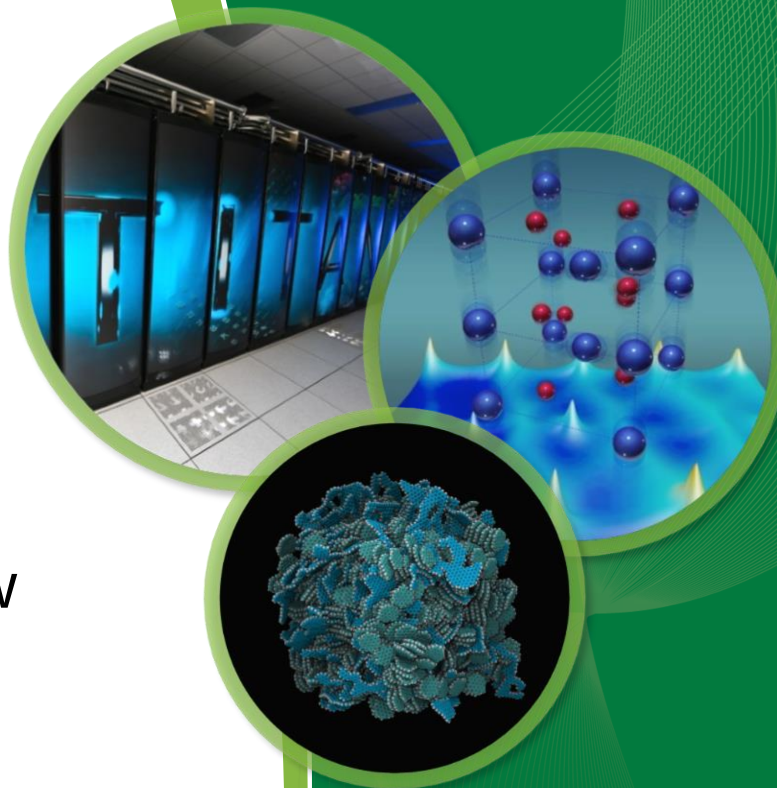
Parallel Evolutionary Optimization for Neuromorphic Network Training



Catherine Schuman

Liane Russell Early Career Fellow

Computational Data Analytics

MLHPC 2016

Salt Lake City, Utah

November 14, 2016

Neuromorphic Computing

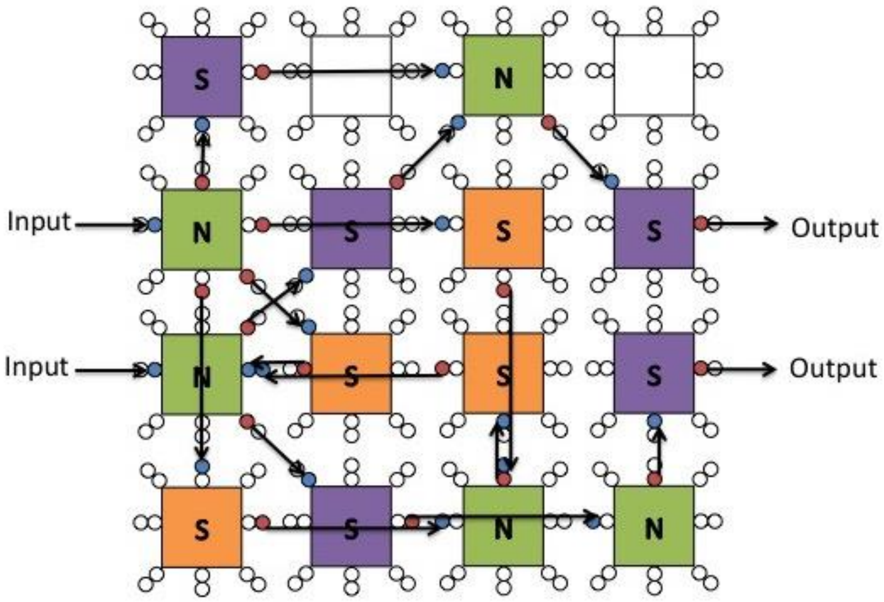
- Neuromorphic computing systems are software/hardware systems that are inspired by biological brains.
  - Neural networks in hardware.
- Goal is to capture important capabilities of the biological brain: real-time processing abilities, generalization of learned information, adaptability to changes in the environment, robustness.
- Neuromorphic hardware: improvements in power, size/portability, computation time, communication costs over neuromorphic simulations.

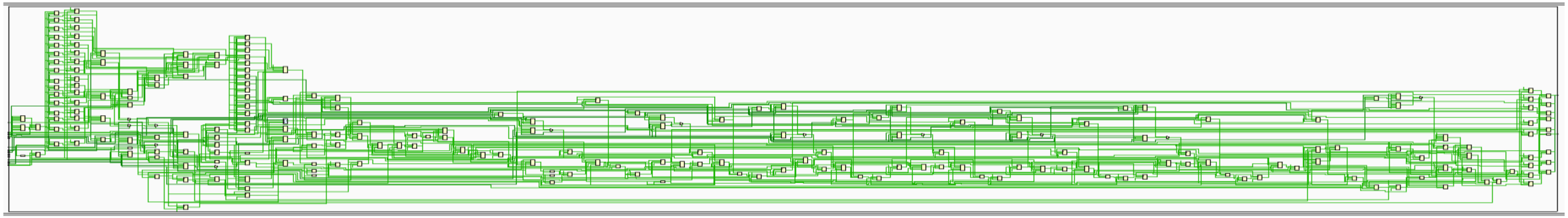
Neuromorphic Computing

- What characterizes a neuromorphic computer?
  - Many simple processor/memory structures (e.g., neurons and synapses).
  - Communication using simple messages (e.g., spikes).
  - Algorithms usually emphasize temporal interaction.
    - Messages have a time-stamp (implicit or explicit).
    - Operation is usually event-driven.

Dynamic Adaptive Neural Network Array (DANNA)

- Array of programmable neuromorphic elements.
- Elements can connect to to 16 neighbors.
- Current: FPGA.
- Future: VLSI, memristors

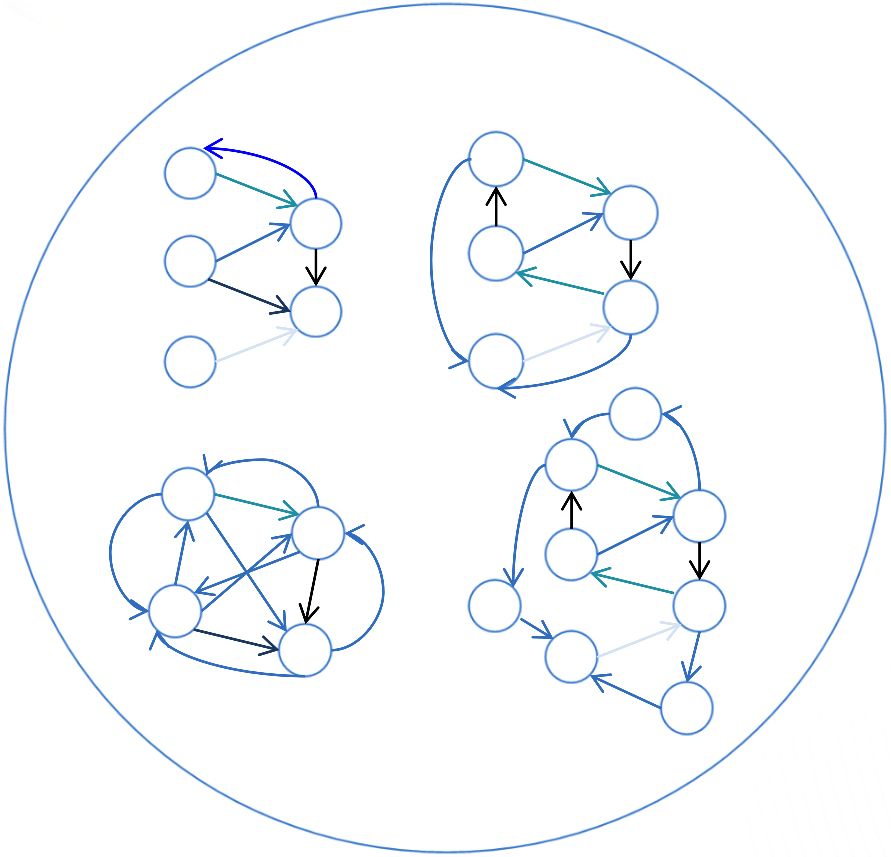




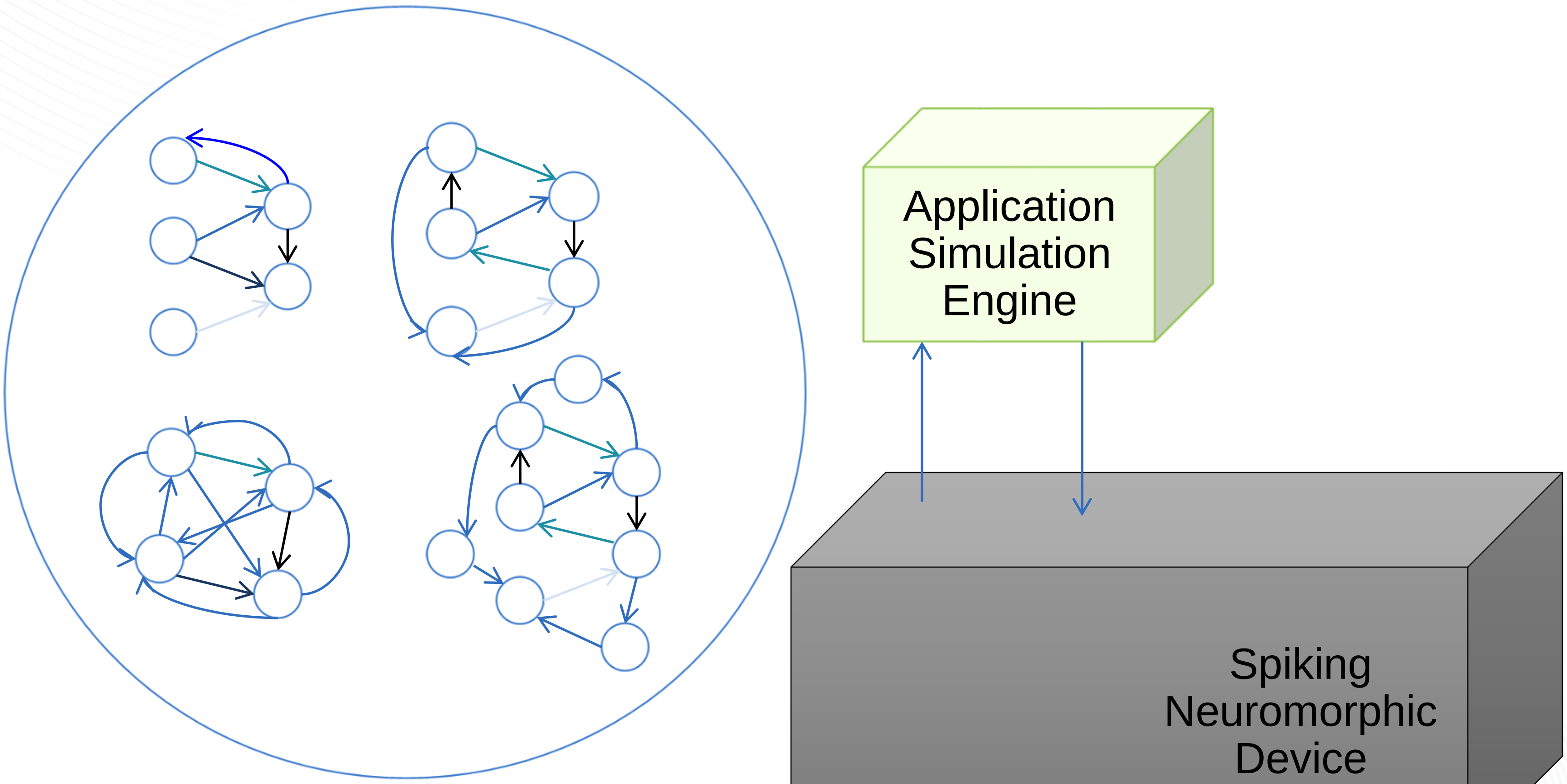
OAK RIDGE National Laboratory

How do we program neuromorphic computers?

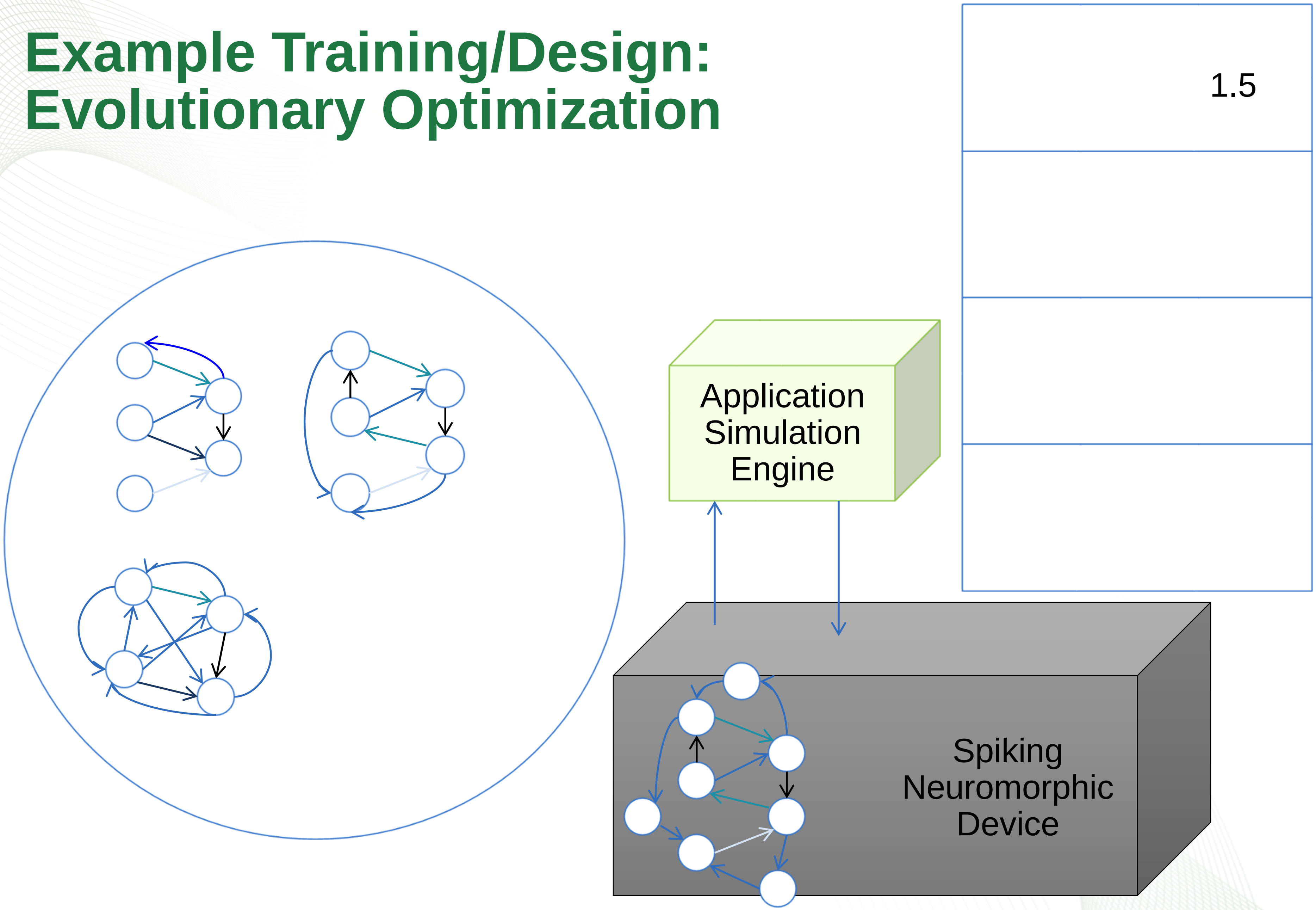
Example Training/Design: Evolutionary Optimization



Example Training/Design: Evolutionary Optimization

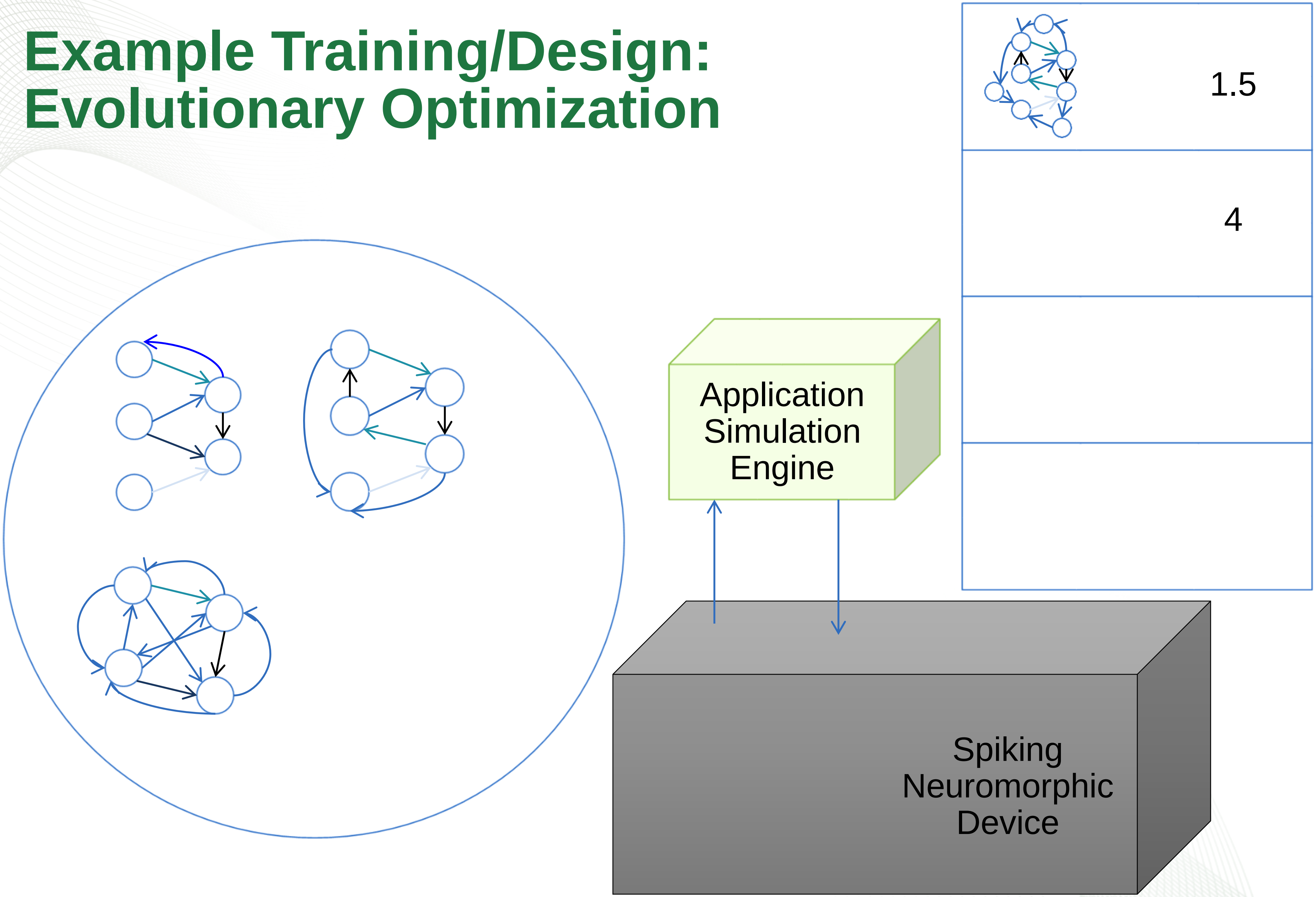


Example Training/Design: Evolutionary Optimization



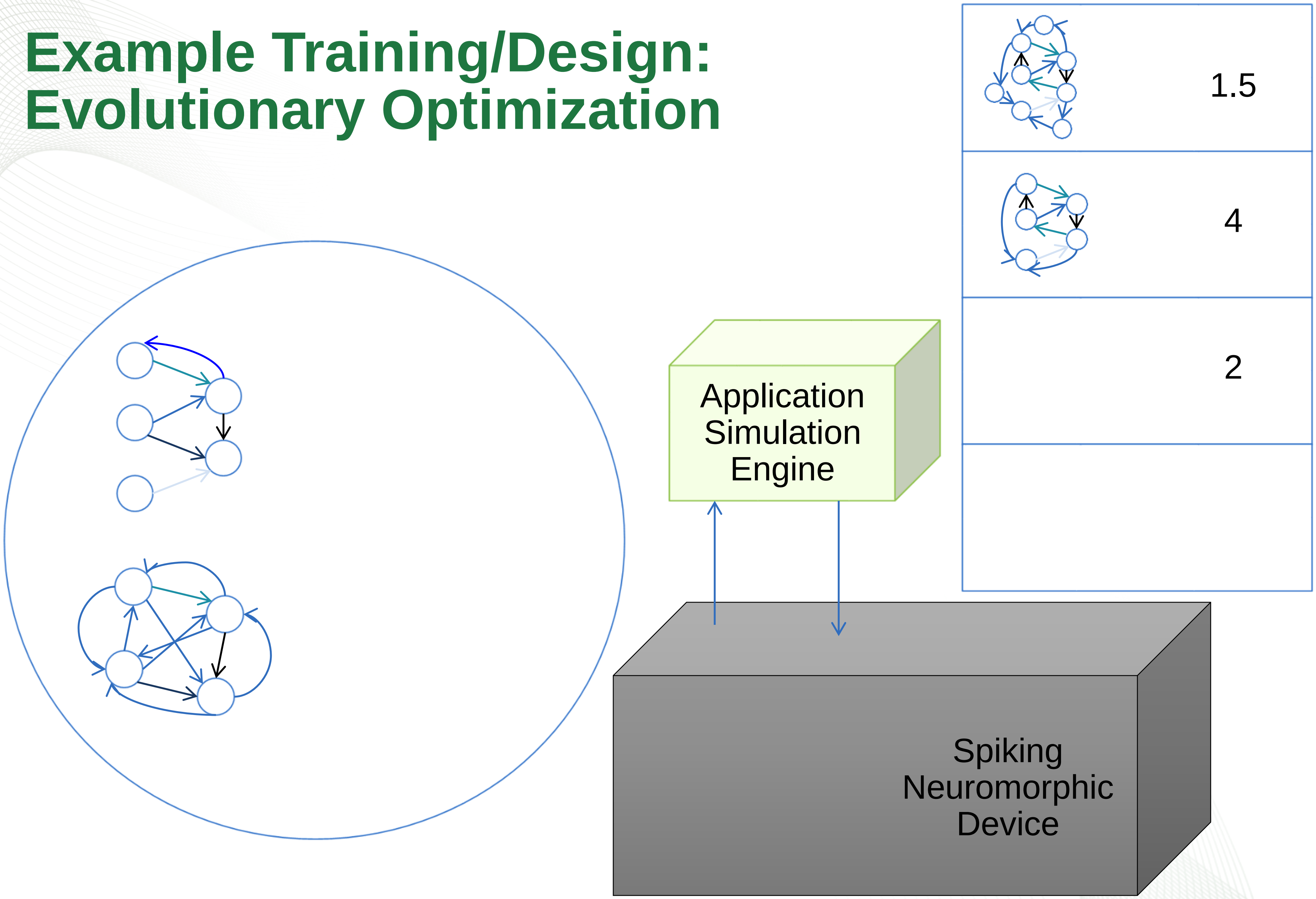
OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization

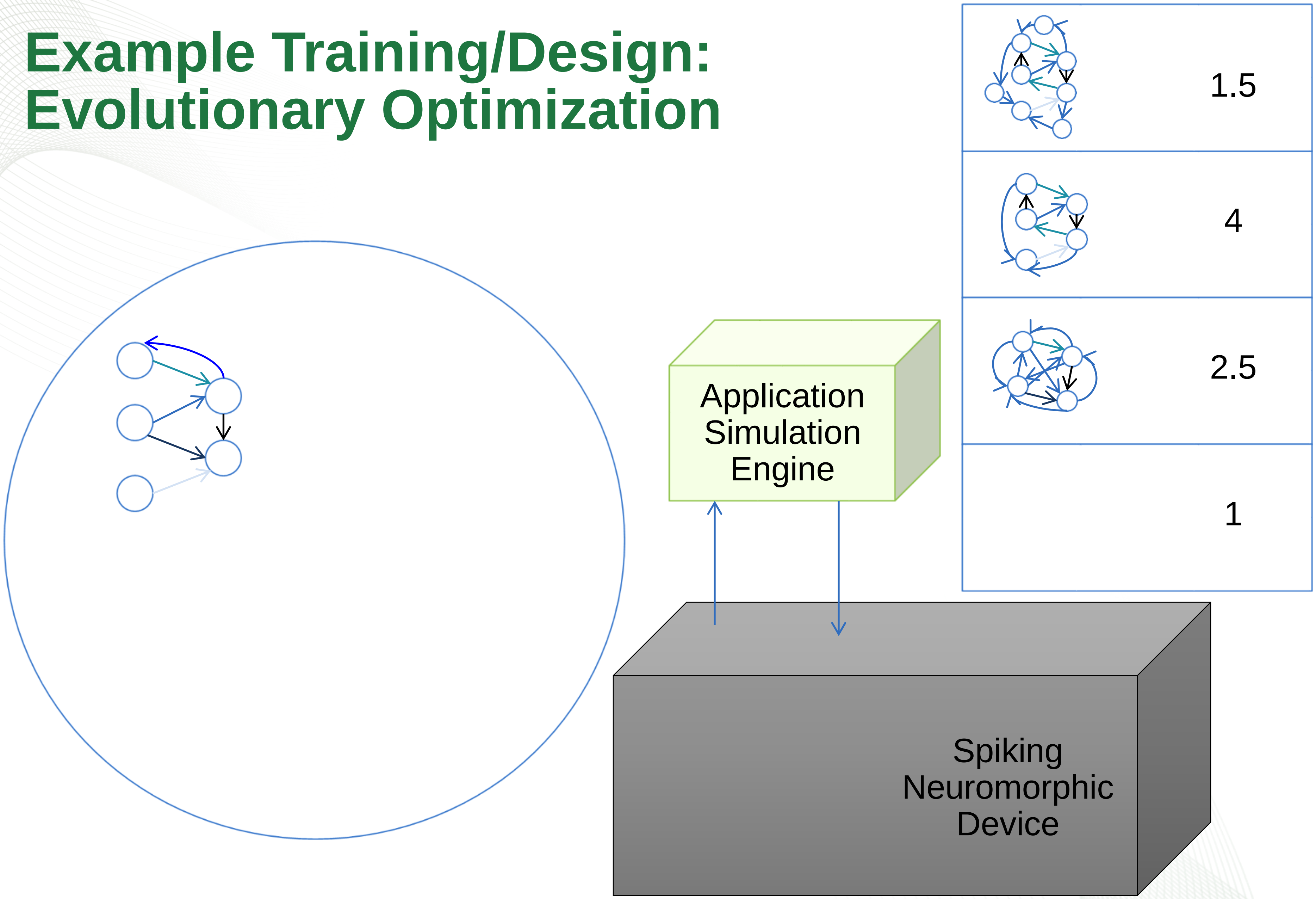


OAK RIDGE National Laboratory

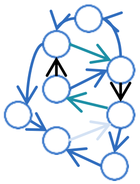
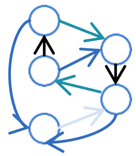
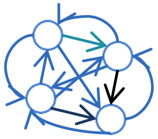
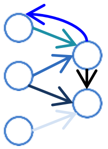
Example Training/Design: Evolutionary Optimization



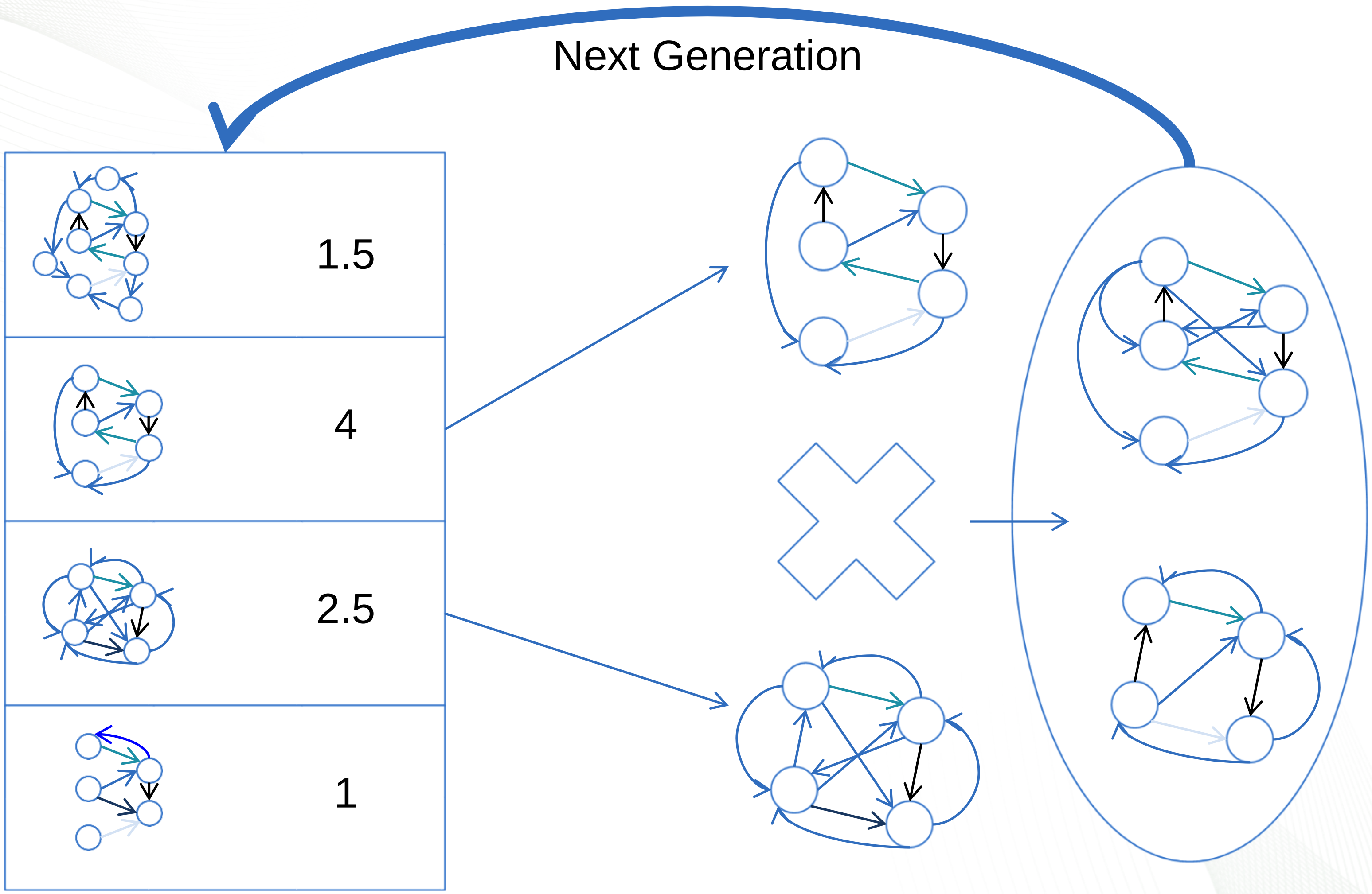
Example Training/Design: Evolutionary Optimization



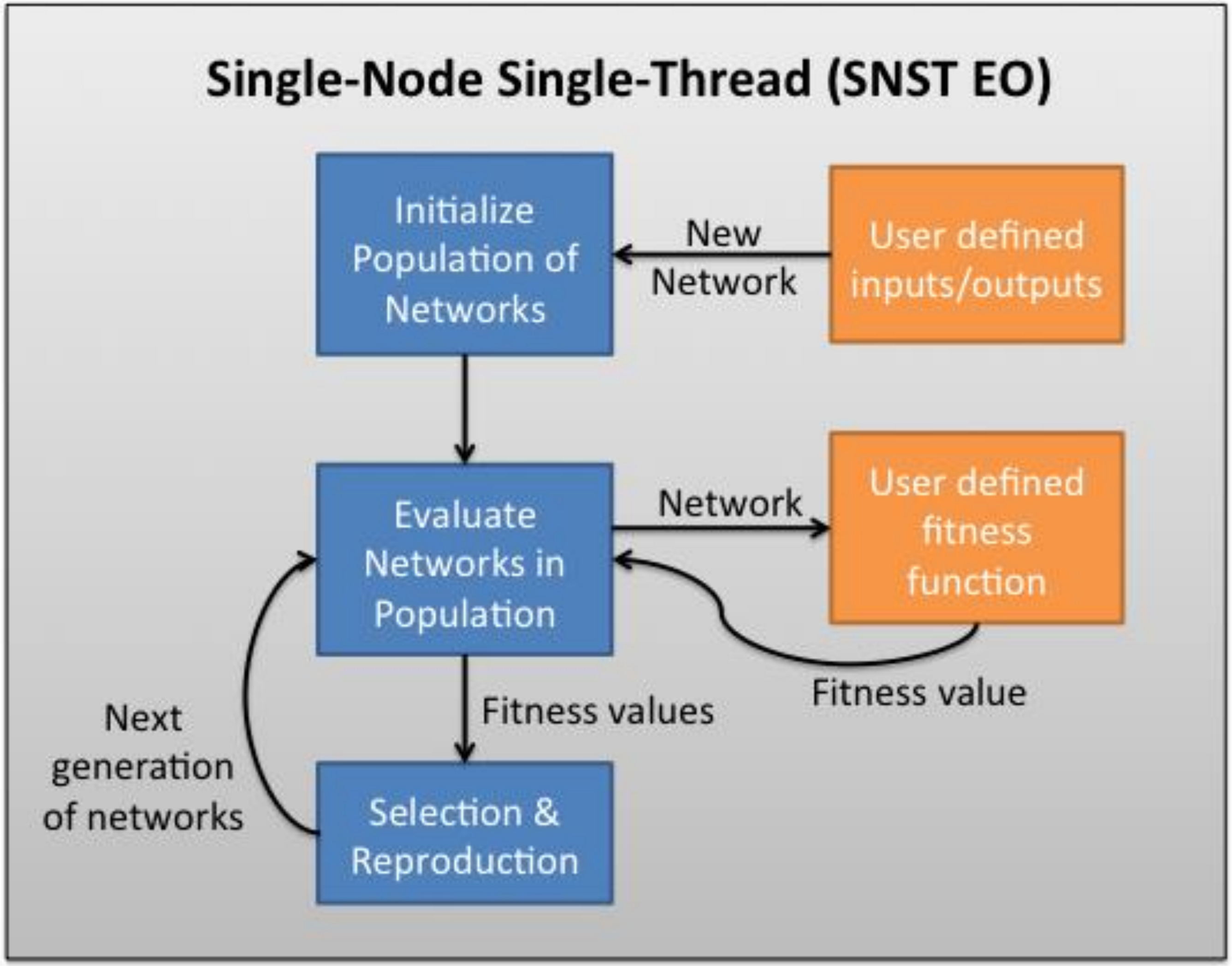
Example Training/Design: Evolutionary Optimization

	1.5
	4
	2.5
	1

Example Training/Design: Evolutionary Optimization



Single Node, Single Thread (SNST EO)



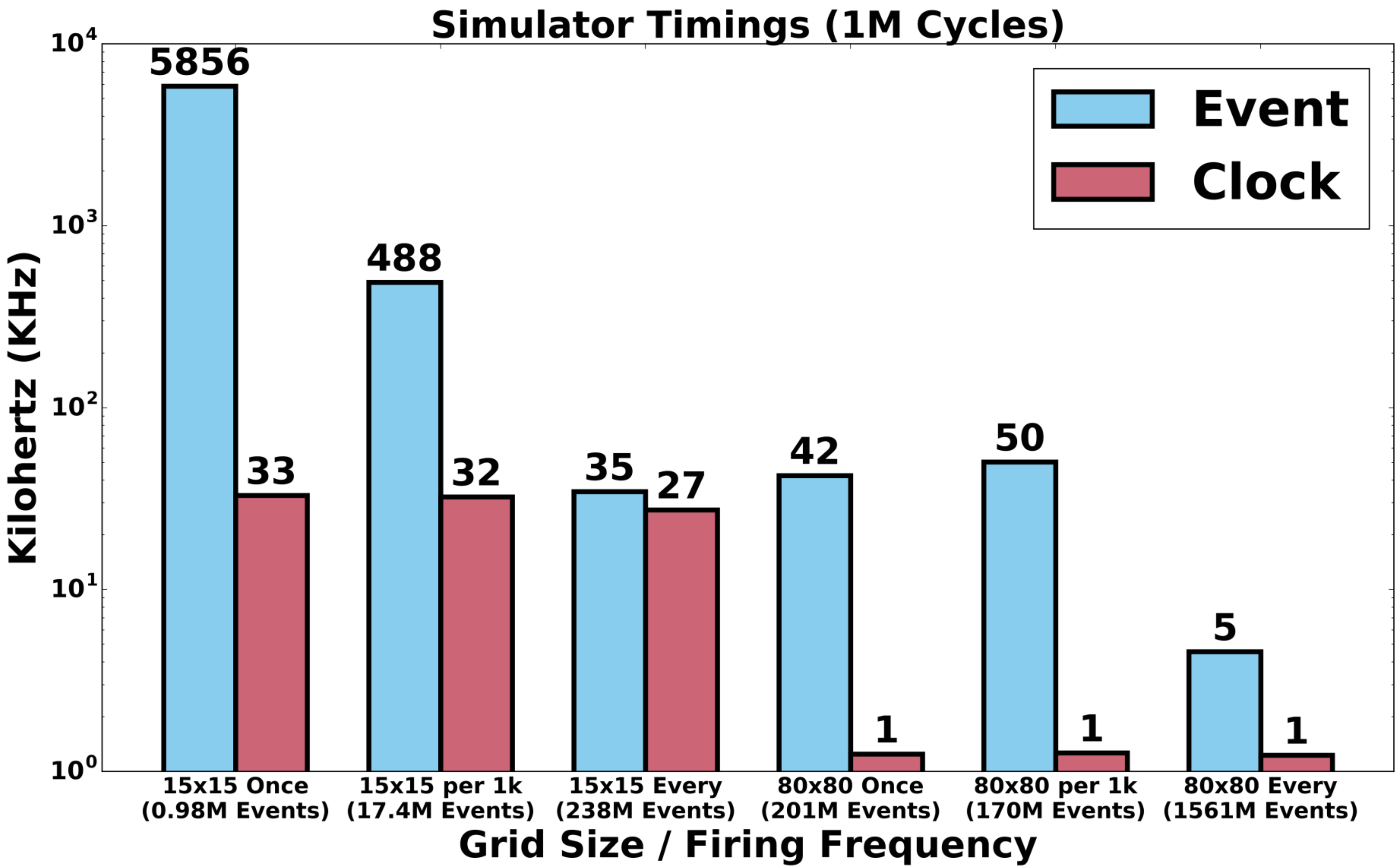
Improving EO Performance

- Fitness evaluation is a bottleneck:
  - Optimize neuromorphic simulator.
  - Parallelize fitness evaluation.
- Solution space for network solutions is large for complex problems:
  - Larger population sizes can allow for better exploration of the search space, leading to solutions more quickly.
  - Subpopulations with communication allow for diversity, but also knowledge sharing.

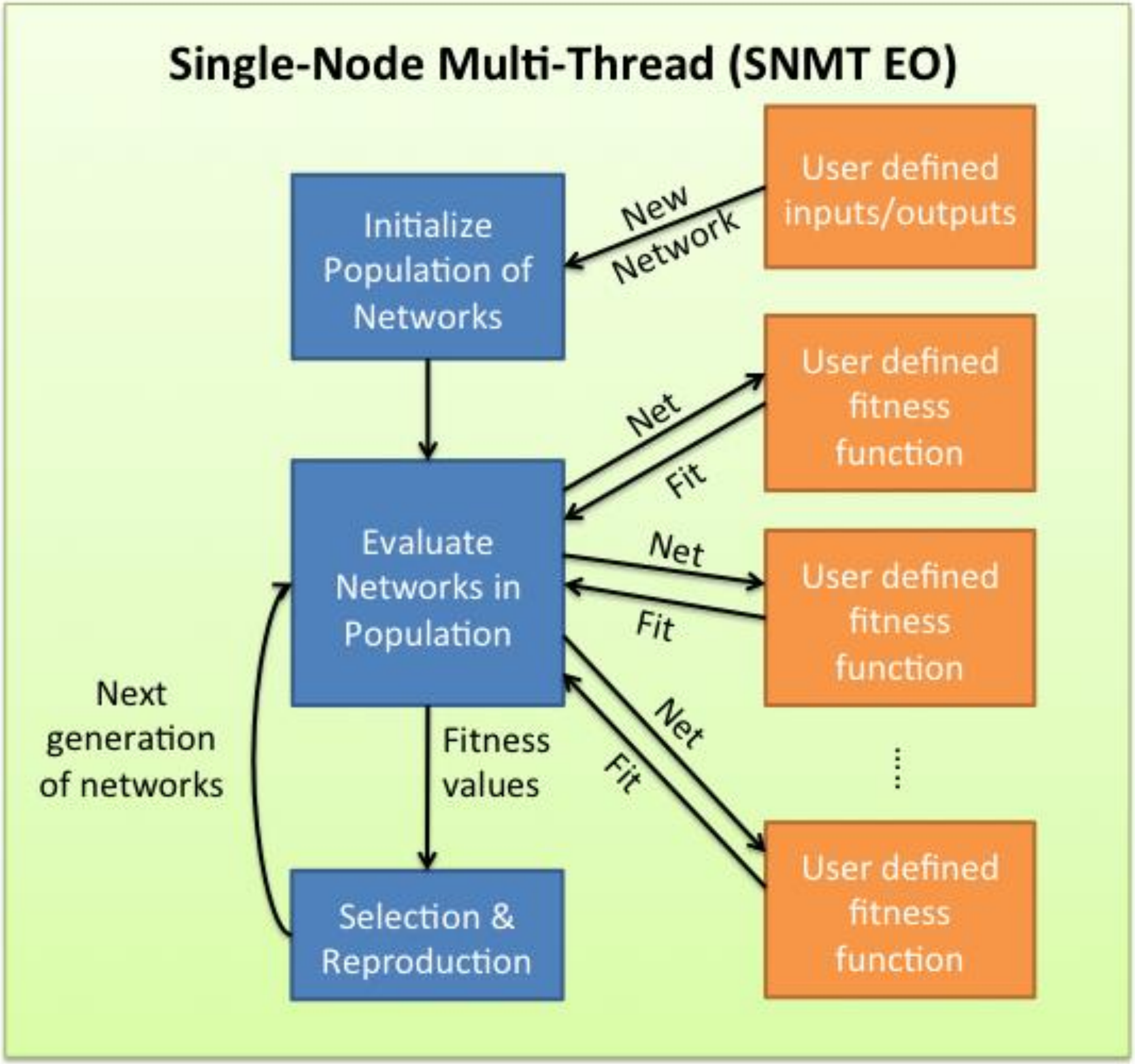
Improving EO Performance

- **Fitness evaluation is a bottleneck:**
  - Optimize neuromorphic simulator.
  - Parallelize fitness evaluation.
- Solution space for network solutions is large for complex problems.
  - Larger population sizes can allow for better exploration of the search space, leading to solutions more quickly.
  - Subpopulations with communication allow for diversity, but also knowledge sharing.

Simulator Performance



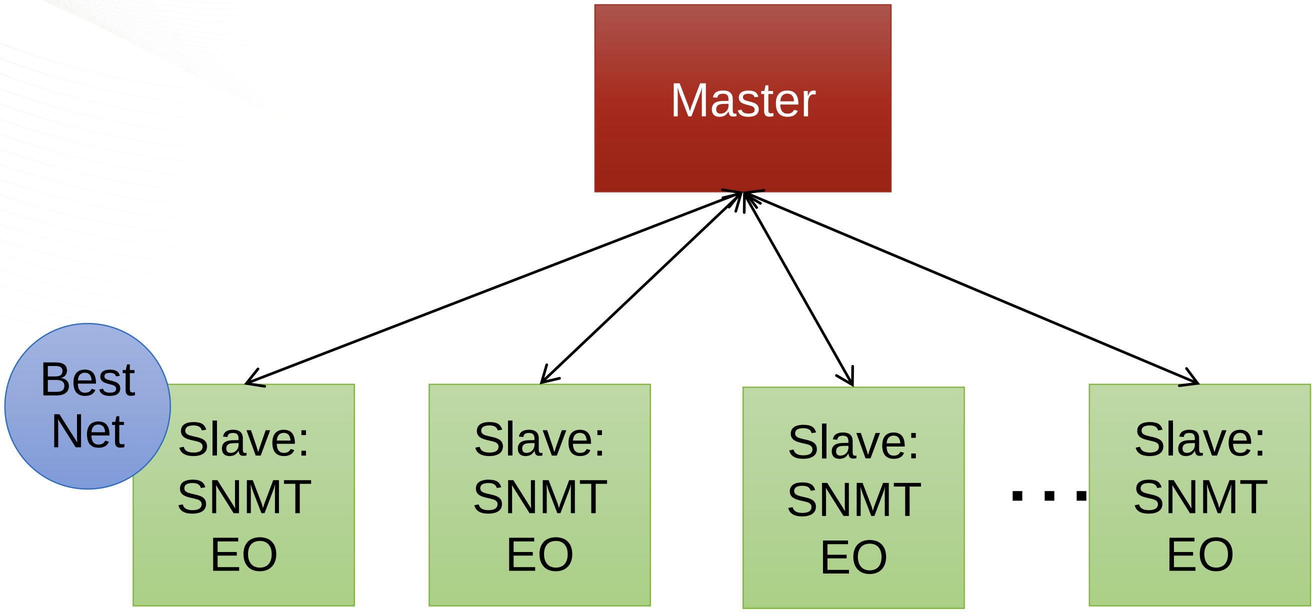
Single-Node Multi-Thread (SNMT EO)



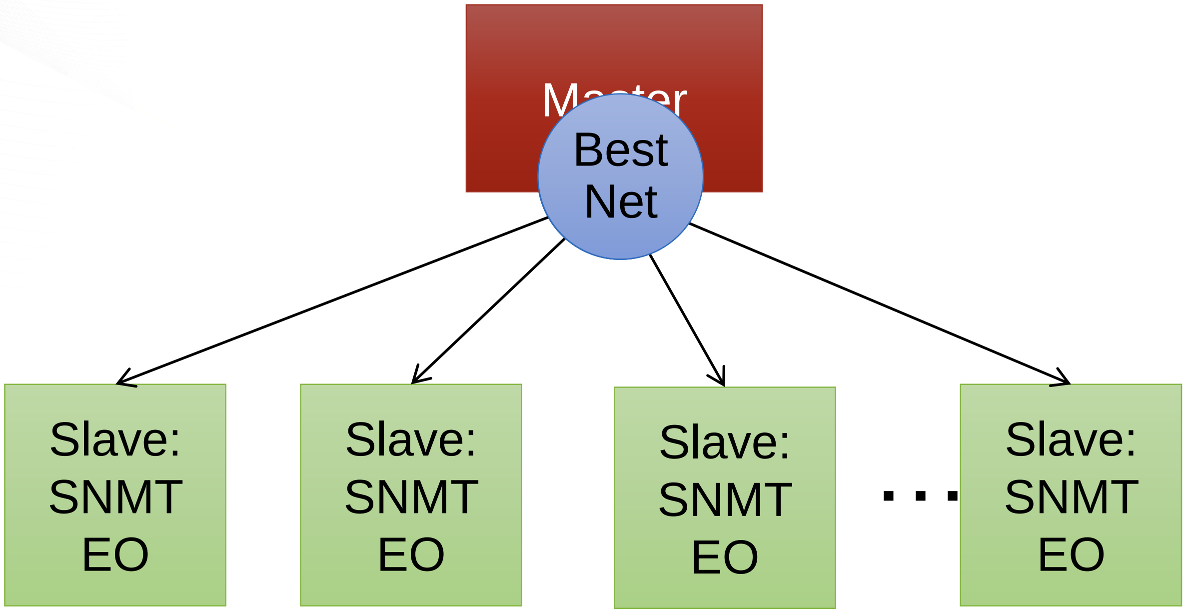
Improving EO Performance

- Fitness evaluation is a bottleneck:
  - Optimize neuromorphic simulator.
  - Parallelize fitness evaluation.
- **Solution space for network solutions is large for complex problems.**
  - Larger population sizes can allow for better exploration of the search space, leading to solutions more quickly.
  - Subpopulations with communication allow for diversity, but also knowledge sharing.

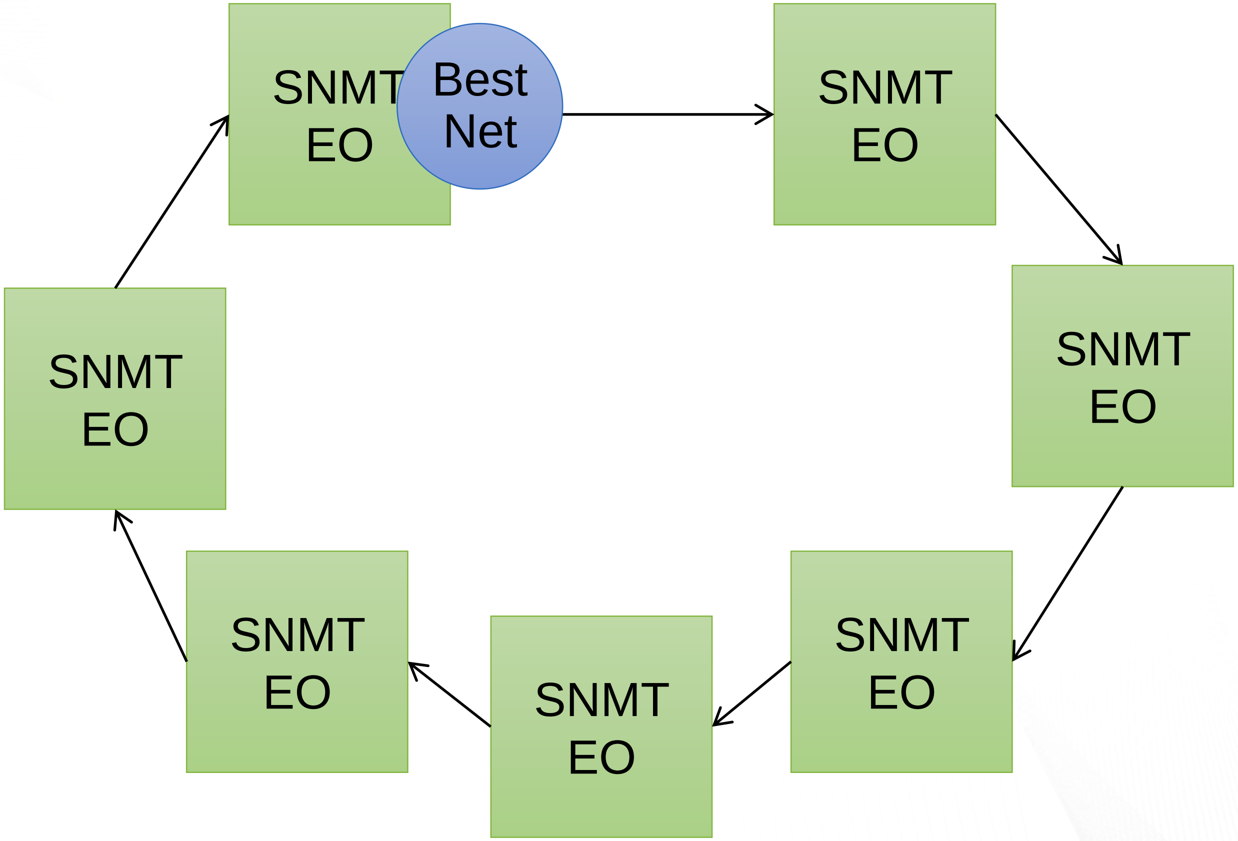
Multi-Node: Master-Slave



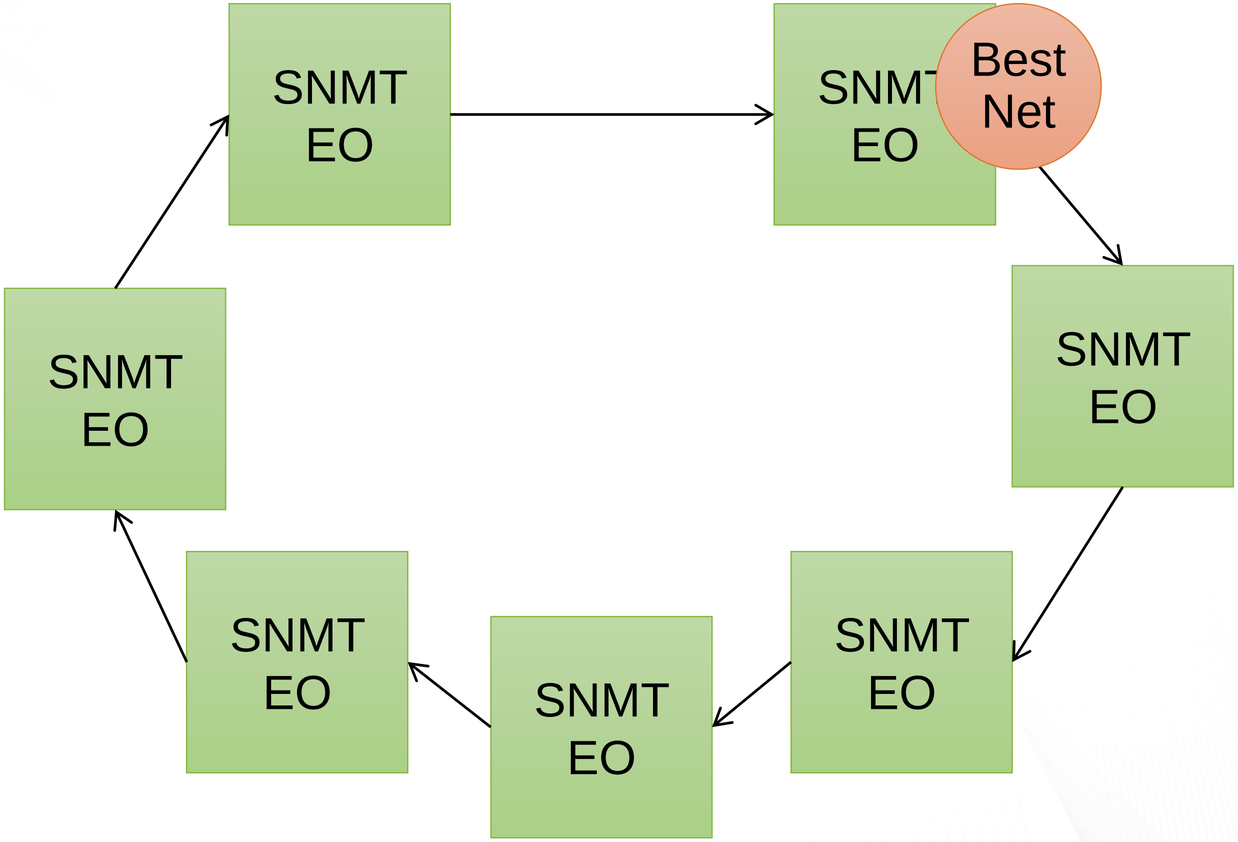
Multi-Node: Master-Slave



Direct Subpopulation to Subpopulation (SP2SP)

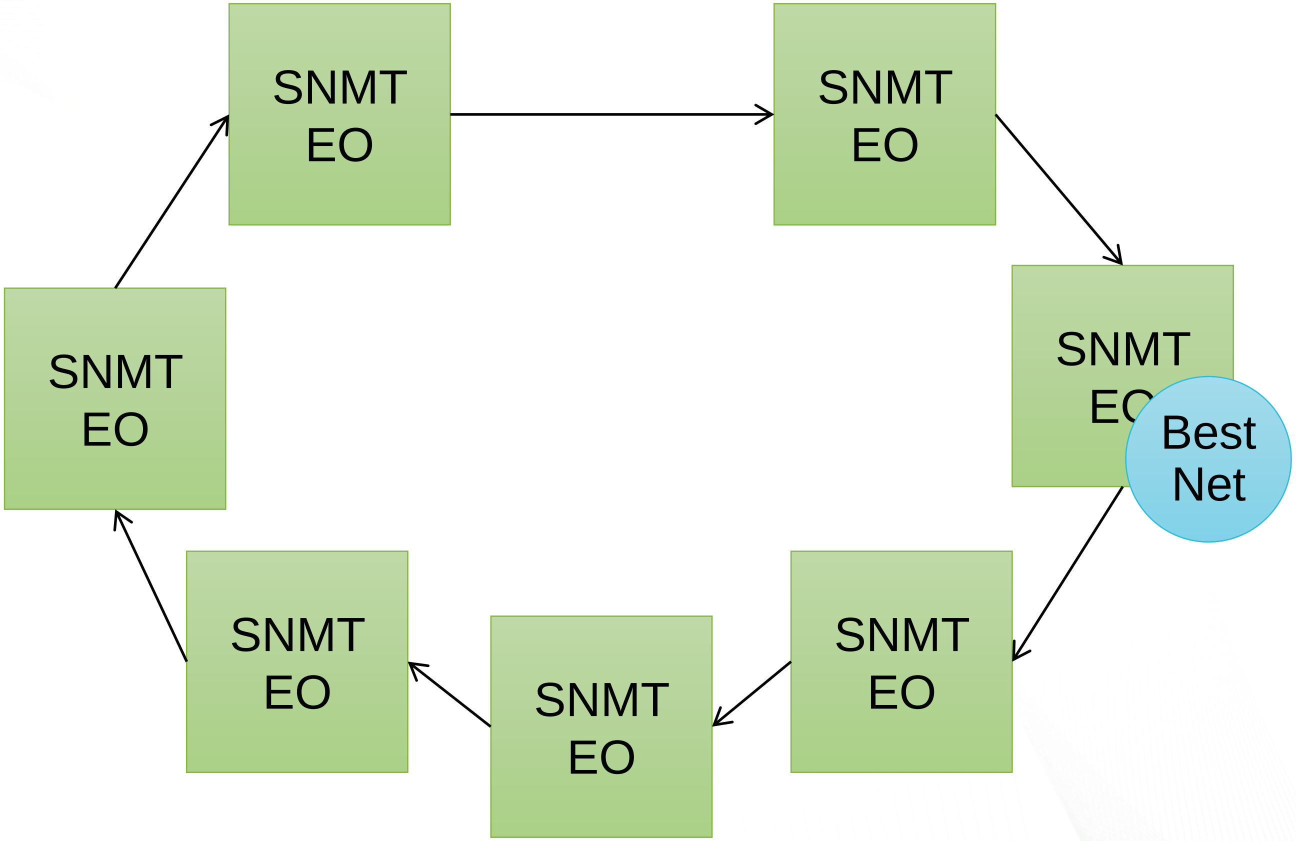


Direct Subpopulation to Subpopulation (SP2SP)



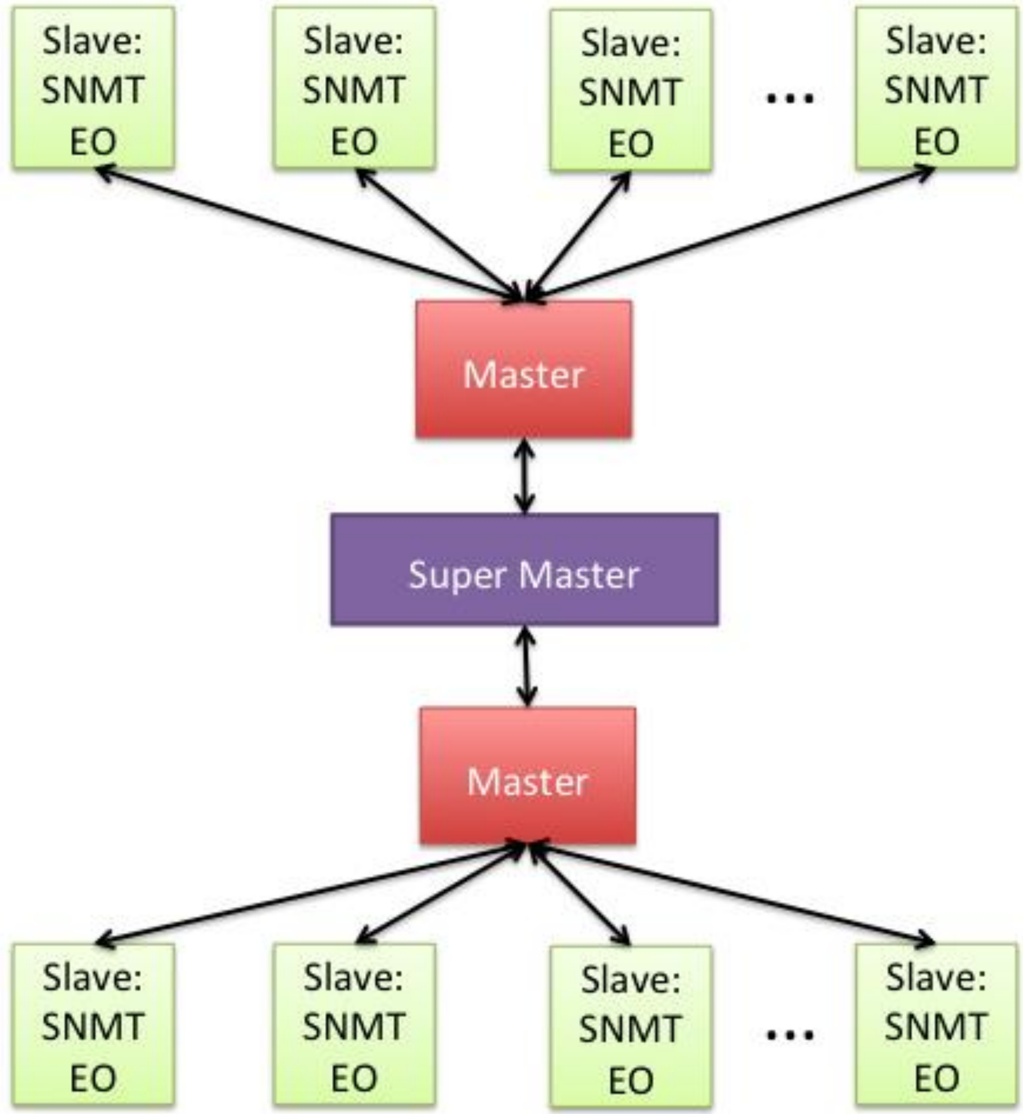
OAK RIDGE National Laboratory

Direct Subpopulation to Subpopulation (SP2SP)



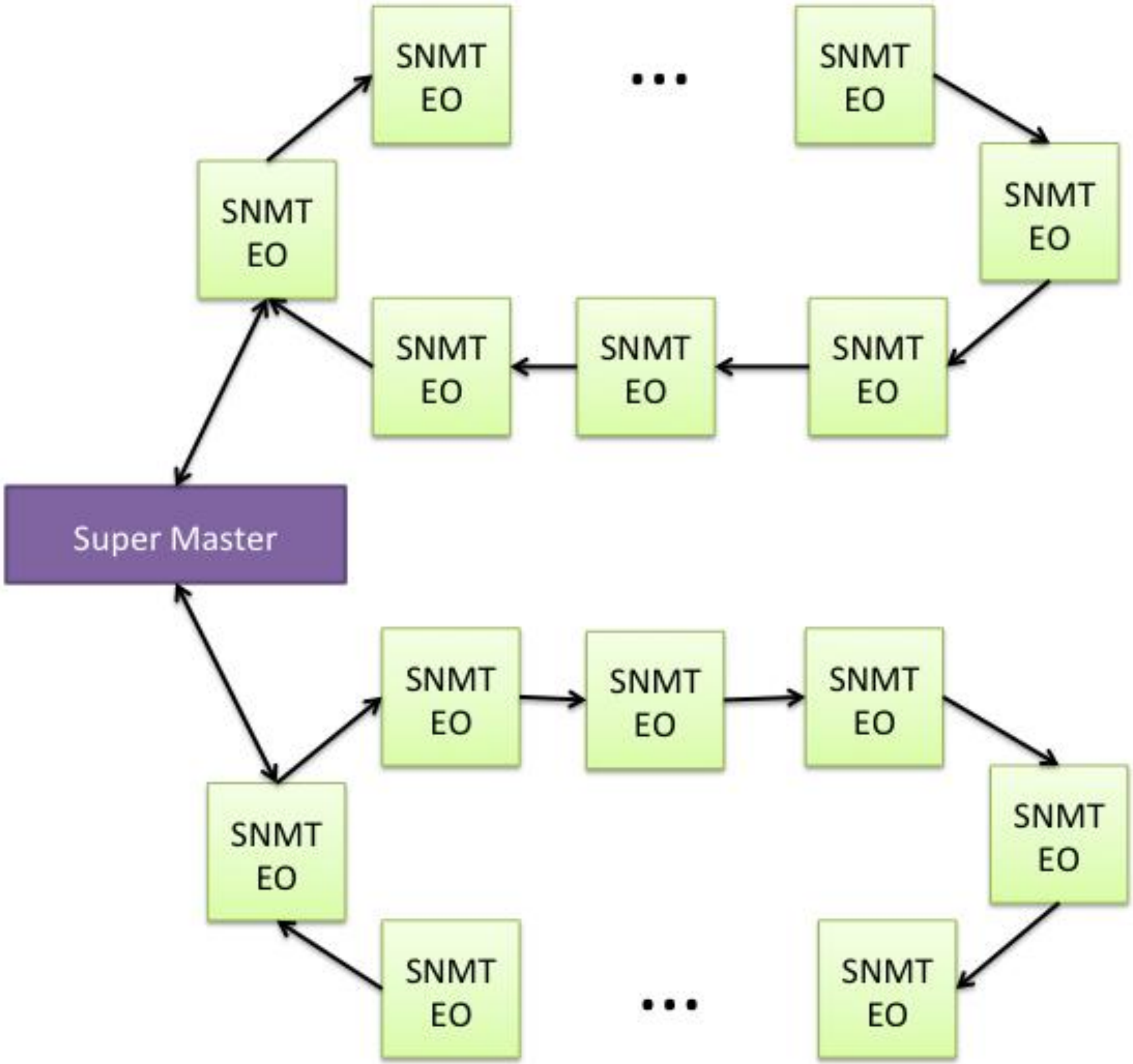
OAK RIDGE National Laboratory

Super Master – Master-Slave



Oak Ridge National Laboratory

Super-Master – SP2SP



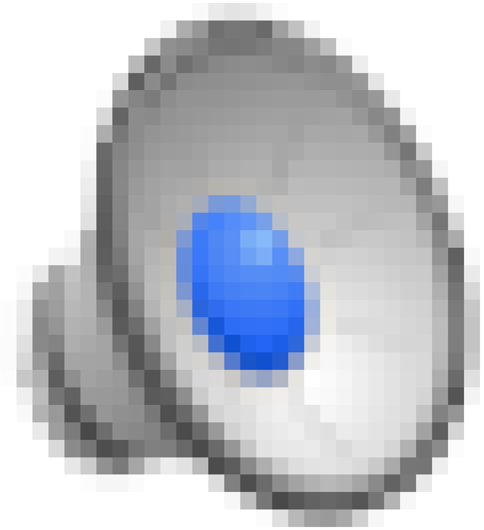
OAK RIDGE National Laboratory

Titan

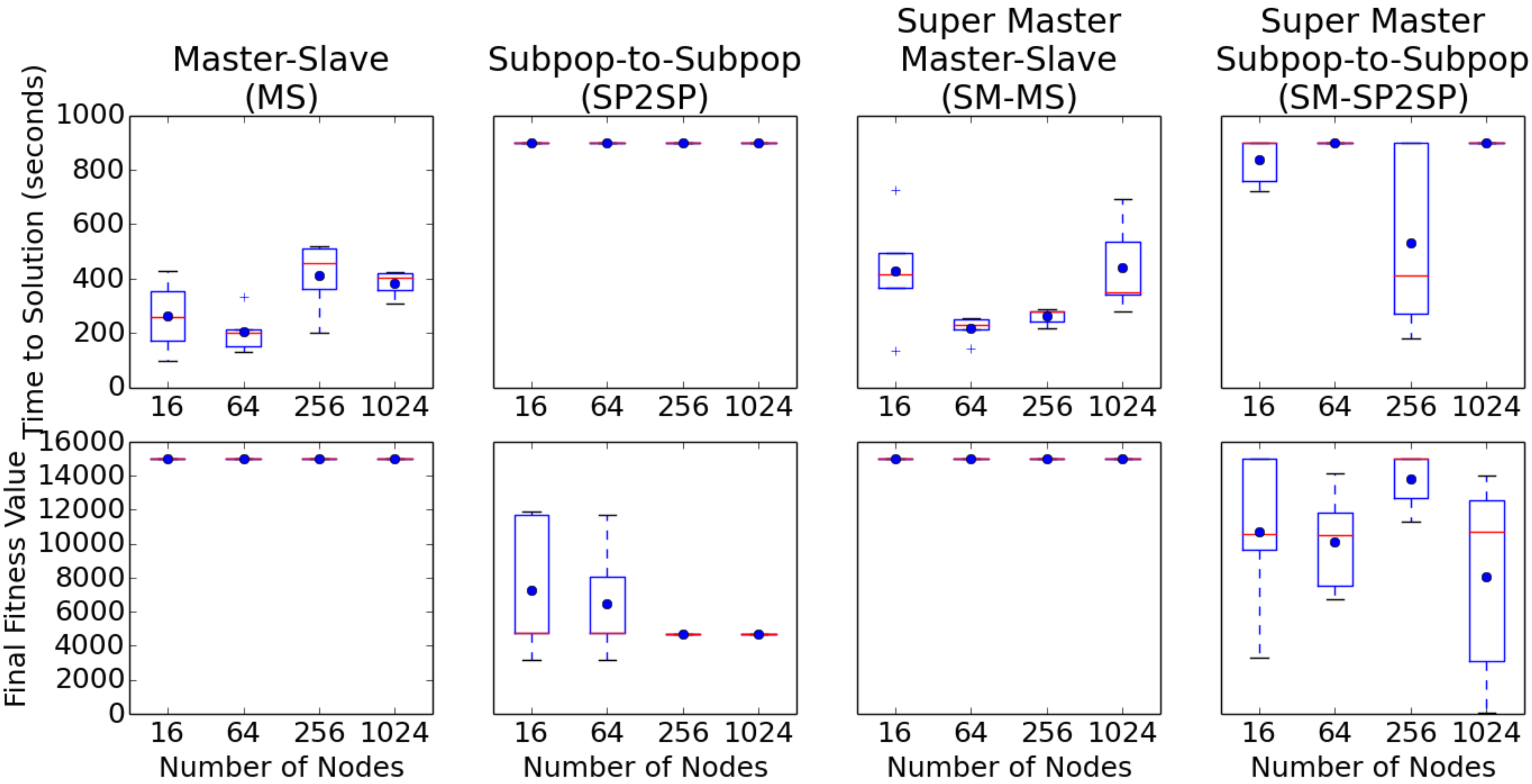
- 18,688 compute nodes – 16 core AMD processors on each node (along with an NVIDIA Kepler GPU).
- 3rd on the Top 500 Supercomputers List in June 2016.

Oak Ridge National Laboratory

Application: Pole Balancing

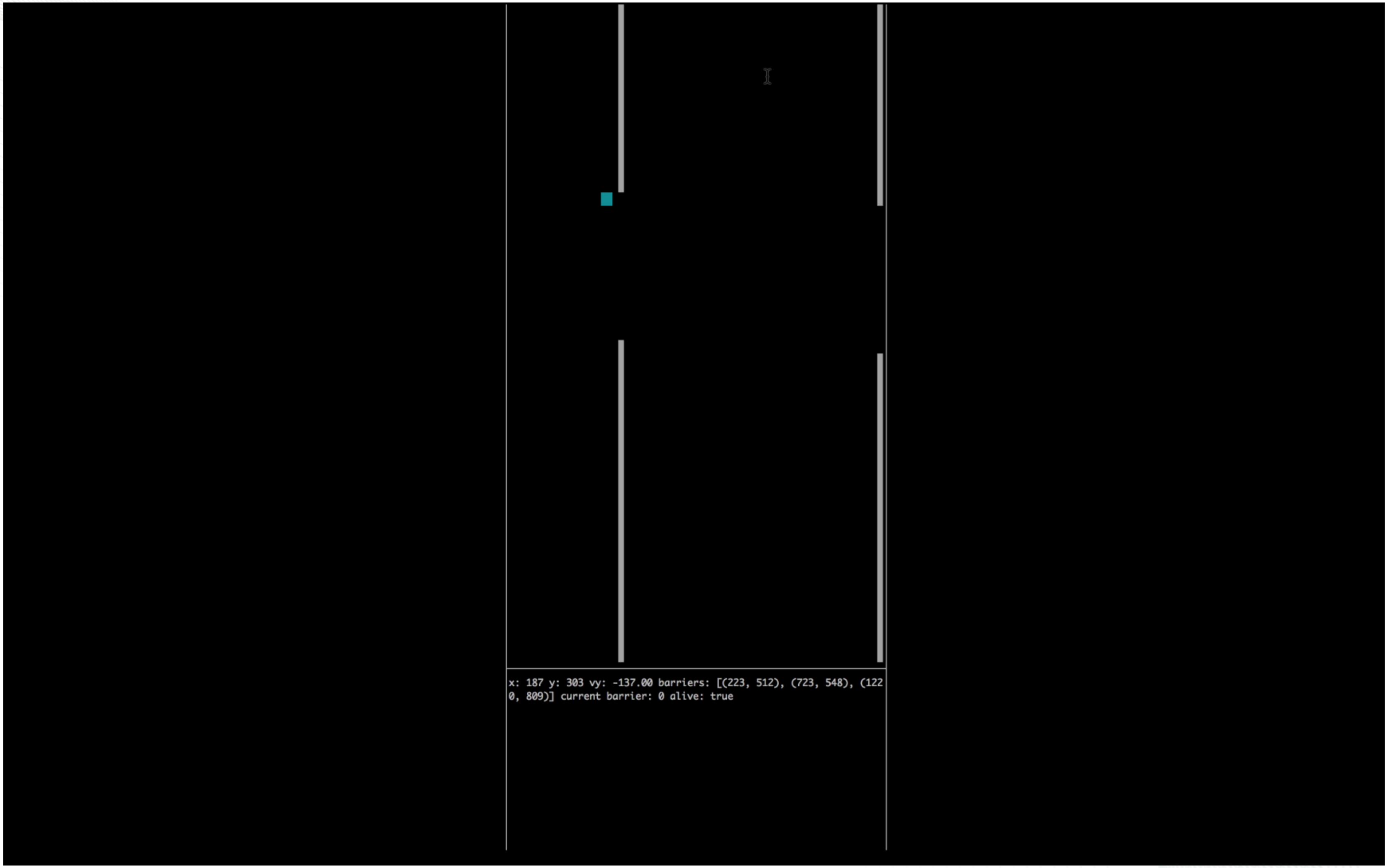


Pole Balancing Results - Titan



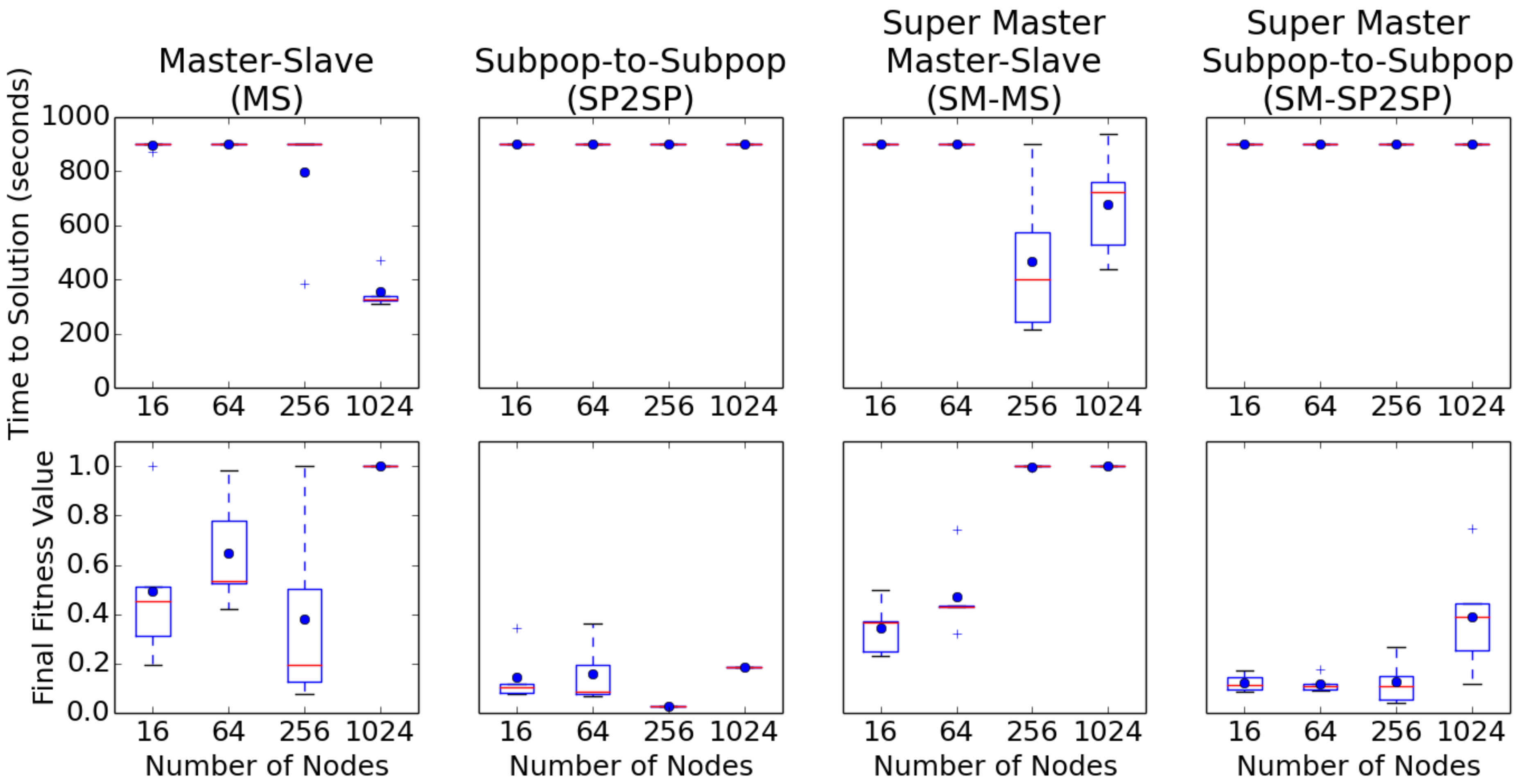
OAK RIDGE National Laboratory

Application: One Dimensional Navigation



OAK RIDGE National Laboratory

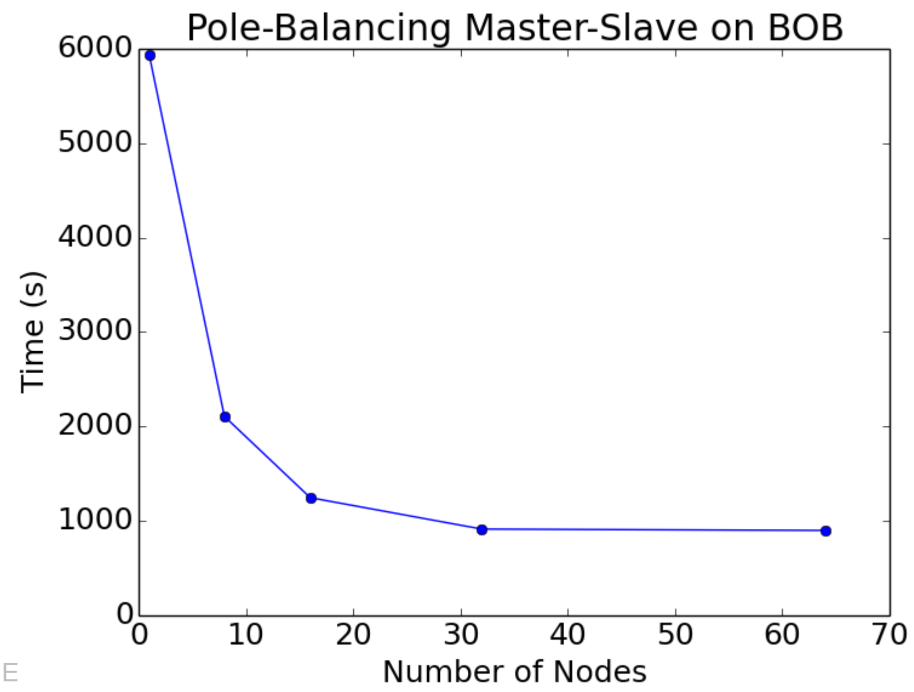
Flappy Bird Results - Titan



OAK RIDGE
National Laboratory

BOB – Raspberry Pi Cluster

- 64 Raspberry Pi 3 Cluster
  - 1.15 GHz quad core ARM Cortex A53 CPU
- Each set of 32 nodes is on a Gigabit Ethernet switch.
  - 32-node sets are connected by a single gigabit link.



OAK RIDGE National Laboratory

Network Structure Analysis

- EO has the by-product of producing lots of networks and their performance characteristics.
- We took 90,000 pole-balancing networks generated on Titan and analyzed input/output paths:

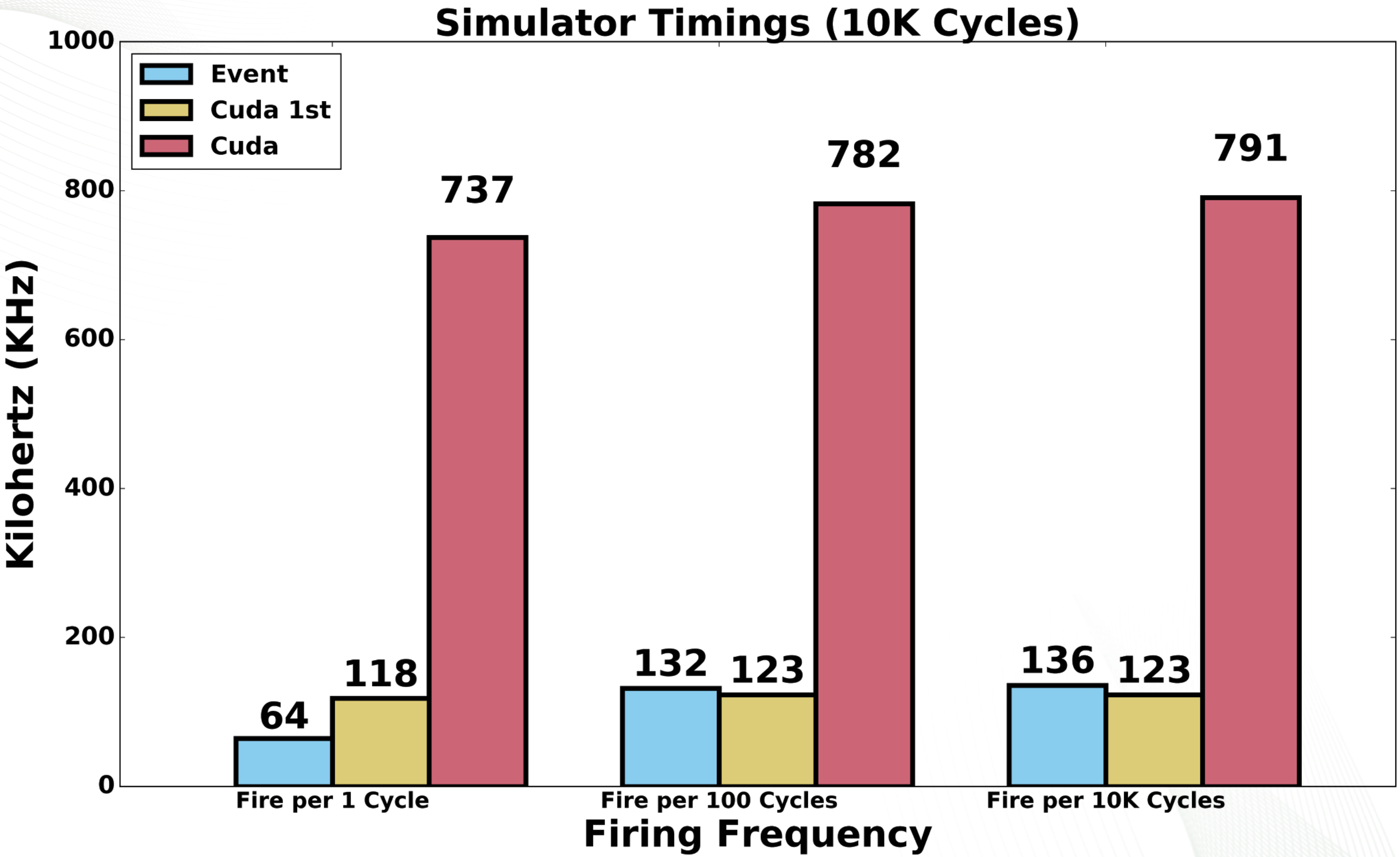
Input Neuron	Output Neuron	Est. Coeff.
(3, 0)	(6, 13)	1.552
(4, 0)	(6, 13)	1.417
(4, 0)	(7, 13)	1.409
(7, 0)	(6, 13)	1.621
(5, 0)	(7, 13)	1.596
(8, 0)	(6, 13)	2.486
(6, 0)	(7, 13)	2.885
(7, 0)	(7, 13)	1.886
(12, 0)	(6, 13)	1.655
(10, 0)	(7, 13)	2.074
(14, 0)	(6, 13)	1.569
(12, 0)	(7, 13)	1.386

OAK RIDGE National Laboratory

Future Work

- GPU simulator is in progress.
  - Can be used in combination with event-based simulator to take advantage of all computing resources to study neuromorphic systems.
- Profiling and optimizing existing implementations.
- Additional exploration of produced networks and their performance characteristics to understand the learning process.
  - What are the characteristics of “good” networks?
  - Can we embed learned information into the training process?

Preliminary GPU Implementation



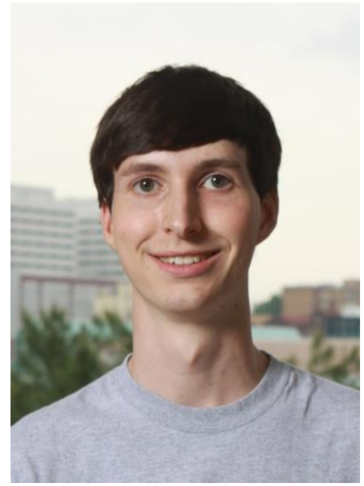
Conclusion

- Neuromorphic computing is clearly one architecture that will be present in the computing landscape of the future.
- There are still many unknowns about neuromorphic computing system, including the most efficient ways to train them.
- Evolutionary optimization (EO) is one way to train neuromorphic networks that is especially amenable for large-scale implementation.
- We implement, test, and demonstrate large-scale parallel EO methods on Titan and BOB.
- We demonstrate one way to utilize the produced results from large-scale EO implementations to study a neuromorphic architecture.

Acknowledgements

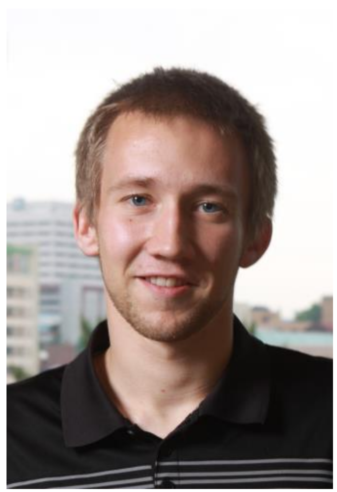
Adam Disney

Susheela Singh



Grant Bruer

Parker Mitchell



Alex Klibisz

Jim Plank

THE UNIVERSITY OF TENNESSEE KNOXVILLE

NC STATE UNIVERSITY

AIR FORCE RESEARCH LABORATORY

OAK RIDGE National Laboratory

Thank You!

Email: schumancd@ornl.gov
Website: neuromorphic.eecs.utk.edu

GPU Improvements

- Batching of DANNA I/O packets
  - This reduces the number of kernel launches required
- Pinned memory for DANNA I/O packets
  - Allows transfers to be concurrent with simulation
- Load array in shared memory
  - Can’t really “block” simulation like a matrix multiply but current arrays are small enough to entirely fit in shared memory
  - In combination with Batching, only have to hit global memory at the beginning and end of the batch
- Neuron and Synapse list
  - To avoid divergence, divide the warps into Neuron warps and Synapse warps that run through their respective lists.