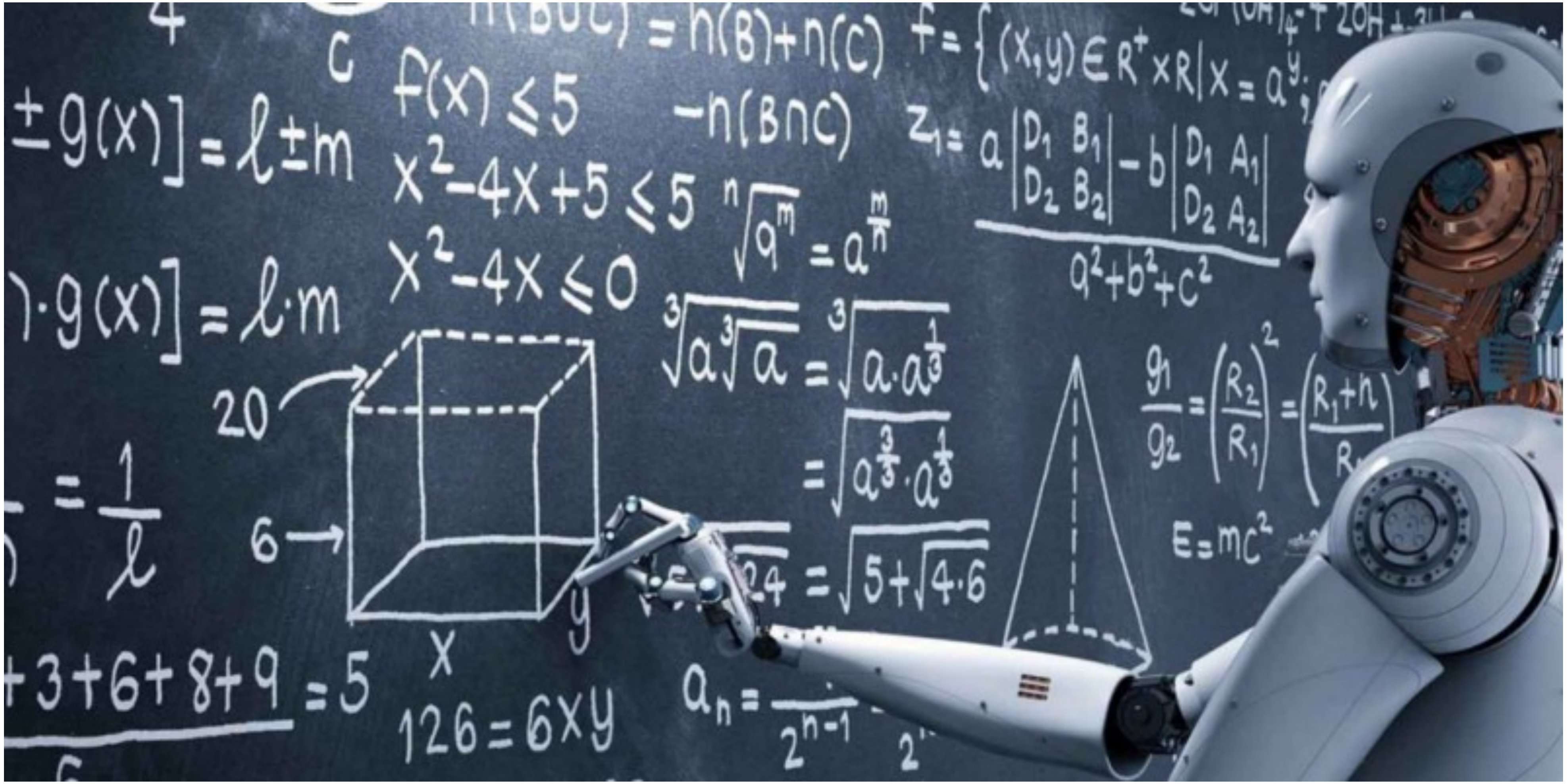
http://synergy.ece.gatech.edu

# Enabling Continuous Learning through Synaptic Plasticity in Hardware

Tushar Krishna
Georgia Tech

EMC$^2$ Workshop
June 23 2019

# The Dream!



# Deep Learning Applications

“AI is the new electricity” – Andrew Ng

**Object Detection**

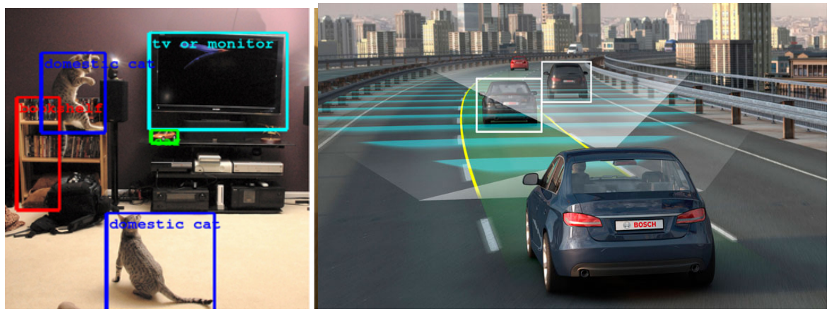
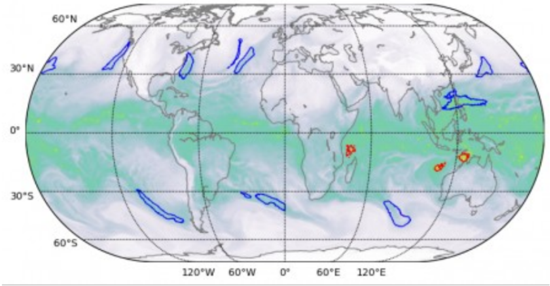
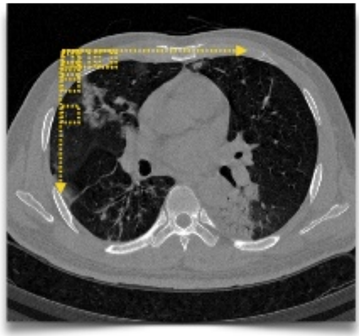



**Image Segmentation**

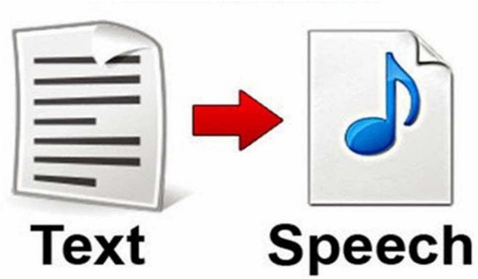



**Medical Imaging**



**Speech Recognition**

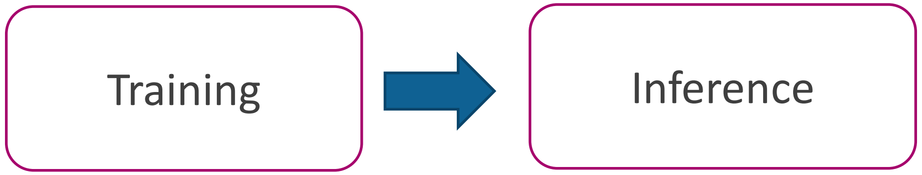

**Text to Speech**



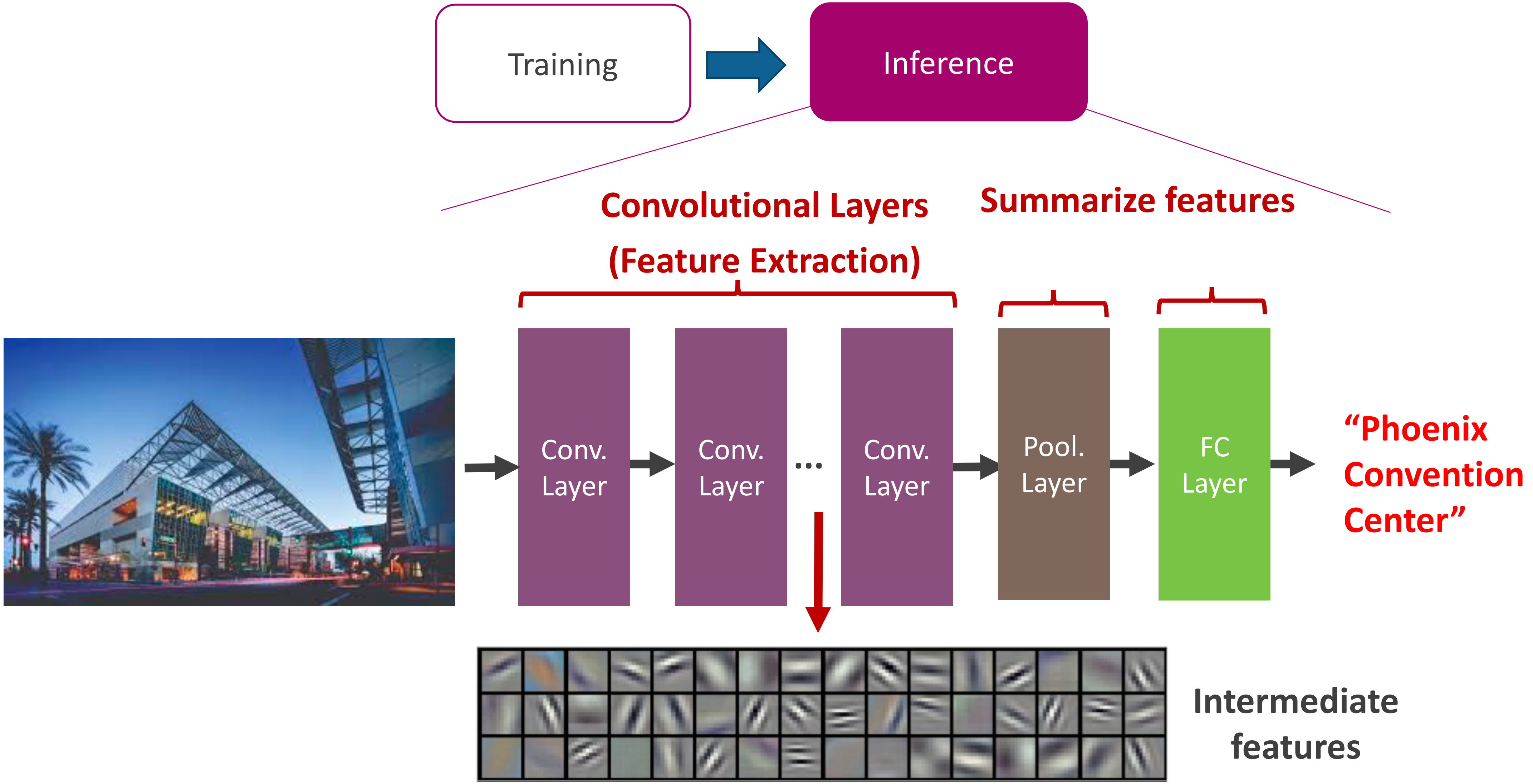

**Recommendations**

**Games**

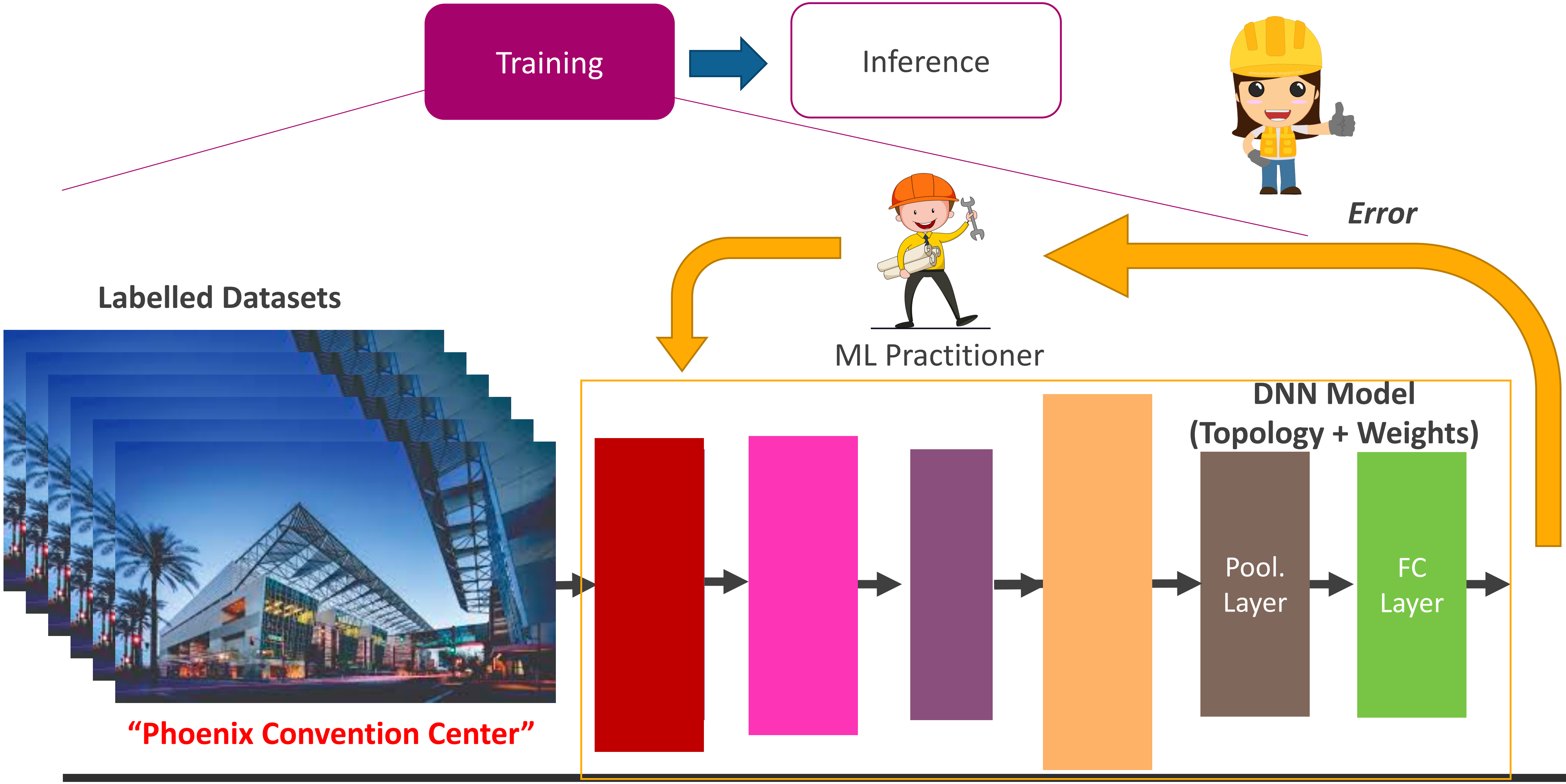
# Deep Learning Landscape



# Deep Learning Landscape



# Deep Learning Landscape



# Computation Platforms

Training → Inference

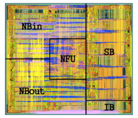
**HPC cluster**

ARM Trillum

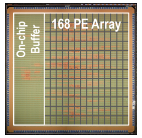
NVDLA

Apple Neural Engine

CambriconX



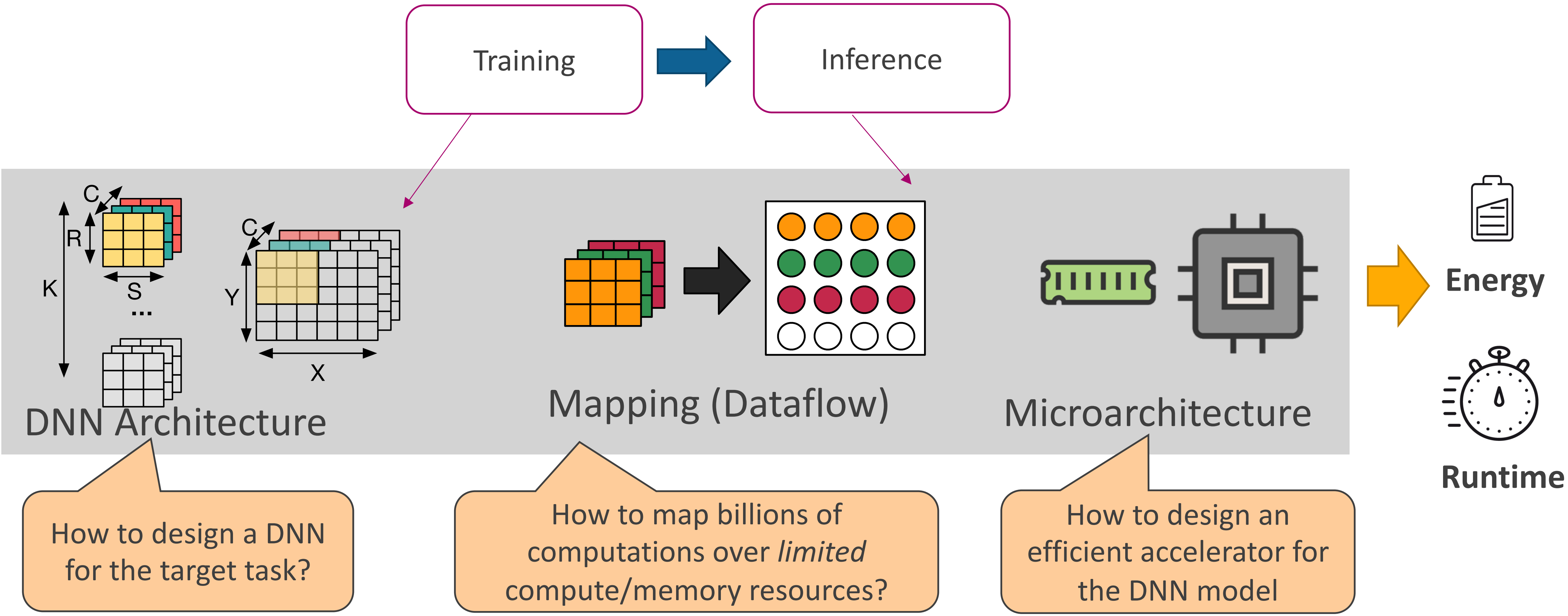

ShiDianNao




Eyeriss

**Accelerators**

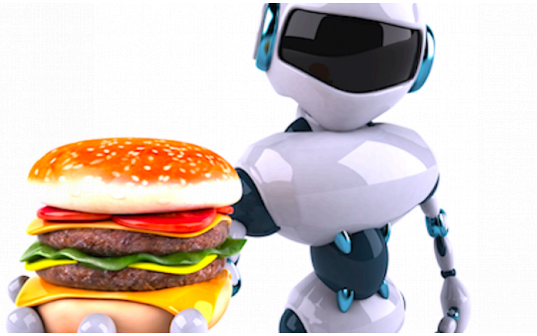
# Efficiency of Deep Learning Systems



# What is Continuous Learning?

Become better and faster with experience

Learn new tasks

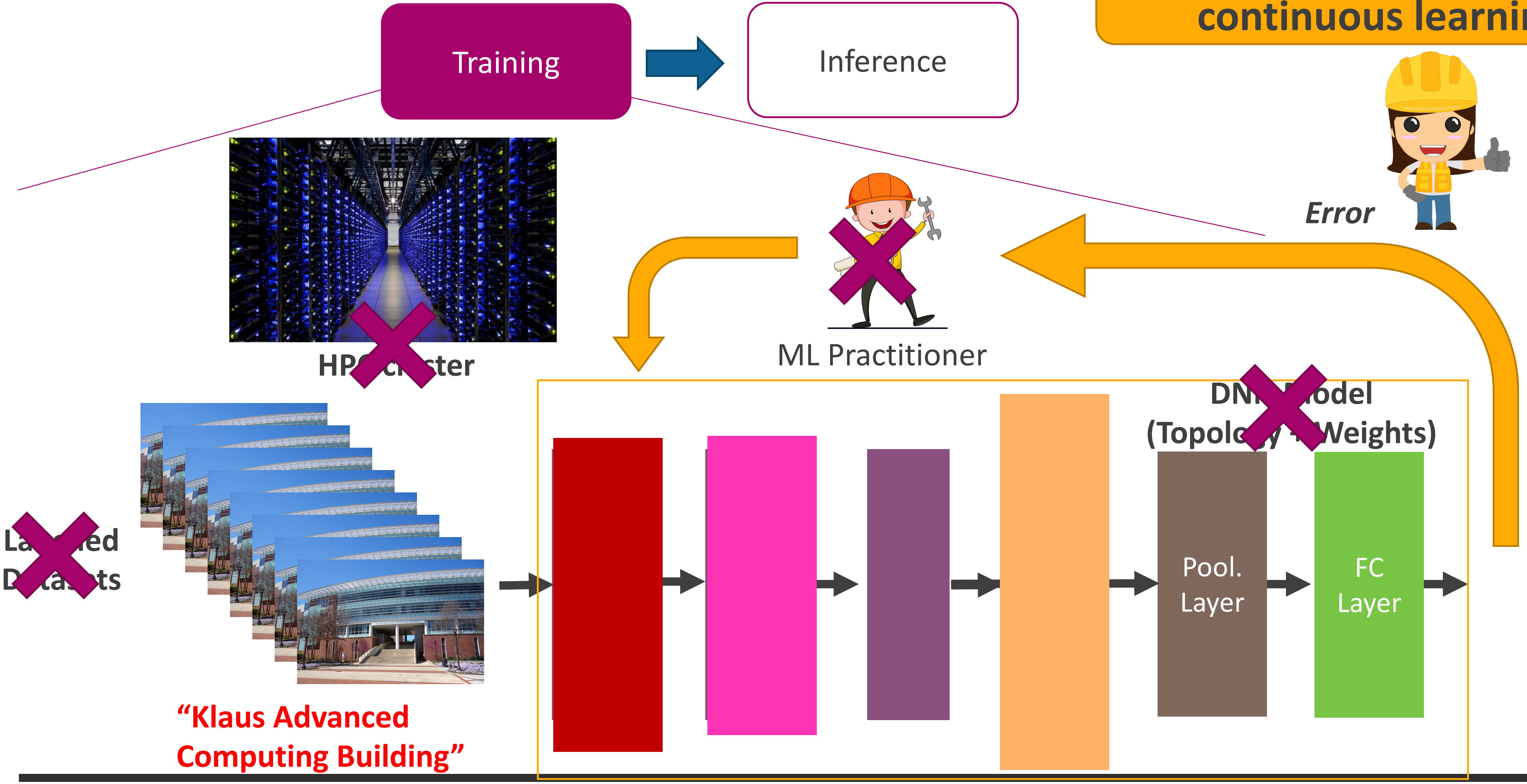


Compute and energy-efficiency

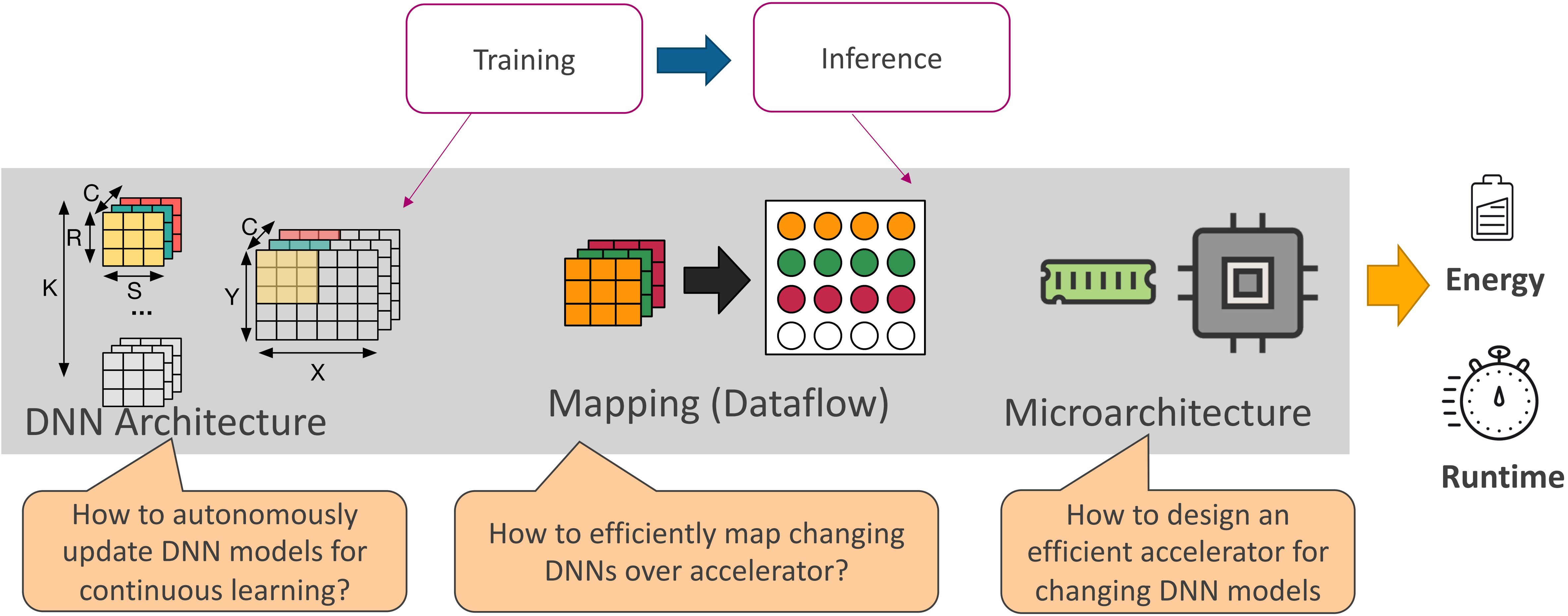
Can we leverage Supervised Deep Learning?

# Deep Learning Landscape

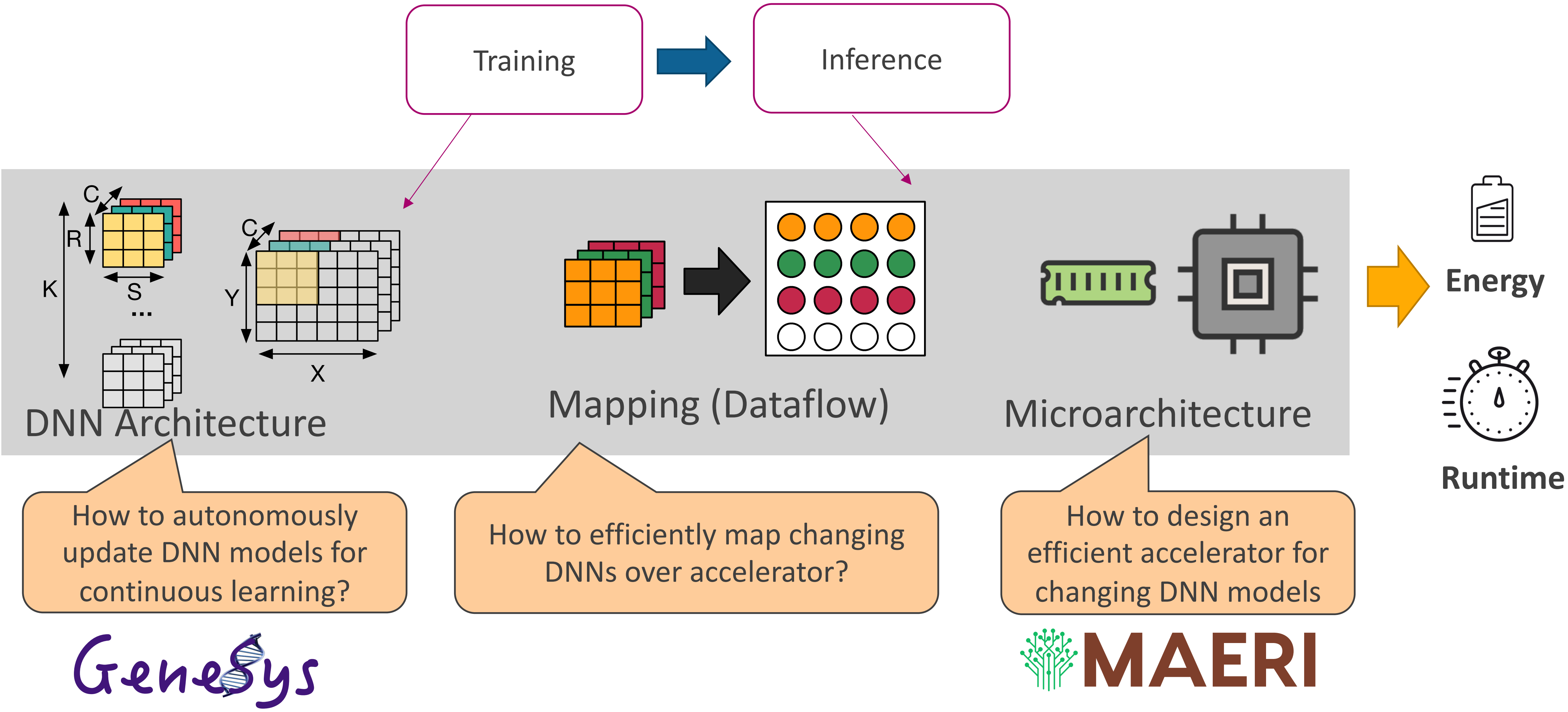
**Deep Learning not viable for continuous learning**



# Efficiency of Continuous Learning Systems

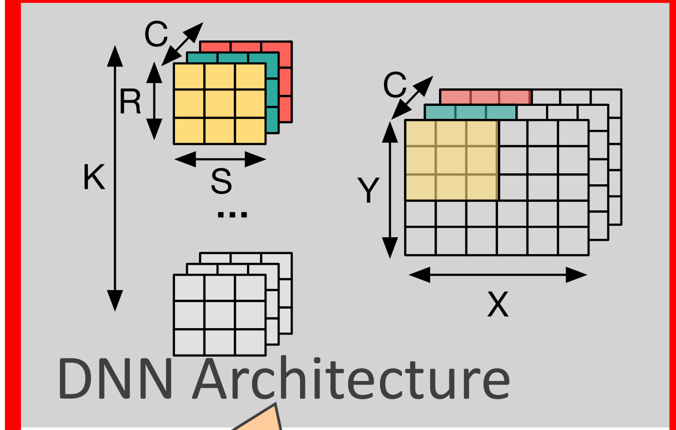


# Outline of Talk



# Outline of Talk

*Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna, **GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware,*** ***MICRO 2018***

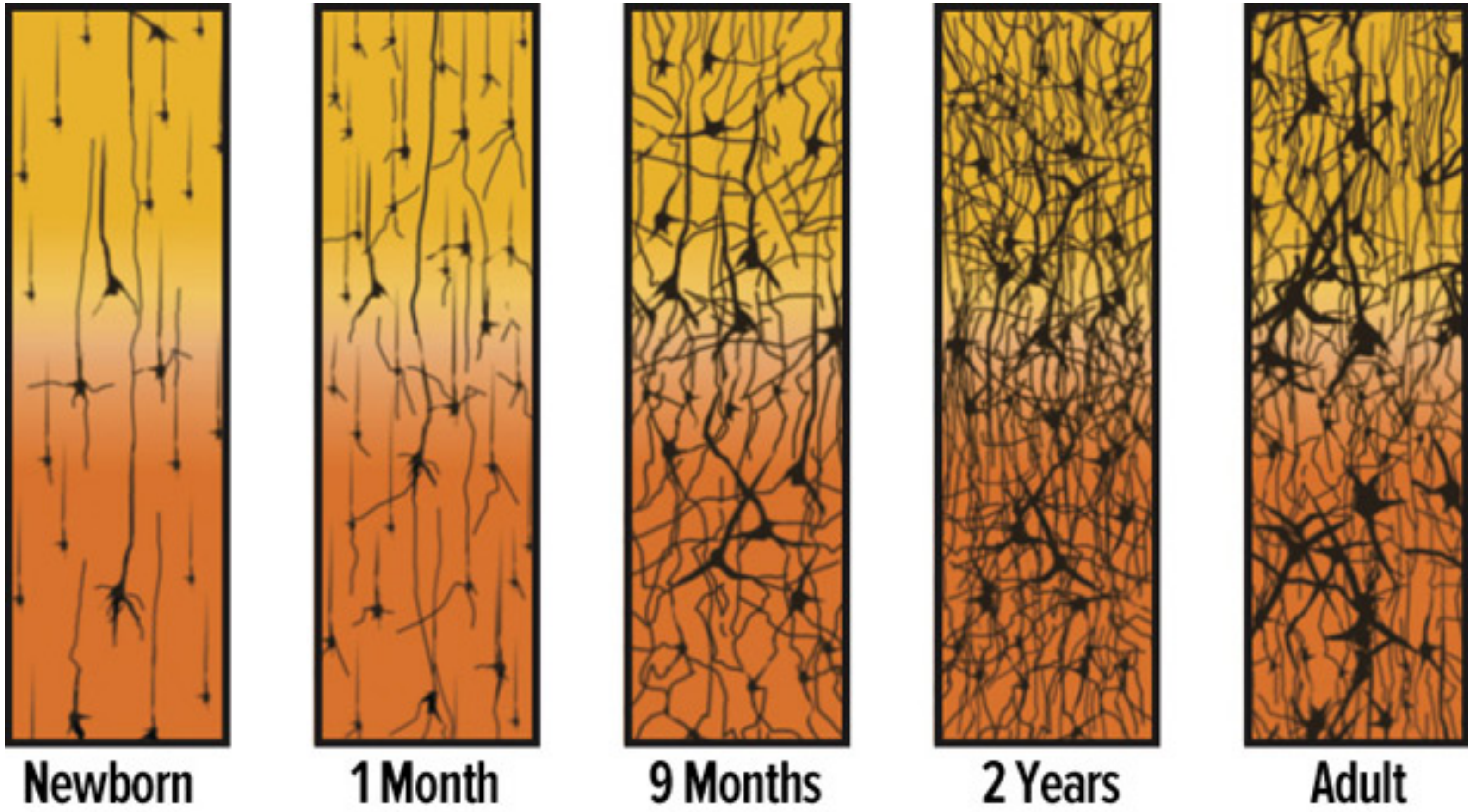



How to autonomously update DNN models for continuous learning?

GeneSys

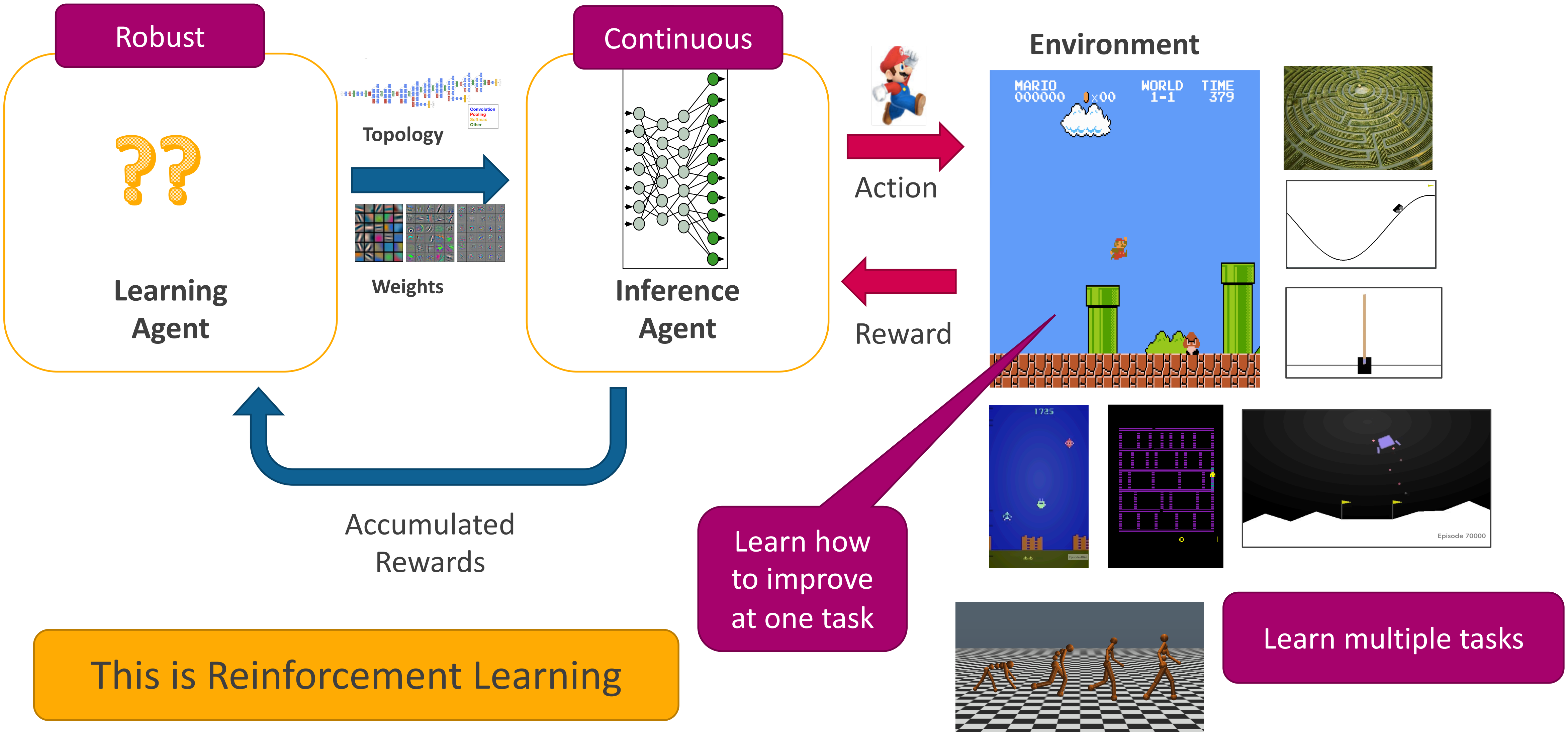
- Continuous Learning Template
- Neuro-Evolutionary Algorithms
  - Algorithm Description
  - Characterizing NEAT
- Microarchitecture
- Evaluations

# Continuous Learning in Brains




Constant synapse formation and pruning

# Template for Continuous Learning



# Conventional RL: Challenges

**Deep NNs** used internally

! Manual hyperparameter tuning

Each update results in **Backpropagation**

! High compute requirement at every update

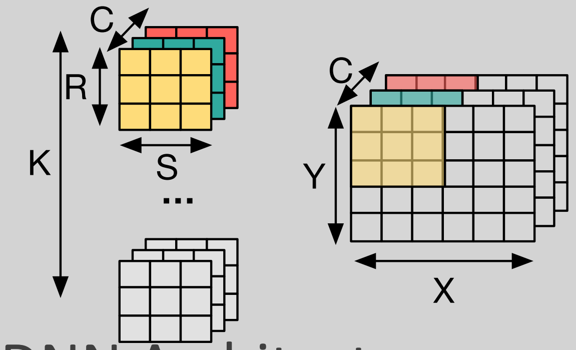
! High memory overhead

! Not scalable

**Not viable for continuous learning on the edge**

# Outline of Talk

*Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna,* ***GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware,*** ***MICRO 2018***



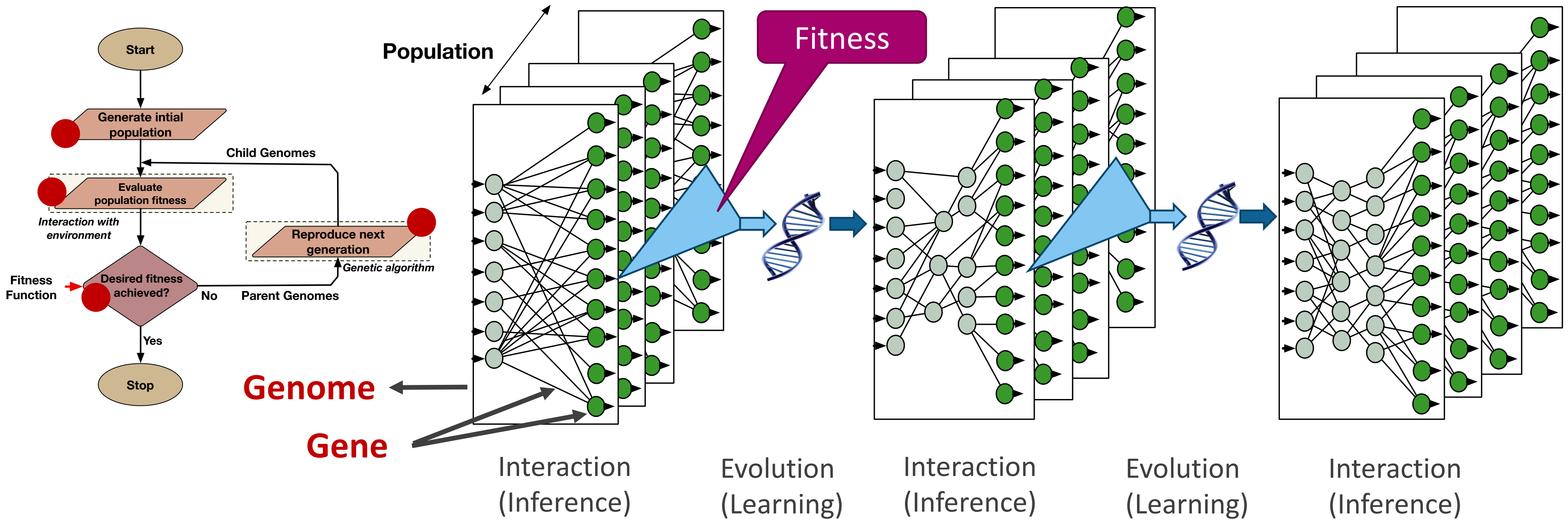

DNN Architecture

How to autonomously design DNN models for continuous learning?

GeneSys

- Continuous Learning` Template
- Neuro-Evolutionary Algorithms
  - Algorithm Description
  - Characterizing NEAT
- Microarchitecture
- Evaluations

# Neuro-Evolutionary (NE) Algorithm



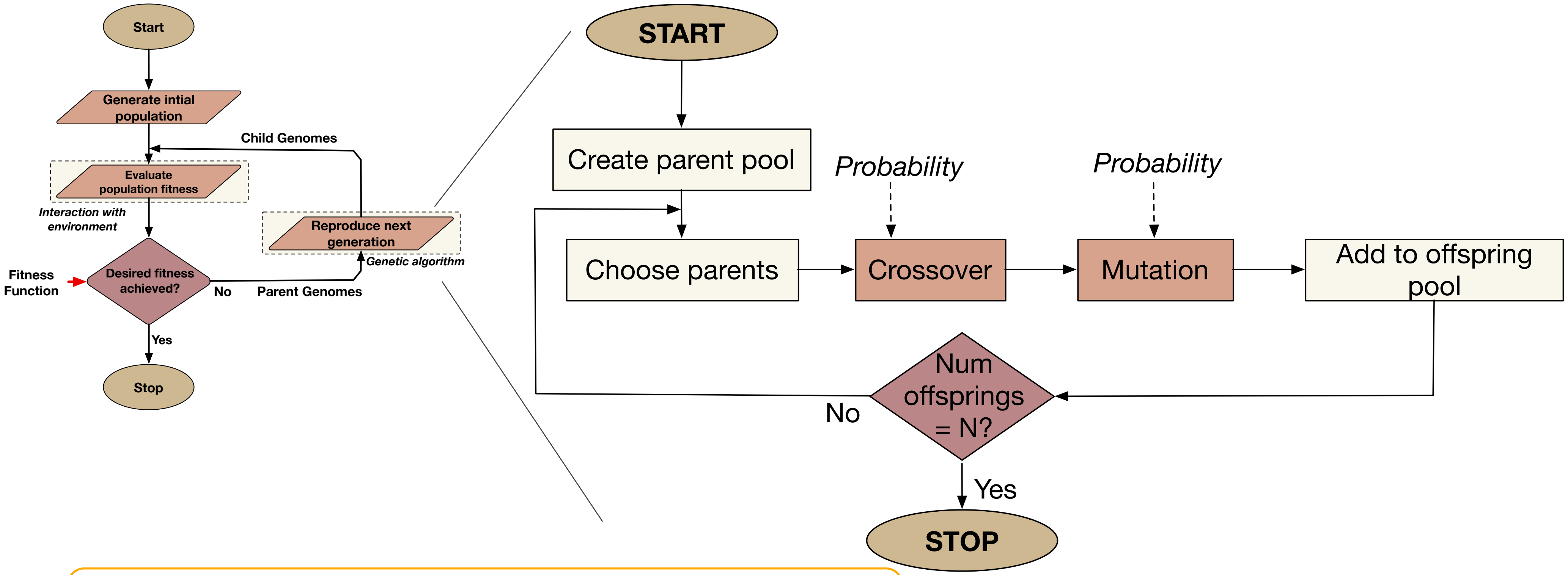

Neural Network (NN) expressed as a graph

**Gene**: Vertex or Edge in the graph

**Genome**: Collection of all genes (i.e., a NN)

[1] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, *10*(2), 99-127.

# Neuro-Evolutionary (NE) Algorithm




Neural Network (NN) expressed as a graph

**NeuroEvolution of Augmented Topologies (NEAT) [1]**

**Gene**: Vertex or Edge in the graph

**Genome**: Collection of all genes (i.e., a NN)

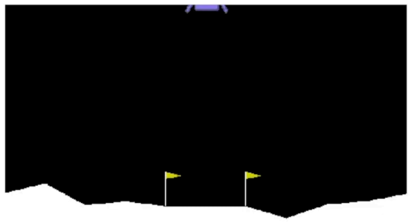
[1] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, *10*(2), 99-127.

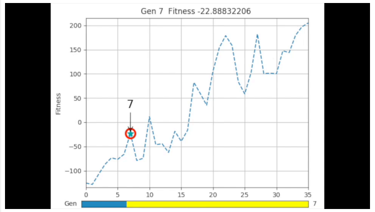
# Properties of NE algorithms

## Algorithmic

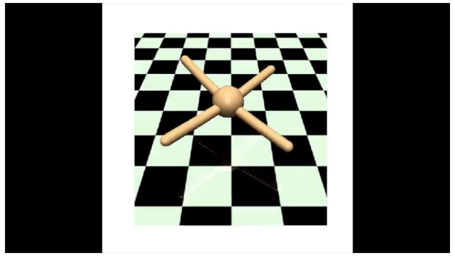
Robustness

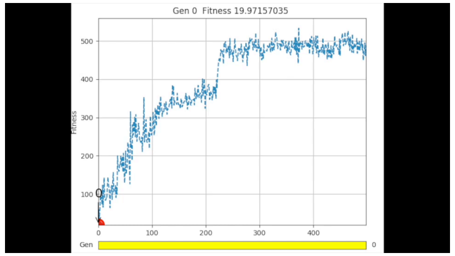
No Training






Change fitness function





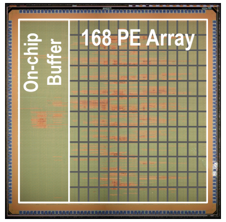

Accuracy?

## Systems

Too much compute!

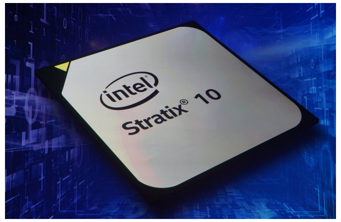
Convergence time?

déjà vu! Looks like Deep Neural Networks in the 90s




Eyeriss

GPU



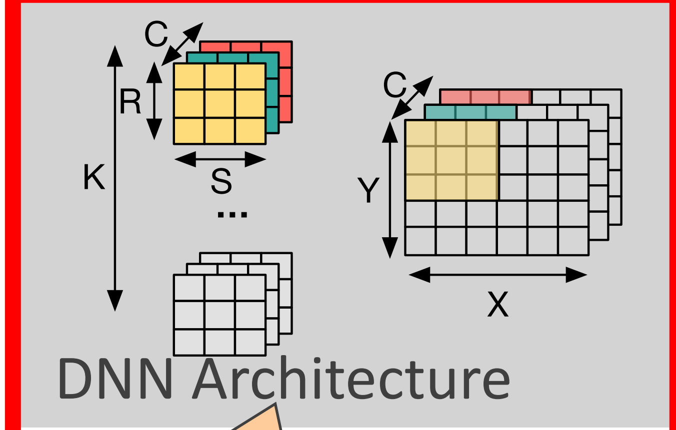

FPGA

**HW solutions enabled Deep Learning**

***Can we do the same with EA?***

# Outline of Talk

*Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna, **GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware**, **MICRO 2018***




How to autonomously design DNN models for continuous learning?

GeneSys

- Continuous Learning
- Neuro-Evolutionary Algorithms
  - Algorithm Description
  - Characterizing NEAT
- Microarchitecture
- Evaluations

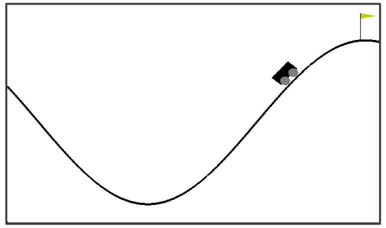
# Characterization of NEAT

**NEAT - Python**

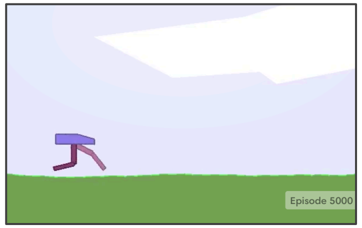
Codebase

OpenAI Gym

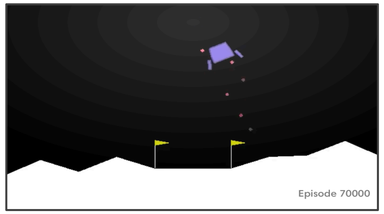
Environments



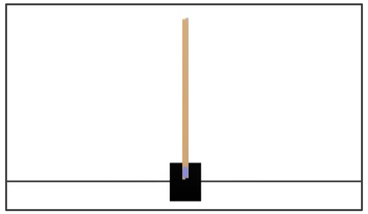
Mountain car




Bipedal

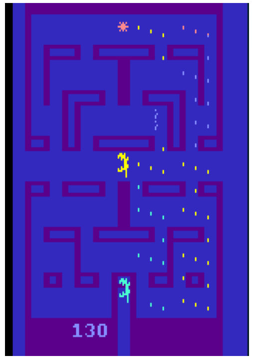



Lunar Lander

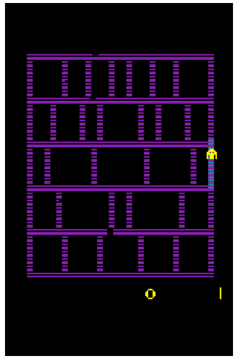


Cart pole


Airraid-RAM




Alien-RAM



Amidar-RAM

Ran each environment till convergence, multiple times

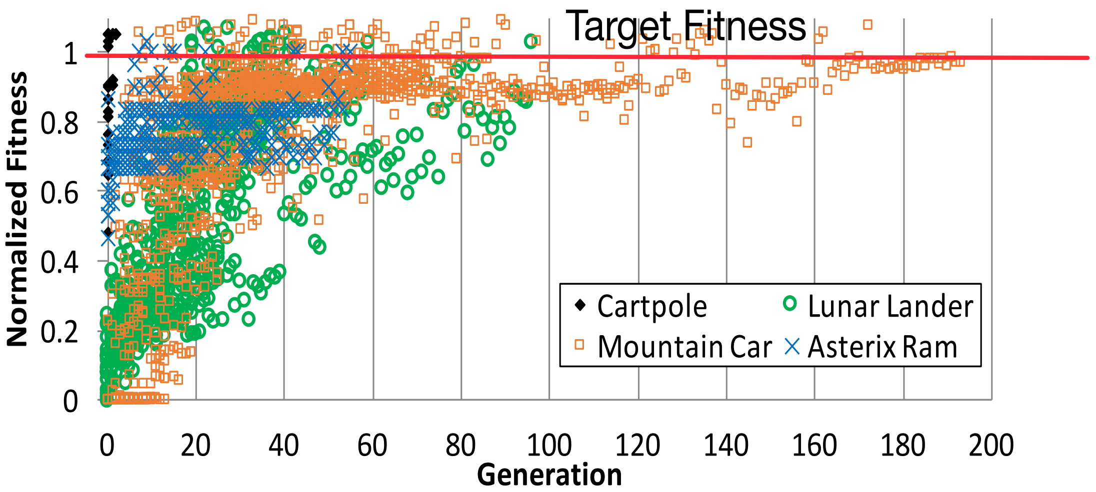
Only changed fitness function between workloads



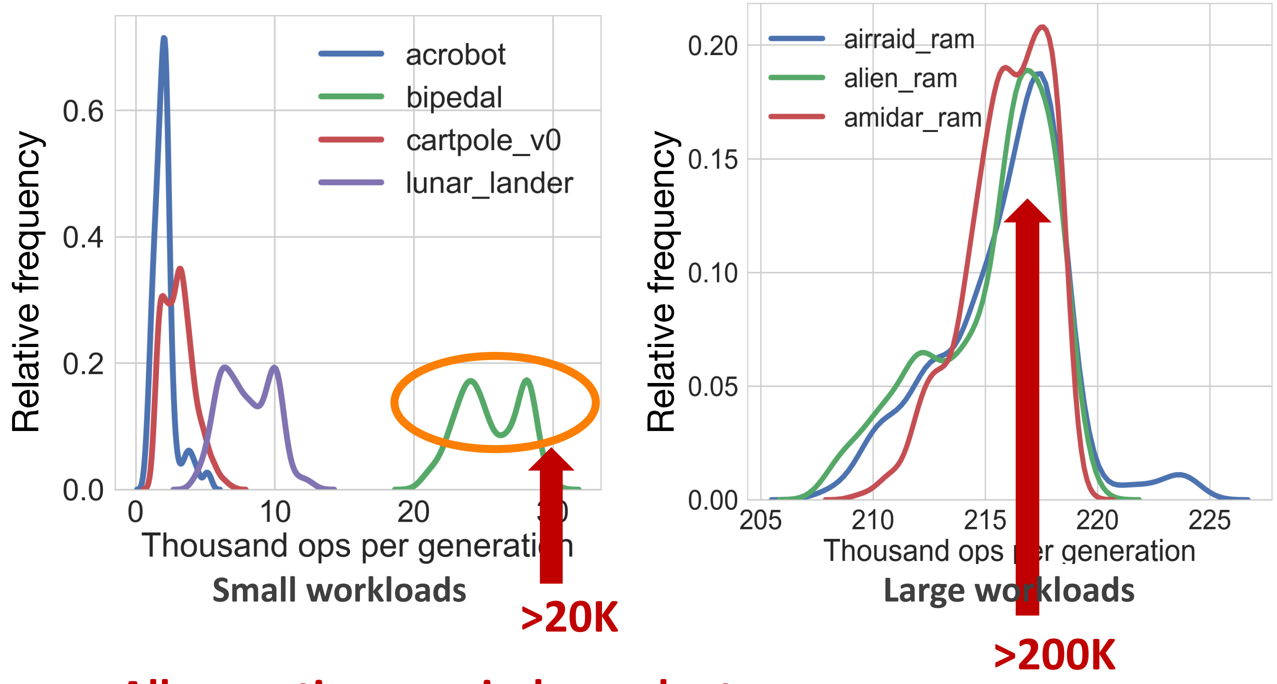
NEAT Python: https://github.com/CodeReclaimers/neat-python

# Characterization of NEAT

## Computations




### Distribution of Operations/Generation




**Inference:**

**Population level parallelism (PLP)**

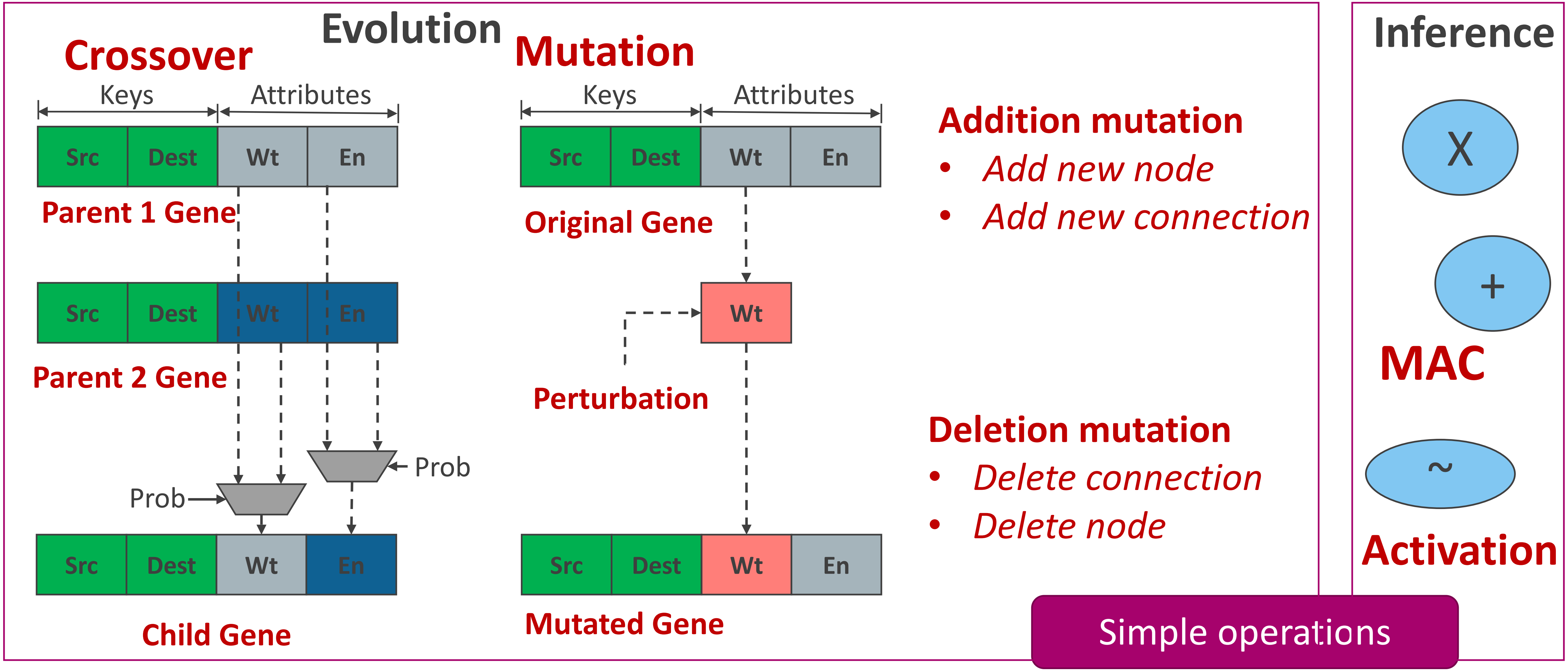
**All operations are independent**

**Large operation level Parallelism**

**Evolution:**

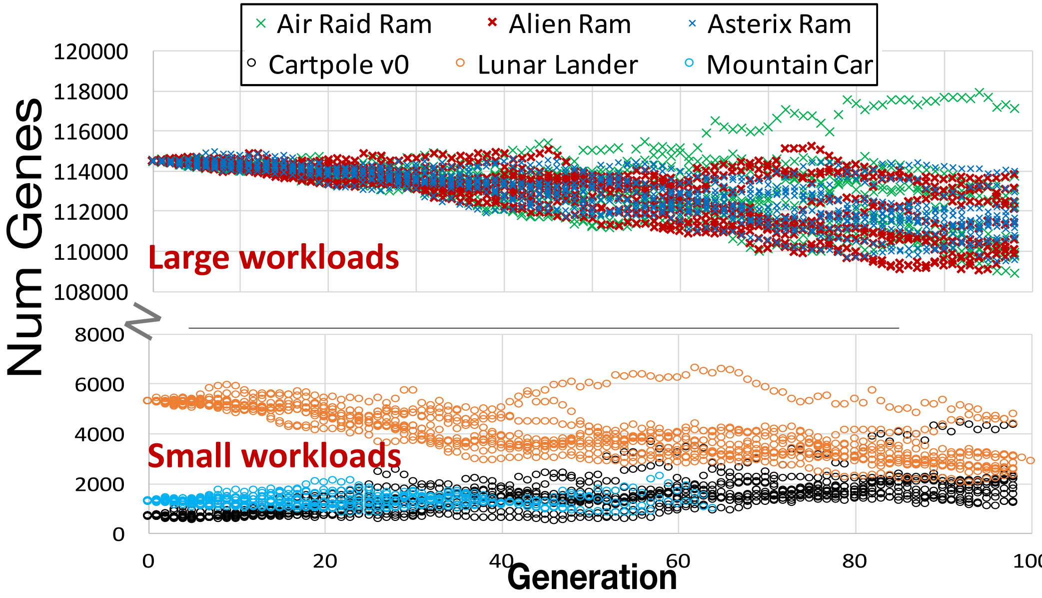
**Gene level parallelism (GLP)**

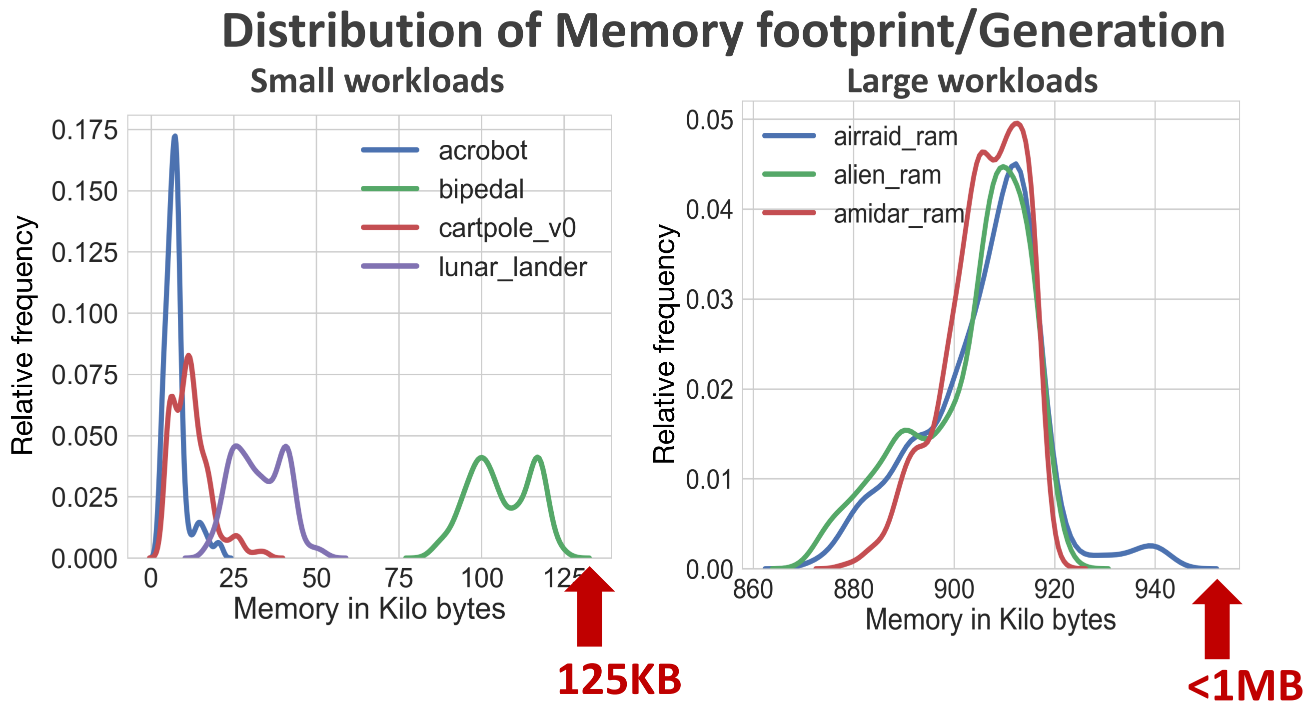
# Operations in NEAT



# Characterization of NEAT

## Memory



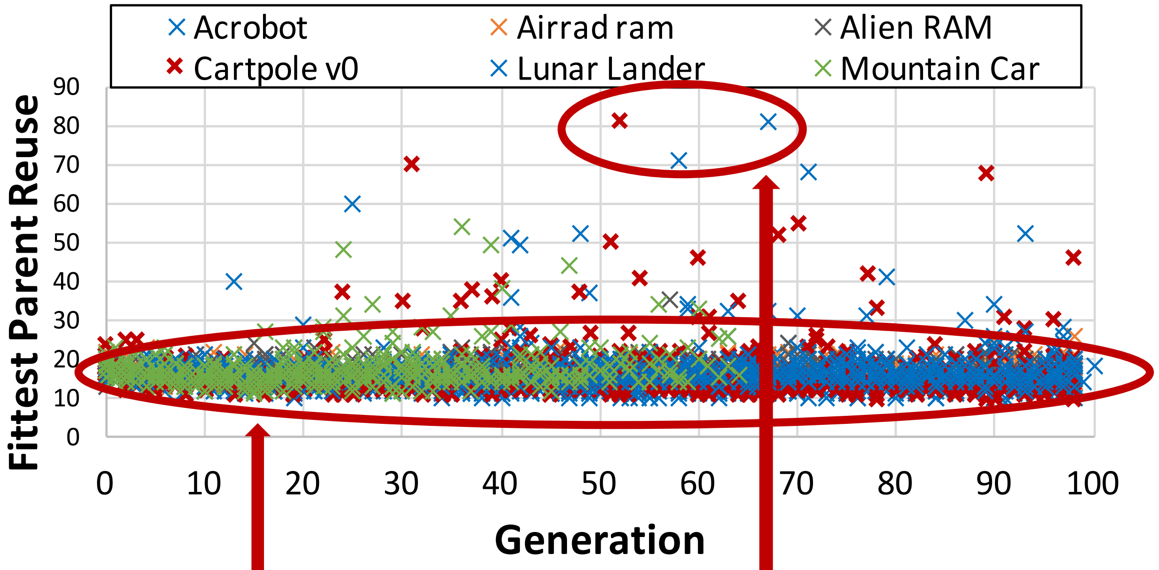

**Entire population can fit on-chip**

**Only need to store the weights and node info**

# Characterization of NEAT

## Memory

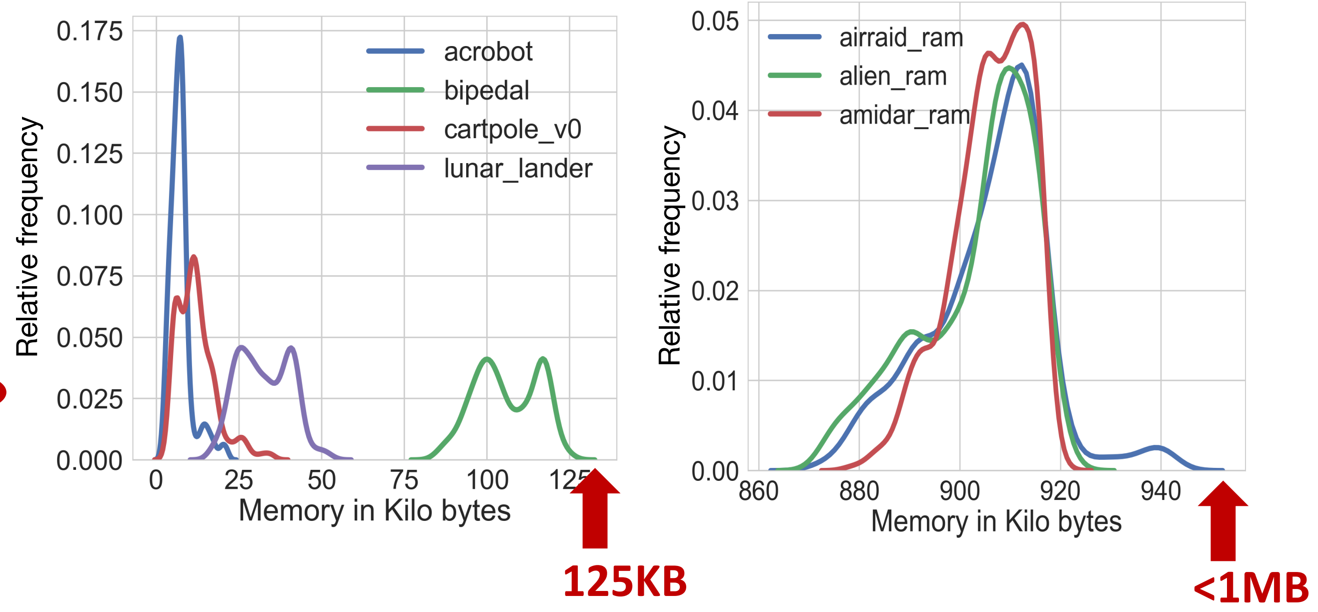
### Opportunity for Reuse




Fittest parent genome is used about ~10-20 times each generation

Even higher in certain cases

### Distribution of Memory footprint/Generation




**Entire population can fit on-chip**

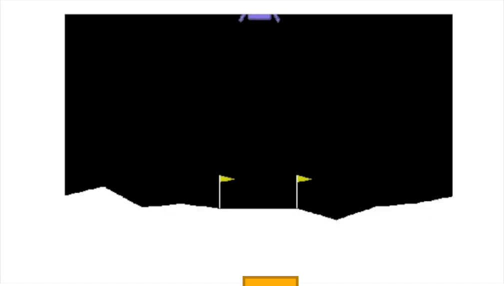
**Only need to store the weights and node info**

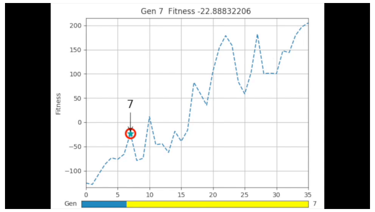
# Properties of NE algorithms

## Algorithmic

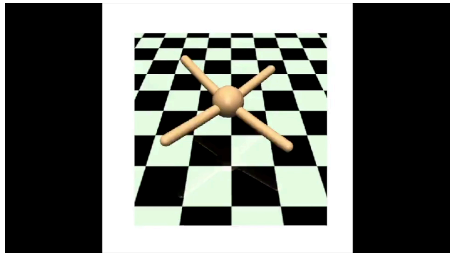
Robustness

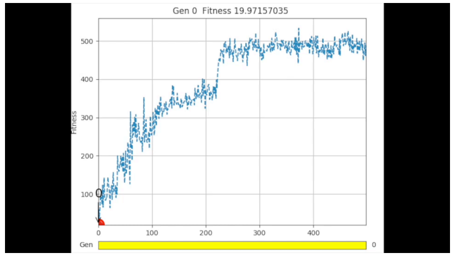
No Training






Change fitness function






## Systems

Massive Parallelism

Genomes within Population

Genes within a Genome

Low Memory Footprint

Only store genomes in current generation

No backprop

No gradiant calculations or storage

Simple HW-friendly Ops

MACs in Inference
Crossover and Mutation in Evolution

**HW-SW Co-Design of NE makes them viable for continuous learning on edge**

# Motivating Hardware Solution

**Massive Parallelism** → **Scalability** → **Faster convergence**

Gene and Population level parallelism

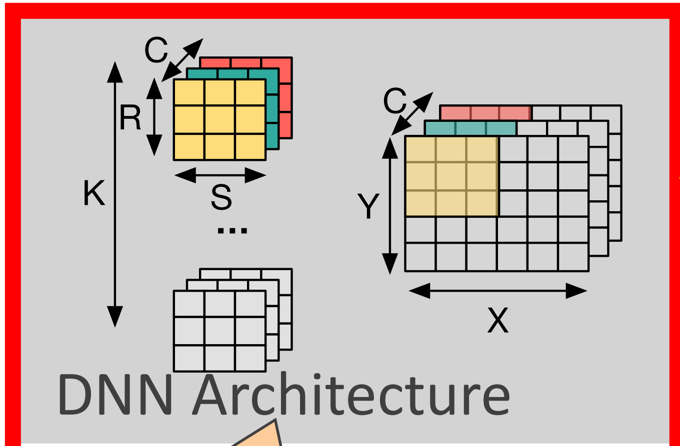
**Power efficiency** → **More deployable compute** → **Target complex problems**

Simple HW friendly operations

**Hardware-Software codesign of NE makes them viable for continuous learning**

# Outline of Talk

*Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna, **GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware**, **MICRO 2018***

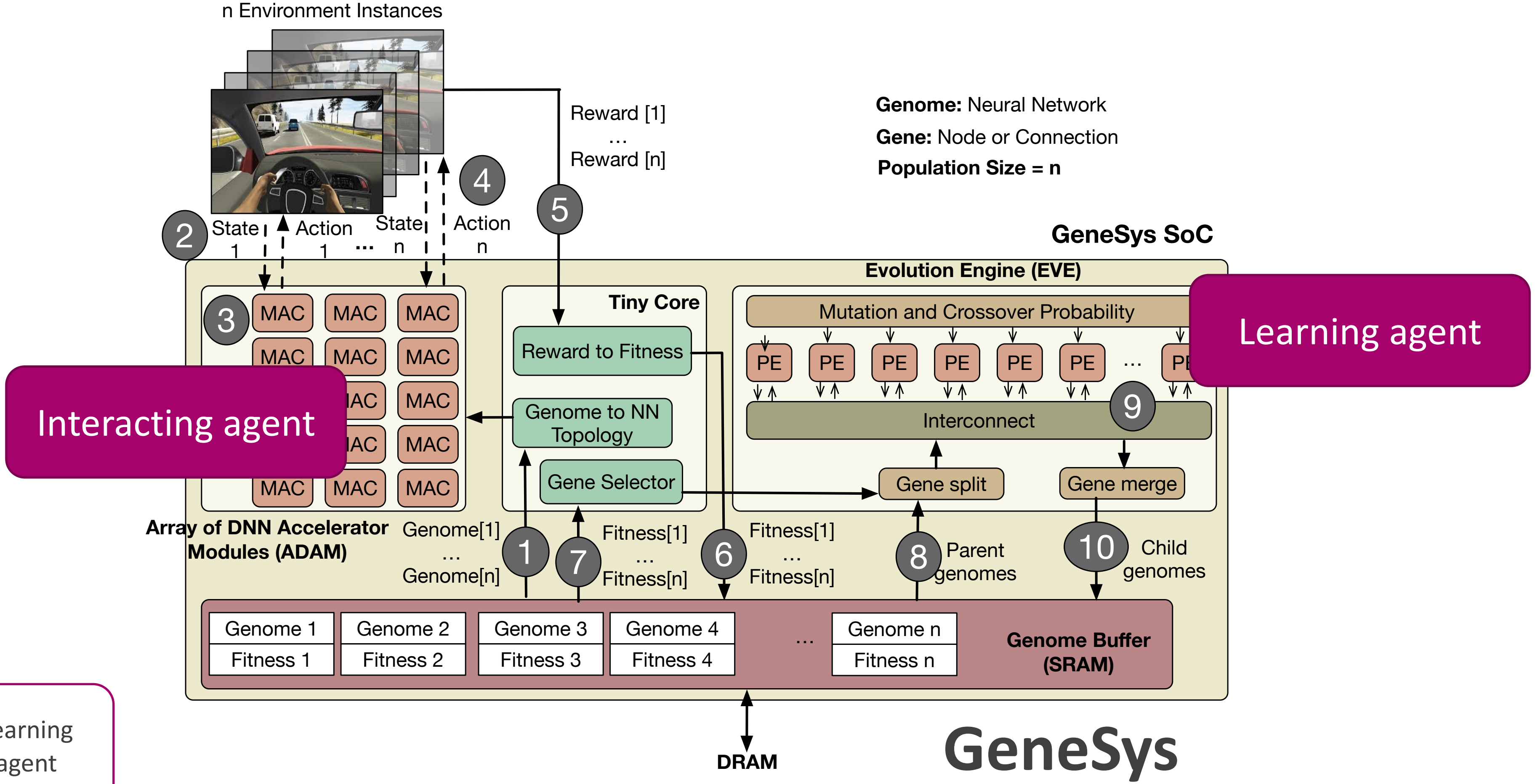


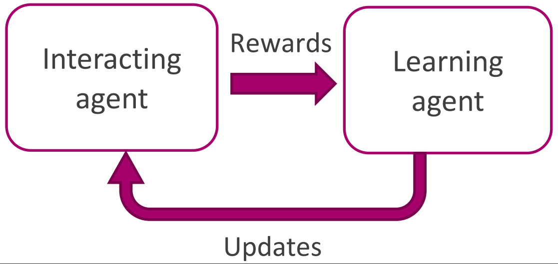

How to autonomously update DNN models for continuous learning?

GeneSys

- Continuous Learning
- Neuro-Evolutionary Algorithms
  - Algorithm Description
  - Characterizing NEAT
- Microarchitecture
- Evaluations

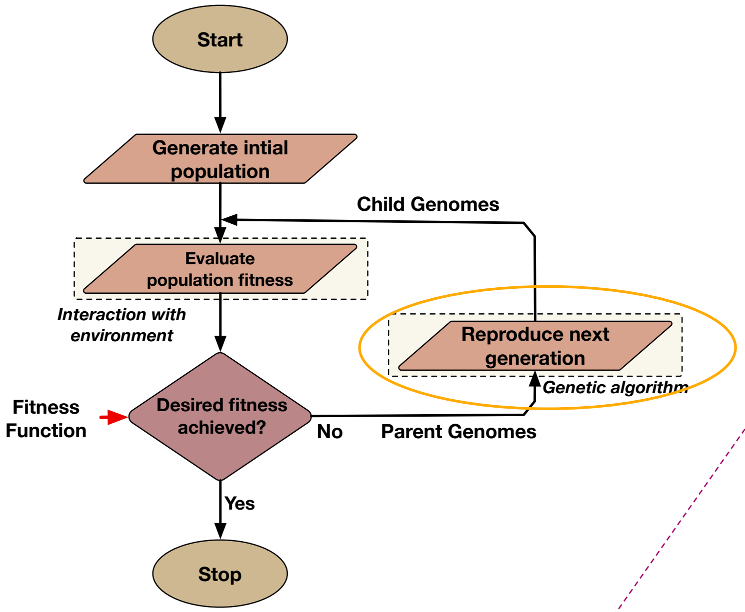
# GeneSys SoC

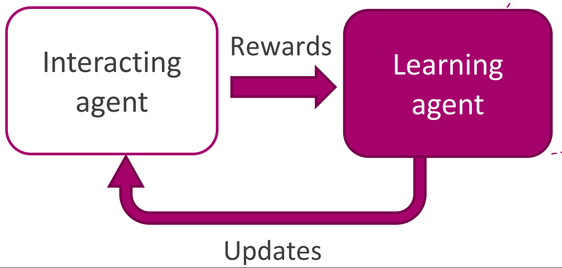


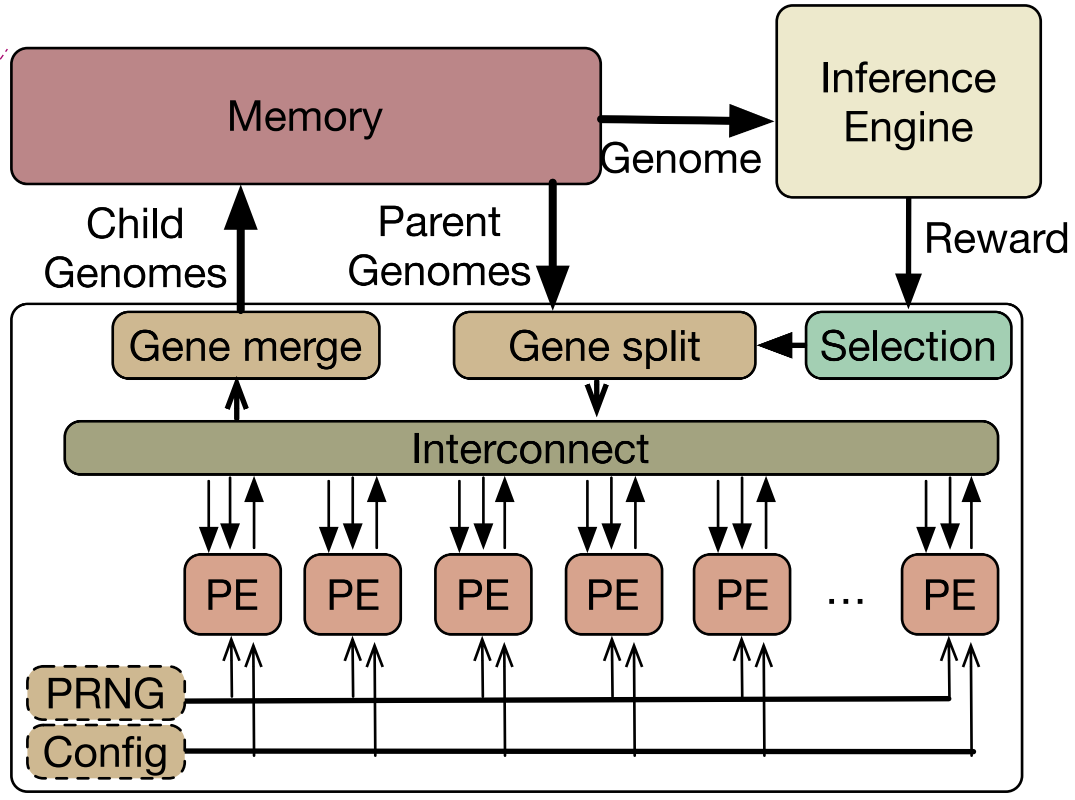

*Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna,* ***GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware****, MICRO 2018*

# Evolution Engine: EvE Microarchitecture



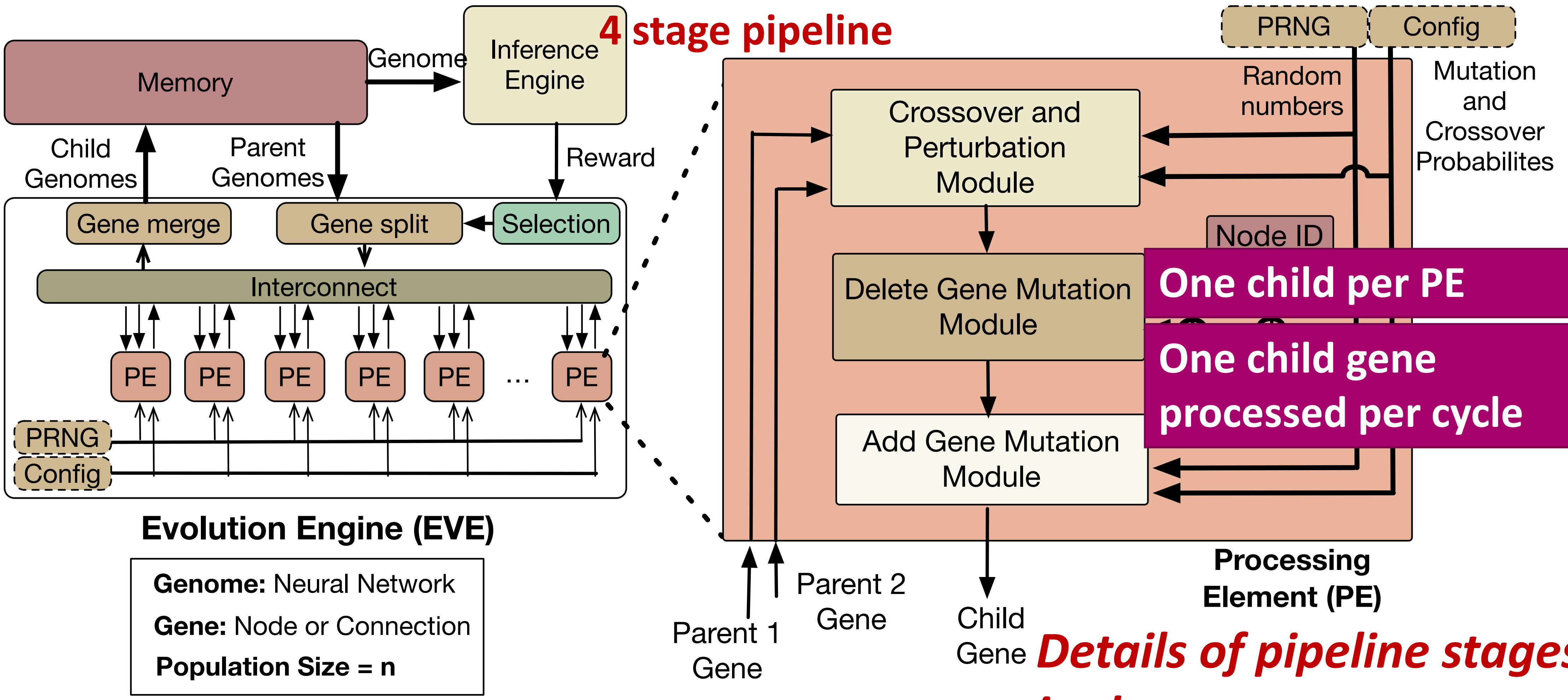

Large number of PE to exploit parallelism




**Evolution Engine (EVE)**

# PE Microarchitecture

**4 stage pipeline**

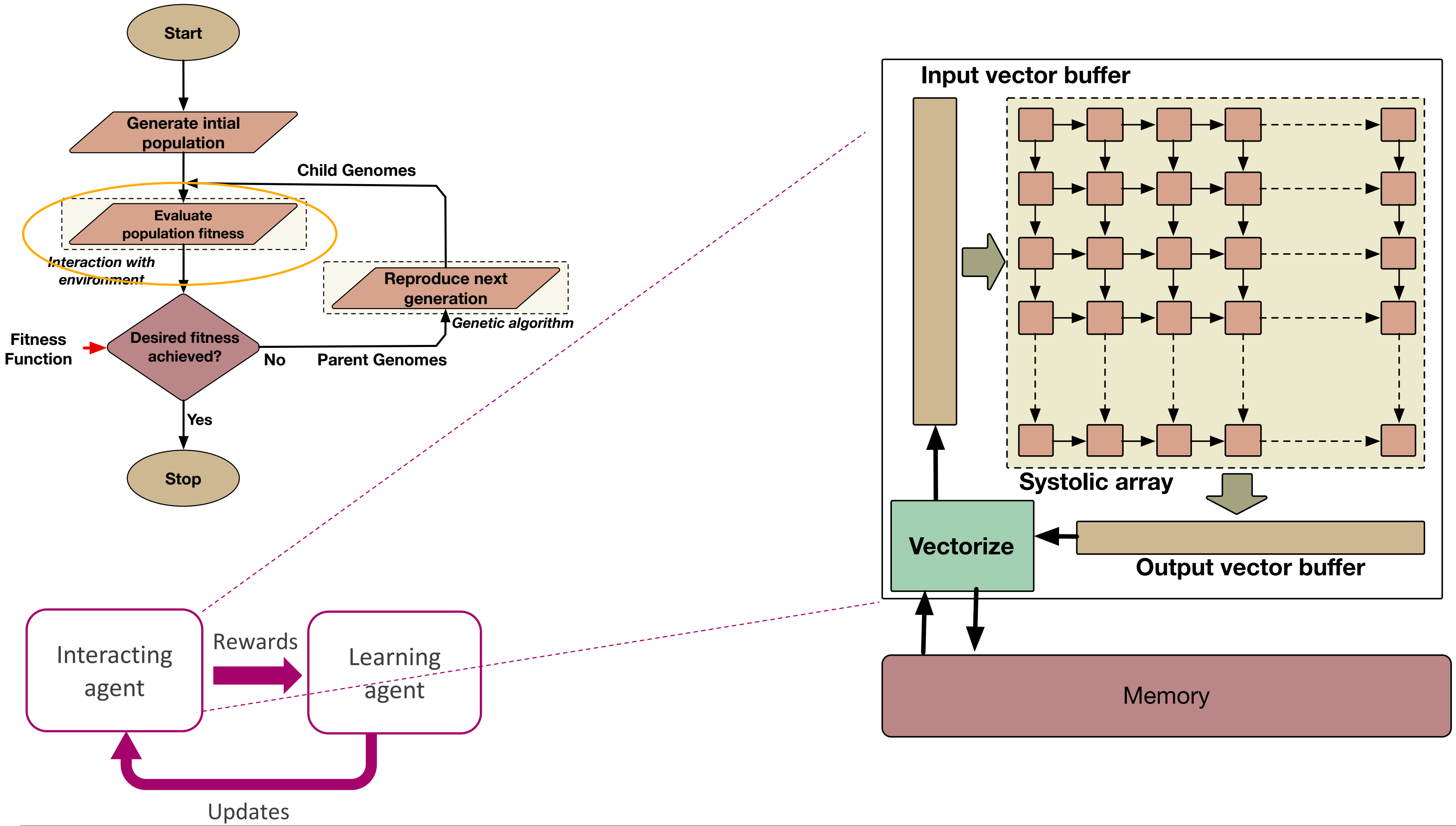



**One child per PE**

**One child gene processed per cycle**

***Details of pipeline stages in the paper***

# Inference Engine: ADAM Microarchitecture




Conventional DNN Inference Accelerator

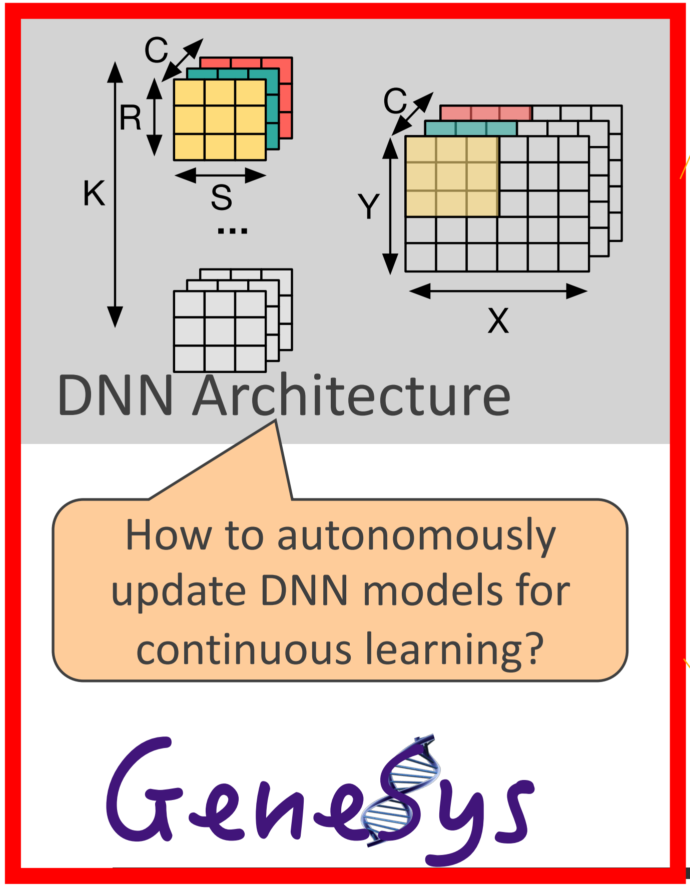
**Exploit Population Level Parallelism**

Networks generated by NEAT are irregular (thus sparse)

Details later in talk!

# Outline of Talk

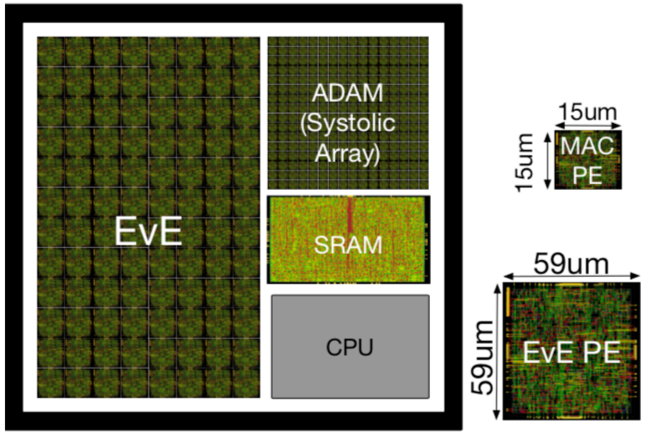
*Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna, **GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware**, **MICRO 2018***

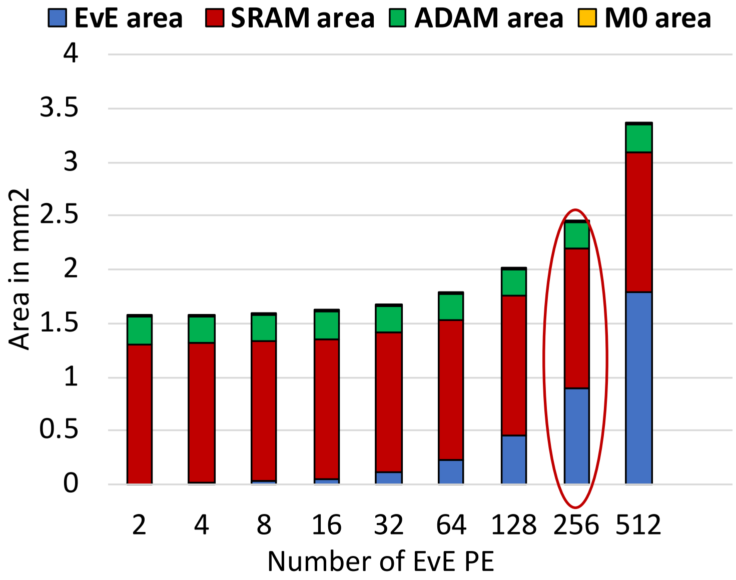



- Continuous Learning
- Neuro-Evolutionary Algorithms
  - Algorithm Description
  - Characterizing NEAT
- Microarchitecture
- Evaluations

# Implementation

| GeneSys Parameters | |
|---|---|
| Tech node | 15nm |
| Num EvE PE | 256 |
| Num ADAM PE | 1024 |
| EvE Area | 0.89 mm2 |
| ADAM Area | 0.25 mm2 |
| GeneSys Area | 2.45 mm2 |
| Power | 947.5 mW |
| Frequency | 200 MHz |
| Voltage | 1.0 V |
| SRAM banks | 48 |
| SRAM depth | 4096 |



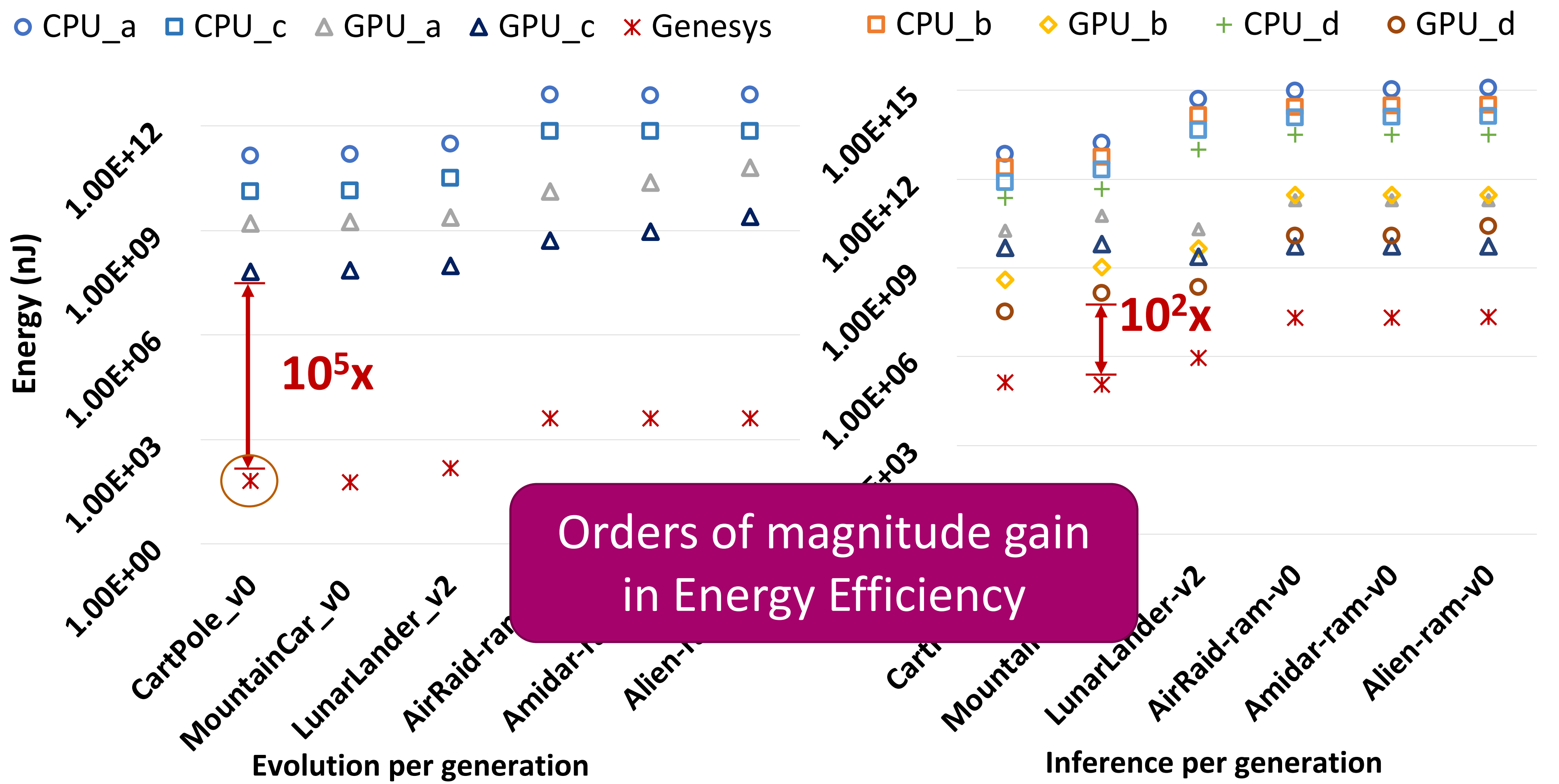



# Evaluations

| Legend | Inference | Evolution | Platform |
|---|---|---|---|
| CPU_a | Serial | Serial | 6th gen i7 |
| CPU_b | PLP | Serial | 6th gen i7 |
| GPU_a | BSP | PLP | Nvidia GTX 1080 |
| GPU_b | BSP + PLP | PLP | Nvidia GTX 1080 |
| CPU_c | Serial | Serial | ARM Cortex A57 |
| CPU_d | PLP | Serial | ARM Cortex A57 |
| GPU_c | BSP | PLP | Nvidia Tegra |
| GPU_d | BSP + PLP | PLP | Nvidia Tegra |
| GENESYS | PLP | PLP + GLP | GENESYS |

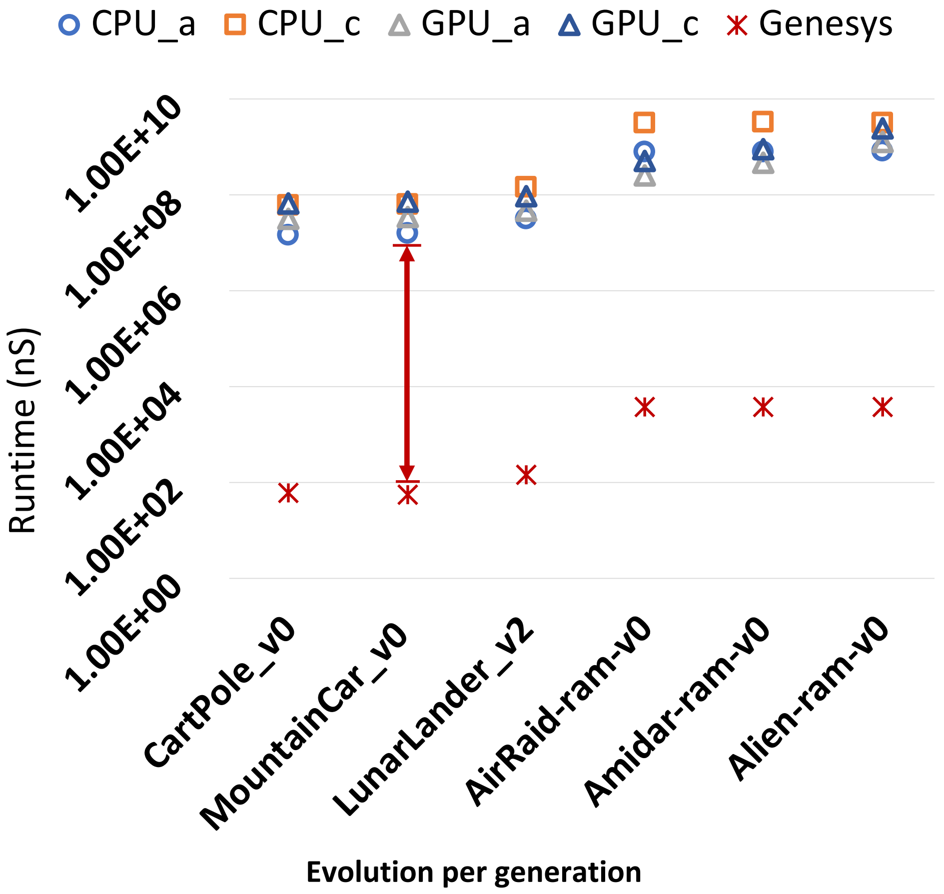
PLP (GLP) - Population (Gene) Level Parallelism
BSP - Bulk Synchronous Parallelism (GPU)

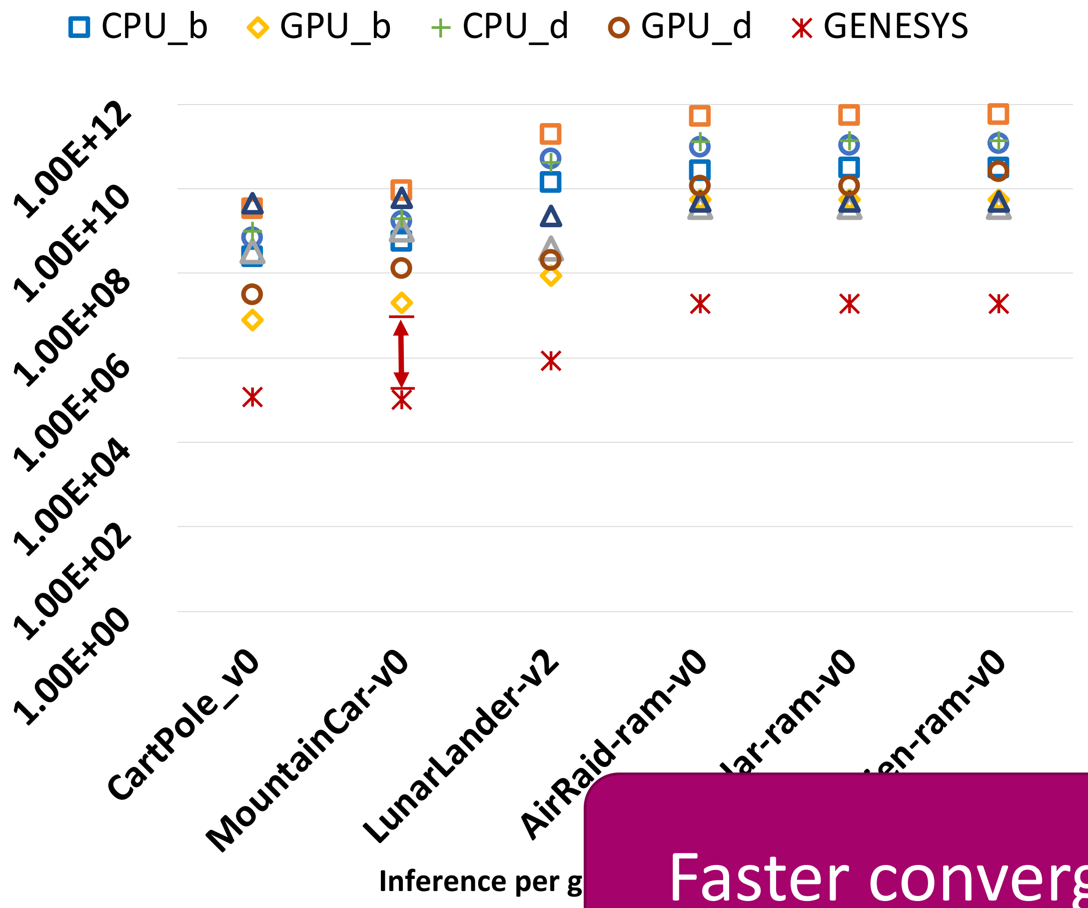
# Evaluations: Energy




Orders of magnitude gain in Energy Efficiency

# Evaluations: Runtime



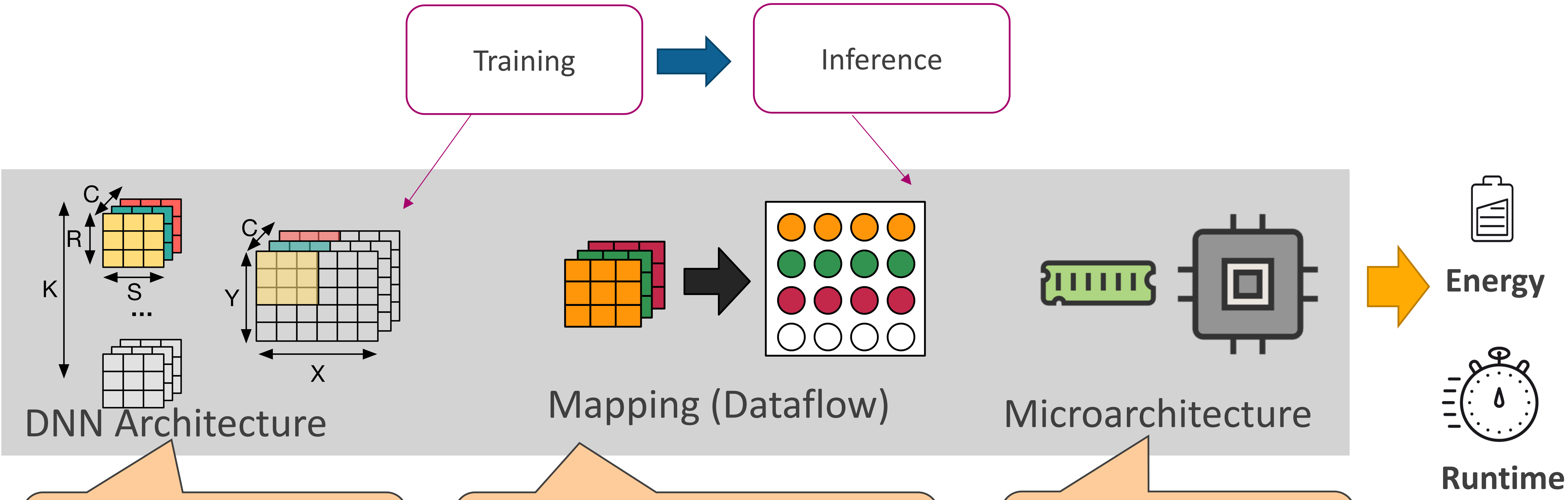




Faster convergence

# Summary for GeneSys

- **Robust**, **Scalable** and **Energy** efficient  solutions needed for continuous learning
  - Look **beyond DL and RL**
- NEs offer promise
  - Parallelism
  - Low-memory Footprint
  - HW friendly
- GeneSys: *100x – 100000x energy efficiency and performance*
  - **More deployable compute**
  - **Enables** AI solutions for a large gamut of problems

# Outline of Talk




How to autonomously design DNN models for continuous learning?

How to efficiently map changing DNNs over accelerator?

How to design an efficient accelerator for changing DNN models

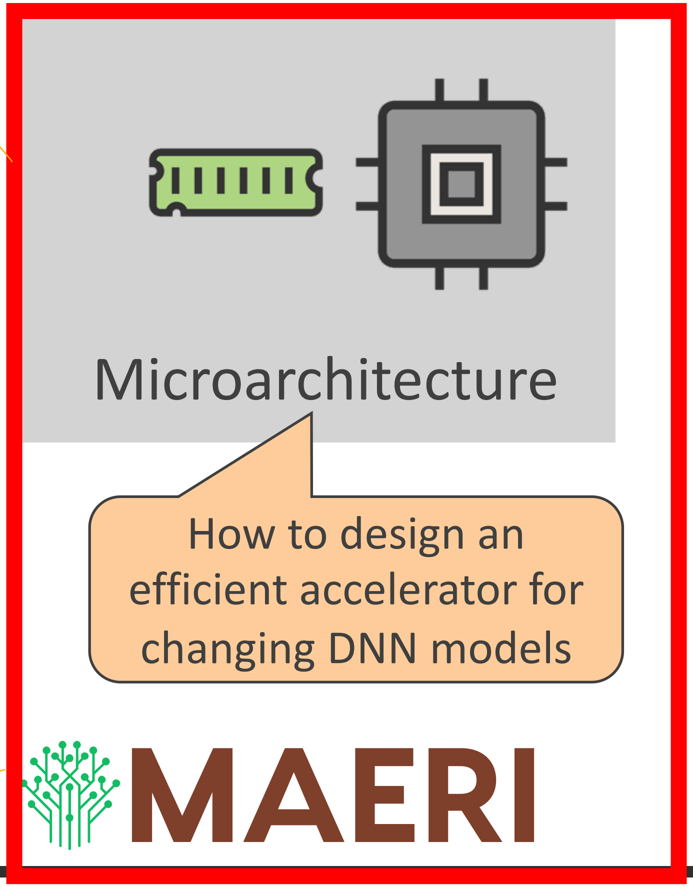
GeneSys

MAERI

# Outline of Talk

- Motivation
  - Irregular Dataflows
  - DNN Computation
- MAERI
  - Abstraction
  - Implementation
  - Operation Example
  - Mapping Strategies
- Evaluations

Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna
**MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects:**
**ASPLOS 2018**, ***IEEE Micro Top Picks 2019 Honorable Mention***



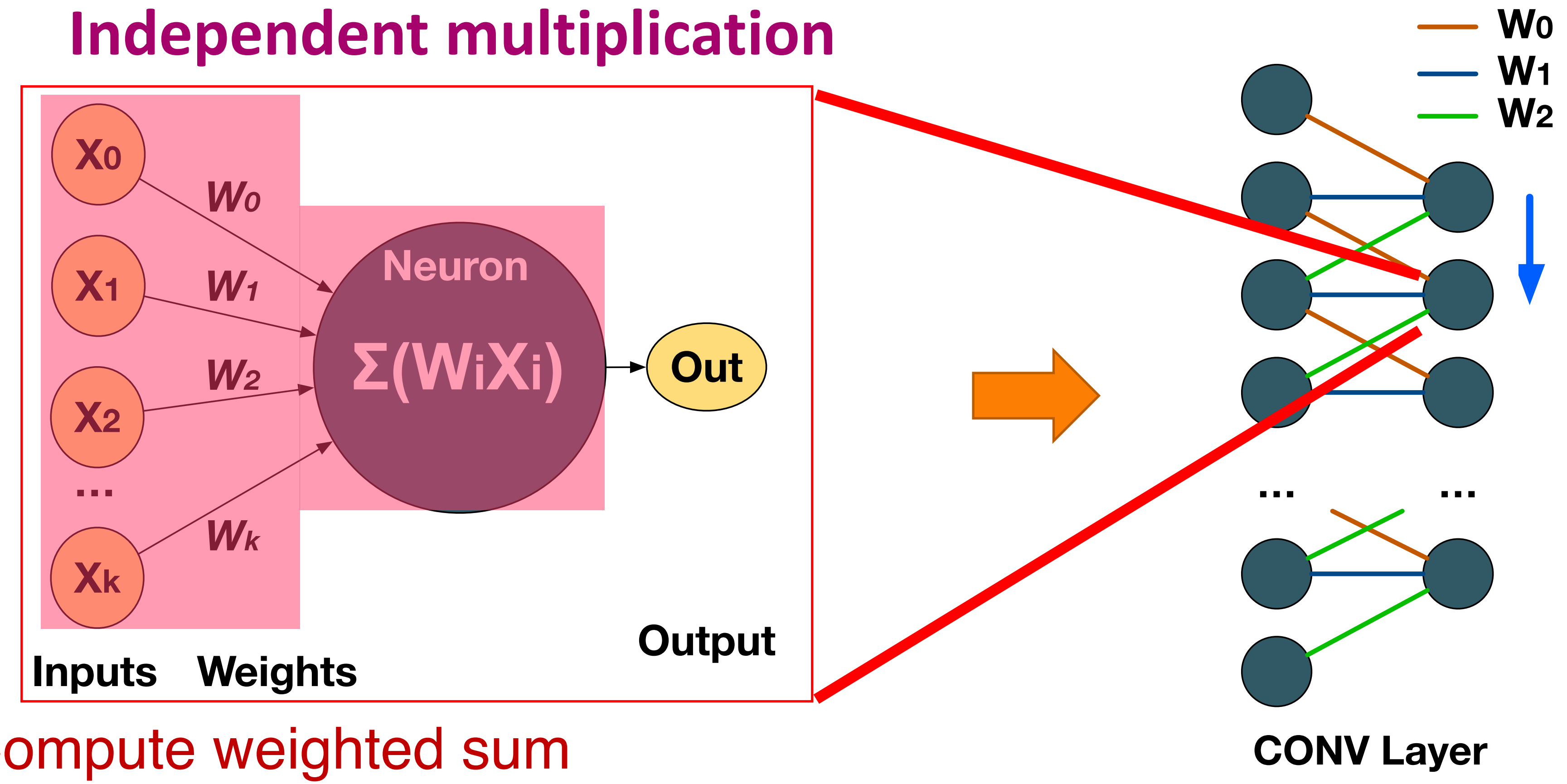
# Myriad Dataflows in DNN Accelerators

- **DNN Topologies**
  - Layer size / shape
  - Layer types: Convolution / Pool / FC / LSTM
  - New sub-structure: e.g., Inception in Googlenet
- **Compiler/Mapper**
  - Loop Scheduling
    - Reordering and Tiling
  - Mapping
    - Output/Weight/Input/Row-stationary
- **Algorithmic Optimization (e.g., Sparsity)**
  - Weight pruning
  - GeneSys

Can we have one architectural solution that can handle arbitrary dataflows and provides ~100% utilization?

# What is the computation in a DNN?

## Independent multiplication



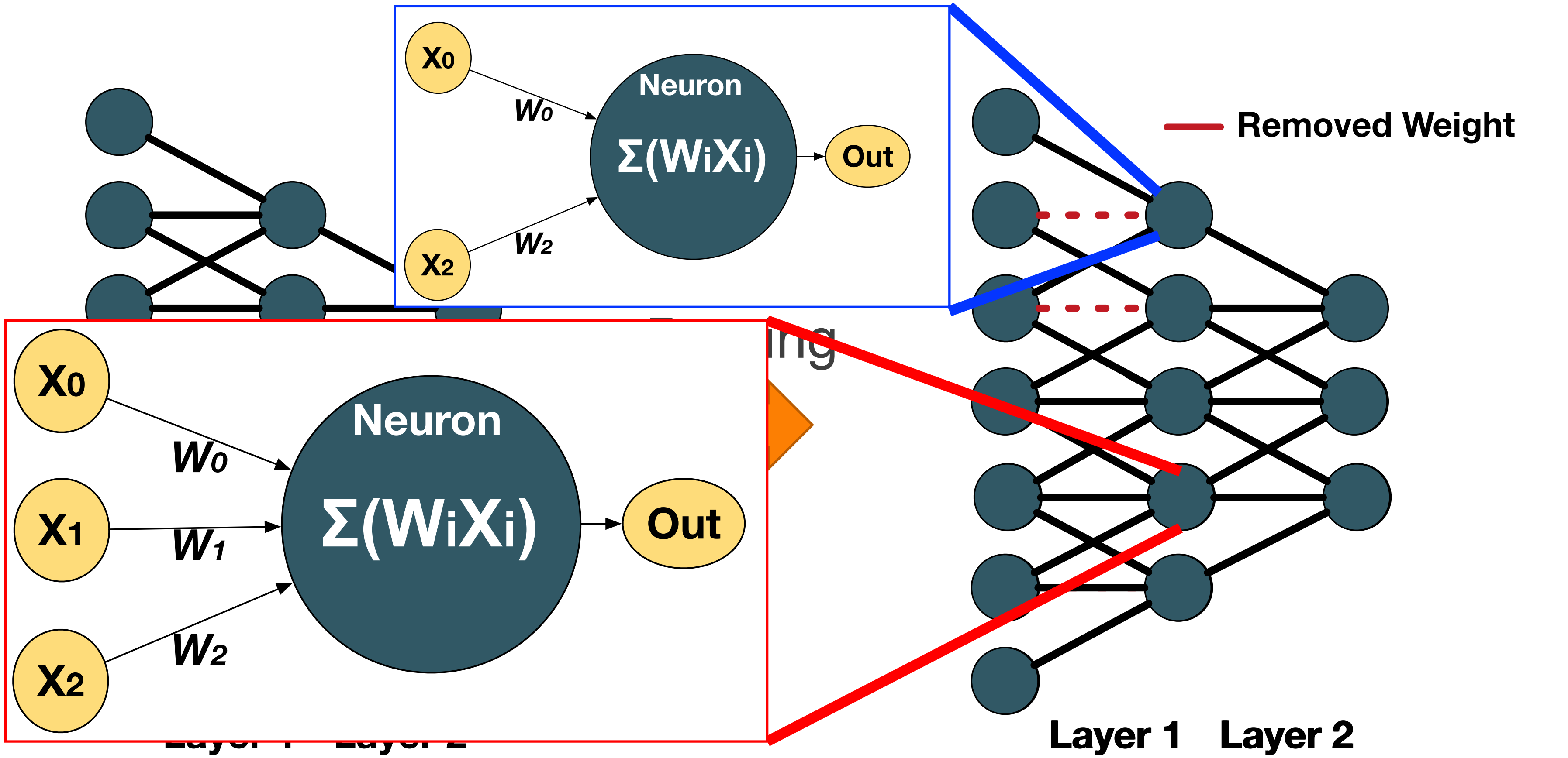

Compute weighted sum

**Accumulation of partial products**

Our Key insight: Each DNN/dataflow translates into neurons of different sizes

# Irregular Dataflow: Pruning

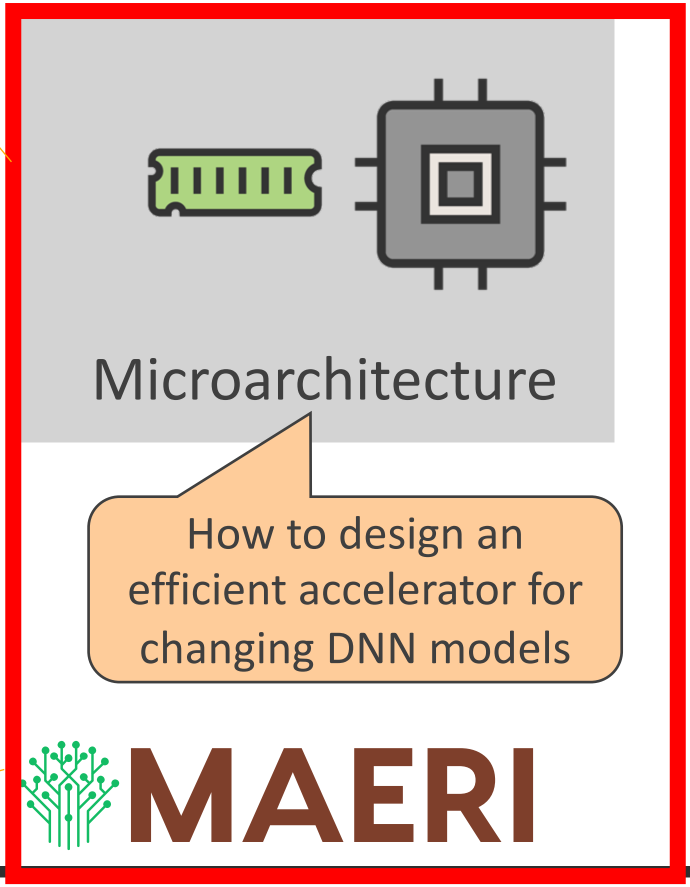
**Example: Weight Pruning (Sparse Workload)**




Our Key insight: Each DNN/dataflow translates into neurons of different sizes

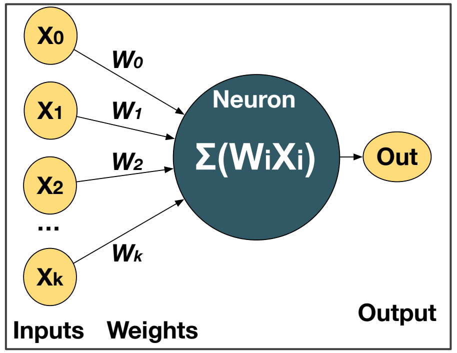
# Outline of Talk

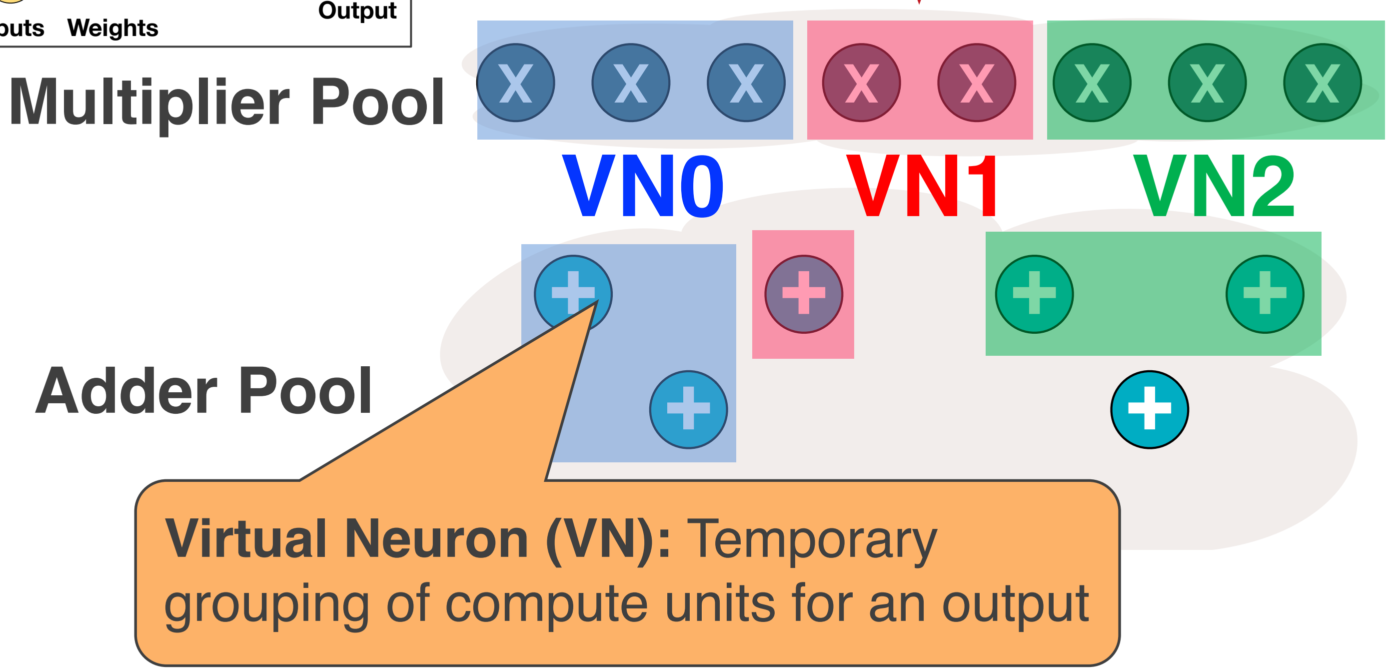
- Motivation
  - Irregular Dataflows
  - DNN Computation
- MAERI
  - Abstraction
  - Implementation
  - Operation Example
  - Mapping Strategies
- Evaluations



# The MAERI Abstraction

MAERI



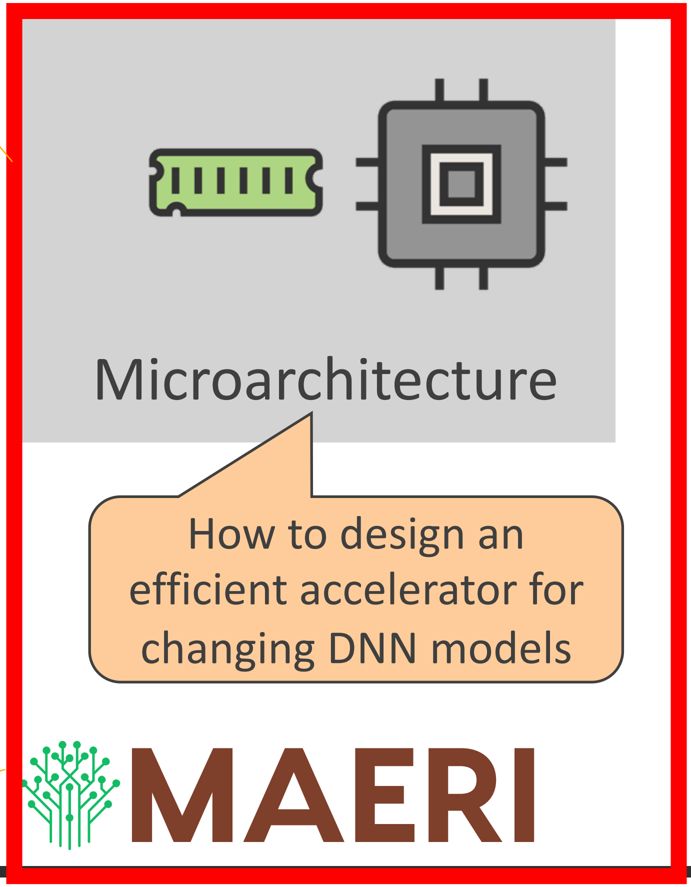

***How to enable flexible grouping?***

**Need flexible connectivity!**

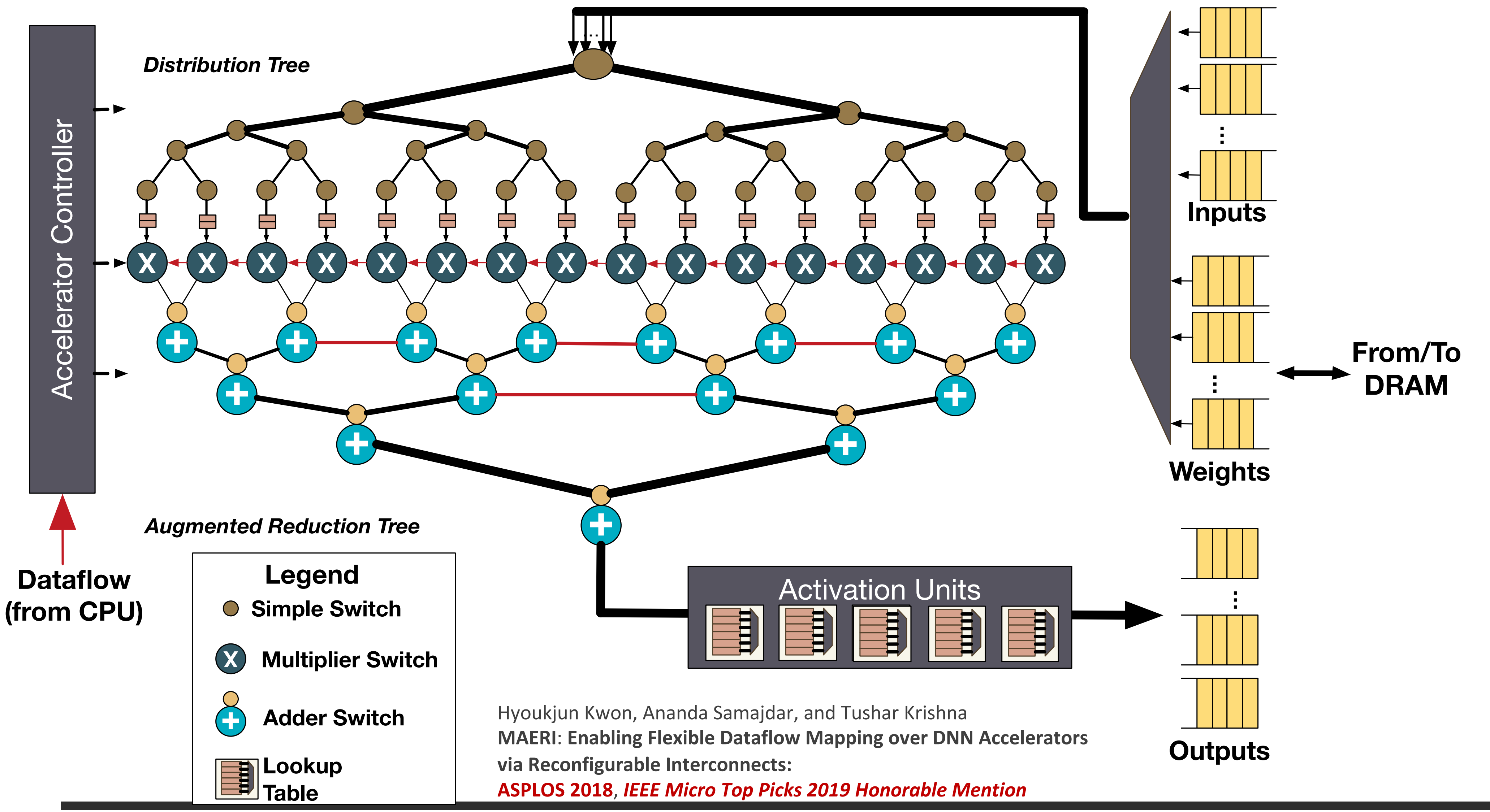
# Outline of Talk

- Motivation
  - Irregular Dataflows
  - DNN Computation
- MAERI
  - Abstraction
  - Implementation
  - Operation Example
  - Mapping Strategies
- Evaluations

Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna
**MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects:**
**ASPLOS 2018**, ***IEEE Micro Top Picks 2019 Honorable Mention***

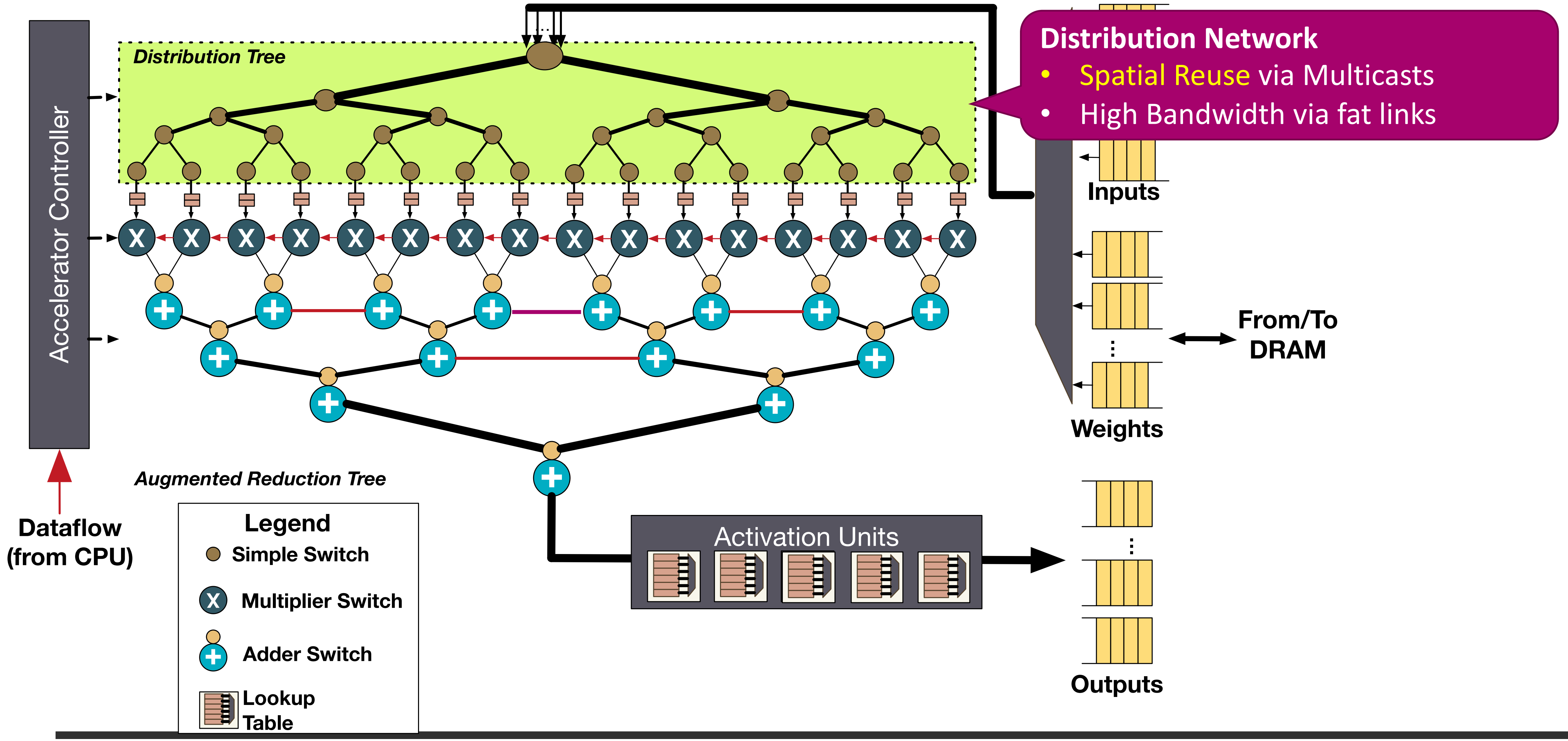


# The MAERI Implementation

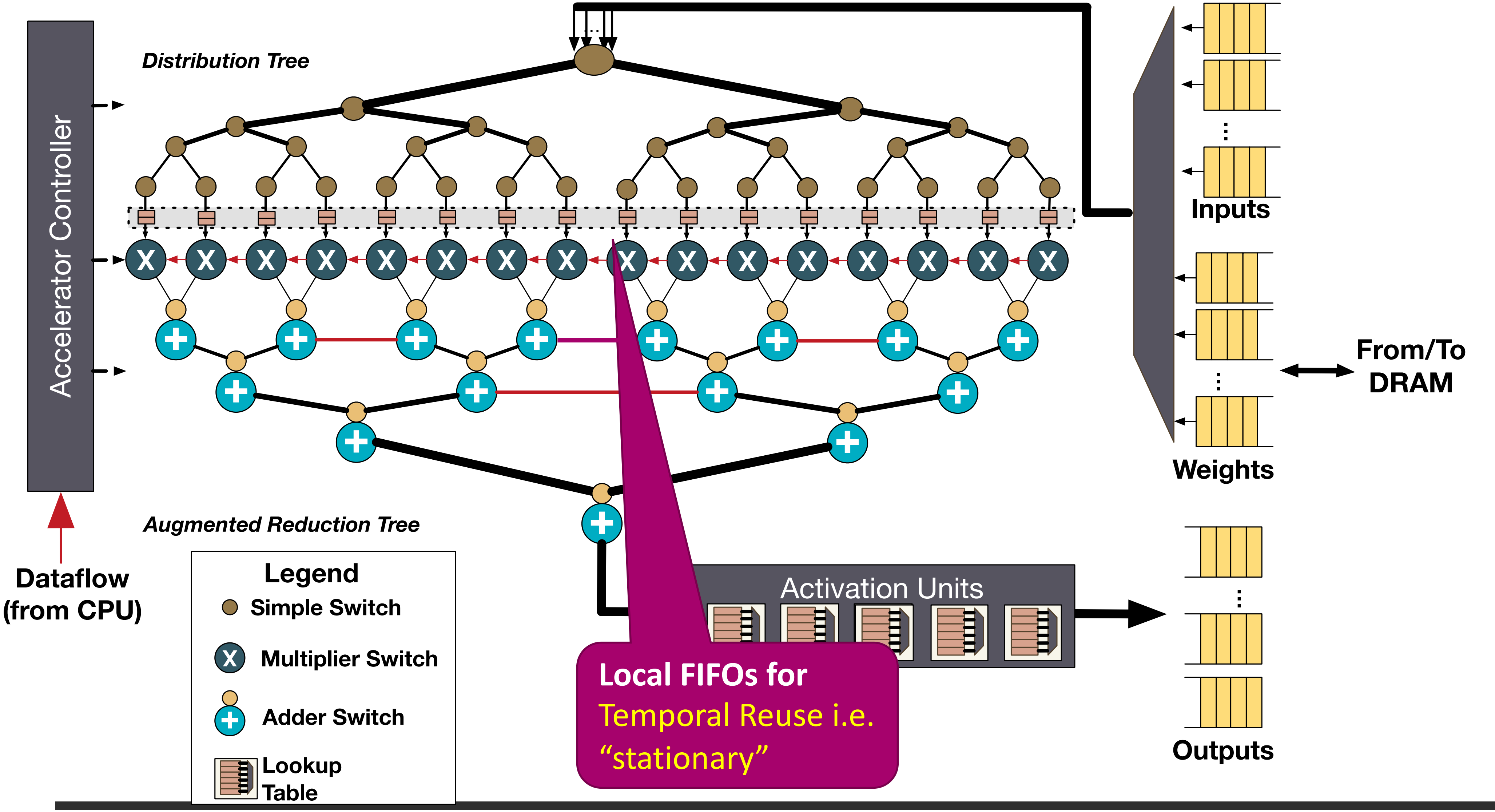



Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna
**MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects:**
**ASPLOS 2018**, ***IEEE Micro Top Picks 2019 Honorable Mention***

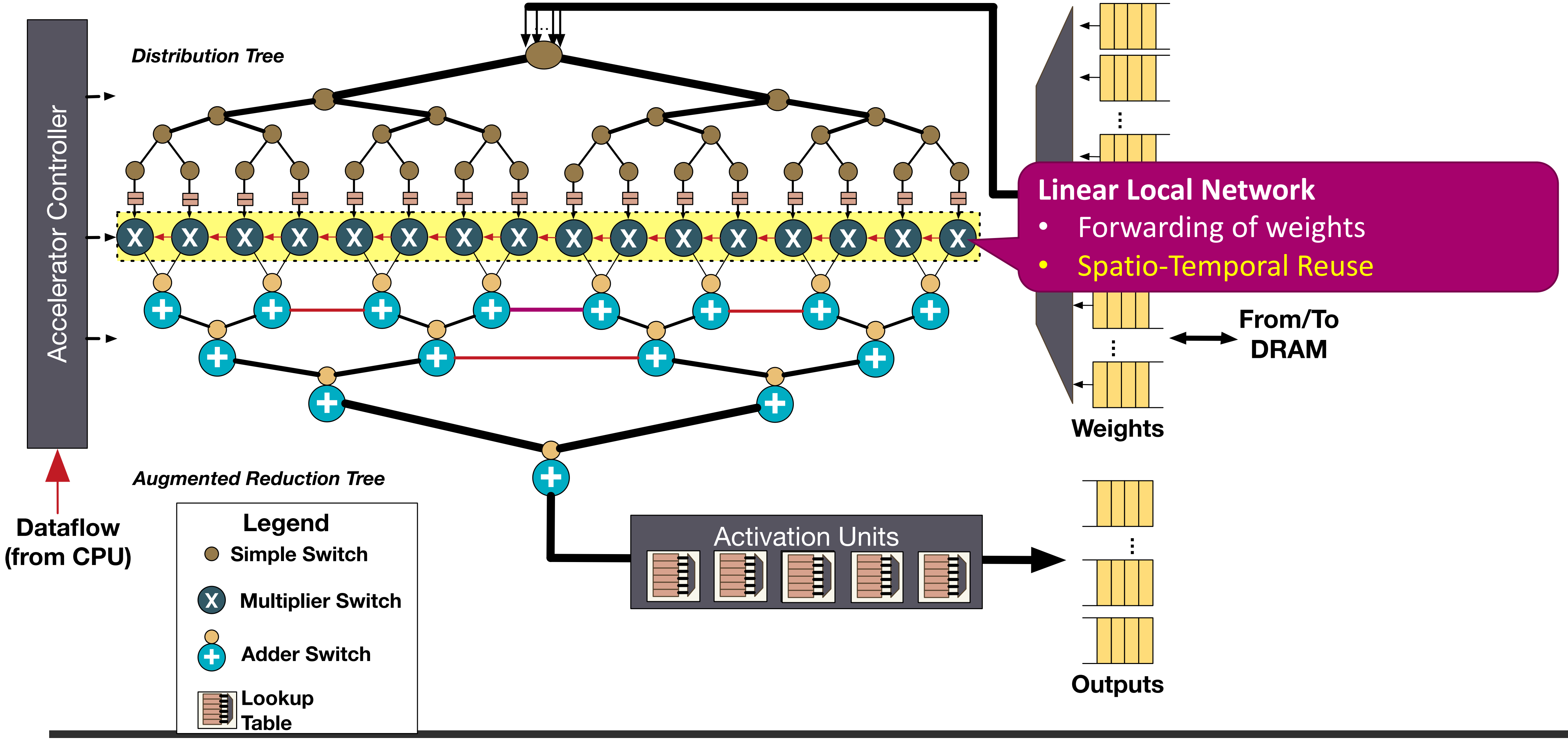
# The MAERI Implementation



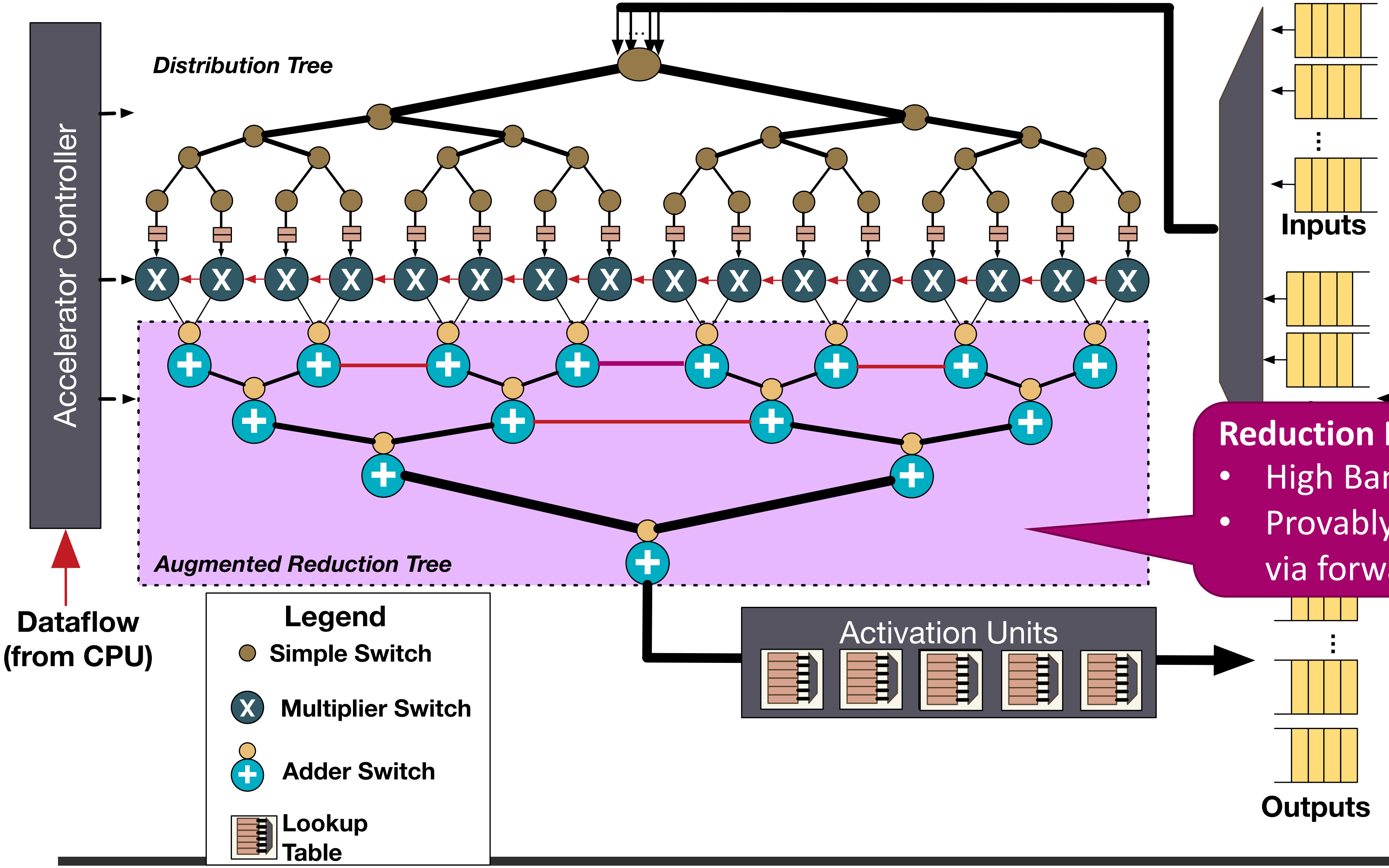
# The MAERI Implementation



# The MAERI Implementation



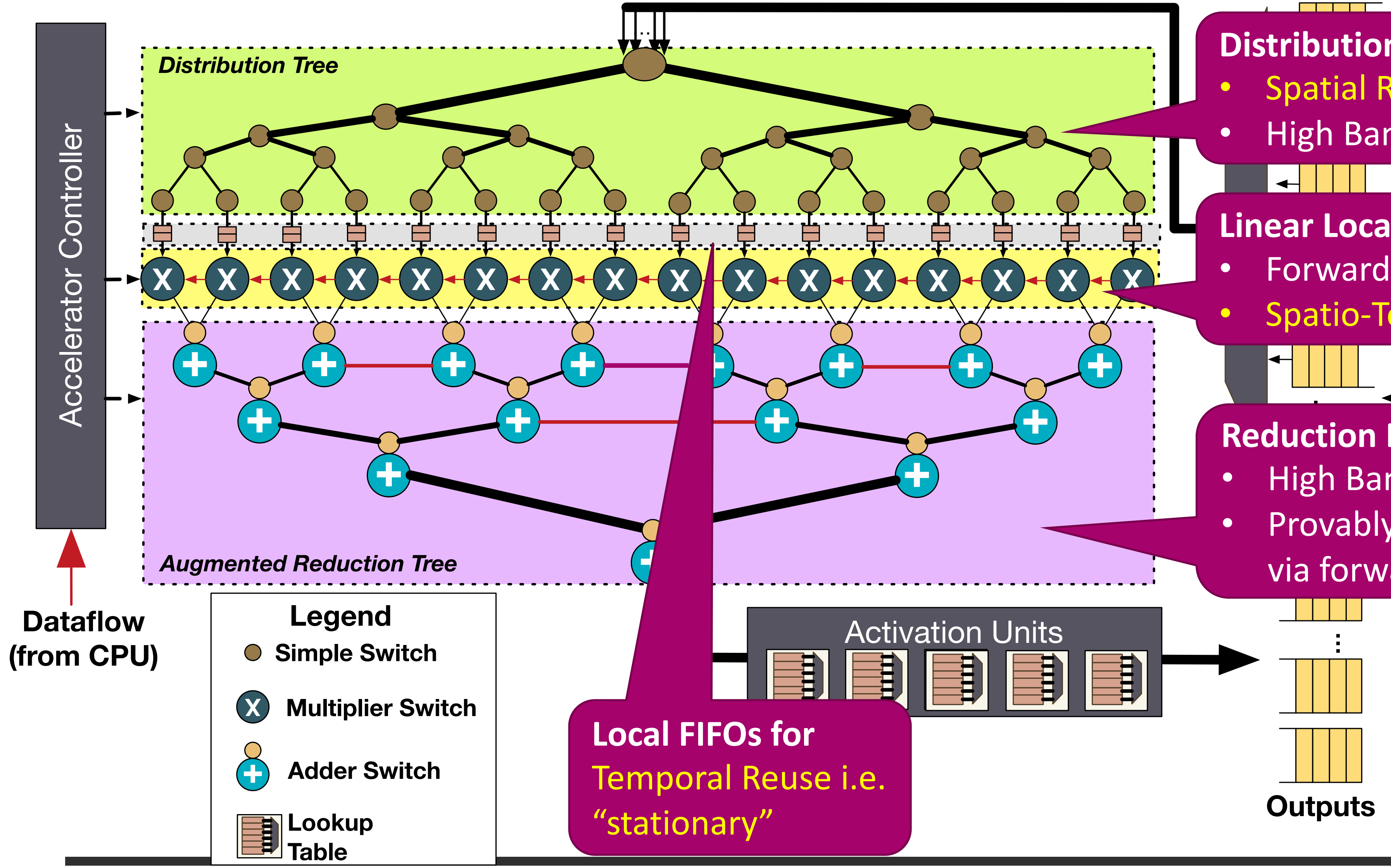
# The MAERI Implementation




**Reduction Network**

- High Bandwidth via fat links
- Provably Non-blocking Reductions via forwarding links

# The MAERI Implementation



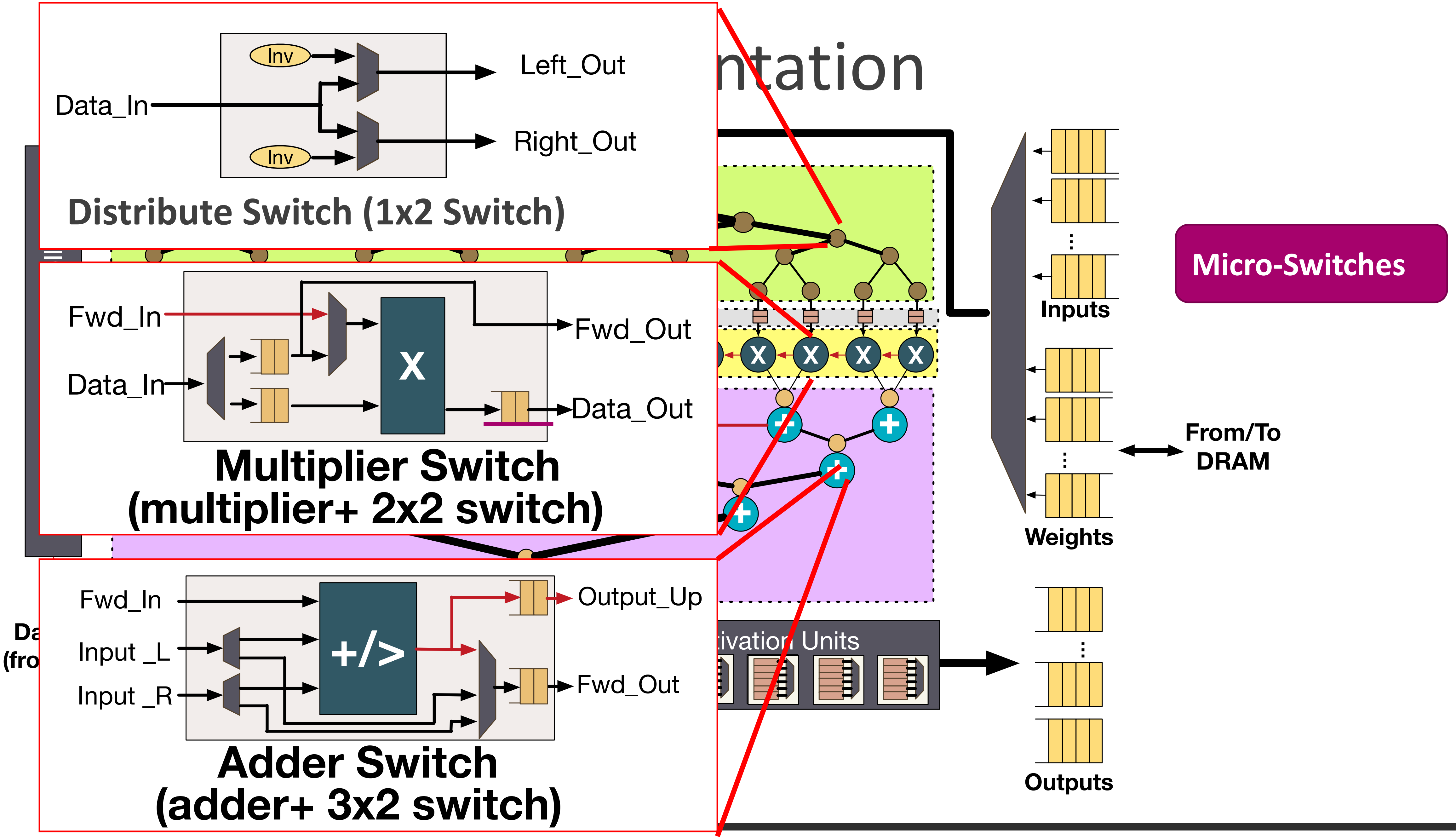

**Distribution Network**
- Spatial Reuse via Multicasts
- High Bandwidth via fat links

**Linear Local Network**
- Forwarding of weights
- Spatio-Temporal Reuse

**Reduction Network**
- High Bandwidth via fat links
- Provably Non-blocking Reductions via forwarding links

**Local FIFOs for** Temporal Reuse i.e. "stationary"

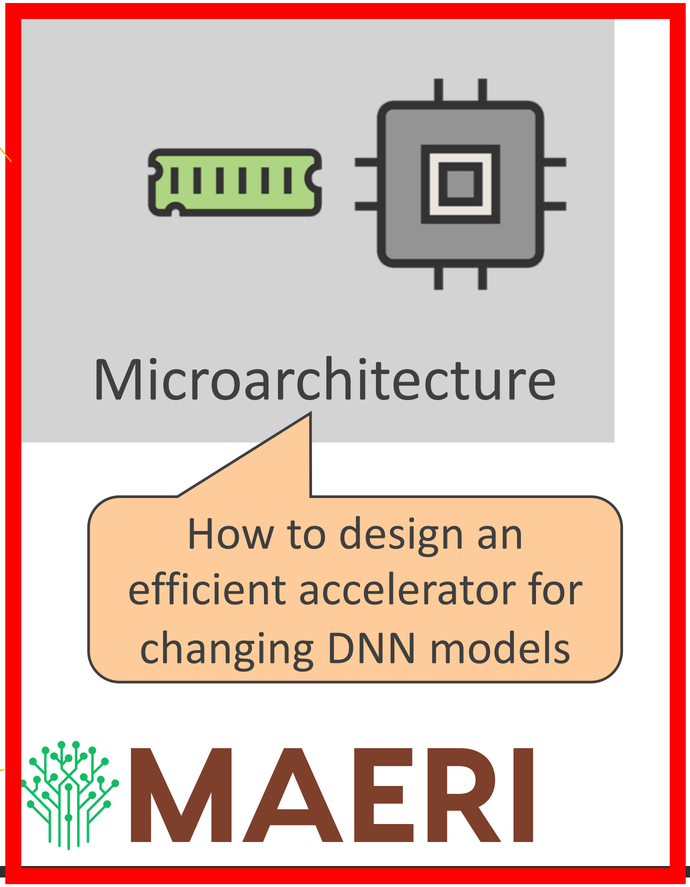
# ntation


Inv
Left_Out
Data_In
Right_Out
Inv
Distribute Switch (1x2 Switch)
Fwd_In
Fwd_Out
Data_In
X
Data_Out
Multiplier Switch
(multiplier+ 2x2 switch)
Fwd_In
Output_Up
Input _L
+/>
Input _R
Fwd_Out
Adder Switch
(adder+ 3x2 switch)
ivation Units
Inputs
Weights
Outputs
From/To
DRAM
Micro-Switches

# Outline of Talk

- Motivation
  - Irregular Dataflows
  - DNN Computation
- MAERI
  - Abstraction
  - Implementation
  - Operation Example
  - Mapping Strategies
- Evaluations

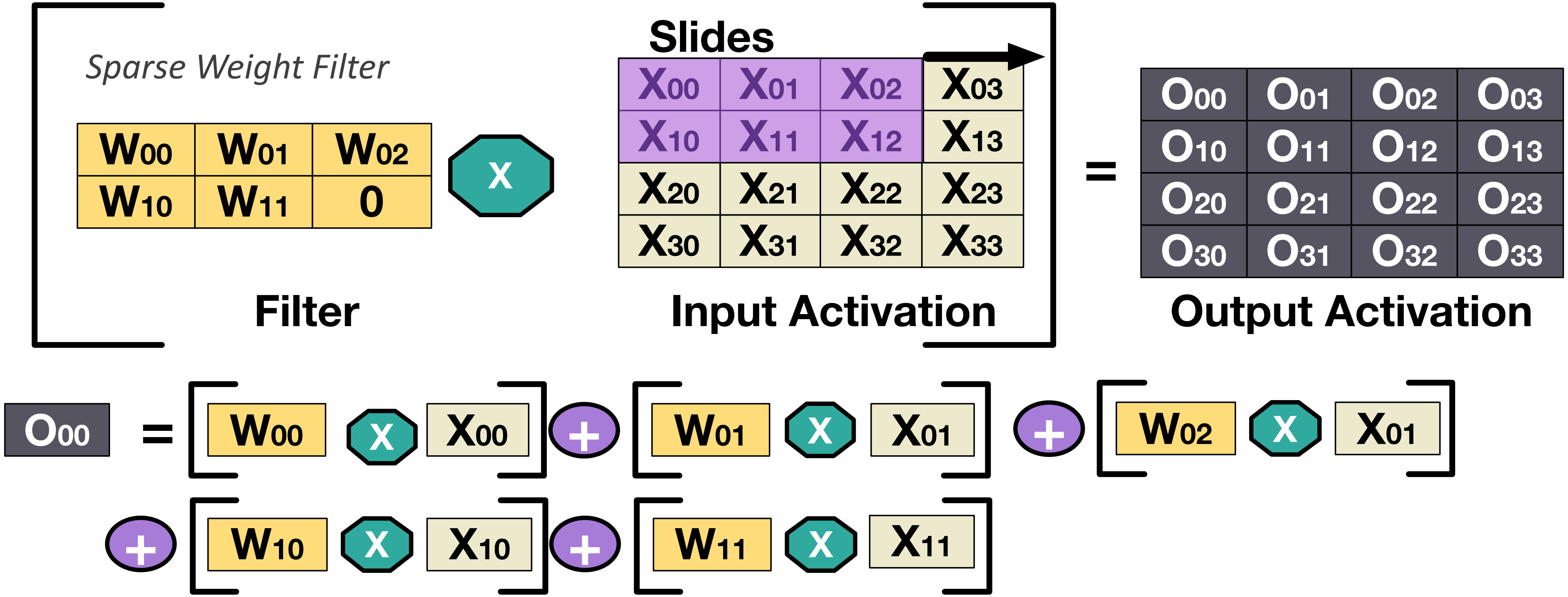
Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna
**MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects:**
**ASPLOS 2018**, ***IEEE Micro Top Picks 2019 Honorable Mention***



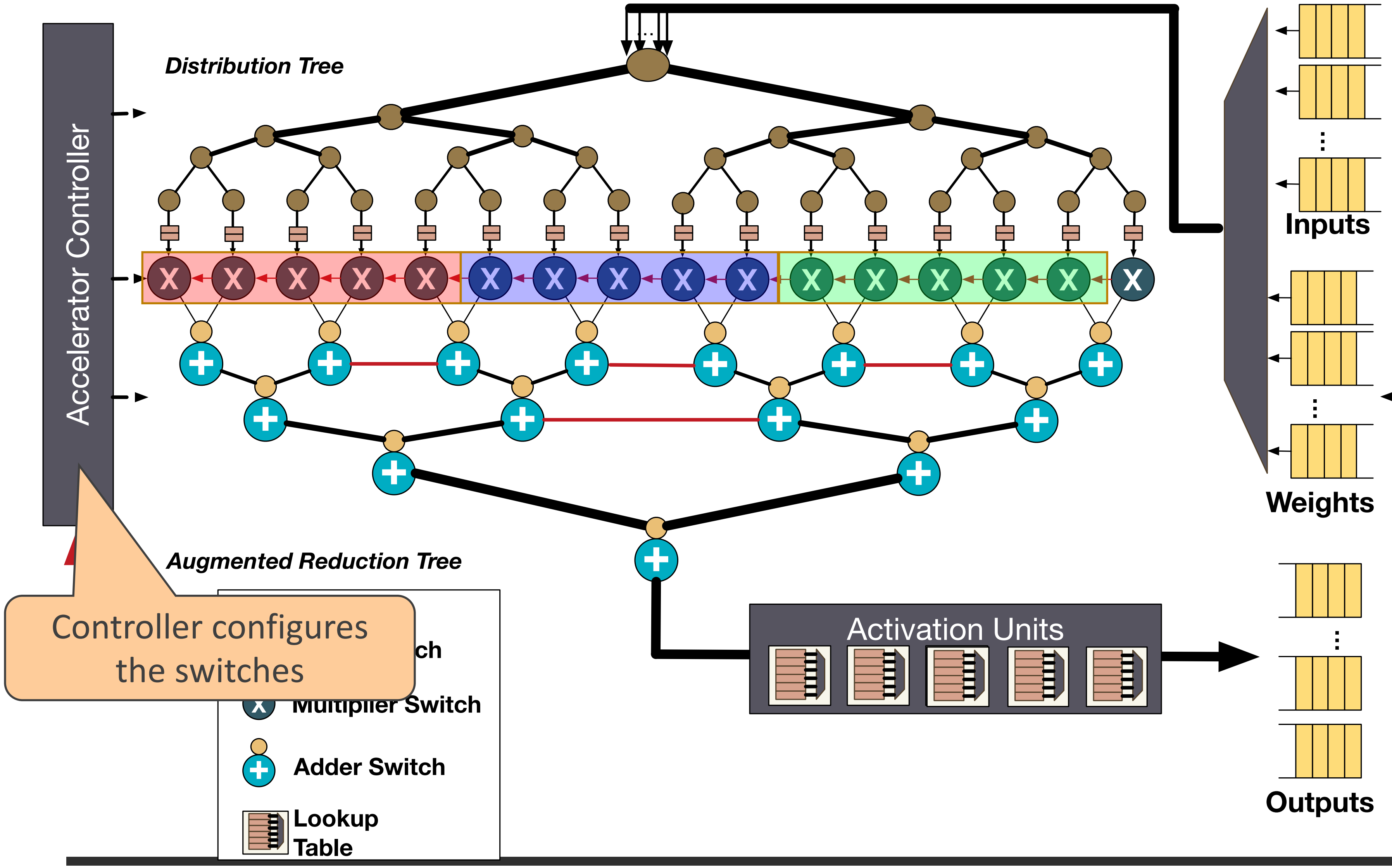
# Example: Computing a CONV layer

- [Communication] **Distribute weights and inputs** (image pixels) to multiplier switches
  - *Assume: weight stationary, conv reuse of inputs via local links*
- [Computation] Compute **partial sums**
- [Computation] **Reduce** partial sums
- [Communication] **Collect** outputs to buffer

# MAERI Operation Example



# MAERI Operation Example



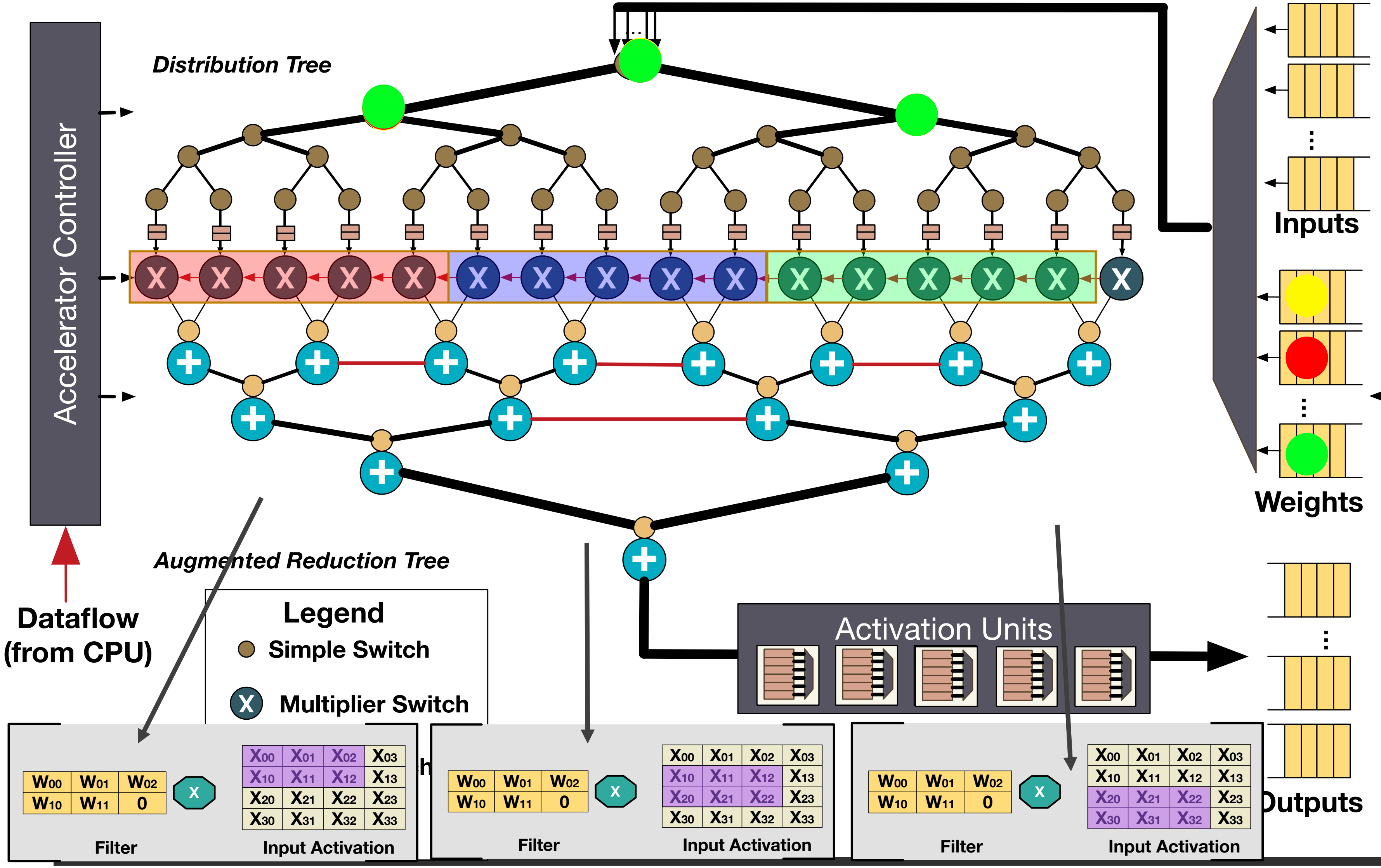

| | | |
|---|---|---|
| $W_{00}$ | $W_{01}$ | $W_{02}$ |
| $W_{10}$ | $W_{11}$ | 0 |

Sparse Weight Filter

**<Step 1>**
Virtual Neuron Construction

VN size = 5

# MAERI Operation Example



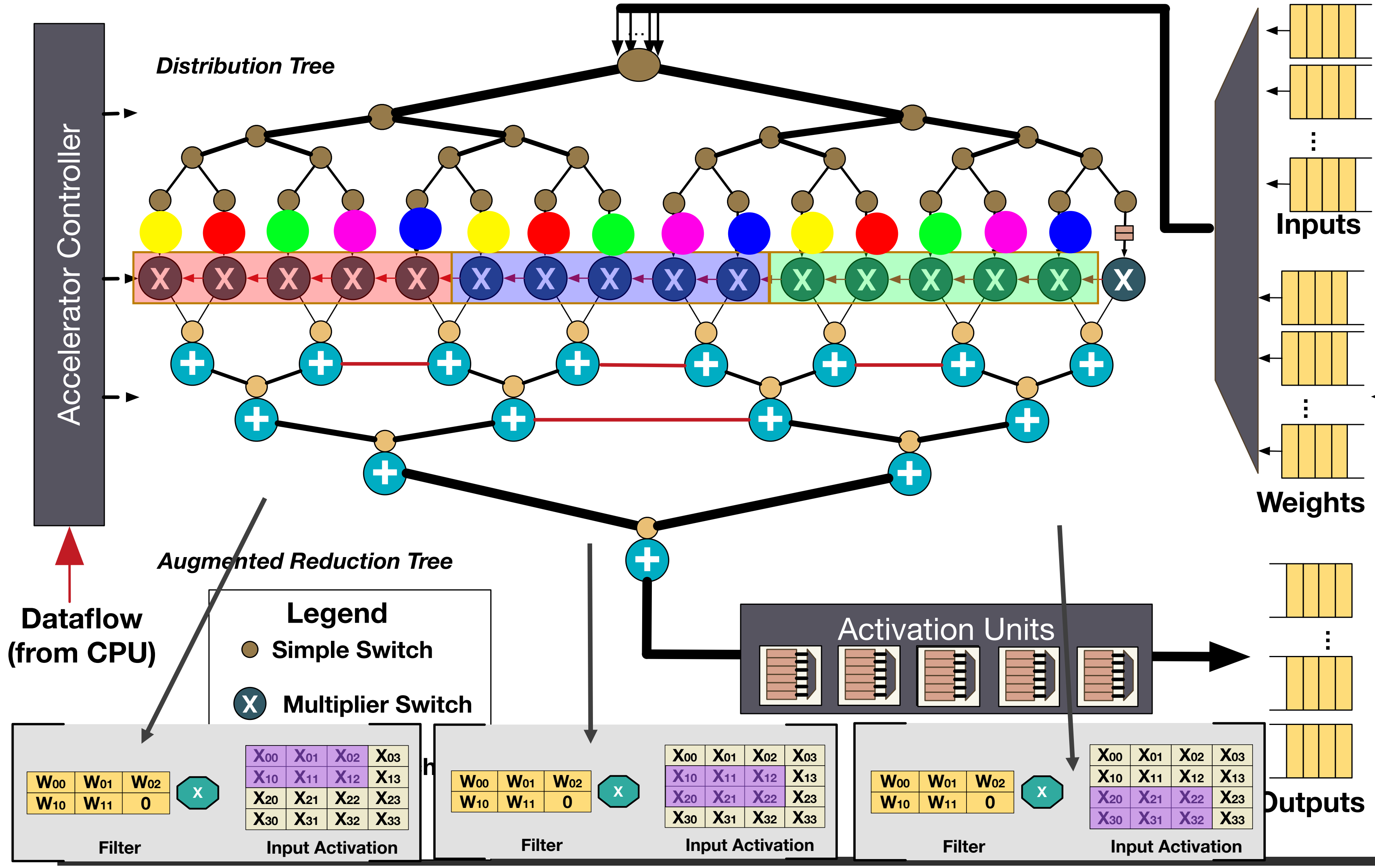

| | | |
|---|---|---|
| $W_{00}$ | $W_{01}$ | $W_{02}$ |
| $W_{10}$ | $W_{11}$ | 0 |

Sparse Weight Filter

**<Step 2>**
**Weight Distribution**

Distribution bandwidth is tunable.
**Suppose BW = 3**

# MAERI Operation Example



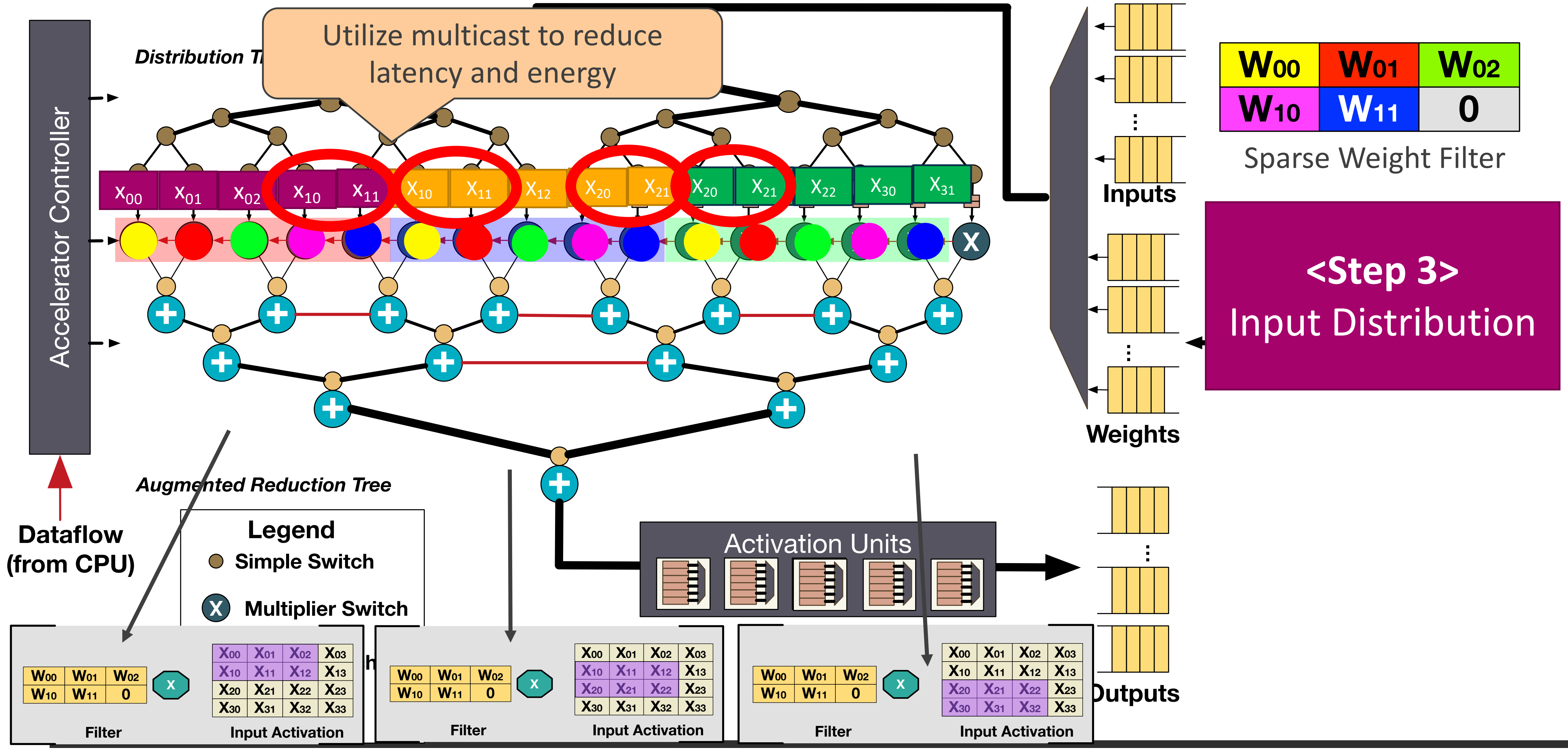

| | | |
|---|---|---|
| $W_{00}$ | $W_{01}$ | $W_{02}$ |
| $W_{10}$ | $W_{11}$ | 0 |

Sparse Weight Filter

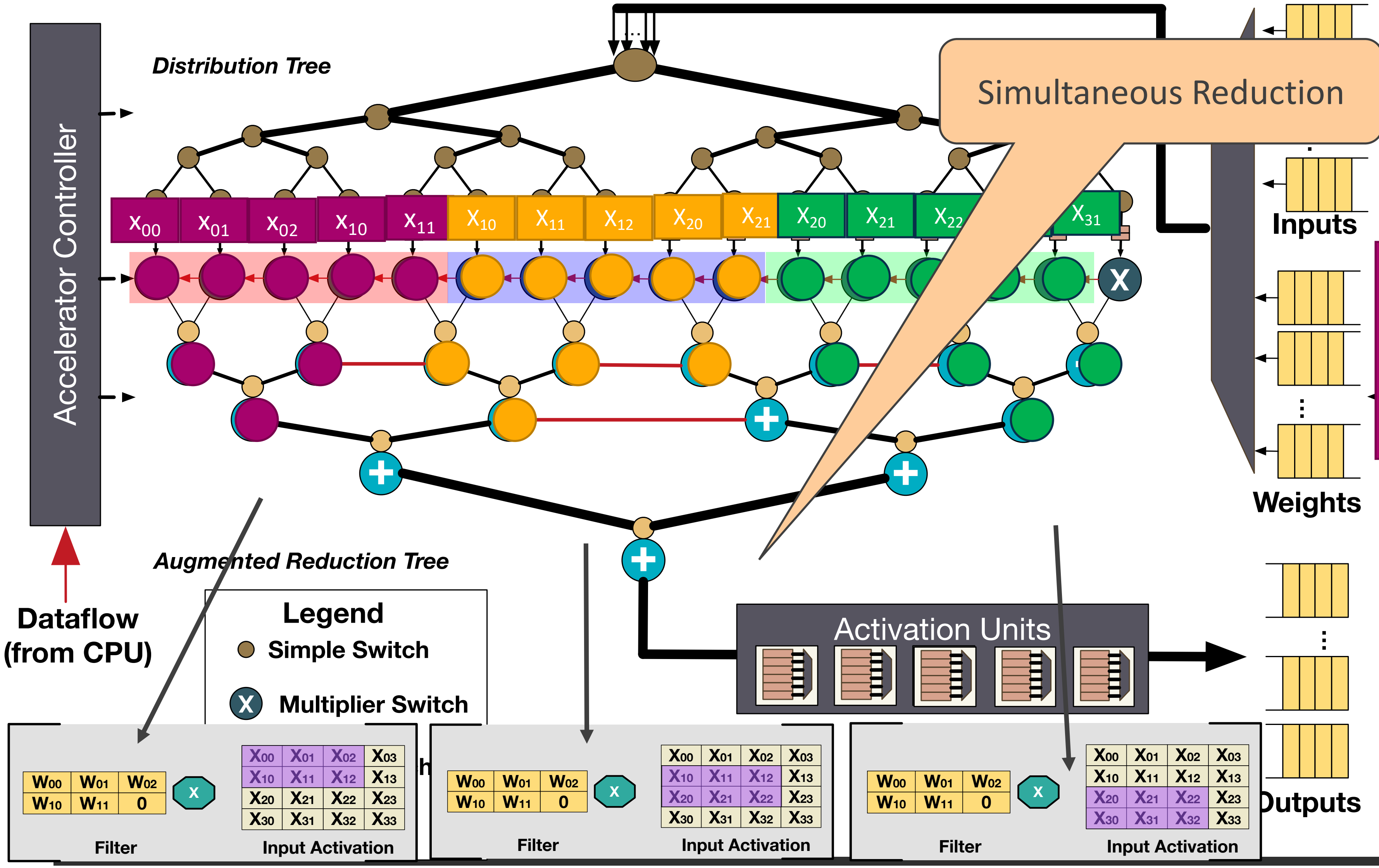
**<Step 2>**
Weight Distribution

Distribution bandwidth is tunable.
**Suppose BW = 3**

# MAERI Operation Example

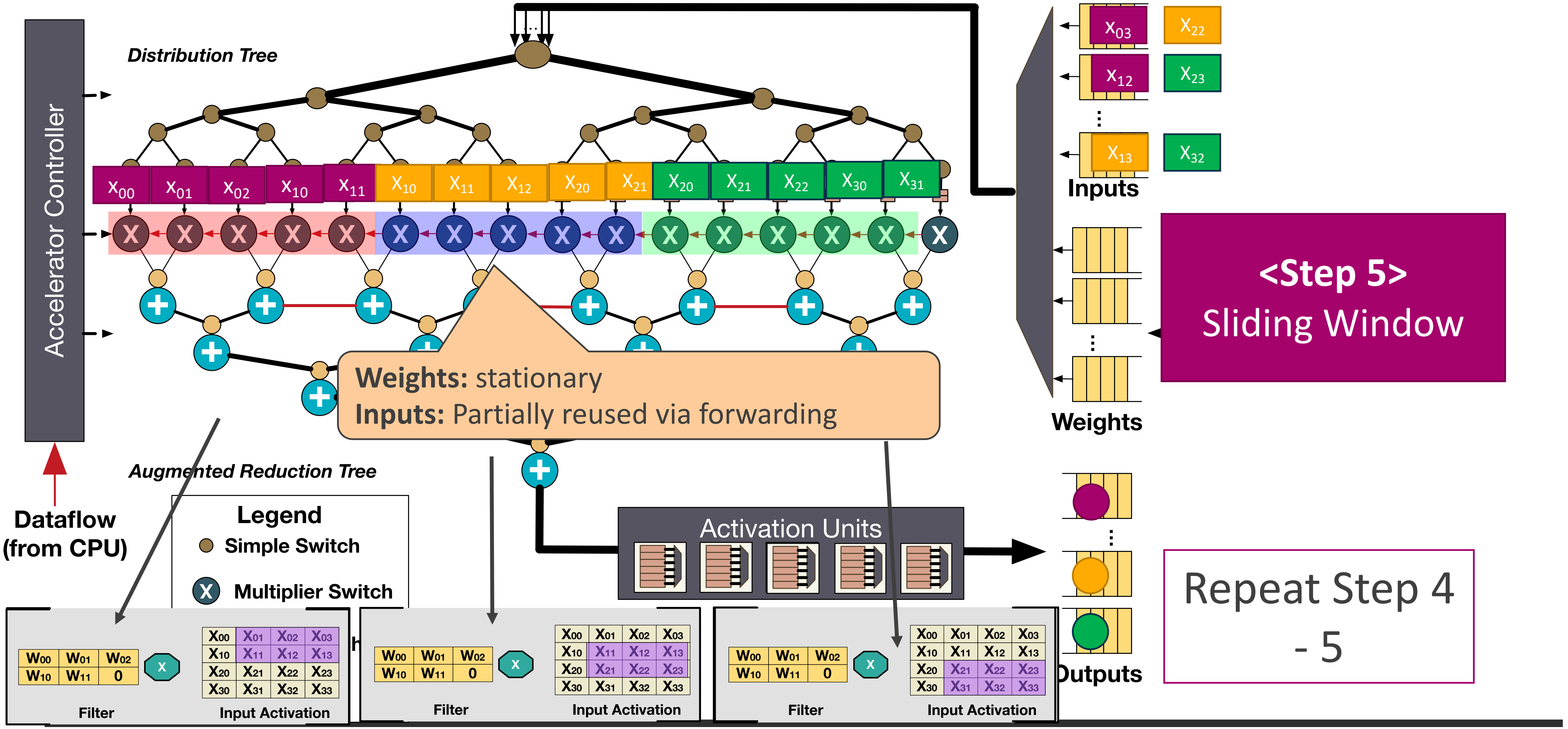


# MAERI Operation Example




<Step 4> Partial sum reduction

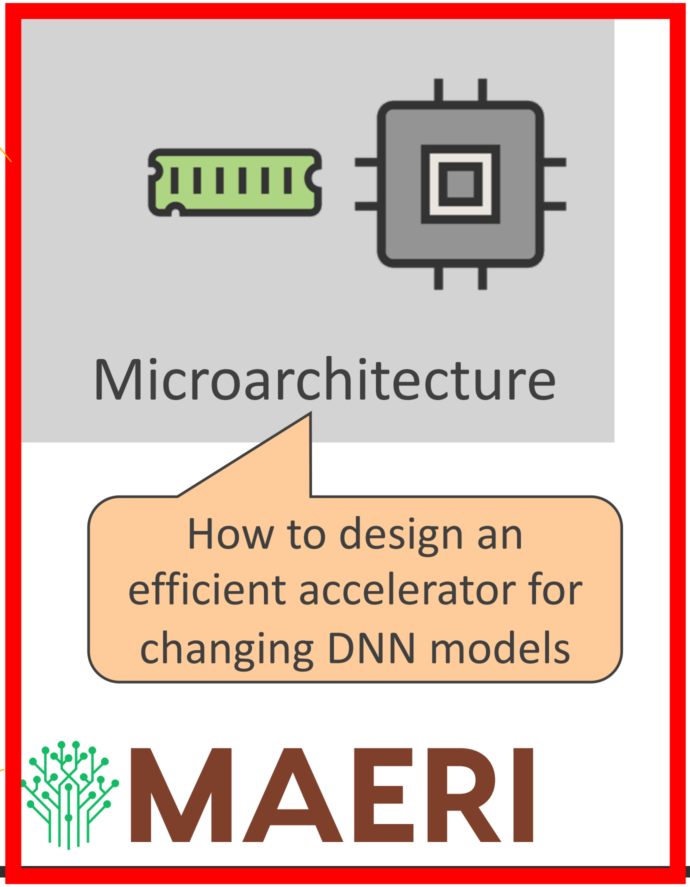
# MAERI Operation Example



# Outline of Talk

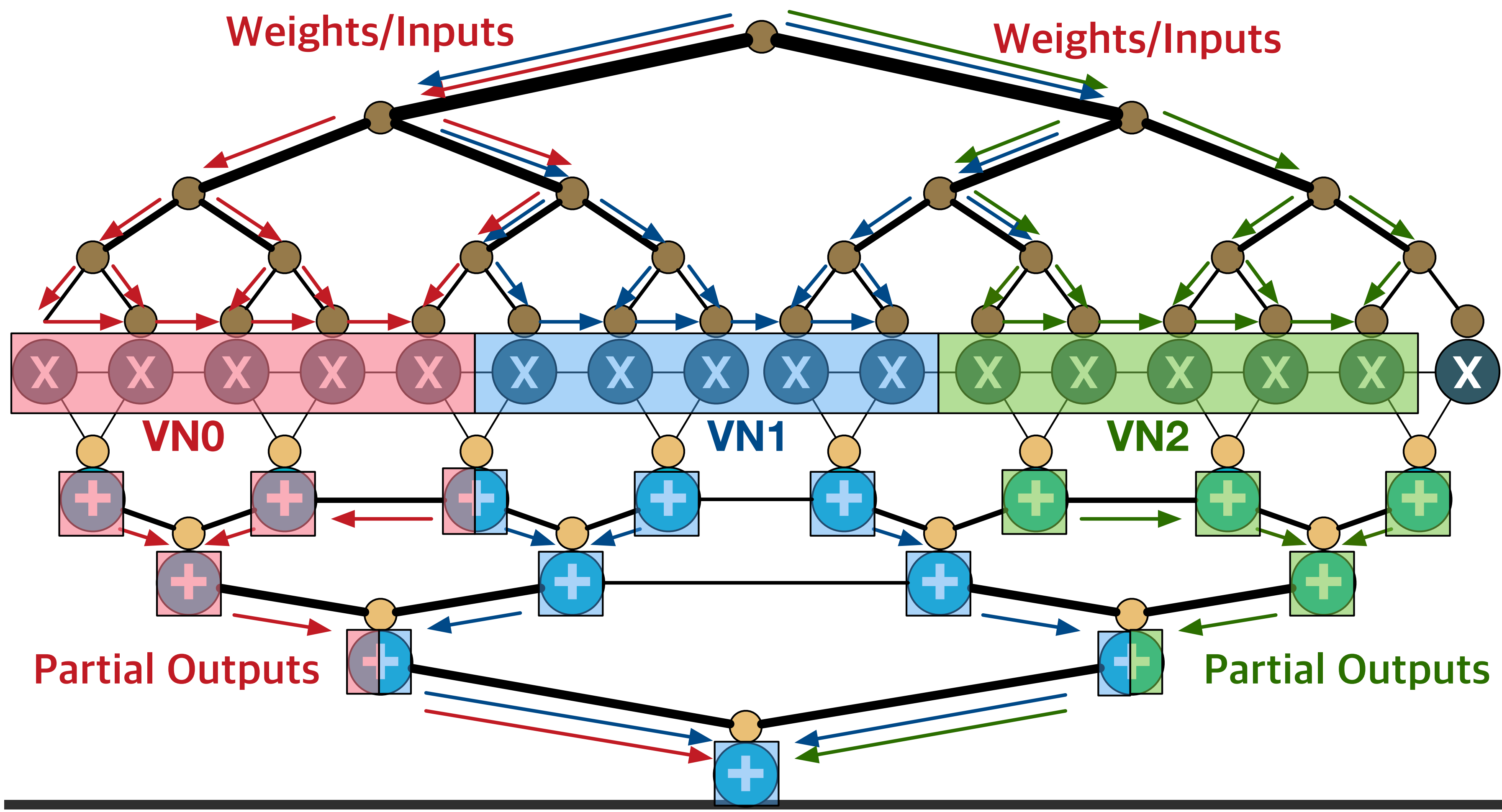
- Motivation
  - Irregular Dataflows
  - DNN Computation
- MAERI
  - Abstraction
  - Implementation
  - Operation Example
  - Mapping Strategies
- Evaluations

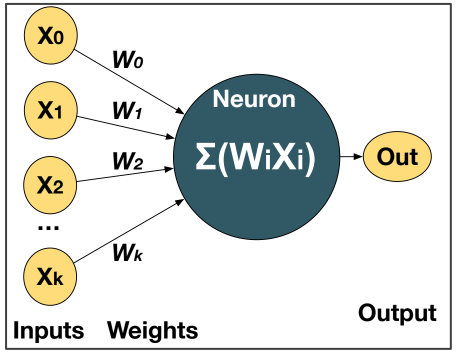
Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna
**MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects:**
**ASPLOS 2018**, ***IEEE Micro Top Picks 2019 Honorable Mention***



# Example Mapping – Dense CNN

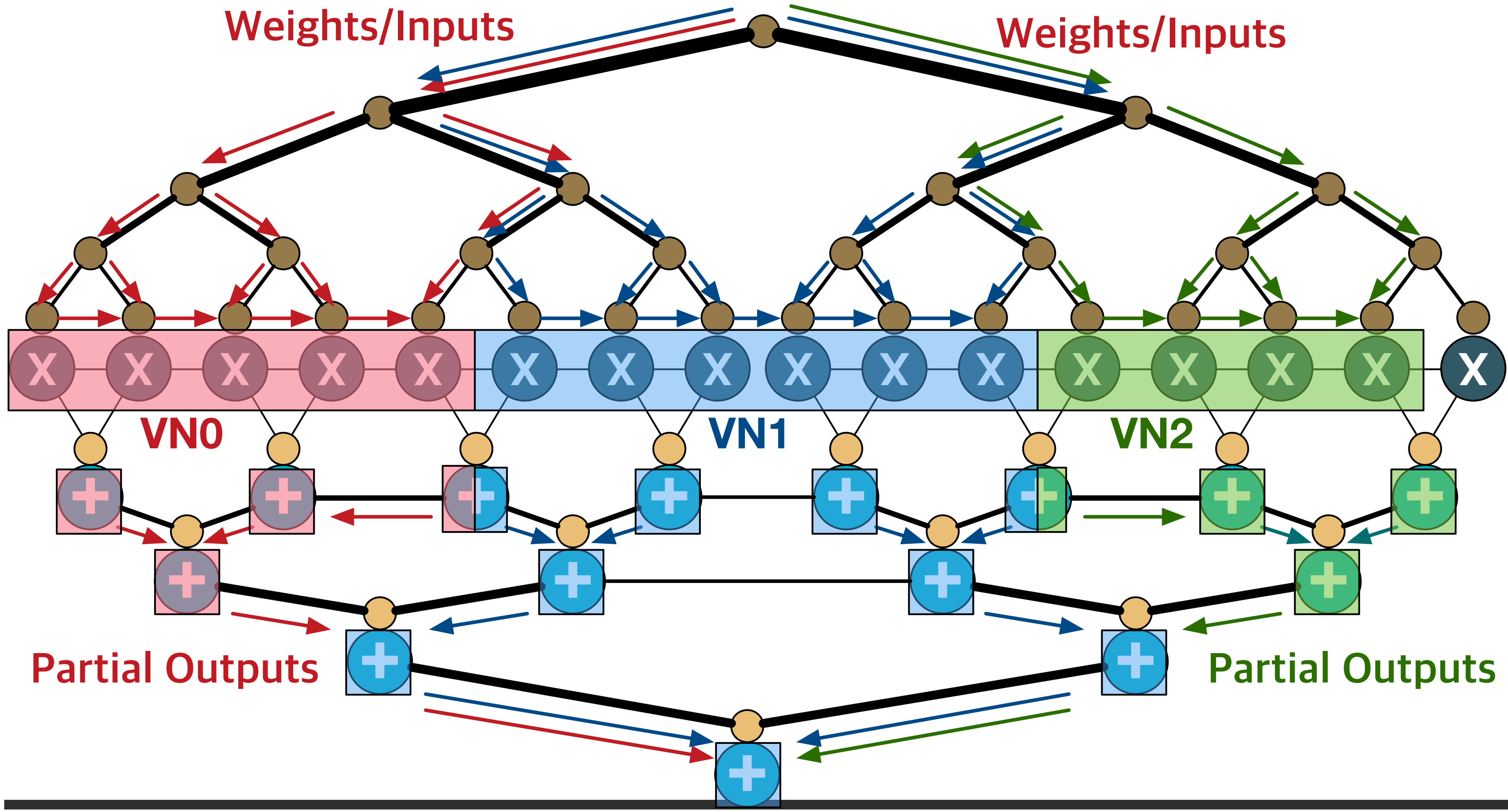
Our Key insight: Each DNN/dataflow translates into neurons of different sizes

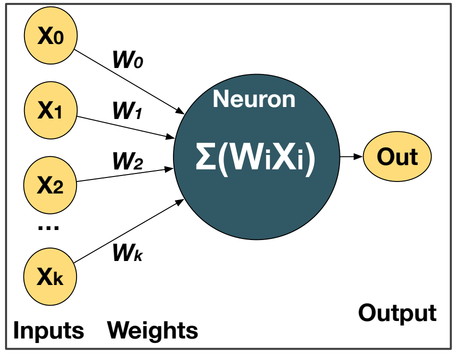


# Example Mapping – Sparse DNN

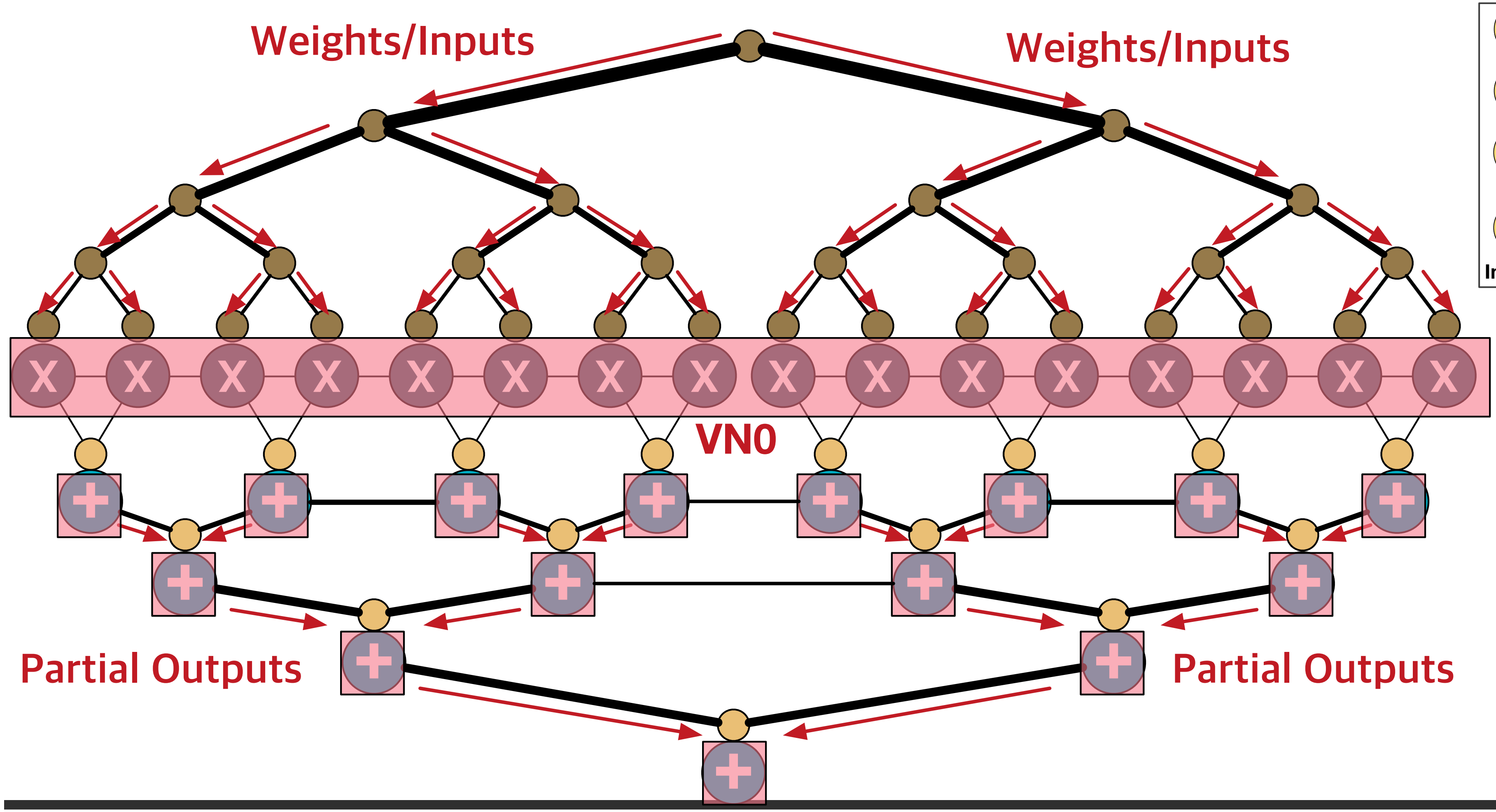
Our Key insight: Each DNN/dataflow translates into neurons of different sizes

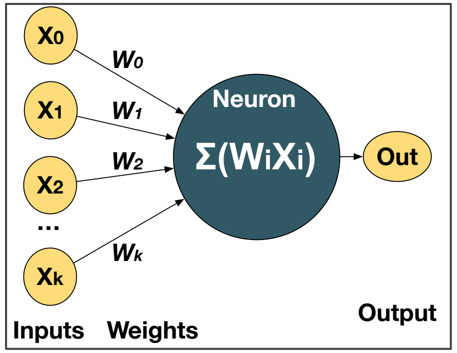


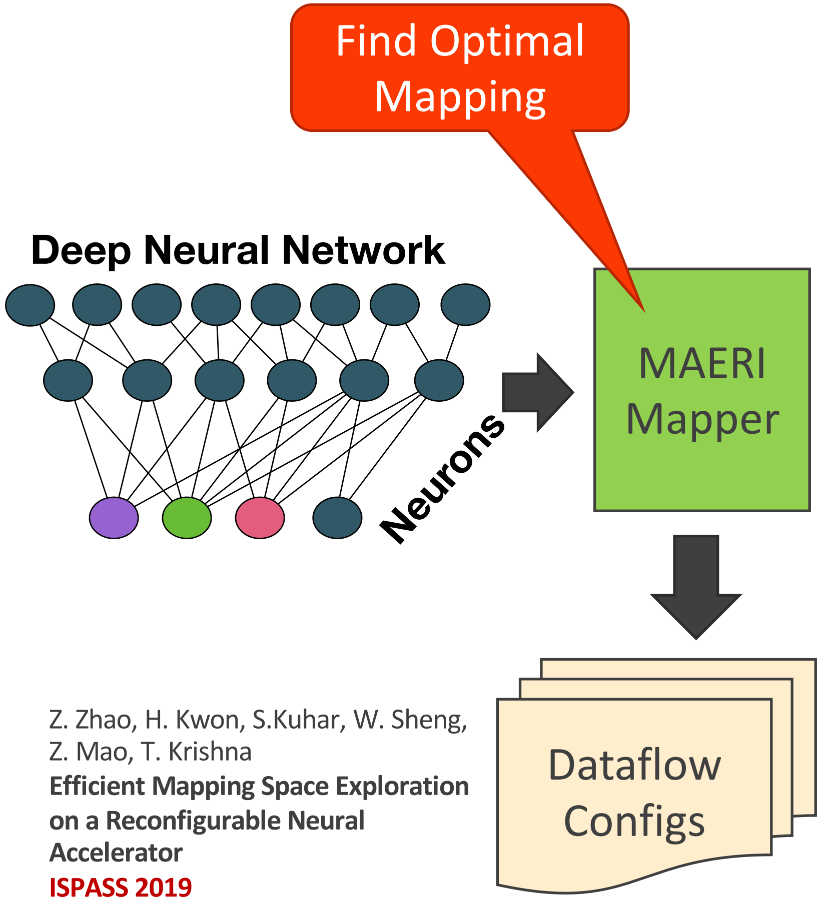
# Example Mapping – LSTM/FC

**Our Key insight: Each DNN/dataflow translates into neurons of different sizes**

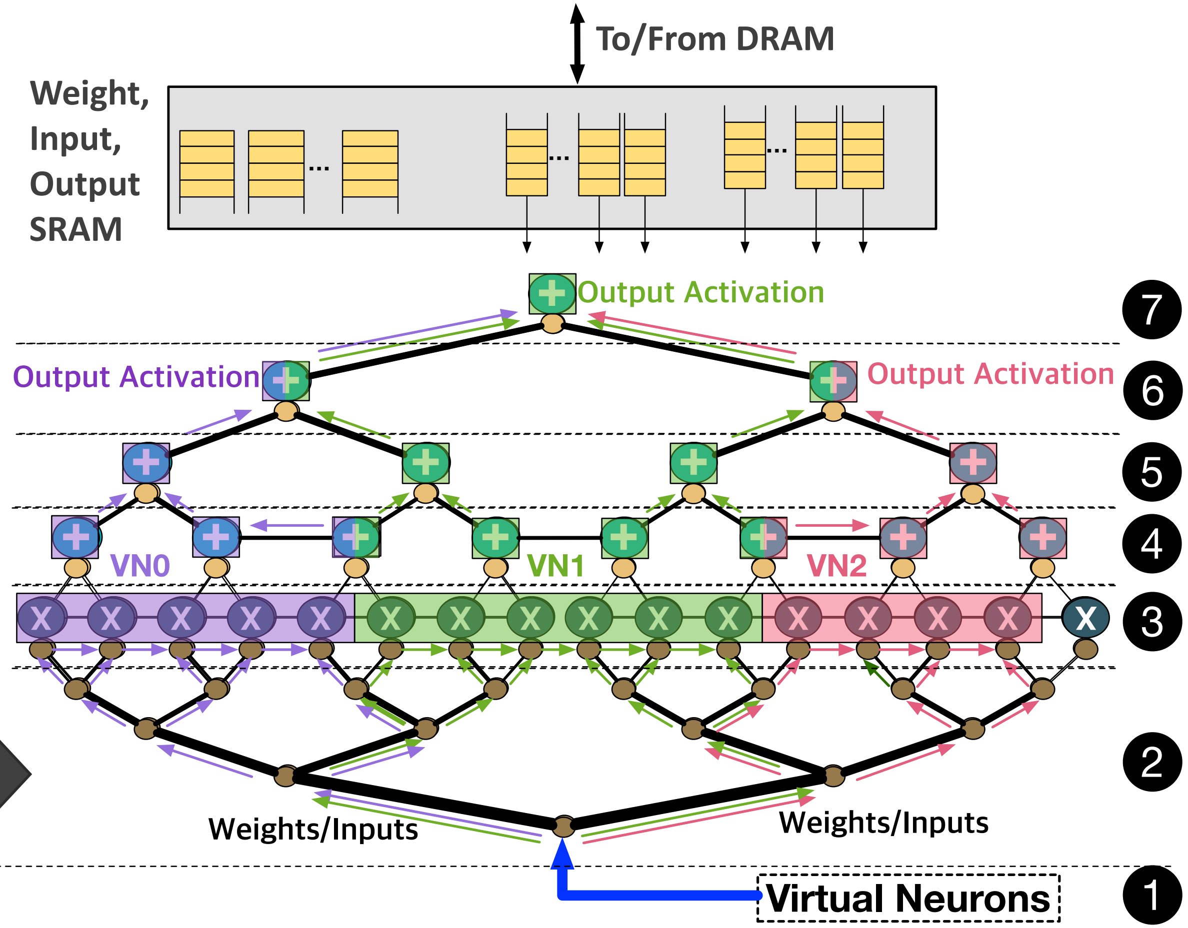


# Searching optimal dataflows for MAERI




Z. Zhao, H. Kwon, S.Kuhar, W. Sheng, Z. Mao, T. Krishna
**Efficient Mapping Space Exploration on a Reconfigurable Neural Accelerator**
ISPASS 2019




~100% Utilization

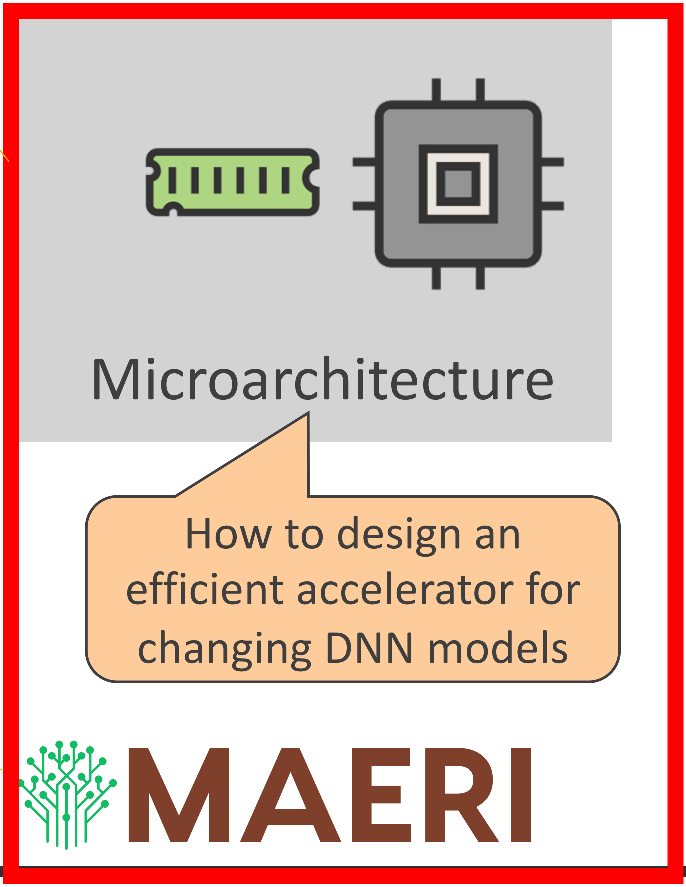
Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna
**MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects:**
ASPLOS 2018, *IEEE Micro Top Picks 2019 Honorable Mention*

MAERI

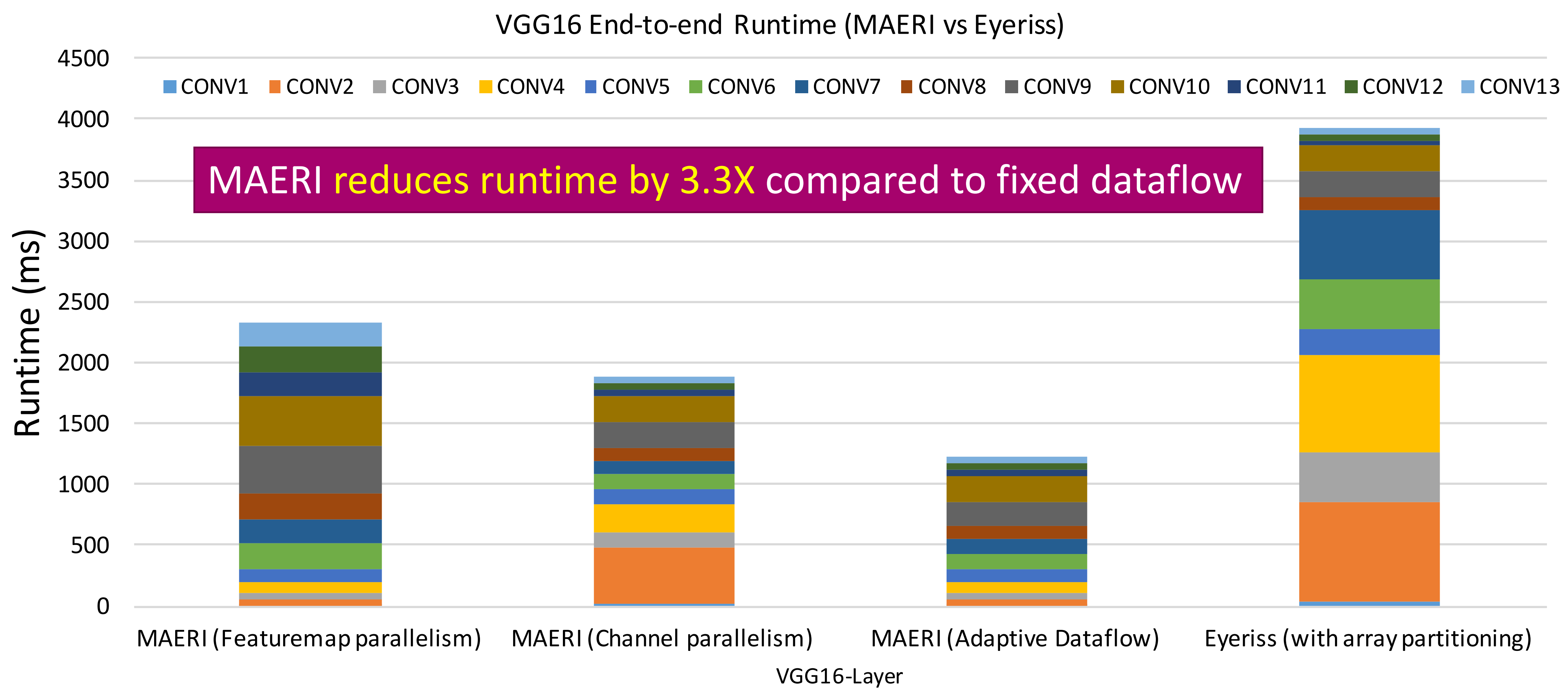
# Outline of Talk

- Motivation
  - Irregular Dataflows
  - DNN Computation
- MAERI
  - Abstraction
  - Implementation
  - Operation Example
  - Mapping Strategies
- Evaluations

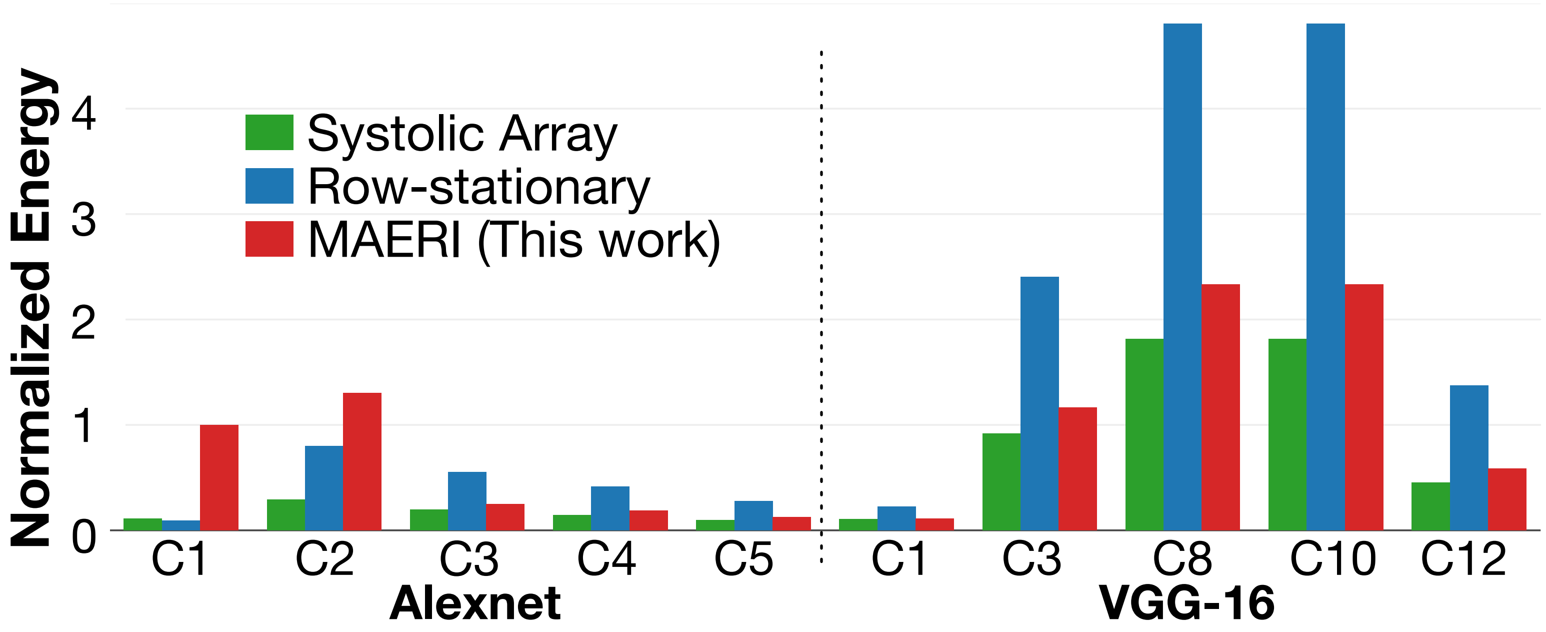
Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna
**MAERI: Enabling Flexible Dataflow Mapping over DNN Accelerators via Reconfigurable Interconnects:**
**ASPLOS 2018**, ***IEEE Micro Top Picks 2019 Honorable Mention***



# End-to-End Performance



# Energy with Convolution Layers




* Normalized to MAERI energy with Alexnet C1

MAERI reduces energy upto 57% and 28% in average compared to Row-Stationary (dense dataflow) and 7.1% in average compared to Systolic Array (sparse dataflow)

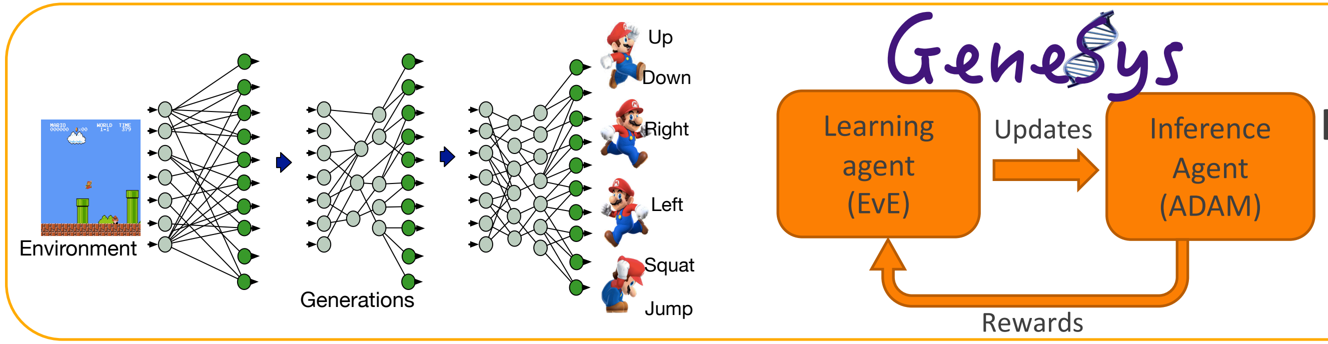
# Summary of MAERI

- DNN models evolving rapidly
  - Multiple layer types
  - Sparsity Optimizations
  - Myriad dataflows for scheduling and mapping

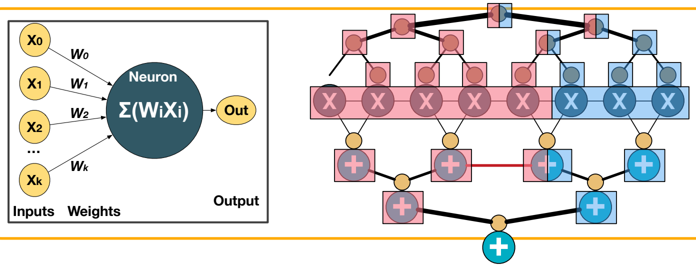
- MAERI enables dynamic grouping of arbitrary number of MACCs (“Virtual Neuron”) via reconfigurable, non-blocking interconnects, providing
  - Future proof to DNN models and dataflows
  - Near 100% compute unit utilization

# Takeaways

AI will be pervasive




HW-SW Co-Design of NE Algorithms shows promise for continuous learning at the edge




MAERI

DNN Accelerator with Configurable Interconnects can map Irregular Dataflows

Thank you!

http://synergy.ece.gatech.edu