 h	2.5 h	2 h
Test duration (samples)	~ 8.76	~ 11.48	~ 7.82
Test duration (time)	4.96 sec	31.86 sec	14.64 sec
Activated neurons	98.71%	82.81%	91.33%
FC Critical neuron faults	99.97%	99.86%	98.99%
FC Critical synapse faults	96.96%	99.42%	97.25%
FC Benign neuron faults	47.26%	82.29%	21.43%
FC Benign synapse faults	78.02%	58.98%	54.40%
Maximum accuracy drop for undetected critical neuron (synapse) faults	0.1%(1.1%)	0.4%(0.9%)	0.3%(1.5%)

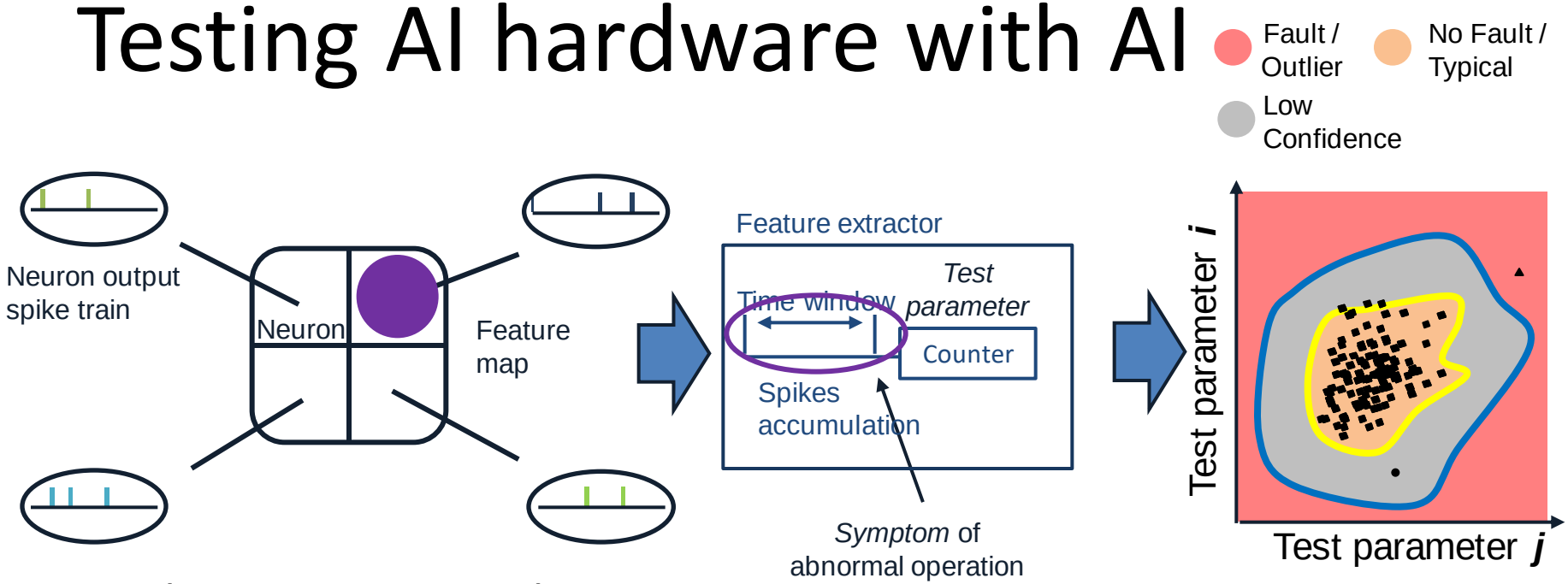


Demonstration on Intel's Loihi



- Test generation in Lava (GPU)
- Test coverage computed on Loihi by emulating different fault scenarios
- 100% critical fault coverage

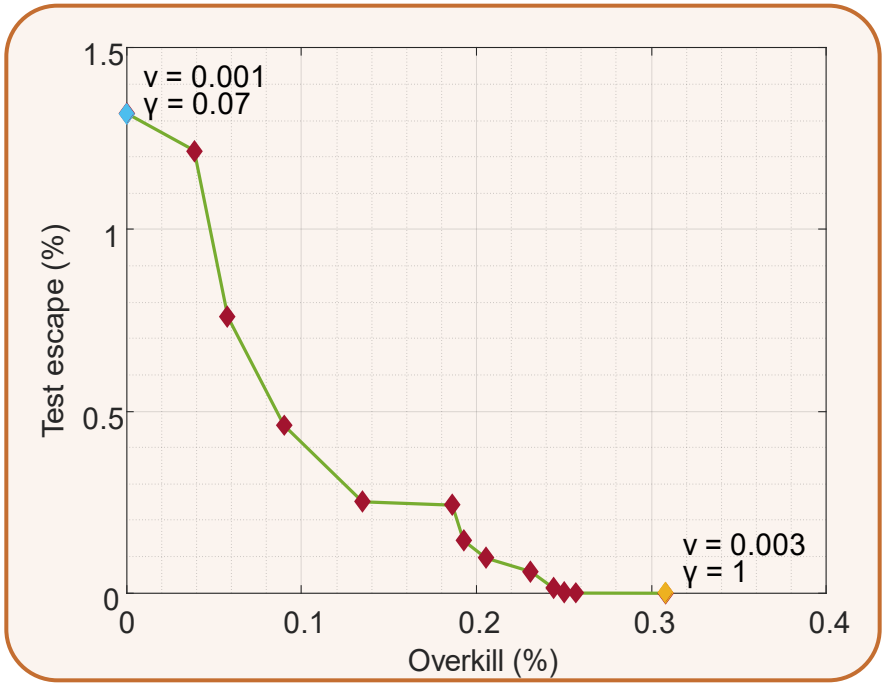
Testing AI hardware with AI

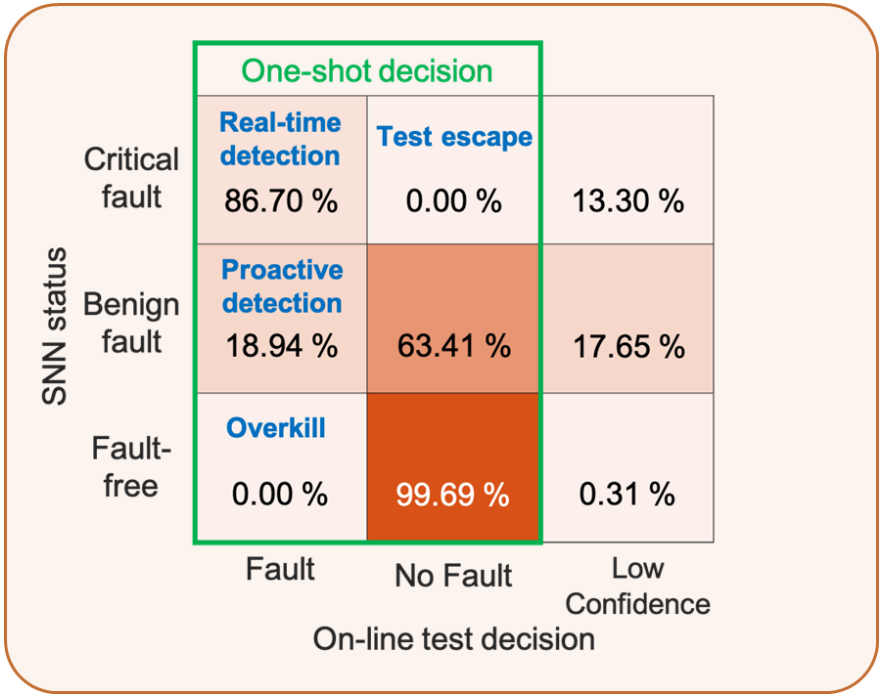


- On-line test approach
- Symptom detection
- Test parameter is the cumulative spike count at feature map output
- Use a system of two one-class classifiers for mapping test parameters to a decision
- One-shot decision (fault or no fault) with high confidence
- If low-confidence execute a reply operation to resolve ambiguity

Validation on our SNN hardware accelerator

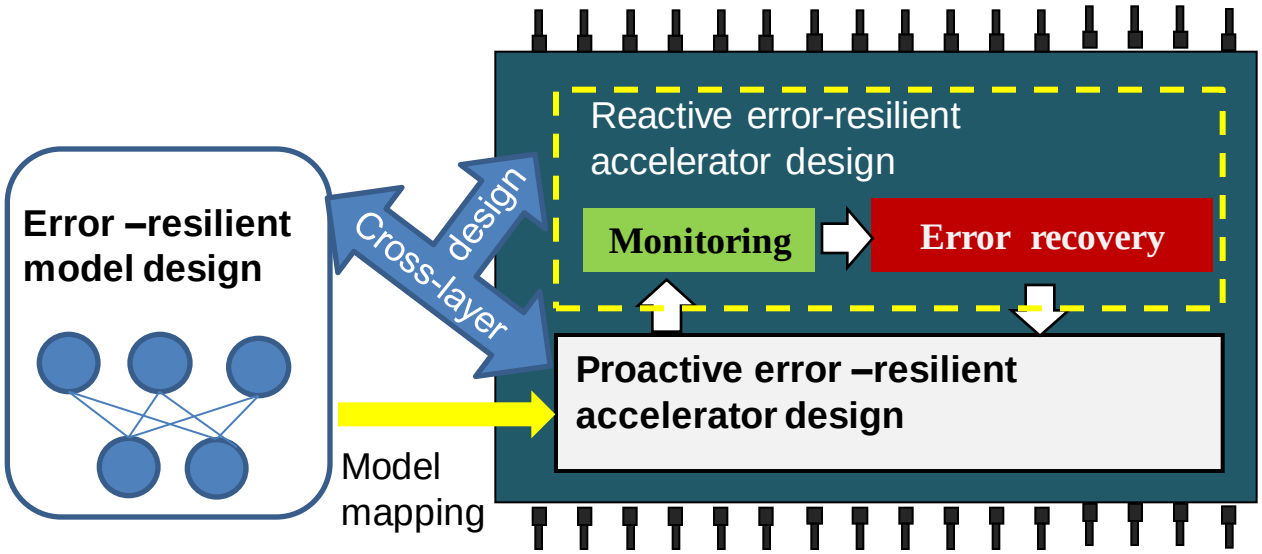
T. Spyrou, *ETS*'23





- One-class classifiers with SVMs running on the processor of the board
- One-shot decisions have 0% test escape and overkill, 86.7% of critical faults detected in real time, 18.94% of benign faults detected proactively
- One reply operation resolves all test ambiguities

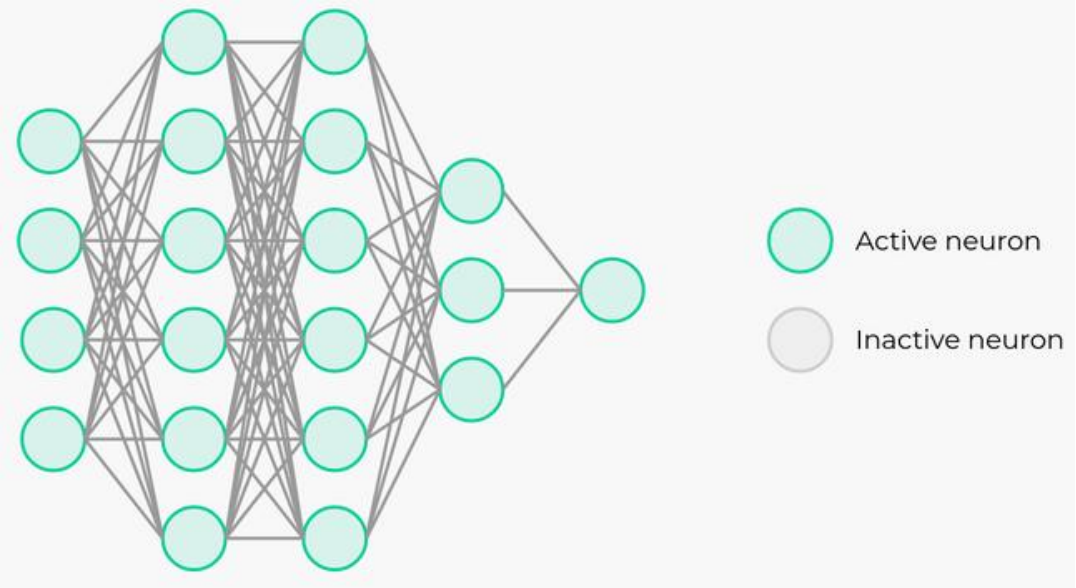
Fault tolerance



- Train AI models with inherit error resilience
- Re-design for proactive error resilience
- Re-design with reactive error resilience
- Cross-layer error-resilience approaches

Training with faults: dropout

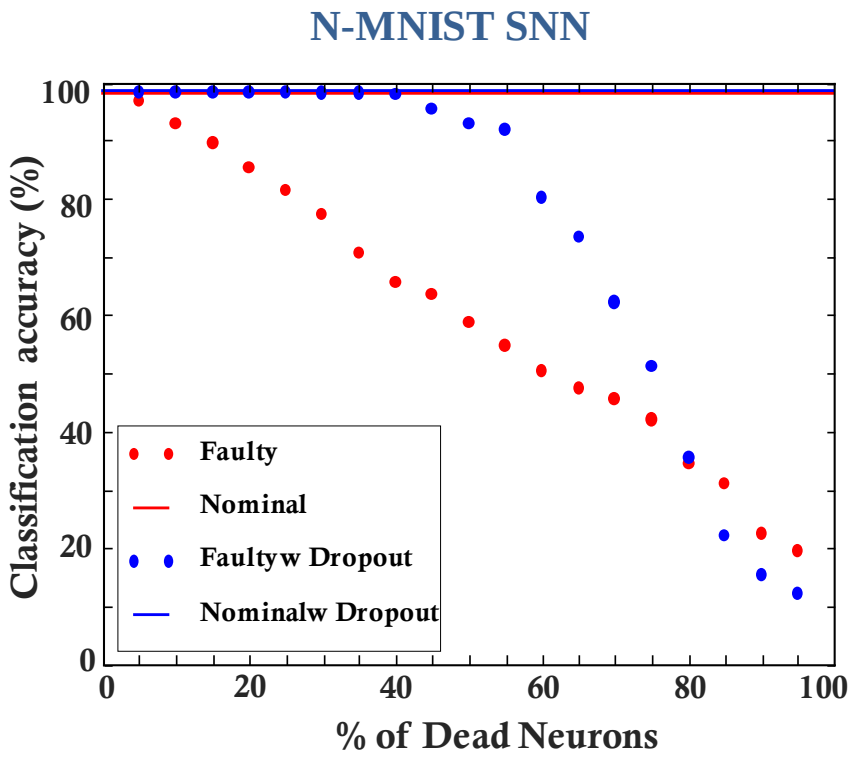
N. Srivastava, *JMLR*'14
T. Spyrou, *DATE*'21

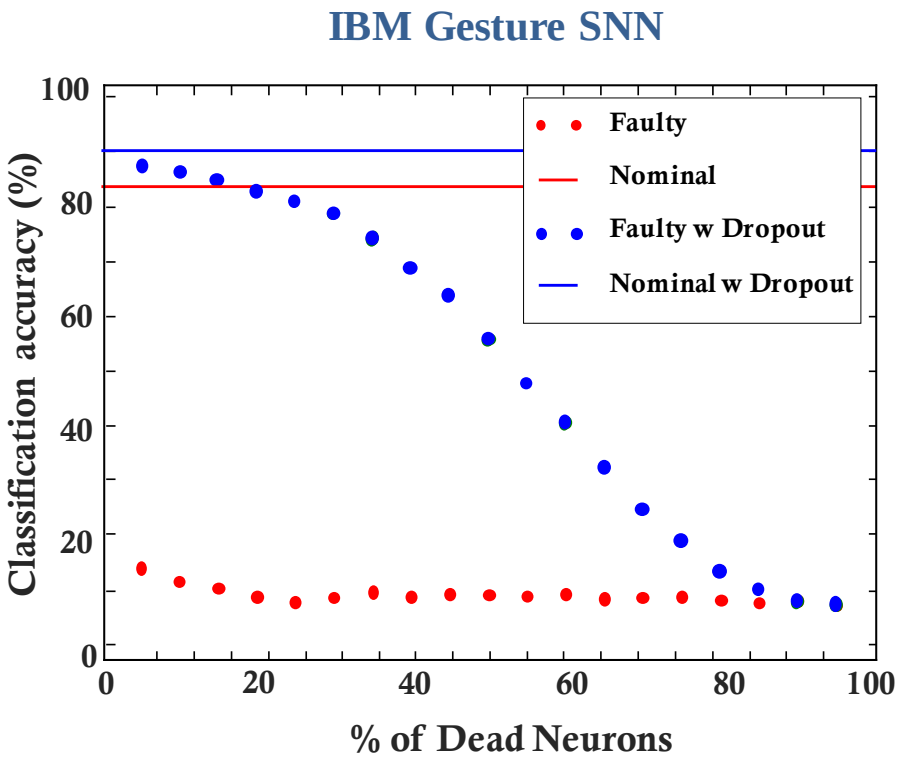


- Training with dropout: temporarily removing neurons during training along with their connections
- Nullifies the effect of dead neuron faults in all hidden layers:
  - Distribution of computational load among the neurons of the network
  - More uniform and sparse spiking activity across the network

Multiple fault scenario with dropout

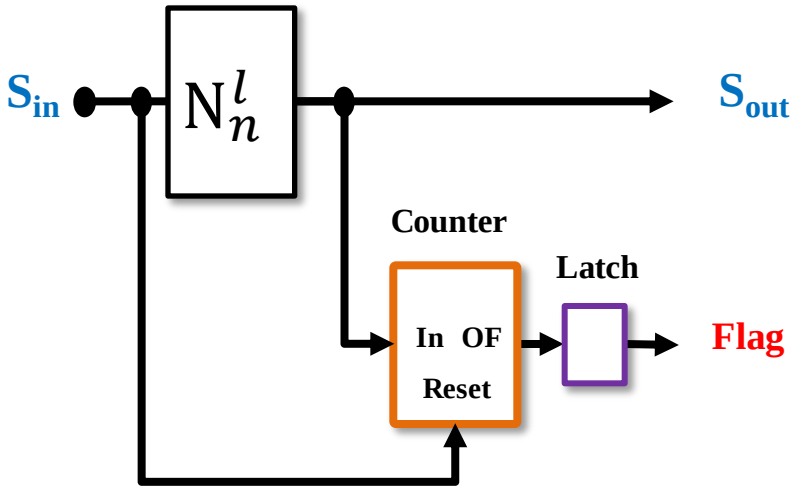
T. Spyrou, *DATE*'21

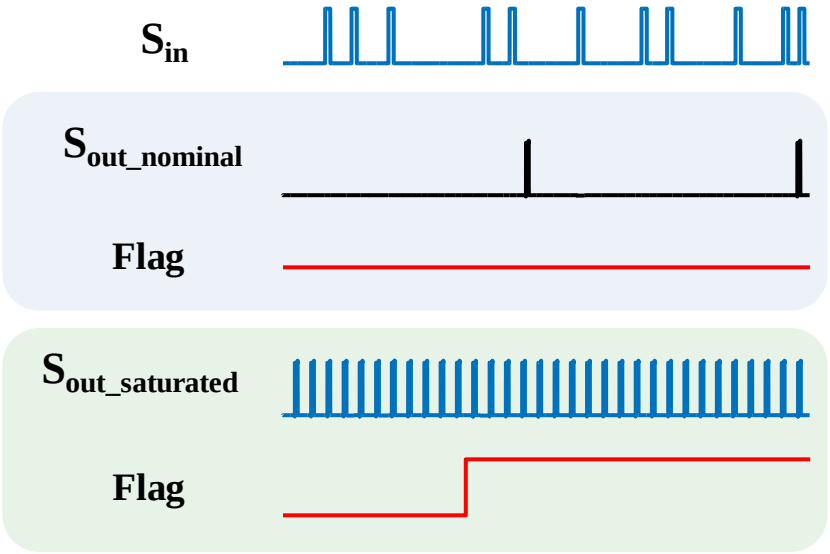




On-line testing using in-situ monitors

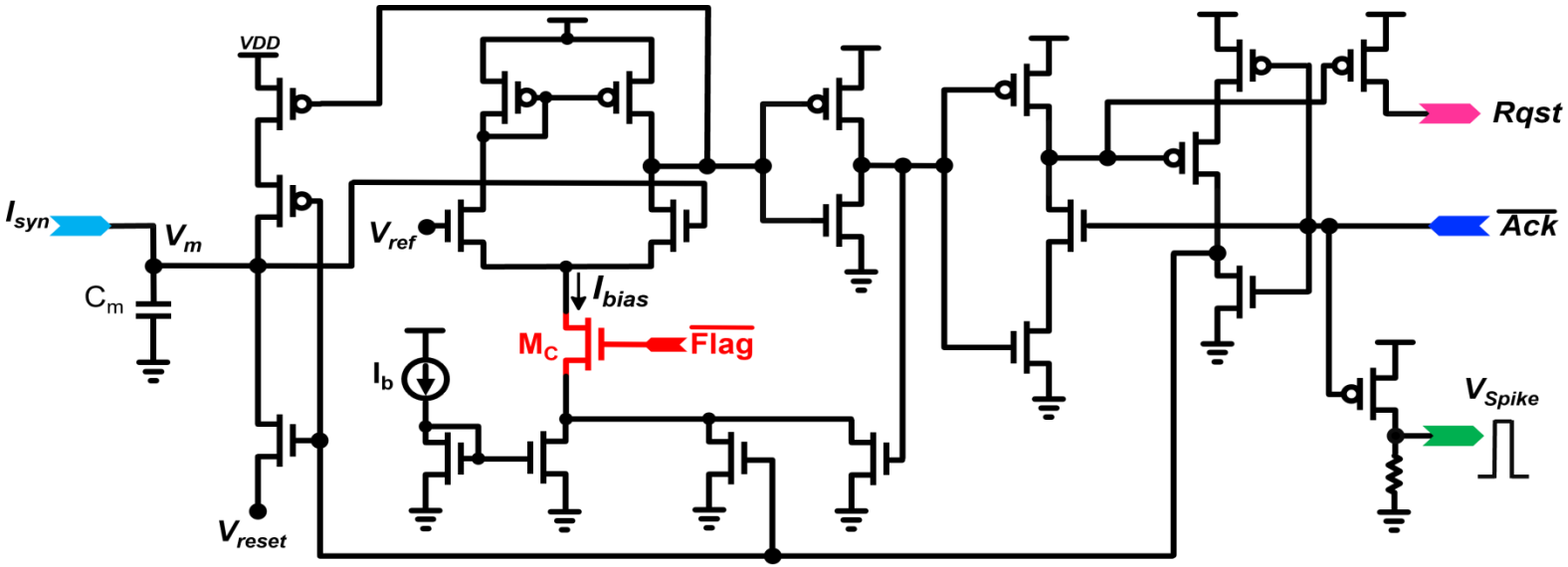
- Count the number of spikes a neuron produces between two successive inputs
- A saturated neuron will produce spikes with higher frequency than usual: counter overflows before an incoming spike resets it again
- Exploits temporal dependency between the input and output of a spiking neuron





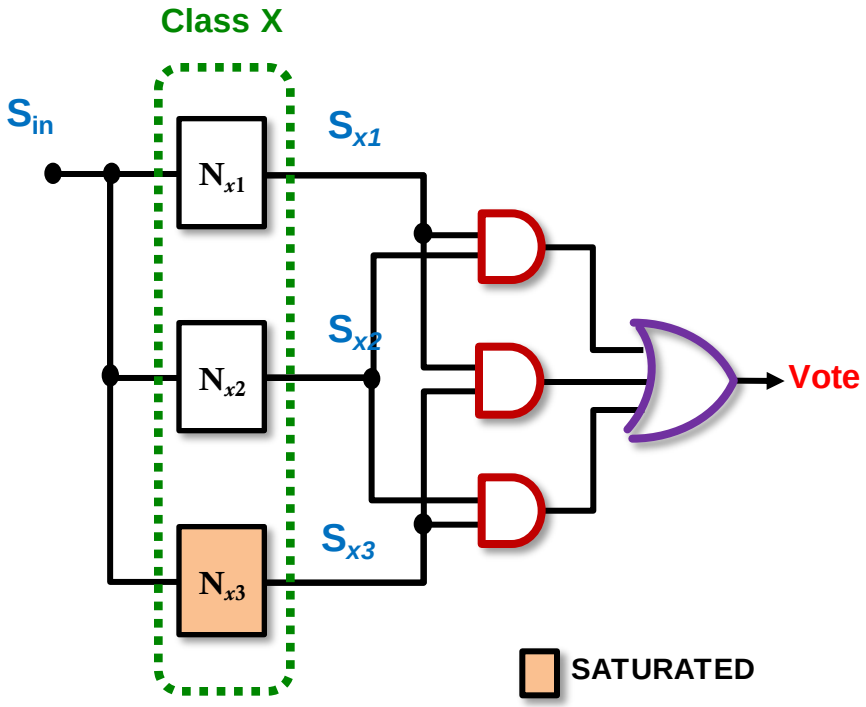
Error recovery using fault masking

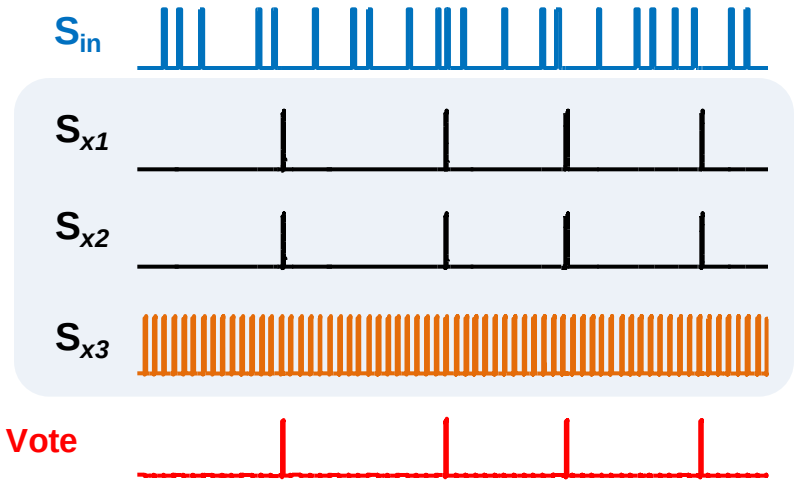
- Saturated neurons are more critical than dead neurons & dead neurons can be nullified using dropout
- "Fault Hopping" concept: saturated neuron fault is translated to a dead neuron fault
- One single transistor is added to the neuron to switch-it off when a saturation "Flag" signal is raised
- Dead neurons do not consume energy



Redundancy-based fault tolerance

- Triple Modular Redundancy:
  - 3 identical neurons vote for the decision of each class
  - majority decides
- Output layer is usually smaller in size than whole network (0.57% for the N-MNIST SNN and 0.04% for the IBM's Gesture SNN)
- Area overhead is negligible





Conclusions

- Advancing the testability and reliability methodologies will ensure that next-generation neuromorphic systems are not only high-performance, but also dependable, verifiable, and resilient in real-world deployments
- Acknowledgments:
  - PhD Students: Sarah Elsayed, Theofilos Spyrou, Spyridon Raptis, Paul Kling, Christos Malogiannis
  - Sorbonne Center for Artificial Intelligence (SCAI)
  - ANR RE-TRUSTING
  - Horizon Europe dAIEDGE
  - EDF ARCHYTAS
  - ANR-NSF CHAMELEON
  - Chist-Era TruBrain

Questions?

haralampos.stratigopoulos@lip6.fr