Cognitive Domain Ontologies Based on Loihi Spiking Neurons Implemented Using a Confabulation Inspired Network

June 25, 2019

Chris Yakopcic (UD), Jacob Freeman (UDRI), Tarek M. Taha (UD), and Scott Douglass (711 HPW)

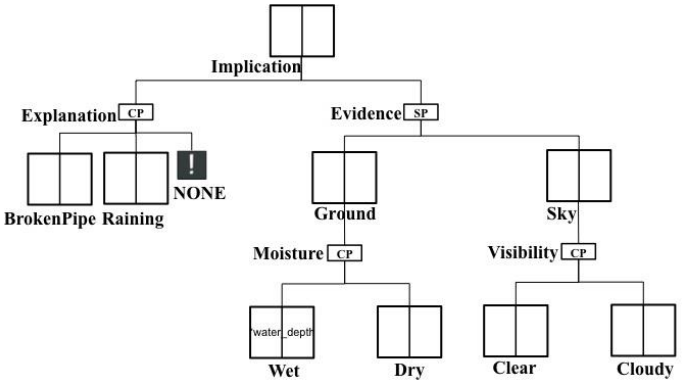
Cognitive Computing at UD

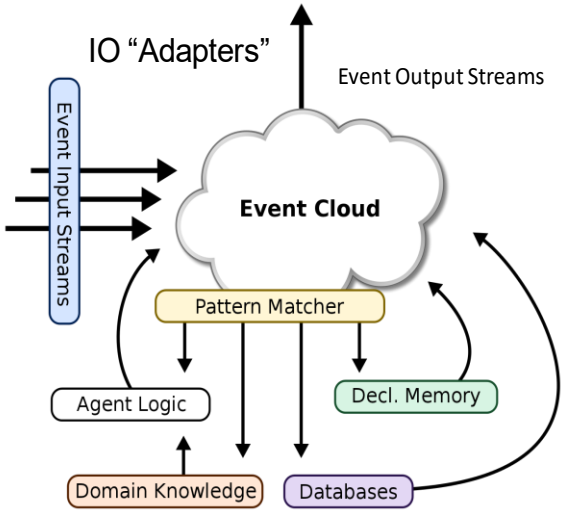
- Neuromorphic and Parallel Cognitive Systems Laboratory
  - Memristor Devices, Circuits, and Systems
  - Deep learning algorithms and applications
    - Medical imaging
    - Image understanding and enhancement
    - Cybersecurity
  - Parallel algorithms for cognitive agents (with AFRL)
  - **Spiking neural network algorithms for cognitive agents (with AFRL)**
  - **Application development for spiking neural processors: Intel Loihi and IBM TrueNorth**

University of Dayton Campus

CECEP Architecture

- The CDO is the decision making engine within the CECEP architecture
- These very simple examples quickly become very complex in realistic systems
  - Billions of possible outcomes

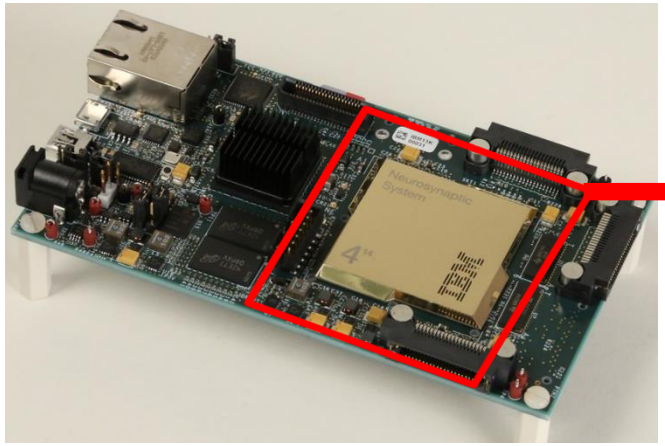


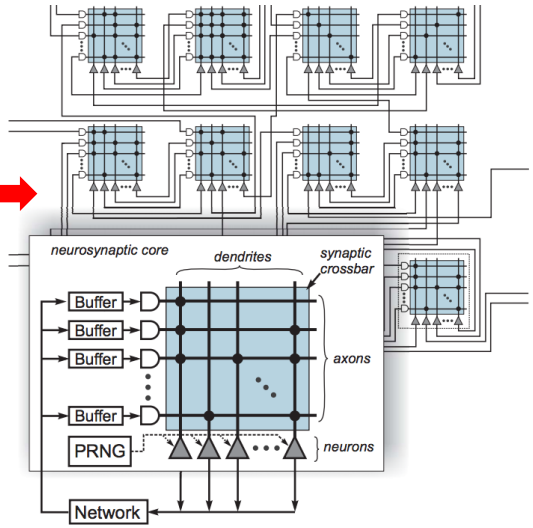


Name	Specification
Raining	IF Implication.explanation = Raining THEN Implication.evidence.Ground.moisture = Wet AND Implication.evidence.Sky.visibility = Cloudy
Broken Pipe	IF Implication.explanation = BrokenPipe THEN Implication.evidence.Ground.moisture = Wet OR (Implication.evidence.Sky.visibility = Clear AND NOT Implication.evidence.Ground.moisture = Dry)
Dry Ground	IFF NOT (Implication.explanation = Raining OR Implication.explanation = BrokenPipe) THEN Implication.evidence.Ground.moisture = Dry
Wet Ground	IFF Implication.evidence.Ground.moisture = Wet THEN Implication.explanation = Raining OR Implication.explanation = BrokenPipe

Objective

- Optimized resource allocation is extremely computationally expensive
- We need low SWaP alternatives, large problems are currently prohibitively expensive to solve.
- This is done using a series of spiking neurons that fire according to the most logical vehicle assignment options
- This work covers a MATLAB implementation of the spiking neuron based algorithm





Allocation Problem Size	Number of Possible Solutions
2 × 2	9
4 × 4	625
6 × 6	117,649
8 × 8	43,046,721
10 × 10	25,937,424,601

Outline

- CECEP Applications
- CDO Solving on a Confabulation Inspired Network
  - Loihi Introduction
  - Method
  - Algorithm
  - Execution on Loihi

Intel Loihi

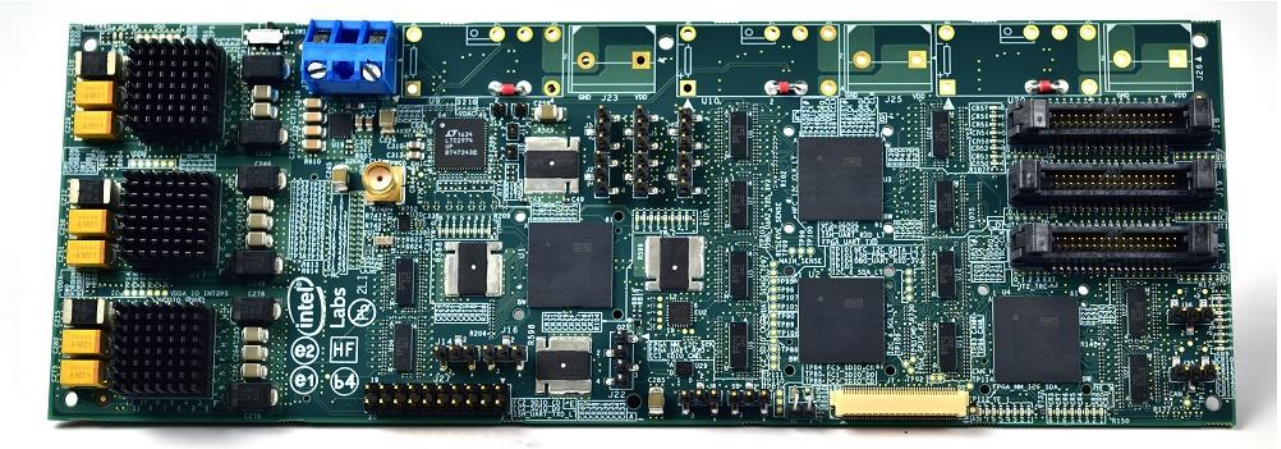
- Loihi Spiking Processor
- Manycore Neurmorphic System
  - 60 mm^2 chip
  - 14 nm profess
  - About 1,000,000 neurons per chip
  - Capable of implementing
    - Hierarchical connectivity
    - Dendritic compartments
    - Synaptic delays
    - Programmable synaptic learning rules

Intel Loihi

- Loihi Spiking Processor
- Manycore Neurmorphic System
  - 60 mm^2 chip
  - 14 nm profess
  - About 1,000,000 neurons per chip
  - USB stick system
    - Extremely portable low power cognitive system

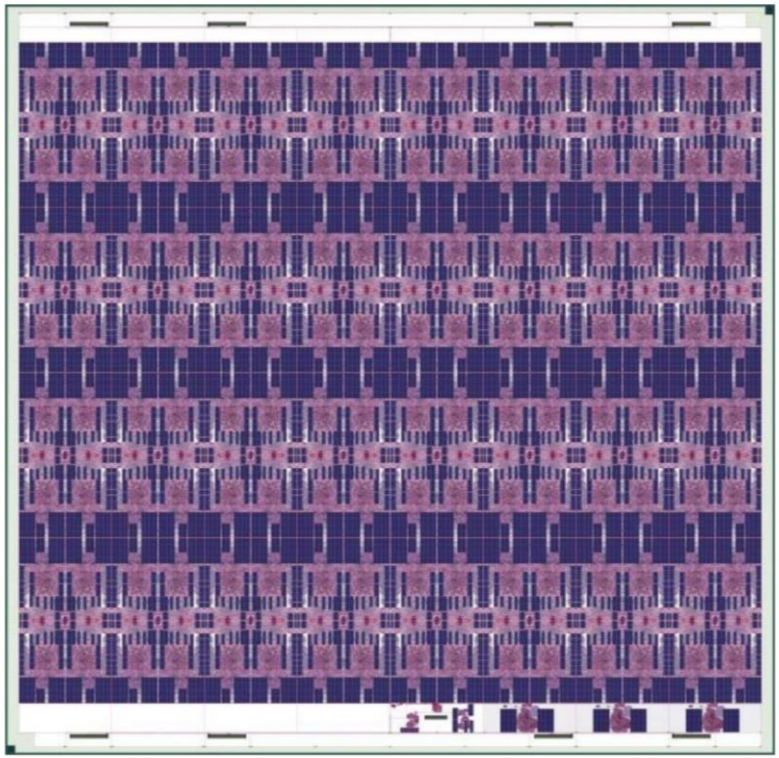
Intel Loihi

- Loihi Spiking Processor
  - 60 mm^2 chip
  - 14 nm profess
  - About 1,000,000 neurons
  - Alternative Multichip System
    - We utilize a multichip architecture via remote login for this work



Intel Loihi

- Loihi Spiking Processor
  - Multicore routing

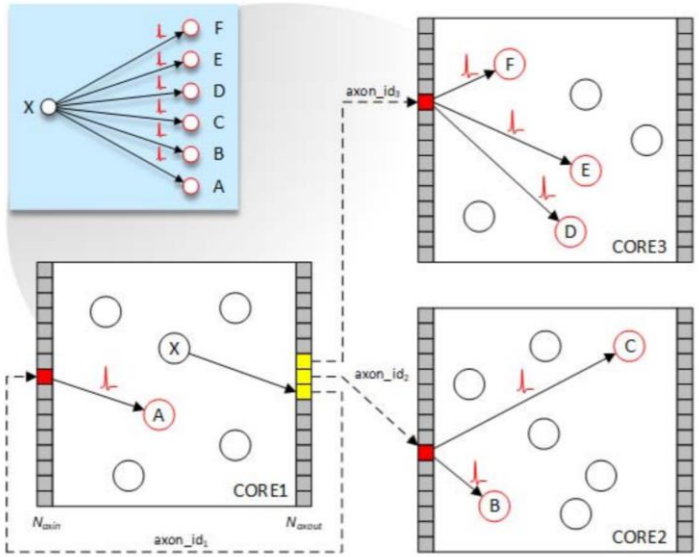


Loihi Chip Plot

Mike Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning" IEEE Micro Jan./Feb. 2018

Intel Loihi

- Loihi Spiking Processor
  - Multicore routing
  - Flexible Python user interface handles low level routing
  - Asynchronous communication
    - Low power design
    - Spikes only occur when needed to propagate information



Neuron to Neuron Routing

Mike Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning" IEEE Micro Jan./Feb. 2018

SWaP Comparison

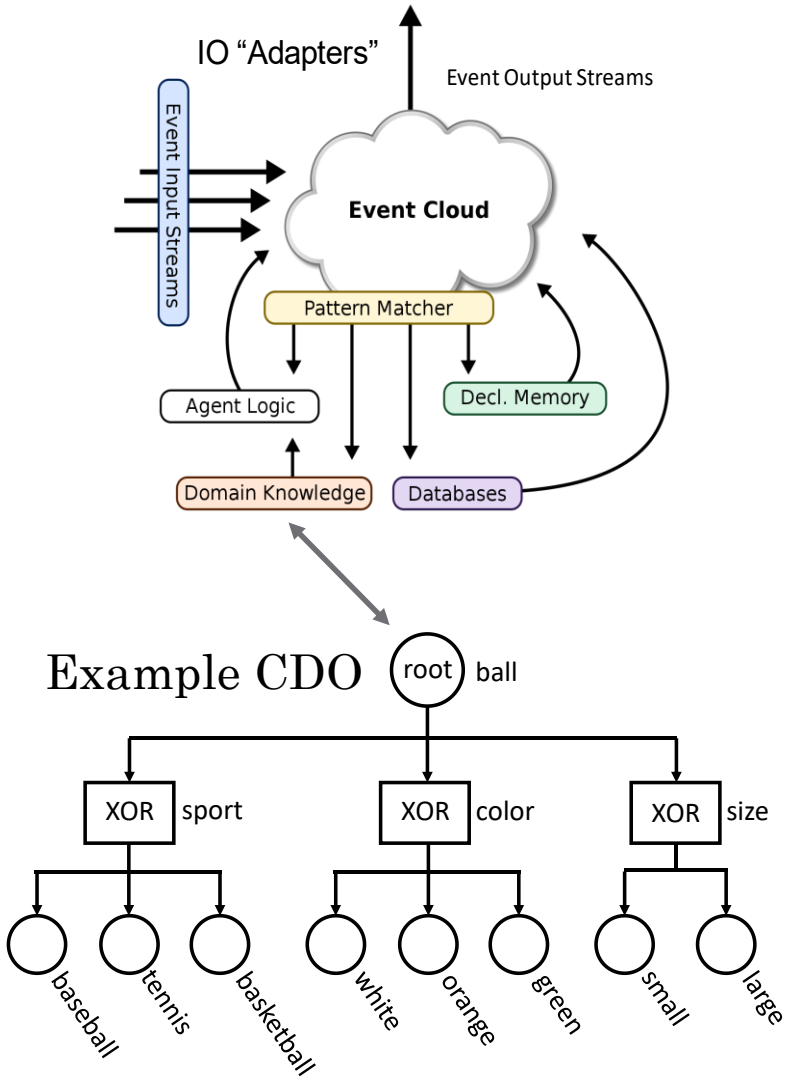
- General System Comparison
  - Exhaustive/Traditional vs. Embedded/Approximate
  - Approximate solution leads to dramatic increase in efficiency

	Trad. CPU / GPU System	Embedded System	Ratio
Size	2240 in^3	24 in^3	93×
Weight	20 lb	0.5 lb	40×
Power	500 W	70 mW	7142×
Accuracy	100%	99%	-

CDOs Using Confabulation

- The CDO is the decision making engine within the CECEP architecture
- Example Ball CDO
  - Demonstration Level
  - Small Scale

(a) If sport is baseball, then size is small and color is white.
(b) If sport is basketball, then size is large and color is orange.
(c) If size is small, then sport is baseball and tennis.



Real World CDO Example

- Complexity quickly grows in CDO structure
  - Solution space grows at a rapid exponential rate

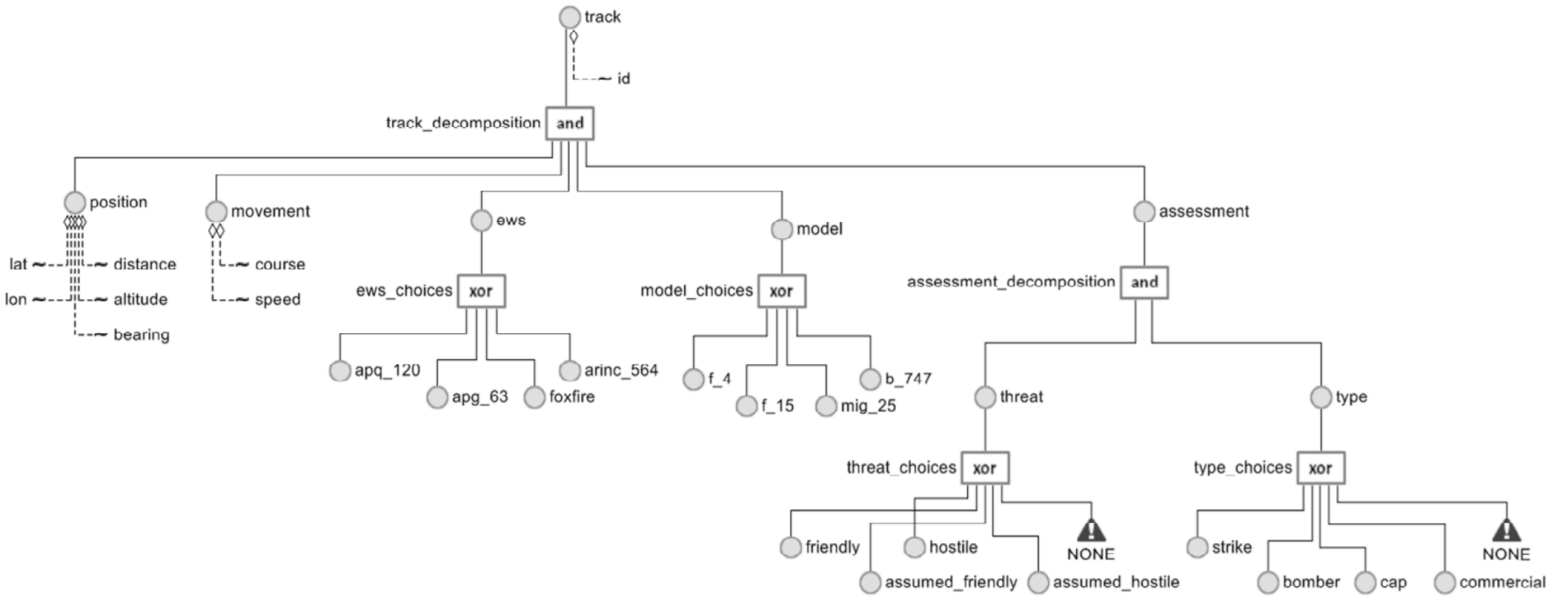
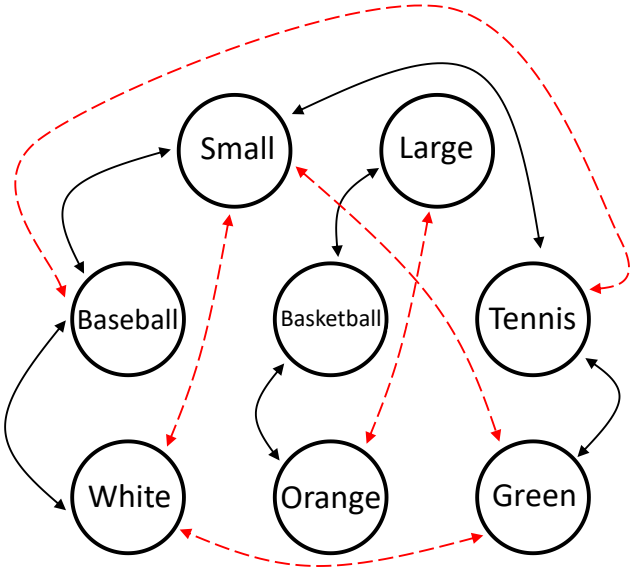


Fig. 2. A cognitive domain ontology capturing knowledge about track entities specified in RML.

T. Atahary, T. Taha, S. Douglass, "Hardware Accelerated Cognitively Enhanced Complex Event Processing", 14th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, July, SNPD 2013.

Loihi Implementation

- Implement the Ball CDO as a confabulation inspired network

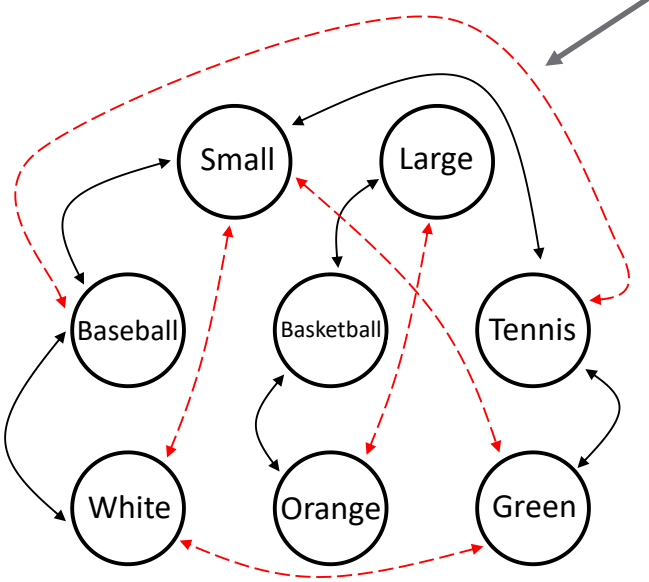


Loihi Implementation

- Implement the Ball CDO as a confabulation inspired network

```
CFD_array = np.array([[0, 0, 1, 0, 1, -1, 0, -1],   small
                      [0, 0, 0, 1, 0, 0, 0, 0],     large
                      [1, 0, 0, 0, -1, 1, 0, 0],    baseball
                      [0, 1, 0, 0, 0, 0, 1, 0],     basketball
                      [1, 0, -1, 0, 0, 0, 0, 1],    tennis
                      [-1, 0, 1, 0, 0, 0, 0, 0],    white
                      [0, -1, 0, 1, 0, 0, 0, 0],    orange
                      [-1, 0, 0, 0, 1, 0, 0, 0]])   green
```

small large baseball basketball tennis white orange green

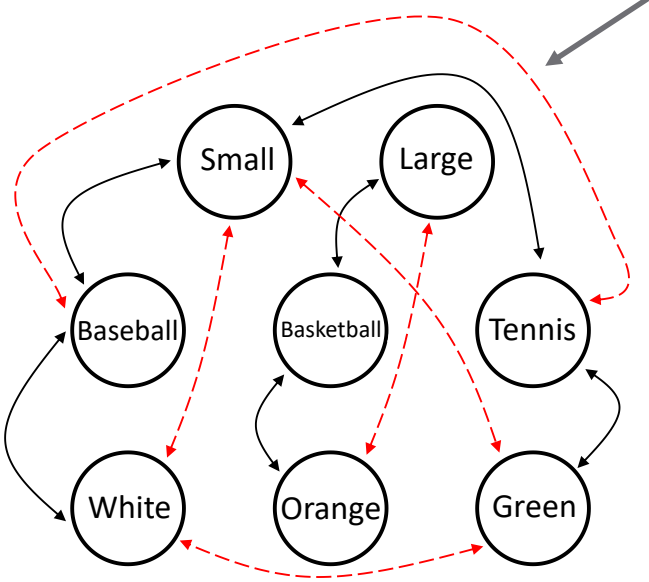


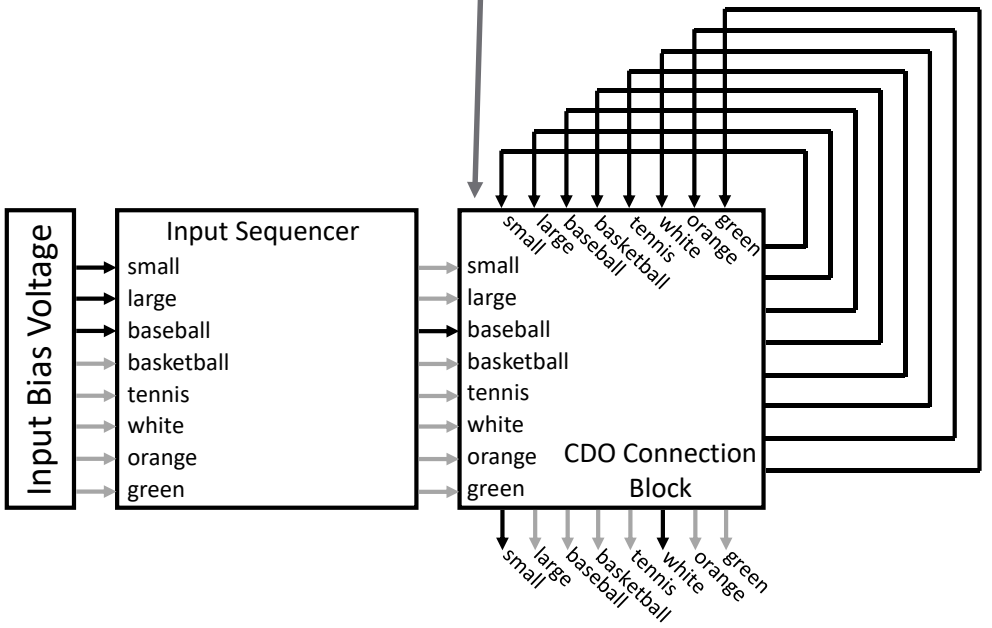
Loihi Implementation

- Implement the Ball CDO as a confabulation inspired network
- Sequencer block used to change input in real time

```
CFD_array = np.array([[0, 0, 1, 0, 1, -1, 0, -1],   small
                      [0, 0, 0, 1, 0, 0, 0, 0],     large
                      [1, 0, 0, 0, -1, 1, 0, 0],    baseball
                      [0, 1, 0, 0, 0, 0, 1, 0],     basketball
                      [1, 0, -1, 0, 0, 0, 0, 1],    tennis
                      [-1, 0, 1, 0, 0, 0, 0, 0],    white
                      [0, -1, 0, 1, 0, 0, 0, 0],    orange
                      [-1, 0, 0, 0, 1, 0, 0, 0]])   green
```

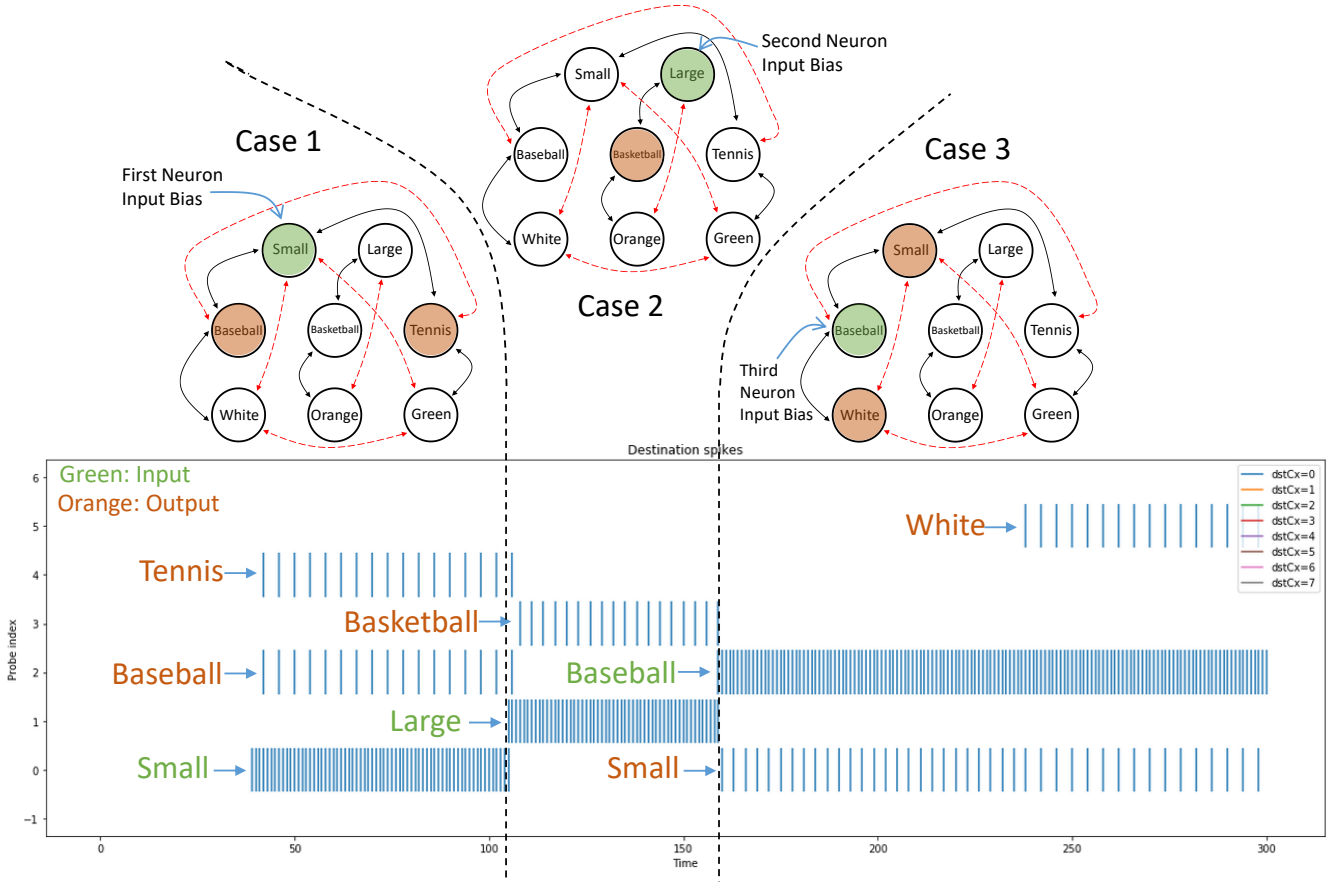
small large baseball basketball tennis white orange green





Spiking Result

- Results show three different example cases
- Inputs can change in real time according where bias is applied



Next Steps

- Confabulation CDO
  - Implement learning
  - Scalability Test
    - Some aspects of the spiking network may cause size limitations
      - Communication delay
      - Negative accumulation (fixed in our most recent work)

- Loihi
  - Constraint Solving
  - Energy / Timing Benchmark

- Next talk shows quantified results using these SNN systems for asset allocation