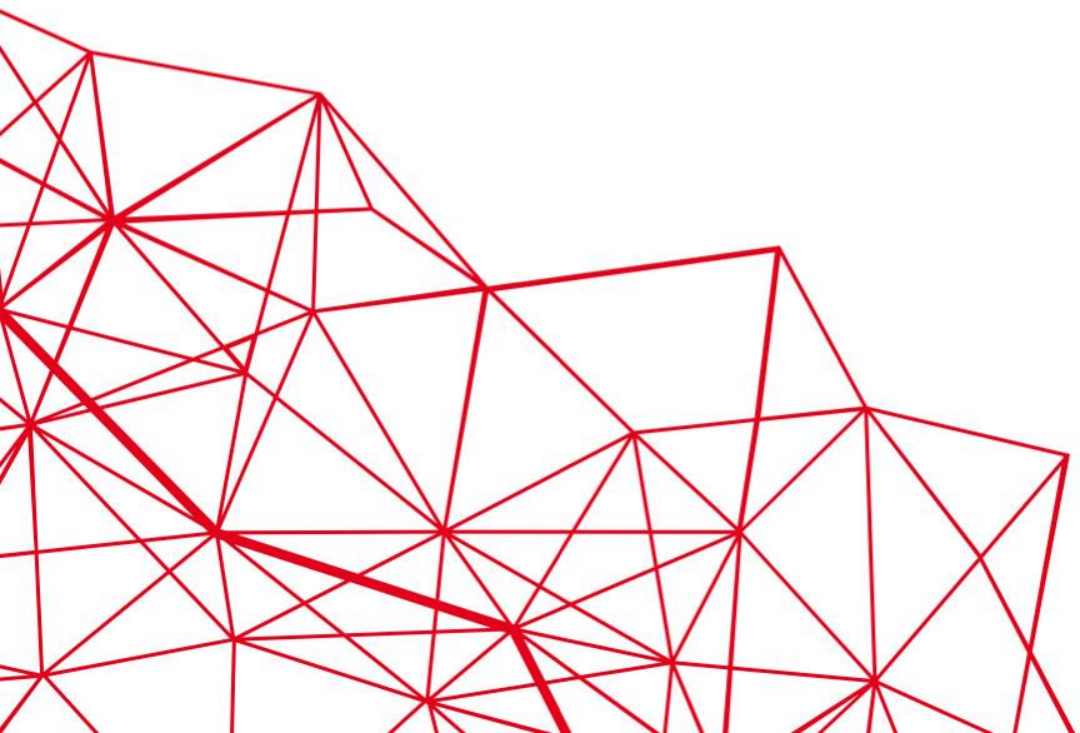
SAND2024-05780C



Large-Scale Neuromorphic Computing at Sandia National Labs from Algorithms to Architectures

Presented by
Craig M. Vineyard (cmviney@Sandia.gov)

Sandia National Laboratories

Exceptional service in the national interest

Large-Scale Neuromorphic Computing at Sandia National Labs from Algorithms to Architectures

PRESENTED BY

CRAIG M. VINEYARD (CMVINEY@SANDIA.GOV)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

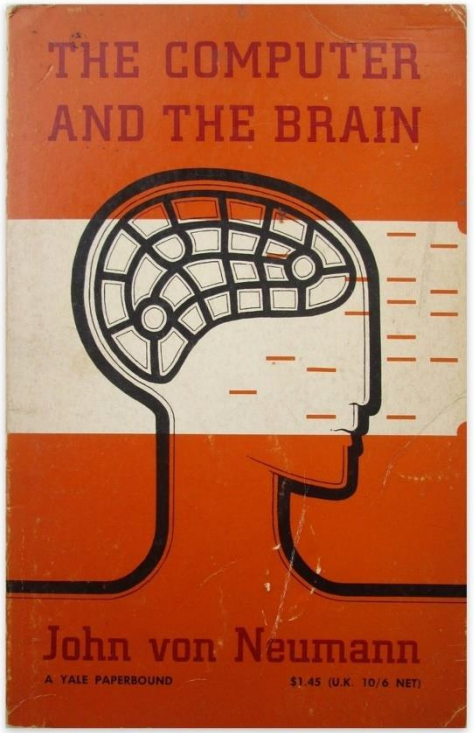
U.S. DEPARTMENT OF ENERGY

NNSA

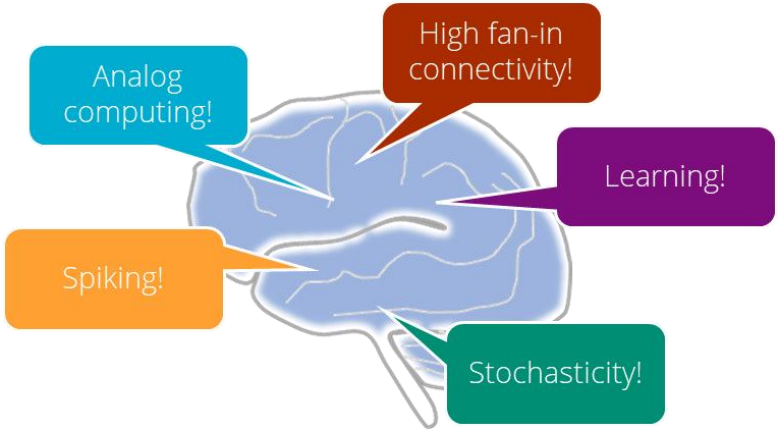
Neural-inspired computing

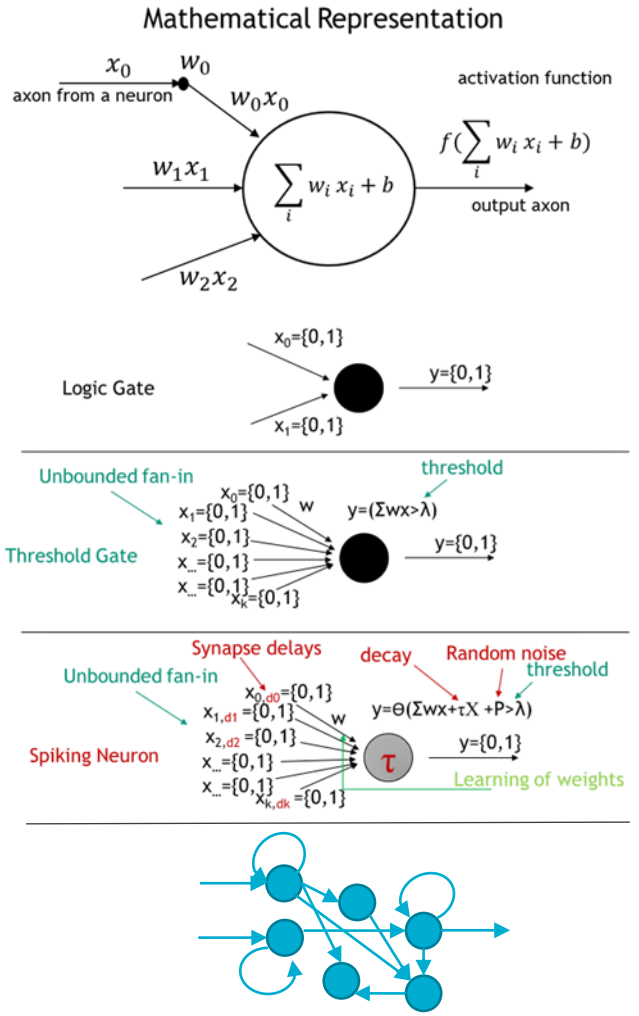
What is neural-inspired, neuromorphic, brain-inspired computing?

- Many terms
- Fundamental notion of taking inspiration from how the brain performs computation

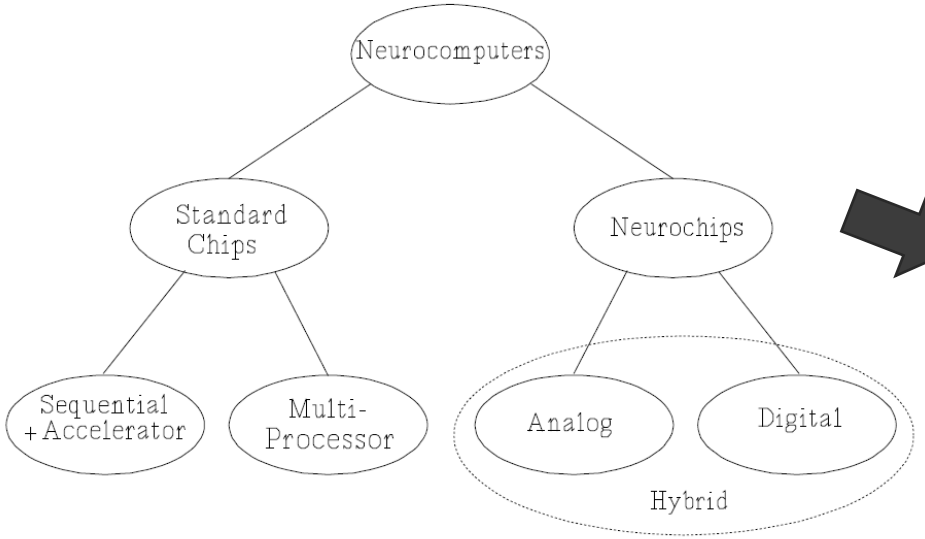


Source: https://digitalminds2016.wordpress.com/wp-content/uploads/2018/02/364-2.jpg?w=325





Neural-inspired computing

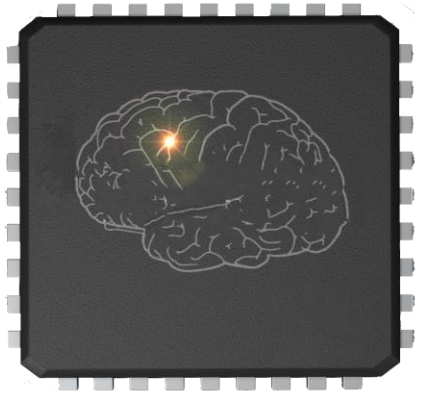


Source: Heemskerk, Jan NH. "Overview of neural hardware." *Neurocomputers for brain-style processing. Design, implementation and application* (1995).

Realized Features of Brain Inspiration in Neuromorphic Hardware

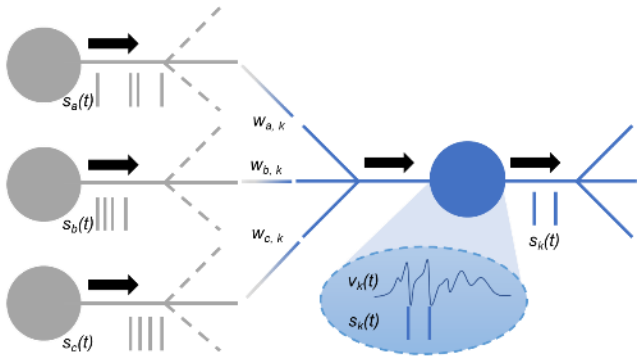
- Event-driven communication
- Graph based connectivity
- Processing in Memory
- In situ learning
- Analog computation
- Post-Moore's Law Devices
- Ubiquitous stochasticity

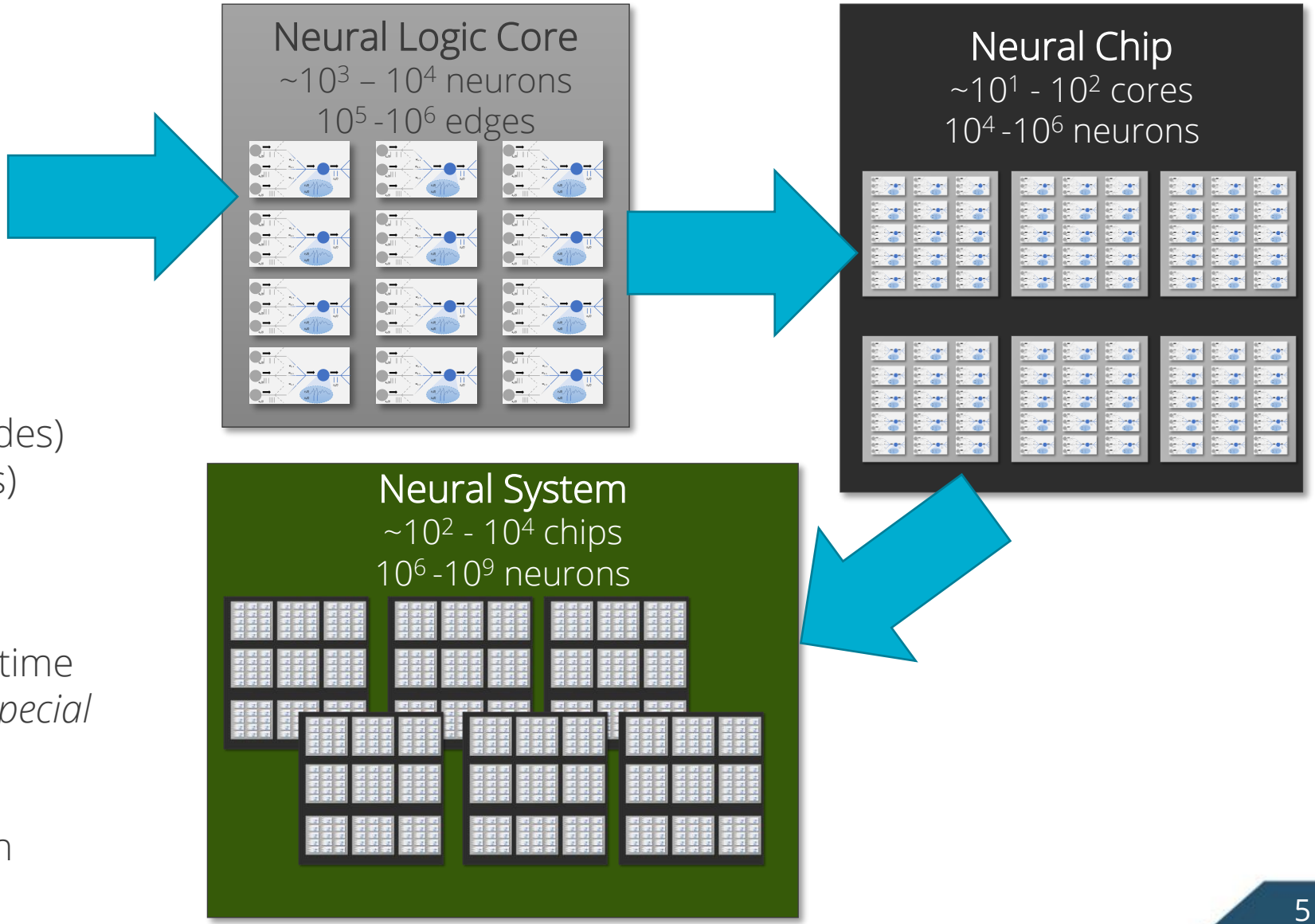
Realized Today for 10x-100x energy savings



Active Research for additional power savings

Spiking neuromorphic systems



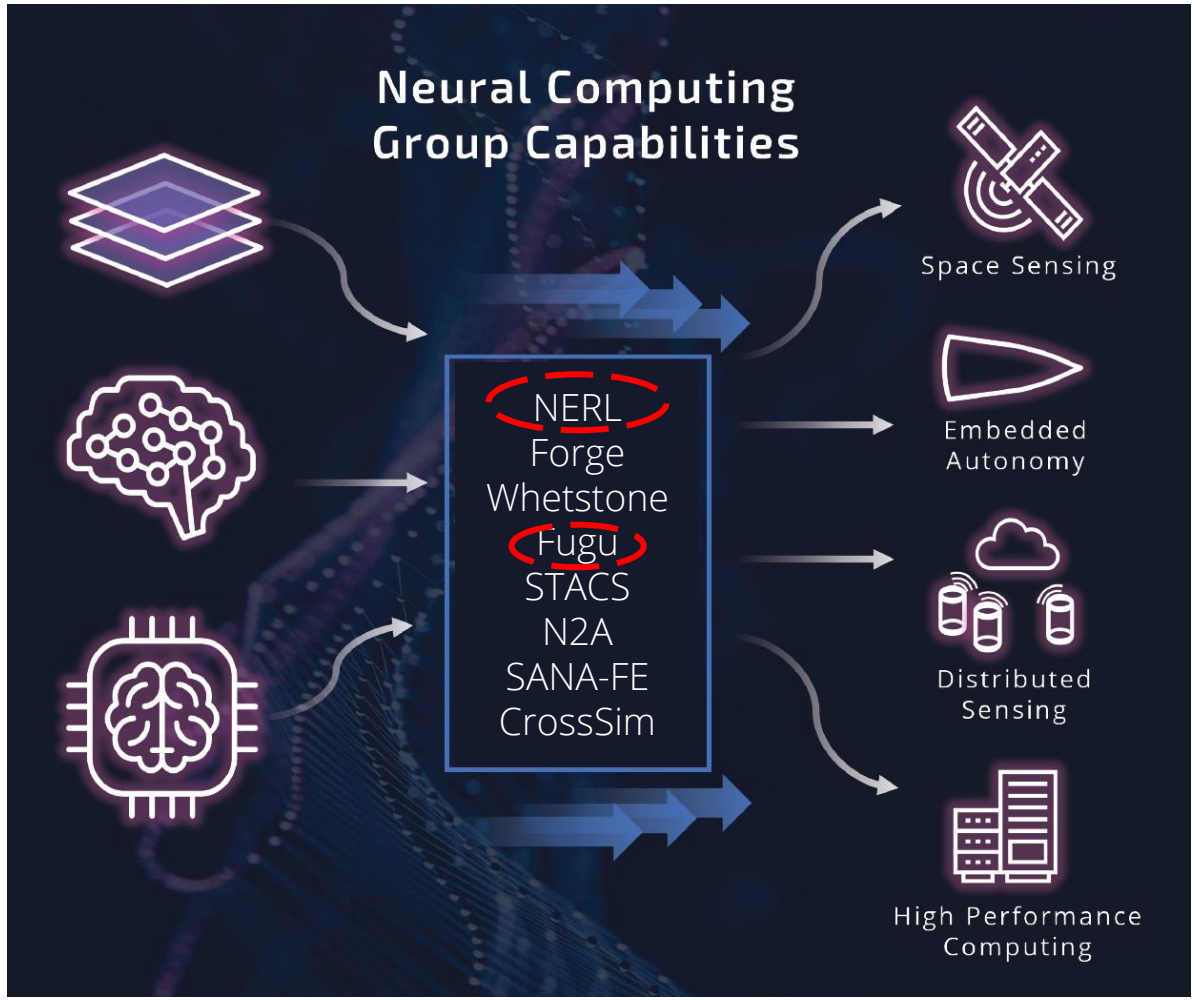


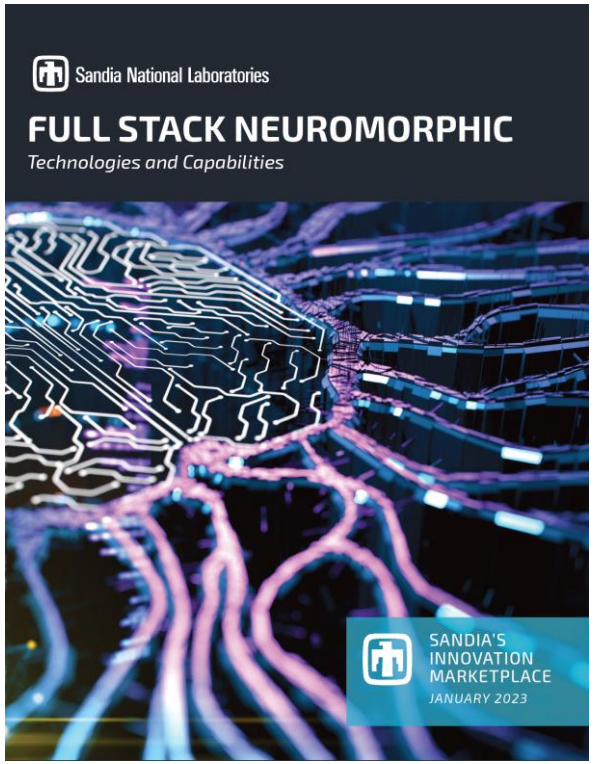
Computational Primitives

- Spiking Neurons (vertices / nodes)
- Synapses (connections / edges)

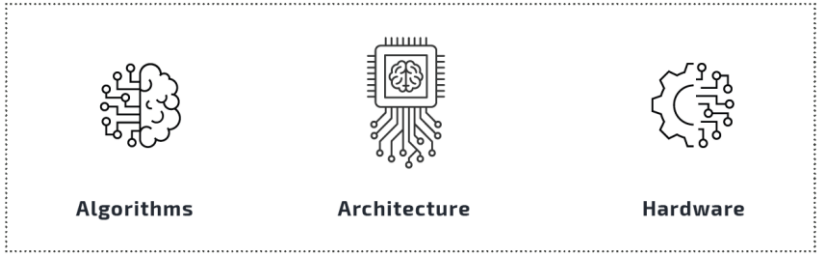
Programmable as arbitrary graphs

- Edges: Directed and weighted
- Nodes: Threshold gate logic + time
- *Artificial neural networks are a special case*
- Programmability, theoretical, analysis and software are open research questions





https://ip.sandia.gov/opportunity/full-stack-neuromorphic/



Sandia's Neural Computing group is building a suite of cross-cutting capabilities to bring neural hardware, algorithms, and AI closer to applications

What is Fugu? And Why?

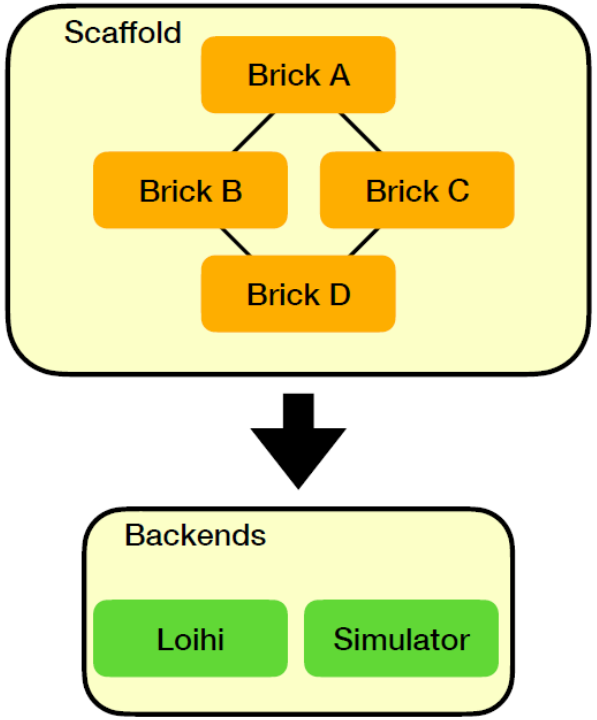
Neuromorphic Challenges

- Neuromorphic platforms remain a challenge to program
- Lack of interoperability between research outputs

Fugu

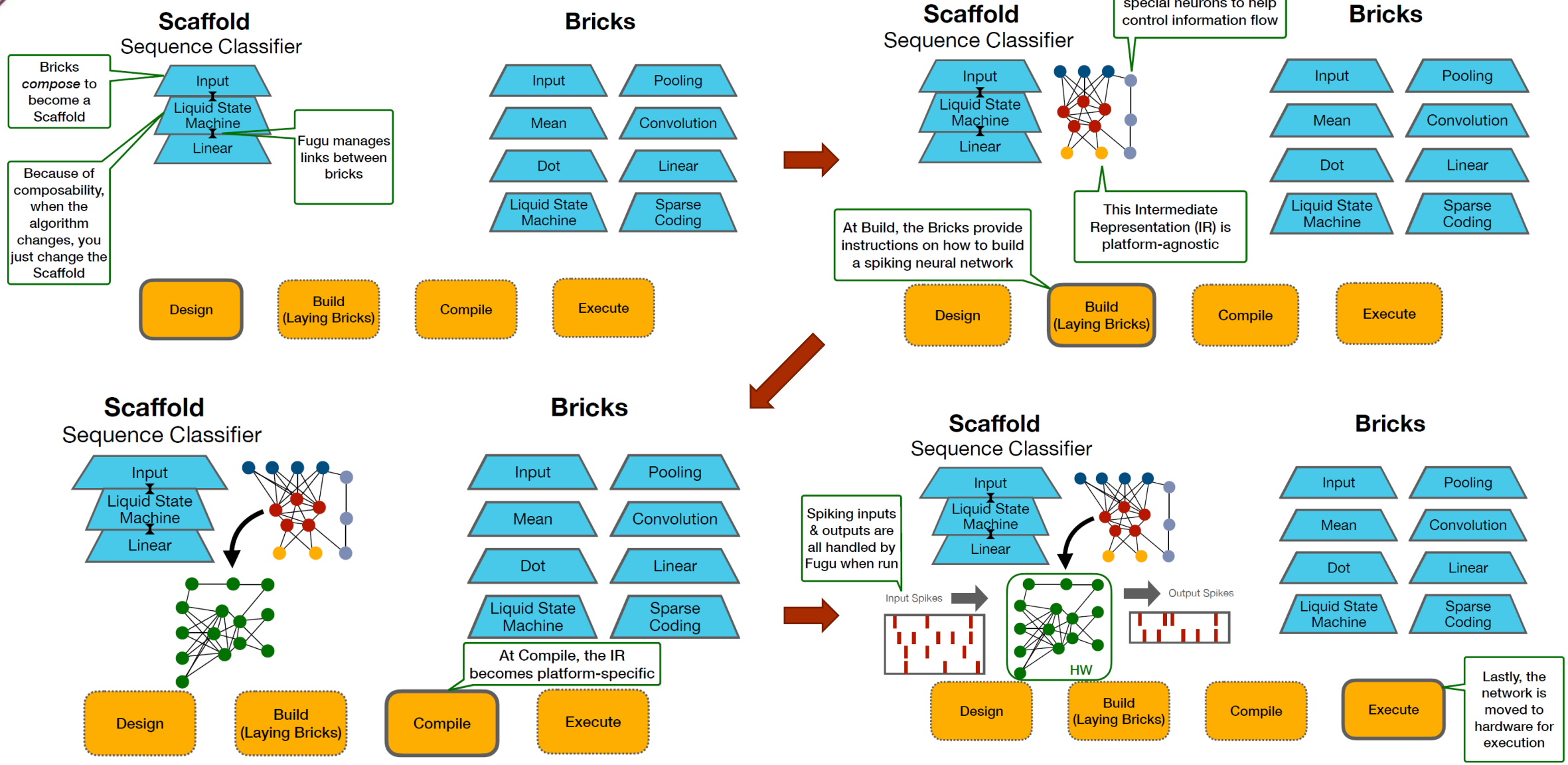
- Open-source library for spiking neural networks
- A unified, (mostly) hardware agnostic, framework to enable neuromorphic algorithm development
  - Bricks: roughly represents a function
  - Scaffolds: represents an application
- Design goals: easy-to-use, lower barrier of entry, improved code efficiency and re-use
- In active development

https://github.com/sandialabs/Fugu

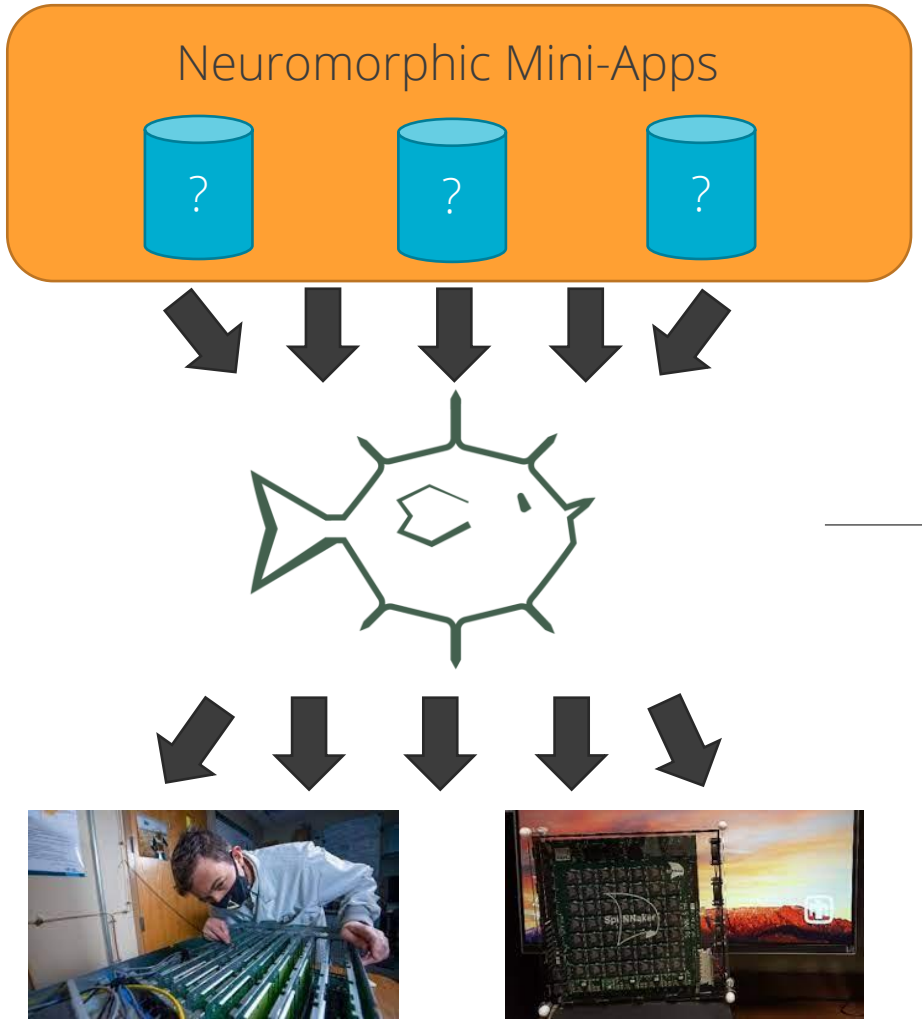


The name Fugu is inspired by the Japanese word for pufferfish; which, of course, have spikes. Furthermore, Fugu is considered a culinary delicacy due to the presence of low levels of the neurotoxin tetrodotoxin, or TTX, which has significant value in studying the electrophysiology mechanisms underlying biological action potentials.

How Fugu works



Fugu addresses two key challenges of neuromorphic programming



Composability

Deploying applications on neuromorphic hardware requires implementing algorithms within neural circuits

- Need to be able to build applications from well designed kernels
- Need to take advantage of features offered by spiking neuron model

Portability

Programming neuromorphic platforms requires a *graph* of neurons (nodes) and synapses (edges)

- Need to represent neural algorithms in common graph format
- Need ability to translate graph into backend specific constraints

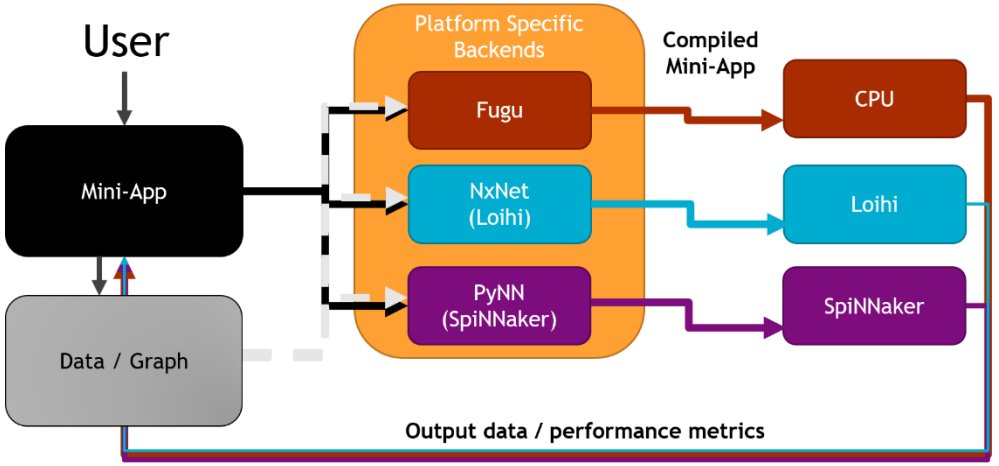
Neural mini-app structure

Single-line Python interface

- python fluence_mini_app.py --run_mode loihi --neural_timesteps 10000 -v 100 -dt .02 -ss .05 -da .2 -M 200

Can run multiple backends from same function

- Currently have worked with Fugu, Loihi, SpiNNaker



Flags to set Mini-App specific parameters

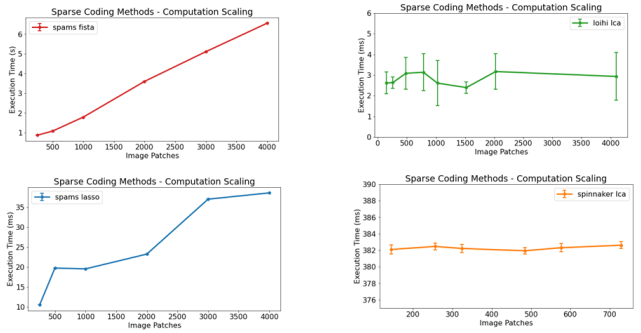
- Scaling parameters (e.g., # neuromorphic timesteps, # of walkers)
- Implementation parameters (e.g., angle precision, time precision)
- Physics parameters (e.g., particle velocity, scattering probabilities)

Neural Mini-Apps

Sparse Coding/Dictionary Learning

- Sparse linear combinations of elements of a given overcomplete basis set
- On neuromorphic LASSO approximated by LCA

Example Results -

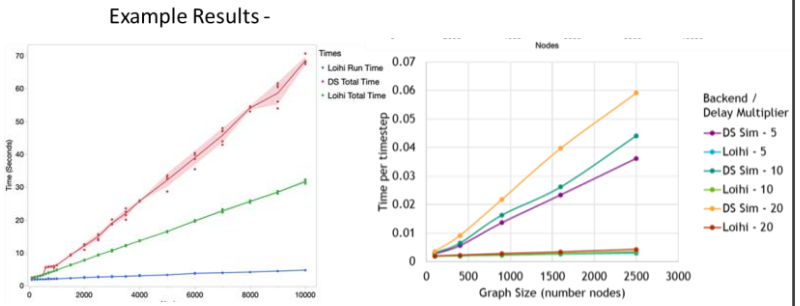


Parameterization	Size of image, Size of image patch, Size of the dictionary, Stride of image patch, Desired sparsity
Scaling	Problem size via # of image patches, Parameters
Metrics	Time for setup, Time for reconstruction, Reconstruction performance, Reconstruction sparsity, Compute resource usage, Energy resource usage

Graph Analysis

- Single Source Shortest Path
- Source neuron spikes and shortest path determined by edges traversed leading to target neuron spiking

Example Results -

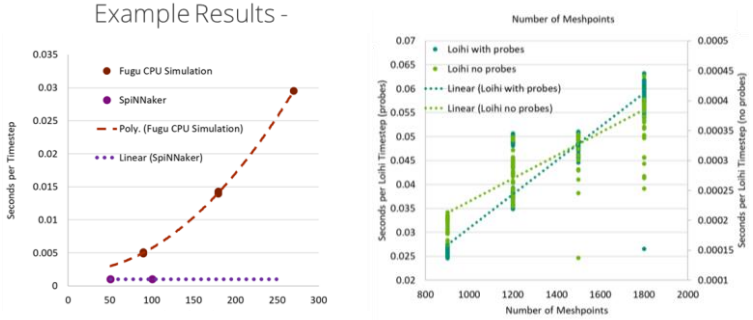


Parameterization	Graph generation (uniformly random tree, small world), Nodes, Weight range, Max runtime, Source, Target
Scaling	Graph scale, Weight/delay range
Metrics	Total time, Time for setup

Random Walk

- Discrete time Markov Chain
- Neuromorphic approach models state & tracks walkers via spike activity

Example Results -

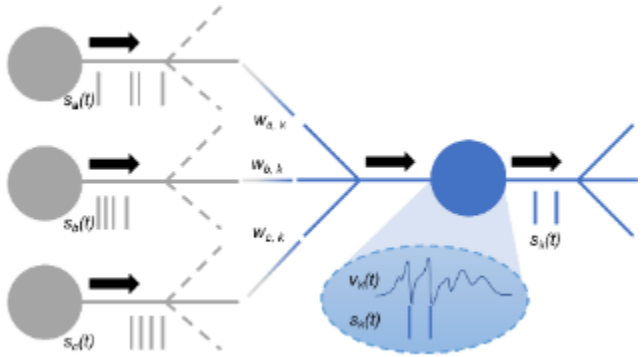


Parameterization	Number of total walkers, Size of direction/relative velocity/angular discretization, Time step size of simulation, Size of the state space, Size of positional discretization
Scaling	Walkers, Mesh size
Metrics	Energy cost of walkers, Time to run, Space to run

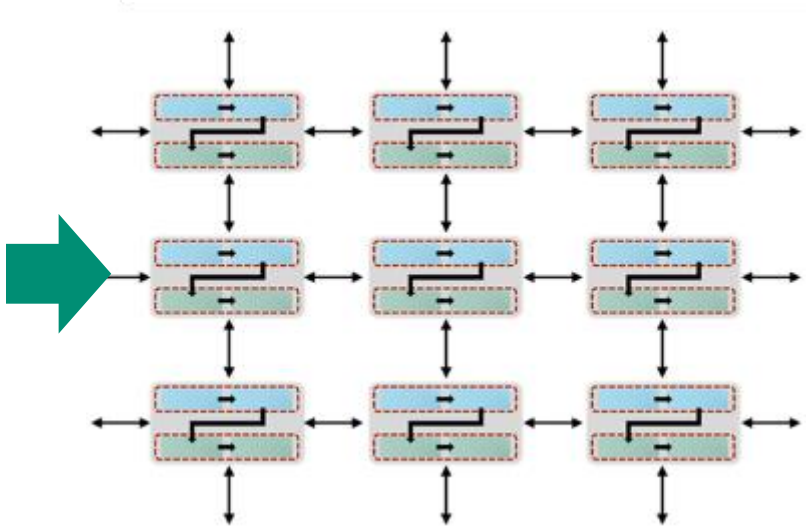
Vineyard, Craig, et al. "Neural Mini-Apps as a Tool for Neuromorphic Computing Insight." *Proceedings of the 2022 Annual Neuro-Inspired Computational Elements Conference*. 2022.

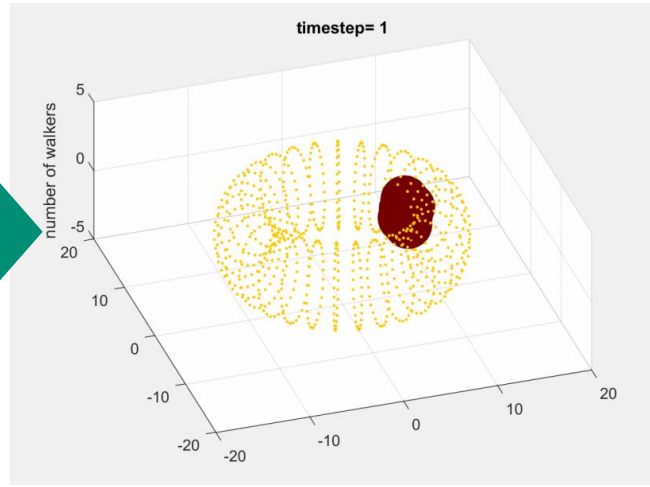
Scientific Computing

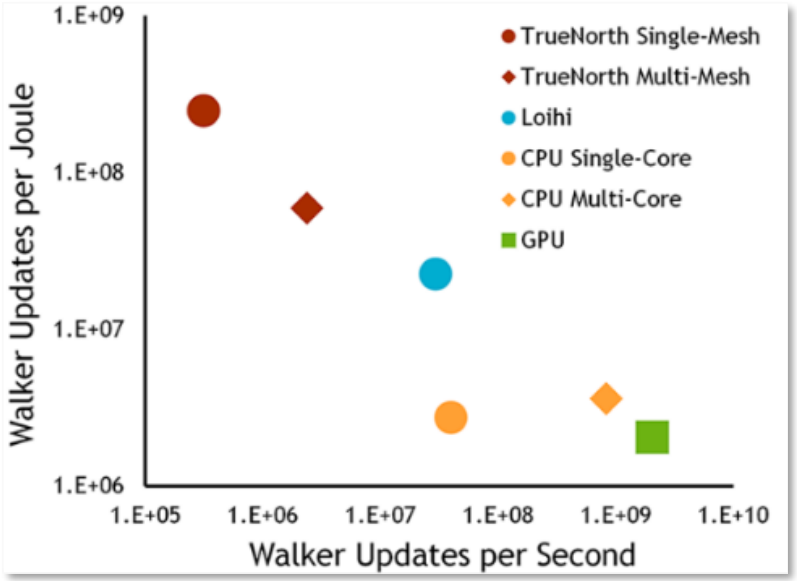
Neuromorphic hardware can simulate Monte Carlo random walks more efficiently than CPUs / GPUs



Leaky Integrate and Fire Neuron

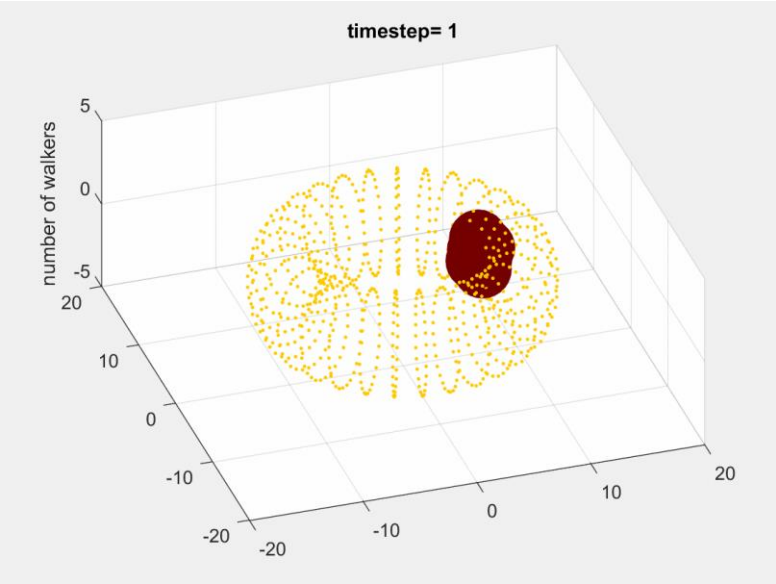


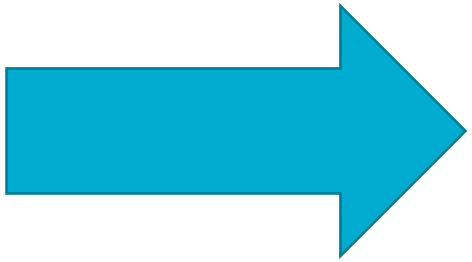


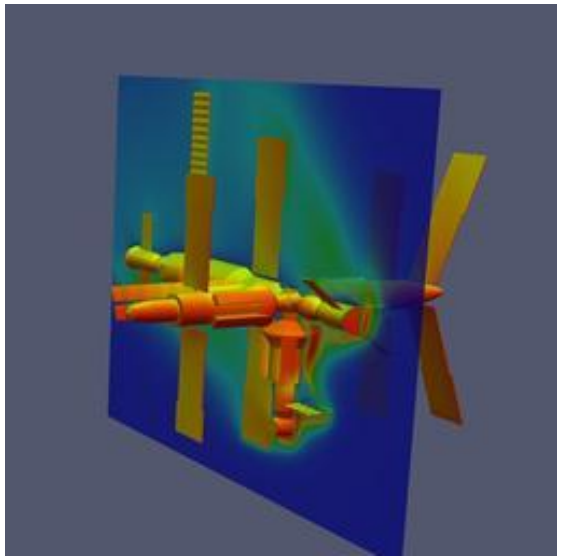


Smith et al., "Neuromorphic Scaling Advantages for energy-efficient random walk computations" Nature Electronics 2022

Will this translate to real world impact?







Random walks on neuromorphic (Smith et al., 2022)

- Brownian motion
- 1000's of particles
- 100's of cells
- 100's of timesteps
- 1 neuromorphic chip

SPARTA simulation of Mir space station (Michael Gallis, Sandia)

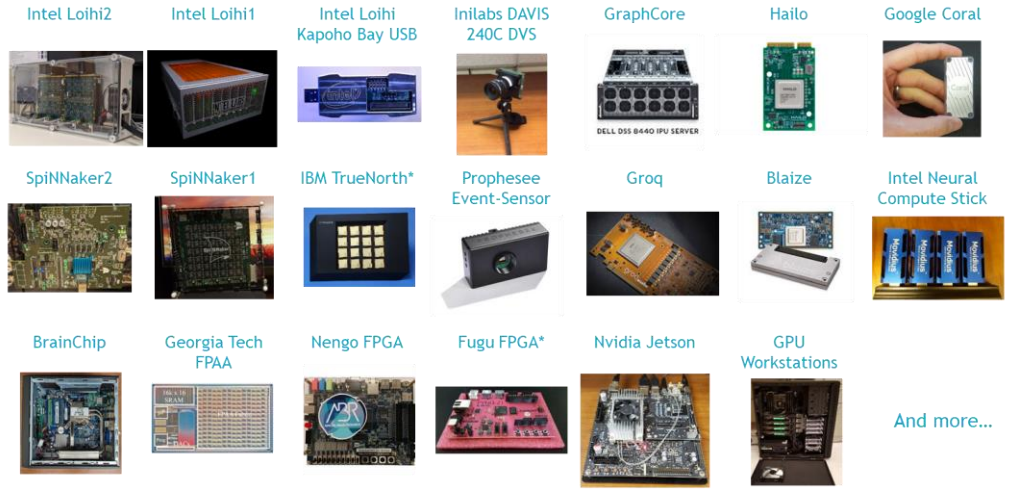
- Gas physics
- 1.6 Billion particles
- 10 million cells
- 500,000 timesteps
- 2048 Xeon cores

Neural Exploration & Research Lab (NERL)

Enables researchers to explore the boundaries of neural computation

- The research conducted in the lab evaluates what is possible with neural hardware and software for national security benefit and the advancement of basic research

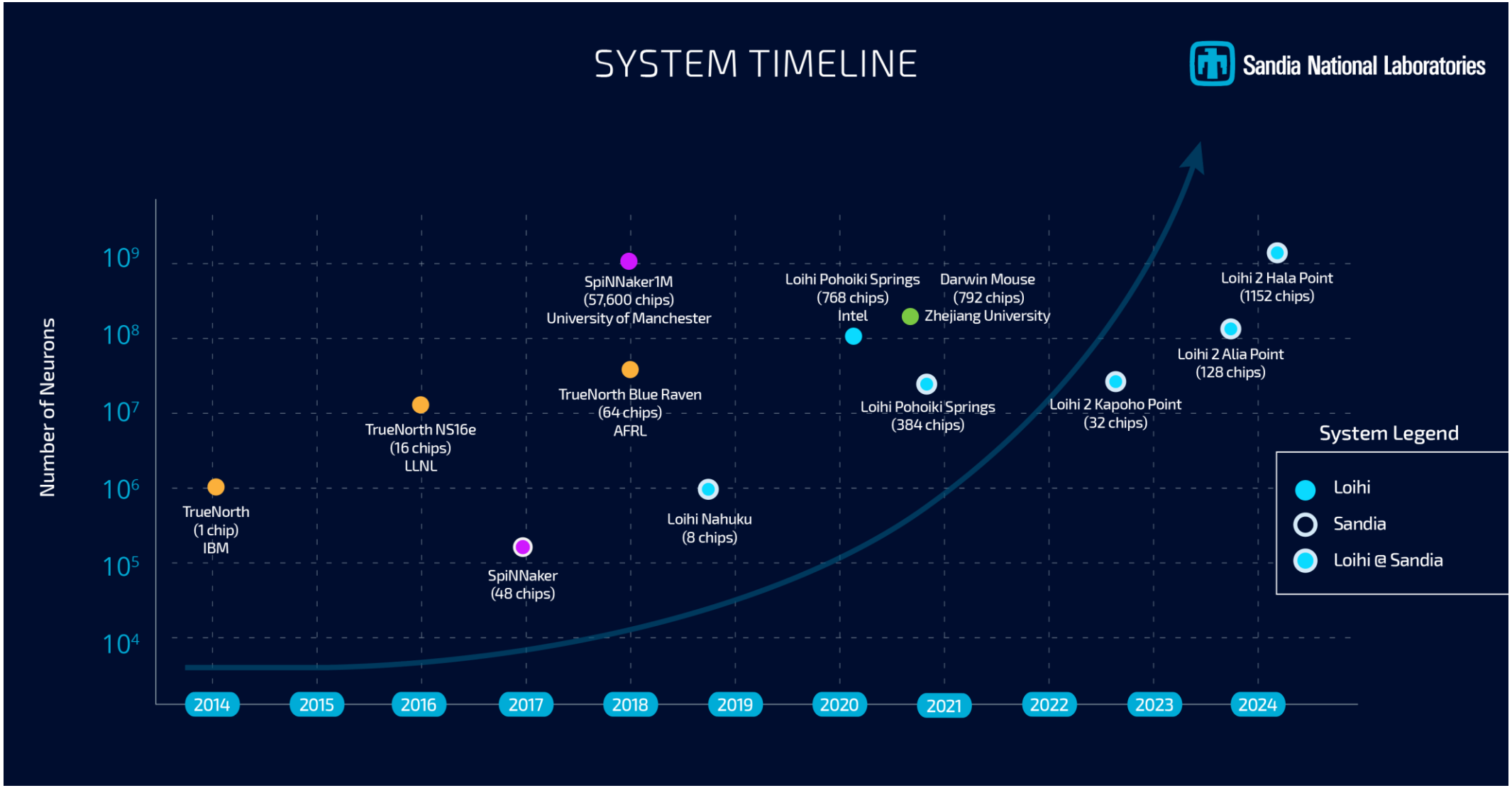
Consists of a variety of neuromorphic hardware & neural algorithms providing a testbed facility for comparative benchmarking and new architecture exploration



https://neuroscience.sandia.gov/neural-exploration-and-research-laboratory-nerl/

Not an exhaustive list

Large-Scale Neuromorphic Systems



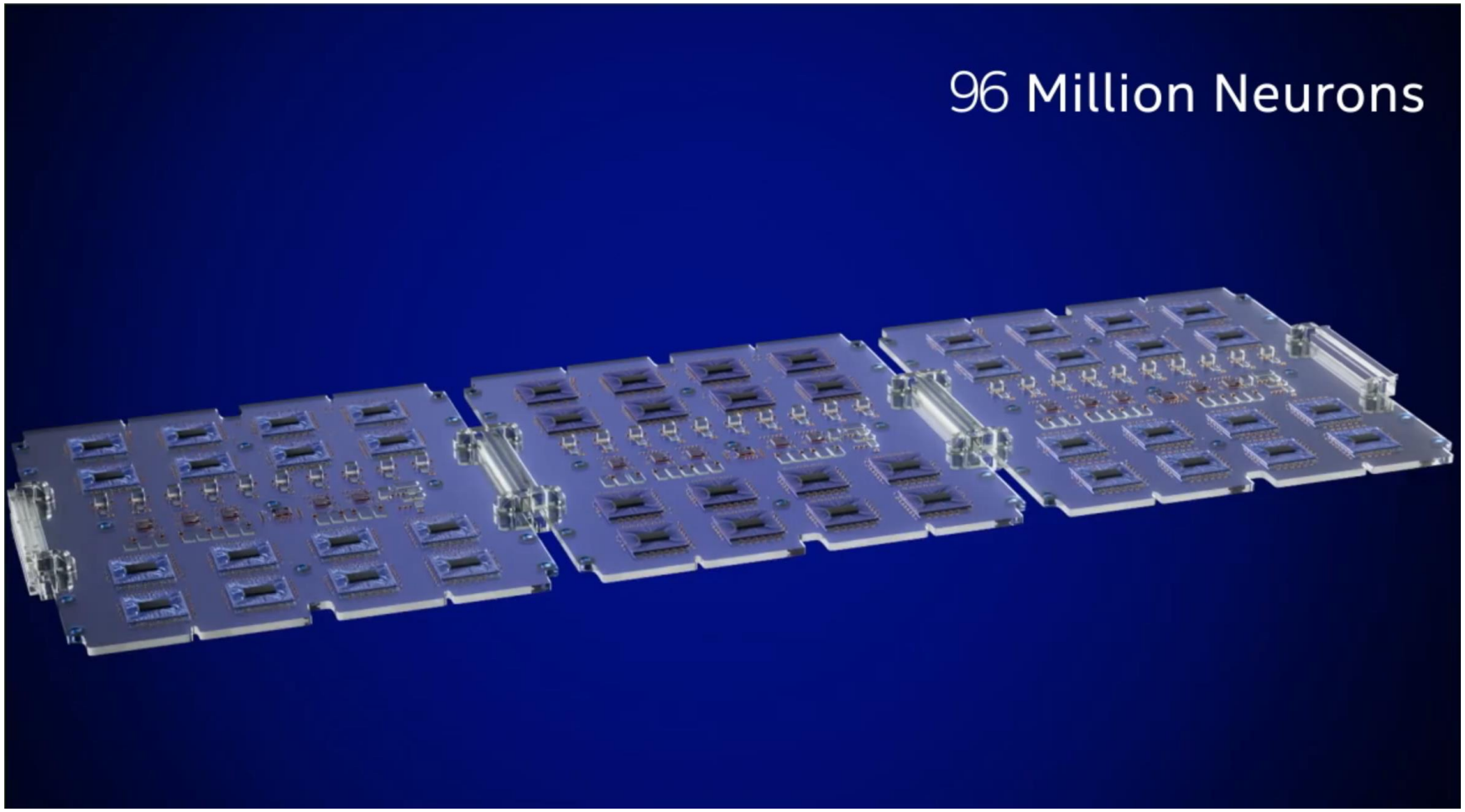
Sandia Labs & Intel - Hala Point

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point



Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Sandia Labs & Intel - Hala Point

Source: https://www.intel.com/content/www/us/en/newsroom/news/intel-builds-worlds-largest-neuromorphic-system.html#gs.84527k

Hala Point Specs & Performance

System

- 1152 Loihi 2 chips
- 140,544 neuromorphic cores
- 2,304 x86 cores
- 6U data center chassis
- 2600 Watts power (max)

Capacity

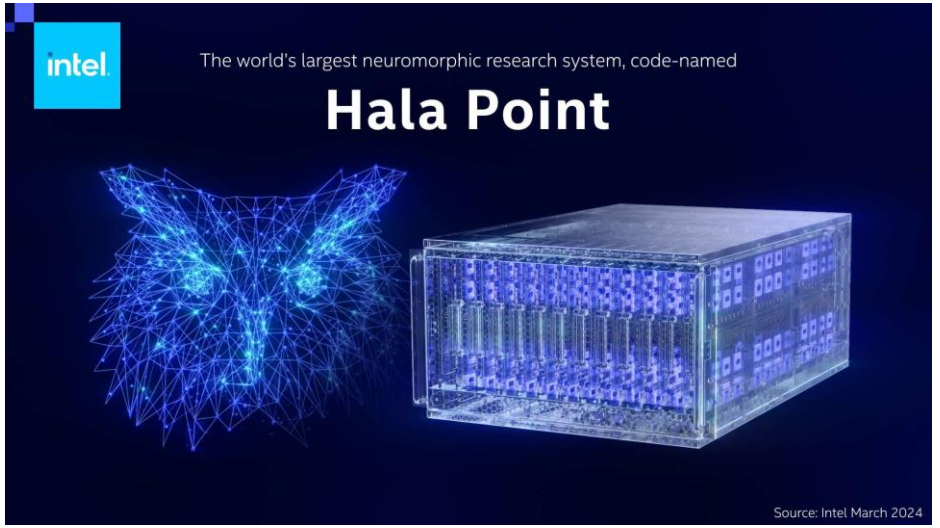
- 1.15 billion neurons
- 128 billion synapses

Speed

- 380 trillion synaptic ops/second
- 240 trillion neuron ops/second
- 16 petabytes/sec memory bandwidth
- 3.5 PB/s inter-core communication bandwidth
- 5 TB/s inter-chip communication bandwidth

Performance Characterization

- Up to 20 quadrillion operations per second (or 20 petaops)
- 15 trillion 8-bit operations per second per watt (TOPS/W)
  - 10:1 sparse connectivity & event-driven activity via sigma-delta neuron model
  - MLP network with 14,784 layers; 2048 neurons/layer, 8-bit weights; random-noise activity



Source: https://www.intel.com/content/dam/www/central-libraries/us/en/images/2024-04/newsroom-intel-hala-point-owl.jpg.rendition.intel.web.1920.1080.jpg

Sandia Labs & SpiNNcloud (SpiNNaker2)

Spiking Neural Network Architecture (SpiNNaker)

- Each SpiNNaker2 chip contains a low-power mesh of 152 Arm-based cores + accelerators
  - Globally-asynchronous-locally-synchronous operation and dynamic voltage regulation for energy efficiency
  - Event-driven mesh communication
- Designed to boost neuromorphic, hybrid, and mainstream AI model computations
- Server board consists of 48 SpiNNaker2 chips
- Large-scale systems with 90 boards for billions of neurons

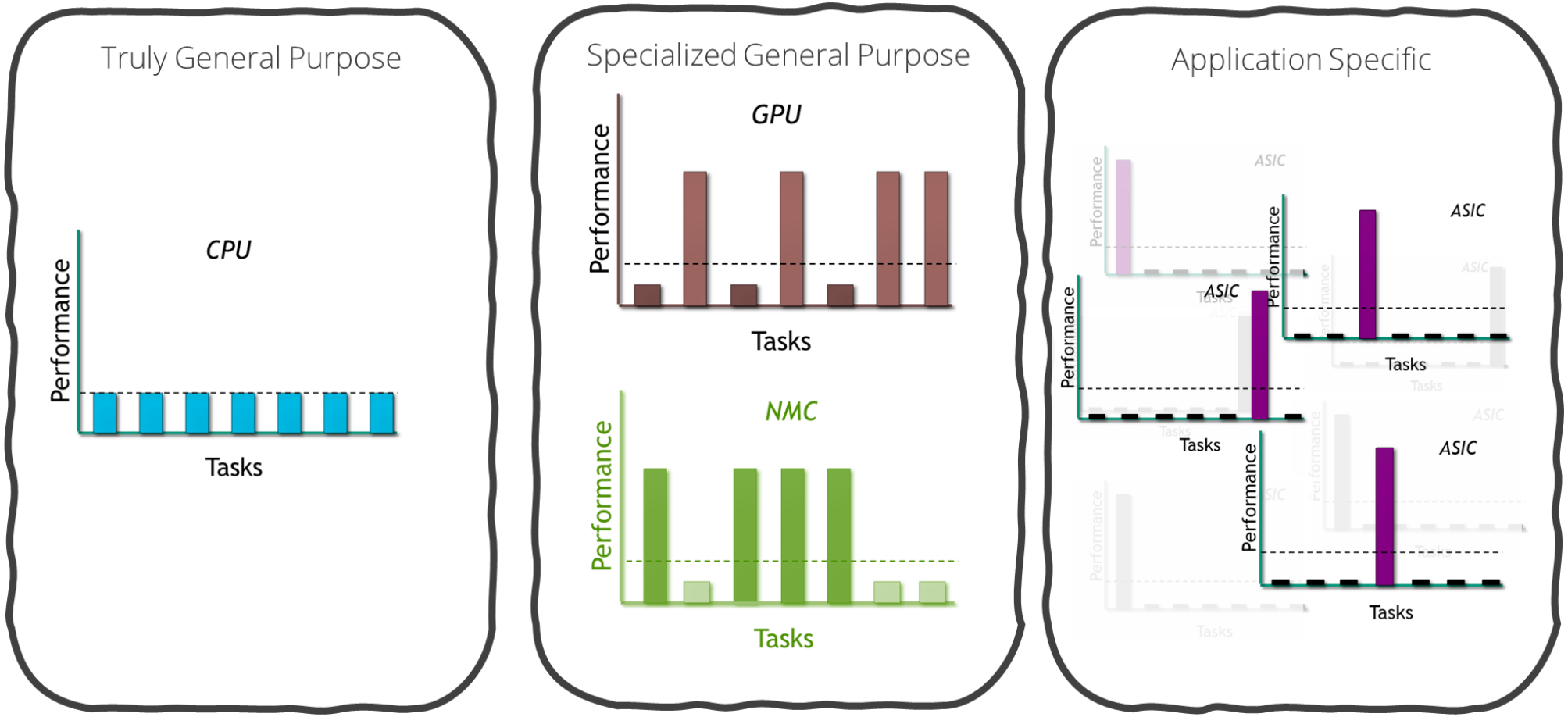
Read more:
https://spinncloud.com/?utm_source=ARM&utm_medium=referral&utm_campaign=CollaborationCampaign&utm_content=blogpost&p=5057

https://www.dnn.de/lokales/dresden/ki-supercomputer-an-der-tu-dresden-58b93825-75fd-46ae-846a-da4e78c9c3e0.html

Conclusions

Future of Neuromorphic?

Neuromorphic is likely similar to GPUs in degree of specialization



Exciting research exploring -
Which applications? How? When?

Thank You!

Neural Exploration & Research Lab (NERL)

https://neuroscience.sandia.gov/

Full Stack NMC Brochure

https://ip.sandia.gov/opportunity/full-stack-neuromorphic/

Fugu

https://github.com/sandialabs/Fugu

Simulation Tool for Asynchronous Cortical Streams (STACS)

https://github.com/sandialabs/STACS

Neurons to Algorithms (N2A)

https://github.com/sandialabs/n2a

Questions?

Backup

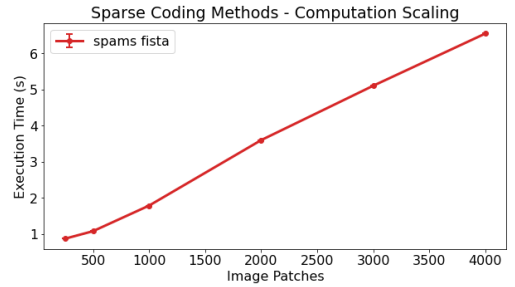
Neural Sparse Coding

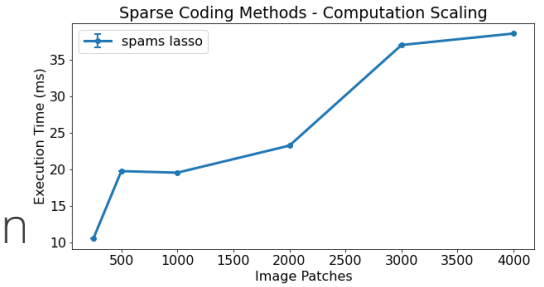
Sparse Coding or Sparse Dictionary Learning

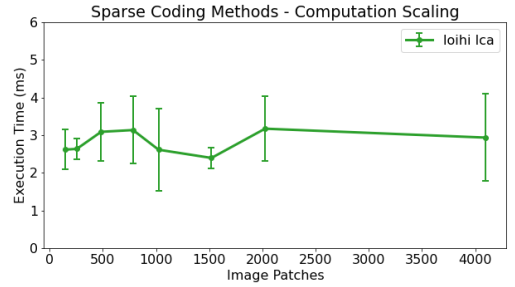
- Method of modeling data by decomposing it into sparse linear combinations of elements of a given overcomplete basis set
- On neuromorphic, the LASSO (least absolute shrinkage and selection operator) computation for sparse coding can be approximated with the spike-based algorithm LCA (locally competitive algorithm)
  - Implemented as rate-coded neurons with inhibitory connections between competing dictionary elements

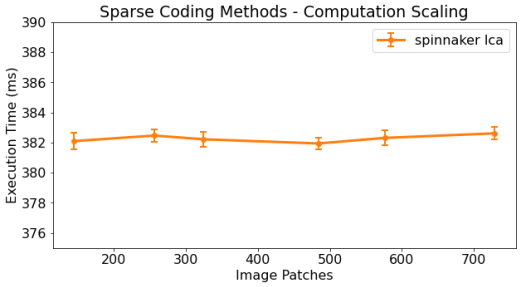
Parameterization	Size of image, Size of image patch, Size of the dictionary, Stride of image patch, Desired sparsity
Scaling	Problem size via # of image patches, Parameters
Metrics	Time for setup, Time for reconstruction, Reconstruction performance, Reconstruction sparsity, Compute resource usage, Energy resource usage

Example Results -









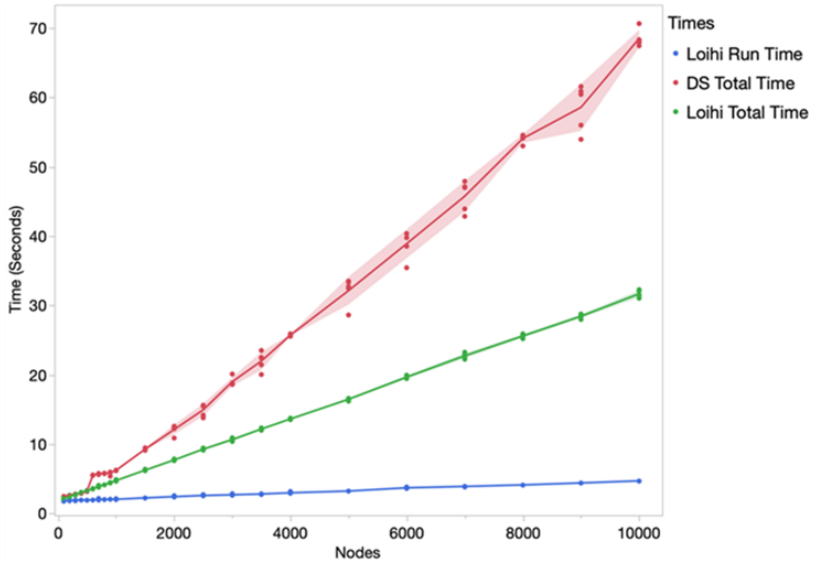
Neural Graph Analysis

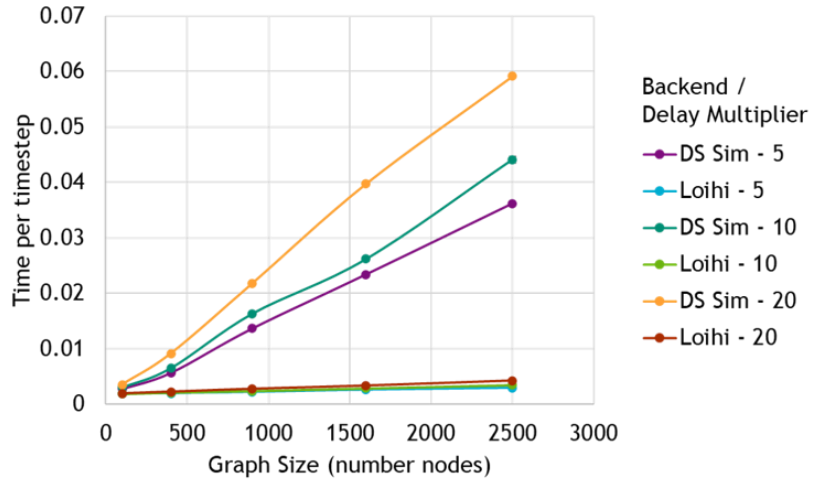
Single Source Shortest Path (SSSP)

- Between a source and target node, what is the shortest path (and path length) that connects the two
- SNN is straightforward – each vertex in the source graph is a neuron, each edge is a synapse between neurons, & graph weights equate to delays
  - The source neuron receives input driving it to spike send ensuing spikes through the SNN
  - Shortest path length is determined when the target spikes & monitoring edges can yield the path

Parameterization	Graph generation (uniformly random tree, small world), Nodes, Weight range, Max runtime, Source, Target
Scaling	Graph scale, Weight/delay range
Metrics	Total time, Time for setup

Example Results -





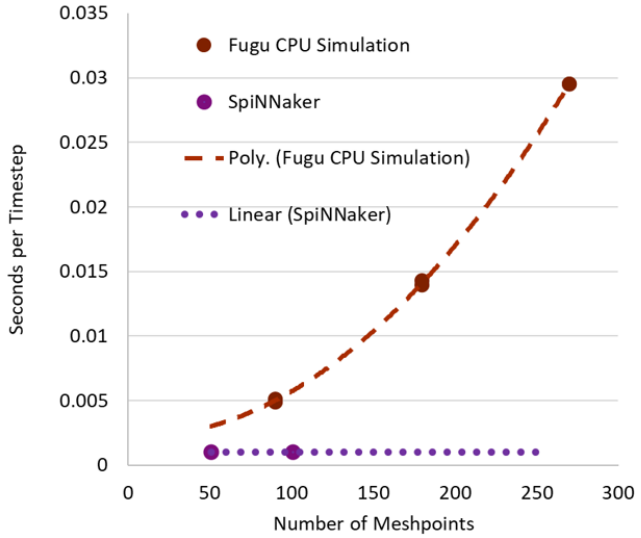
Neural Random Walk

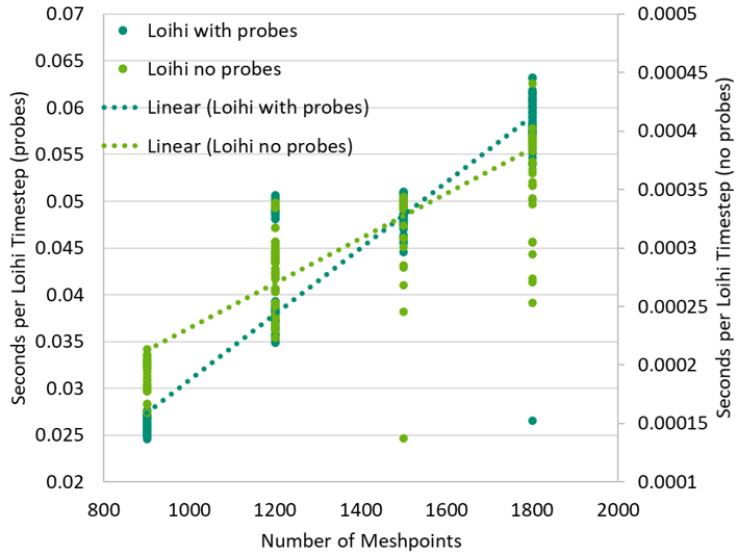
Discrete time Markov Chain (DTMC)

- Particle Angular Fluence: the time-integrated flux of particles traveling through media given as a function of position and velocity
- Particles travel at a constant speed and experience relative velocity scattering over a small region of space
- Conventional approach models walkers & tracks states – neuromorphic models state & tracks walkers

Parameterization	Number of total walkers, Size of direction/relative velocity/angular discretization, Time step size of simulation, Size of the state space, Size of positional discretization
Scaling	Walkers, Mesh size
Metrics	Energy cost of walkers, Time to run, Space to run

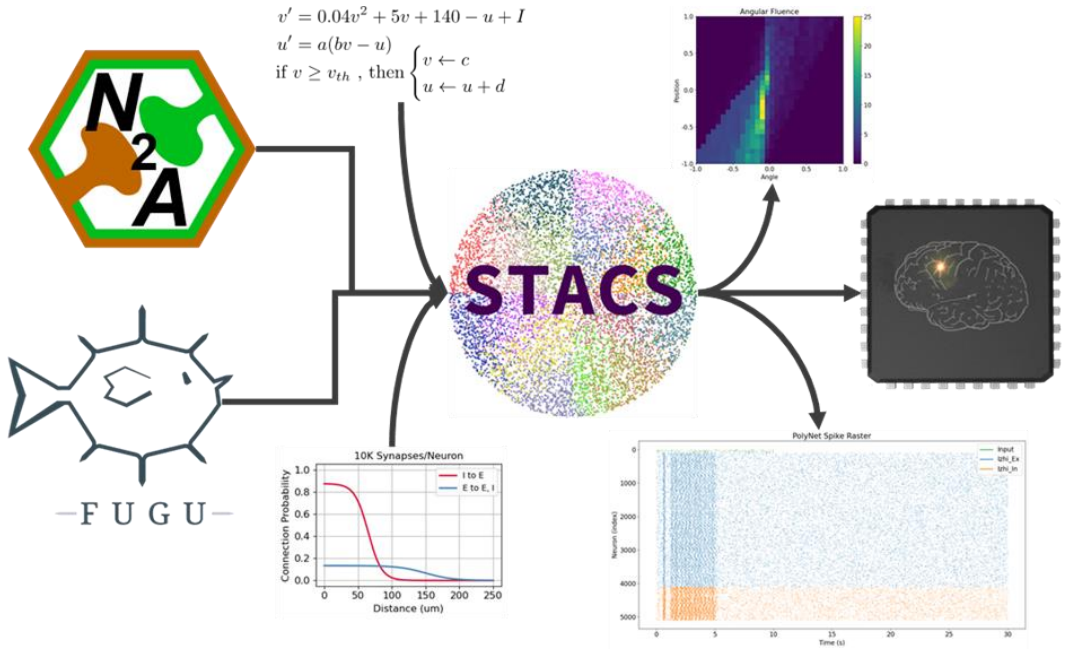
Example Results -





Simulation Tool for Asynchronous Cortical Streams (STACS)

- Large-scale spiking neural network simulator built on top of the Charm++ parallel programming framework



Description Languages
- PyNN, NeuroML, NineML, etc.

Software Frameworks
- N2A, Fugu, Lava, Nengo, etc.

Able to interoperate with software frameworks through:
- Translating between network description languages
- As a simulation backend

Network Simulators
- NEST, NEURON, Brian, GeNN, etc.

STACS is primarily a spiking neural network simulator

Data Formats
- NIR, SONATA, NetworkX, GEXF, etc.

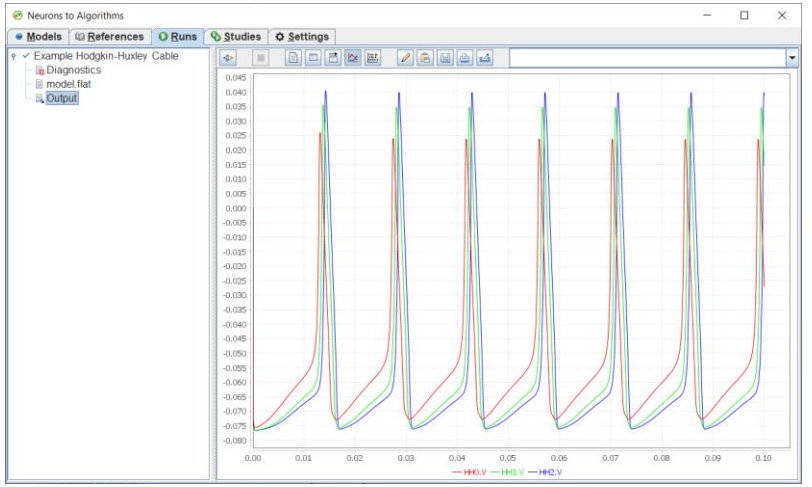
Hardware Platforms
- Loihi, SpiNNaker, BrainScaleS, etc.

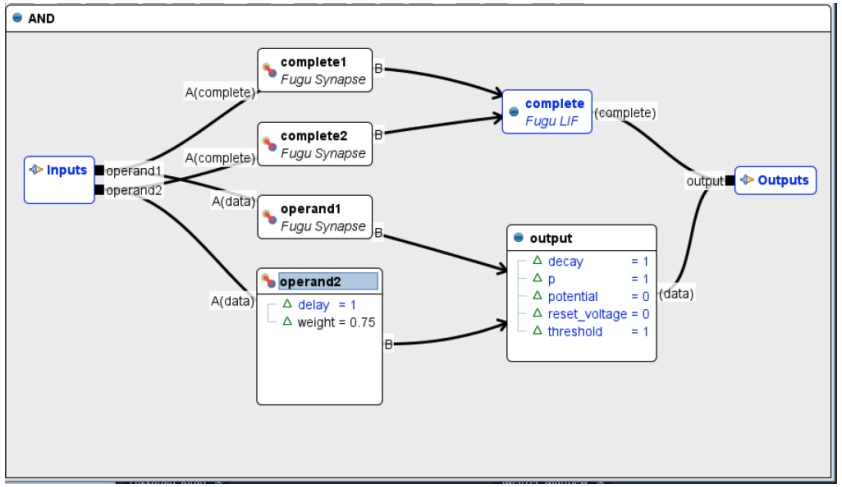
Partition-based SNN-dCSR data format supports external tool interoperability:
- Graph partitioners & network analysis
- Also enables mapping to neuromorphic hardware platforms

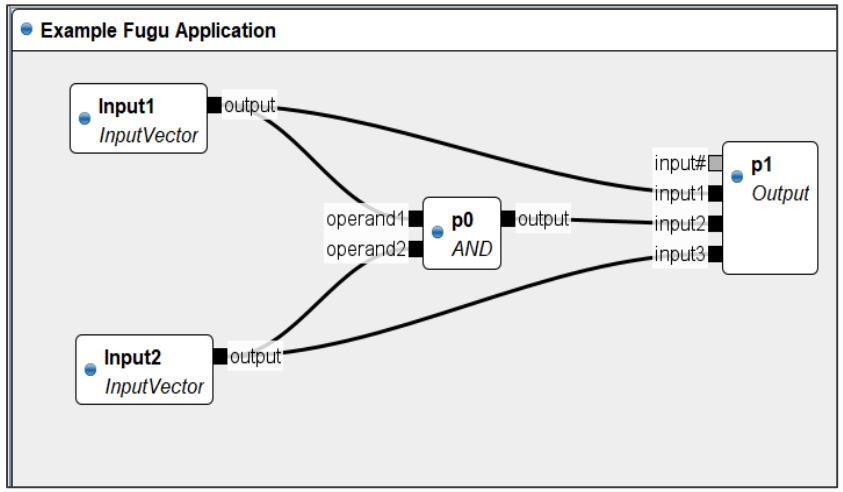
- Available at: https://github.com/sandialabs/STACS

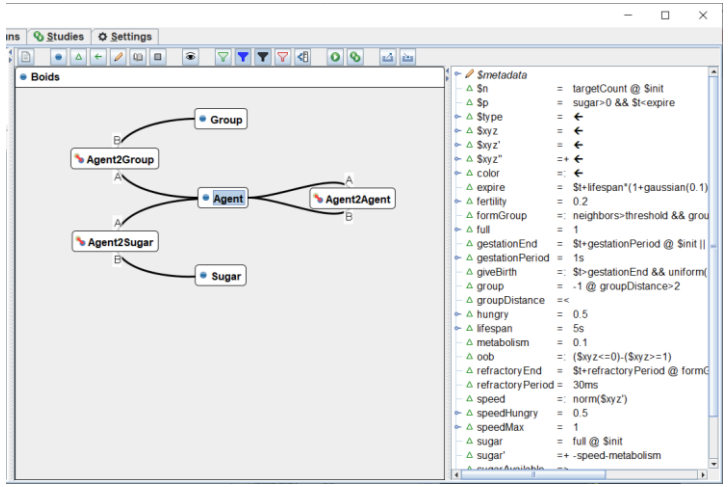
Neurons to Algorithms (N2A)

- Neural programming language and workbench
- Object-oriented, declarative language
  - Parts defined with simple set of equations
  - No need to program
  - Build complex structures from simple ones by reusing parts
  - Backends for major neuromorphic devices (work-in-progress)









Available at: https://github.com/sandialabs/n2a