Embracing the Hairball: An Investigation of Recurrence in Spiking Neural Networks for Control

Catherine Schuman,
Charles Rizzo, Garrett Rose,
James Plank
Department of EECS

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

TENNLab

- Four PIs at UTK:
  - Dr. Ahmed Aziz (Devices)
  - Dr. Garrett Rose (Architectures and Devices)
  - Dr. Jim Plank (Software and Applications)
  - Dr. Katie Schuman (Algorithms and Applications)
- Affiliated faculty at:
  - University of Albany
  - George Mason University
  - University of Mississippi
  - Florida International University
  - Oak Ridge National Laboratory

TENN LAB
NEUROMORPHIC
ARCHITECTURES. LEARNING. APPLICATIONS.

https://neuromorphic.eecs.utk.edu/

What is the role of recurrence in spiking neural networks?

Recurrence in today's spiking neural networks

- Liquid state machines/reservoir computing
  - Sparse, recurrent randomly generated reservoirs that are not trained by the algorithm
- Winner-take-all networks
  - Used in a variety of contexts, but typically with fixed weights
- Recurrence inspired by or mapped from networks like LSTMs
  - Trained using conventional or adapted recurrent neural network approaches

Training recurrent neural networks is hard!

Training recurrent neural networks is hard!

But…is it worth it?

How do we evaluate recurrence in SNNs?

- We wanted to evaluate across multiple control applications in a simple SNN simulator
- Why control?
  - Control applications usually benefit from some form of memory, though not all actually require memory to be successful
- We wanted to compare feed-forward networks with recurrent neural networks where any two neurons could be connected by a synapse

TENNLab Neuromorphic Software Framework

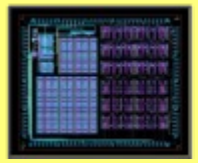
Software Core: Common Interfaces, Input/Output Coding, Network Compiler

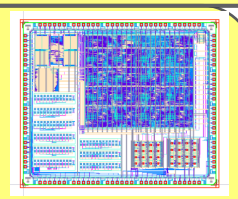
Plank, J. S., Schuman, C. D., Bruer, G., Dean, M. E., & Rose, G. S. (2018). The TENNLab exploratory neuromorphic computing framework. *IEEE Letters of the Computer Society*, *1*(2), 17-20.

TENNLab Neuromorphic Software Framework

Software Core: Common Interfaces, Input/Output Coding, Network Compiler

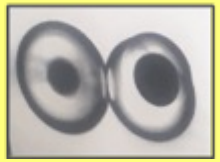
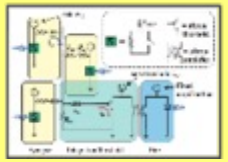
Architectures/Devices





mrDANNA DANNA2 RAVENS

Our Hardware


Biom SOENS Caspian

Collaborators' Hardware

Simulators: RISP, Brian, NEST, BindsNET, Nengo, NengoLoihi

Plank, J. S., Schuman, C. D., Bruer, G., Dean, M. E., & Rose, G. S. (2018). The TENNLab exploratory neuromorphic computing framework. *IEEE Letters of the Computer Society*, *1*(2), 17-20.

Reduced Instruction Spiking Processor (RISP)

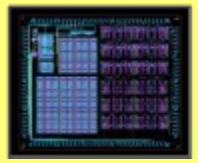
- RISP is implemented with a C++ discrete event simulator.
- Simple leaky integrate and fire neuron is implemented with programmable leak and thresholds.
  - We turn off leak on the RISP neurons
  - Neuron thresholds vary between -1 and 1.
- RISP's synapses have programmable weights and delays.
  - RISP synapse weights to vary between -1 and 1.
  - Synaptic delay values are integers between 1 and 5.

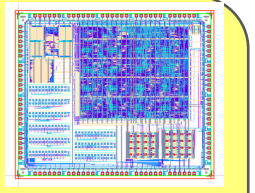
TENNLab Neuromorphic Software Framework

Algorithms: EONS, LEAP, Whetstone, Supervised STDP, Decision Tree, Reservoir

Software Core: Common Interfaces, Input/Output Coding, Network Compiler

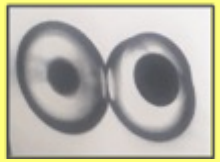
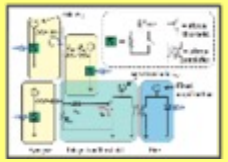
Architectures/Devices





mrDANNA DANNA2 RAVENS

Our Hardware


Biom SOENS Caspian

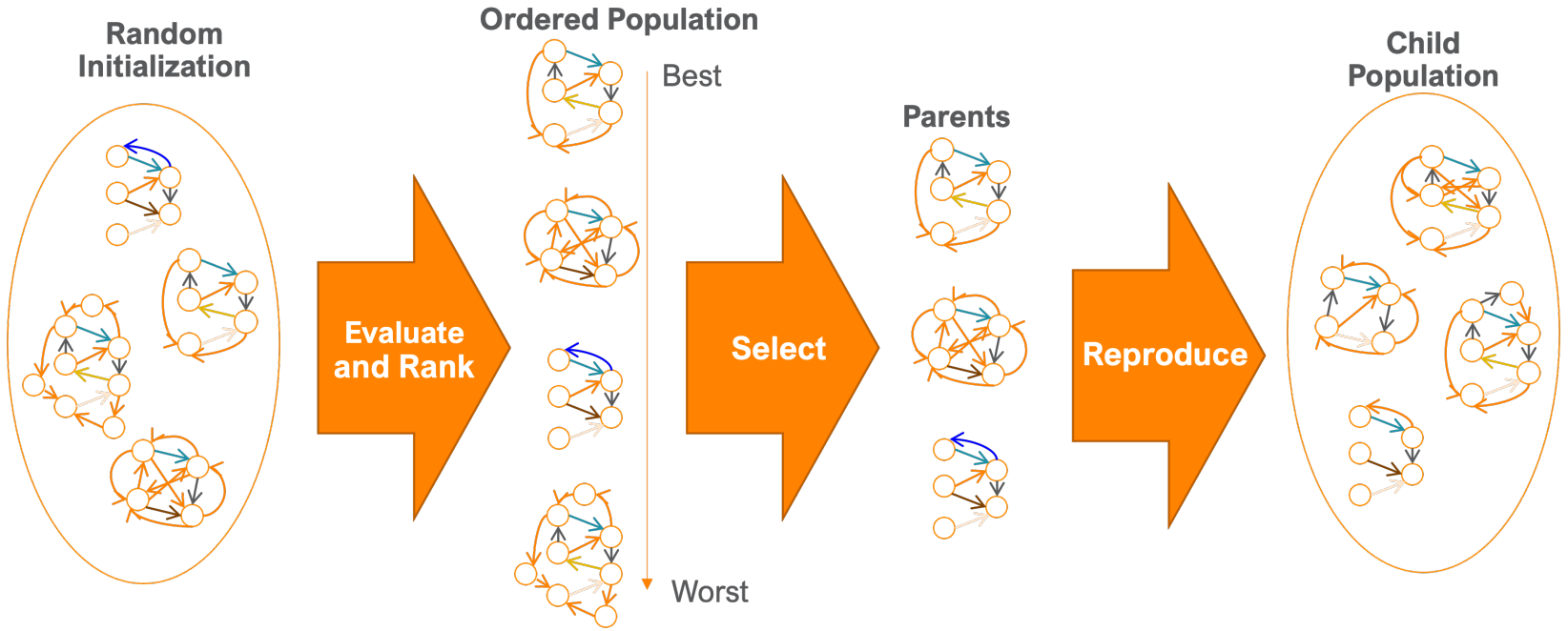
Collaborators' Hardware

Simulators: RISP, Brian, NEST, BindsNET, Nengo, NengoLoihi

Plank, J. S., Schuman, C. D., Bruer, G., Dean, M. E., & Rose, G. S. (2018). The TENNLab exploratory neuromorphic computing framework. *IEEE Letters of the Computer Society*, *1*(2), 17-20.

Evolutionary Optimization for Neuromorphic Systems

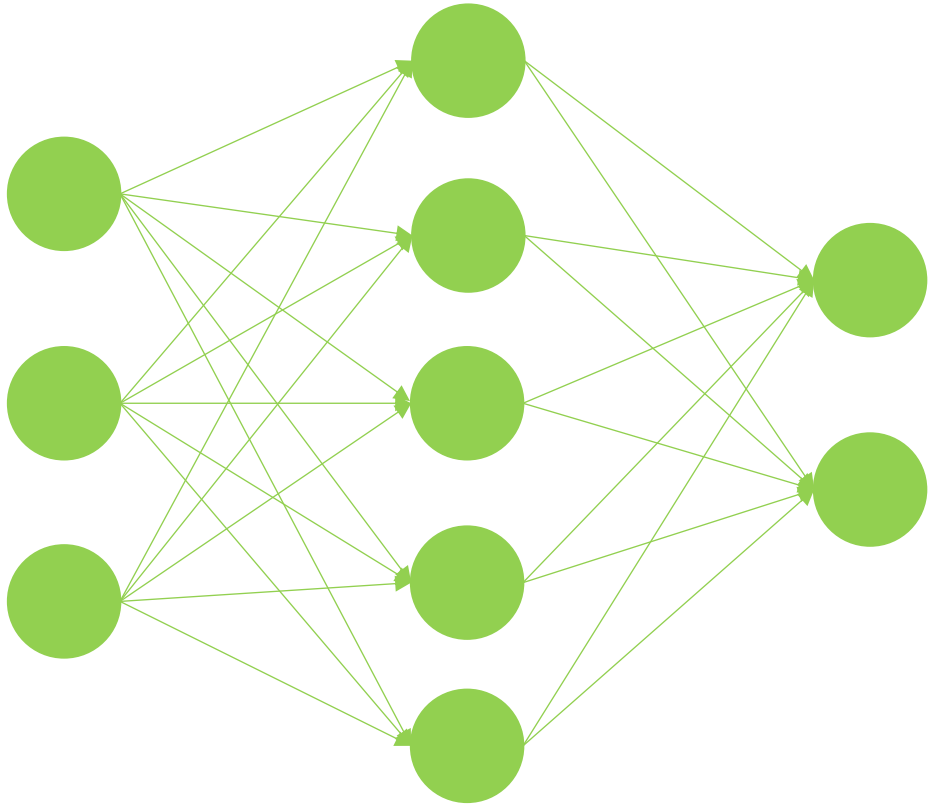
- Evolved structure, evolved parameters
- Evolve with the following number of starting hidden neurons: 0, 1, 2, 5, 10
- Do not allow for EONS to add hidden neurons, but they can be deleted



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.

Library for Evolutionary Algorithms in Python

- Fixed structure, evolved parameters
- Same inputs/outputs as EONS evolved networks for each application
- Evaluate with the following numbers of hidden neurons: 1, 5, 10, 20, and 50



Coletti, Mark A., Eric O. Scott, and Jeffrey K. Bassett. "Library for evolutionary algorithms in Python (LEAP)." In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 1571-1579. 2020.

TENNLab Neuromorphic Software Framework

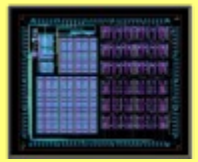
Static Data Classification | Temporal Data Classification | Event Detection | Control | General Computation

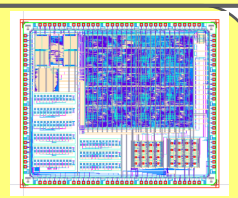
Applications

Algorithms: EONS, LEAP, Whetstone, Supervised STDP, Decision Tree, Reservoir

Software Core: Common Interfaces, Input/Output Coding, Network Compiler

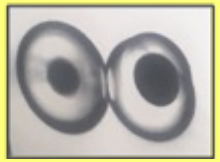
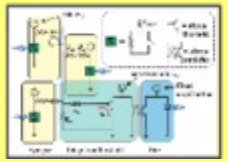
Architectures/Devices





mrDANNA **DANNA2** **RAVENS**

Our Hardware

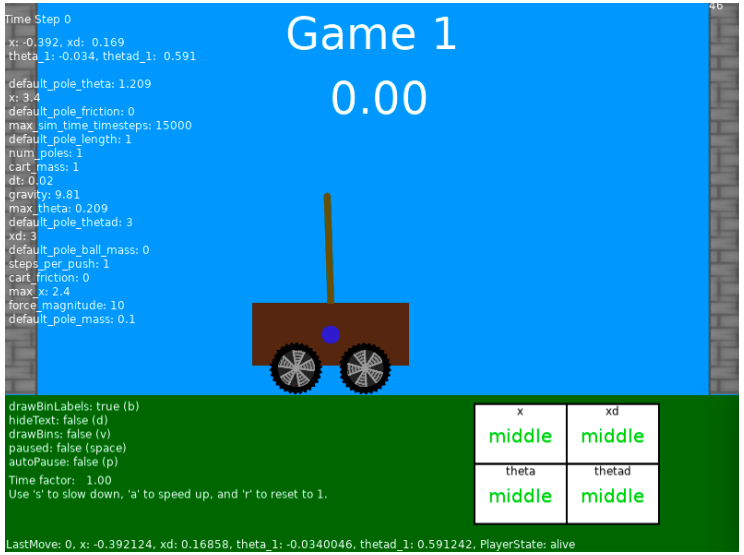

Biom **SOENS** **Caspian**

Collaborators' Hardware

Simulators: RISP, Brian, NEST, BindsNET, Nengo, NengoLoihi

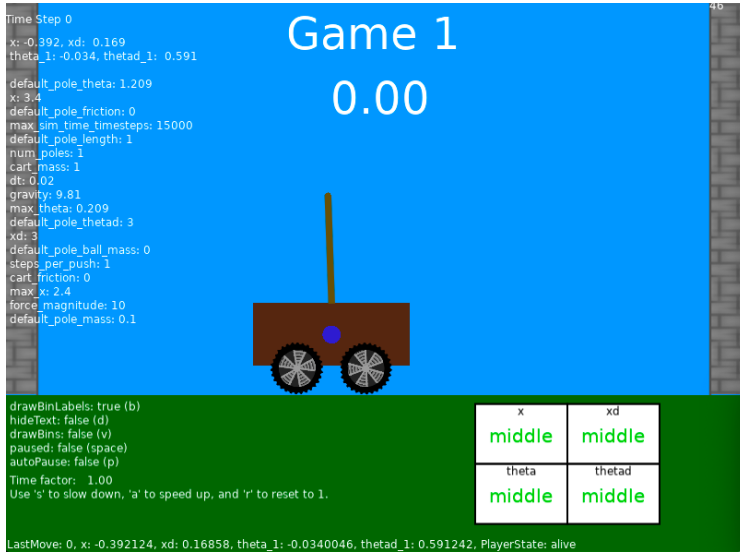
Plank, J. S., Schuman, C. D., Bruer, G., Dean, M. E., & Rose, G. S. (2018). The TENNLab exploratory neuromorphic computing framework. *IEEE Letters of the Computer Society*, *1*(2), 17-20.

Control



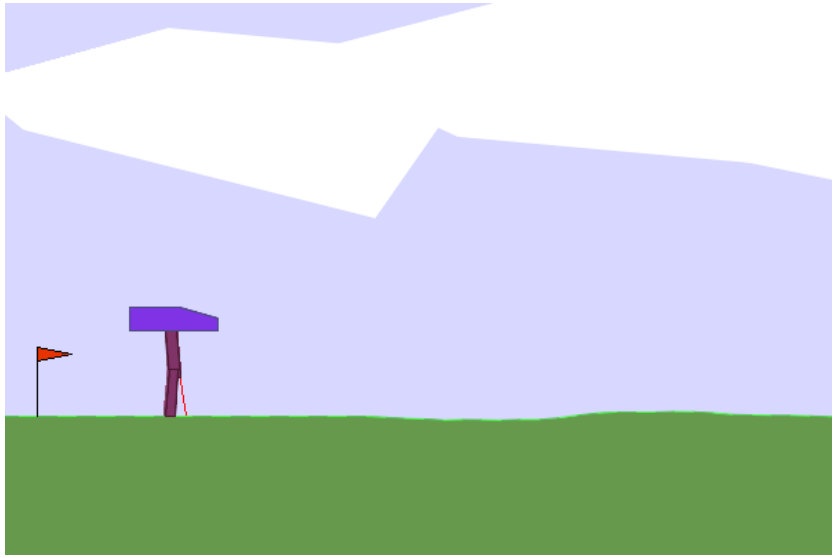
TENNLab Pole Balancing

- Observations: 4
- Actions: 2



TENNLab Pole Balancing
No Velocities

- Observations: 2
- Actions: 2



OpenAI Gym Bipedal Walker

- Observations: 24
- Actions: 4

Bipedal Walker source: https://gymnasium.farama.org/environments/box2d/bipedal_walker/

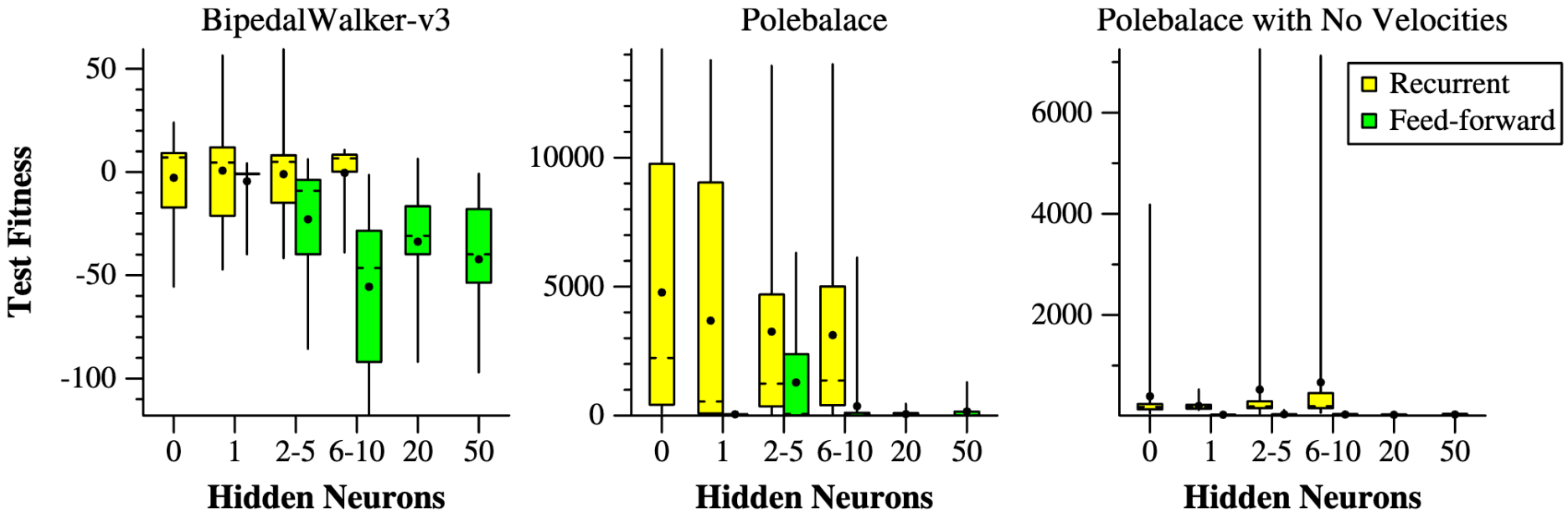
Experimental Setup

- For each algorithm (EONS and LEAP), each number of hidden neurons (0, 1, 2, 5, and 10 for EONS and 1, 5, 10, 20, and 50 for LEAP), and each application (pole balancing with and without velocities, and bipedal walker), we ran 20 tests with different random starting seeds.
- Each evolution:
  - Population size of 100
  - 300 generations
- The testing fitness is the average score across 1000 random episodes.

Results

Key Observation:

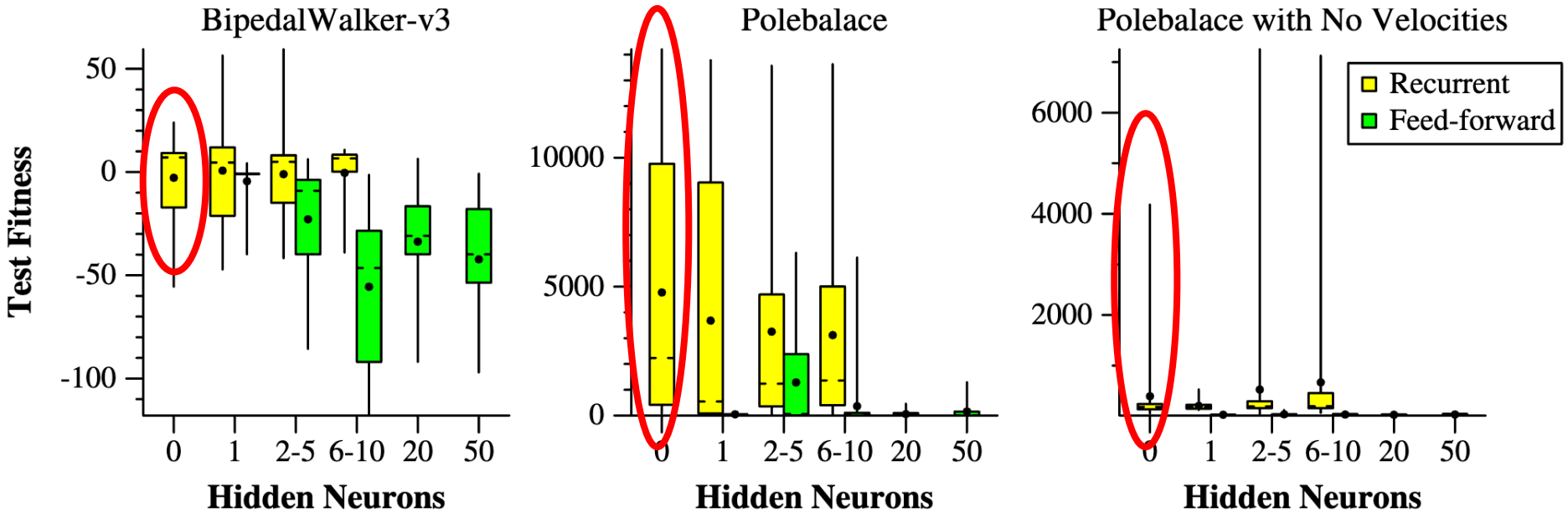
Recurrent networks achieve the best performance on each task



Results

Key Observation:

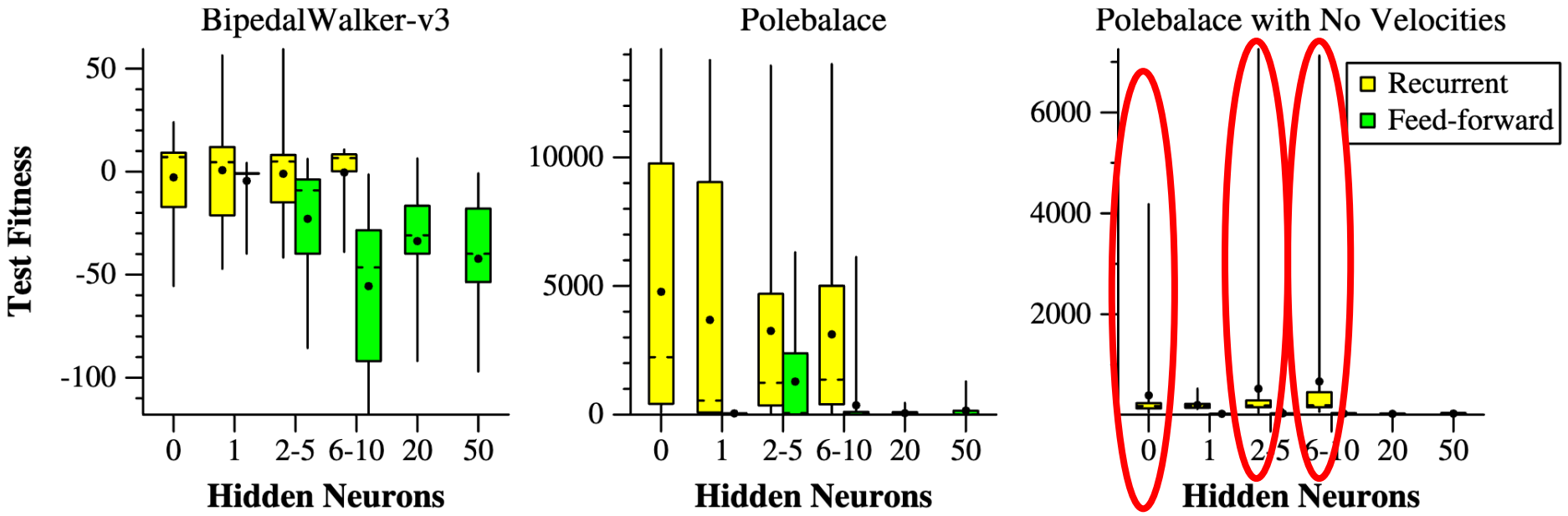
Recurrent networks with zero hidden neurons on average outperform all feed-forward networks



Results

Key Observation:

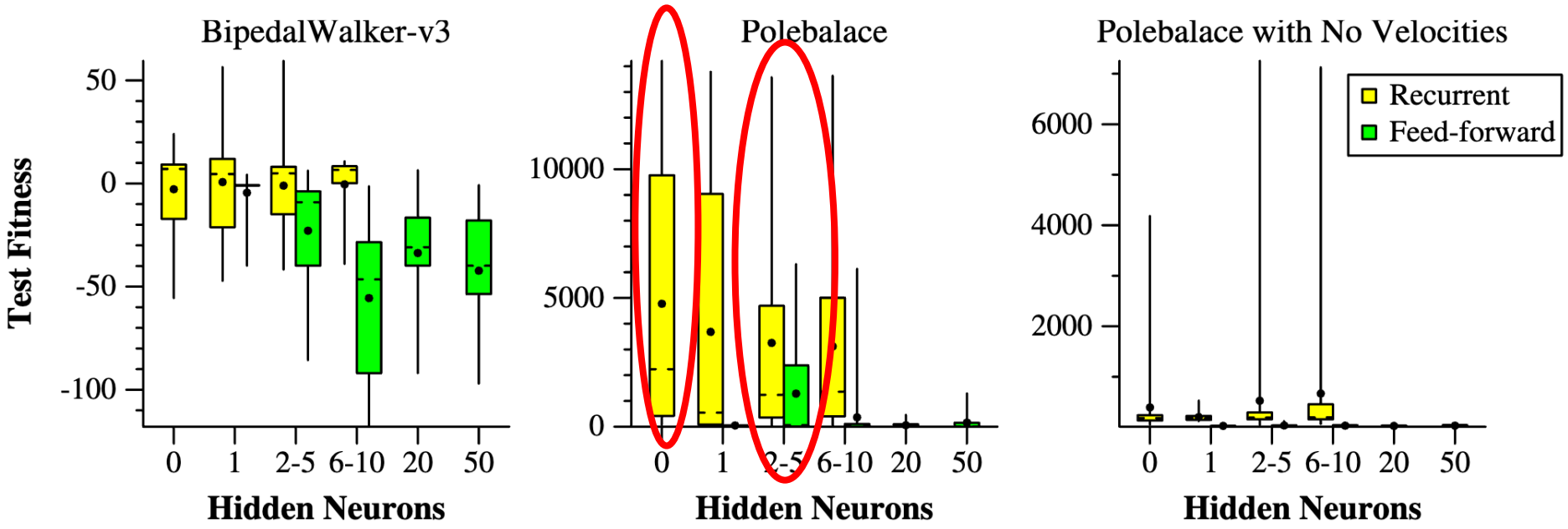
Pole balancing with no velocities requires recurrence (memory)



Results

Key Observation:

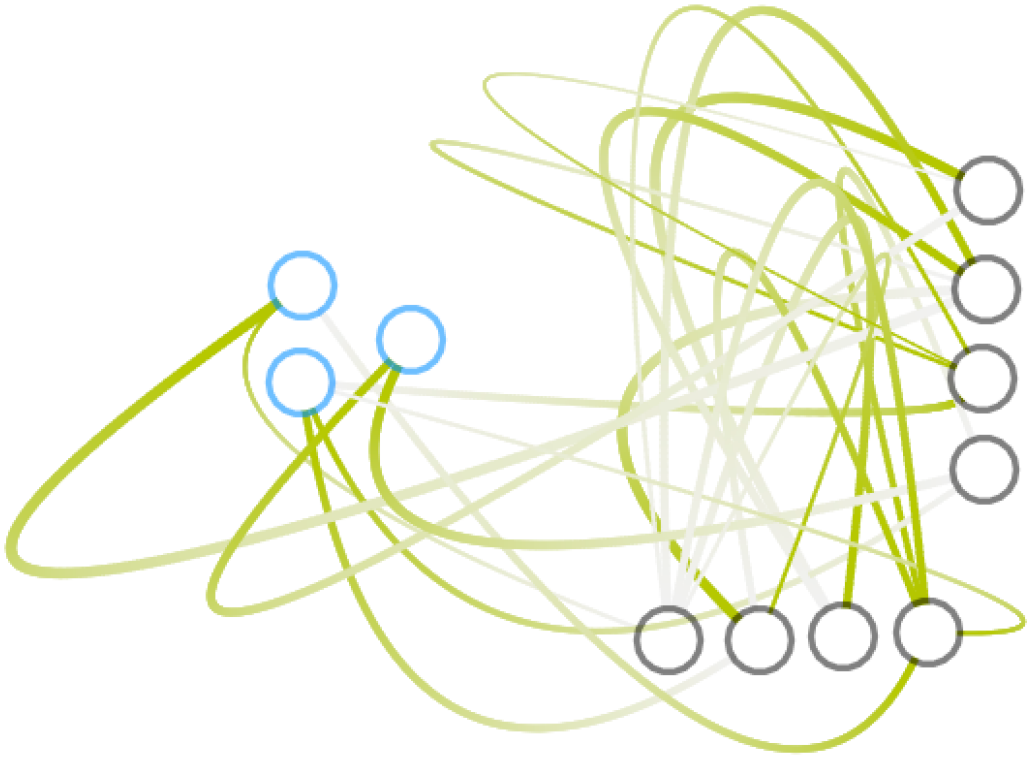
Even tasks that do not require recurrence perform better WITH recurrence



Network: Pole balancing with velocities

Key Observation:

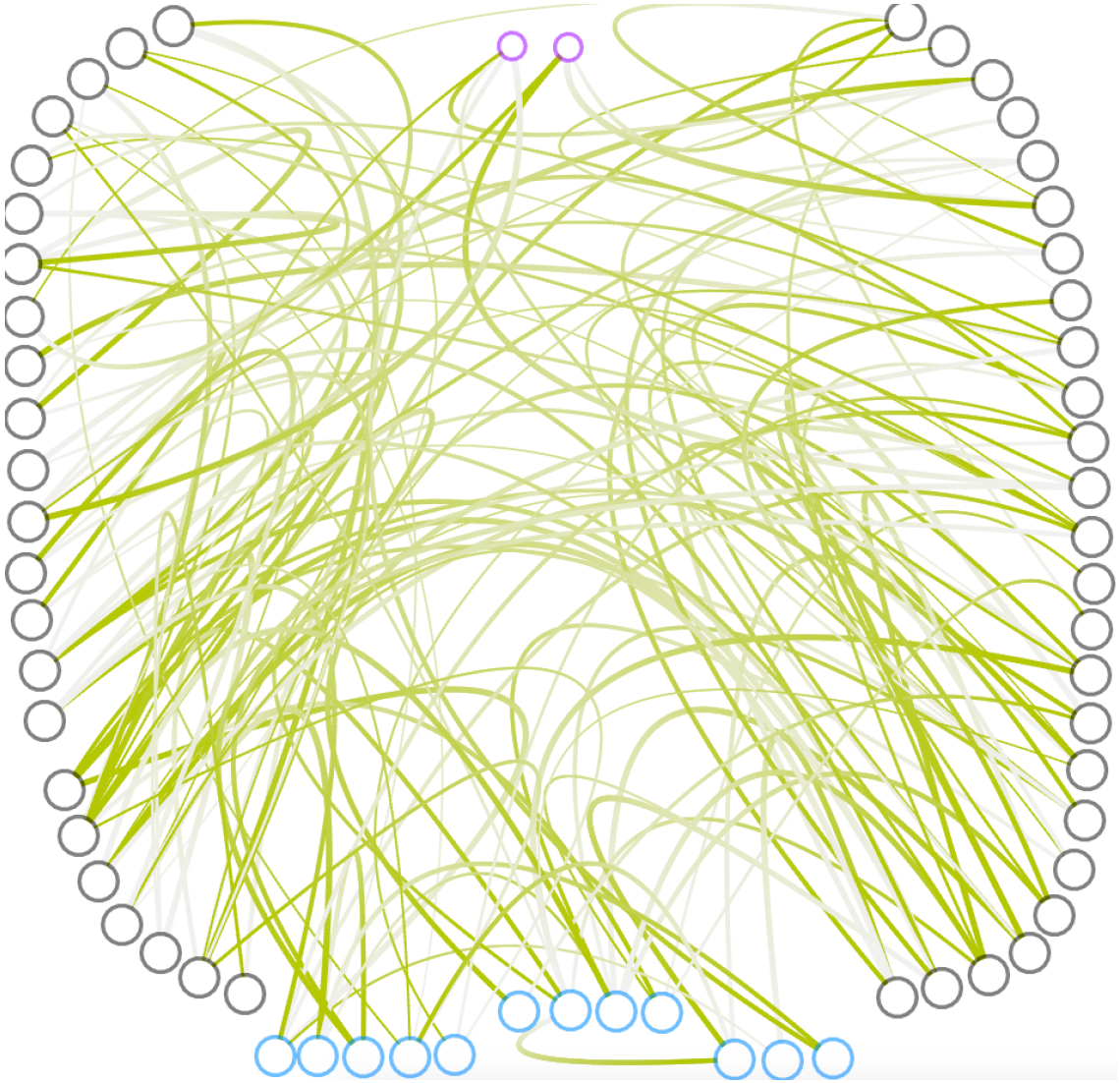
Best performing network for pole balancing with velocities has ***no hidden neurons at all***, but significant recurrent connectivity



Network: Bipedal-Walker

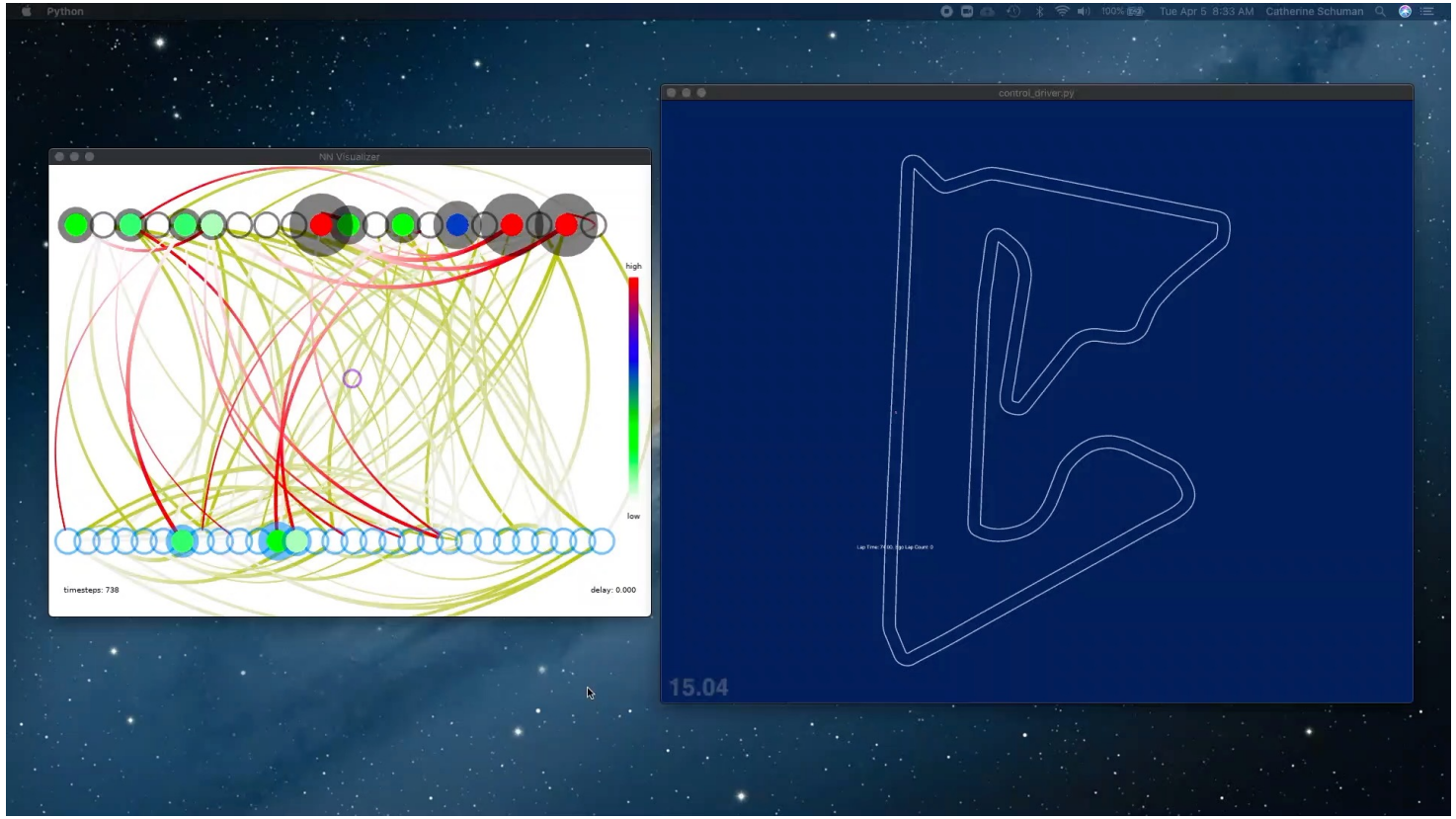
Key Observation:

Best performing network for bipedal walker has two hidden neurons and significant recurrent connectivity (input to input, output to output, output to input, etc.)



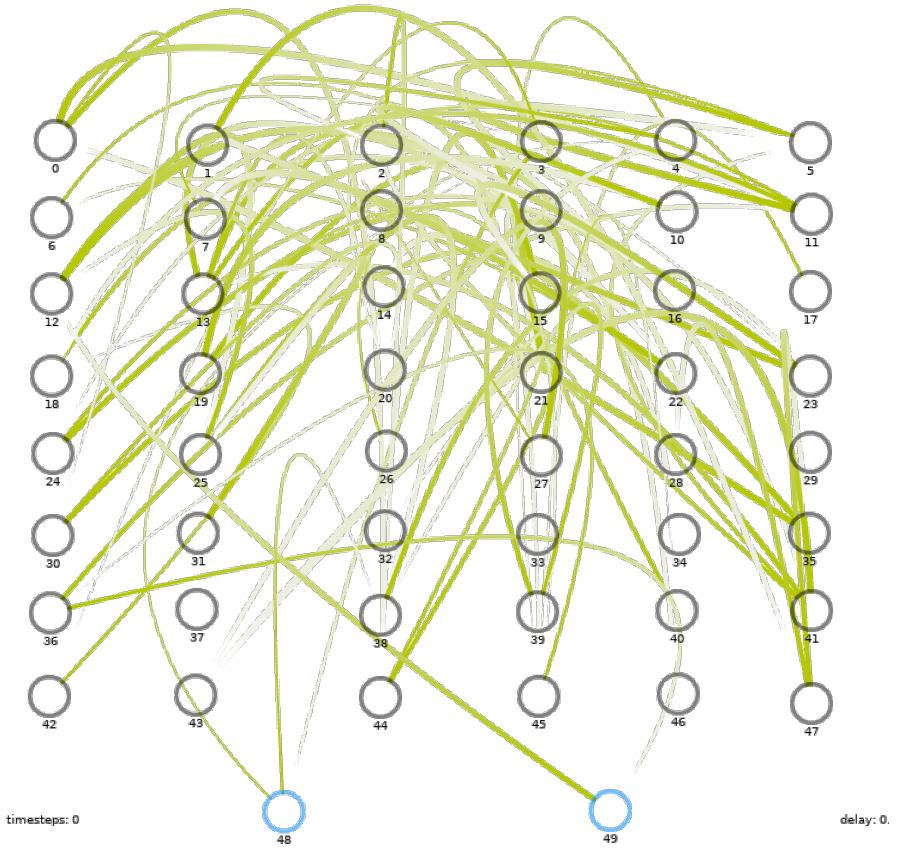
Some additional recurrence anecdotes…

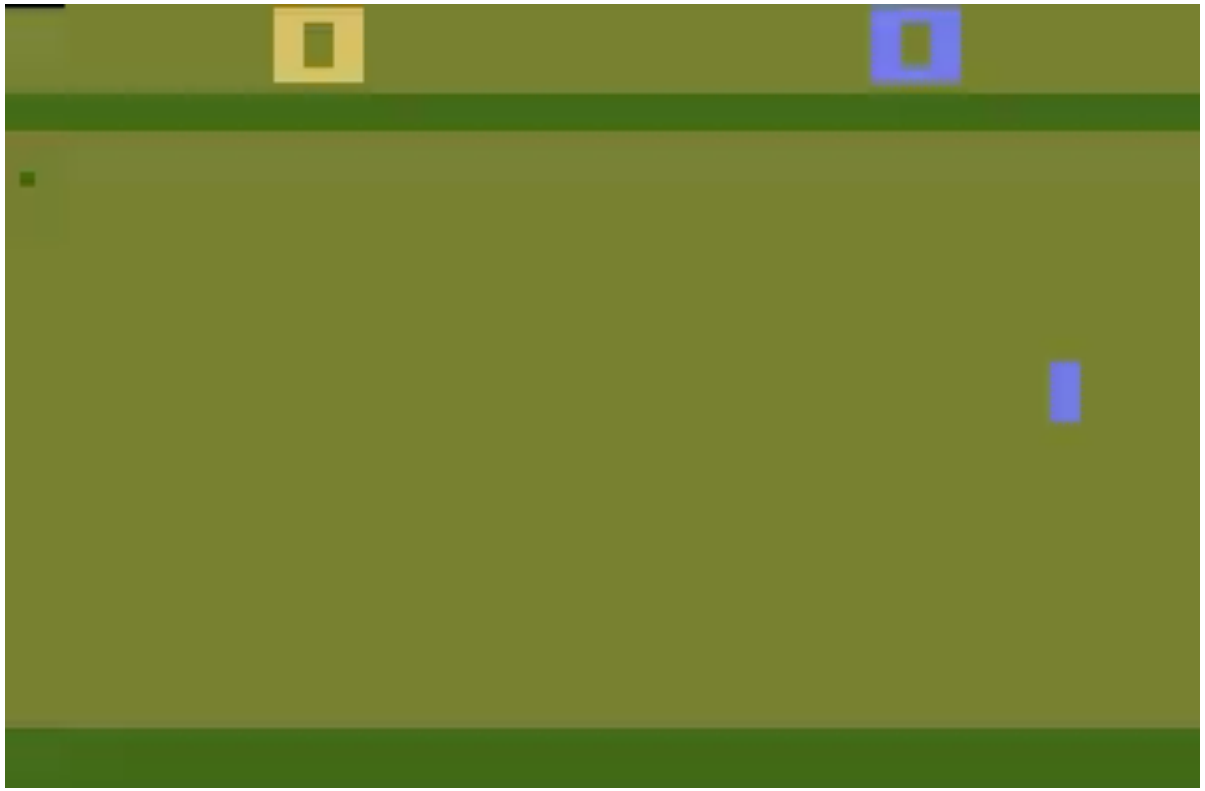
Recurrence Anecdotes: Autonomous Racing



Schuman, Catherine, Robert Patton, Shruti Kulkarni, Maryam Parsa, Christopher Stahl, N. Quentin Haas, J. Parker Mitchell et al. "Evolutionary vs imitation learning for neuromorphic control at the edge." Neuromorphic Computing and Engineering 2, no. 1 (2022): 014002

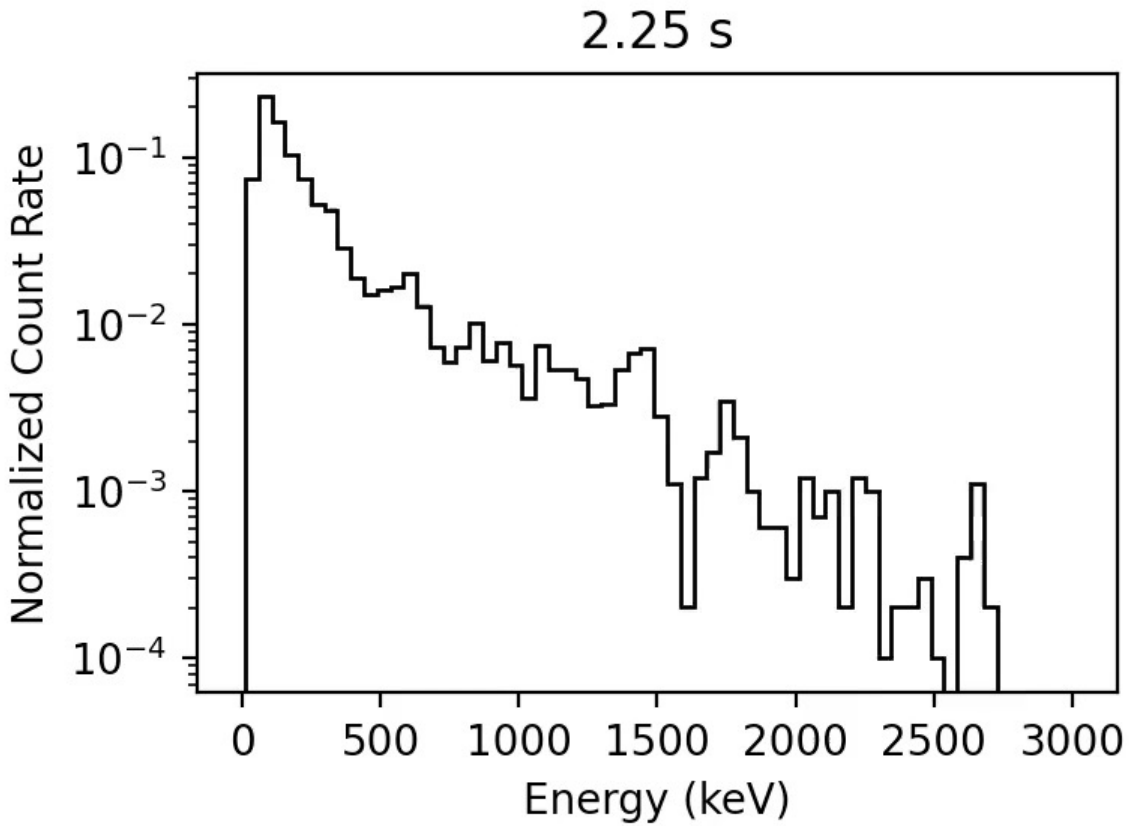
Recurrence Anecdotes: Pong





Rizzo, Charles P., Catherine D. Schuman, and James S. Plank. "Neuromorphic downsampling of event-based camera output." In Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference, pp. 26-34. 2023.

Recurrence Anecdotes: Radiation Detection



Ghawaly, James, Aaron Young, Andrew Nicholson, Brett Witherspoon, Nick Prins, Mathew Swinney, Cihangir Celik, Catherine Schuman, and Karan Patel. "Performance Optimization Study of the Neuromorphic Radiation Anomaly Detector." In Proceedings of the 2023 International Conference on Neuromorphic Systems, pp. 1-7. 2023.

Conclusions

- For these three control applications, the recurrent networks can result in not only ***better performance***, but better performance with ***significantly smaller networks***
- Our main goal with this work was to demonstrate that allowing for more free-form recurrent networks can give better performance with potentially smaller networks
- Our secondary goal was to encourage the community to explore the development of:
  - Algorithms that can produce and leverage these sorts of network architectures
  - ***Hardware that can implement them effectively***

Acknowledgements



This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under award number DE-SC0022566, and in part on research sponsored by Air Force Research Laboratory under agreement number FA8750-21-1-1018.

ICONS 2024

- Arlington, VA
- July 30-August 2, 2024
- Submission Deadline:
  - April 30, 2024
- https://iconsneuromorphic.cc/

ICONS
International Conference on Neuromorphic Computing Systems

Thank you!

Questions?

Contact:
Email: cschuman@utk.edu

Website: neuromorphic.eecs.utk.edu

TENN LAB
NEUROMORPHIC
ARCHITECTURES. LEARNING. APPLICATIONS.