WESTERN SYDNEY UNIVERSITY

OPTIMISED EVENT-DRIVEN SNN ARCHITECTURES

tinyML Asia 2023

Yeshwanth Ravi Theja Bethi
International Centre for Neuromorphic Systems (ICNS)

Collaborators

Dr. Saeed Afshar
ICNS

Prof. André van Schaik
ICNS

Dr. Ali Mehrabi
ICNS

The problem of learning at edge

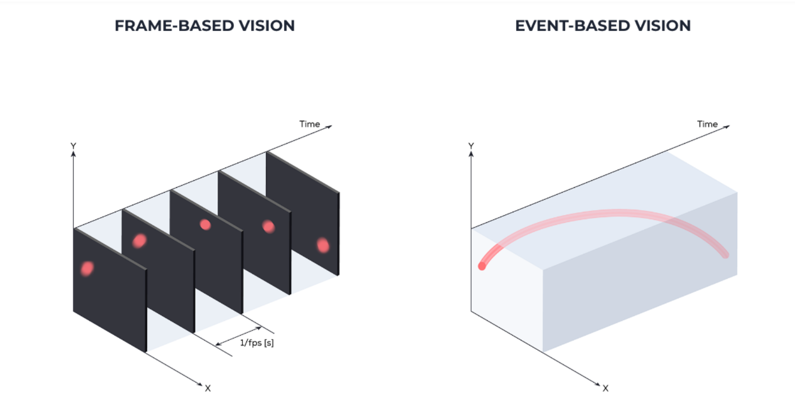
- Error-Backpropagation is computationally expensive -> High Power
- Requires a symmetric backward pathway to pass full precision values
- Backpropagation requires precise and synchronous orchestration of gradient calculation and weight updates. It leads to a ‘locking effect’ when implemented in hardware
- The von-Neumann bottleneck. We need co-located memory and compute in hardware
- Possible for small networks, but not scalable
- The result: Most solutions are only inference at edge rather than learning at edge

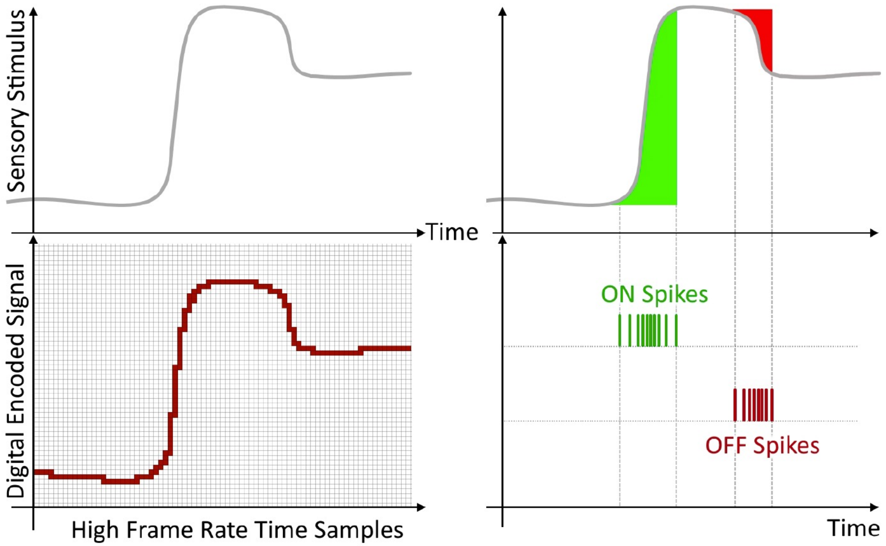
Lessons from nature

- Communication sparsity
- Computation sparsity
- Massive parallelism
- Local Independence

Communication Sparsity

- Information coding in spikes
- Spike only when necessary
- Neuromorphic sensors have proved to be energy efficient solutions for sampling data with high temporal precision yet generating sparse data rates
- Timing is key

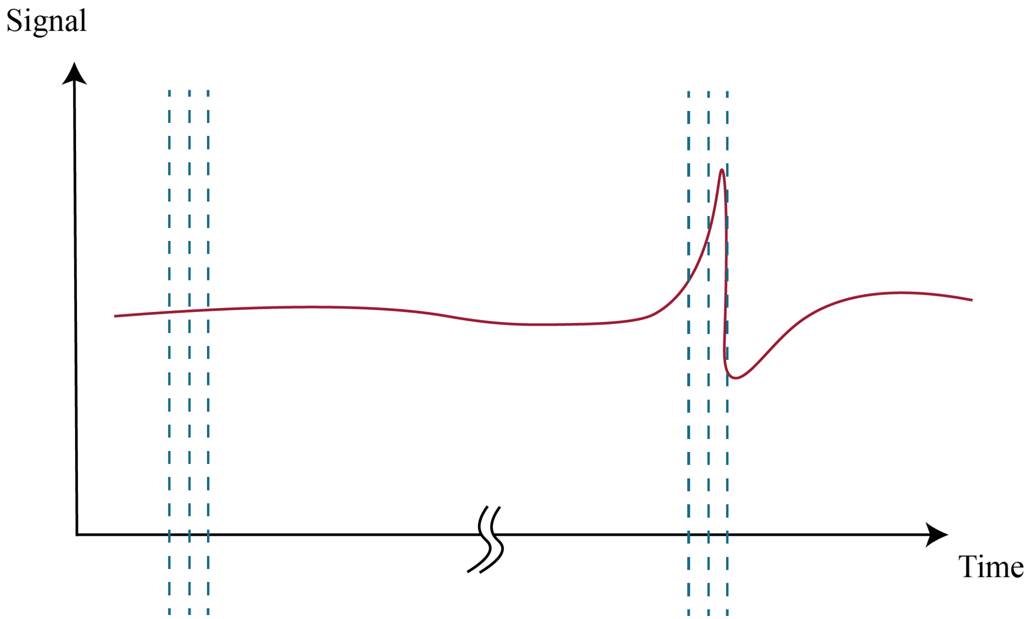




WESTERN SYDNEY UNIVERSITY

Computation Sparsity

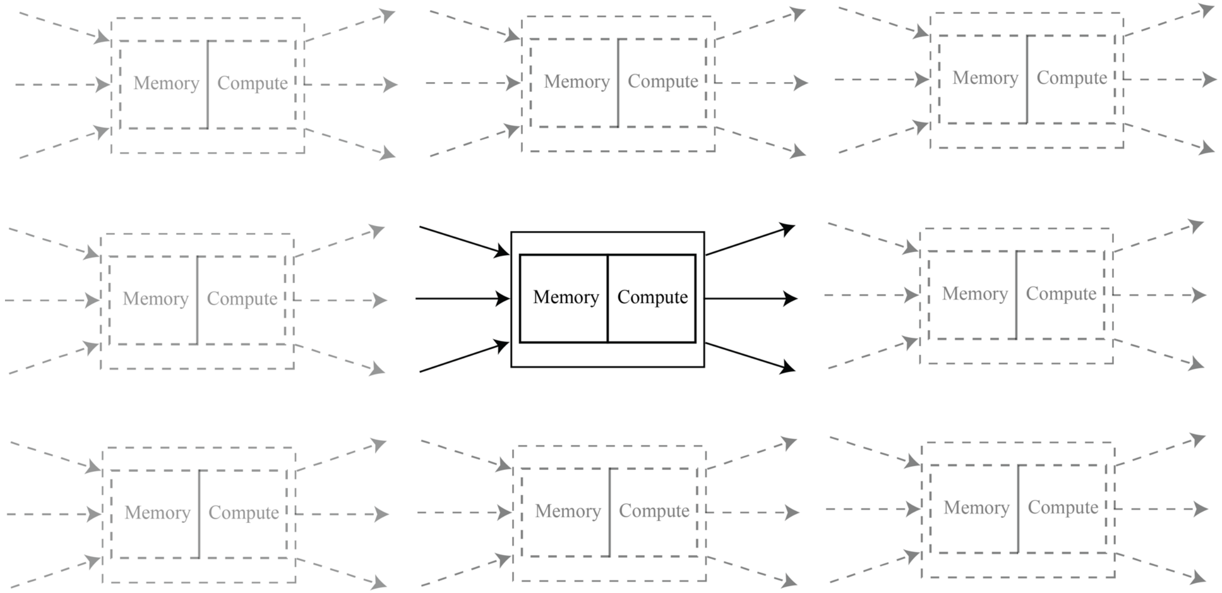
- Asynchronous computing at the arrival of spikes
- Long periods of boredom with sudden bursts of activity!
- Sensors like event-based cameras solve it at the sensing end. Need similar capabilities at the processing end of things.
- Compute only when necessary



WESTERN SYDNEY UNIVERSITY

Local Independence and Parallelism

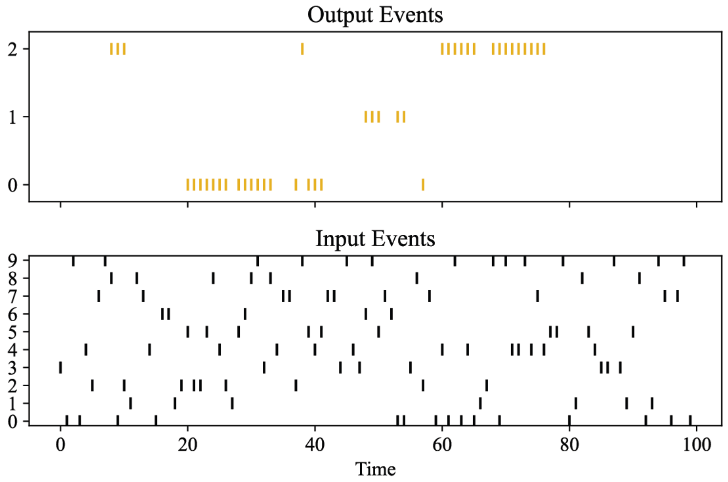
- Local learning vs Global learning
- Scalable network architectures that can be decomposed into entities that can operate independently
- The smaller the better
- The similar the better
- Fault tolerant and extensible



WESTERN SYDNEY UNIVERSITY

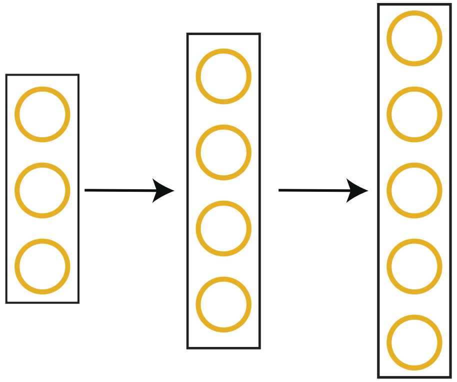
Event-driven Neural Architectures

Event-driven



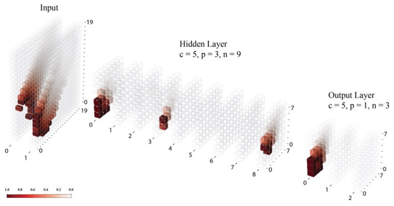
Asynchronous

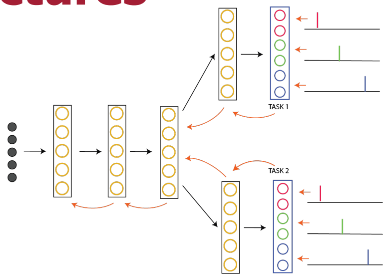
Neural

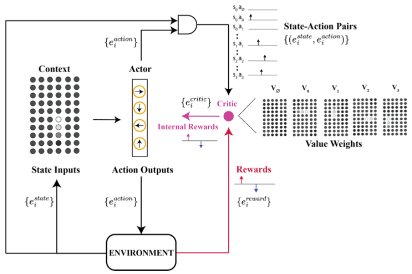


Distributed and Scalable

Architectures

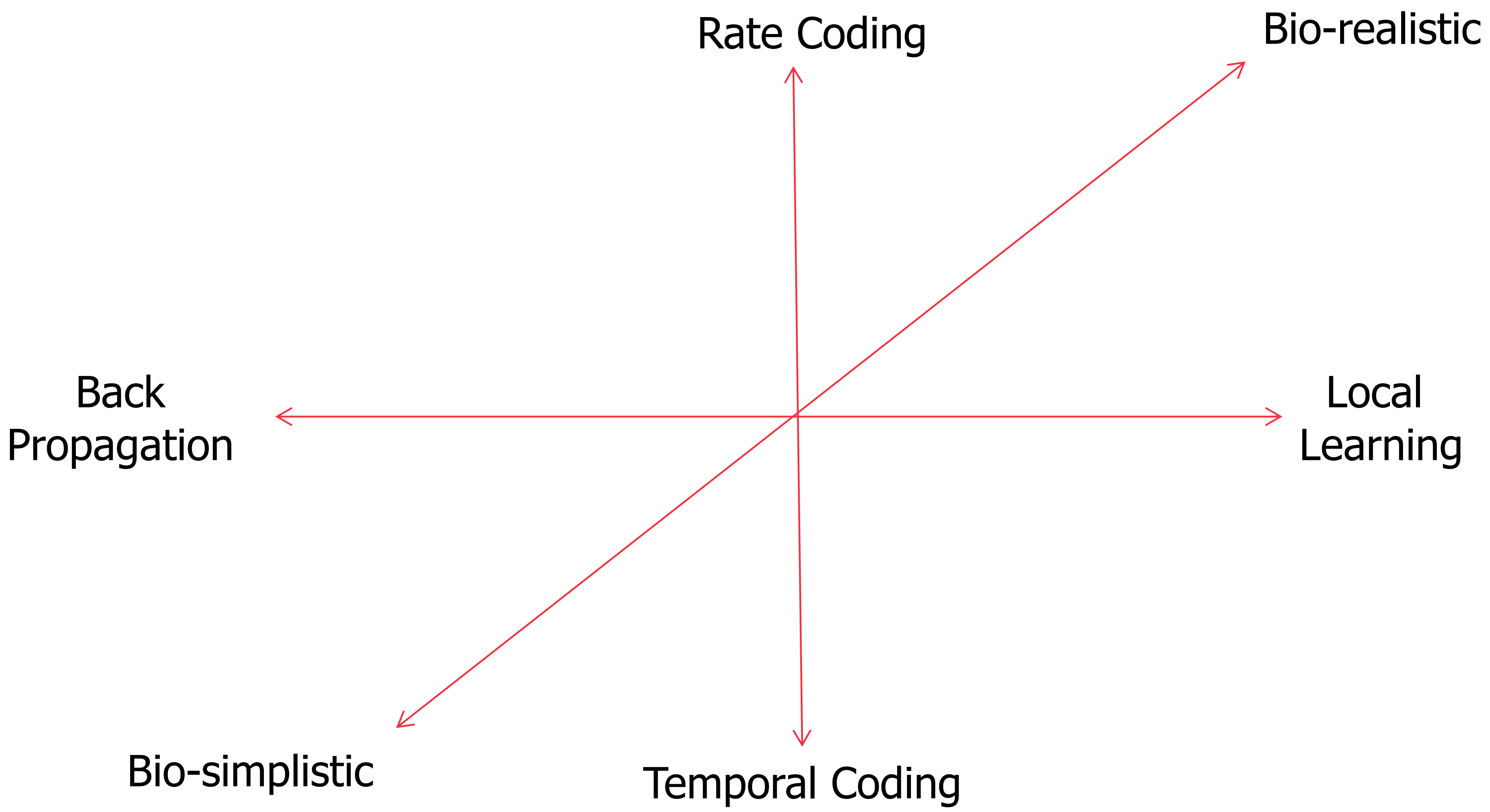






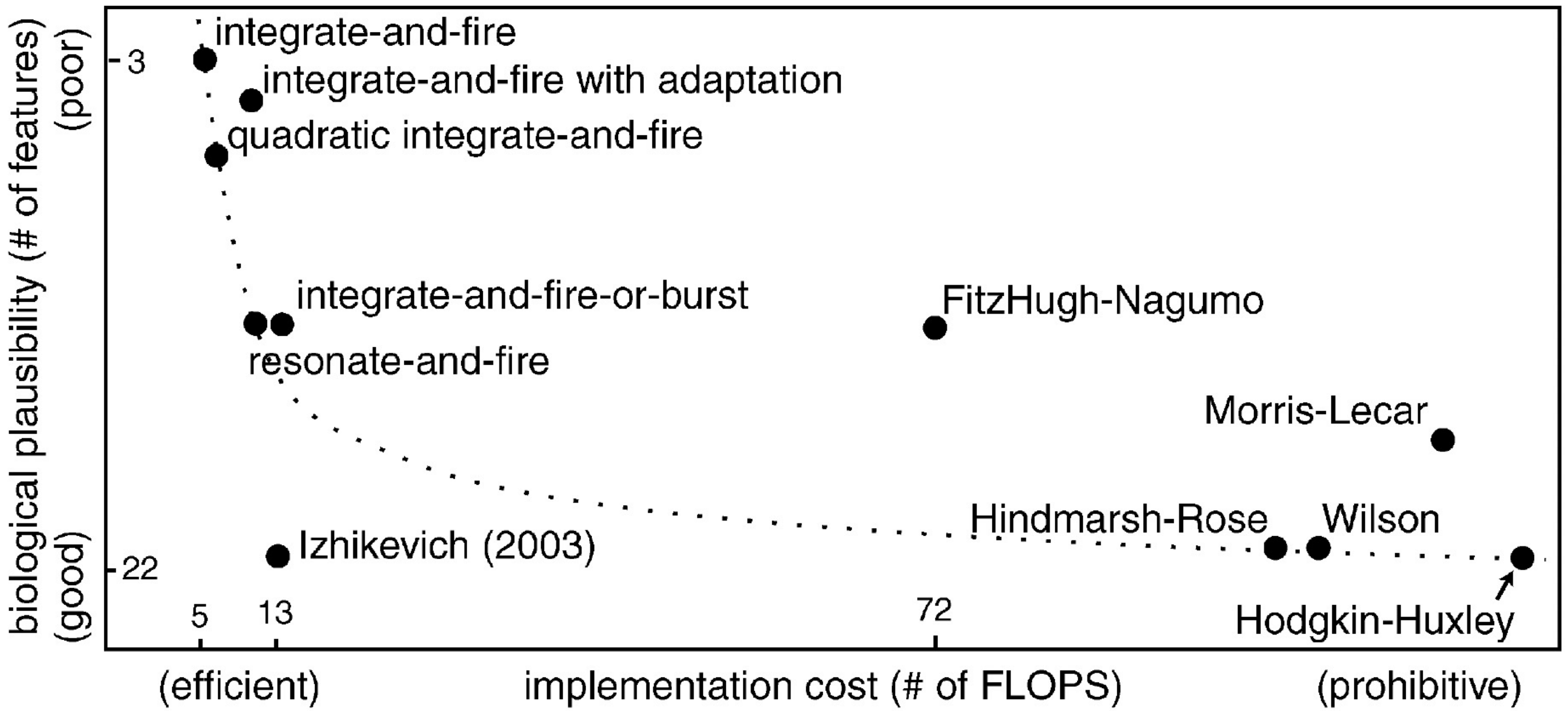
Compositional and Inductive Priors

The SNN Zoo


Rate Coding
Bio-realistic
Back Propagation
Local Learning
Bio-simplistic
Temporal Coding

WESTERN SYDNEY UNIVERSITY

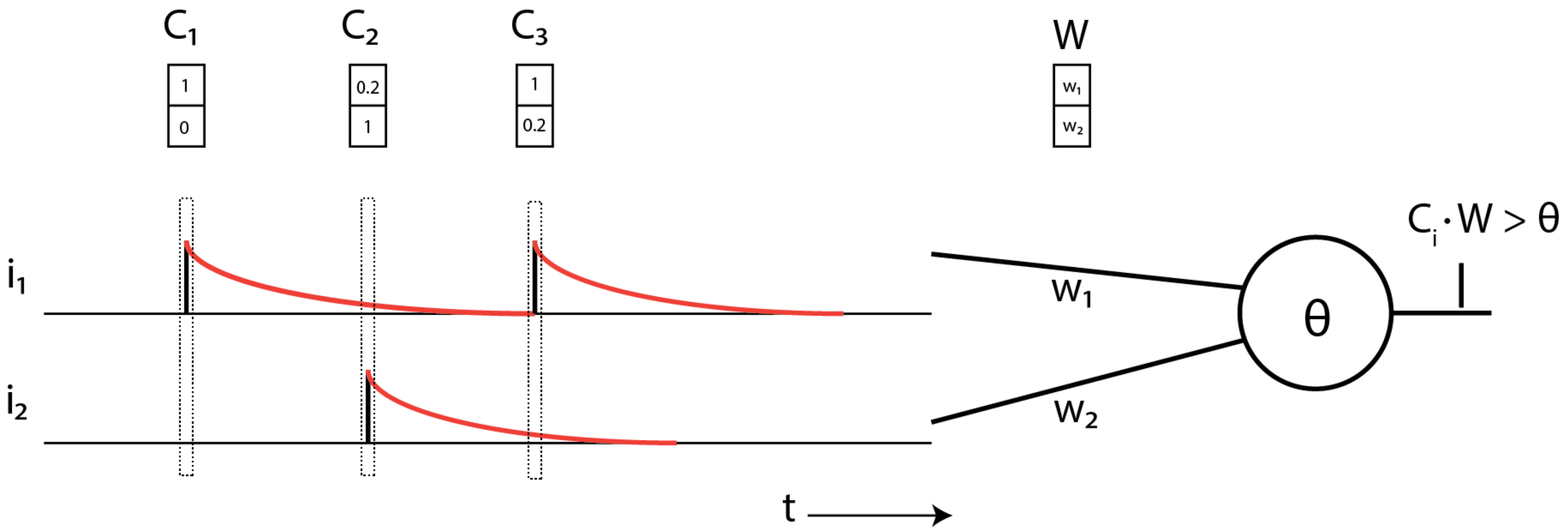
Abstraction of Spiking Neural Networks



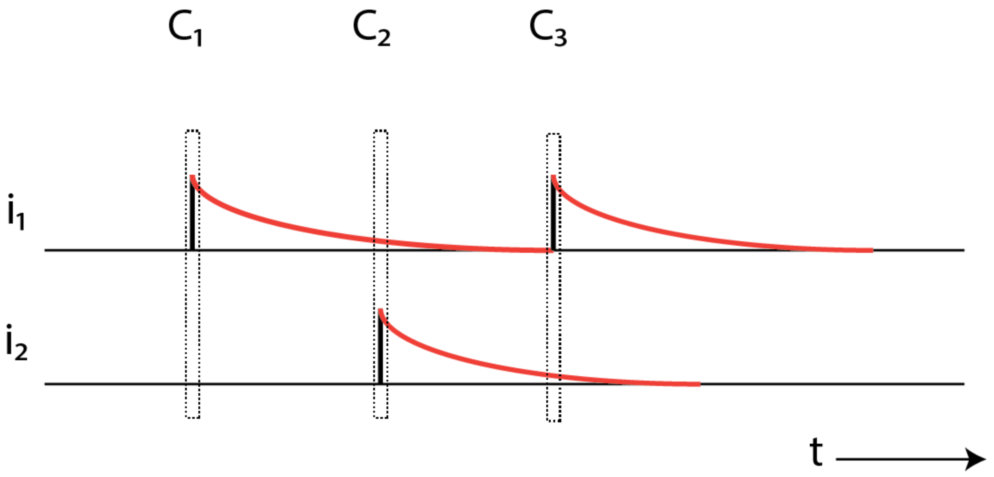
Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? IEEE transactions on neural networks, 15(5).

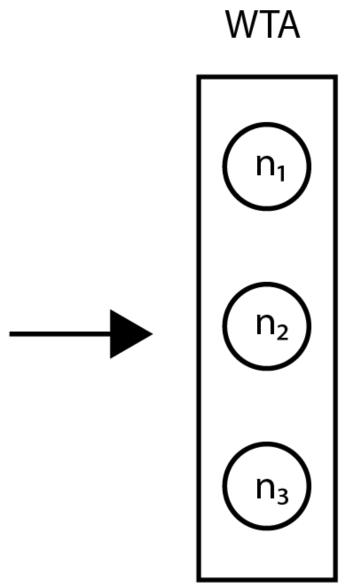
WESTERN SYDNEY UNIVERSITY

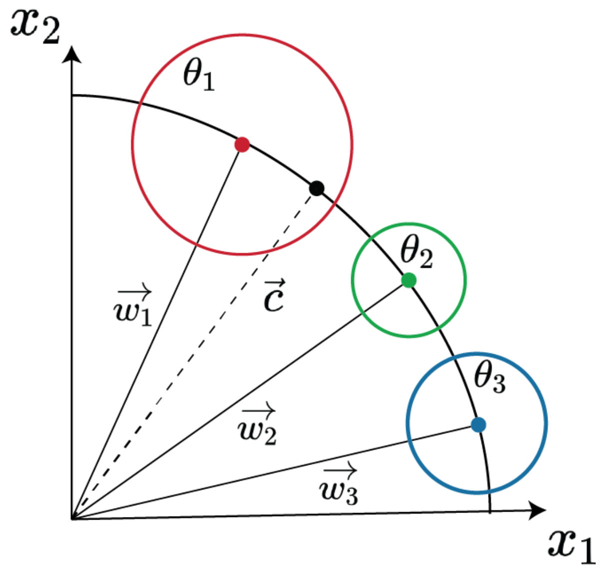
Timesurfaces and dot products for abstraction of spiking neurons

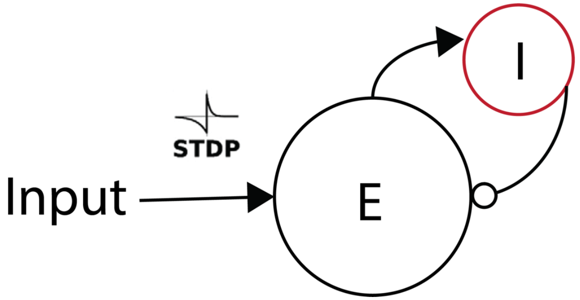


Winner-Takes-All and Selection Thresholds



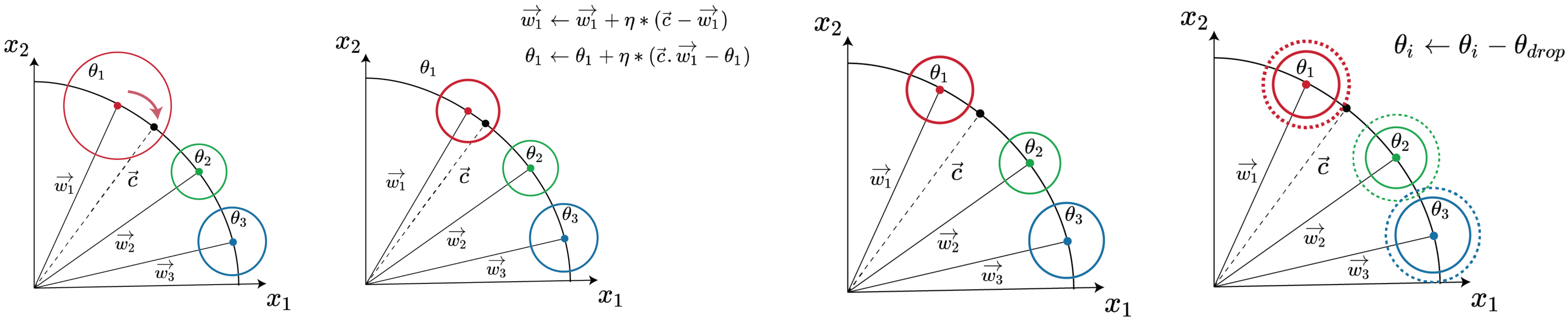






WESTERN SYDNEY UNIVERSITY

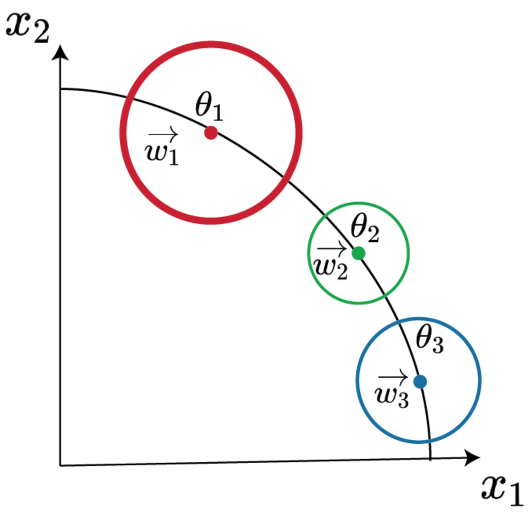
Unsupervised feature extraction using selection thresholds

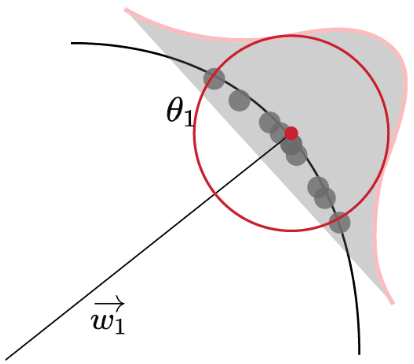


Afshar, Saeed, et al. (2020) "Event-based feature extraction using adaptive selection thresholds." Sensors

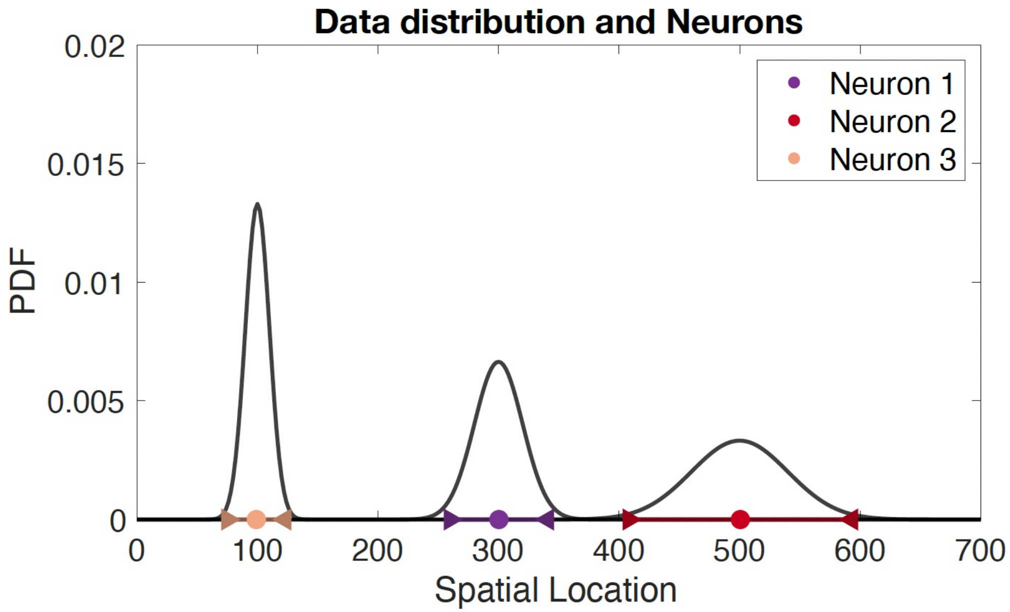
WESTERN SYDNEY UNIVERSITY

Role of weights and thresholds





$$\vec{w_1} \leftarrow \vec{w_1} + \eta * (\vec{c} - \vec{w_1})$$
$$\theta_1 \leftarrow \theta_1 + \eta * (\vec{c}.\vec{w_1} - \theta_1)$$

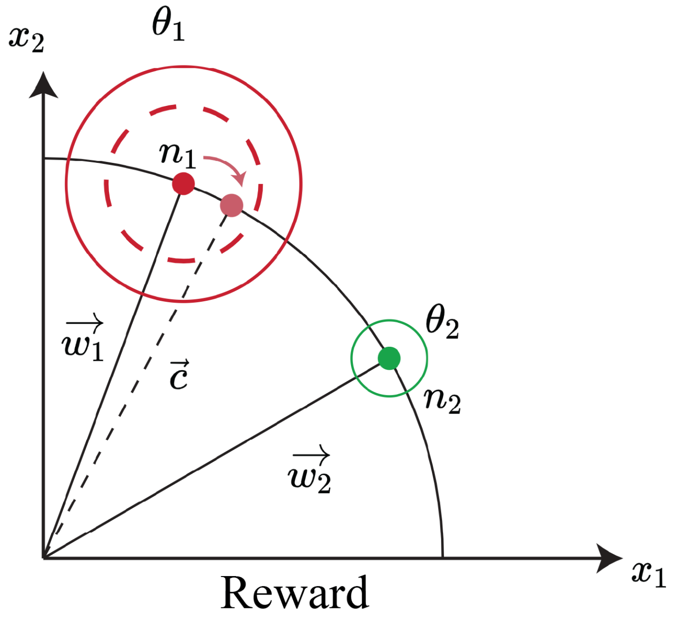


WESTERN SYDNEY UNIVERSITY

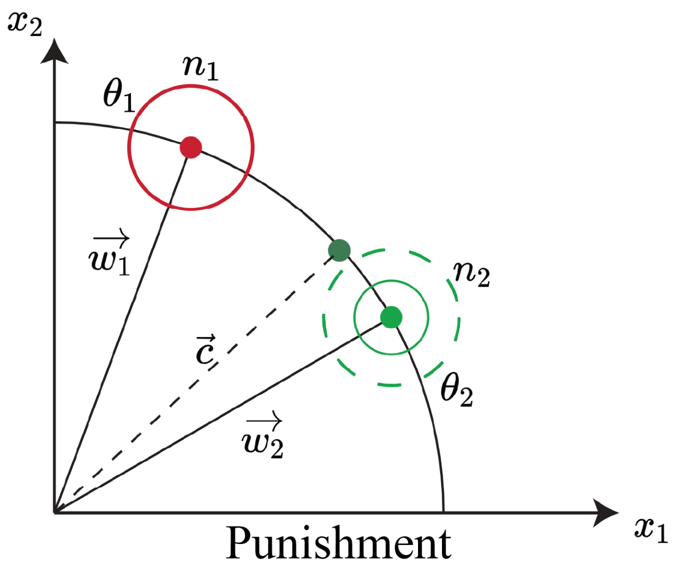
Supervised learning

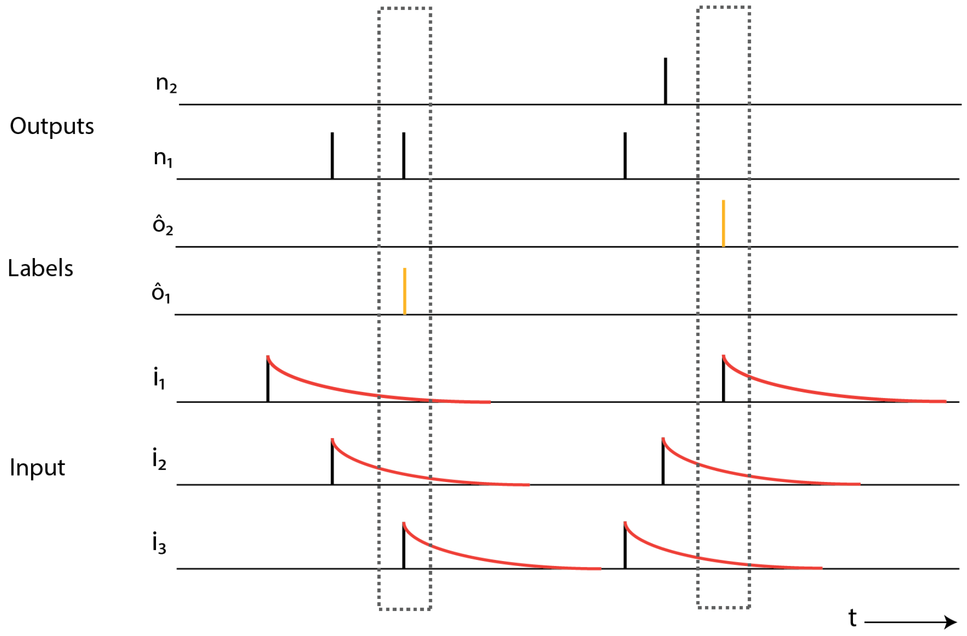
$$\overrightarrow{w_1} \leftarrow \overrightarrow{w_1} + \eta * (\vec{c} - \overrightarrow{w_1})$$

$$\theta_1 \leftarrow \theta_1 + \eta * (\vec{c}.\overrightarrow{w_1} - \theta_1)$$



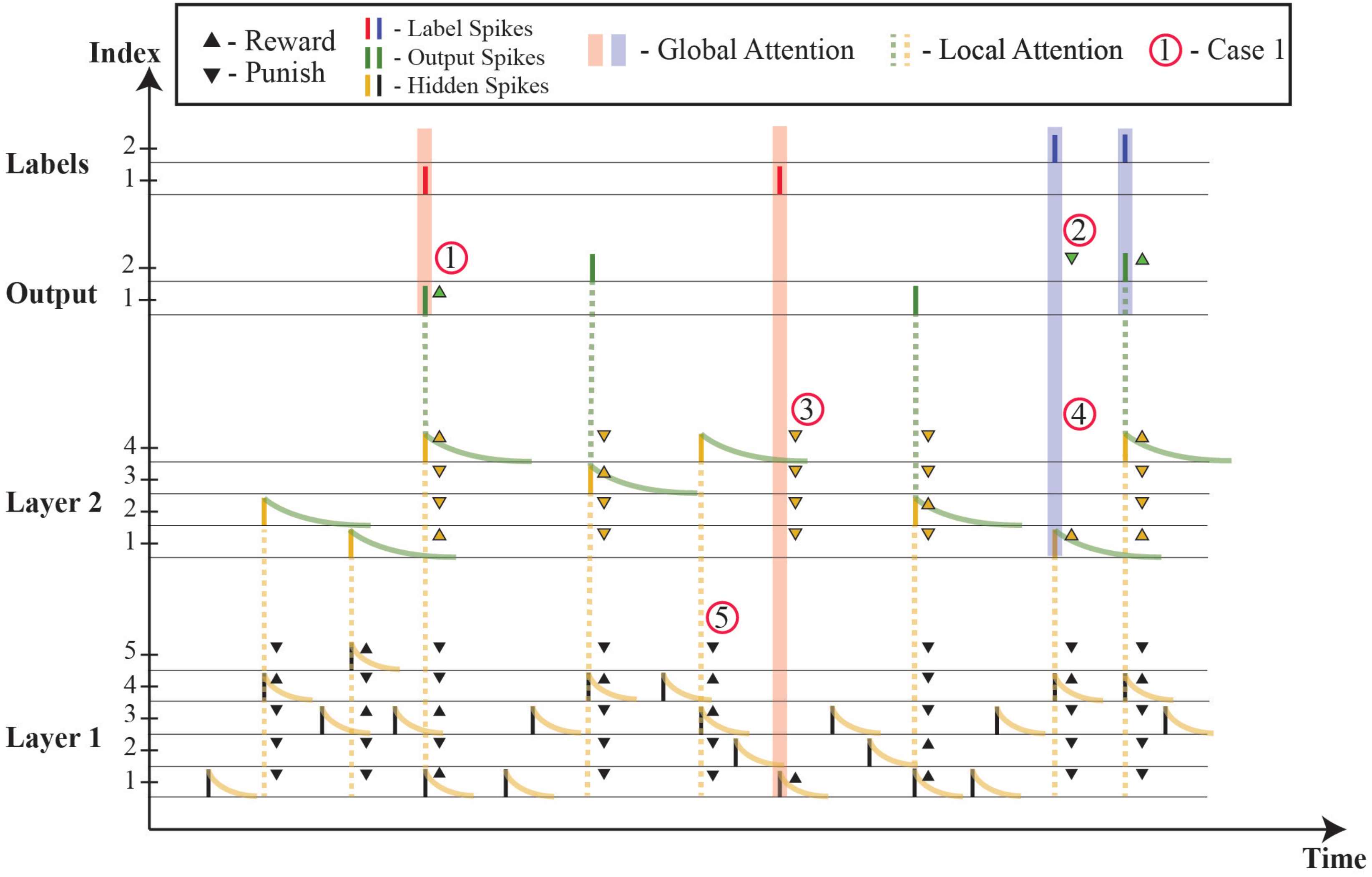
$$\theta_2 \leftarrow \theta_2 - \theta_{drop}$$





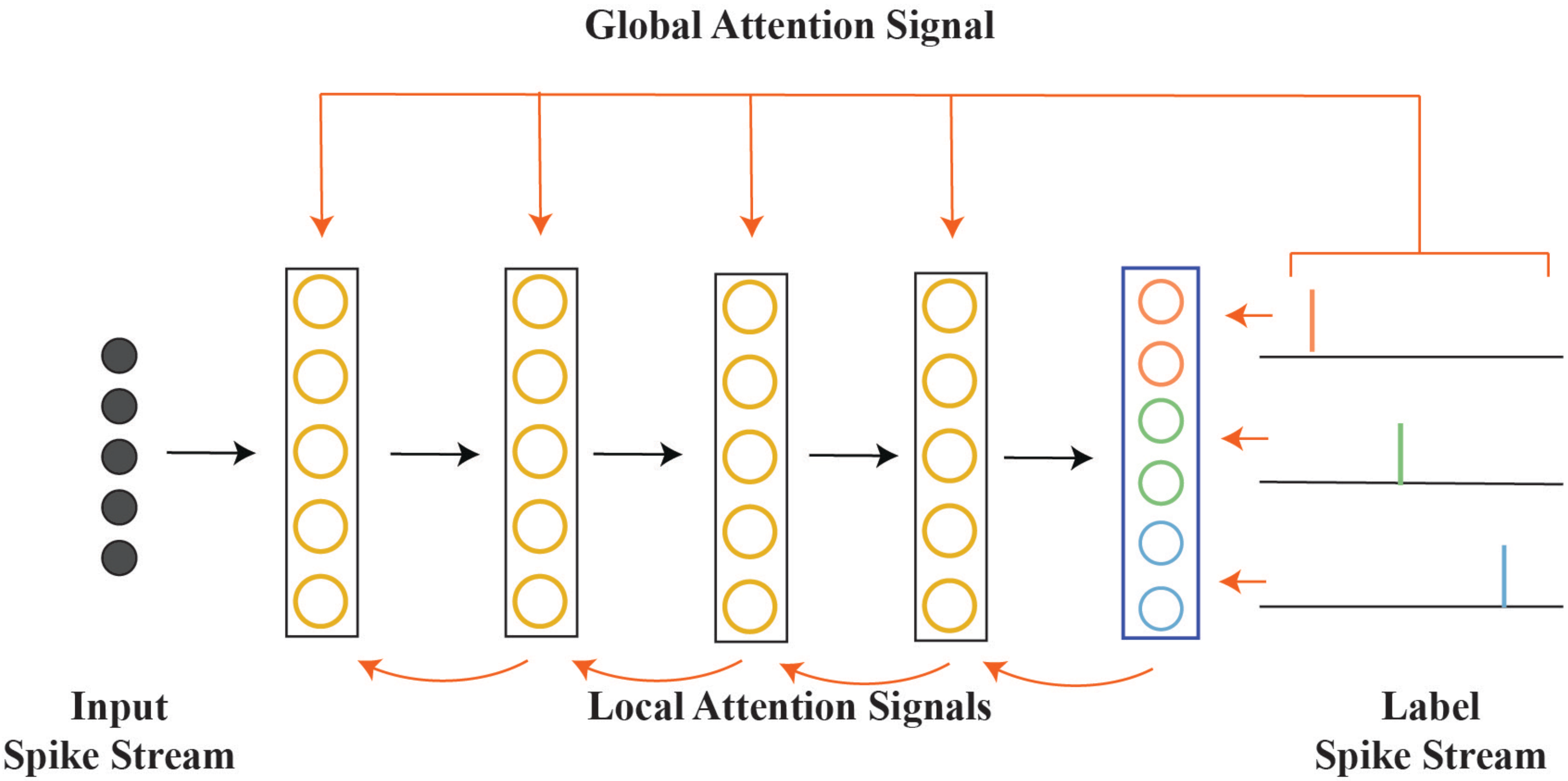
WESTERN SYDNEY UNIVERSITY

Spike-Timing-Dependent Threshold Adaptation



WESTERN SYDNEY UNIVERSITY

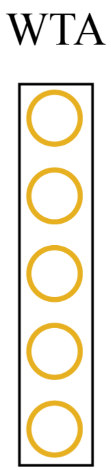
Optimised Deep Event-driven SNN Architecture (ODESA)



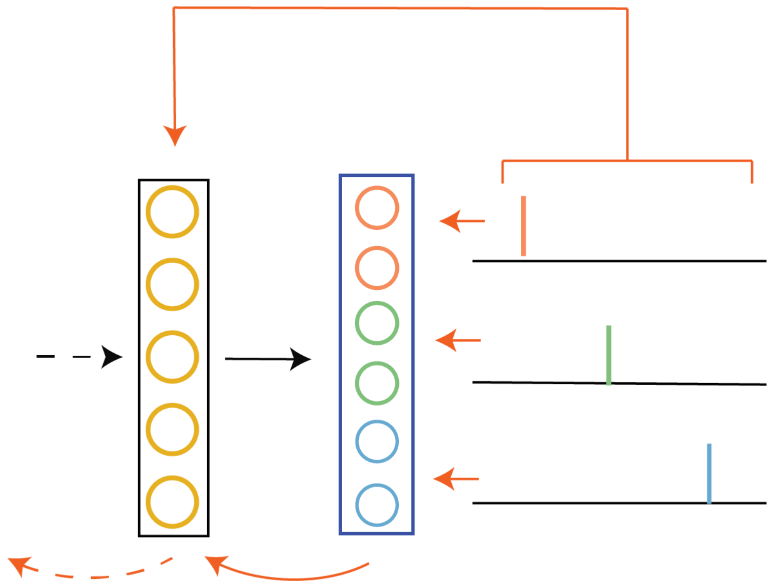
Bethi, Y., Xu, Y., Cohen, G., Van Schaik, A. and Afshar, S., 2022. An optimized deep spiking neural network architecture without gradients. IEEE Access, 10

WESTERN SYDNEY UNIVERSITY

Local Competition and Global Cooperation



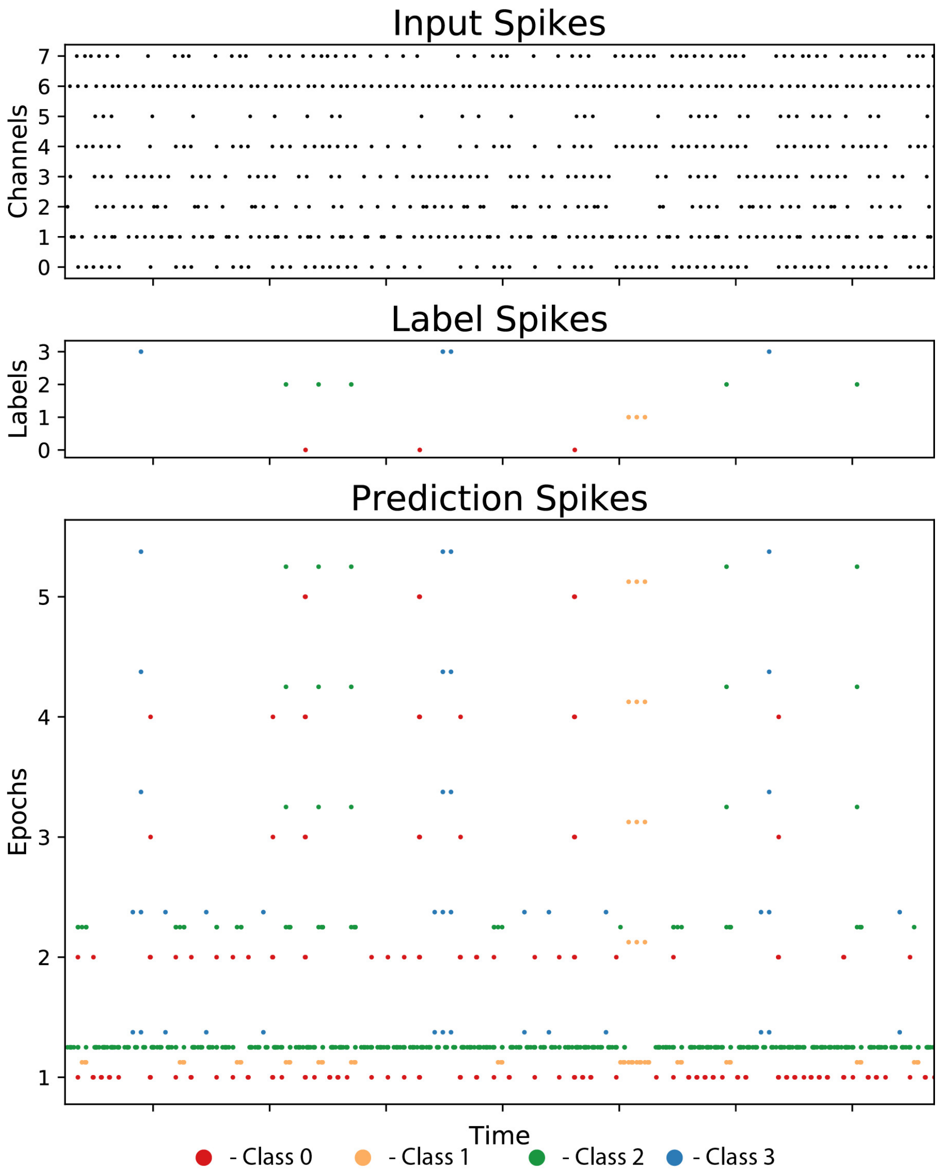
Local competition via Winner-Takes-All



Global cooperation through binary attention signals

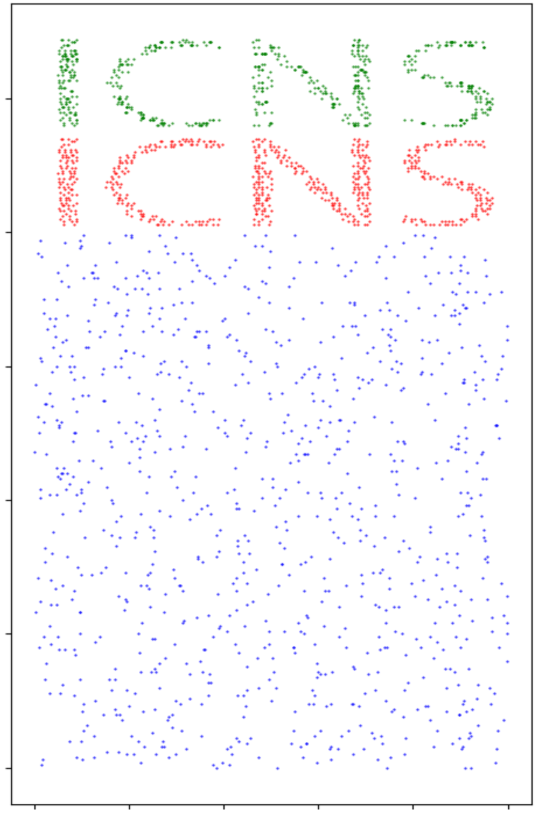
WESTERN SYDNEY UNIVERSITY

ODESA results



WESTERN SYDNEY UNIVERSITY

ODESA Results



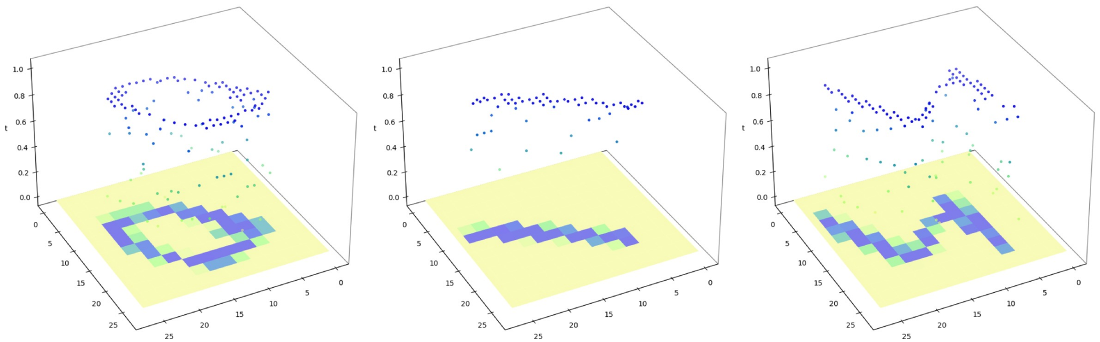
Predicted Spikes

Target Spikes

Input Spikes

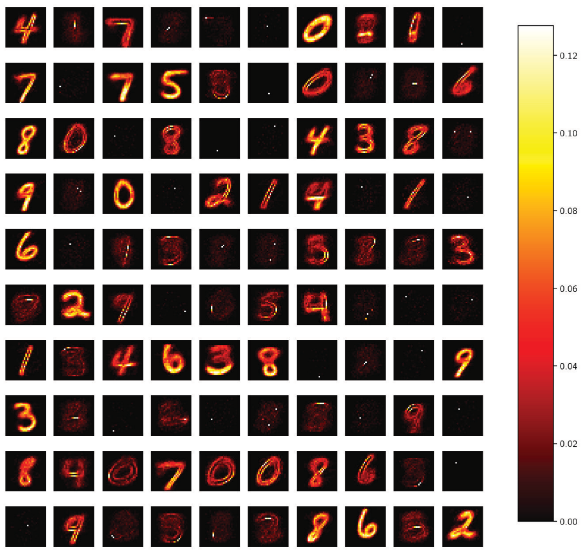
Oxford Spike Pattern
0.67 IOU per input spike

ODESA Results

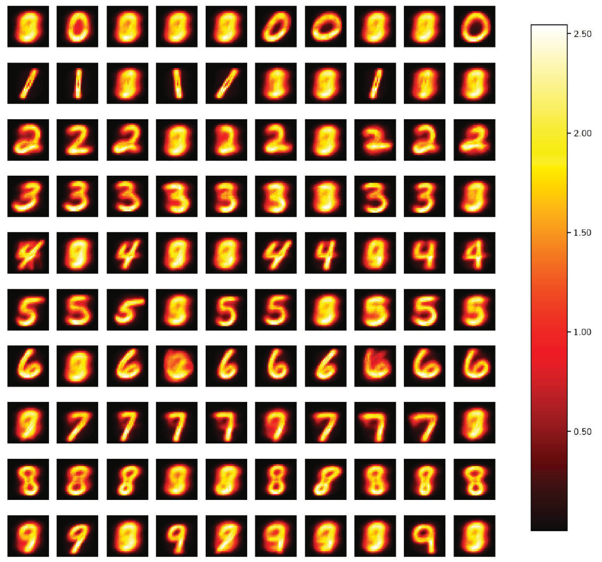


Method	Pre-Processing	Test Accuracy
Two Layer with Exponential STDP + *a posteriori* neuron selection [63]	Rate Coding	95.00%
Spiking RBM + Contrastive Divergence, Linear Classifier [64]	Thresholding + Rate coding	89.00%
Spiking RBM + Contrastive Divergence [65]	Thresholding + Spike Coding	91.90%
Dendritic neurons + Morphology Learning [66]	Rate Coding	90.30%
Multi-layer hierarchical network + STDP with Calcium variable [67]	Orientation Detection + Spike Coding	91.60%
Spiking CNN + Tempotron Rule [68]	Scaling, Orientation, Thresholding + Spike Coding	91.30%
Three Layer SNN + STDP-Backpropagation [69]	Rate Coding	97.20%
Deep SNN + Backpropagation [70]	Rate Coding	98.60 %
Two Layer SNN + SGD [71]	Latency based Spike Encoding	96.92%
ODESA (Our method)	Latency based Spike Encoding	93.24%

100 Random neurons out of W_H

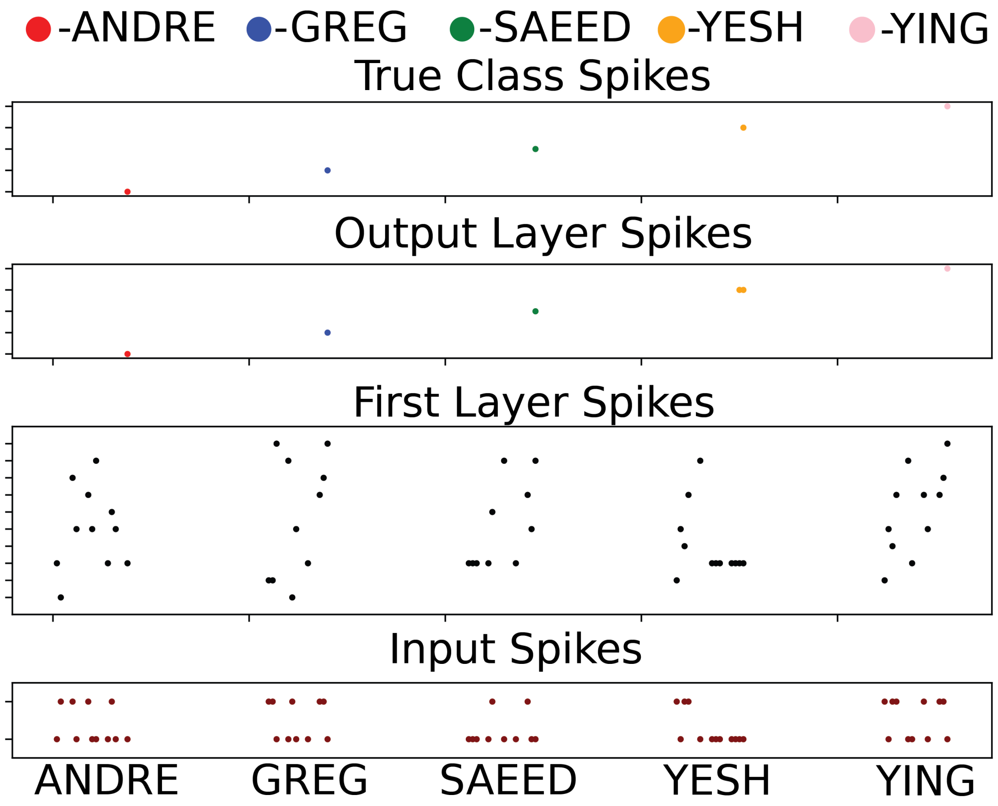


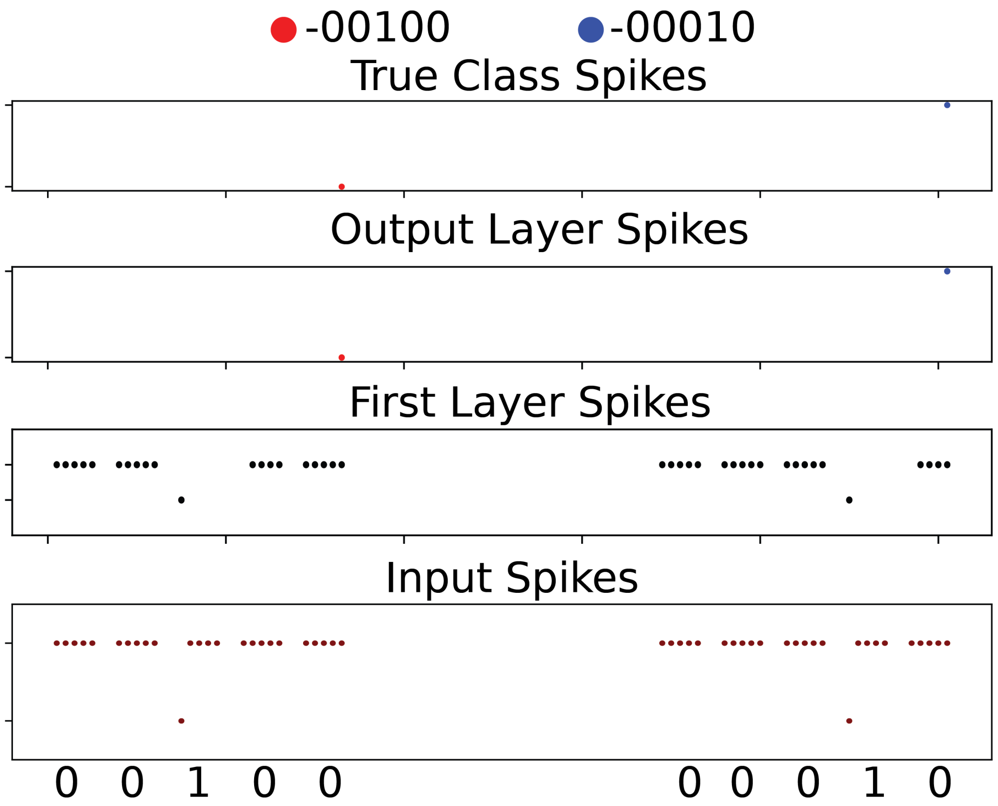
$W_O \times W_H$



WESTERN SYDNEY UNIVERSITY

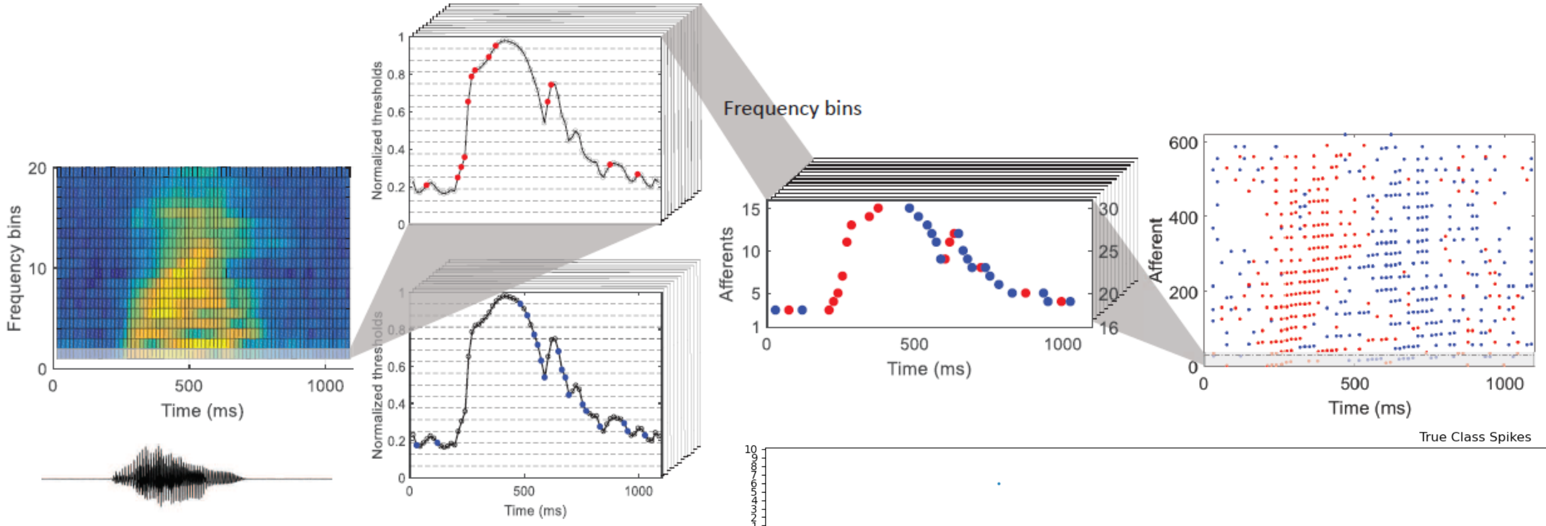
ODESA Results

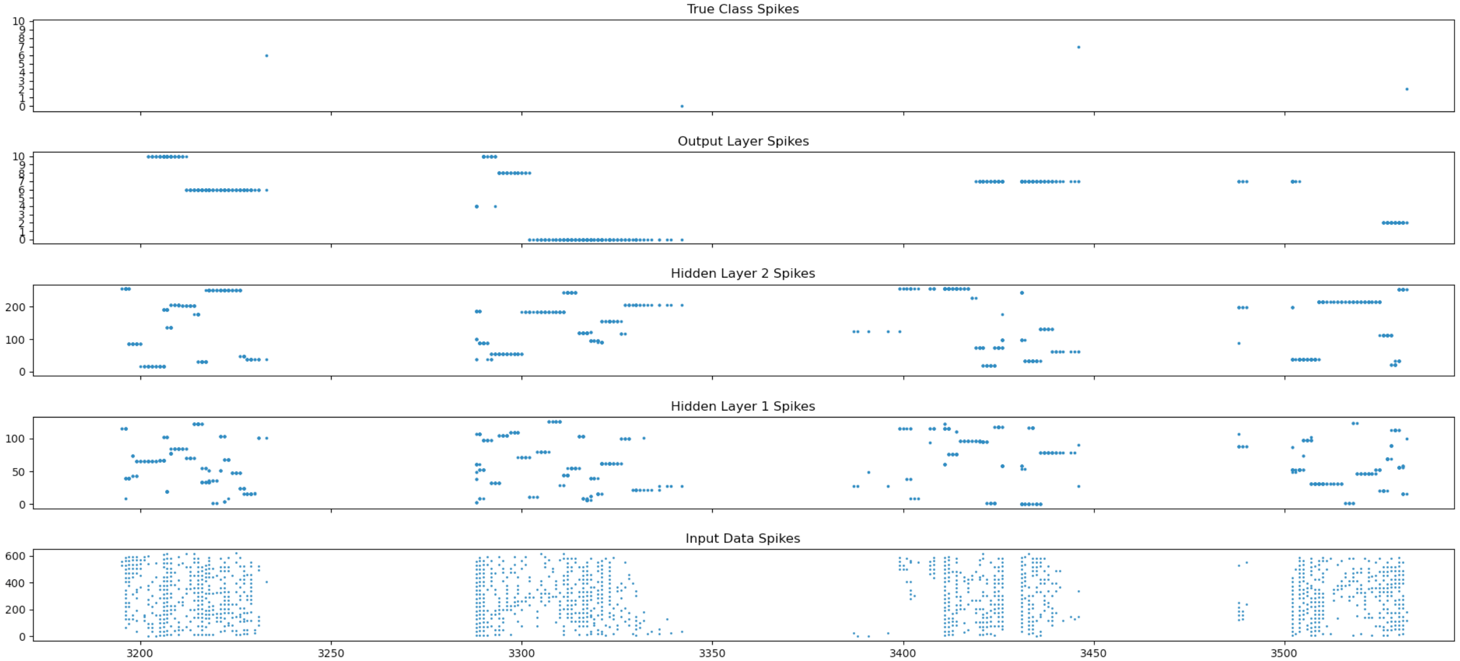




WESTERN SYDNEY UNIVERSITY

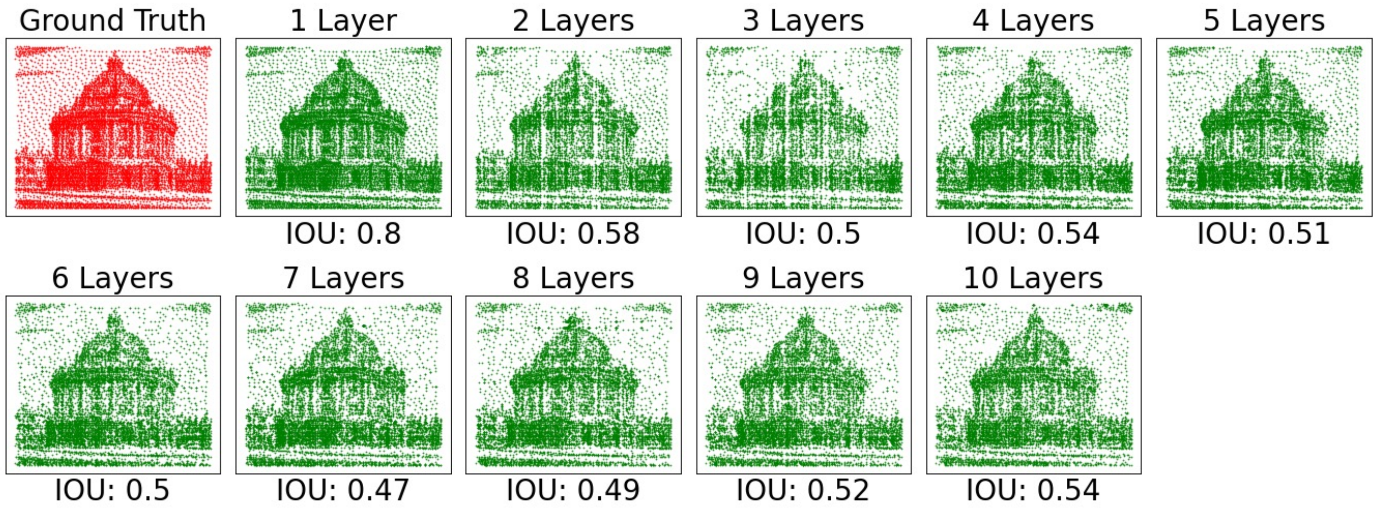
ODESA Results

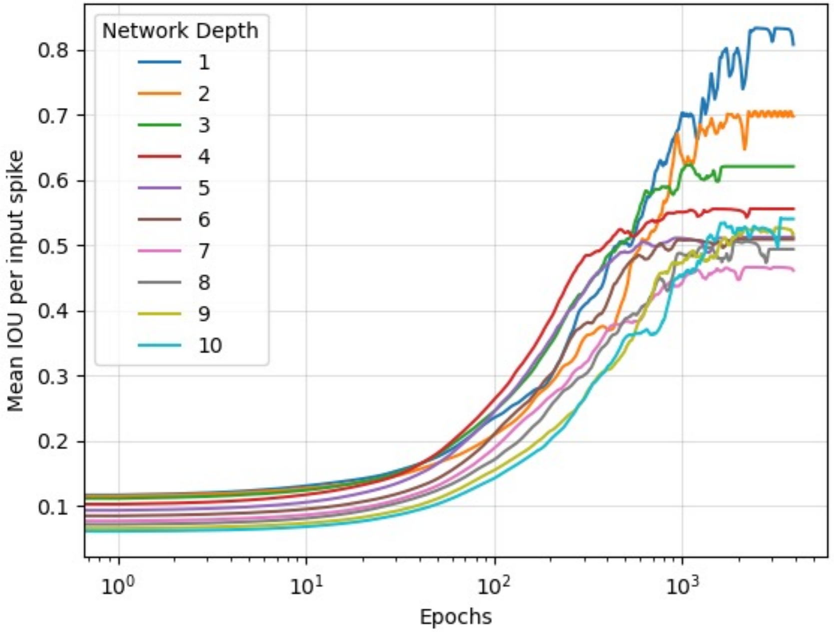




WESTERN SYDNEY UNIVERSITY

How deep is deep?





WESTERN SYDNEY UNIVERSITY

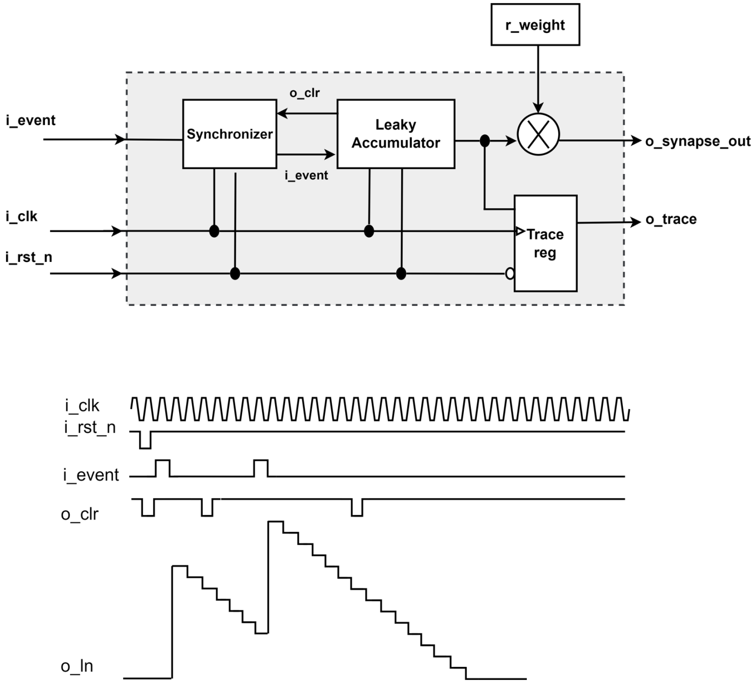
Advantages of ODESA

- Online continuous learning
- Learning hierarchical spatio-temporal features simultaneously at different time scales
- An end-to-end spiking architecture
- Event-driven and computationally efficient
- Gradient-free learning. An alternative to Error Backpropagation, and gradient descent
- Sparse activity due to hard WTA
- Modular and extensible

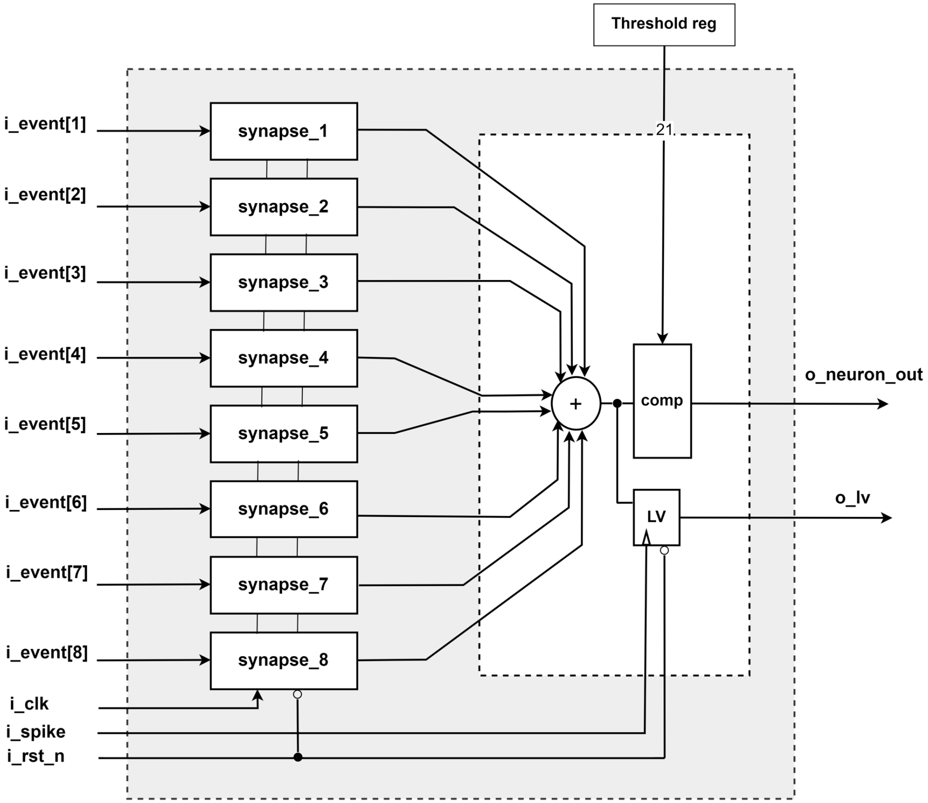
```
from ODESA.FullyConnected import FullyConnected as HiddenLayer
from ODESA.Classifier import Classifier as OutputLayer
from ODESA.FCModel import FCModel as Model

# Instantiate the hidden layers
input_layer = HiddenLayer(input_layer_context_rows,
                          input_layer_context_cols,
                          input_layer_n_neurons,
                          input_layer_eta,
                          input_layer_threshold_open,
                          input_layer_tau,
                          input_layer_trace_tau,
                          input_layer_thresh_eta,
                          cumulative_ts=True)

# Instantiate the output layer
output_layer = OutputLayer(output_layer_context_rows,
                           output_layer_context_cols,
                           output_layer_n_neurons_per_class,
                           output_layer_n_classes,
                           output_layer_eta,
                           output_layer_tau,
                           output_layer_threshold_open,
                           thresh_eta = output_layer_thresh_eta,
                           cumulative_ts=True)

# Initalize a Feast Model to have all the layers.
model = Model()
# Add hidden layers. Copy the statement to add more layers to the network.
model.add_hidden_layer(input_layer)
# Finally add the output layer. Every model has to have an output layer.
model.add_output_layer(output_layer)


# Each labelled event can be passed to the model using the forward function. If the event has no lal
# 'label' should be equal to '-1'.
hidden_layer_winners, output_layer_winner, output_class = model.forward((x,y,ts,1),label)
```

WESTERN SYDNEY UNIVERSITY

A digital implementation of ODESA



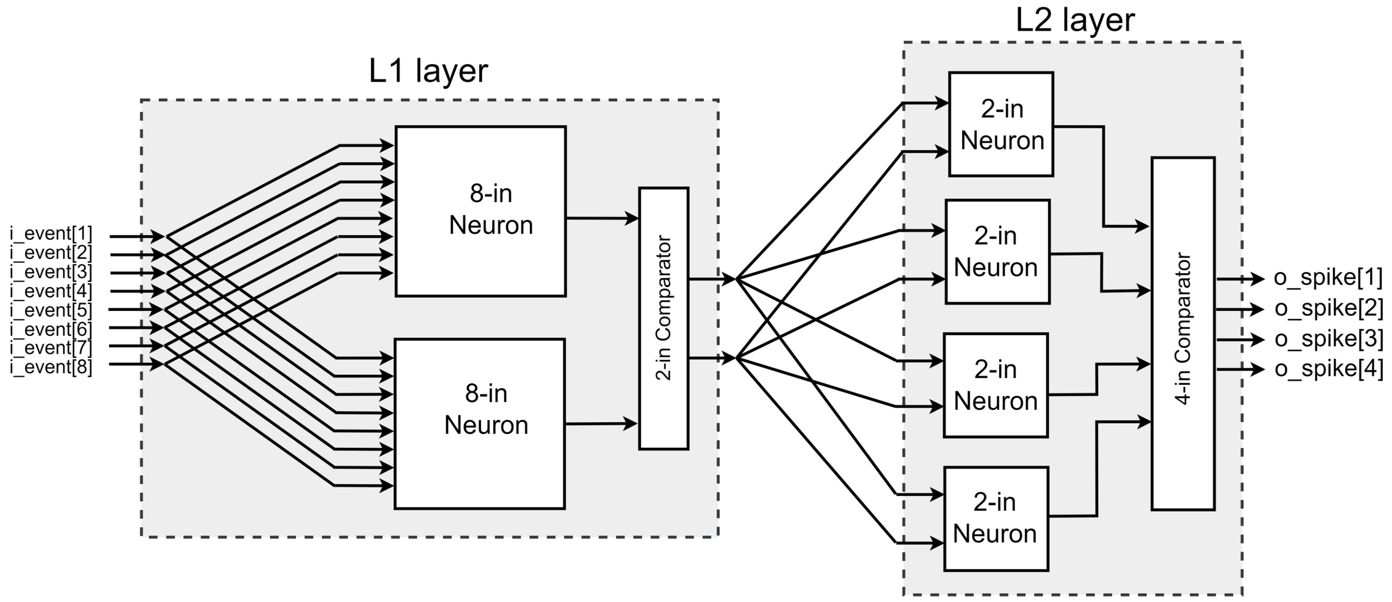
Synapse Design



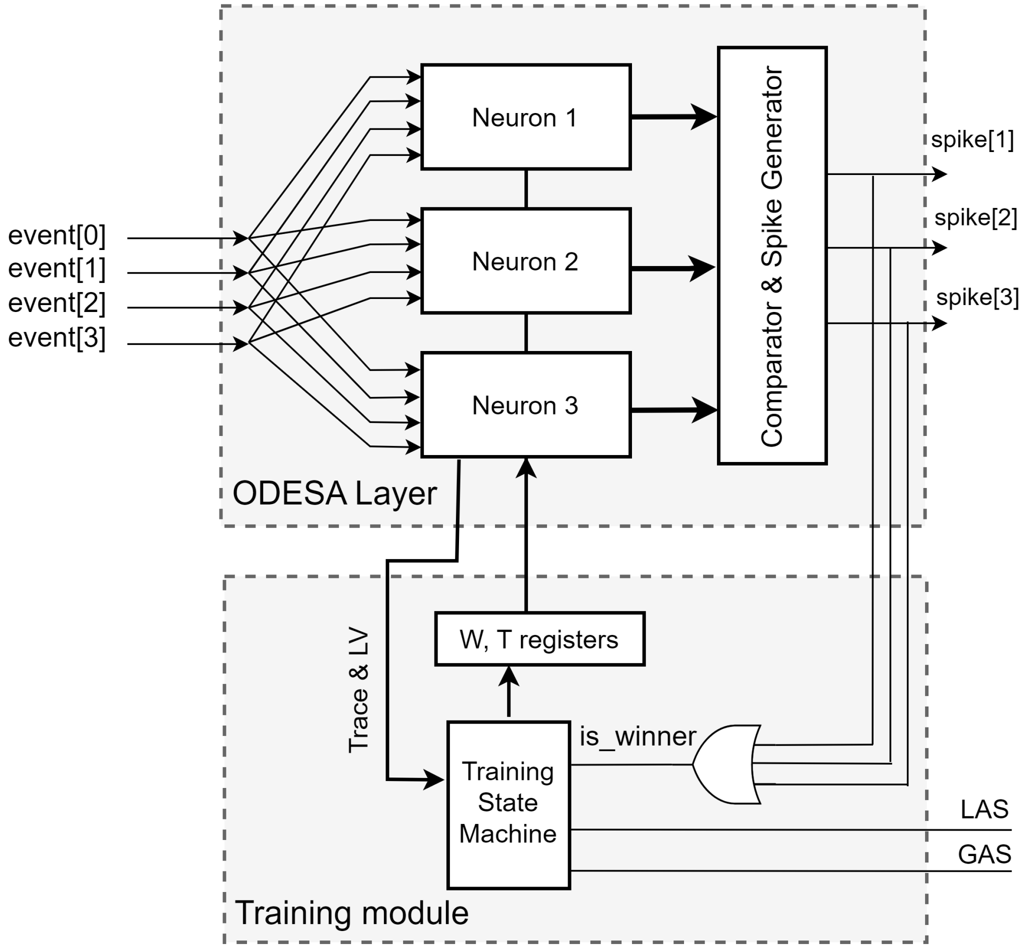
Neuron Design

WESTERN SYDNEY UNIVERSITY

A digital implementation of ODESA

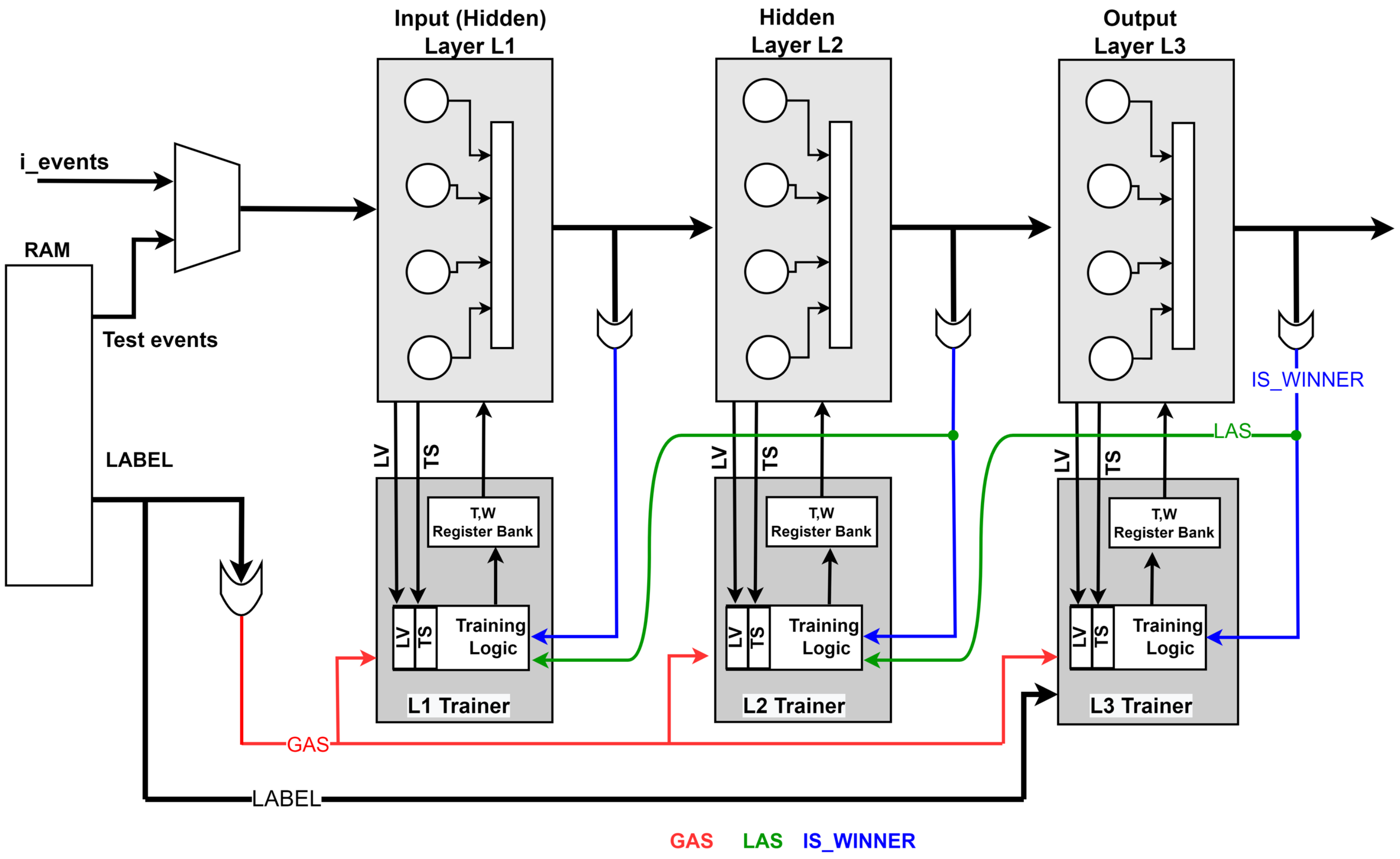


Layer Design



WESTERN SYDNEY UNIVERSITY

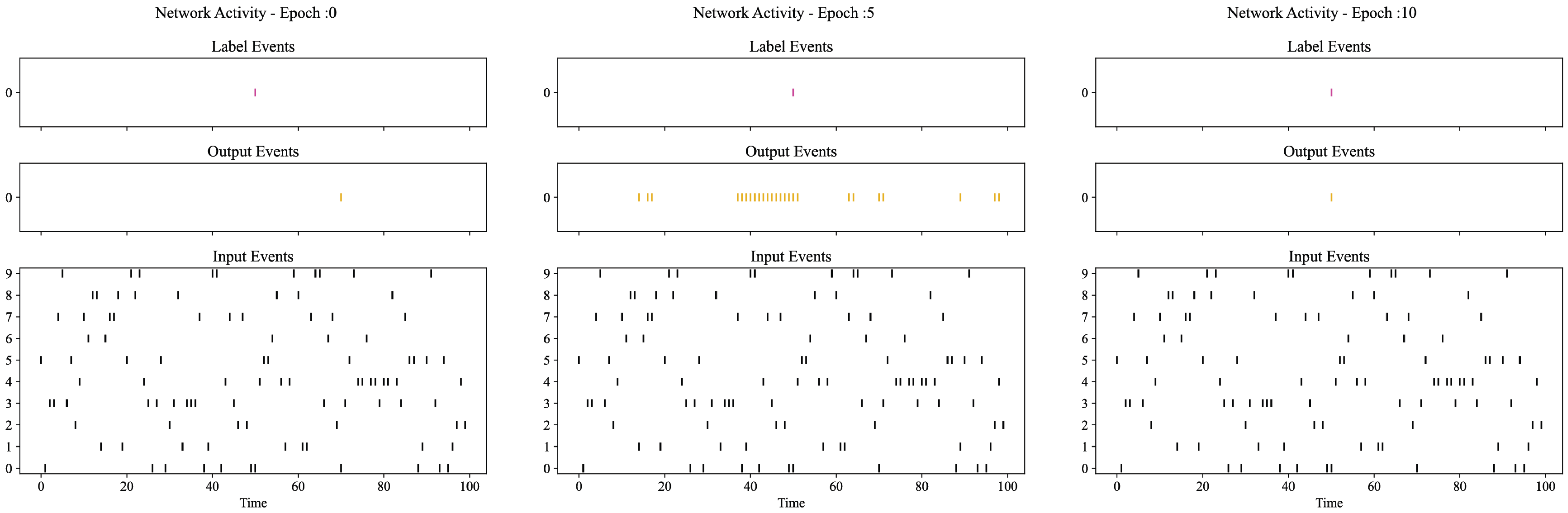
A digital implementation of ODESA



Mehrabi, Ali, et al. (2023) "An Optimized Multi-layer Spiking Neural Network implementation in FPGA Without Multipliers." Procedia Computer Science

WESTERN SYDNEY UNIVERSITY

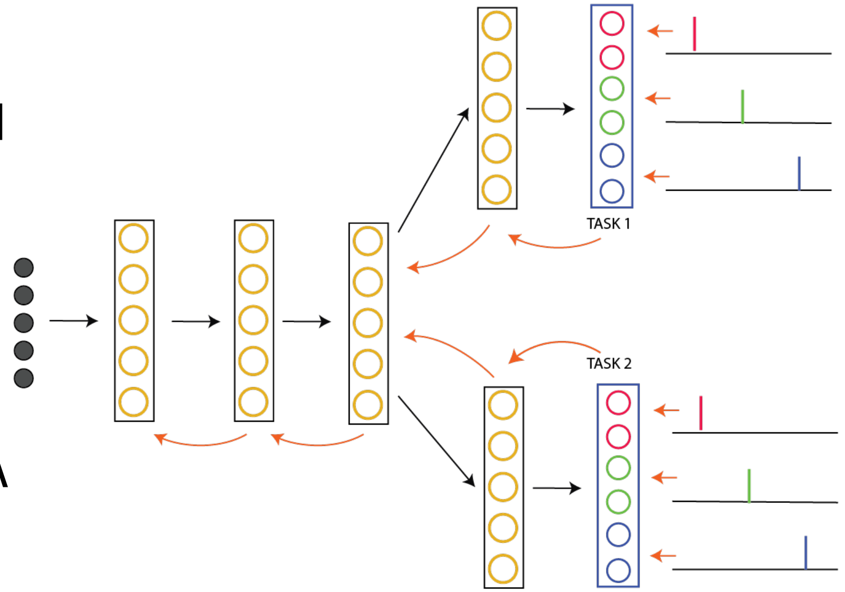
Conclusion



WESTERN SYDNEY UNIVERSITY

Conclusion

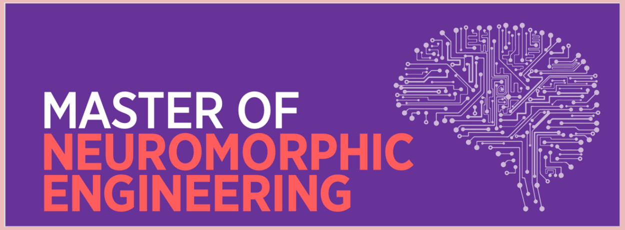
- Spike-timing-dependent threshold adaptation is the underlying principle for all the ODESA architectures
- The operation and memory of each individual computational component of ODESA networks are independent of each other, and thus enable scalability unlike ANNs
- The entirety of computation in all the layers and components of ODESA architectures is event-driven and thus asynchronous.
- The entirety of communication between each component in ODESA architectures including the feedback signals is binary event-based
- The solutions offer blueprints for future neuromorphic hardware design



WESTERN SYDNEY
UNIVERSITY

Thank You!

Check out: https://www.westernsydney.edu.au/icns/masters_program

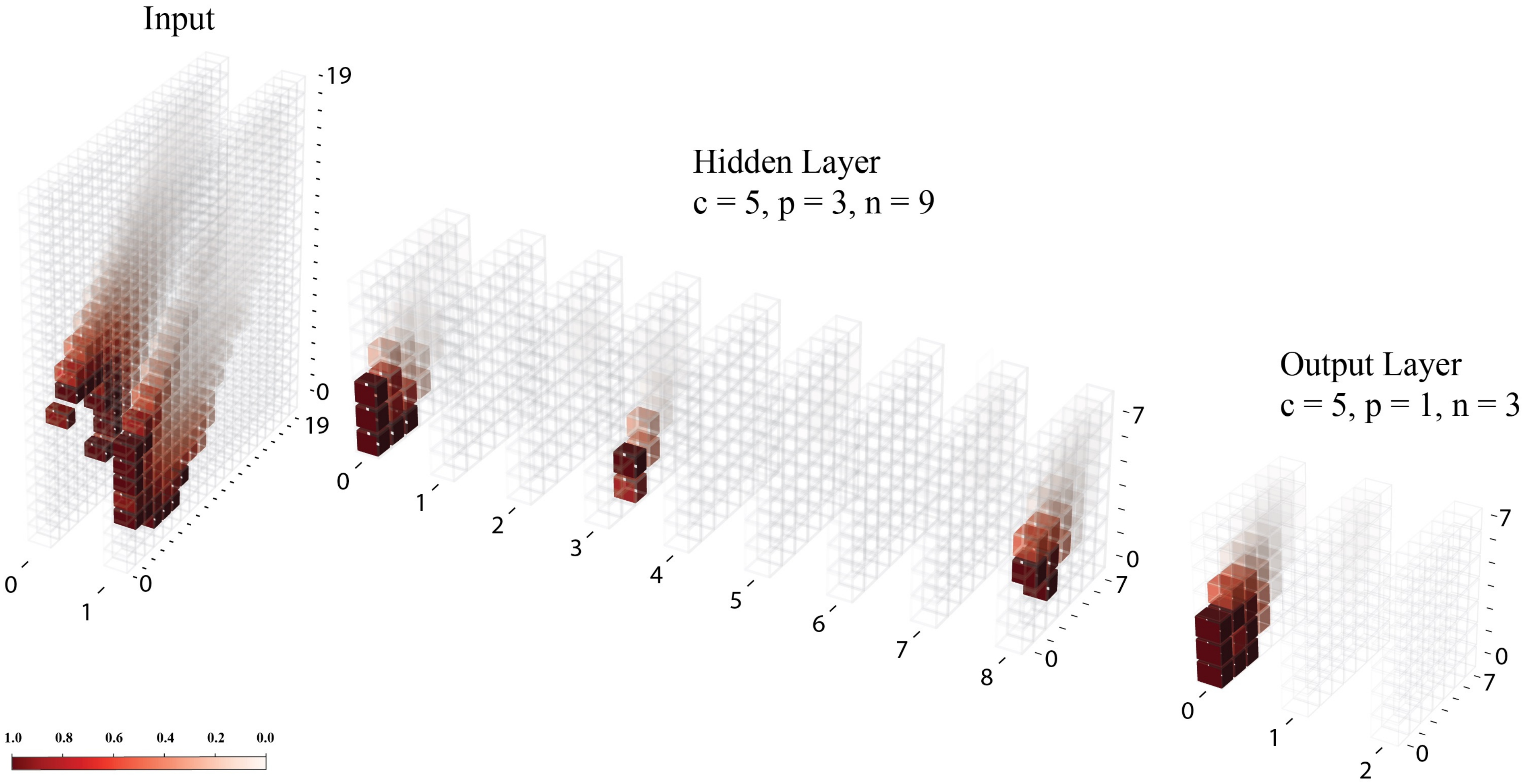


Email: Y.Bethi@westernsydney.edu.au

Additional Slides

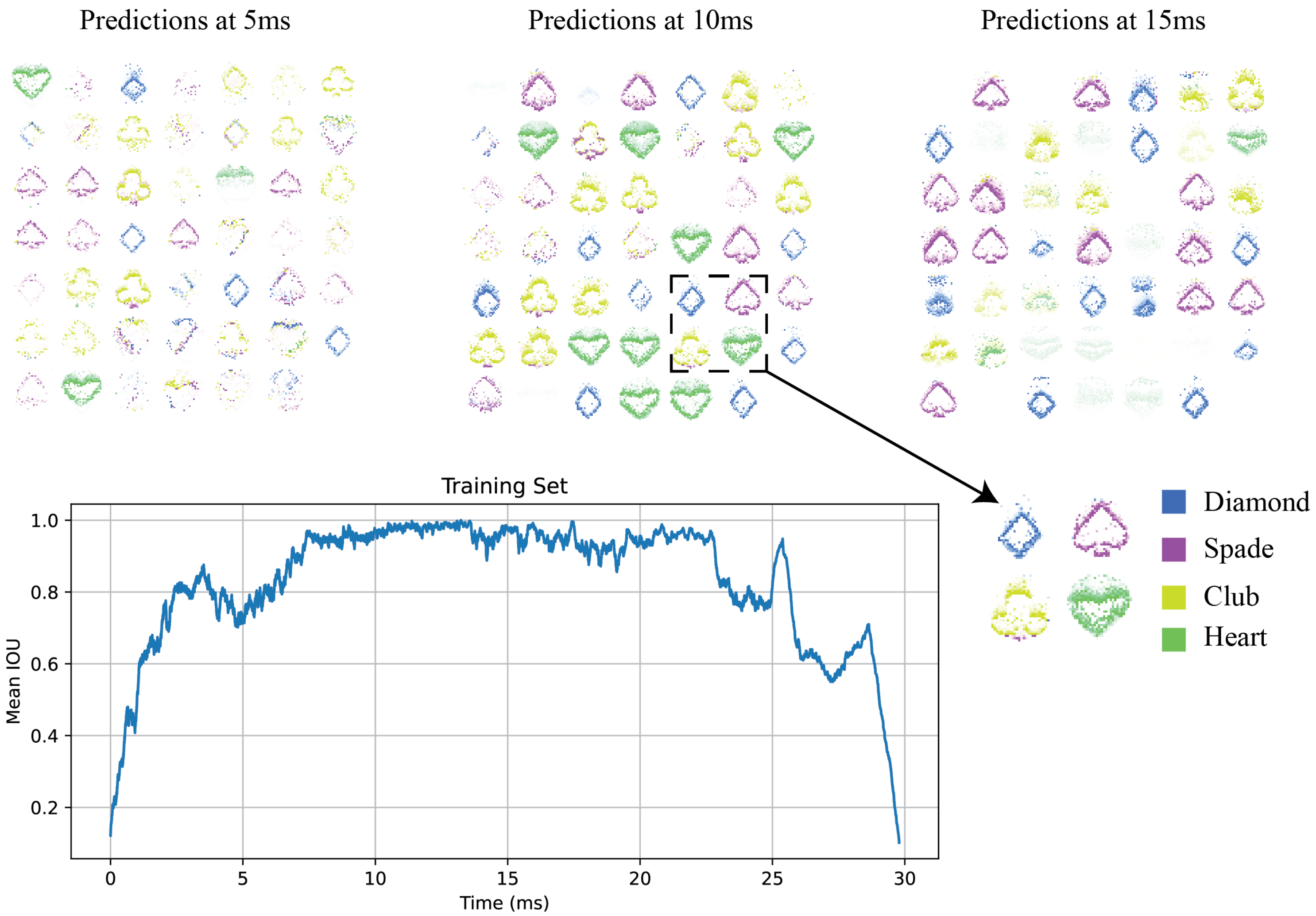
WESTERN SYDNEY UNIVERSITY

Convolutional ODESA



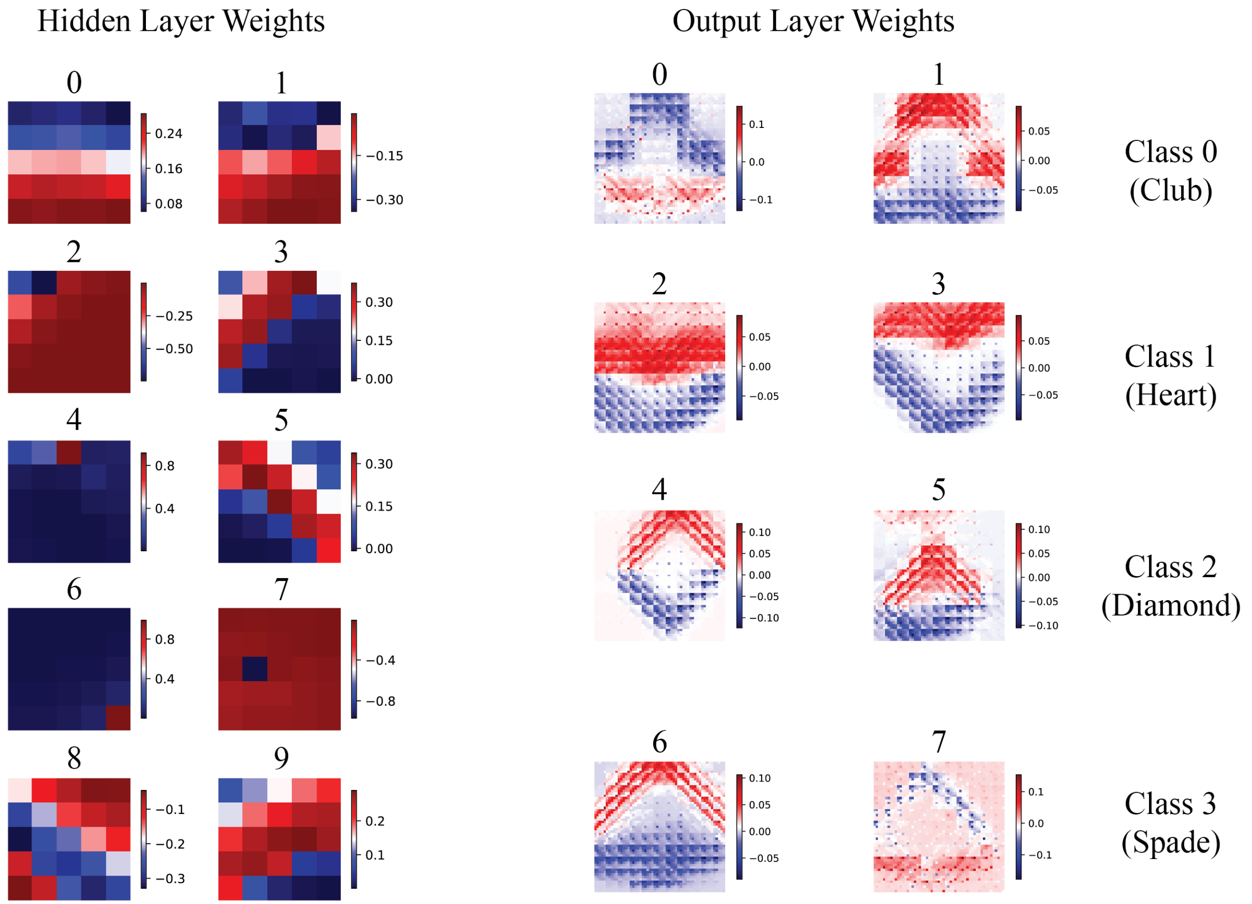
WESTERN SYDNEY UNIVERSITY

Conv Results

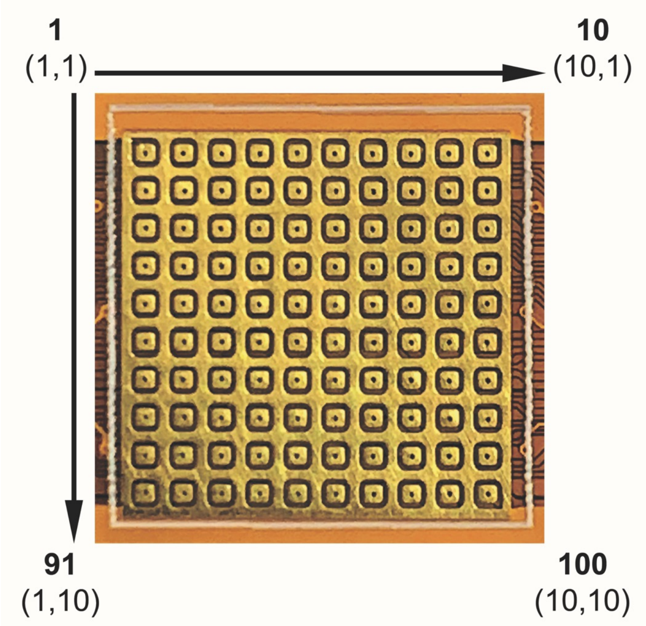


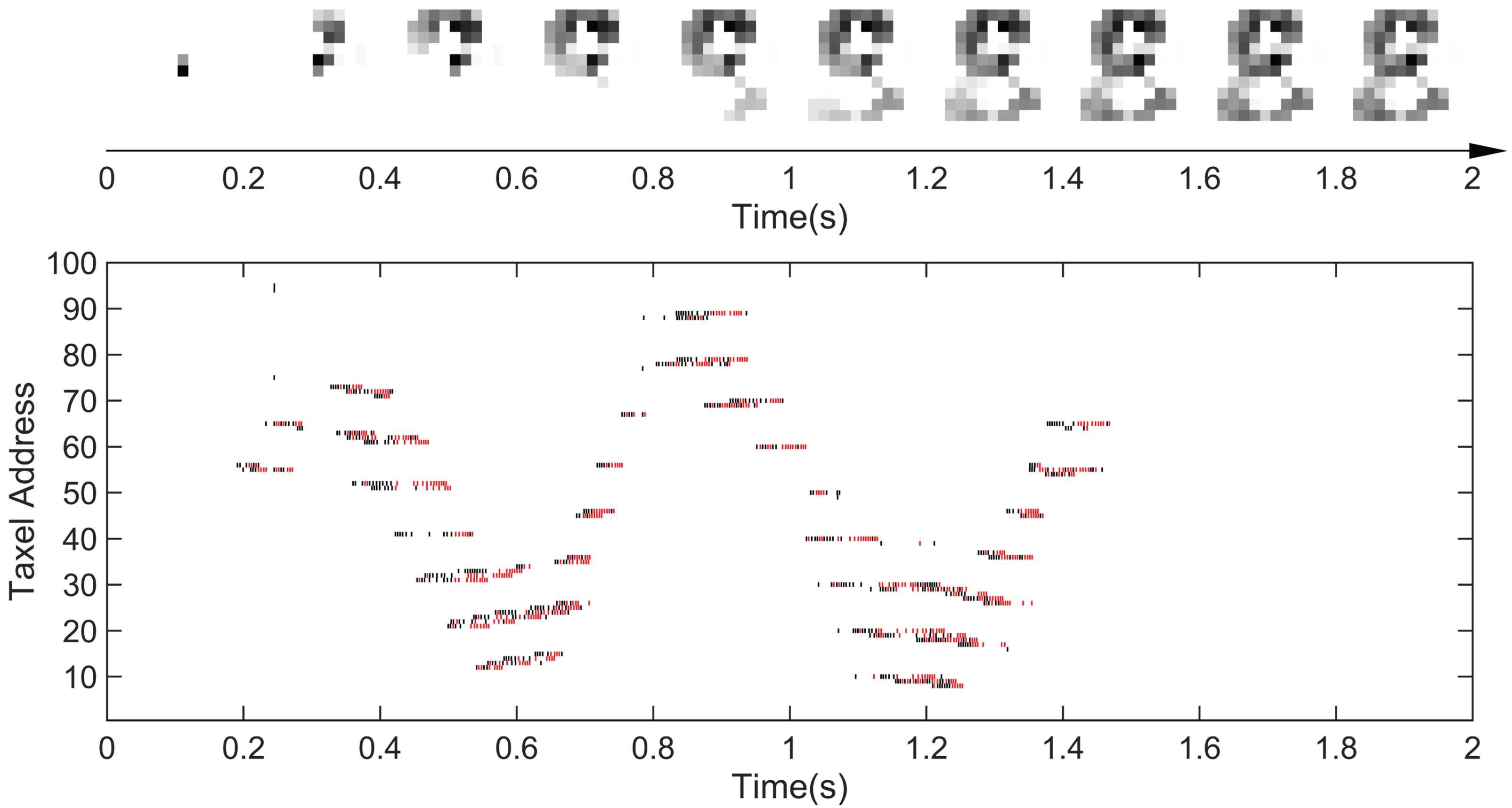
WESTERN SYDNEY UNIVERSITY

Conv Results



Tactile MNIST

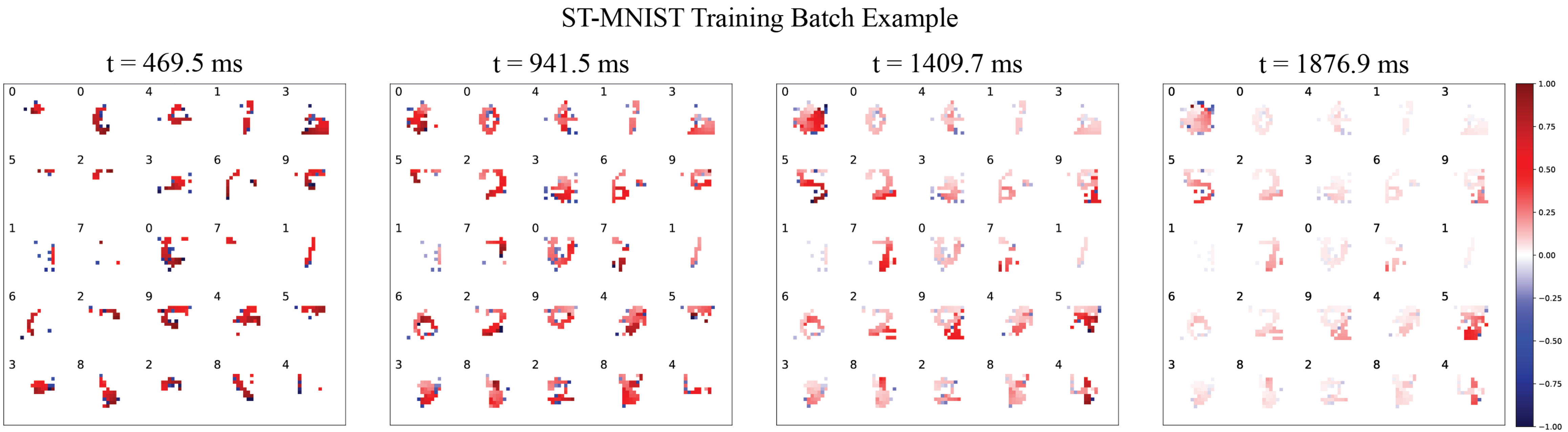




Figures from: *See, H.H., Lim, B., Li, S., Yao, H., Cheng, W., Soh, H. and Tee, B.C., 2020. ST-MNIST--The Spiking Tactile MNIST Neuromorphic Dataset. arXiv preprint arXiv:2005.04319.*

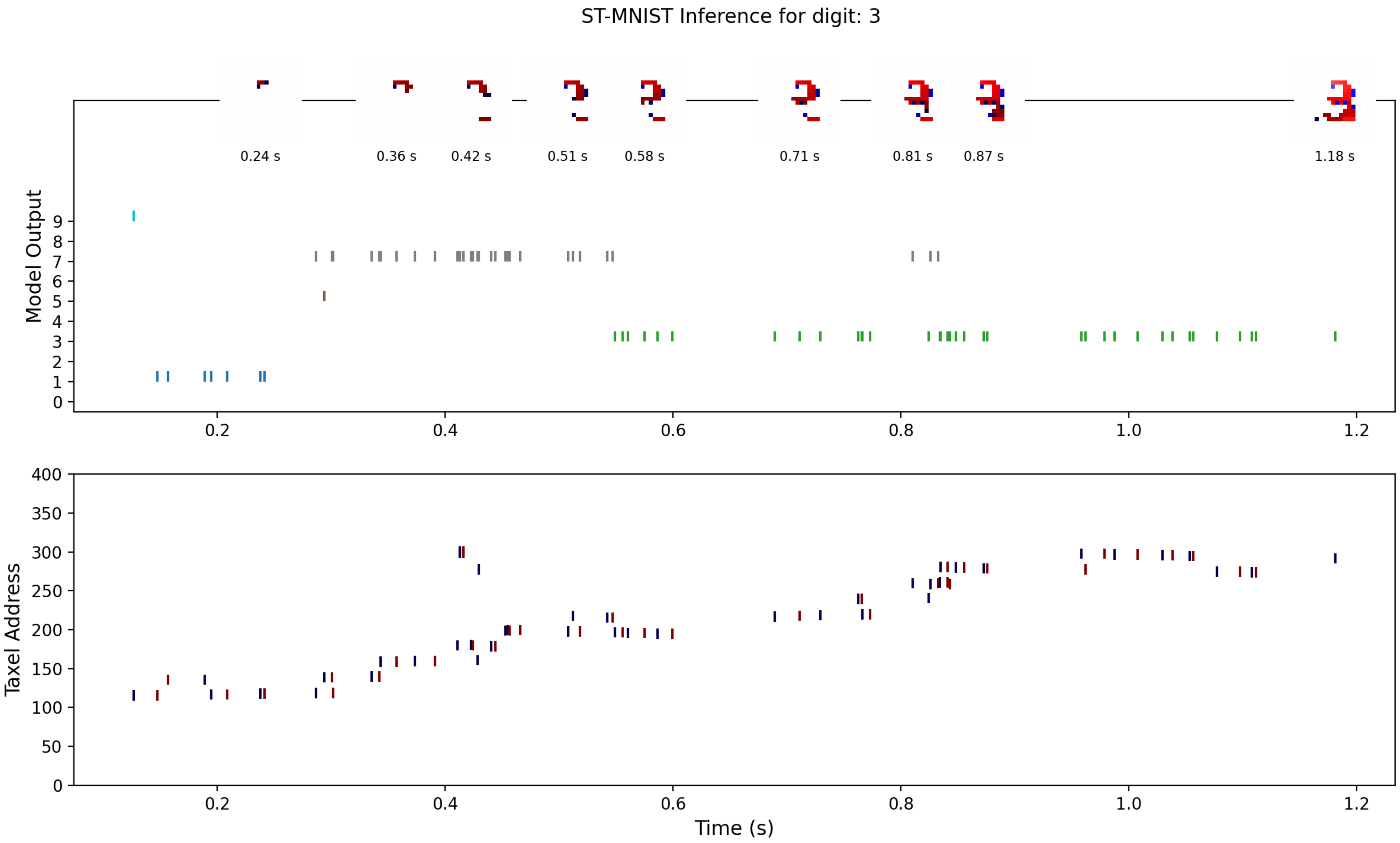
WESTERN SYDNEY UNIVERSITY

Tactile MNIST



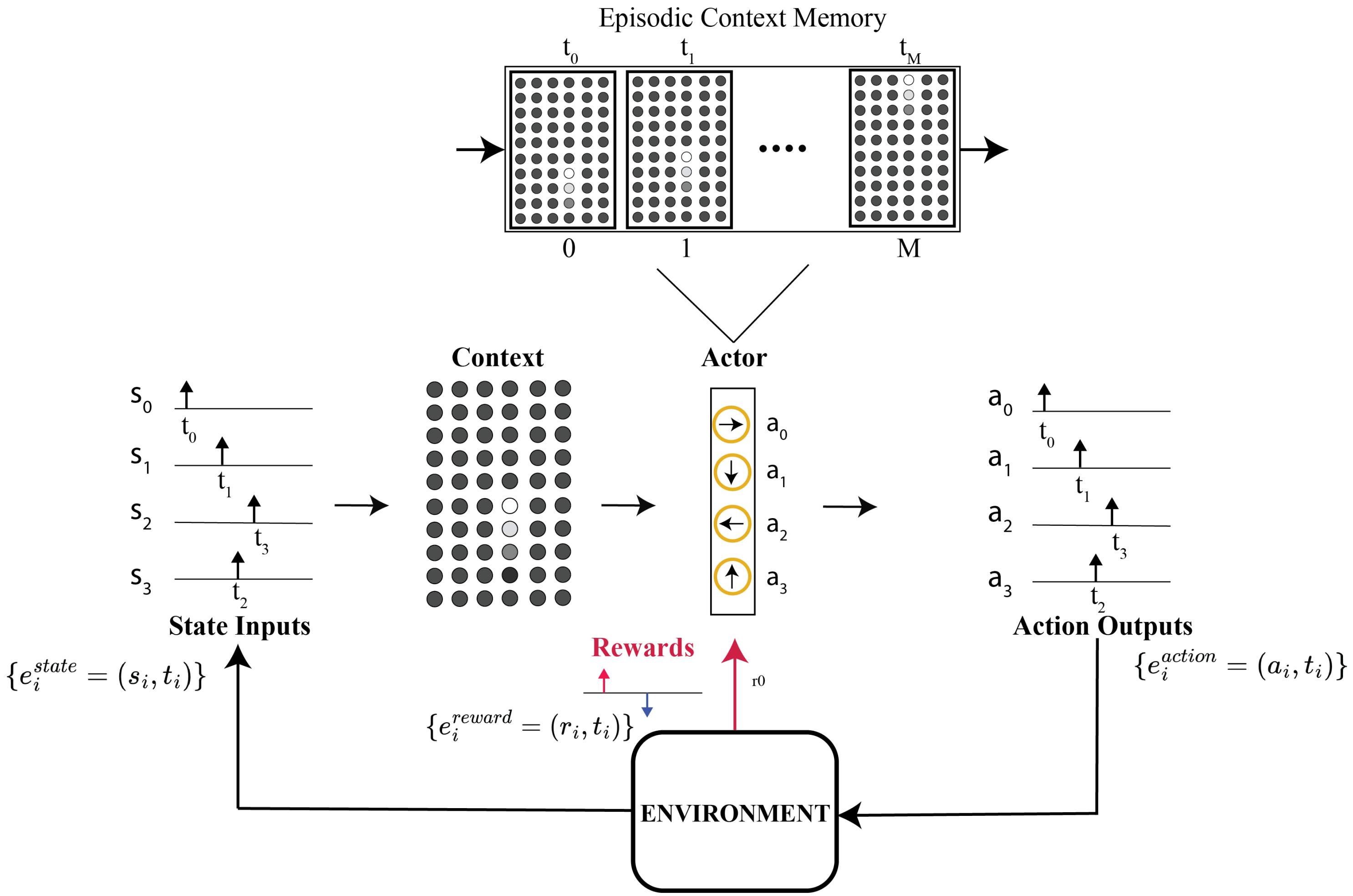
WESTERN SYDNEY UNIVERSITY

Conv Results



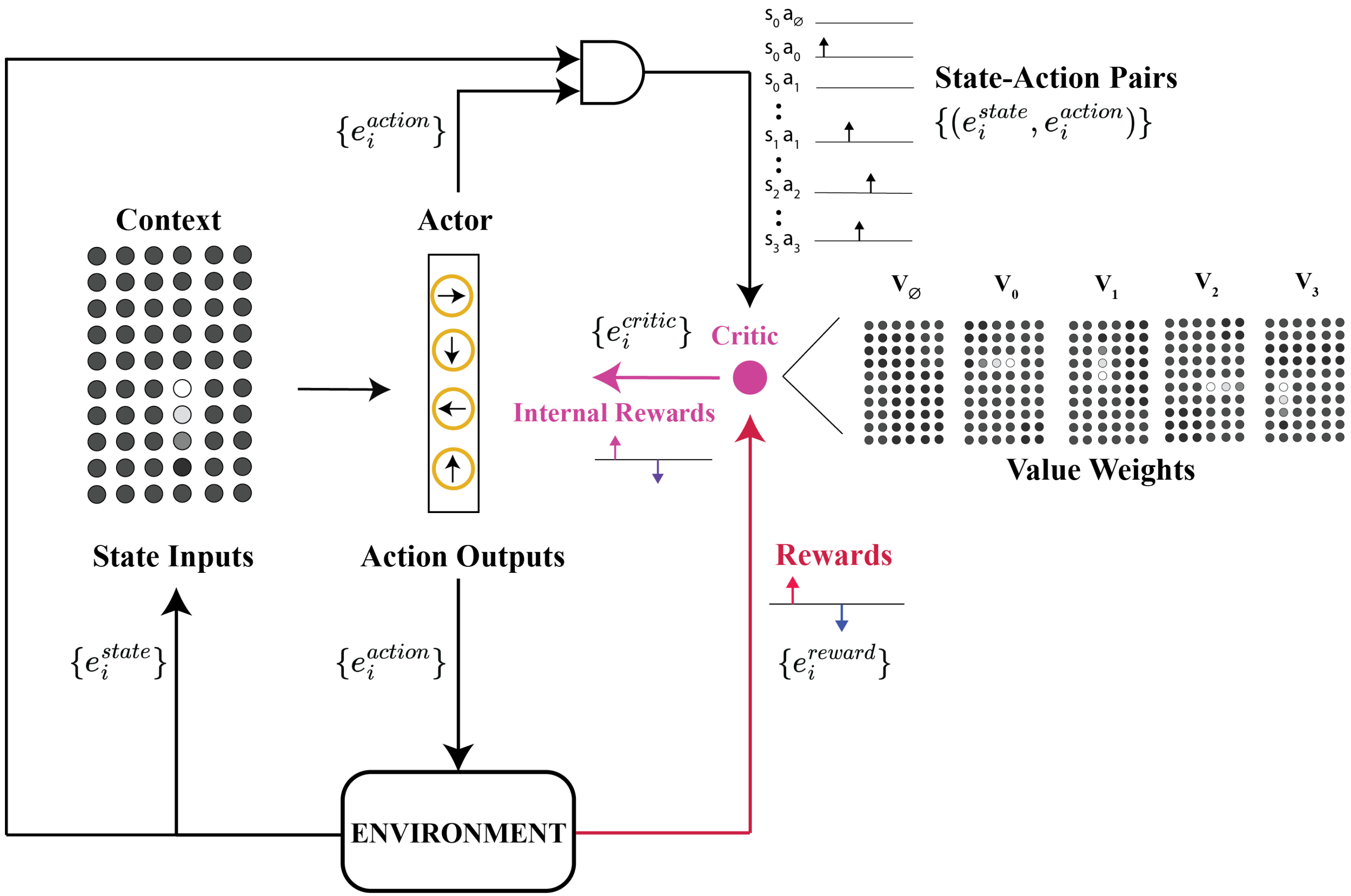
WESTERN SYDNEY UNIVERSITY

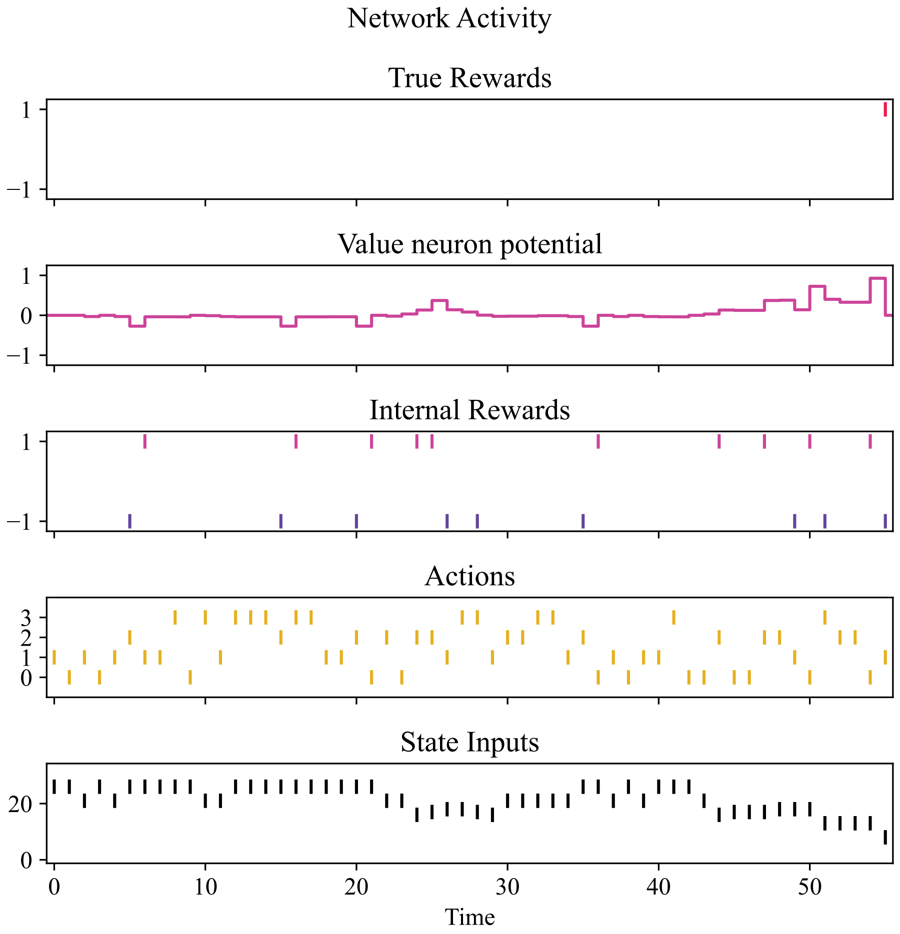
Actor layer with Episodic Context Memory



WESTERN SYDNEY UNIVERSITY

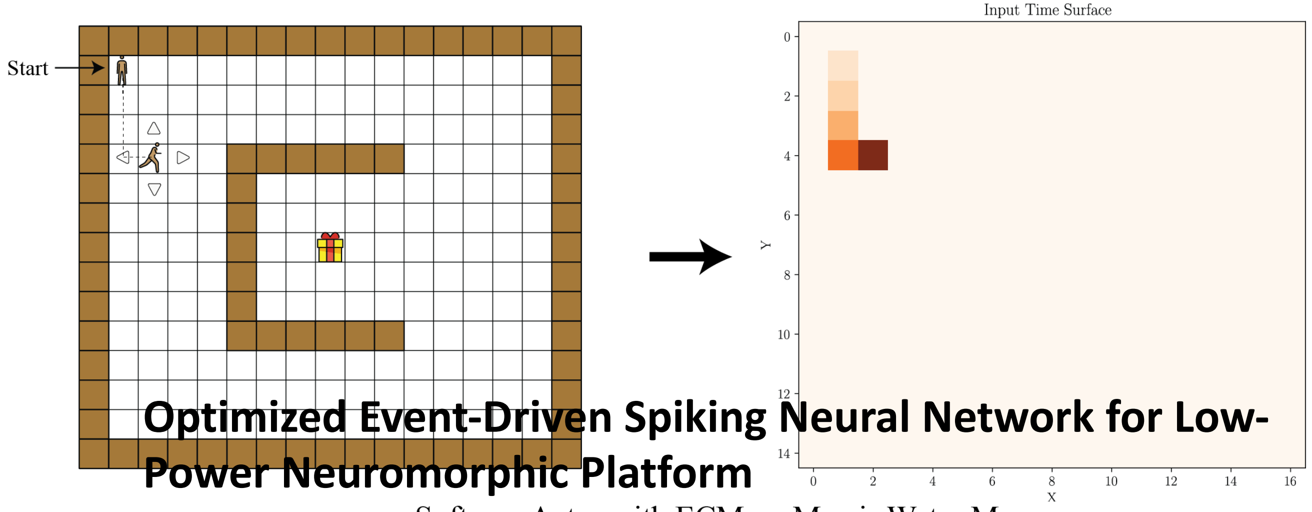
Actor Critic Architecture





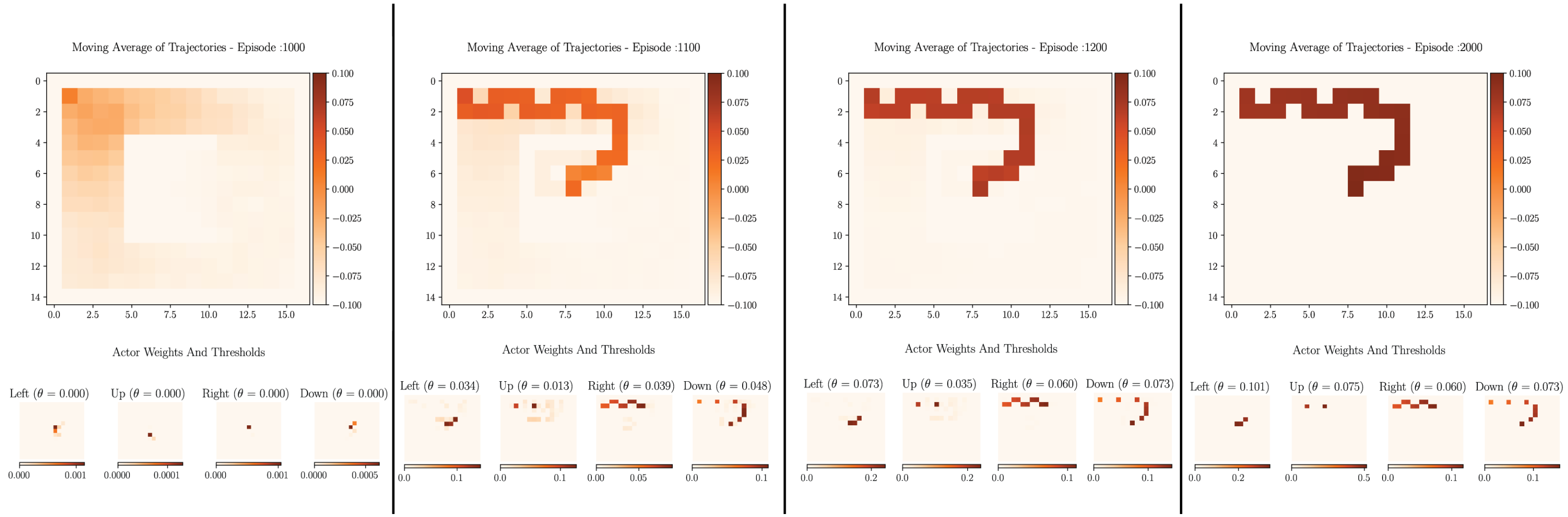
WESTERN SYDNEY UNIVERSITY

Grid Worlds



Optimized Event-Driven Spiking Neural Network for Low-Power Neuromorphic Platform

Softmax Actor with ECM on Morris Water Maze



TINY ML ASIA
Seoul, South Korea
November 16, 2023

Copyright Notice

www.tinyml.org