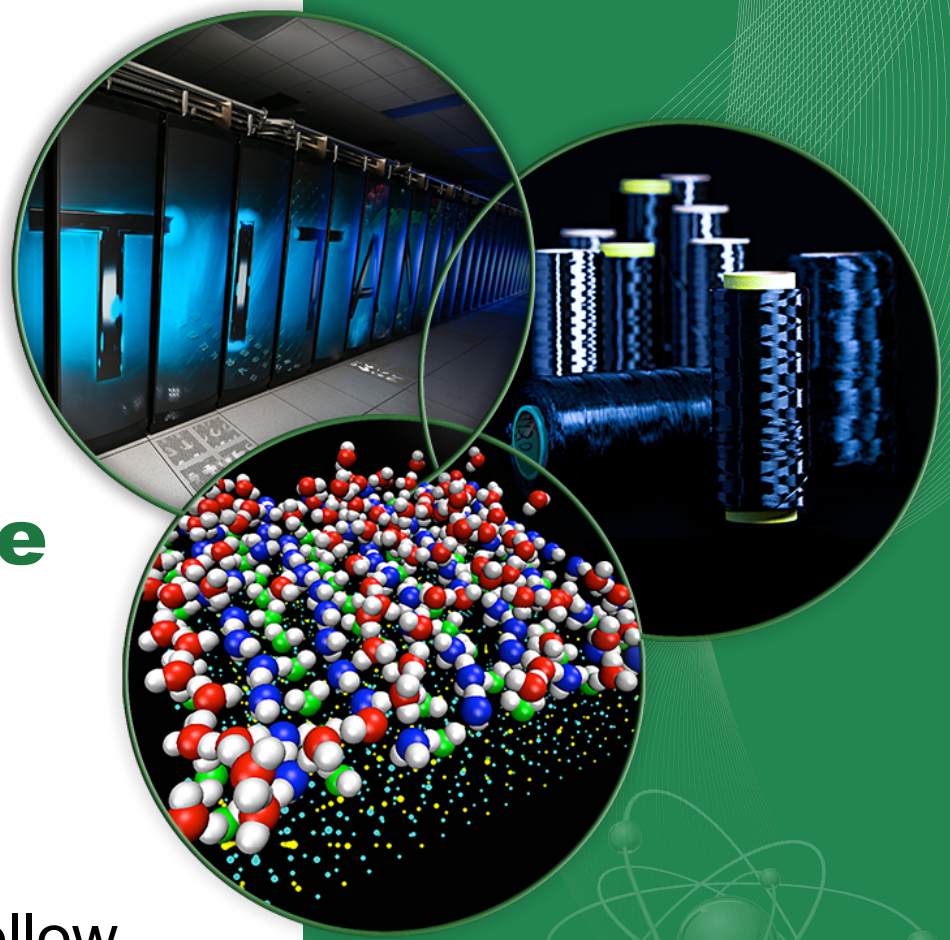
Roadmap for Neuromorphic Computing: A Computer Science Perspective



Catherine Schuman

Liane Russell Early Career Fellow

Computational Data Analytics

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

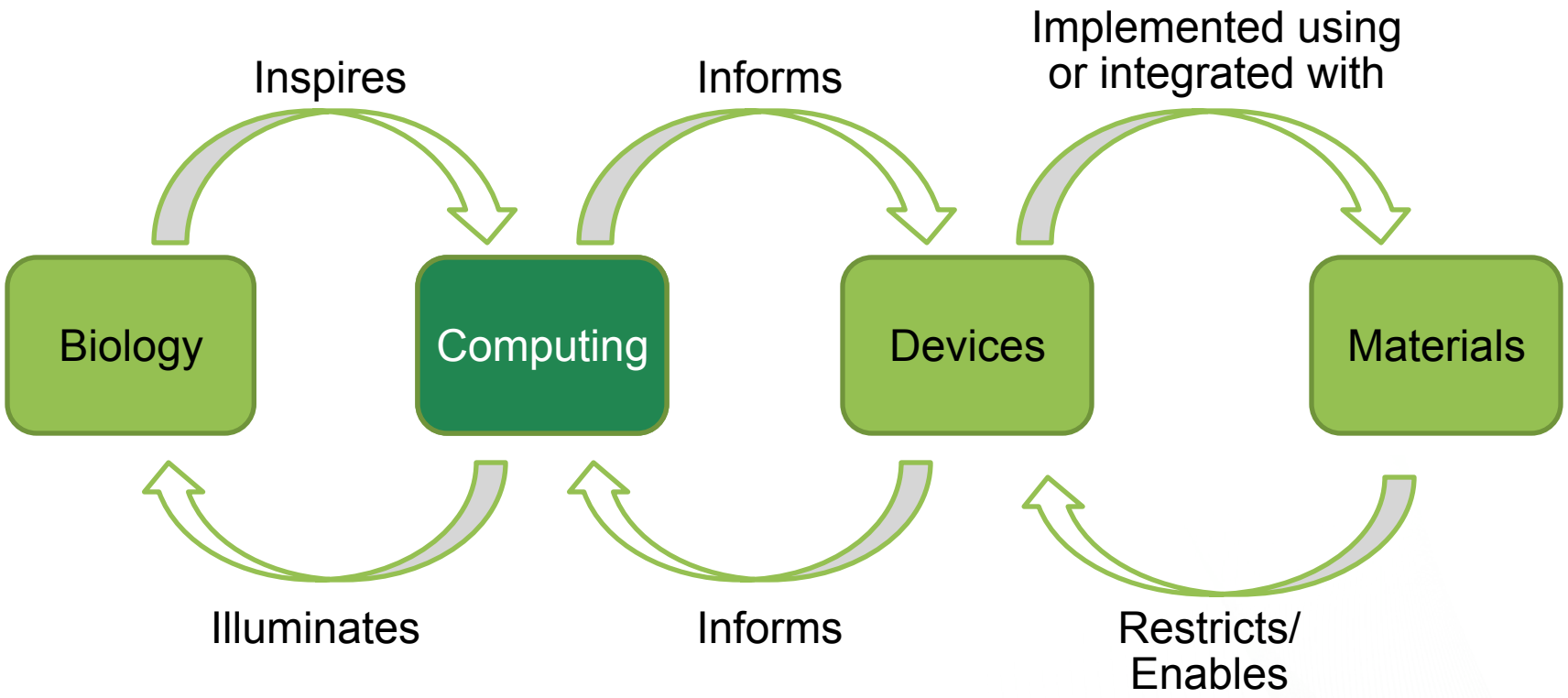
OAK RIDGE National Laboratory

Overview

- Introduction
- Major Research Questions:
  - What are the computational primitives?
  - What degree of programmability is required at the device level?
  - How do we program neuromorphic computers?
  - How do we make neuromorphic systems more usable and accessible?
  - What applications are most appropriate for neuromorphic computers?
- Conclusions

OAK RIDGE National Laboratory

Neuromorphic Computing Community

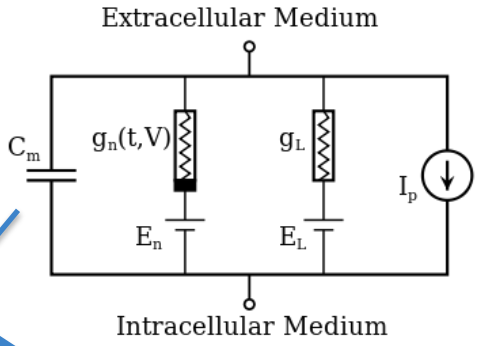


OAK RIDGE National Laboratory

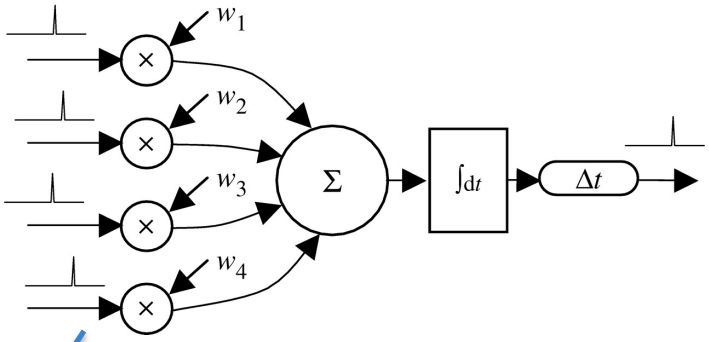
What are the computational primitives?

Neuron/Synapse Model Choices

Biologically Accurate

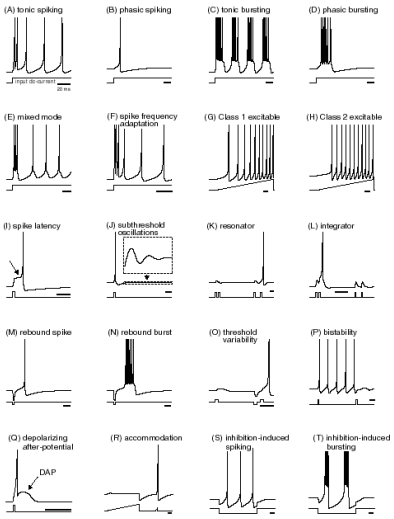


Hodgkin-Huxley

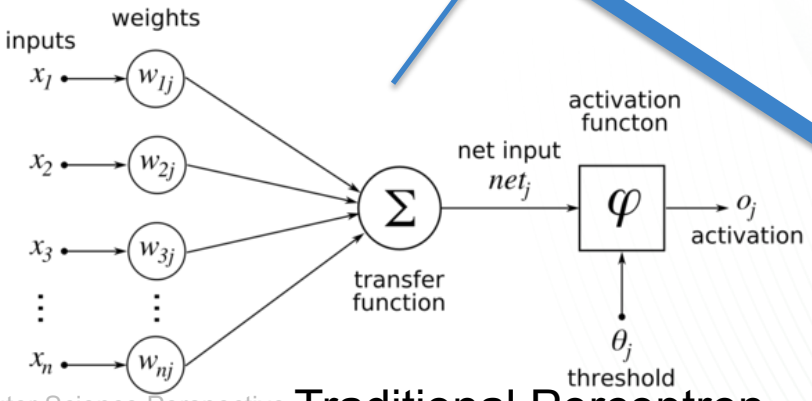


Leaky Integrate and Fire

Izhikevich



NIDA



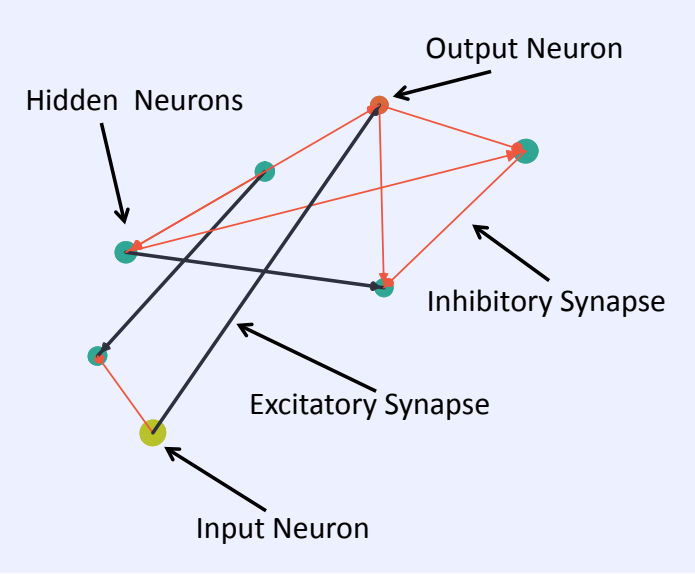
Traditional Perceptron

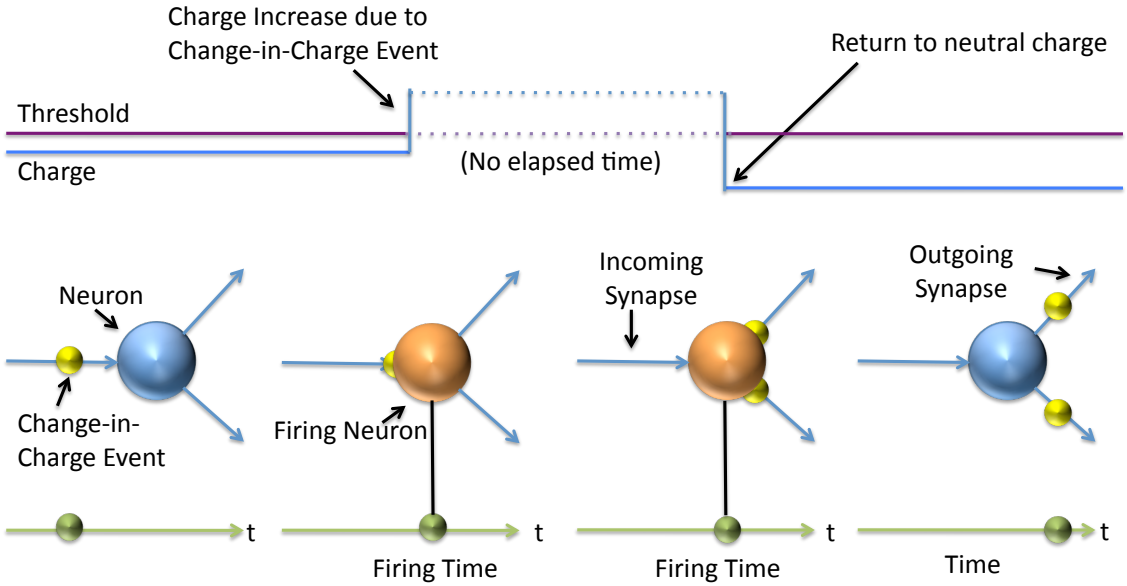
Biologically Inspired

OAK RIDGE National Laboratory

Example: Neuroscience-Inspired Dynamic Architecture (NIDA)

- Spiking neural network embedded in 3D space.
- Simple neuron and synapse implementation.
- Flexible structure.





OAK RIDGE National Laboratory

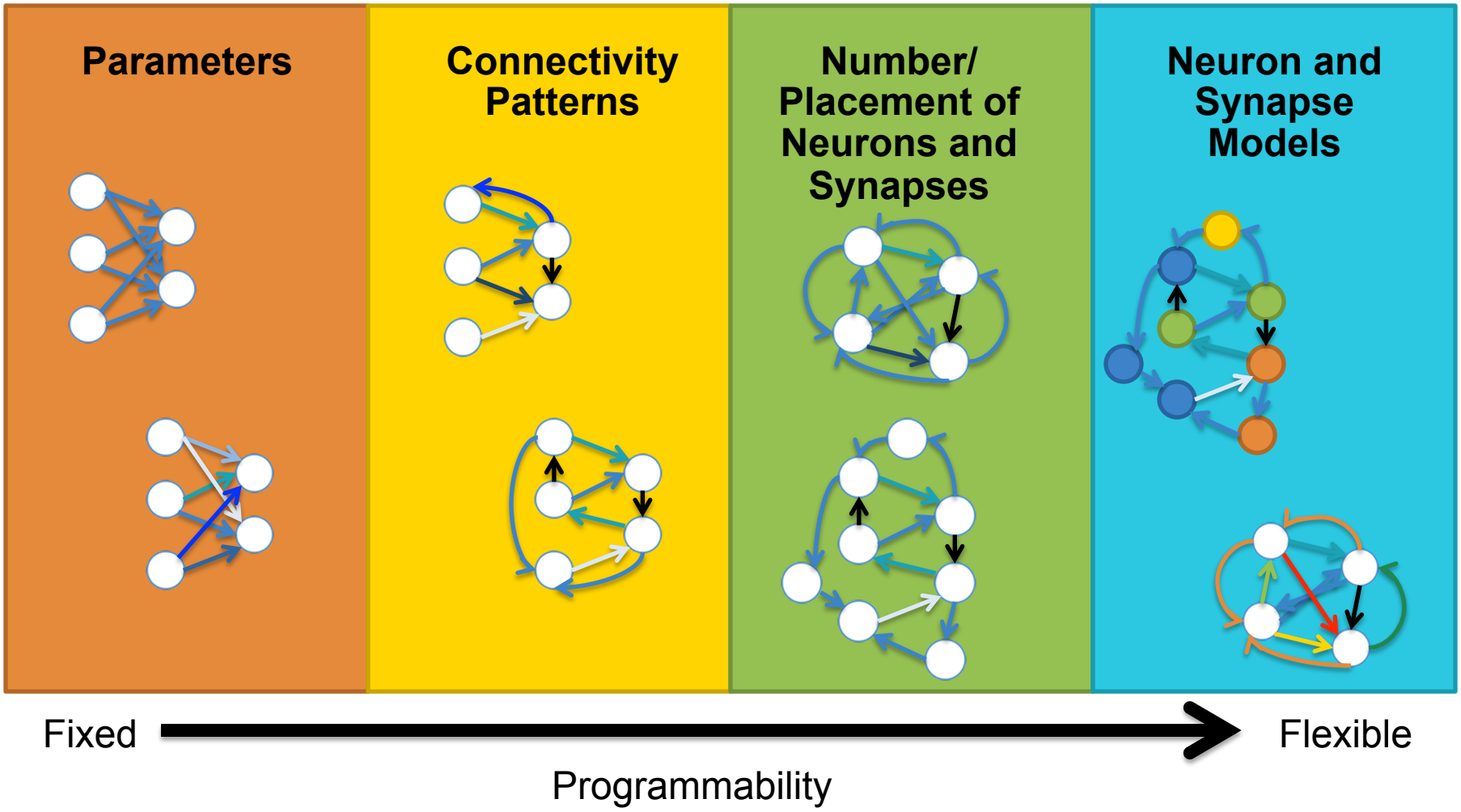
Open Questions

- What neuron and synapse models are most appropriate?
  - Are they application specific?
- What effect does the selection of the computational primitives have on programmability?
- How biologically-accurate should the models be?
- How does the choice of model/computational primitive affect the programming method or algorithm?
- How does the chosen device/material affect the choice of computational primitives?

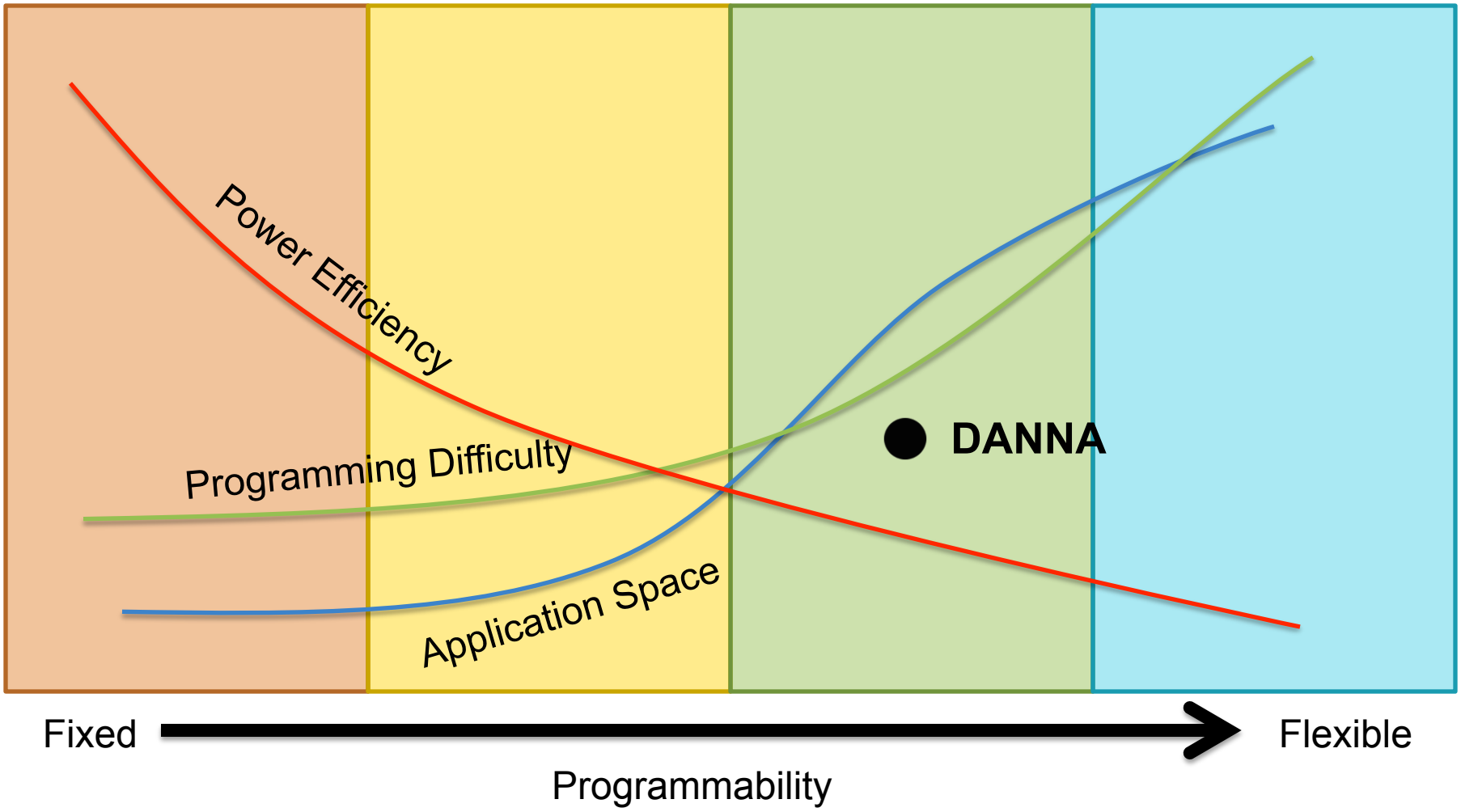
OAK RIDGE National Laboratory

What degree of programmability is required at the device level?

Programmability at Device Levels

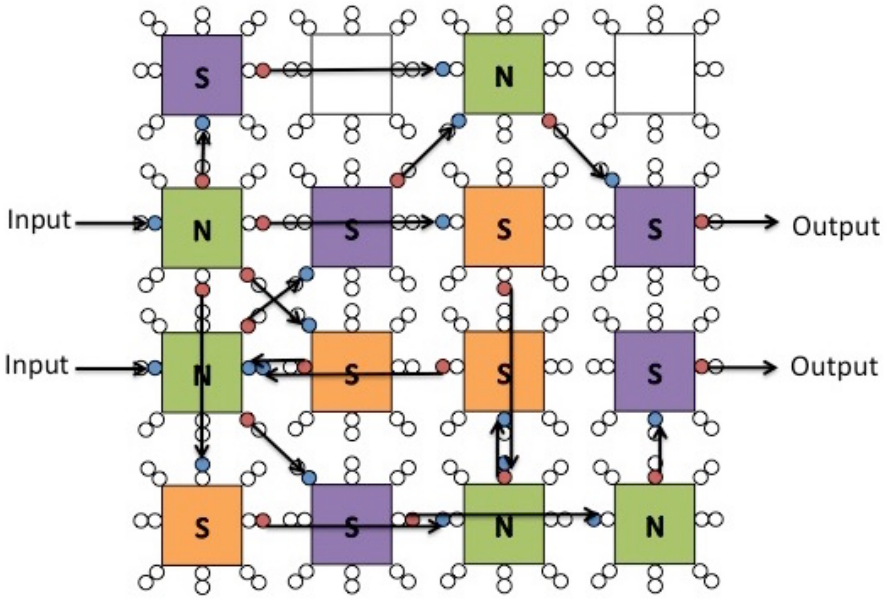


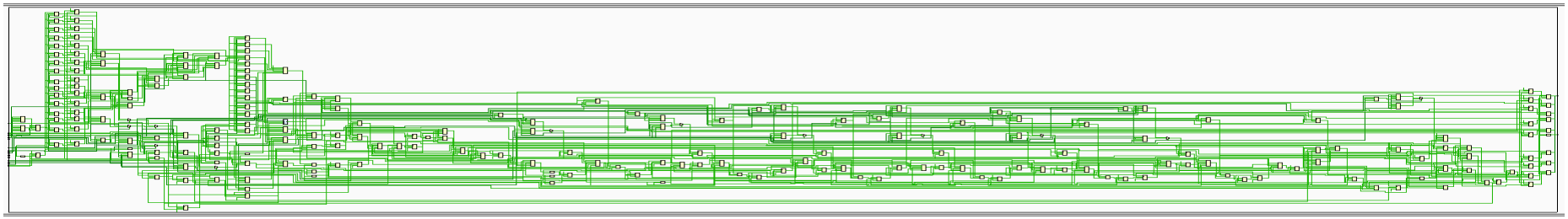
Programmability at Device Levels



Example: 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

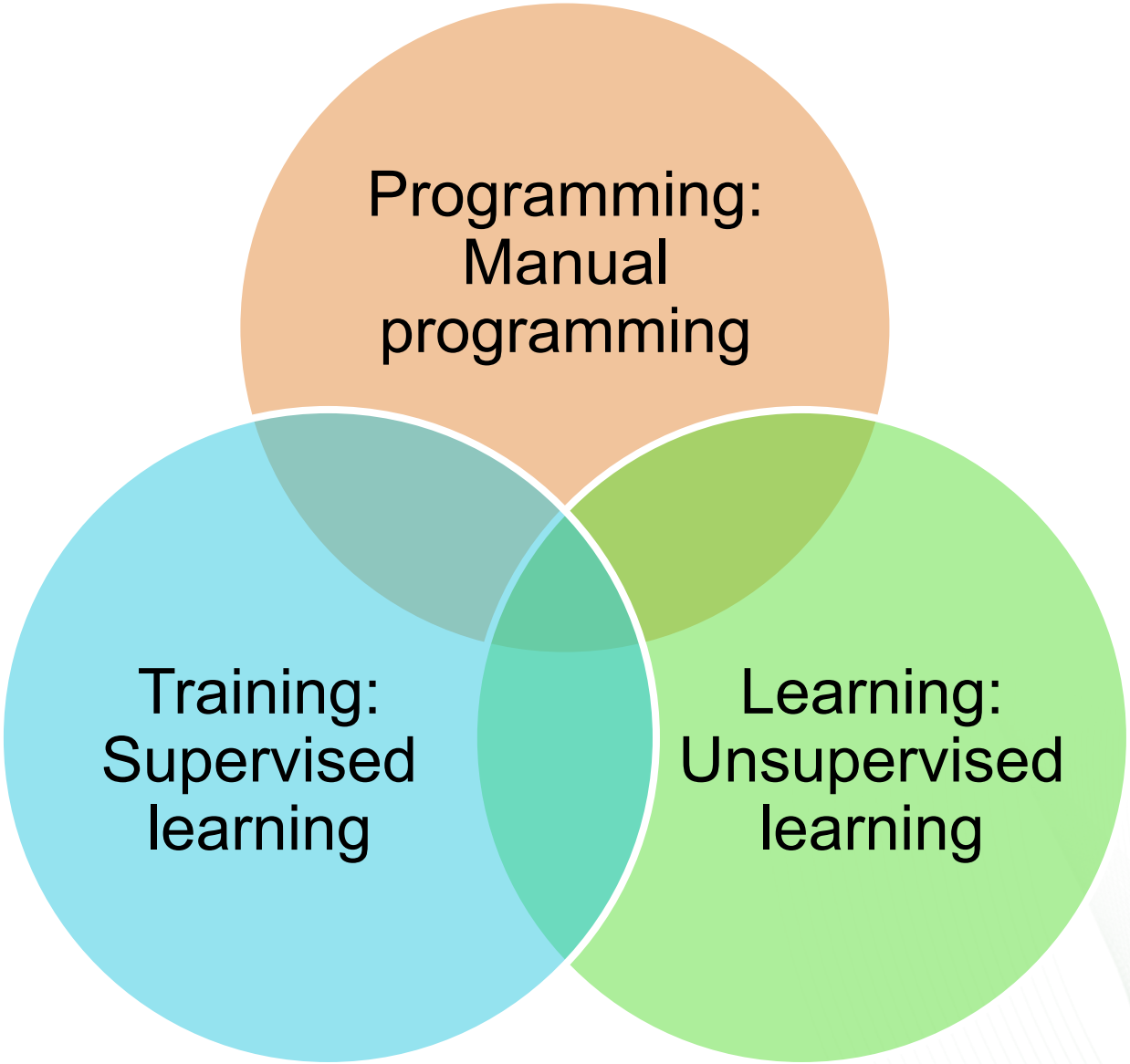
Open Questions

- Do neuromorphic devices for different applications need varying levels of programmability?
- What elements need to be programmable?
  - Parameters
  - Connectivity patterns
  - Structure (number of neurons and synapses)
- How does programmability affect the model selection?
- What is the impact of device programmability on the programming and training algorithms?
- What device types and materials enable programmability?

OAK RIDGE National Laboratory

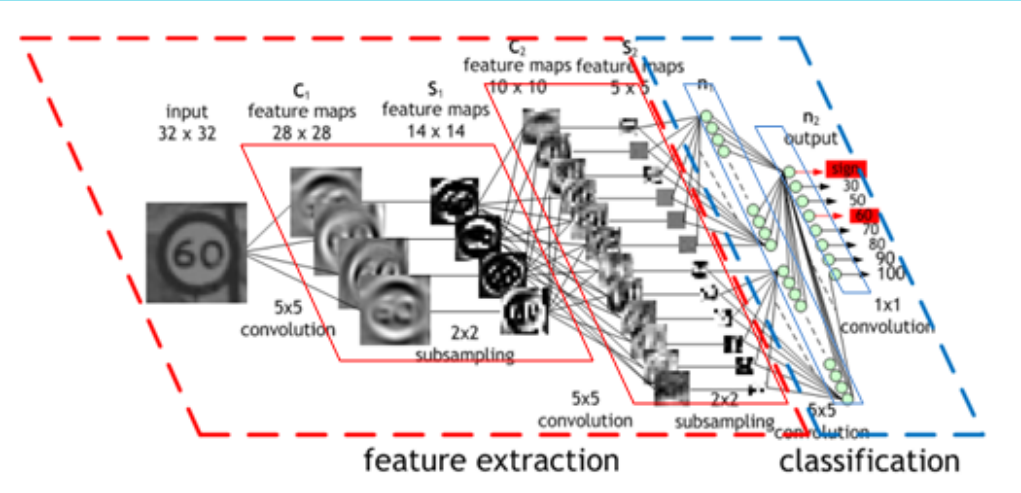
How do we program neuromorphic computers?

Training/Learning/Programming



OAK RIDGE National Laboratory

Training/Learning/Programming



State-of-the-Art Deep Learning Networks
(Trained)

Forced translation
(Programmed)

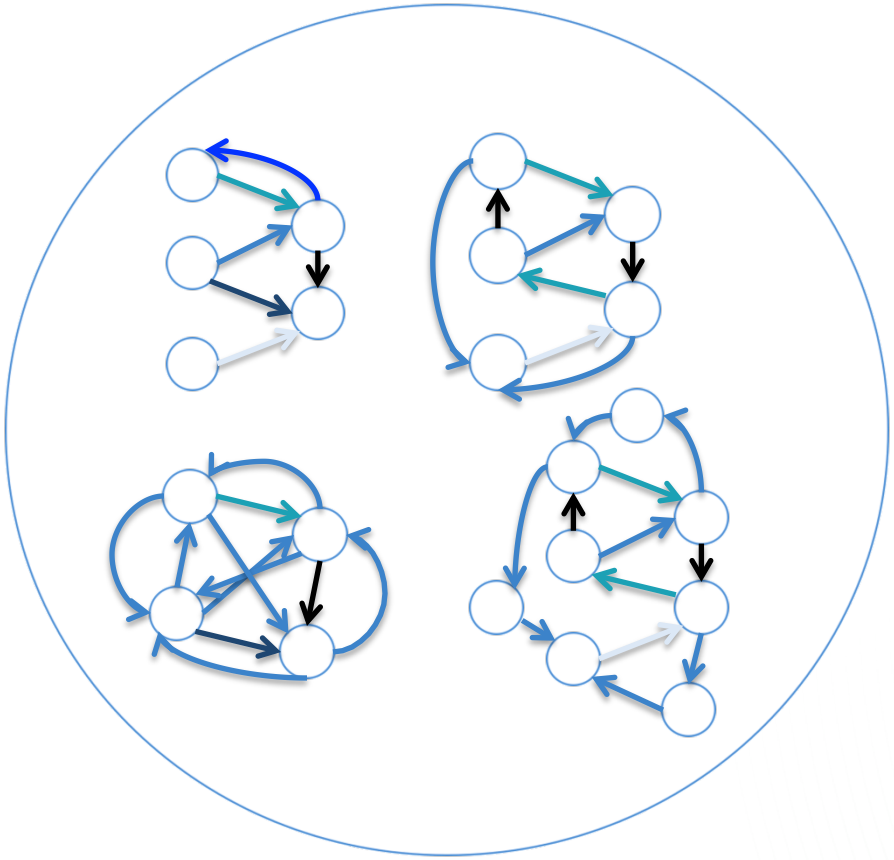
- Does not use the device in training.
- Does not train directly for the device.

Spiking
Neuromorphic
Device

Conv Net Image Source: https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/

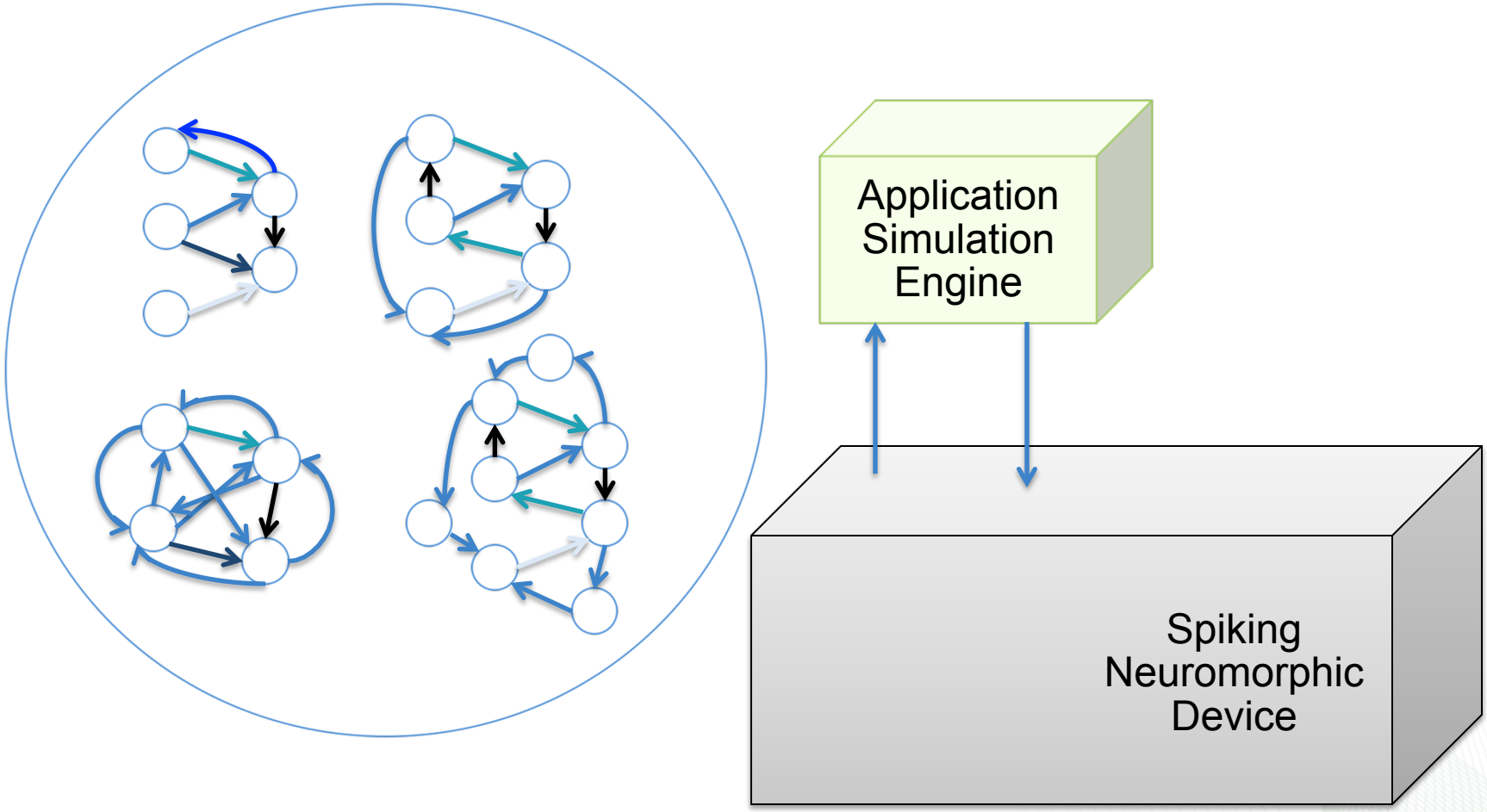
OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization



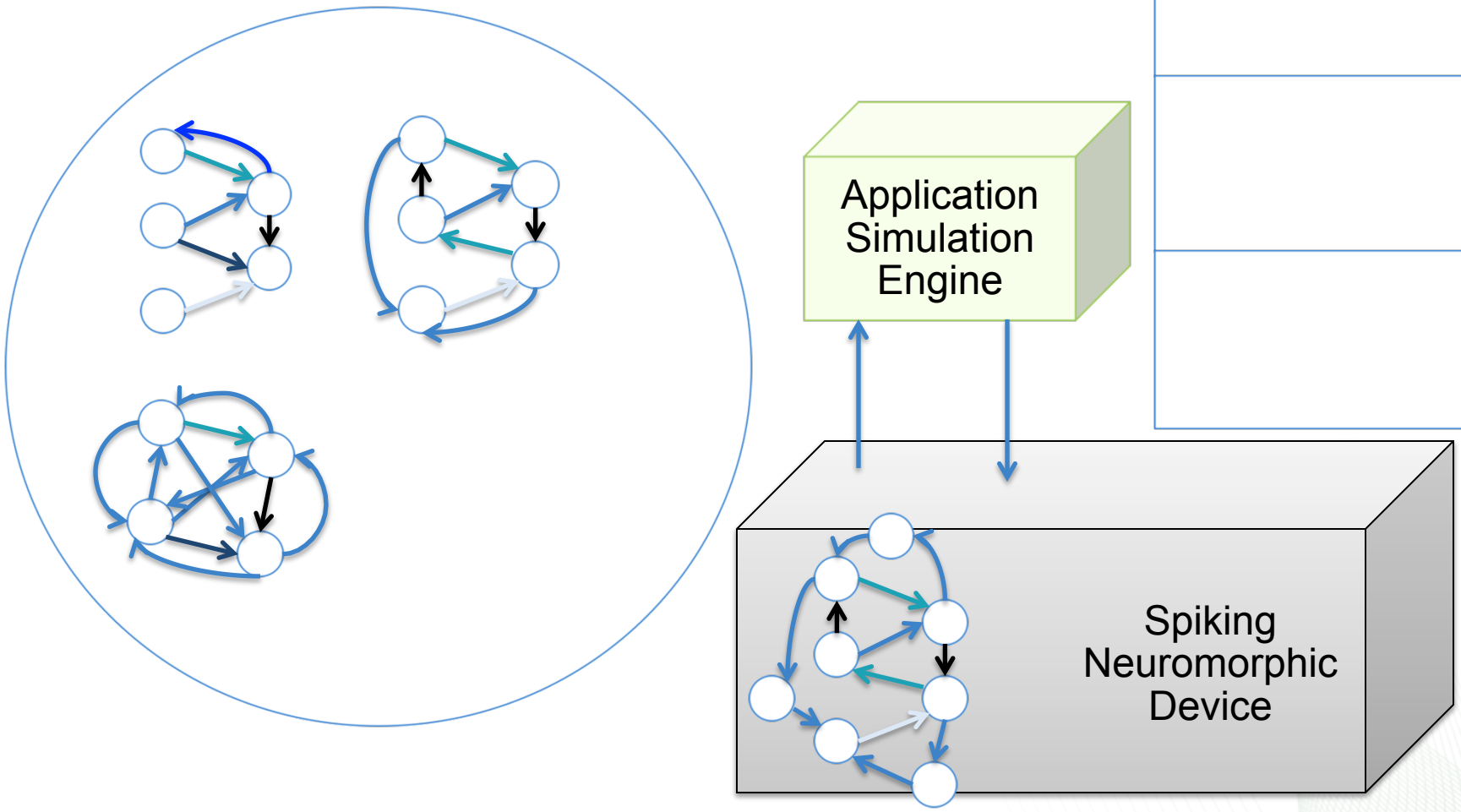
OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization



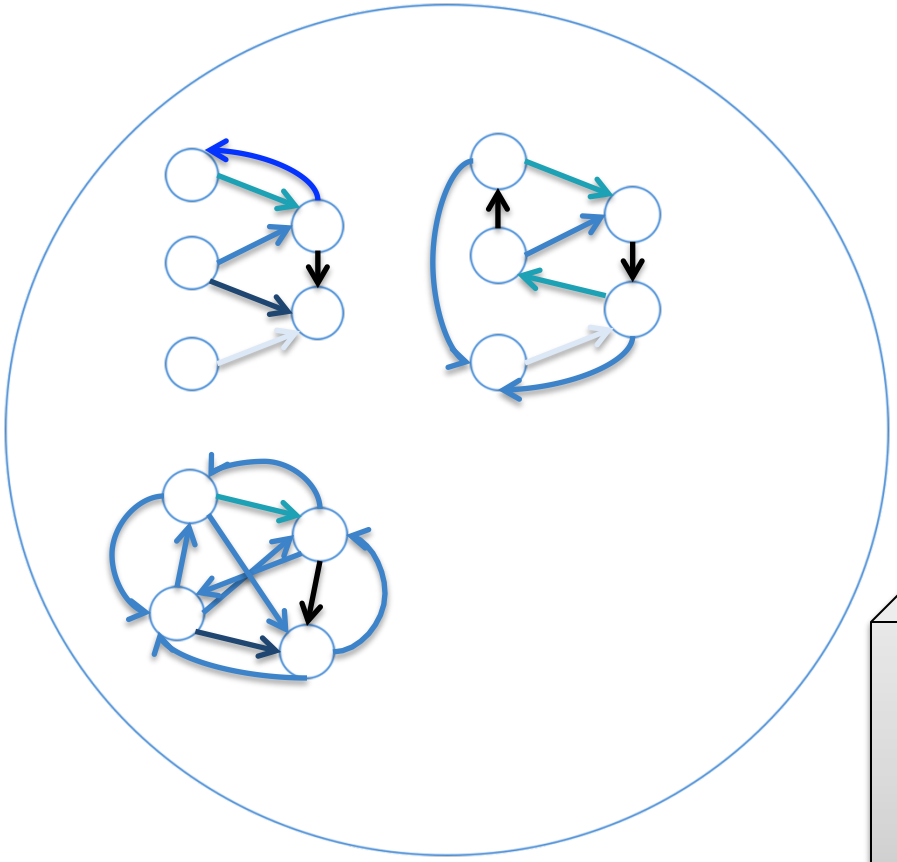
OAK RIDGE National Laboratory

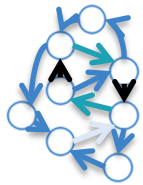
Example Training/Design: Evolutionary Optimization



OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization





1.5

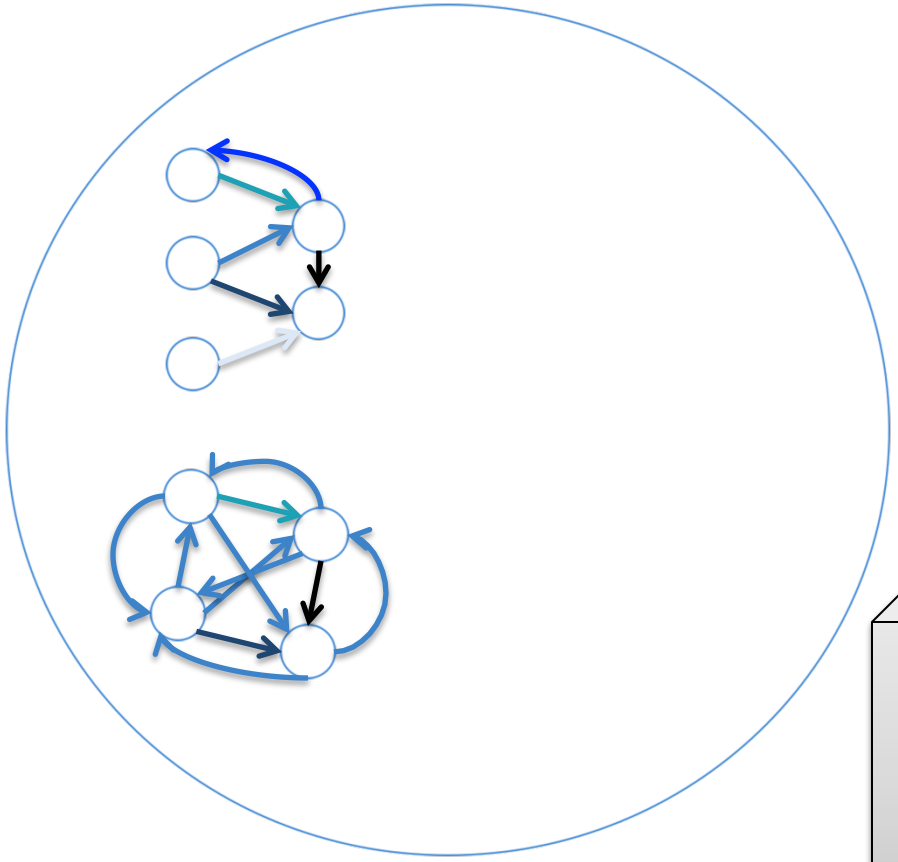
4

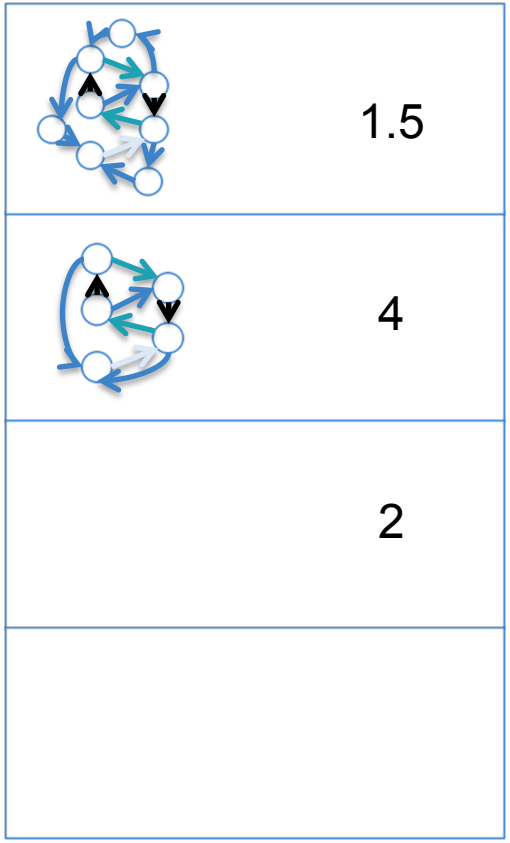
Application Simulation Engine

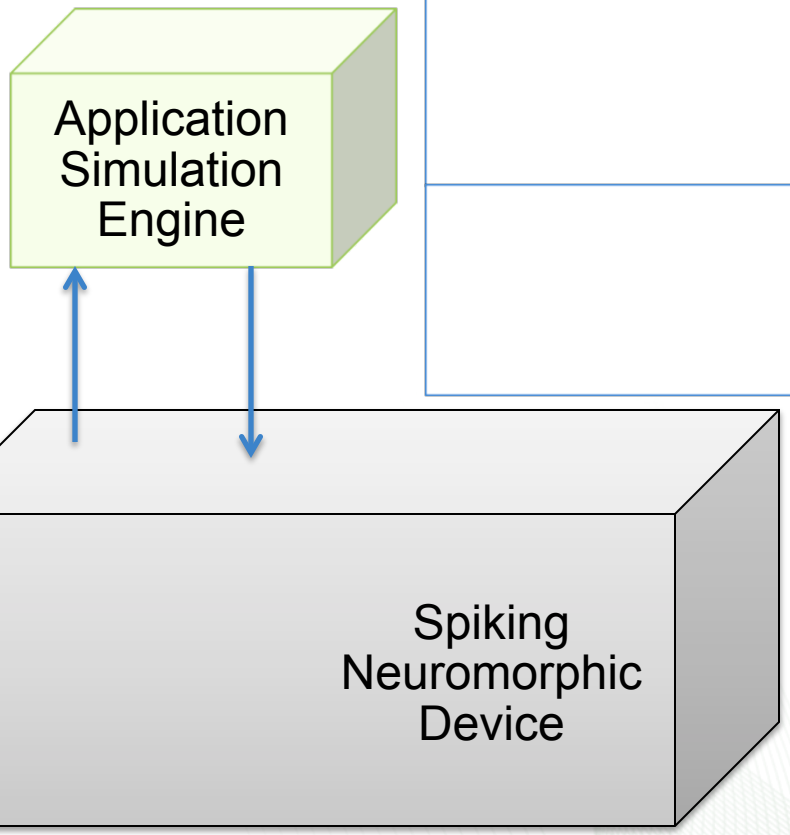
Spiking Neuromorphic Device

OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization

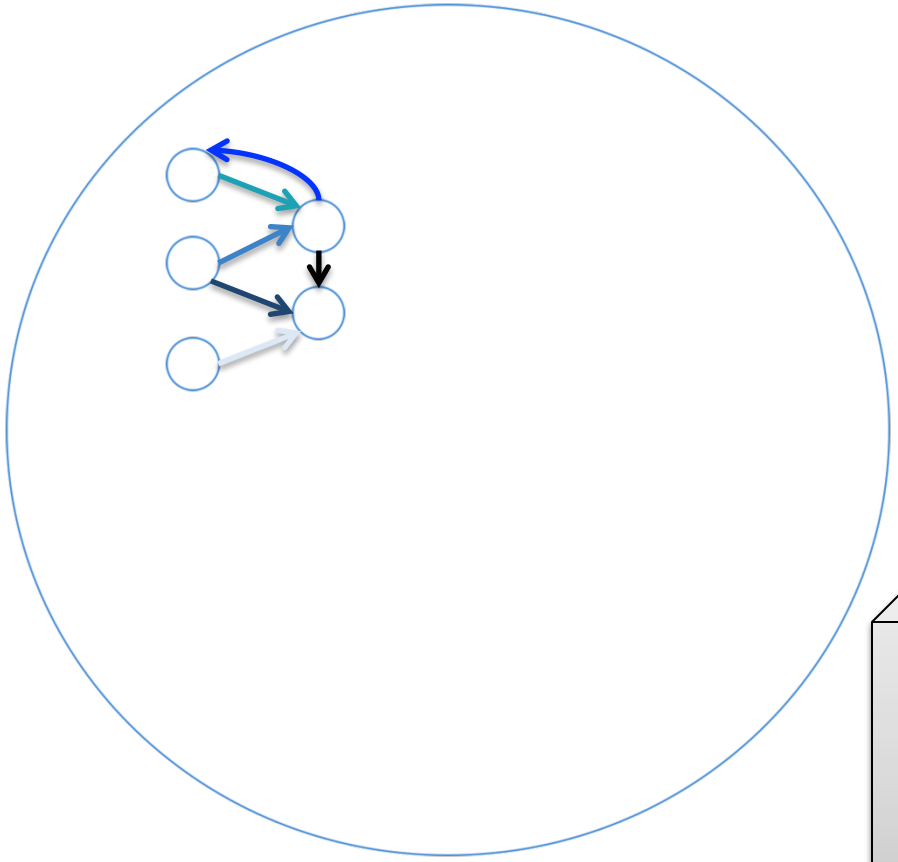


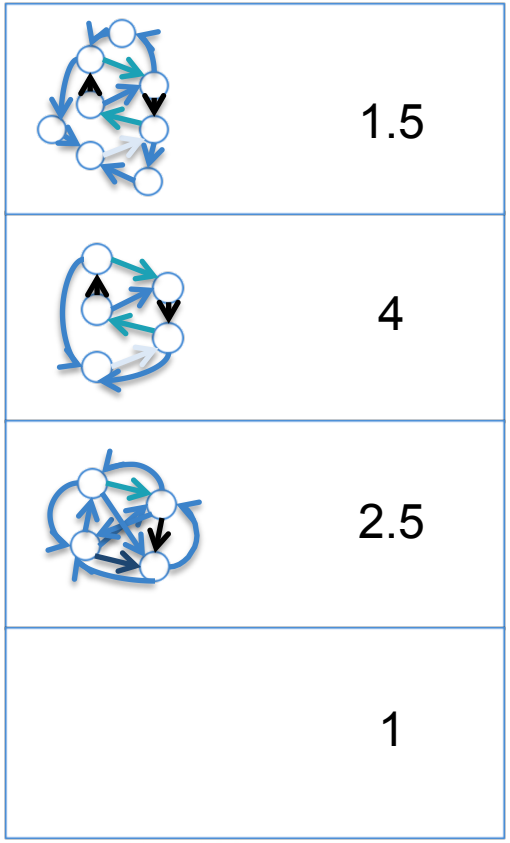


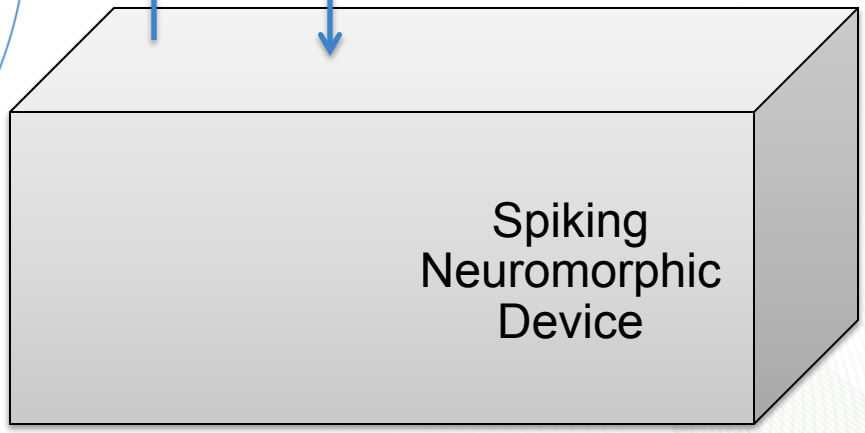


OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization

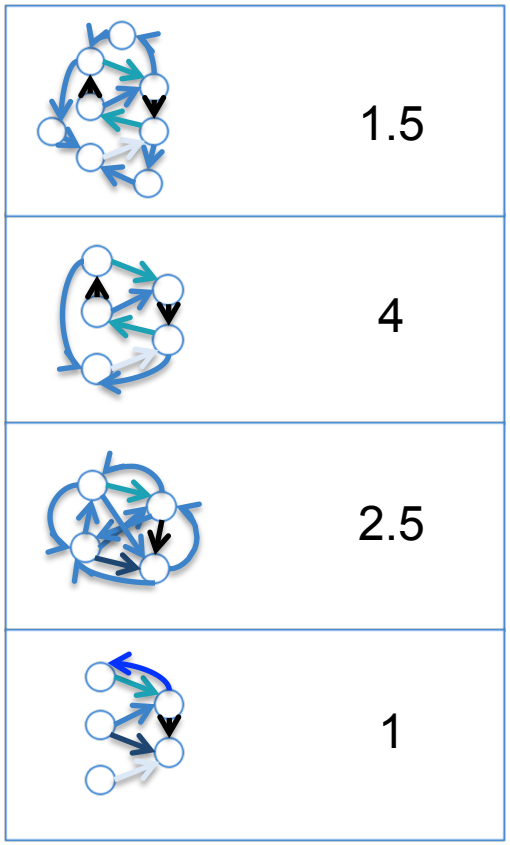






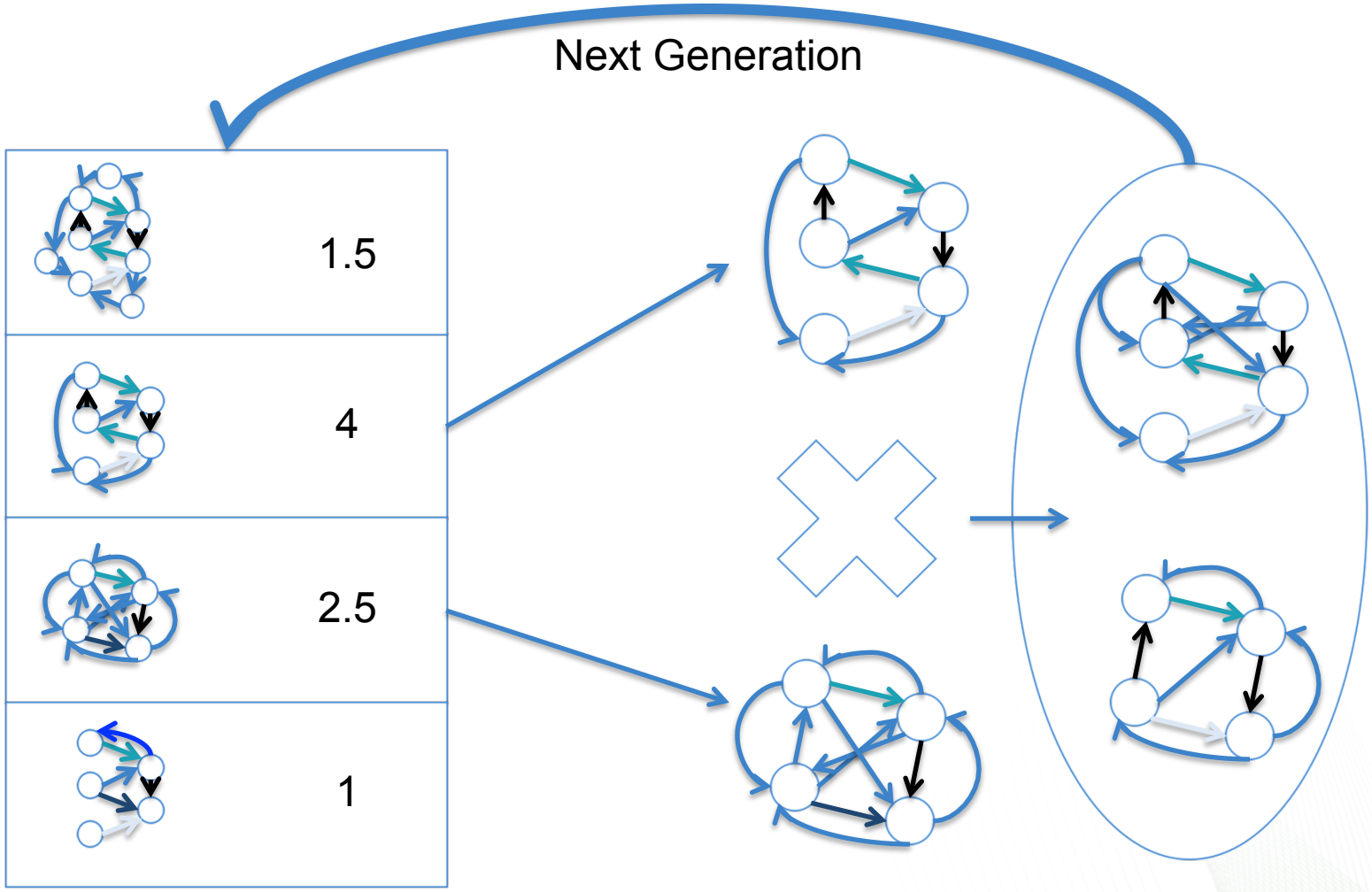
OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization



OAK RIDGE National Laboratory

Example Training/Design: Evolutionary Optimization



OAK RIDGE National Laboratory

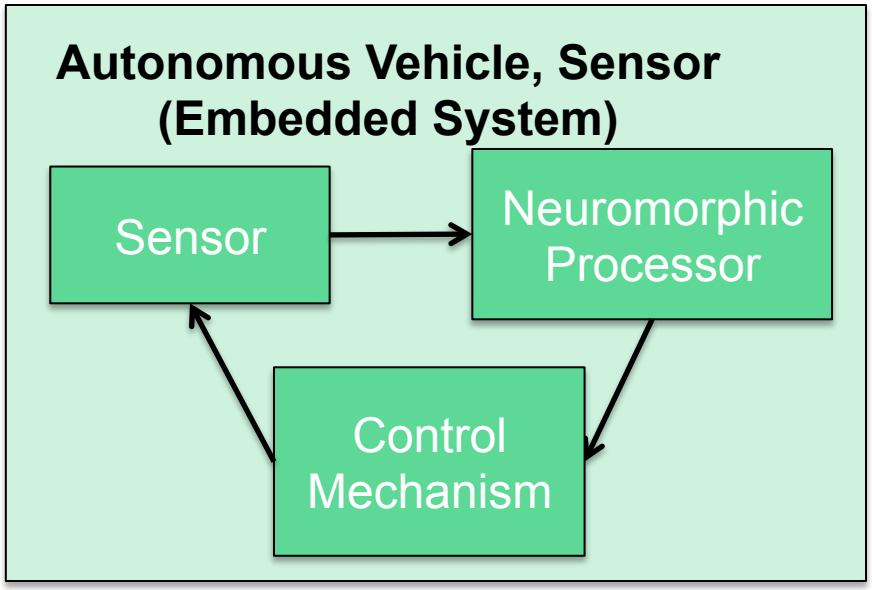
Open Questions

- What are the tradeoffs between programming, training, and learning?
- How important is on-chip learning vs. off-chip learning?
- How important is on-line learning?
- Which biologically-inspired learning mechanisms are important for learning?
  - Spike-timing dependent plasticity, neurogenesis, neuromodulation.
- Which optimization methods are appropriate for training?
  - Gradient-based methods, evolutionary optimization.
- How does model selection influence the programming method?

OAK RIDGE National Laboratory

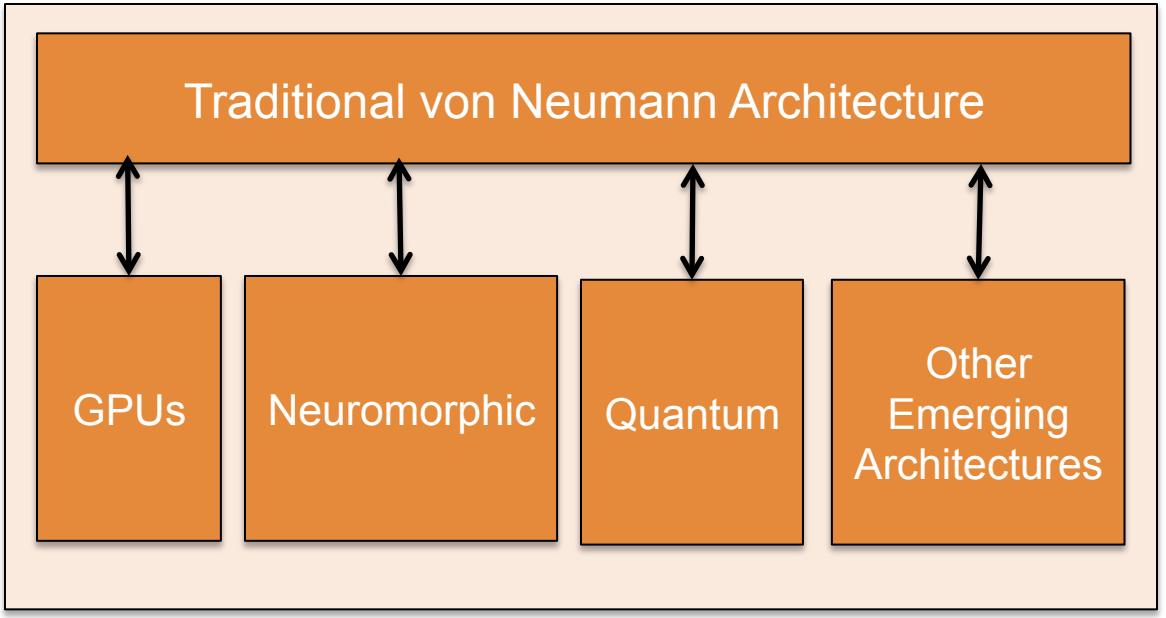
How do we make neuromorphic systems usable and accessible?

Use Cases

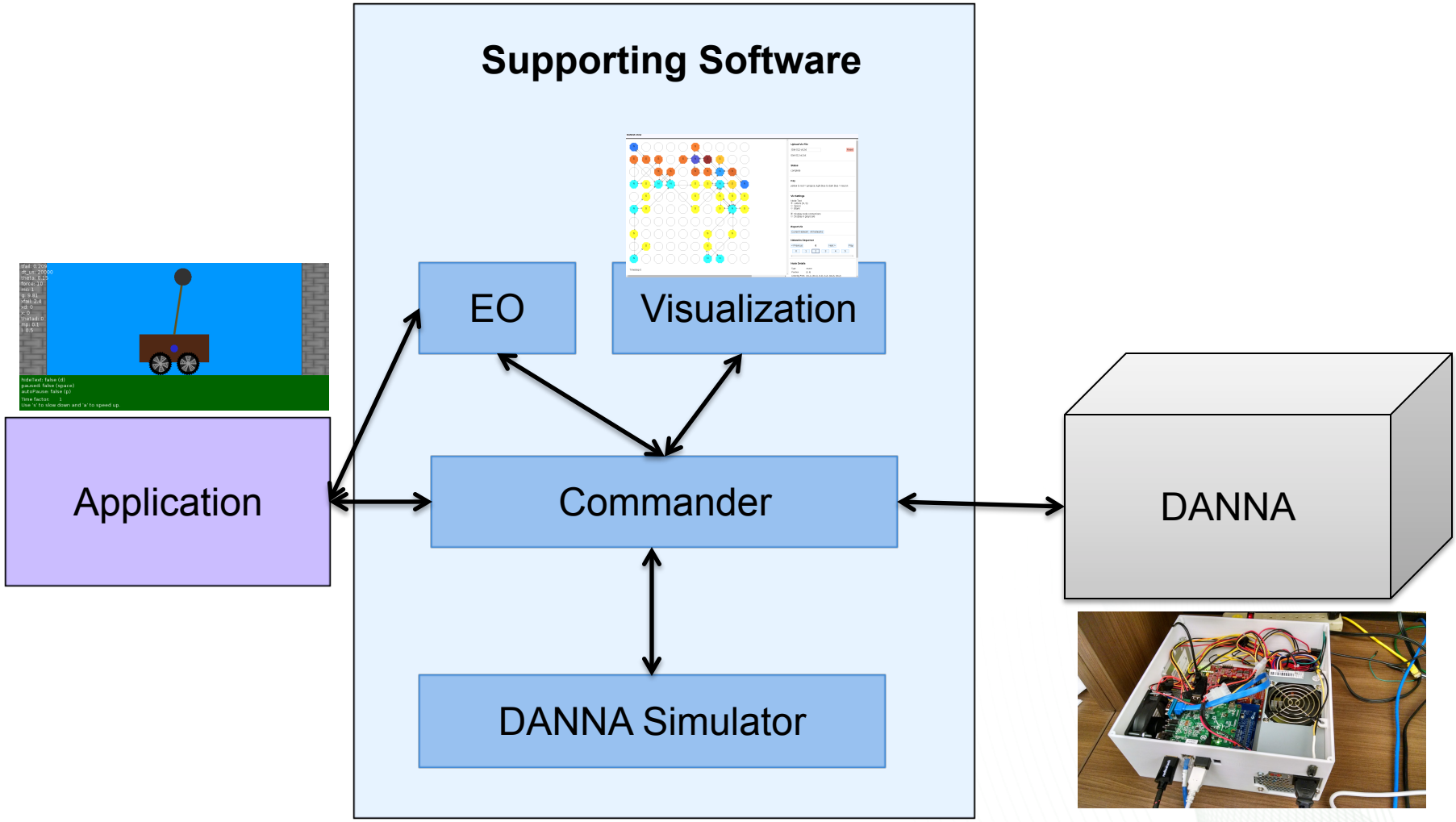


- Highly customized.
- Application-specific communications.
- Pre-training/learning.
- On-chip and on-line learning.

- More programmable.
- More flexible programming mechanisms.
- Flexible and adaptable communication schemes.

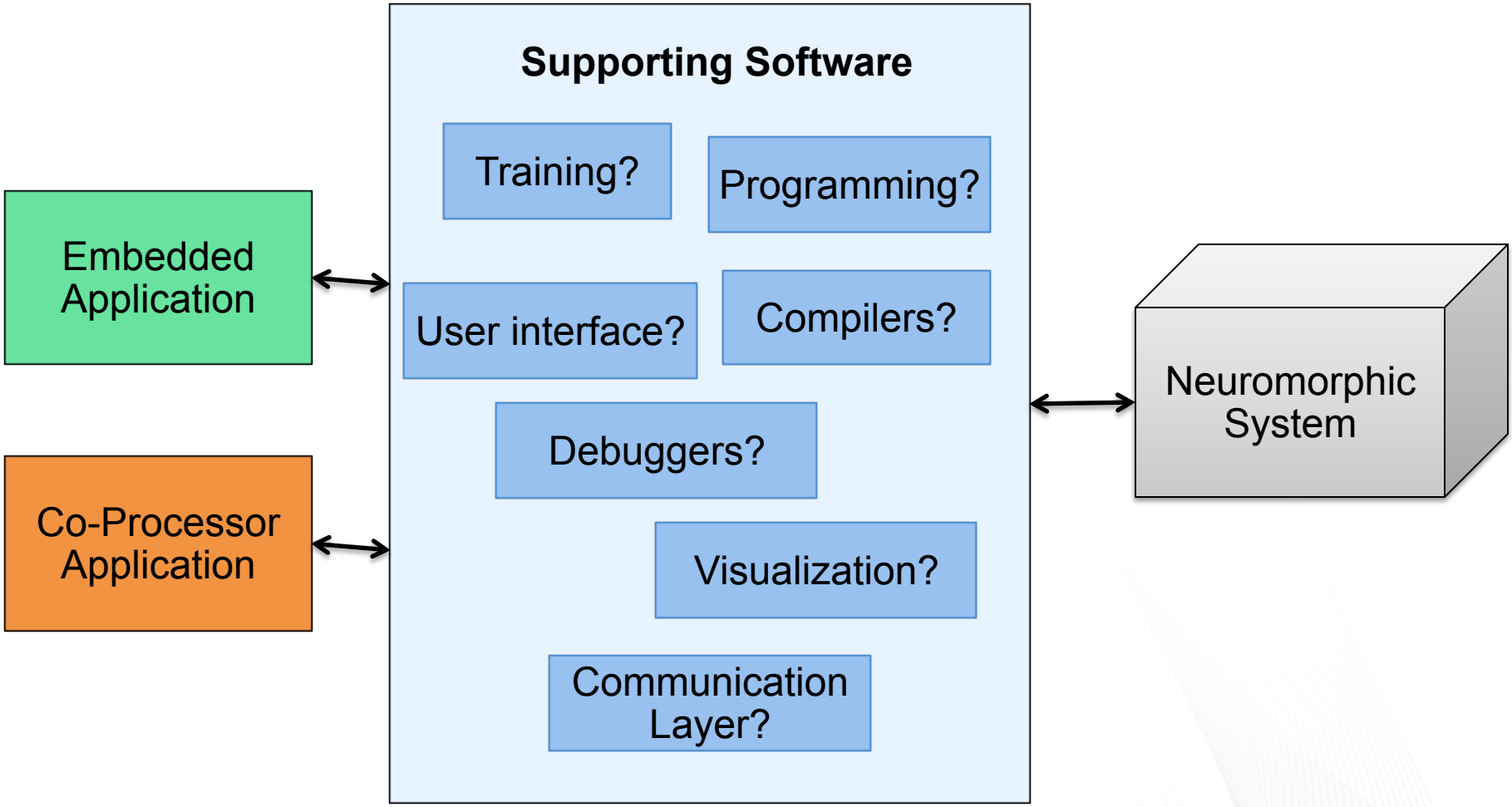


DANNA Software Components



OAK RIDGE National Laboratory

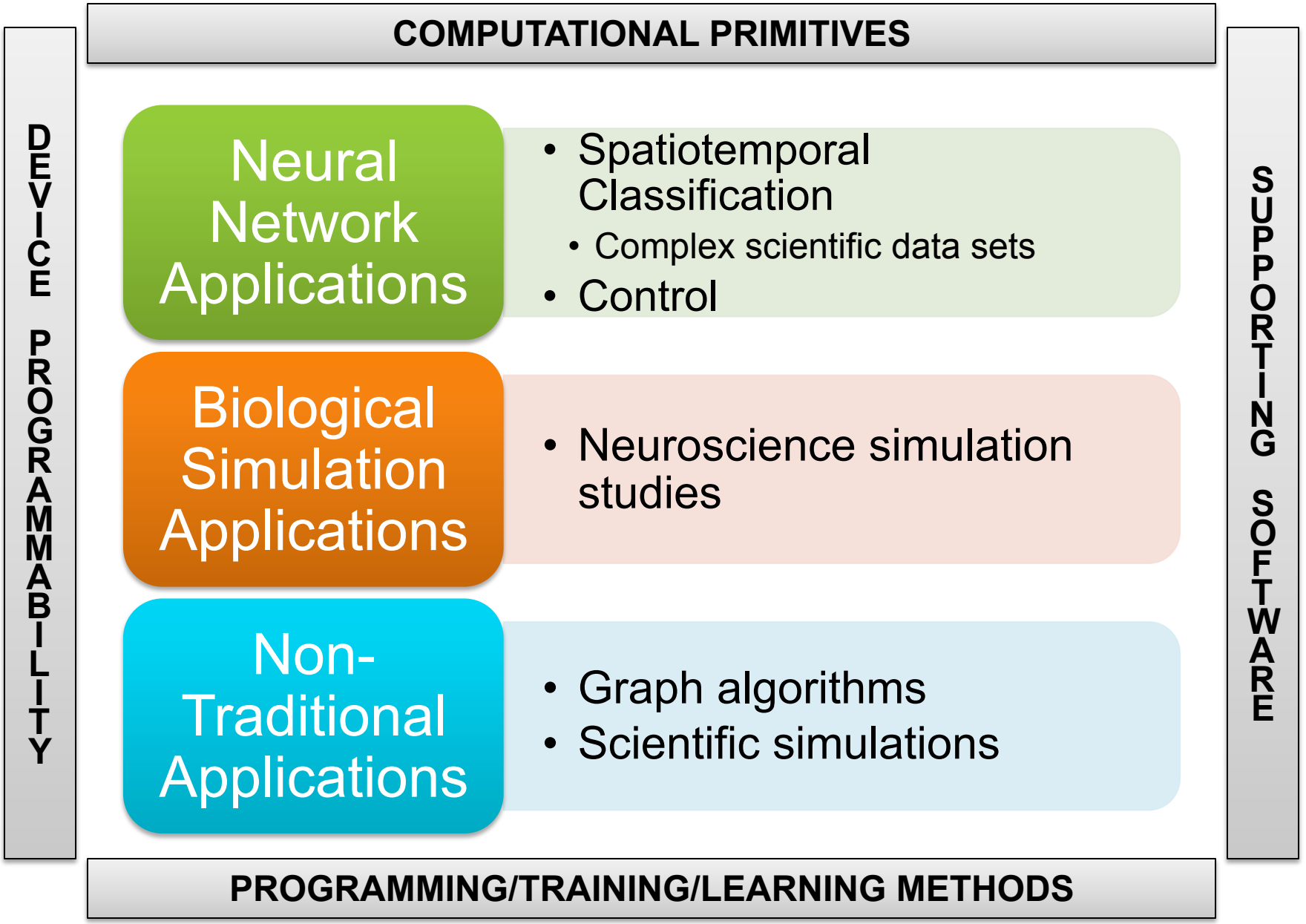
Open Questions



OAK RIDGE National Laboratory

What applications are most appropriate for neuromorphic computers?

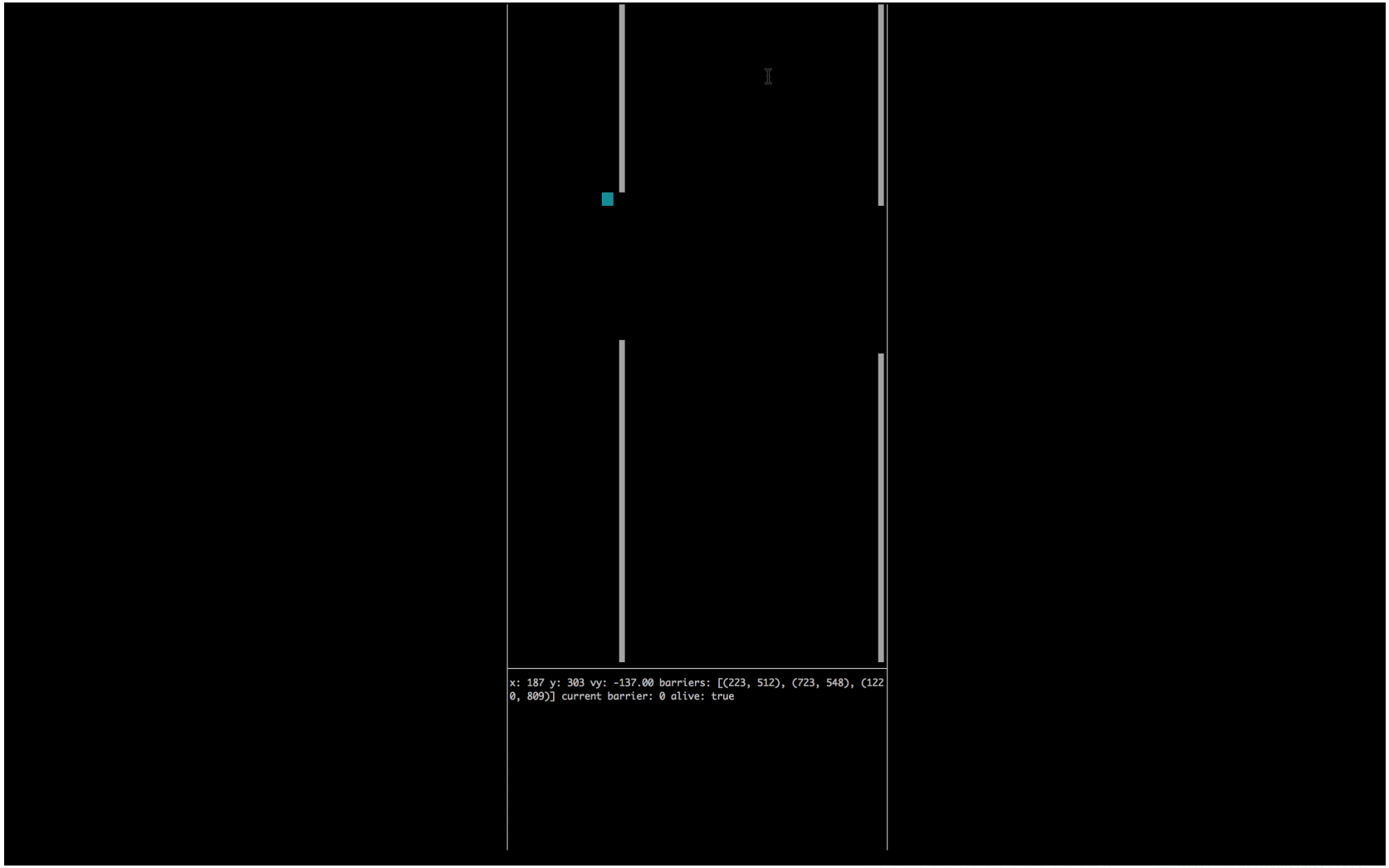
Applications



Control: Pole Balancing

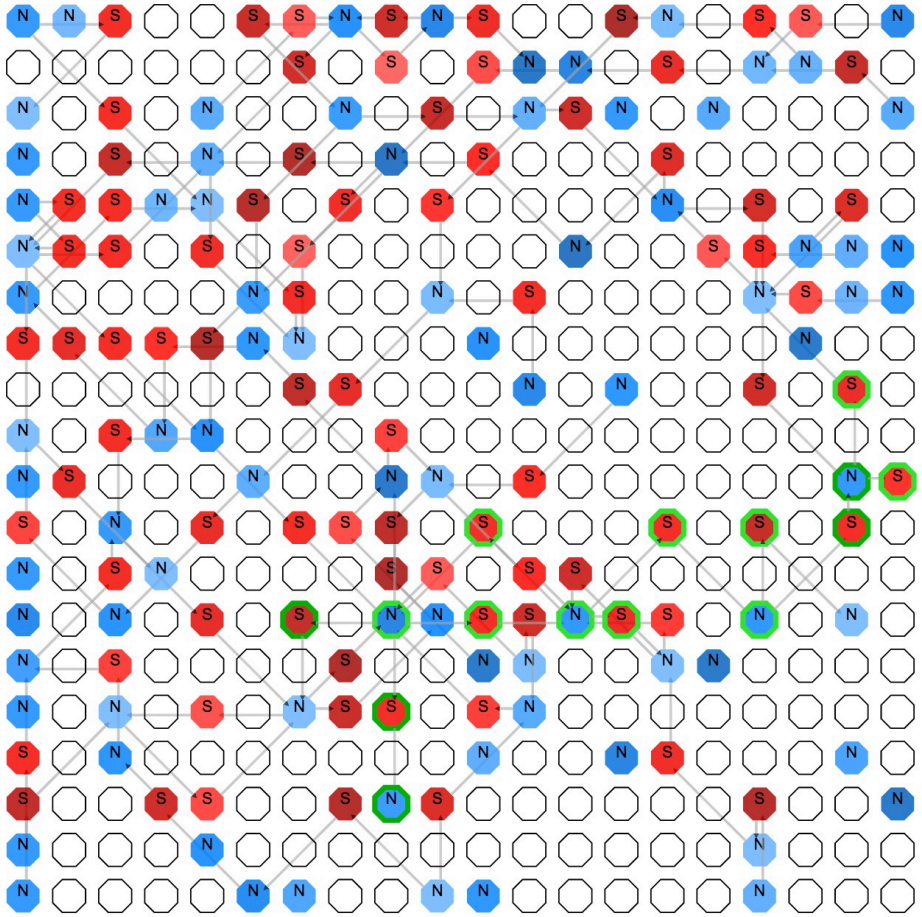
OAK RIDGE National Laboratory

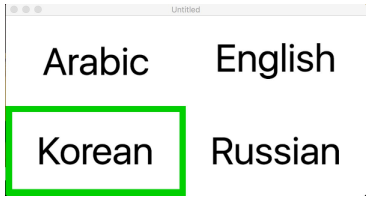
Control: One-Dimensional Navigation

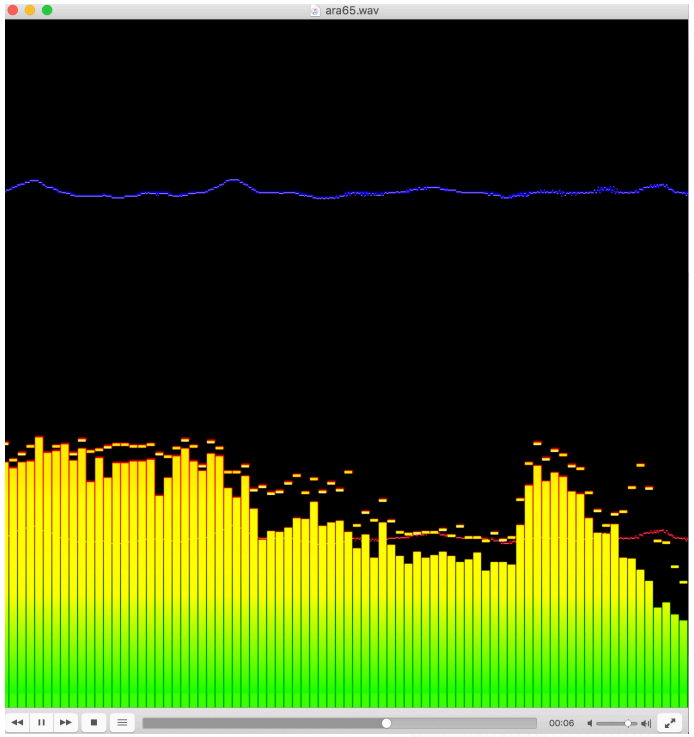


OAK RIDGE National Laboratory

Classification: Language Identification







OAK RIDGE National Laboratory

Open Questions

- What are the “killer apps”?
  - Which applications fully utilize and showcase the capabilities of neuromorphic systems?
- What can neuromorphic systems do (theoretically)?
- What should neuromorphic systems do (practically)?
- To what extent should there be different types of neuromorphic implementations for different applications?
- How do computational primitives, device programmability, programming methods, and supporting software restrict/enable applications?

OAK RIDGE National Laboratory

Moving Forward

- Large-scale software simulations (amenable for HPCs) are key for studying neuromorphic systems.
  - Study computational primitives.
  - Study device programmability implications.
  - Study programming/training/learning methods.
- Compare neuromorphic models and devices:
  - Definition of common metrics.
  - Definition of a diverse set of benchmark applications.
    - Spatiotemporal data sets.
    - Control simulations.
- Development of supporting software and systems is key for usability and accessibility.
  - Can be developed alongside simulations!

OAK RIDGE National Laboratory

Why DOE? Scientific User Facilities

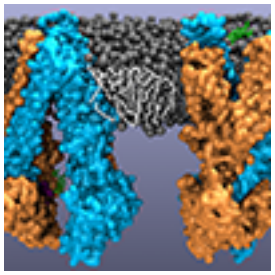
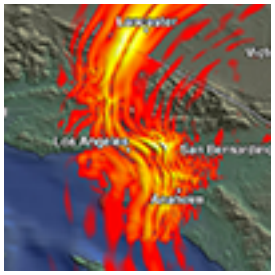
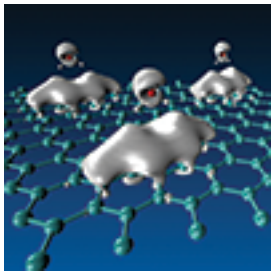
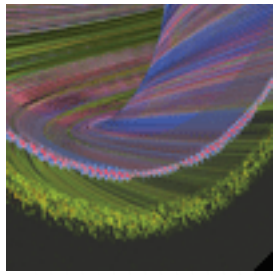
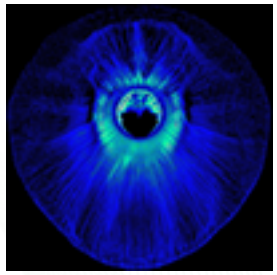
- Access to world-class HPC systems.
  - Researchers who know how to build large-scale simulation systems and supporting software.
- Access to world-class materials science user facilities.

Why DOE? People

- Collaboration opportunities with materials science and device researchers (at both national labs and affiliated universities).

Why DOE? Applications

- Widening the scope of potential applications for neuromorphic computing to include world-class science problems.
  - Climate
  - Nuclear
  - Medical
  - Materials science

Summary

- There are many fundamental questions associated with neuromorphic computing even within computer science.
  - What are the computational primitives?
  - What degree of programmability is required at the device level?
  - How do we program neuromorphic computers?
  - How do we make neuromorphic systems more usable and accessible?
  - What applications are most appropriate for neuromorphic computers?
- DOE is poised to address these questions.

OAK RIDGE National Laboratory

Workshop Agenda and Goals

- What are the fundamental **computing** questions that need to be addressed in order for neuromorphic computing to be successful as a new architecture?

Wednesday	Thursday	Friday

• Short Breakout Sessions
  • Architectures
  • Algorithms
  • Applications
	• Presentations
	• Panel Discussion
		• Breakout Sessions

OAK RIDGE National Laboratory

Thank You!

Email: schumancd@ornl.gov