Abisko: Deep Codesign of an Energy-Optimized, High Performance Neuromorphic Accelerator

Jeffrey S. Vetter, ORNL (PI)
Alec Talin, Sandia National Laboratories
David Brooks, Harvard University
Yu Cao, Arizona State
Sung Kyu Lim, Georgia Tech

Salishan Conference
29 Apr 2022

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

ASCR Program Manager: Robinson Pino

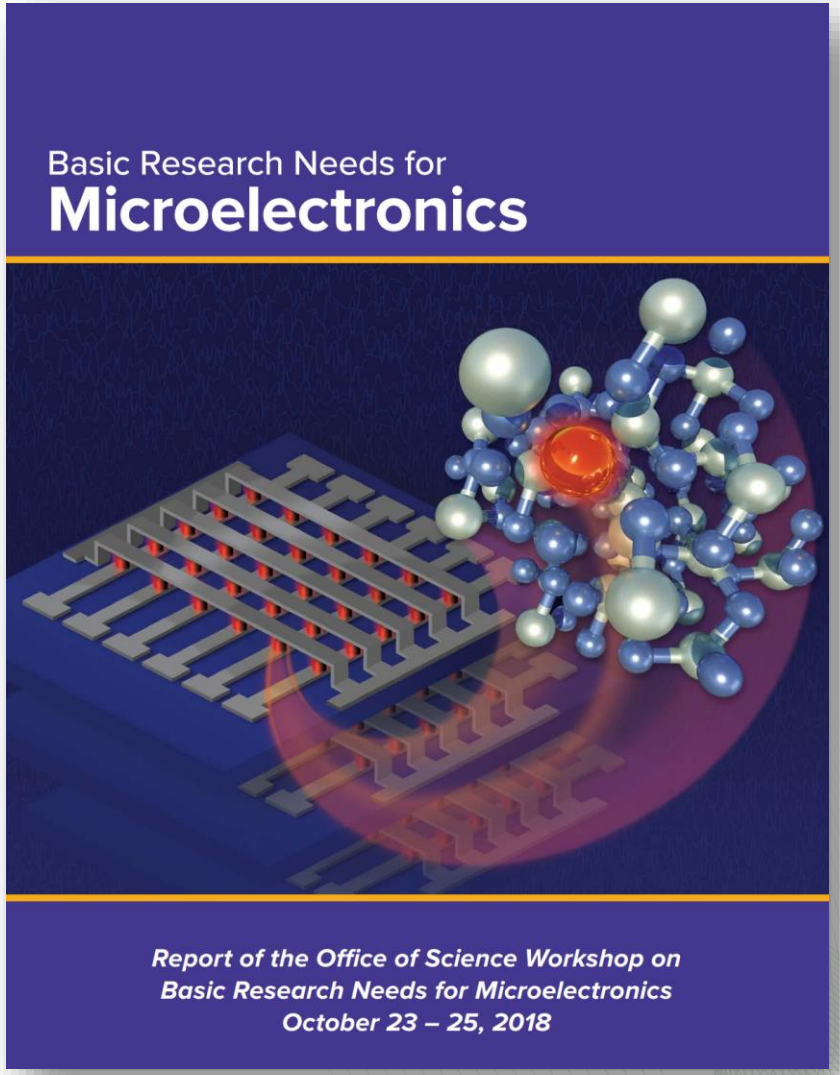
Overview

- Many factors are driving improved design of future computer systems
  - Electronics scaling, power, **business models**, etc.
  - Massive demand for next-generation HPC systems (e.g., ModSim, AI, Data, Omniverse)
  - Microelectronics is recognized as a critical factor in economic wellness and national security
- **Domain-specific computing** (extreme heterogeneity) is a highly likely outcome
- DOE has championed **codesign in HPC** for at least a decade
  - Enable integrated design and implementation of end-to-end solutions, then iterate!
  - Domain-specific computing needs codesign
- **Abisko** is a new microelectronics codesign project with the ambitious goals
  - Develop better techniques for codesign from algorithms to devices and materials
  - Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
  - Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
  - Design language abstractions and runtime support for SNN chiplet
- Abisko is an **interdisciplinary** project that includes scientists from applications, algorithms, software, architectures, devices and circuits, and materials!

https://www.osti.gov/biblio/1822198-reimagining-codesign-advanced-scientific-computing-unlocking-transformational-opportunities-future-computing-systems-science

Basic Research Needs for Microelectronics (2018 Workshop)

- Five Priority Research Directions
  1. Flip the current paradigm [codesign]
  2. Revolutionize memory and data storage
  3. Reimagine informal flow unconstrained by interconnects
  4. Redefine computing by leveraging unexploited physical phenomena
  5. Reinvent the electricity grid through new materials, devices, and architectures



https://www.osti.gov/biblio/1616249-basic-research-needs-microelectronics-report-office-science-workshop-basic-research-needs-microelectronics-october

PRD #1: Flip the Paradigm

- *"Define innovative material, device, and architecture requirements driven by applications, algorithms, and software."*

- Optimize and integrate end to end solutions across multiple levels of abstraction for efficiency.

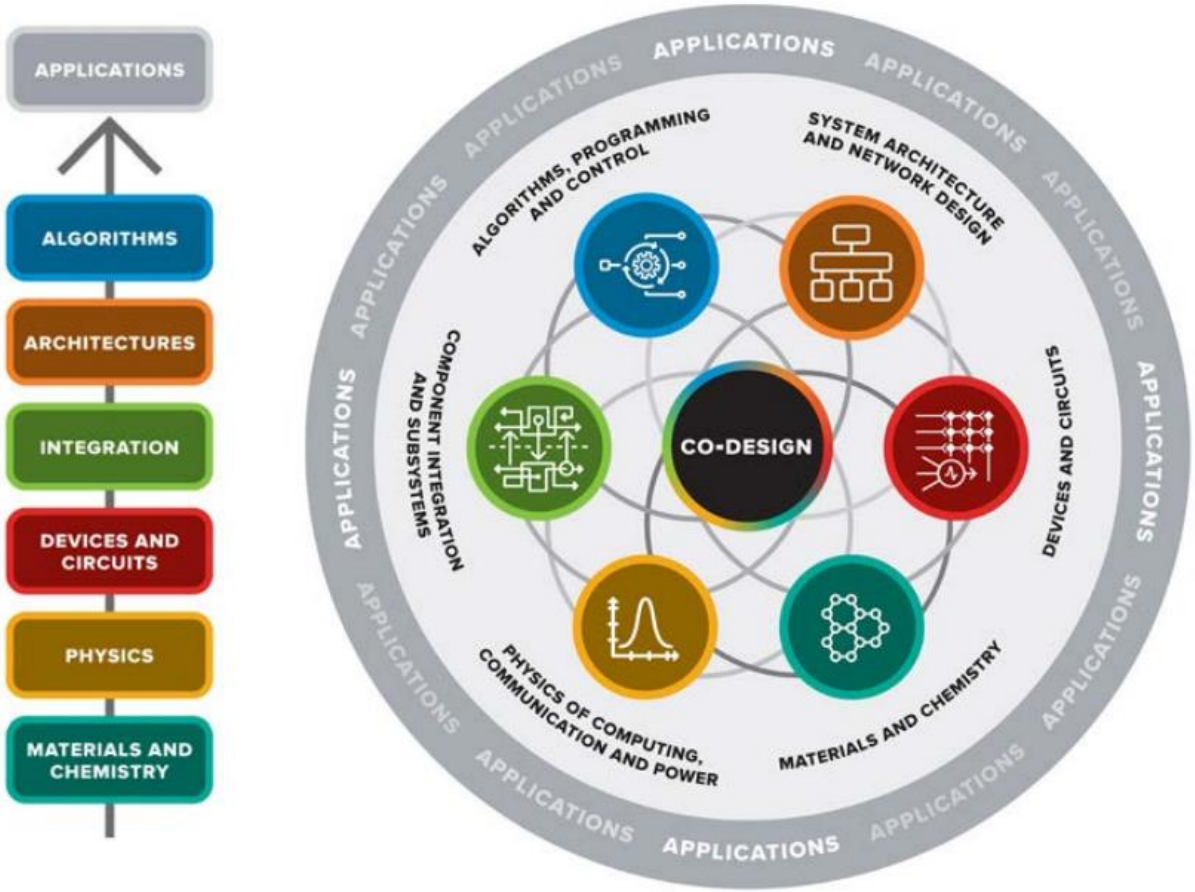
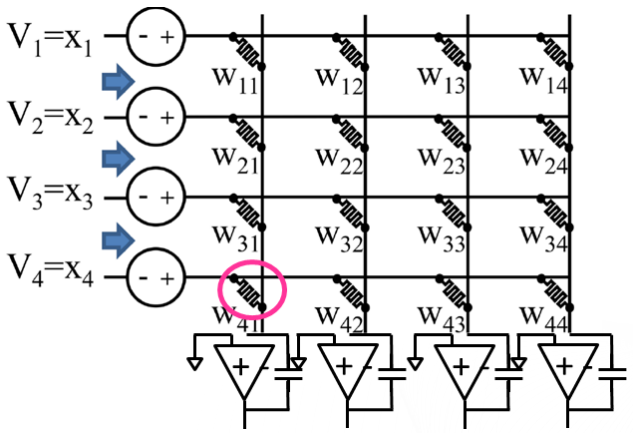


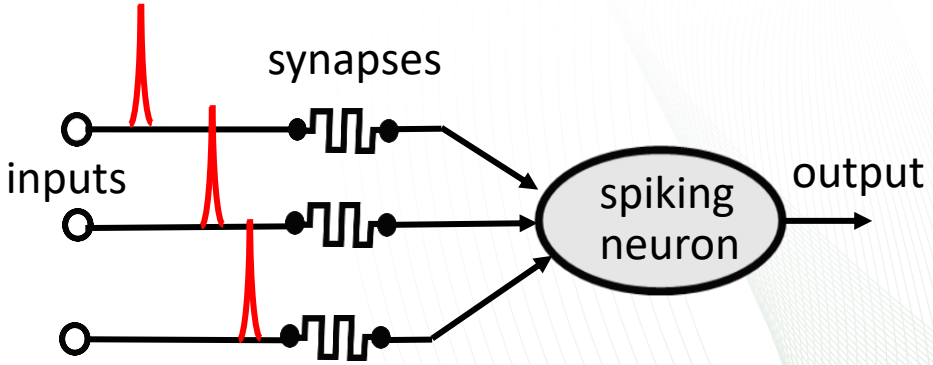
Figure 2. Co-design framework: From the traditional hierarchy of abstraction levels (left) to a holistic system framework (right)

https://www.osti.gov/biblio/1616249-basic-research-needs-microelectronics-report-office-science-workshop-basic-research-needs-microelectronics-october

PRD #4: Redefine computing by leveraging unexploited physical phenomena

- *"leverage novel physical processes to perform useful computation"*
- Categories
  - Optimization machines
  - Computational Models
  - Partitioning b/w non-von Neumann and von Neumann architectures
- Examples
  - Ising machines, Spiking neural networks, Analog MAC with Crossbar, optical FFT, …





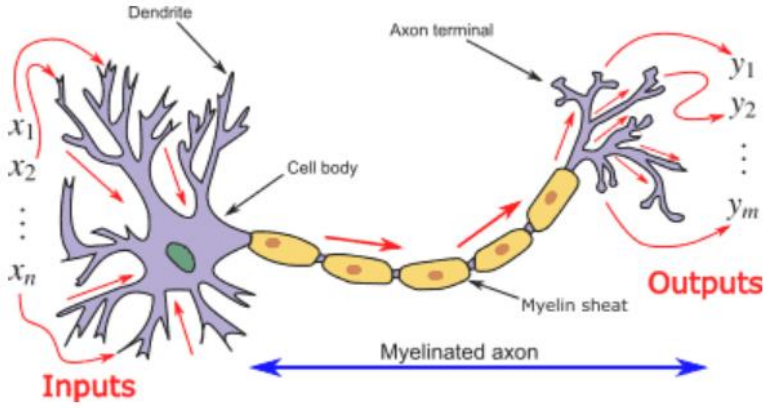
Figures courtesy of Abisko, Alec Talin

Abisko

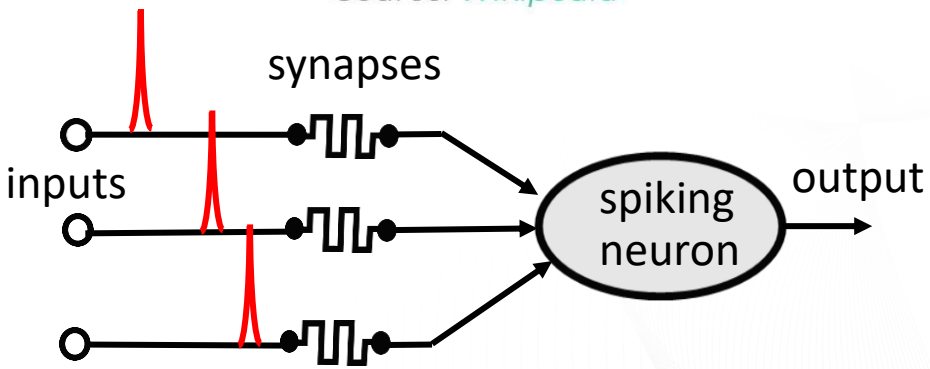
- A Microelectronics Codesign Project

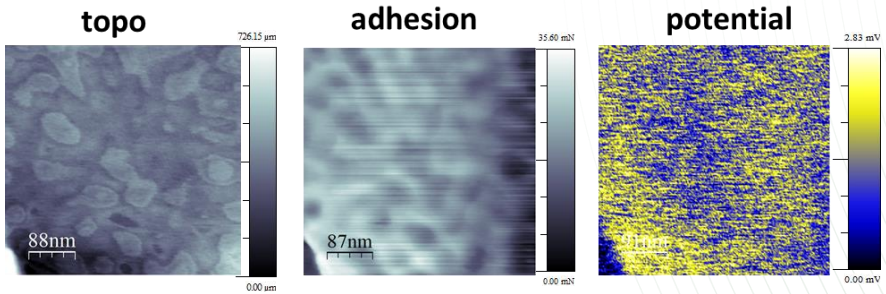
Abisko Vision

- Develop better techniques for codesign from algorithms to devices and materials
- Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
- Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
- Design language abstractions and runtime support for SNN chiplet



Source: Wikipedia



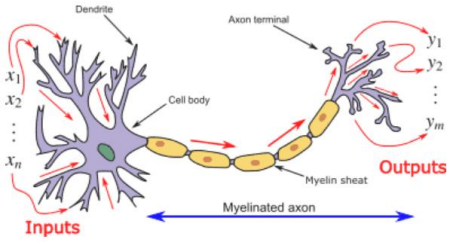


0.5*0.5 um

Abisko Microelectronics Codesign Overview

Oak Ridge National Laboratory | Sandia National Laboratories | Arizona State University | Georgia Tech | Harvard University | Collaborator: Fermilab

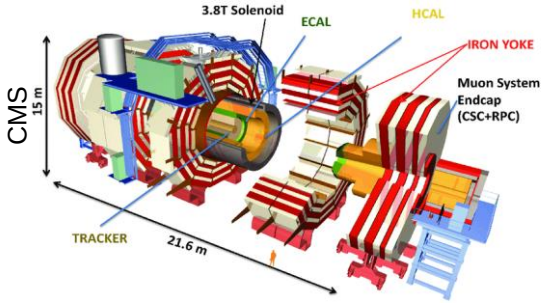
1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet



Source: Wikipedia

Applications

Motivation
- Transportation
- CMS Sensors



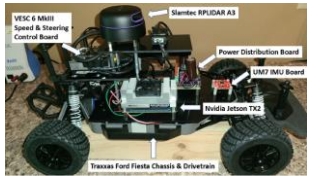
Motifs, Composition

Algorithms

Algorithms
- ML: SLAYER, Whetstone, EONS, eProp, STDP
- Non-ML: Graph algorithms, CSP
- Simulators: NEST, Brian2

nest :: | BRIAN

F1Tenth



API, Motifs

Software

Software
- DSL and API for neuromorphic co-processing
- Built on LLVM and MLIR
- Portable across Abisko chiplet, GPUs, etc.

LLVM COMPILER INFRASTRUCTURE | MLIR | XACC

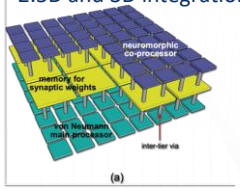
ISA, IR

Architecture

Architecture
- Design neuromorphic chiplet
- RISC-V neuromorphic extensions
- Heterogeneous integration with contemporary technologies

RISC-V

2.5D and 3D integration



Simulation/Emulation

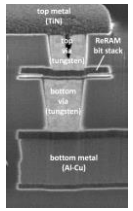
ALADDIN | gem5

Circuit scale up, Interconnects, PDK

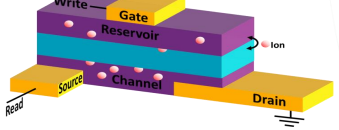
Devices and Circuits

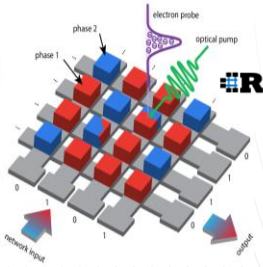
Devices and Circuits
- ion insertion (reversible doping) sets analog states
- mRaman captures transition linear, non-linear switching
- Will extend to 36x36 x-bar array
- Electronic and other optical spectroscopies

TaOx ReRAM



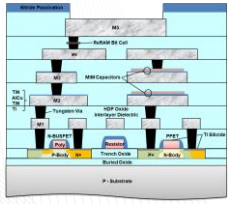
ECRAM





ROSS SIM

MesaFAB ReRAM

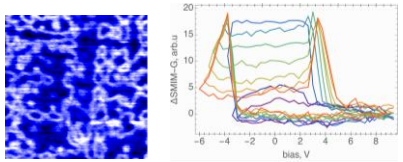


Compact models

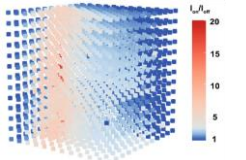
Materials

Materials
- Non-equilibrium probes to few nm
- Data-driven modeling
- On-demand neuromorphism

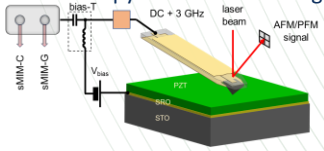
Domain wall memristor



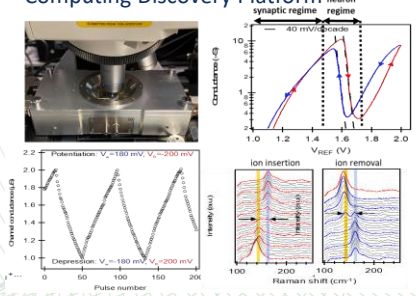
Computational data mining



CNMS scanning probe microscopy and chemical imaging

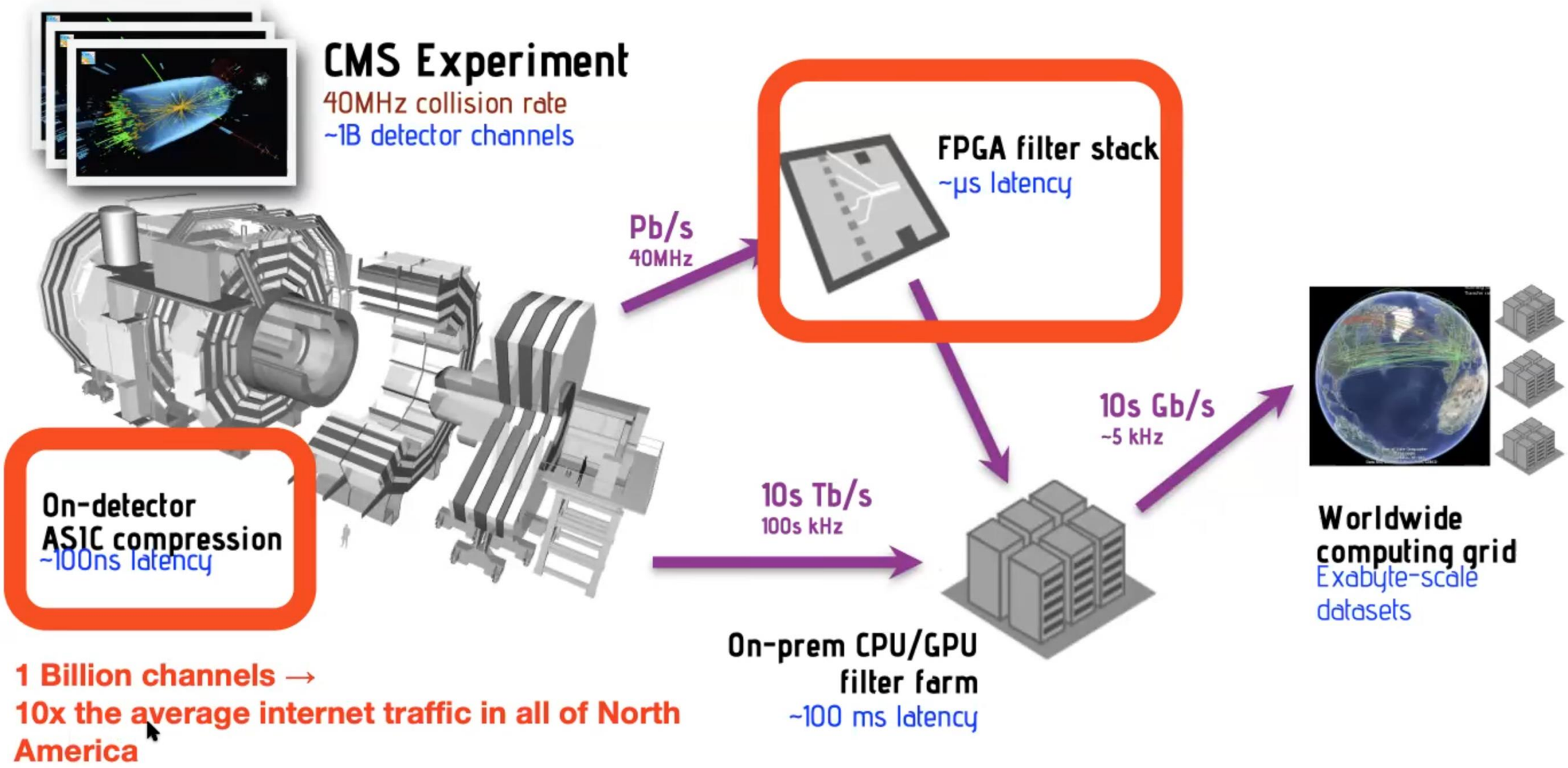


Computing Discovery Platform



Applications

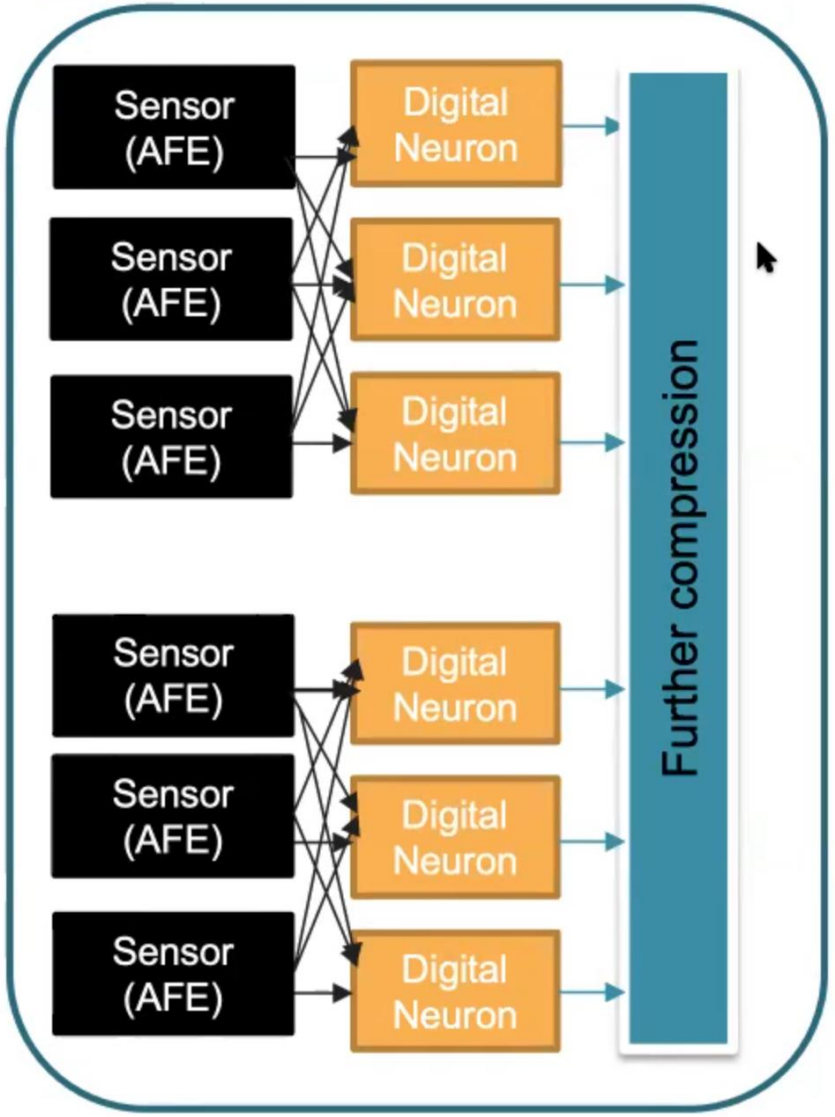
Slide courtesy of Farah Fahim (FermiLab)



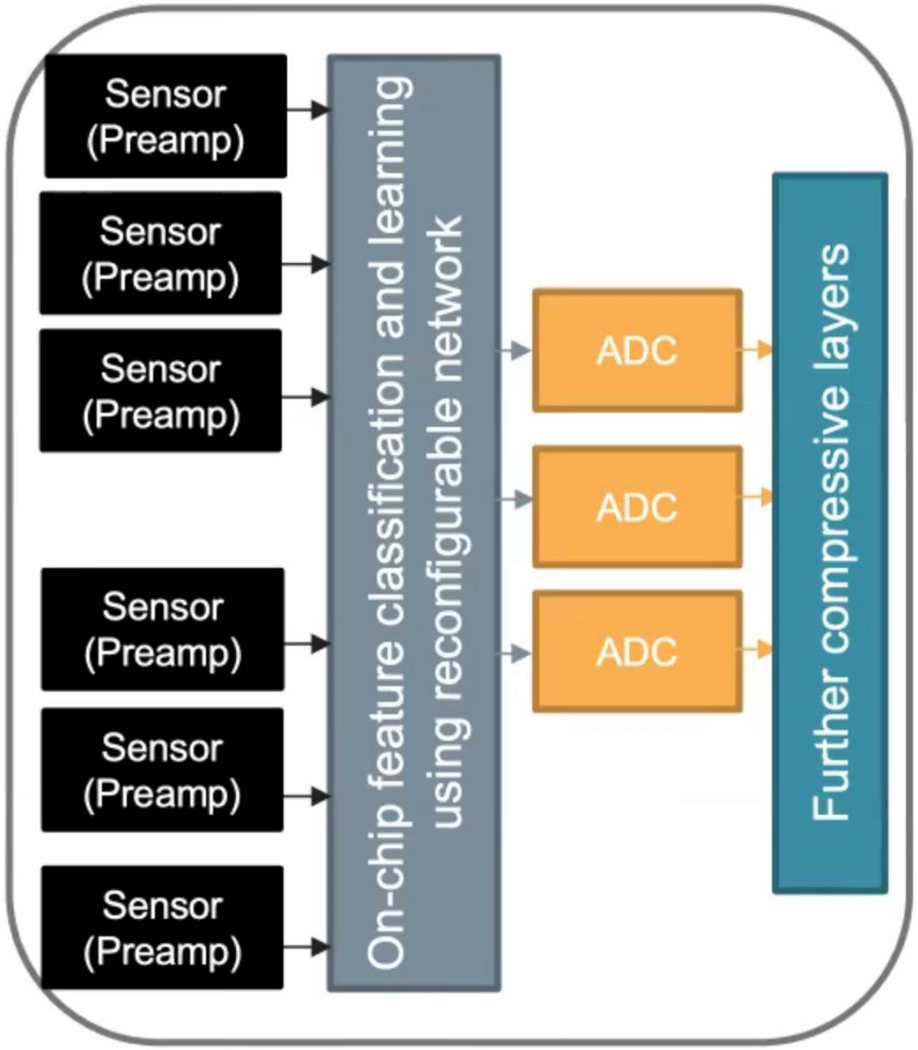
Slide courtesy of Farah Fahim (FermiLab)

Pixel Detector: Proposed ML implementation

Digital neuromorphic implementation



Analog – Mixed Signal implementation using floating gates or memristive cross-bar arrays

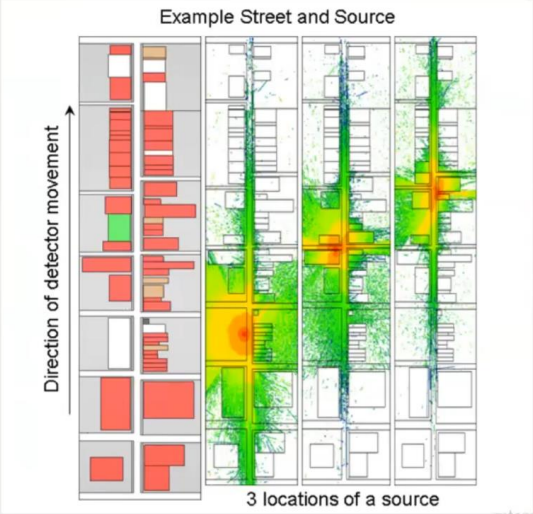
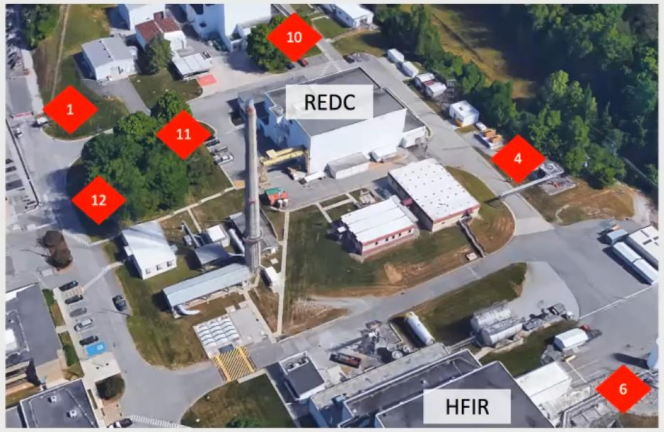


- Ability to work in the latent space (downstream resources)
- Reconfigurability vs. pruning?
- On-chip inference vs. on-chip training?
- Light weight models?
- Can lead to self calibrating detectors?

NeuroRad Project at ORNL

Slide courtesy of NeuroRad project @ ORNL
James Ghawaly

- 1: Develop a neuromorphic-capable radiation anomaly detection algorithm and evaluate on both simulated and real-world data.
- 2: Integrate neuromorphic algorithm on μCaspian board and integrate board with low power radiation detection system.

Datasets	
DOE Urban Search Challenge [1]	HFIR/REDC Static Monitors [2]
• Single 2"x4"x16" NaI(Tl) detector moving through urban street. • 9700 training runs, 15840 testing runs 	• Multiple static sensor "nodes" each with a single 2"x4"x16" NaI(Tl) detector, placed around ORNL HFIR/REDC facility. • <200 source encounters 

OAK RIDGE National Laboratory

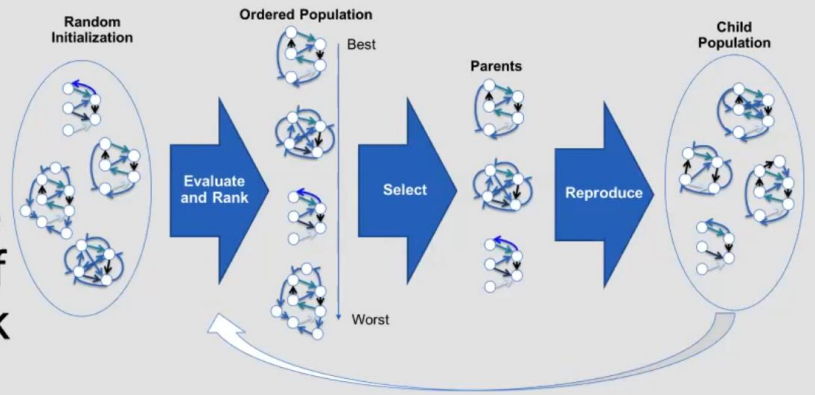
Algorithms

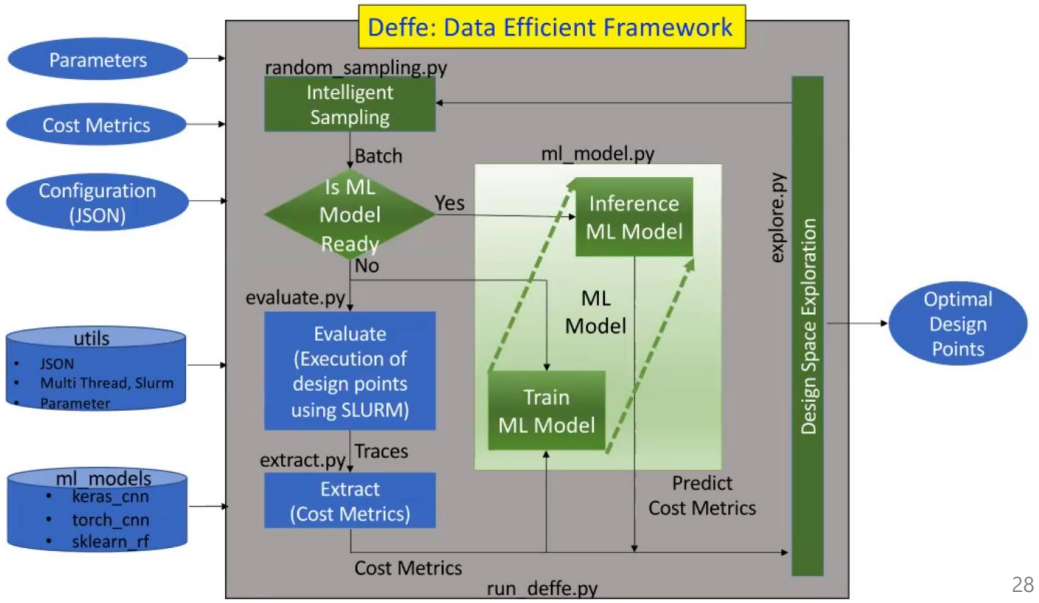
Algorithms

- Find the best algorithms for specific problems (like CMS sensors)
  - Include comparison against SOA techniques
- Optimize algorithmic options for specific application
  - Identify encoding of input vector
  - Evaluate different configurations with simulation
- Training, Inference, Online
- Interact with software and architecture teams
- Tools
  - **EONS** (Evolutionary optimization) for training
  - **Deffe** for Hyperparameter optimization using transfer learning

EONS

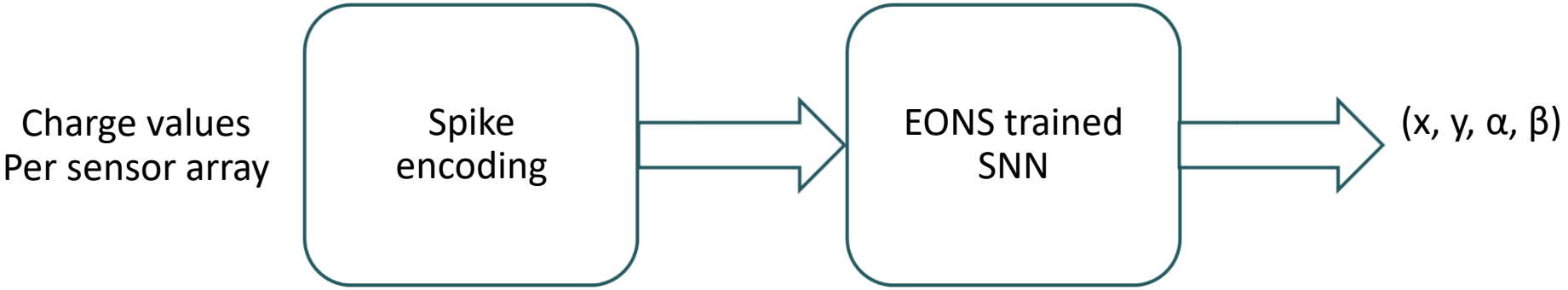
- Generates relatively sparse networks
- Evolves the structure of the network



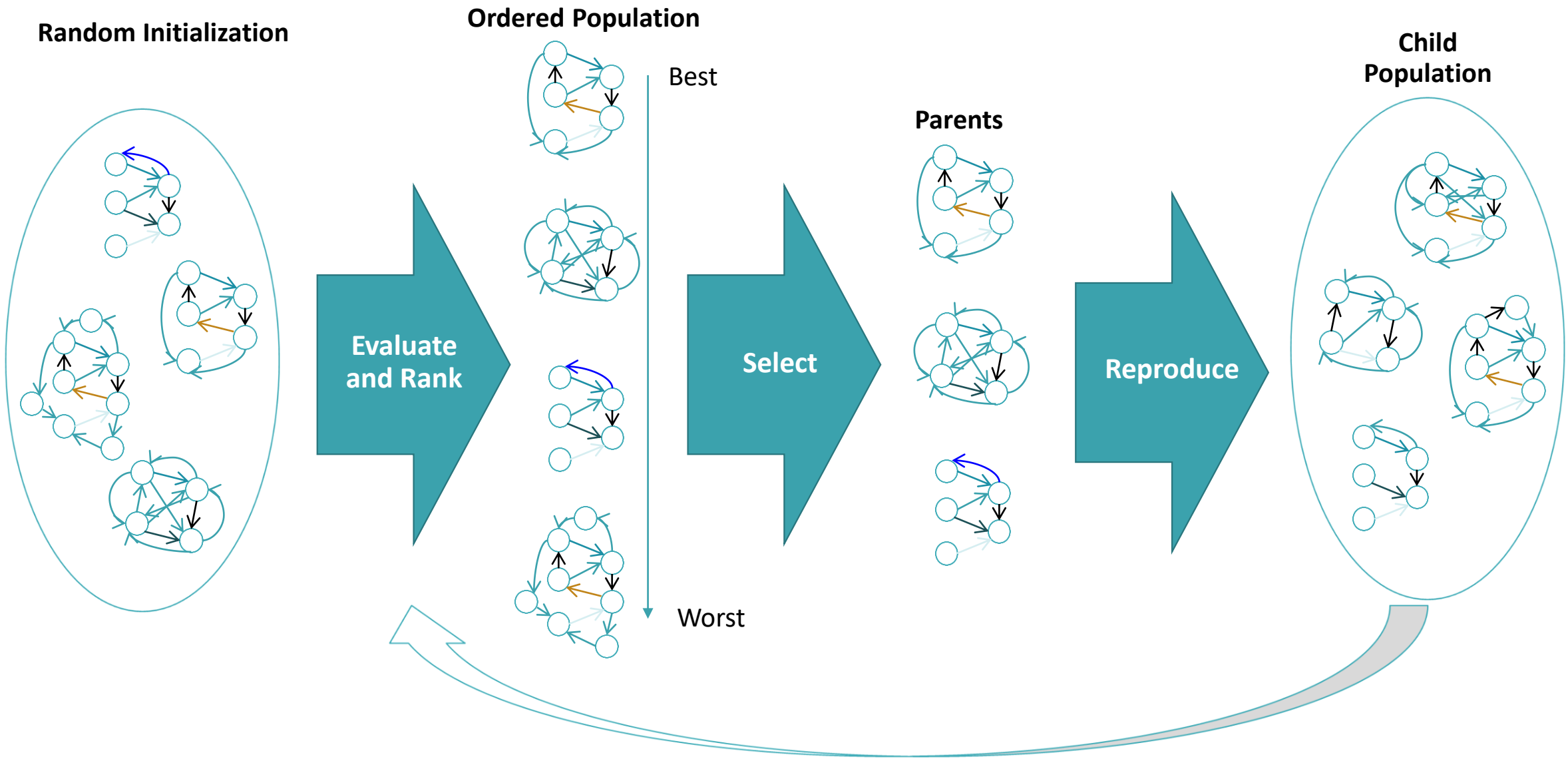


Neuromorphic Approach for Smart Pixel Detection

- Goal
  - Charge values from the sensor every 25ns
  - Data Compression, send only particle track information – (x, y, α, β)
  - PPA: in-sensor pixel detection before ADC - hence, detection model needs to be small
- First approach
  - Develop simulation for test data
  - Apply EONS to identify configuration for SNN algorithm
  - Explore spike encoding of charge values – may need to support rational numbers
- Compared against other approaches
  - Regression, Spiking convolution NN, unsupervised learning (STDP), Spike-based Object detection algorithms



Evolutionary Optimization of Neuromorphic Systems (EONS)

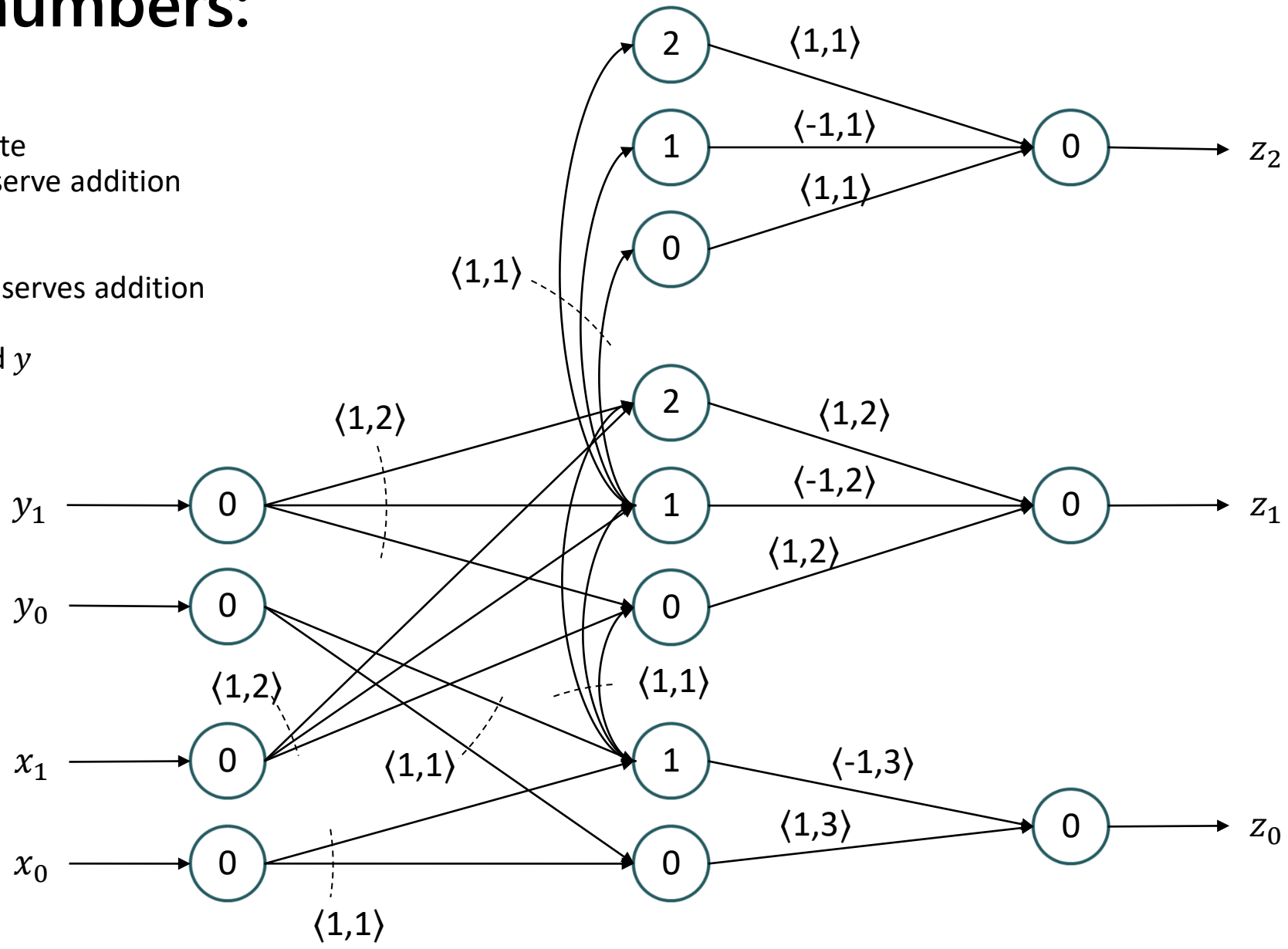


[Schuman et al. 2020 ACM NICE Workshop]

Encoding rational numbers: the Virtual Neuron

- Current encoding methods are inadequate
  - Rate-based encoding does not preserve addition
  - Binning loses information
- Virtual neuron uses binary encoding, preserves addition
- Takes two 2-bit numbers as inputs: x and y
- Returns a 3-bit number as output: z
- Implemented in NEST simulator

x_1	x_0	y_1	y_0	z_2	z_1	z_0	Sum
0	1	0	1	0	1	0	1+1=2
0	1	1	1	1	0	0	1+3=4
1	0	1	1	1	0	1	2+3=5
1	1	1	1	1	1	0	3+3=6

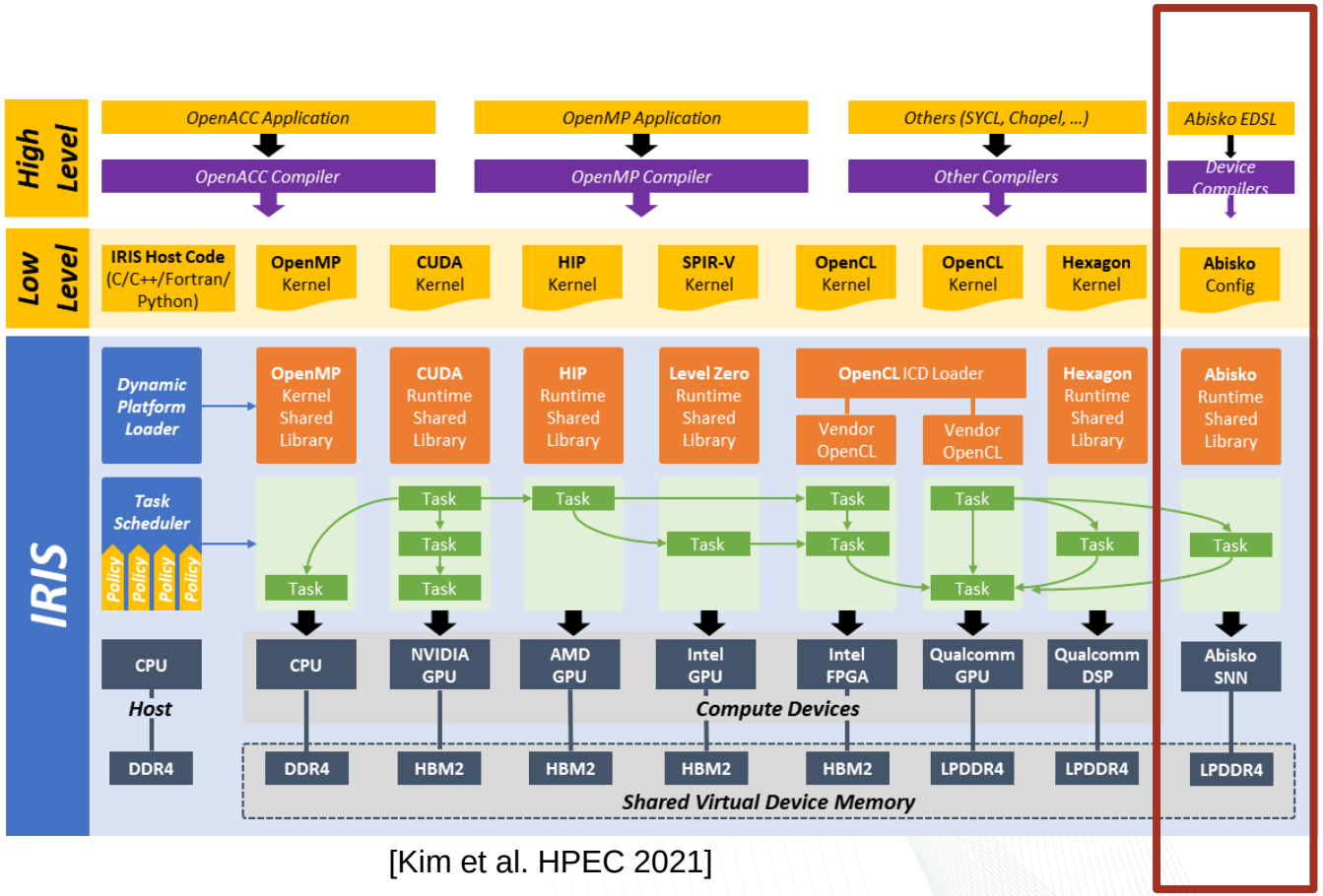


[P. Date et al. Submitted]

Software

Software

- Develop a holistic software stack for neuromorphic coprocessing on heterogeneous architectures with a focus on
  - Language optimizations and code gen with LLVM and MLIR
  - Runtime portability and integration with IRIS
- Portable across GPU, FPGA, SoC, and Abisko chiplet simulator
- Based on successful experiences with Quantum computing at ORNL:
  - XACC, QCOR
- Building embedded DSL (Domain Specific Language) with LLVM and MLIR



[Kim et al. HPEC 2021]

XACC/QCOR Approach for Algorithmic Portability across Many Quantum Architectures

```
#define __qpu__ [[clang::syntax(qcor)]]

__qpu__ void bell(qreg q, int shots) {
   ... quantum DSL here ...
}
```

```
void bell(qreg q, int shots) {
   void internal_bell_call(qreg, int);
   internal_bell_call(q, shots);
}

class bell : public QuantumKernel<bell, qreg, int> {
public:
    bell(qreg q, int shots)
       : QuantumKernel<bell,qreg,int>(q, shots) {}
    ~bell() {
       ... execute on backend ....
    }
};

void internal_bell_call(qreg q, int shots) {
   class bell kernel(q, shots);
}
```

QuantumKernel

Runtime - libqrt.so

QuantumRuntime

qcor SyntaxHandler rewrites function to QuantumKernel subtype.

Program call to bell function is a call to another internal function that instantiates a temporary instance of the new QuantumKernel sub-type.

Investigating Software Abstractions for SNN

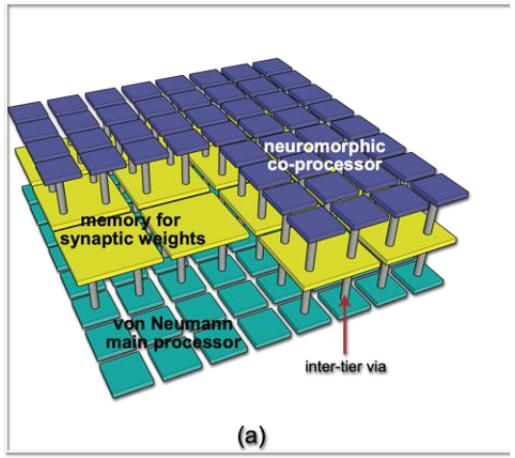
- Prototyping solutions in multiple SNN frameworks to understand what might constitute EDSL feature set
  - NEST, PyNN, BRIAN, Nengo
  - TENNlab
  - LAVA
  - *FUGU*
- Additional Questions in SNNs
  - Hyperparameter optimization of SNN
  - Spike encoding

```
# Create the neurons
neurons_dict = {}
i = 0
for n in sorted(network.nodes()):
    node = network.get_node(n)
    neurons_dict[node.id] = i
    i += 1
    neuron = nest.Create(self.config["neuron_model"])
    self.nest_neurons.append(neuron)
    threshold = float(node.get("Threshold"))
    nest.SetStatus(neuron, {'V_th' : threshold})

    self.setup_neuron_parameters(neuron)

# Create the synapses
for e in network.edges():
    edge = network.get_edge(e[0], e[1])
    pre = self.nest_neurons[neurons_dict[e[0]]]
    post = self.nest_neurons[neurons_dict[e[1]]]
    w = float(edge.get("Weight"))
    d = float(edge.get("Delay"))
    if (self.config["stdp"] == False):
        nest.Connect(pre, post, syn_spec={'weight' : w, 'delay' : d})
    else:
        nest.Connect(pre, post, syn_spec={'weight' : w, 'delay' : d, "model": "stdp_synapse"})
```

Architecture and Integration

Architectures

- Design chiplet for SNN that can be easily integrated with contemporary technologies
  - Heterogeneous integration
  - Compatible with existing processes
- Extensive advances in chiplets, packaging, and heterogeneous integration recently
  - Open Domain-Specific Architecture
  - TSMC SoIC-CoW, Intel Foveros
- Using open toolchain and architecture to explore chiplet designs: RISC-V, OpenLane
- Simulate/emulate with existing simulators like Gem5 and Aladdin
- Initial work with uCaspian boards



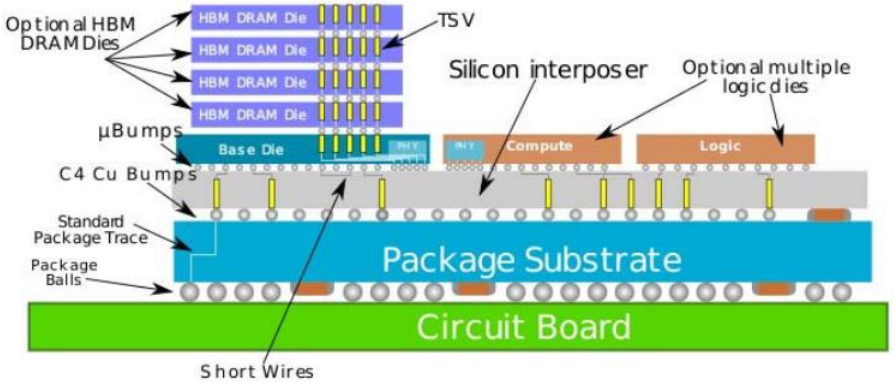
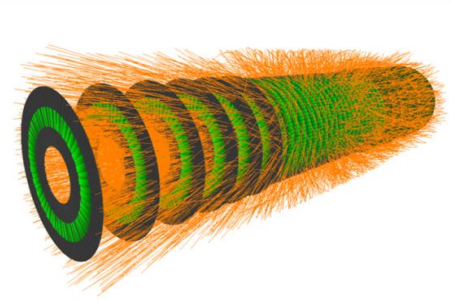


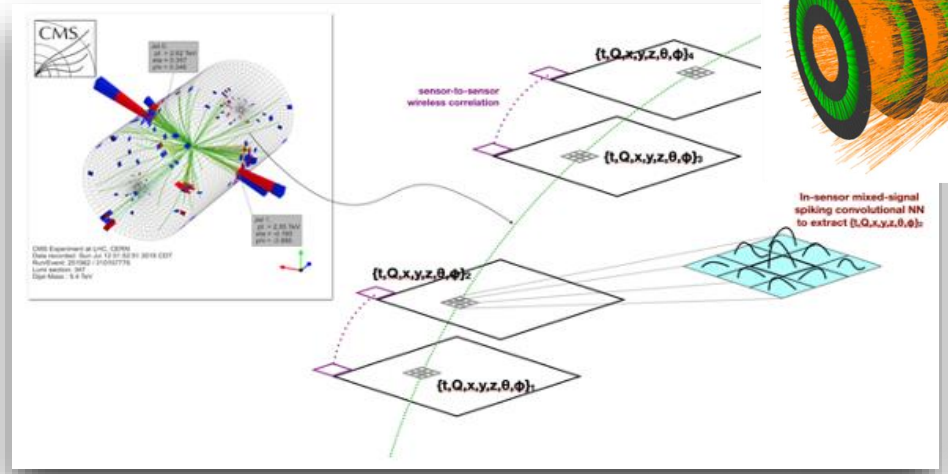
Figure 21. An example showing the use of 2D and 3D interconnections (courtesy TSMC)

[IEEE HIR 2021]

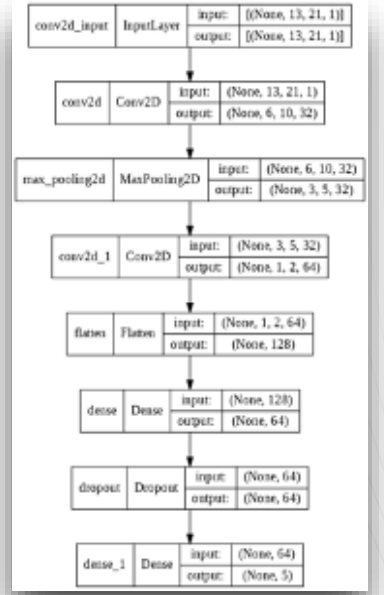
Architecture for Smart Pixel Driver

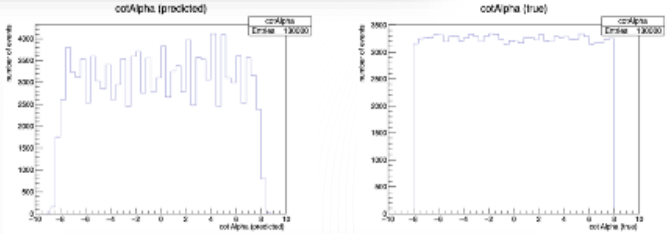
- CMS Experiment from FemiLab: 25ns latency, ~1B detector channels
  - Active ongoing effort to design customized ASIC for data acquisition and compression
  - Active ongoing effort to establish POR ML method on particle trajectory reconstruction

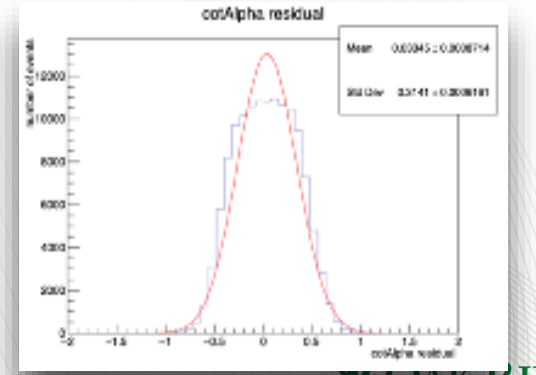




- Establish baseline specs (PPA) in computing intensity required using POR ML method
- work with Algorithms team to explore SNN techniques to better meet other constraints
- Investigate and define integration between ML accelerator cores and von-Neumann cores



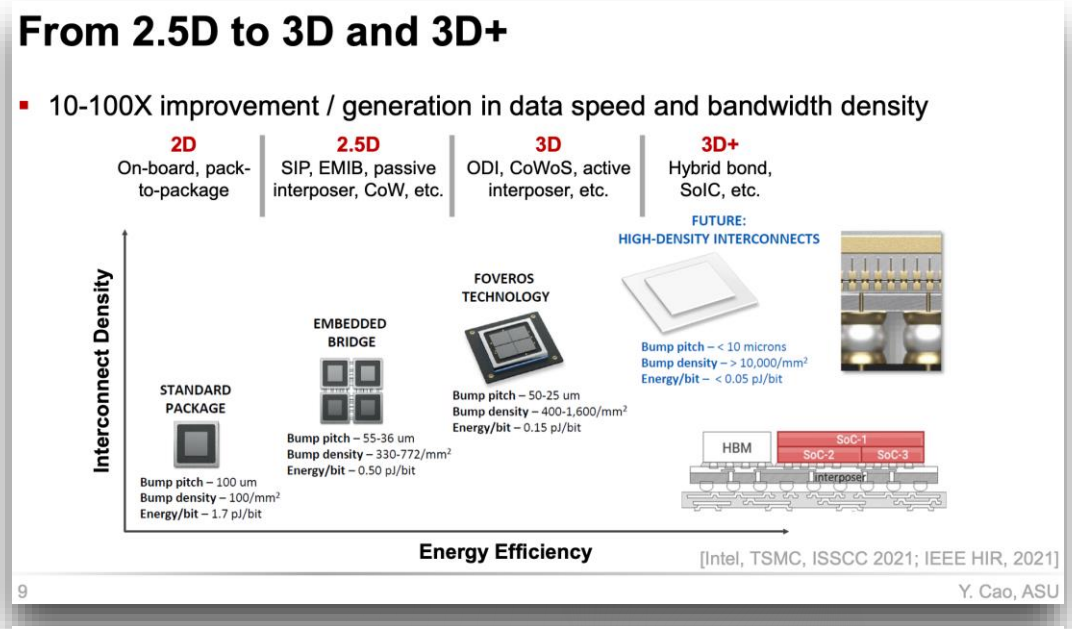


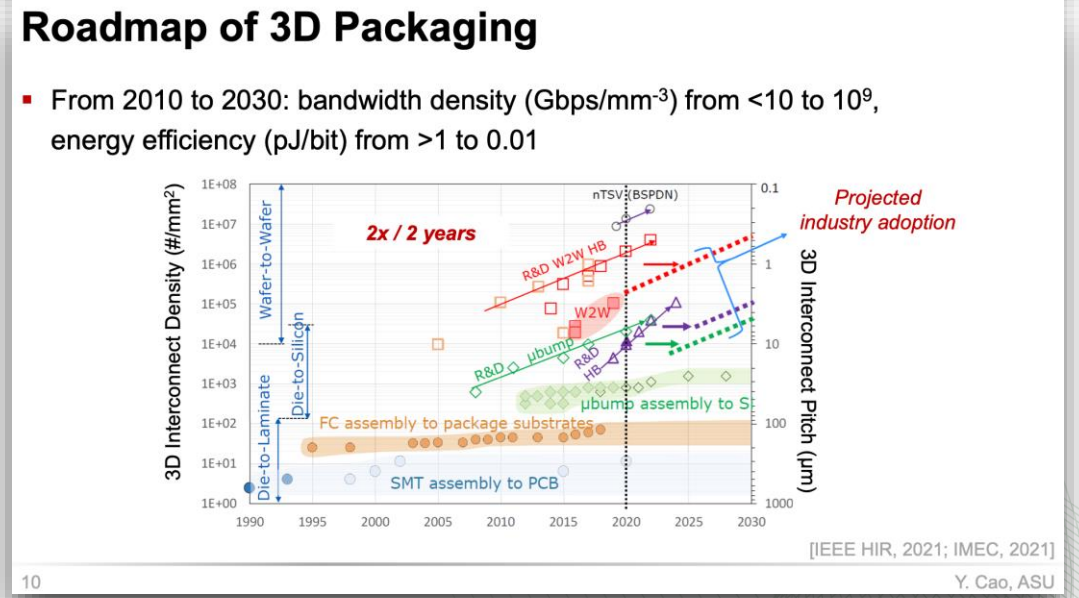


Tracking Integration and Packaging Technology

- Advanced packaging is clearly one of the main technology drivers of semiconductor scaling soon
- Underlying technology is the main uncertainty for neuromorphic accelerator

	SPIKING	NON-SPIKING
DIGITAL	CMOS-friendly (Loihi) latency and energy constraints	traditional GPU/FPGA/NN accelerators
ANALOG	interface to the rest of world, repeatability	Interface, repeatability



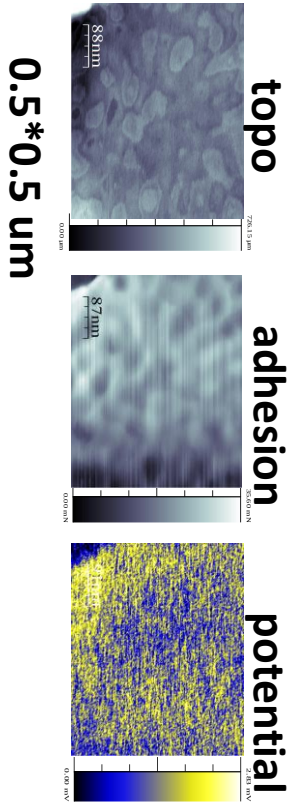


Devices and Circuits Materials

Materials, Devices, Circuits

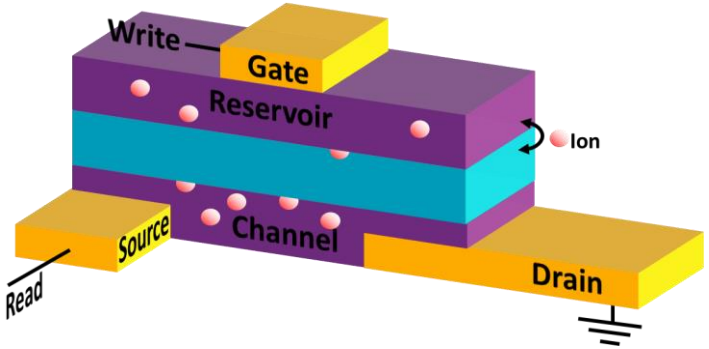
- Goals
  - Harness the interplay between mobile defects (ions and vacancies) and electronic properties to realize functional elements for spiking and non-spiking analog neuromorphic networks
  - Create and validate small network models; generate device and network data for co-design
  - Understand and mitigate radiation induced degradation mechanisms at the device and circuit level

Materials

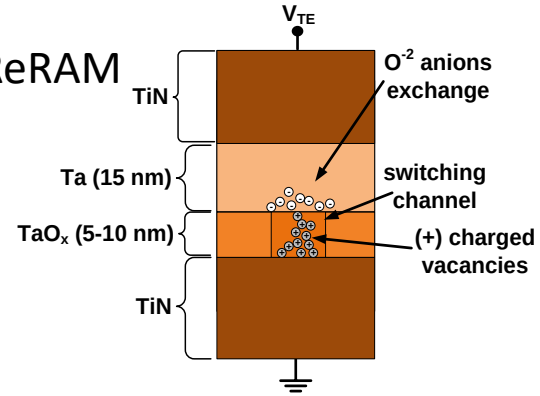


Devices

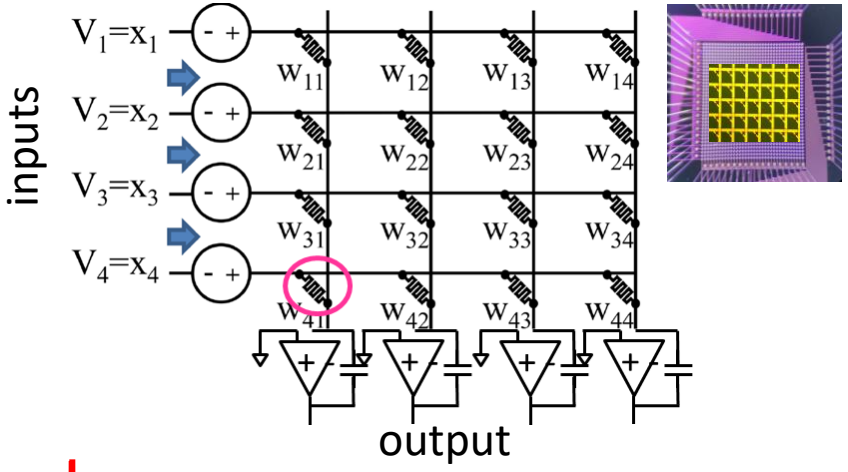
1) ECRAM

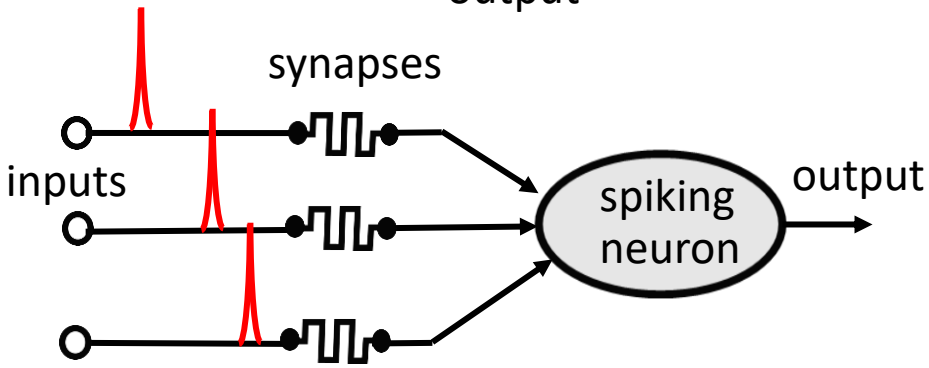


2) ReRAM



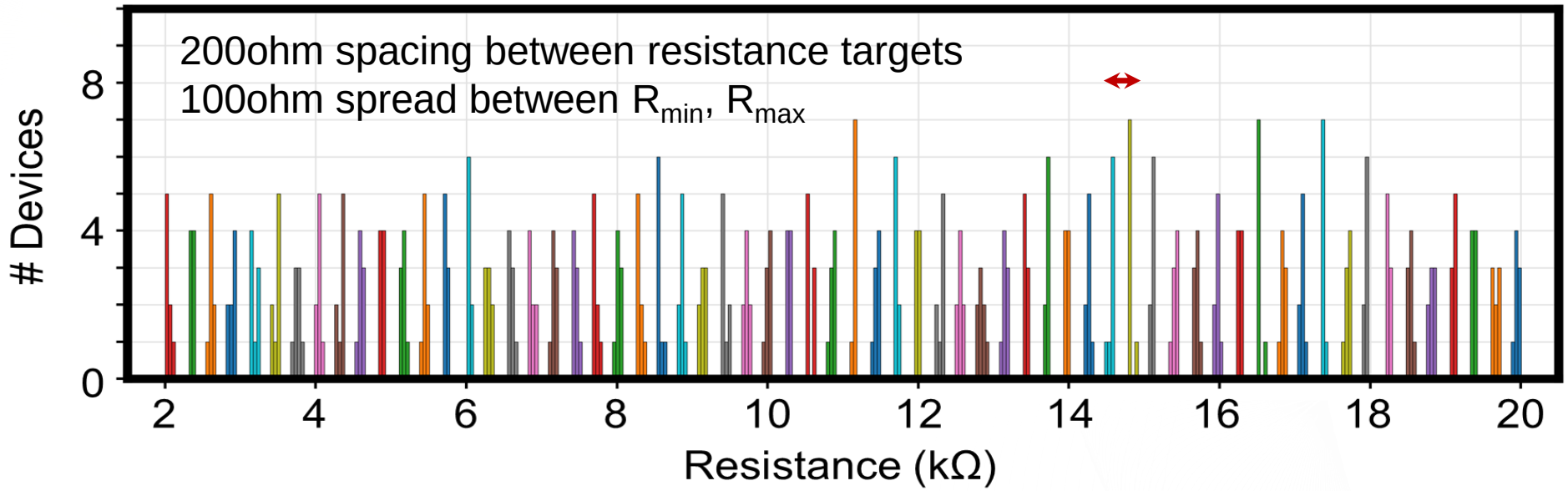
Circuits

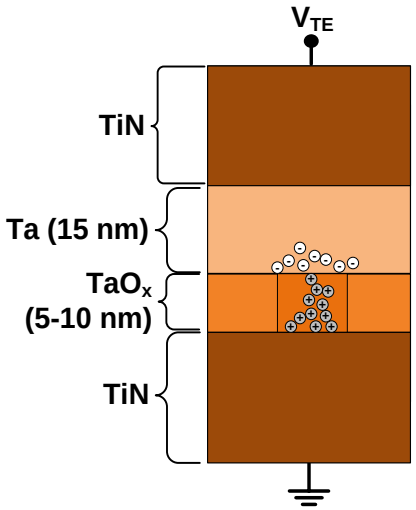


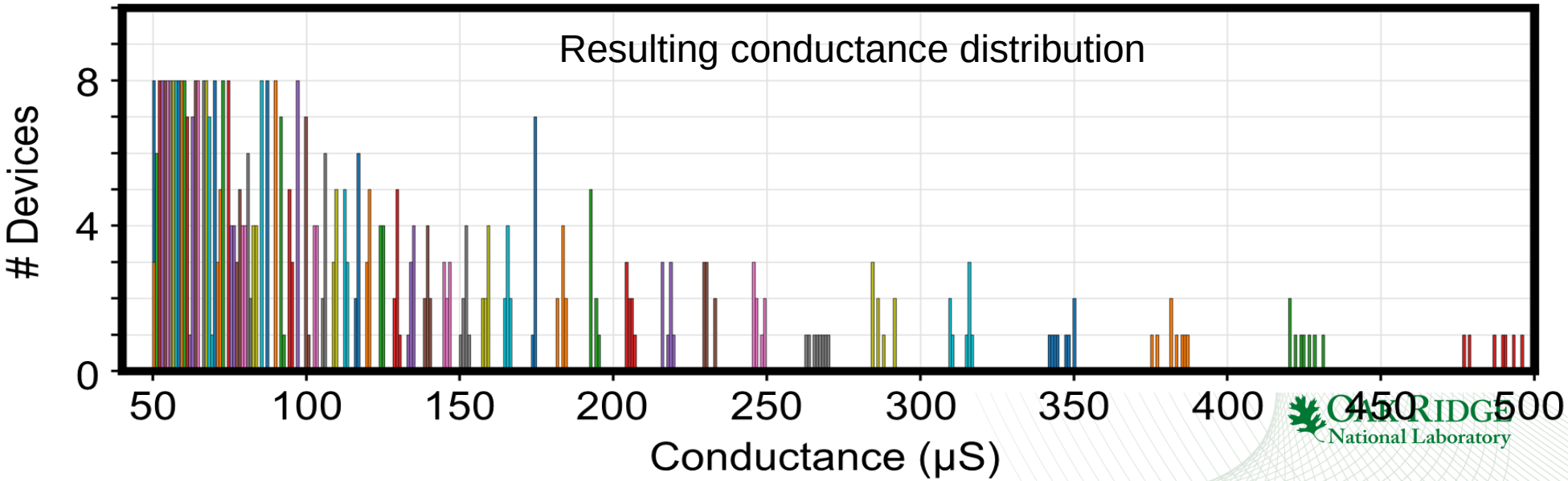


Experimental TaOx ReRAM Conductance Distributions

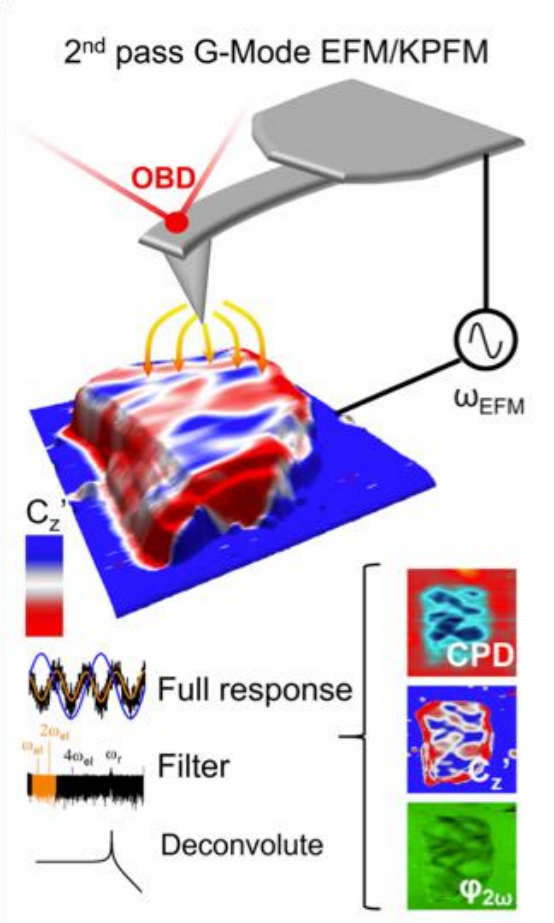
Developed TaOx weight mapping and programming routine for optimizing inference accuracy





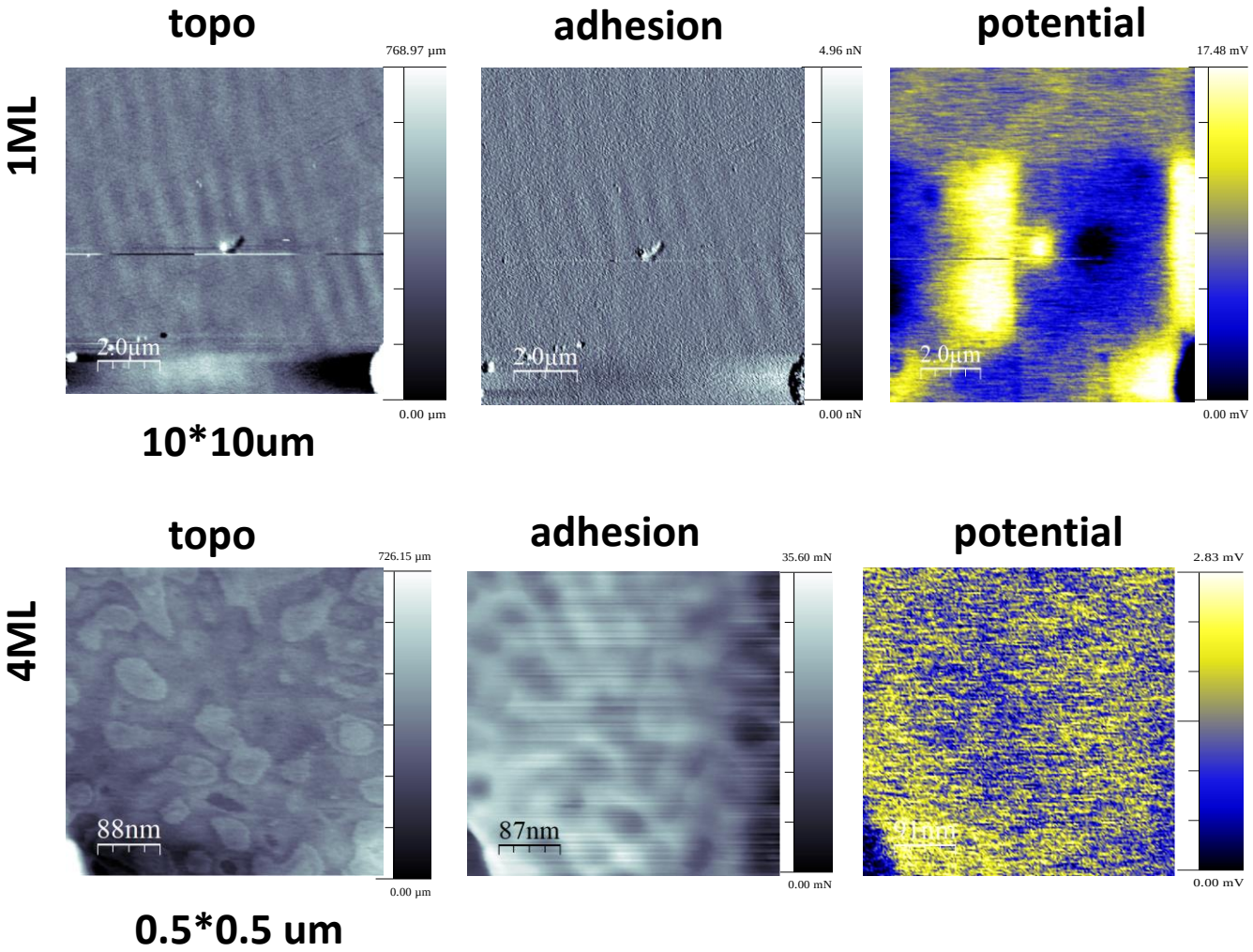


Kelvin Probe Force Microscopy (KPFM) on PB thin films



The principles of the measurement procedure in KPFM technique using two pass mode

M.Checa et al, APL , **2021**

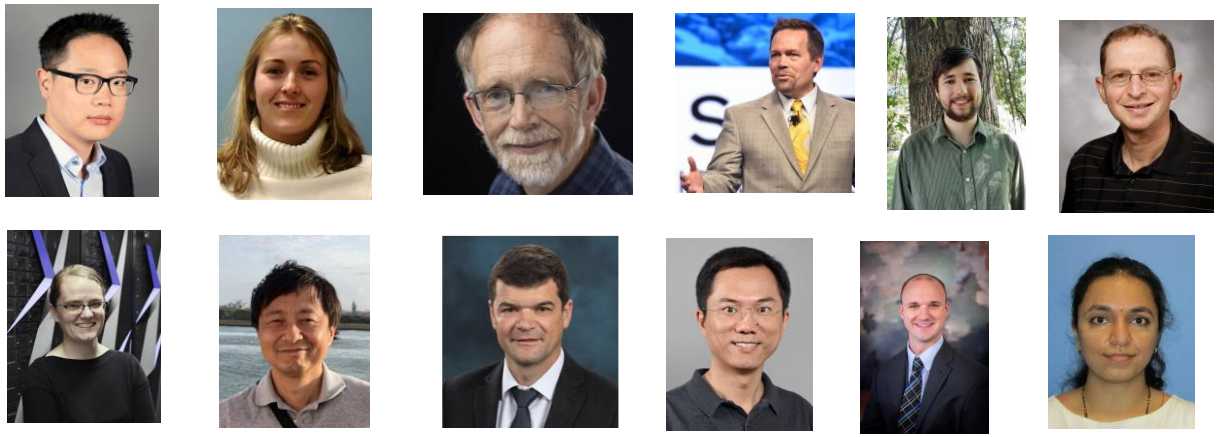


Next step: nanoscale ionic effects from dielectric spectroscopy

Conclusions

Summary

- **Abisko** is a new microelectronics codesign project with the ambitious goals
  - Develop better techniques for codesign from algorithms to devices and materials
  - Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
  - Explore new devices and materials for the SNN chiplet
  - Design language abstractions and runtime support for SNN chiplet
- Truly interdisciplinary team working across the stack
- More information
  - vetter@computer.org
  - https://vetter.github.io
- We are hiring!
  - See https://jobs.ornl.gov
  - Send an email to me.



. . .

Brooks, David	Cao, Kevin	Comish, John
Date, Prasanna	Fahim, Farah (collaborator)	Flynn, Michael
Ghawaly, James (collaborator)	Hornick II, Michael	Huber, Joseph
Hysmith, Holland	Ievlev, Anton	Kulkarni, Shruti
Lim, Sung-Kyu	Liu, Frank(Arch)	Maksymovych, Petro (Materials)
Marinella, Matthew	Miniskar, Narasinga Rao	Ovchinnikova, Olga S.
Schuman, “Katie” (Algo)	Sumpter, Bobby	Talin, Alec (Devices)
Tallada, Marc Gonzonlas (Software)	Tran, Nhan (collaborator)	Tripathy, D
Vetter, Jeffrey S.	Wei, Gu-Yeon	Young, Aaron

Thanks!