Evolution at the Edge: Real-Time Evolution for Neuromorphic Engine Control

Karan P. Patel, Ethan Maness, Tyler Nitzsche,
Emma G. Brown, Brett Witherspoon, Aaron Young,
Bryan Maldonado, Brian Kaul,
James S. Plank, Catherine D. Schuman

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Outline

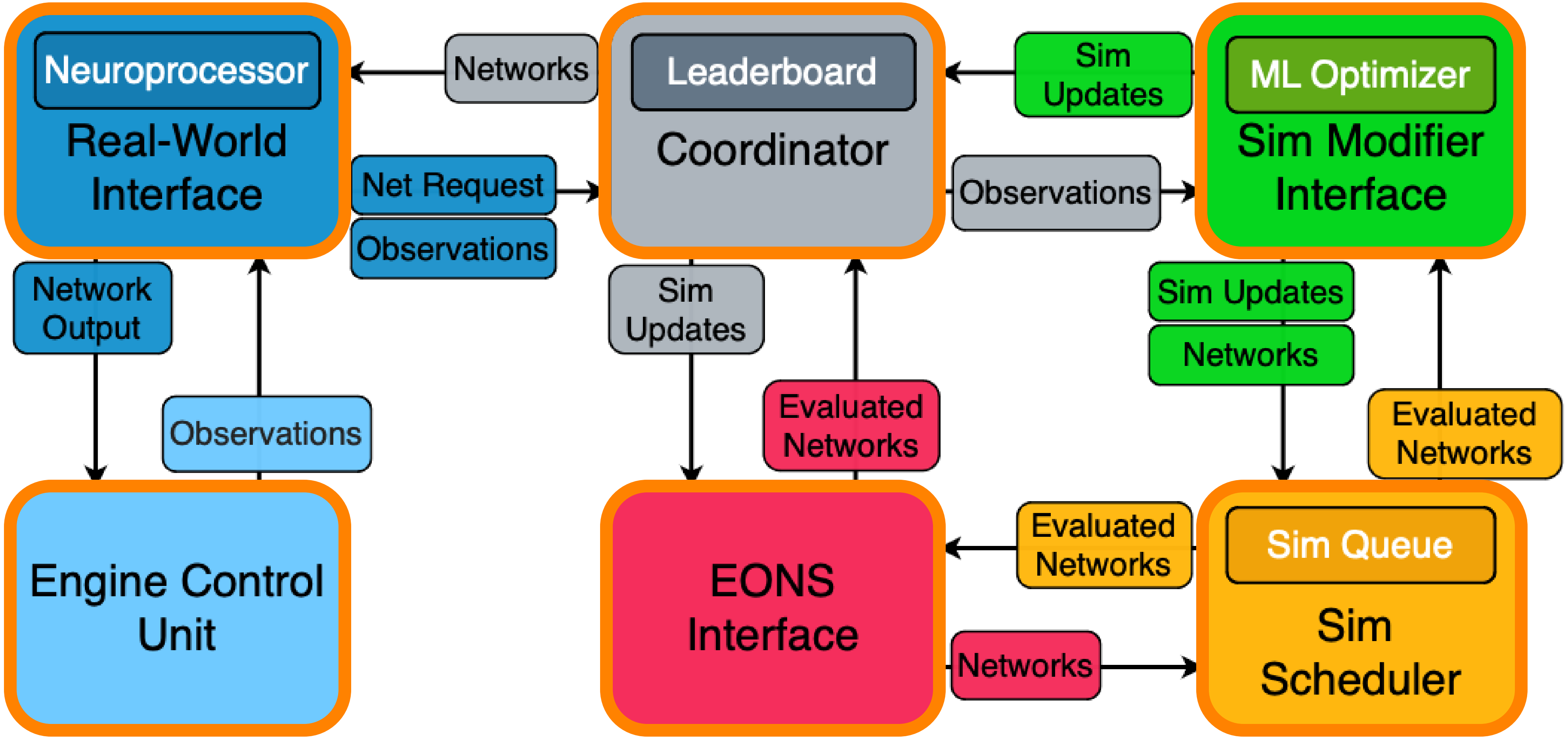
1. Introduction
2. Software Framework
3. Hardware Framework
4. Engine Control Application
5. Results
6. Conclusions & Future Works

What is the goal?

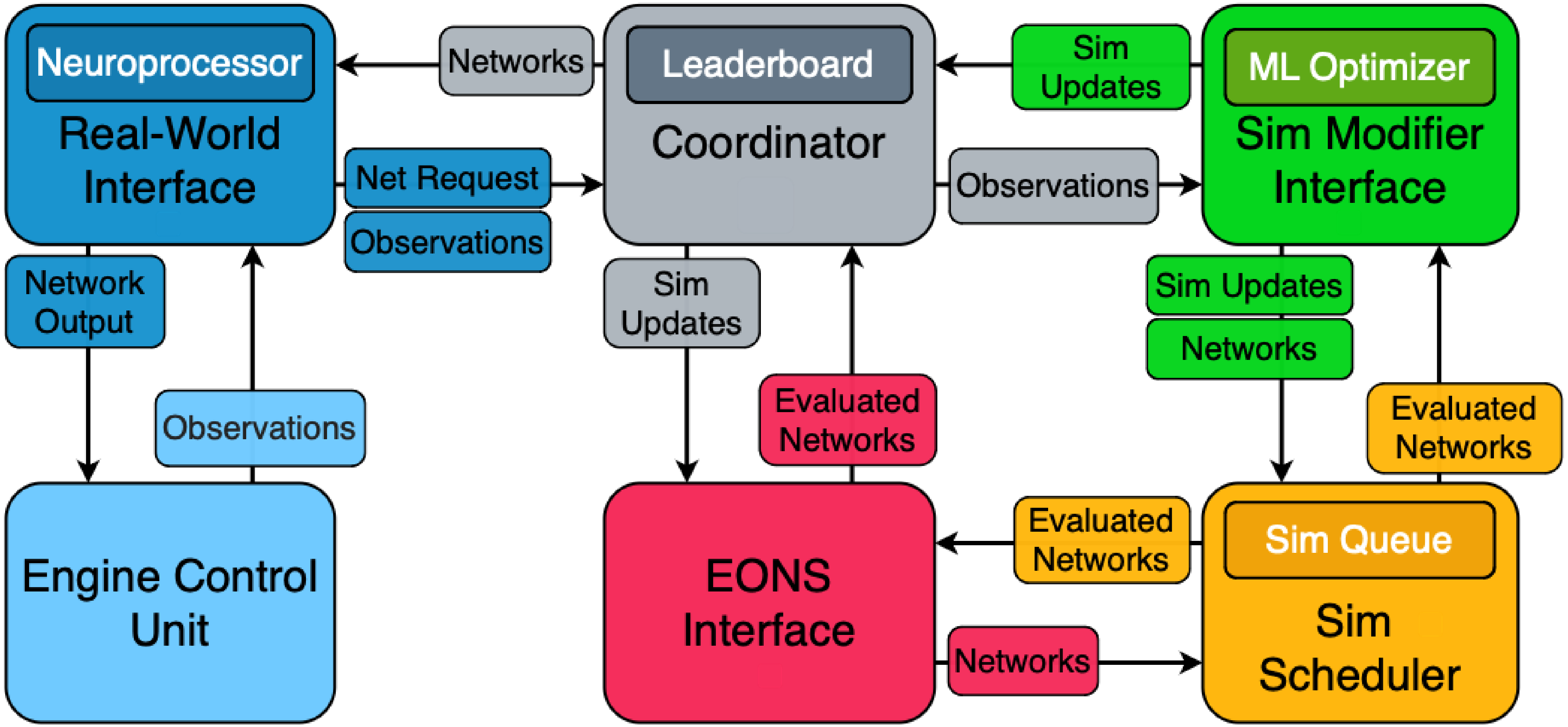
Design a system to perform real-time evolution of SNNs at the edge and apply it to a real application

- Design software framework for network training
- Design application specific simulator
- Design hardware system to process networks and interact with target application

Neuromorphic Optimization using Dynamic Evolutionary Systems

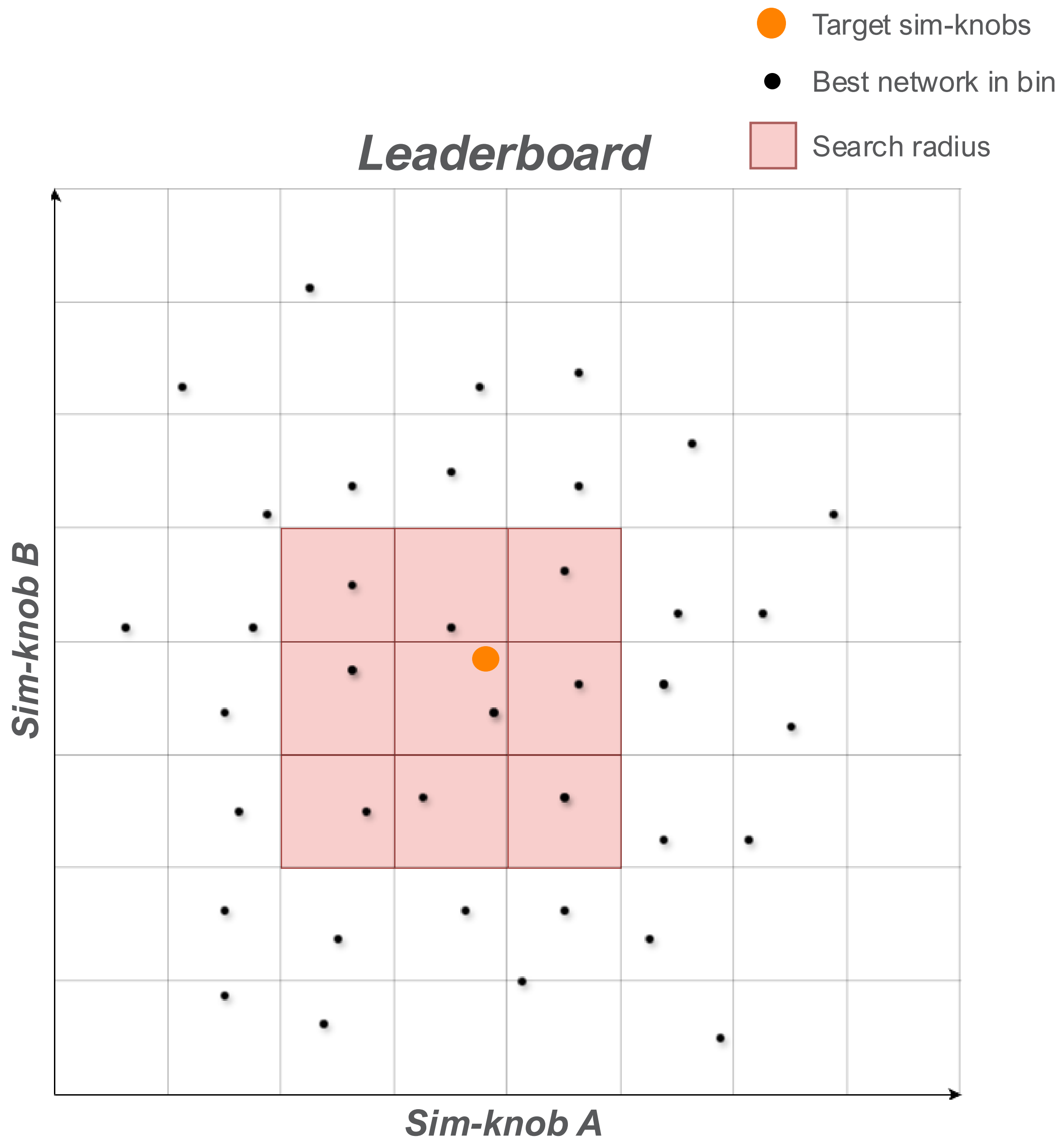


Neuromorphic Optimization using Dynamic Evolutionary Systems

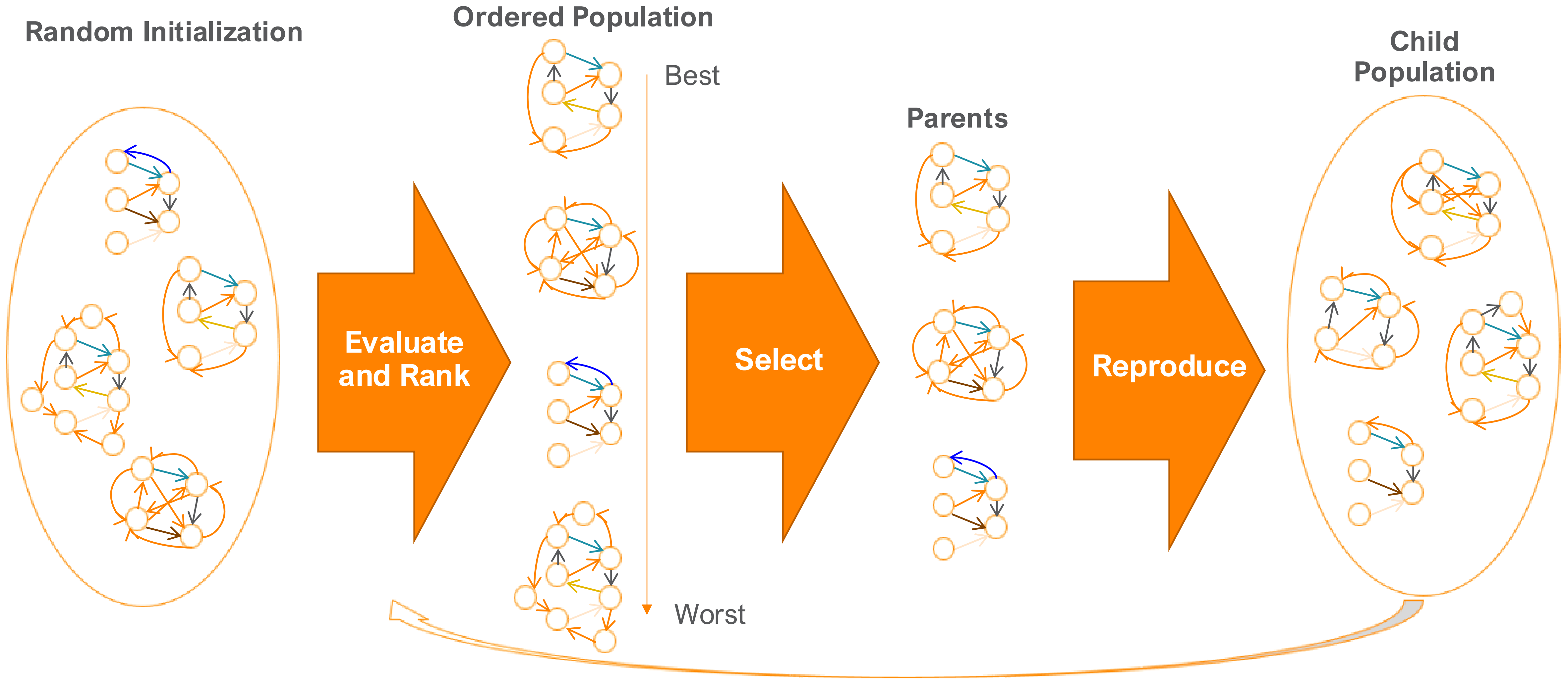


NODES: Coordinator

- Communicates between the various processes
- Coordinates data between processes
- Maintains the *Leaderboard*
  - Stores top-performing SNNs based on app-specific hyperparameters (*sim-knobs*)
  - Retrieves appropriate network based on the *sim-knobs*



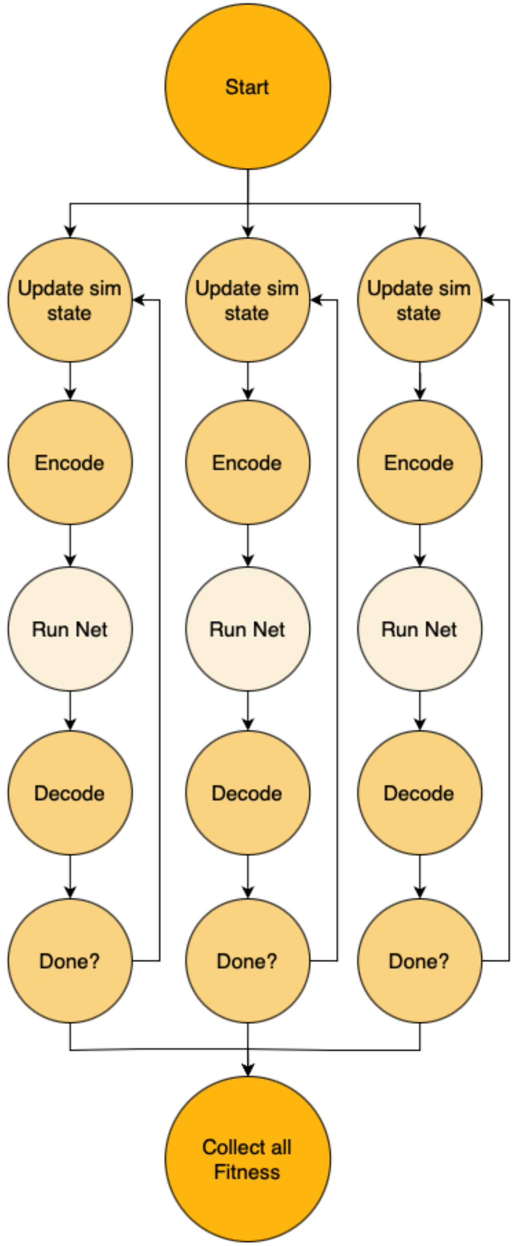
NODES: Evolutionary Optimization for Neuromorphic Systems



Schuman, Catherine D., et al. "Evolutionary optimization for neuromorphic systems." *Proceedings of the Neuro-inspired Computational Elements Workshop*. 2020.
Schuman, Catherine D., et al. "An evolutionary optimization framework for neural networks and neuromorphic architectures." 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016.

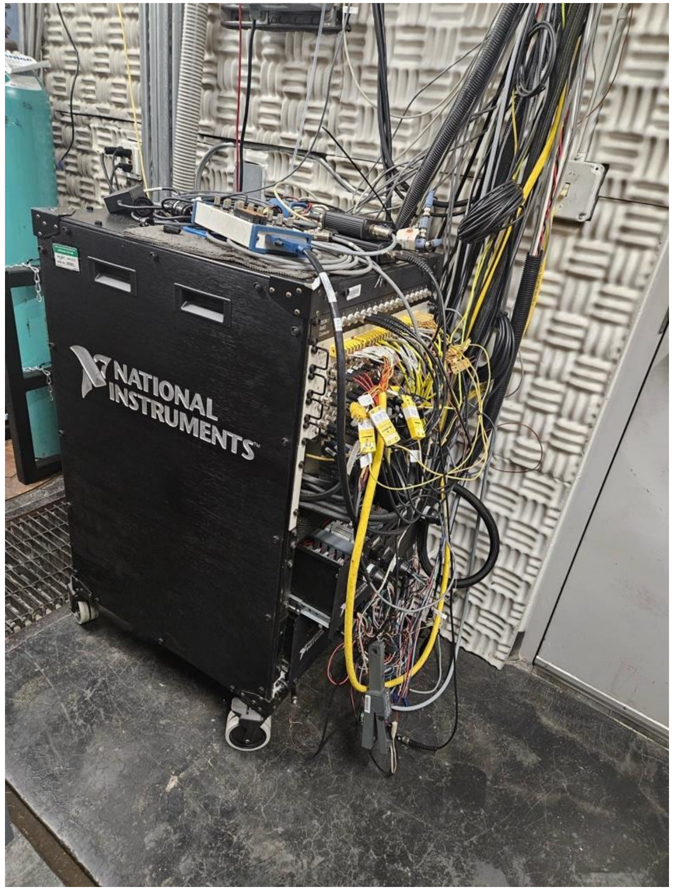
NODES: Simulation Scheduler

- Evaluate networks on various threads:
  1. Queues up networks to be evaluated (sent from ***EONS*** or ***Sim Modifier***)
  2. Load networks onto simulated/hardware neuro-processor
  3. Encode input from application simulator into spikes
  4. Process spikes throughout network
  5. Decode output spikes into application actions
  6. Apply action and step application simulation
  7. Simulation produces new input observations and calculates reward using fitness function
  8. Repeat for *n* timesteps and return network and average reward to the ***Coordinator's Leaderboard***



NODES: Real-World Interface

- Interfaces with the target application on the specified deployment environment
  - Requests network from ***Coordinator*** when the current network performs poorly
  - Loads ***Coordinator*** network onto hardware system
  - Communicates data from real-world environment to ***Sim Modifier***

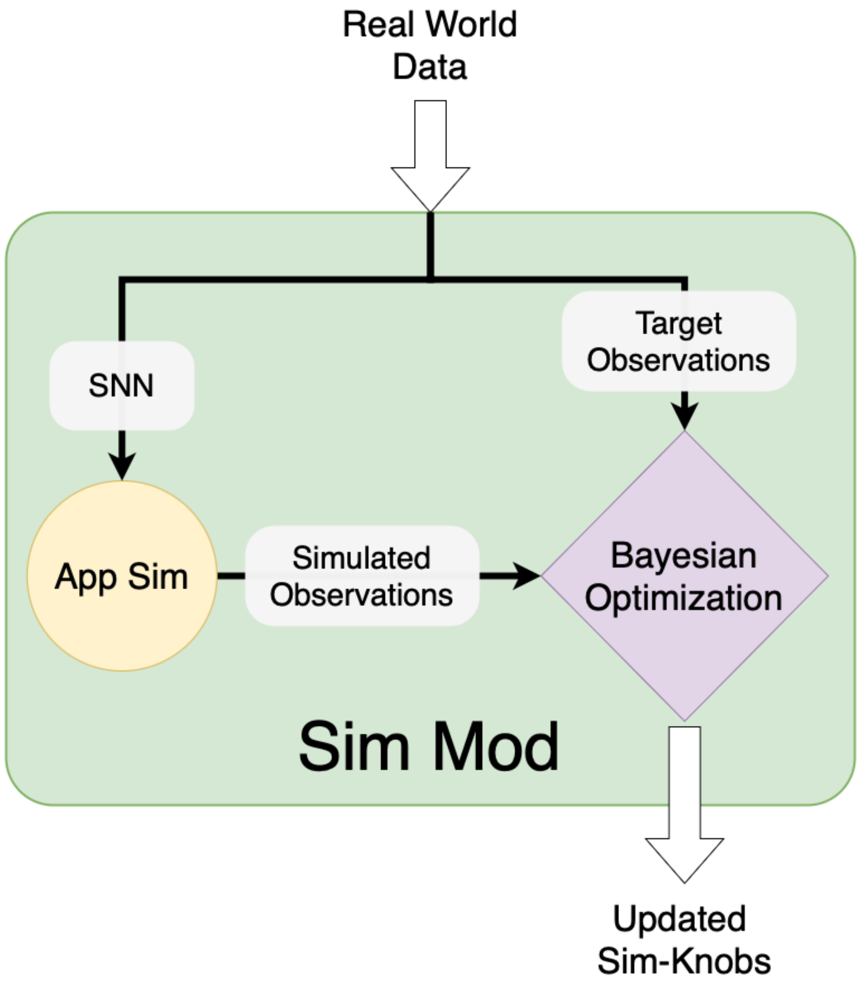


Control Box/ECU

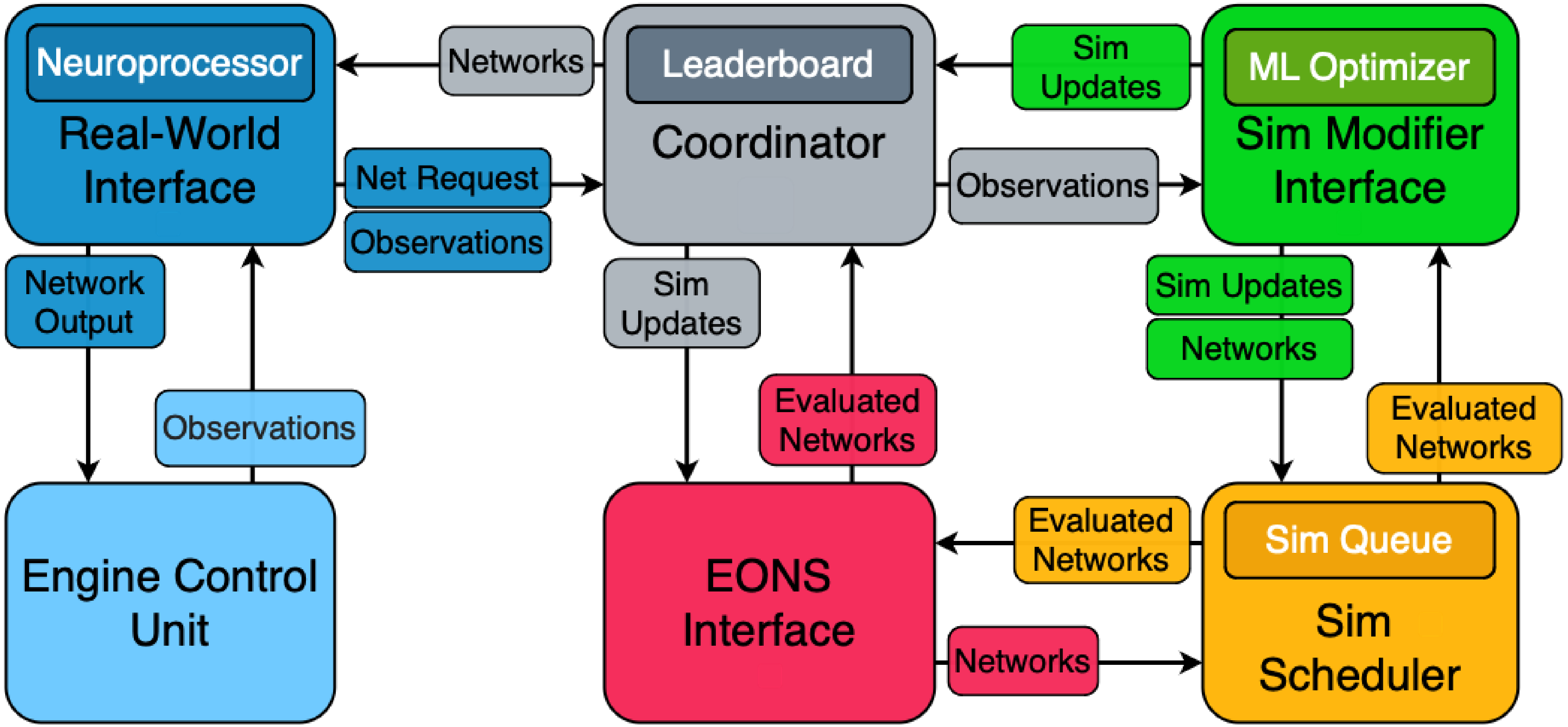
THE UNIVERSITY OF TENNESSEE KNOXVILLE

NODES: Simulation Modifier

- Takes real-world data to tune *sim-knobs* to modify training simulator to better align with deployment environment.
- Optimization process:
  1. Gathers small subset of observations from the real-world data to act as target observations
  2. Run a series of app simulations using the active SNN running on the hardware
  3. Compare the simulated observations to the target observations using MSE
  4. Apply Bayesian optimization to converge to *sim-knobs* that minimize loss
     - Finds parameters that approximate conditions that generate target observations
  5. Update EONS training simulator with new *sim-knobs*



Neuromorphic Optimization using Dynamic Evolutionary Systems

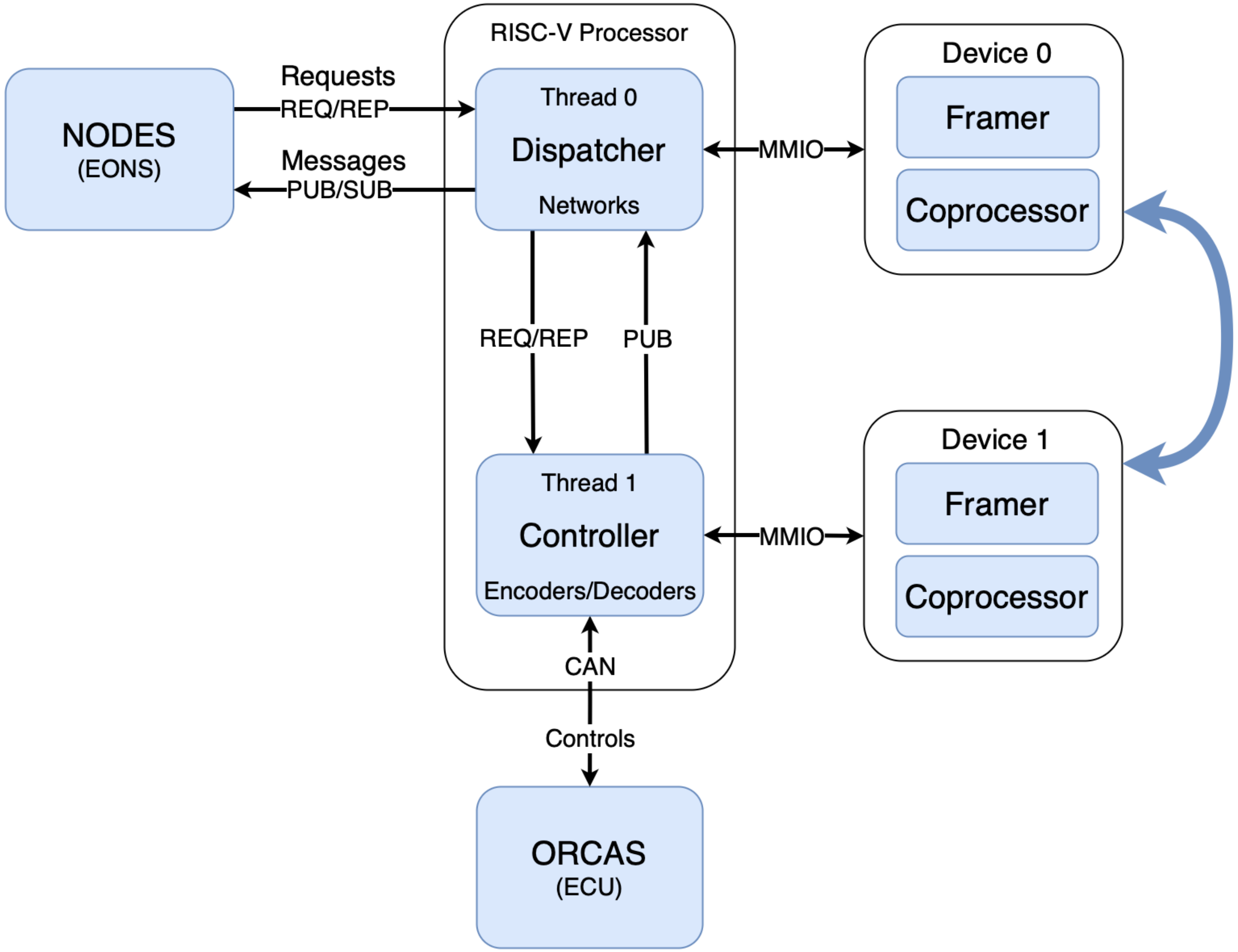


Hardware Setup: Overview

- Firebox hosts and interfaces between NODES, neuro-processors, and ECU (for diesel engine application)
- Raspberry Pi 5
  - NODES Framework
  - Application Simulation
- Beagle V-Fire
  - RISC-V processor used for real-time control and communication
  - FPGA fabric supports multiple µCaspian neuromorphic processors
  - CAN controller allows network outputs to control ECU

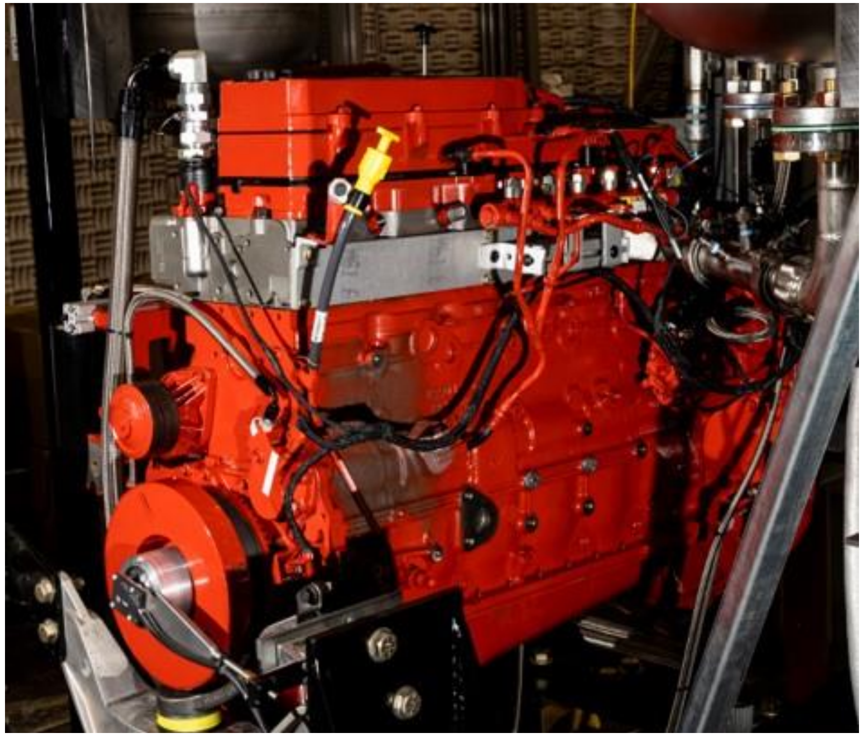
Hardware Setup: Network Swapping

- Controller
  - Performs spike encoding and decoding
  - Engine actions communicated to ECU via CAN
- Dispatcher
  - Loads networks onto inactive coprocessors
  - Swaps active coprocessors, effectively swapping networks that control the engine



Engine Control Application

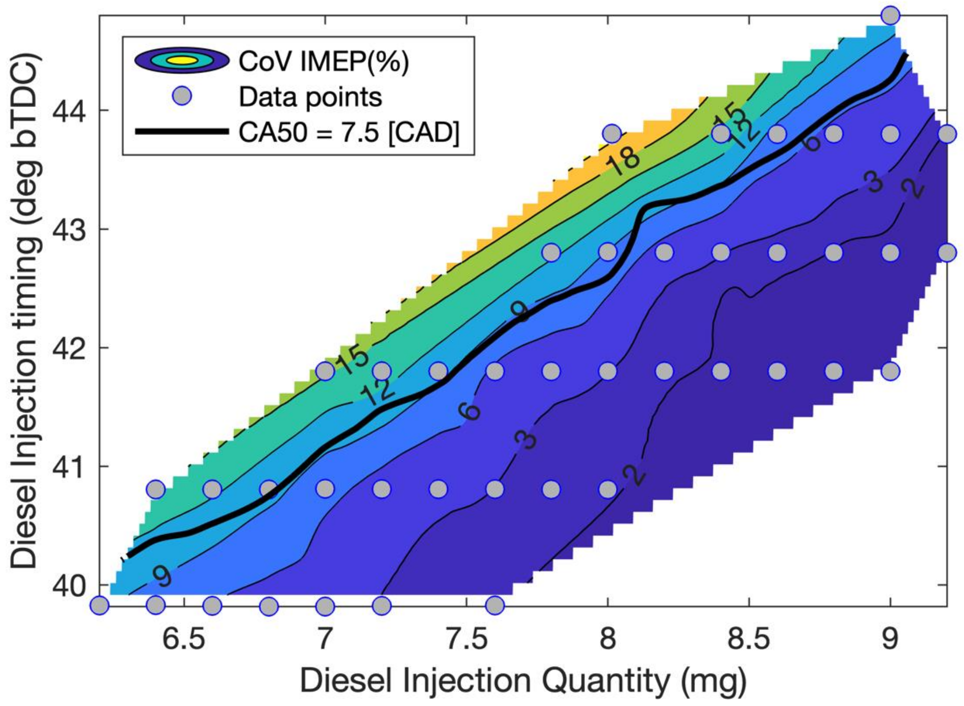
- Use neuromorphic controller to achieve stable, clean combustion in a dual-fuel (diesel and ammonia) engine
  - Ammonia-based fuel is promising in reducing carbon emissions
  - Substantially different combustion properties than traditional fuels
  - Requires adjustments of control parameters (fuel quantity, injection timing, etc.)

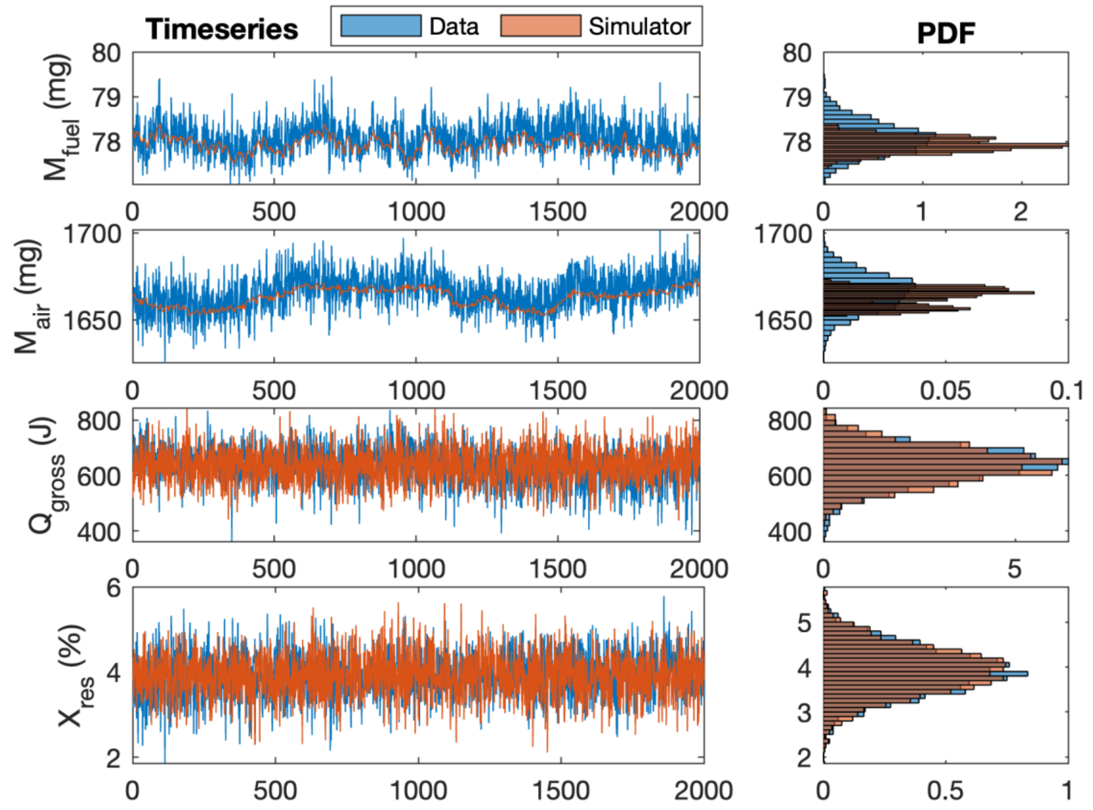


Cummins ISB 6.7L Engine

Engine Data and Simulation

- Data collected from a Cummins ISB single-cylinder engine (1200 RPM)
- 42 operating conditions were recorded to build robust combustion model





THE UNIVERSITY OF TENNESSEE KNOXVILLE

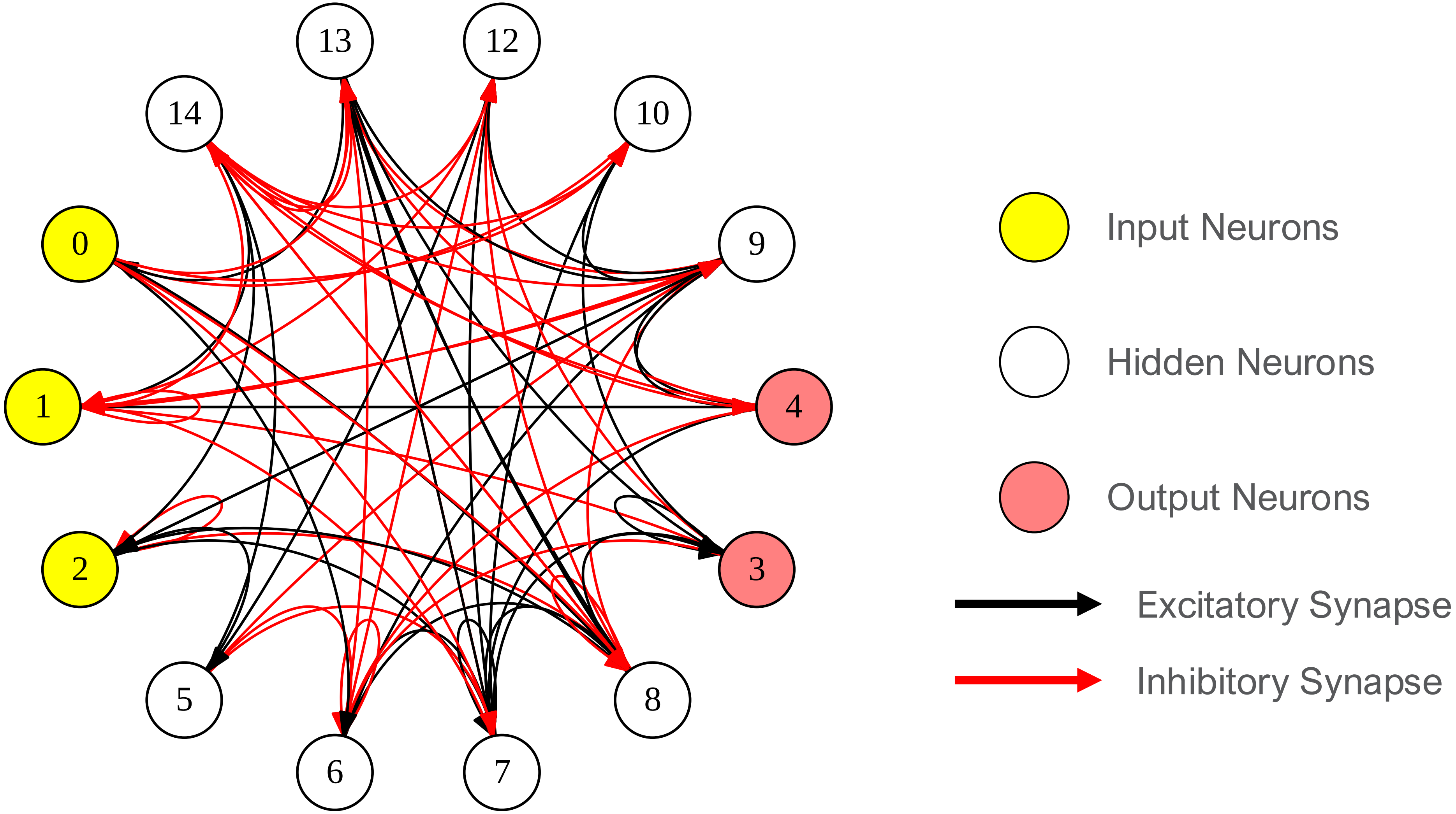
Fitness Function (Minimization)

- Network Inputs
  - M_{fuel} : Total Fuel Mass
  - M_{air} : Air Mass
  - η_c : Combustion Efficiency
- Network Outputs
  - SOI^{dsl} : Diesel Start of Injection
  - m_{in}^{dsl} : Controlled Diesel Fuel Mass
- Sim-Knobs
  - Total Fuel Mass Multiplier (0.9 – 1.1)
  - Air Mass multiplier (0.9 – 1.1)

$$cost = \overbrace{\alpha(CA50 - 7.5)^2}^{\text{Optimize Combustion Timing}} + \overbrace{(\eta_c - 1)^2}^{\text{Optimize Combustion Efficiency}}$$

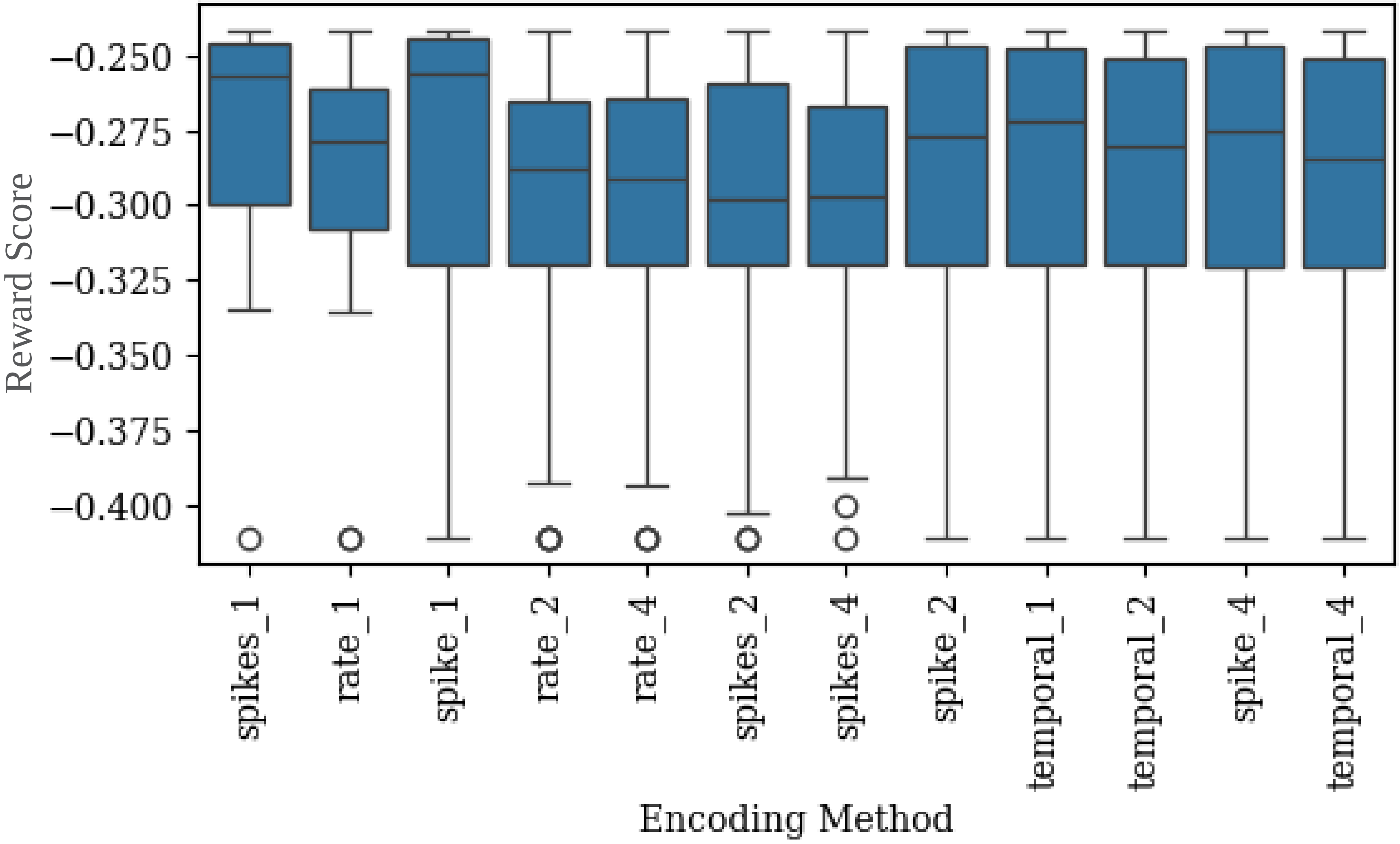
- $CA50$: Crank angle where 50% of fuel is burned
- η_c : Combustion efficiency
- α : Hyperparameter weight

Results: Offline Training



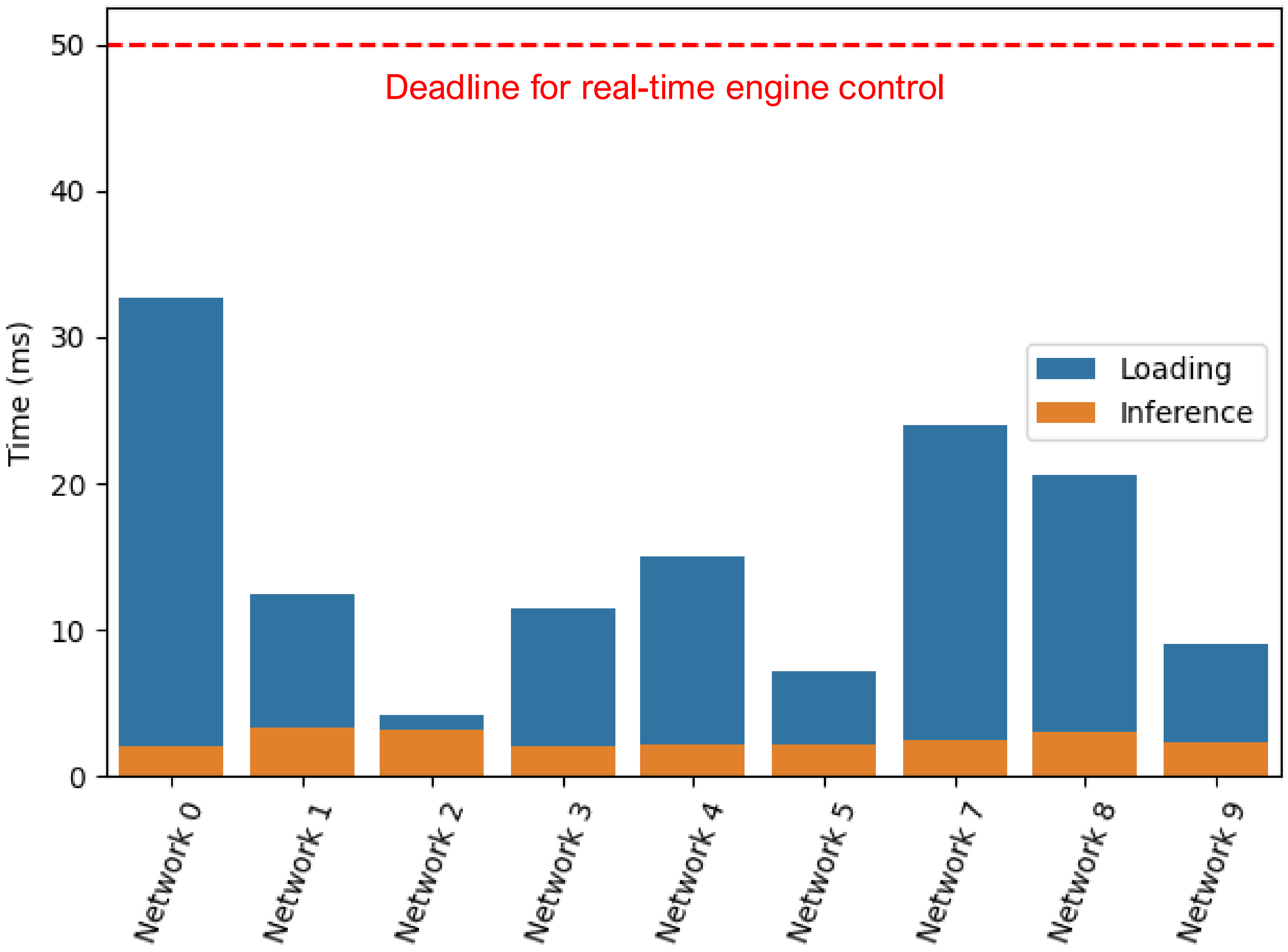
SNN trained on engine simulator with α = 0.02

Results: Hyperparameter Tuning



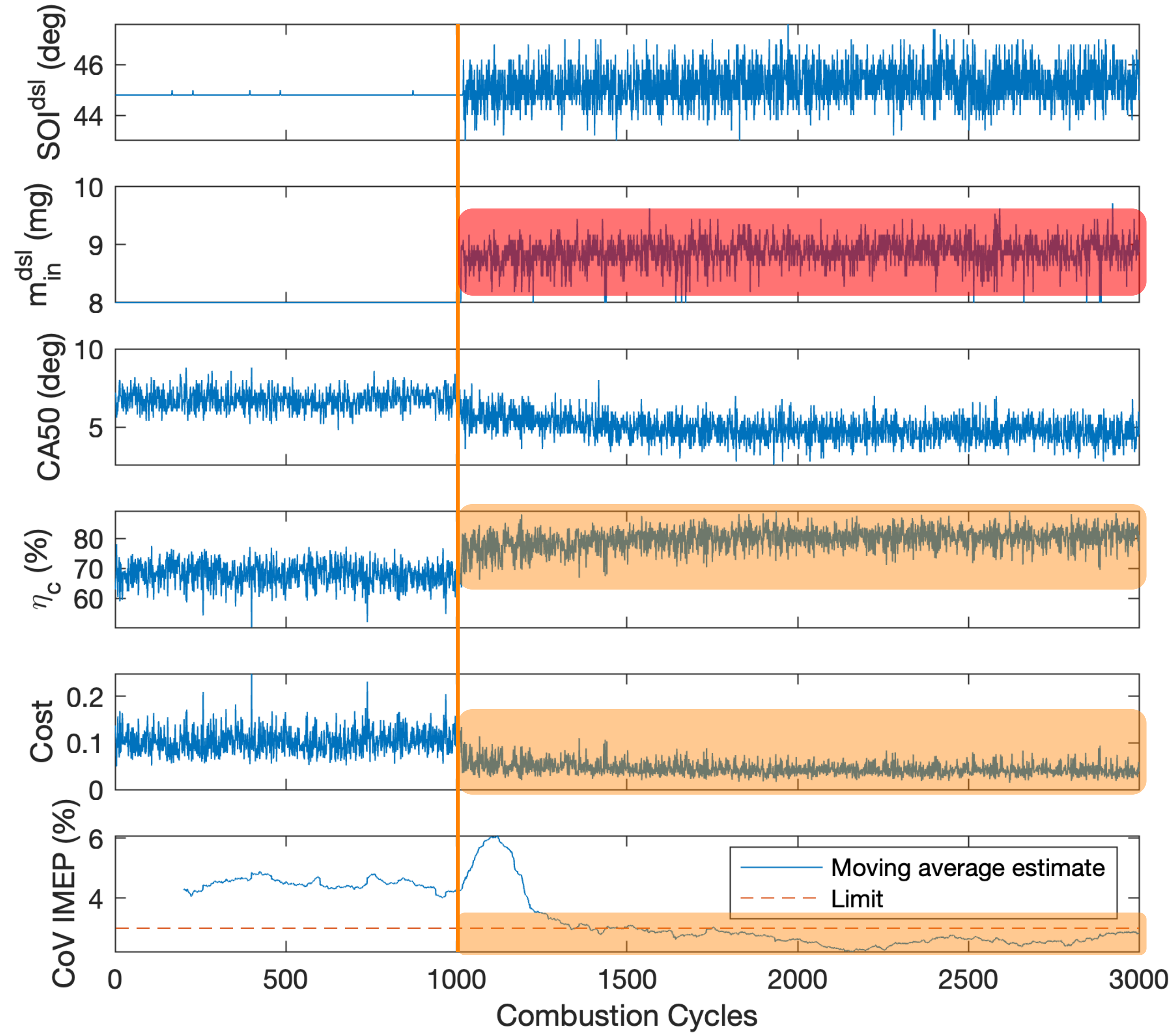
Input encoding hyperparameter tuning

Results: Load and Inference Timing



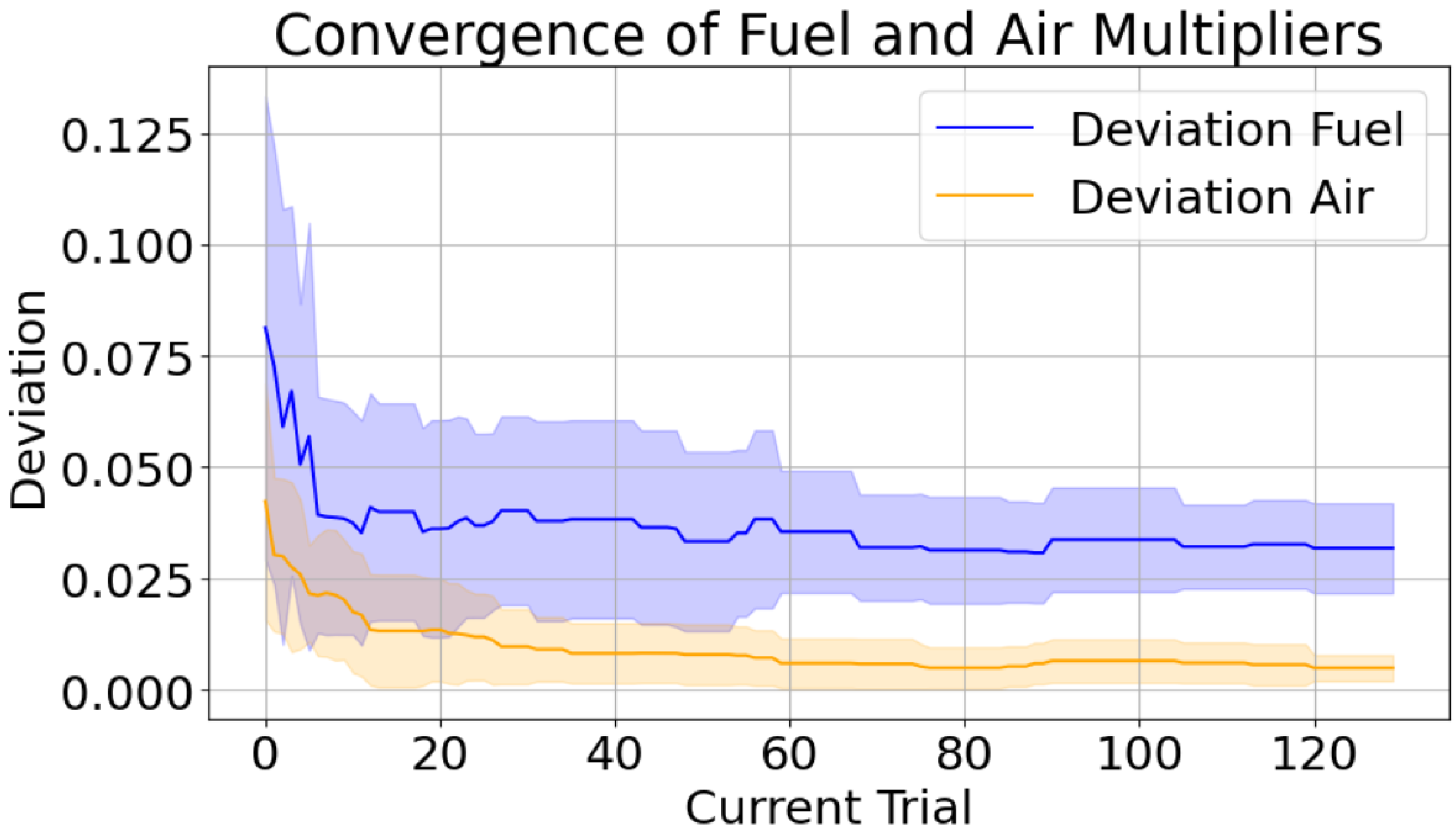
Average network loading and inference times for various networks

Results: Deployed Network Control



Experimental results of neuromorphic control (after ~1000 cycles)

Results: Sim Modifier Optimization

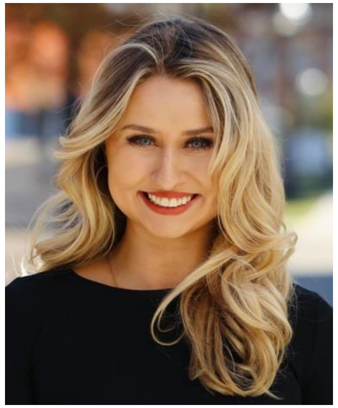


Conclusion & Future Works

- Key Takeaways:
  - Developed hardware and software infrastructure for online neuromorphic hardware system
  - Framework can be applied to various control applications
  - Successfully demonstrated the system on cycle-to-cycle dual-fuel engine control
- We intend to explore:
  - Further research cost function
  - Explore additional learning algorithms
  - Apply to different applications
  - Iterate on NODES

Acknowledgements

- This work was supported by DOE EERE grant number DE-EE0009177 and the UT-Oak Ridge Innovation Institute seed program.



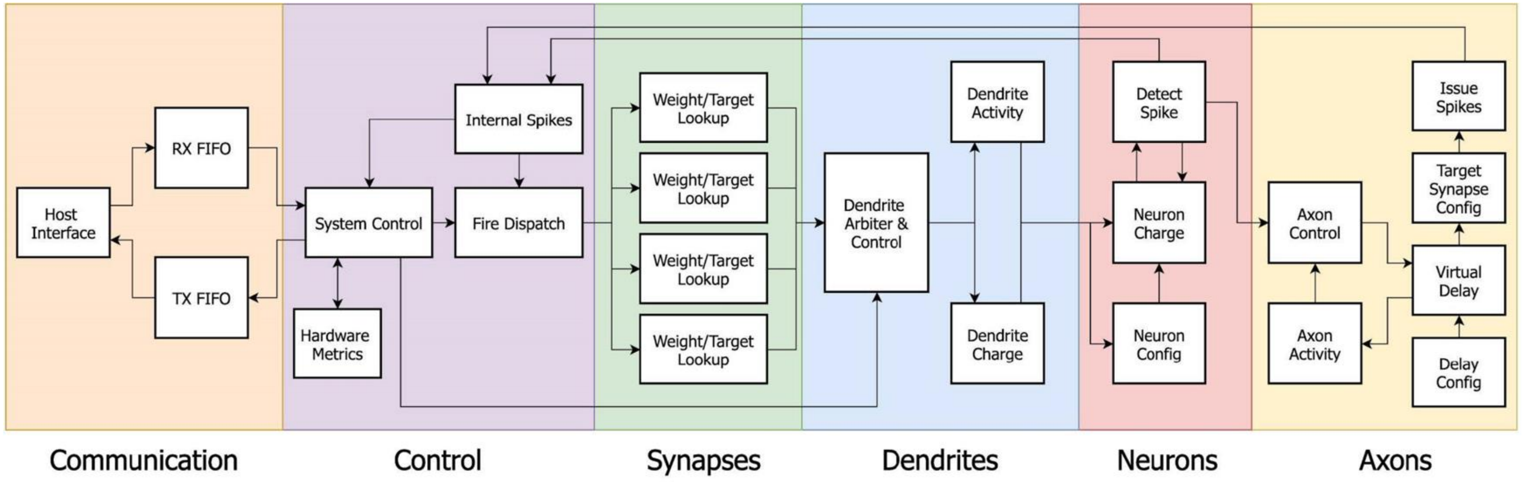


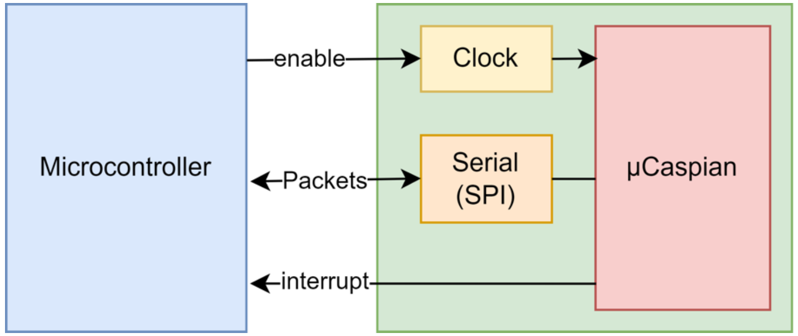


Questions?
Karan P. Patel: kpatel68@vols.utk.edu
THE UNIVERSITY OF TENNESSEE KNOXVILLE

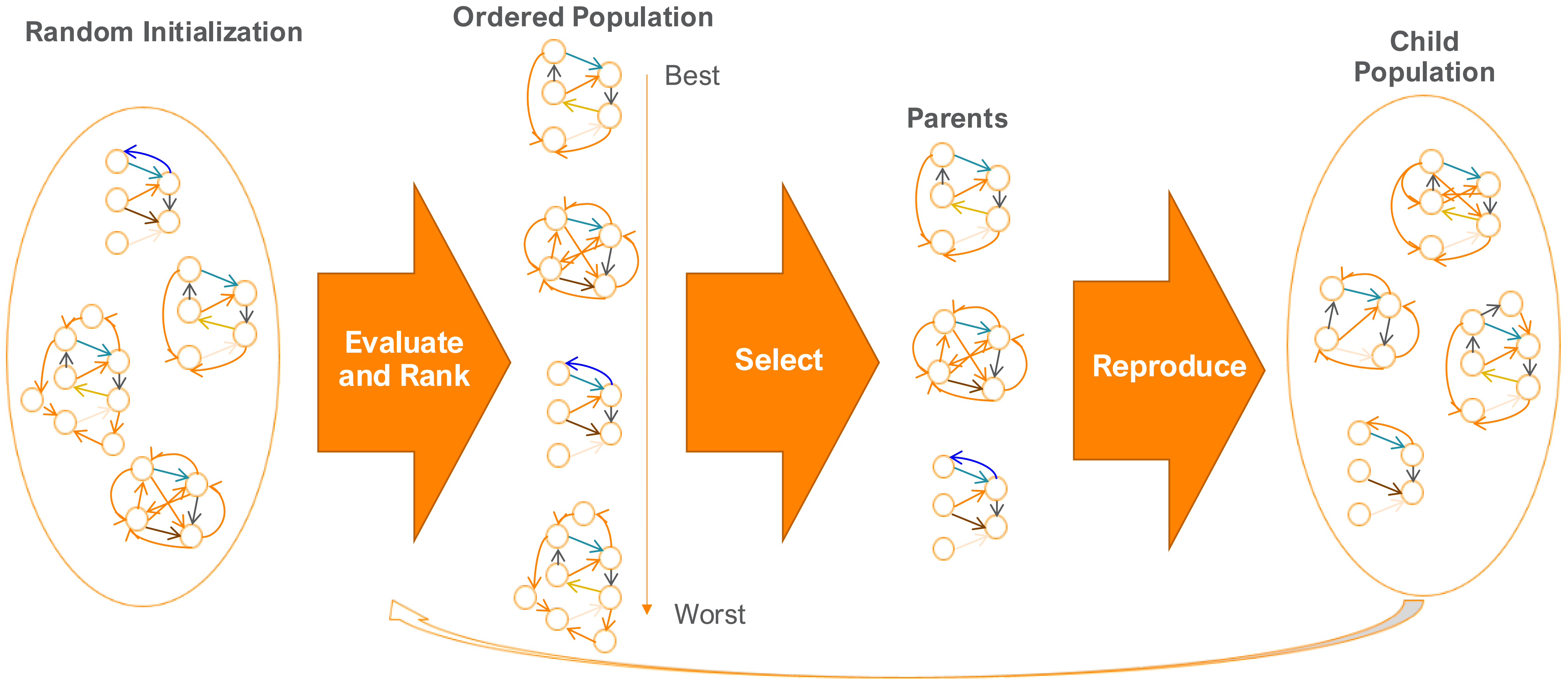
µCaspian

- Variable network cycle frequency based on activity in the network.
- Features:
  - IF-Neurons.
  - All-to-all connection support with 256 neurons and 4096 synapses.
  - Reconfigurable networks.



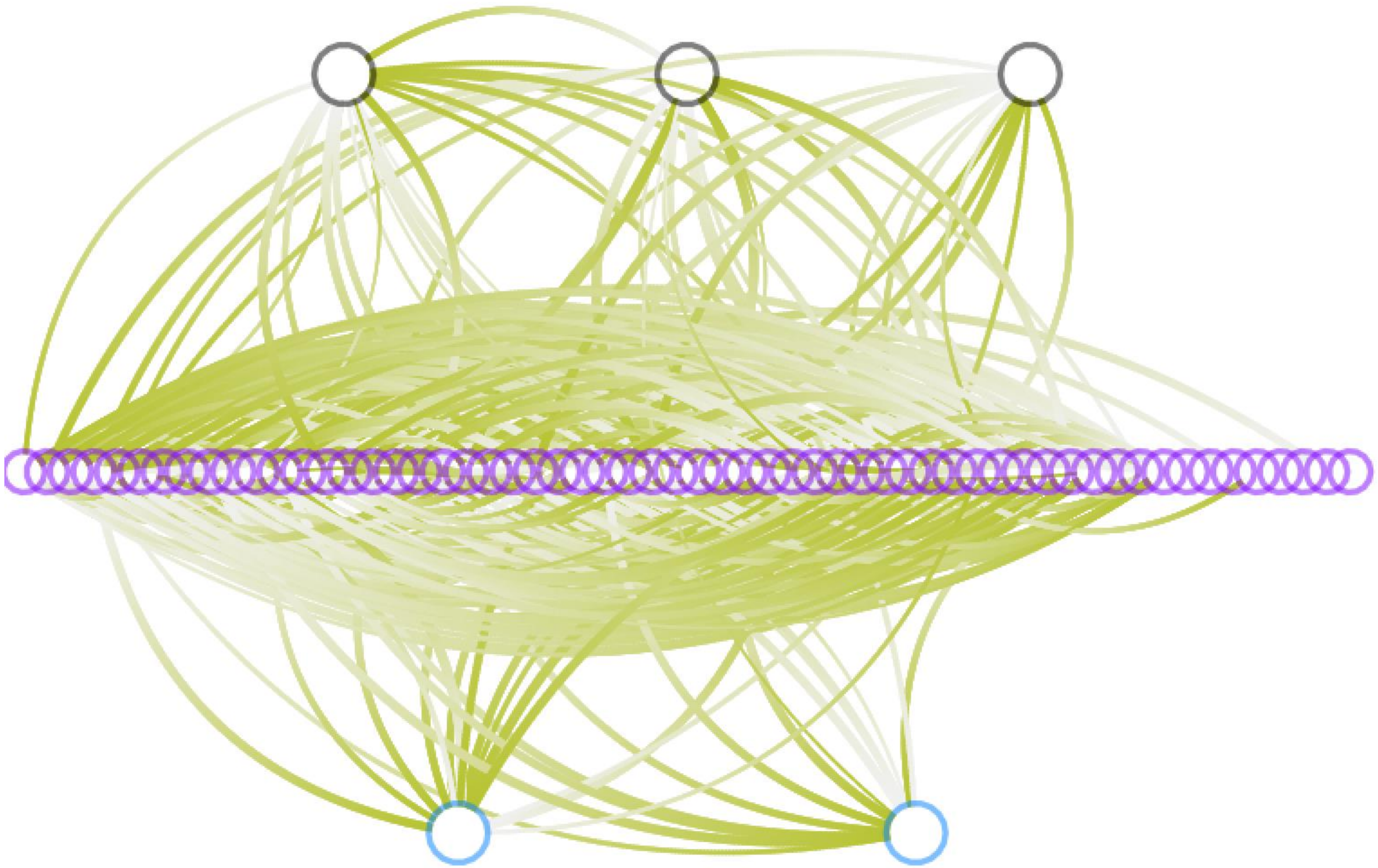


NODES: Evolutionary Optimization for Neuromorphic Systems



Schuman, Catherine D., et al. "Evolutionary optimization for neuromorphic systems." *Proceedings of the Neuro-inspired Computational Elements Workshop*. 2020.
Schuman, Catherine D., et al. "An evolutionary optimization framework for neural networks and neuromorphic architectures." 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016.

Results: Offline Training



SNN trained on engine simulator with α = 0.02