Georgia Tech School of Electrical and Computer Engineering
College of Engineering

http://synergy.ece.gatech.edu

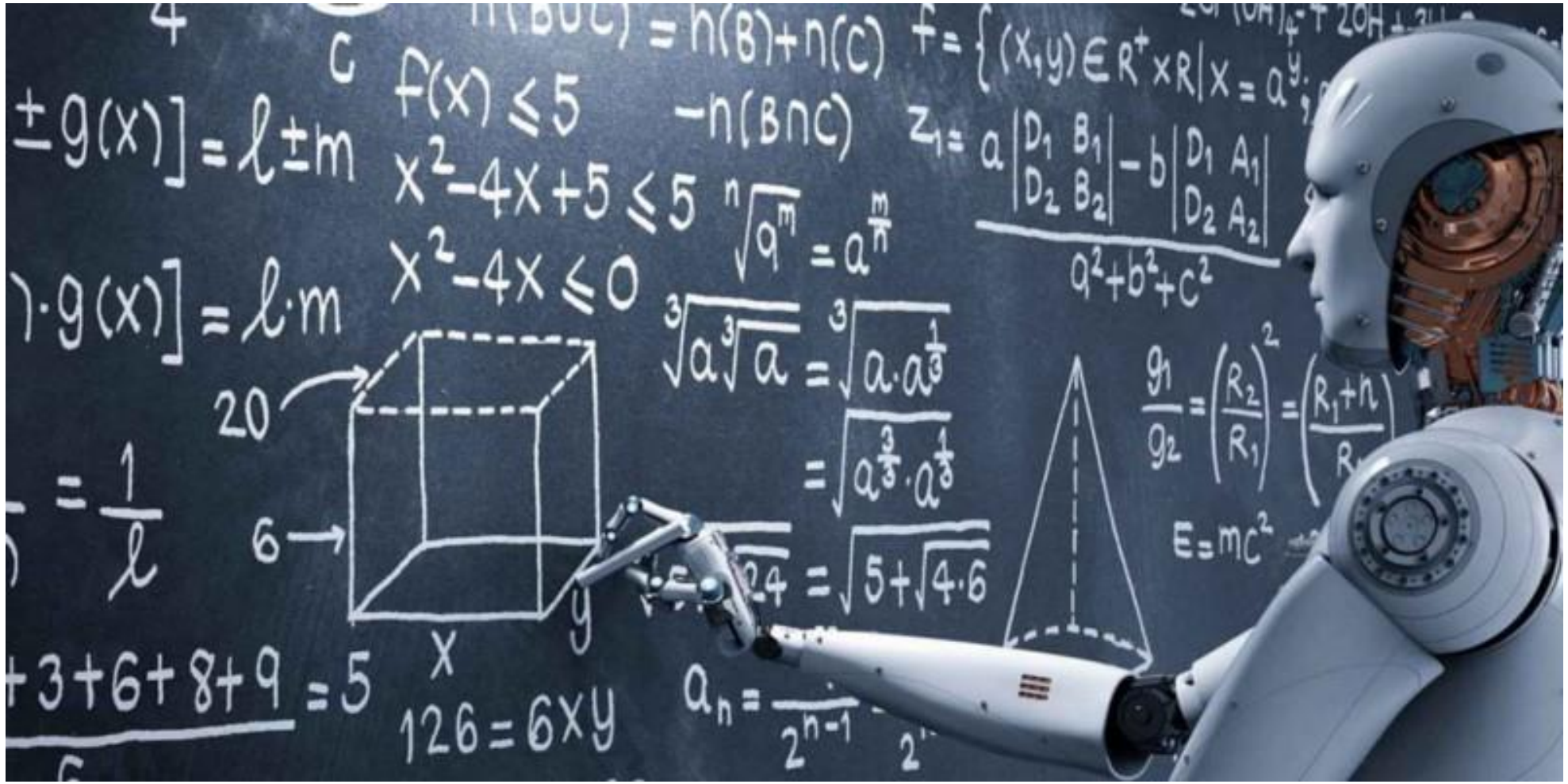
GeneSys

ARM Research Summit
September 16 ,2019

Enabling Continuous Learning through Neural Network Evolution in Hardware

Tushar Krishna
Assistant Professor
ON Semiconductor Professor
School of ECE
Georgia Institute of Technology

The Dream!



General-Purpose AI capable of emulating Human-like Continuous Learning

What is Continuous Learning?

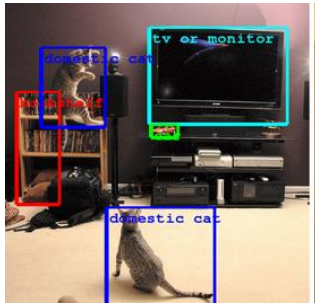
Become better and faster with experience

Learn new tasks

Autonomous

Deep Learning Applications

Object Detection



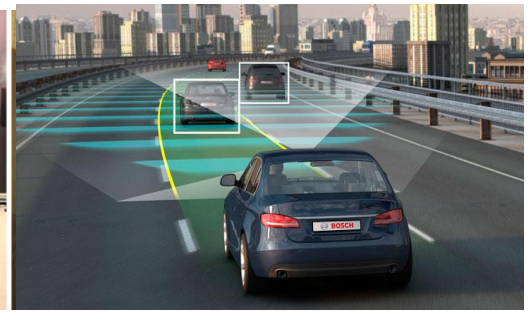
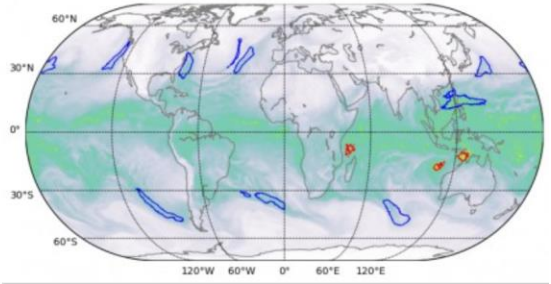
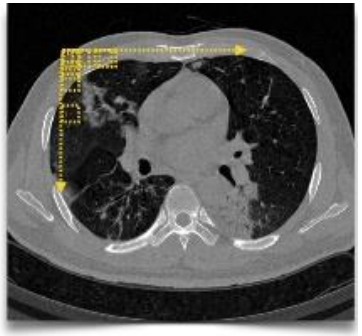


Image Segmentation

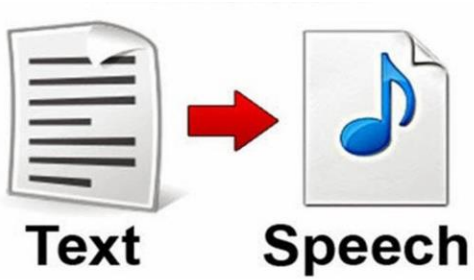


Medical Imaging



Speech Recognition

Text to Speech



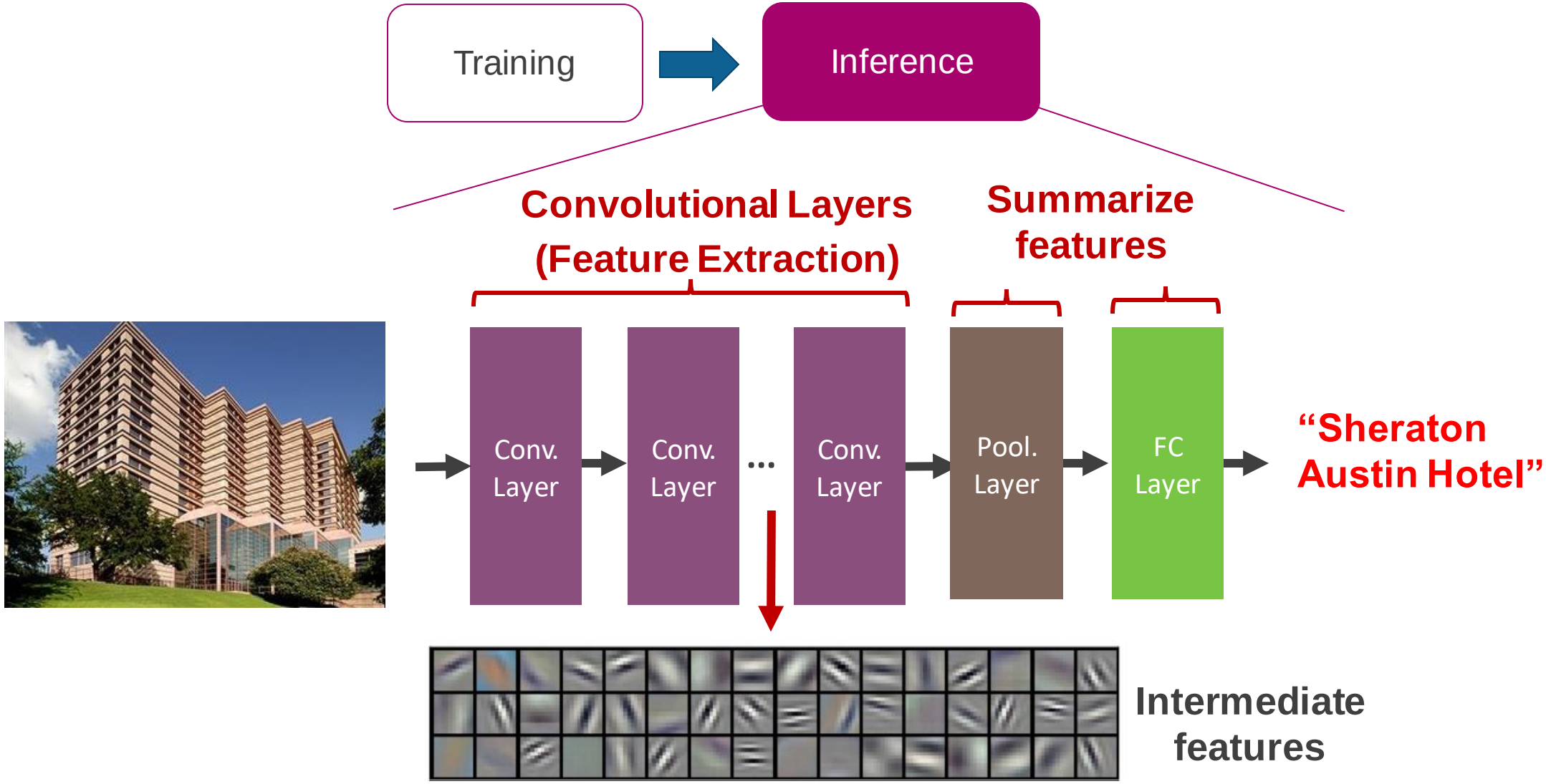
Recommendations

Games

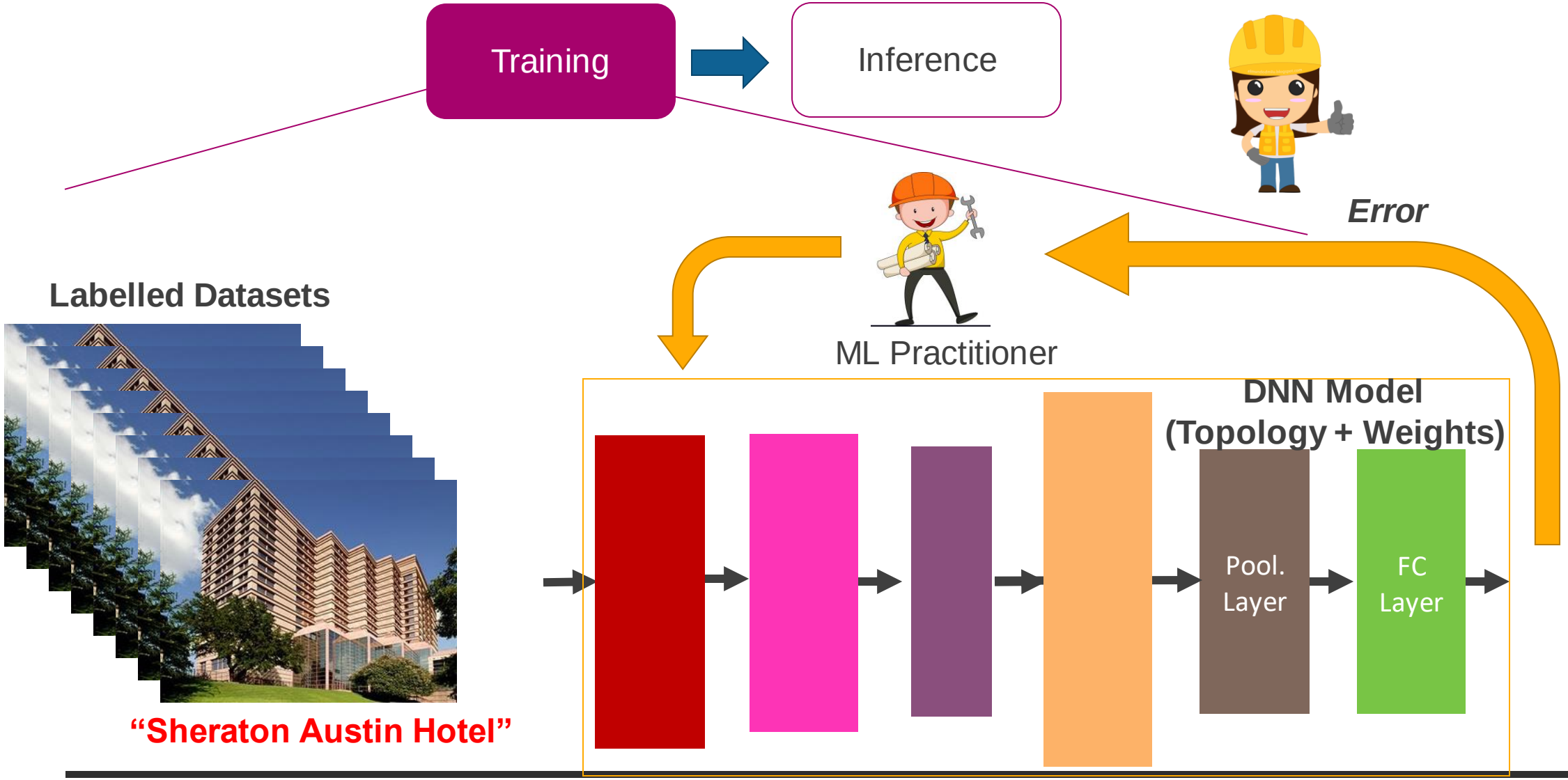
Deep Learning Landscape

Training → Inference

Deep Learning Landscape



Deep Learning Landscape



Computation Platforms

Training → Inference

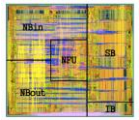
HPC cluster

ARM Trillum

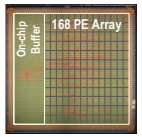
NVDLA

Apple Neural Engine

CambriconX



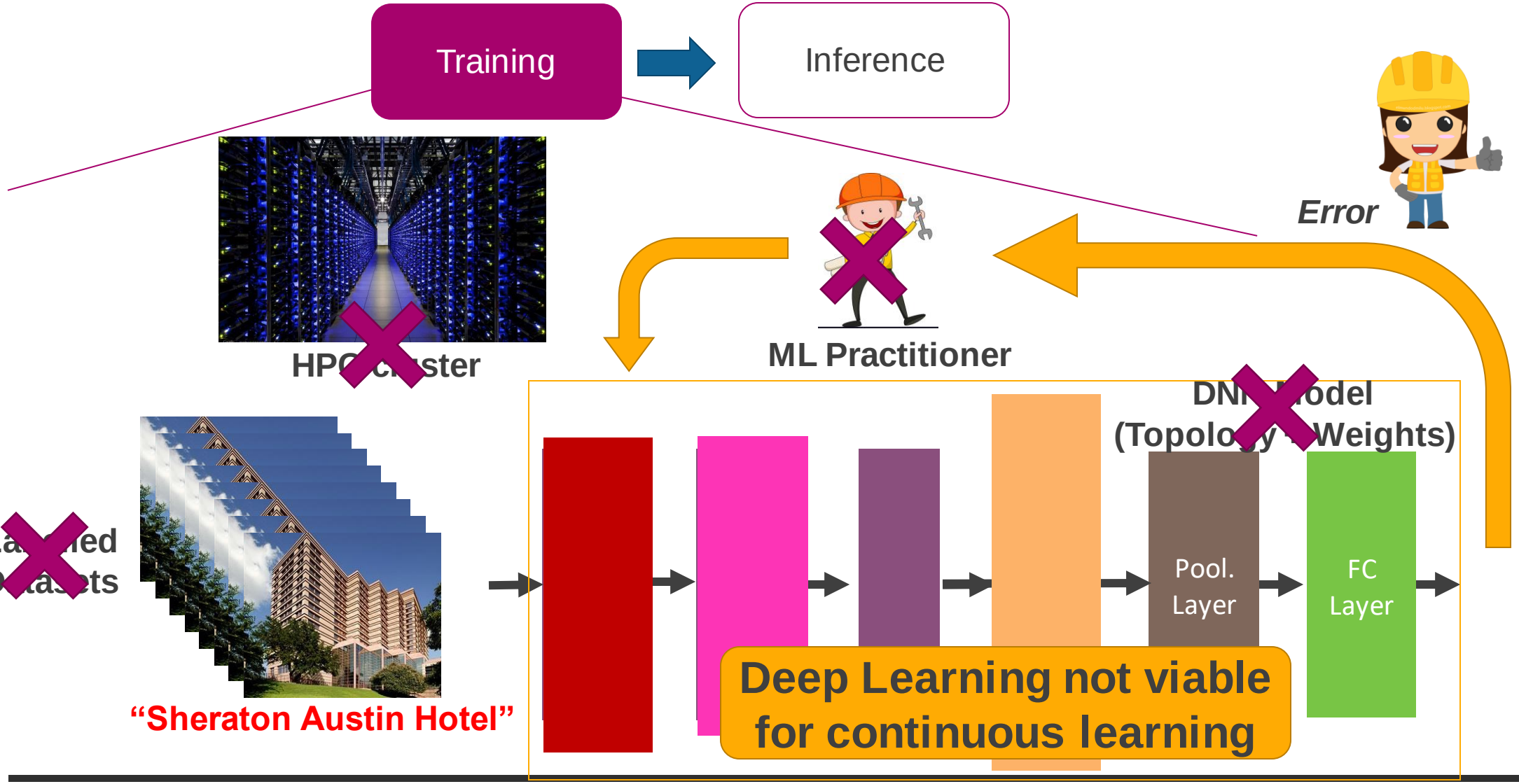
ShiDianNao



Eyeriss

Accelerators

Deep Learning → Continuous Learning?



Challenges for Continuous Learning

- **Algorithm Challenge**
  - No labeled datasets
  - No human-in-the-loop
  - Continuously Changing DNN Model

- **System Challenge**
  - On-Device Learning => Low-energy Hardware Platform

Candidates for Continuous Learning

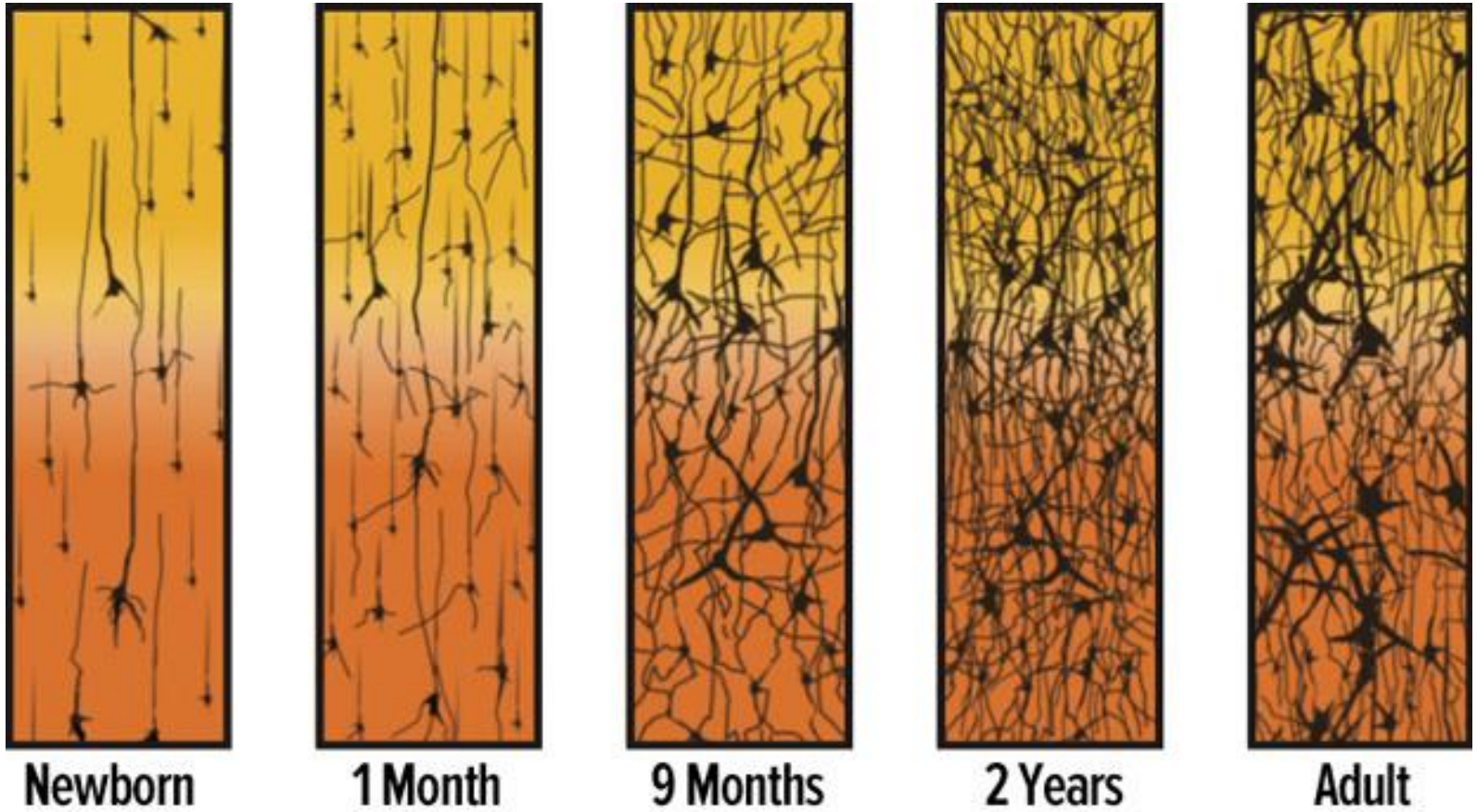
	Data	Hyper-param Tuning	DNN Plasticity	Compute	Memory
Supervised DL	Labeled X	Manual X	Designed for one problem X	Backprop (gradients) X	Backprop (gradients) X

Outline of Talk

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

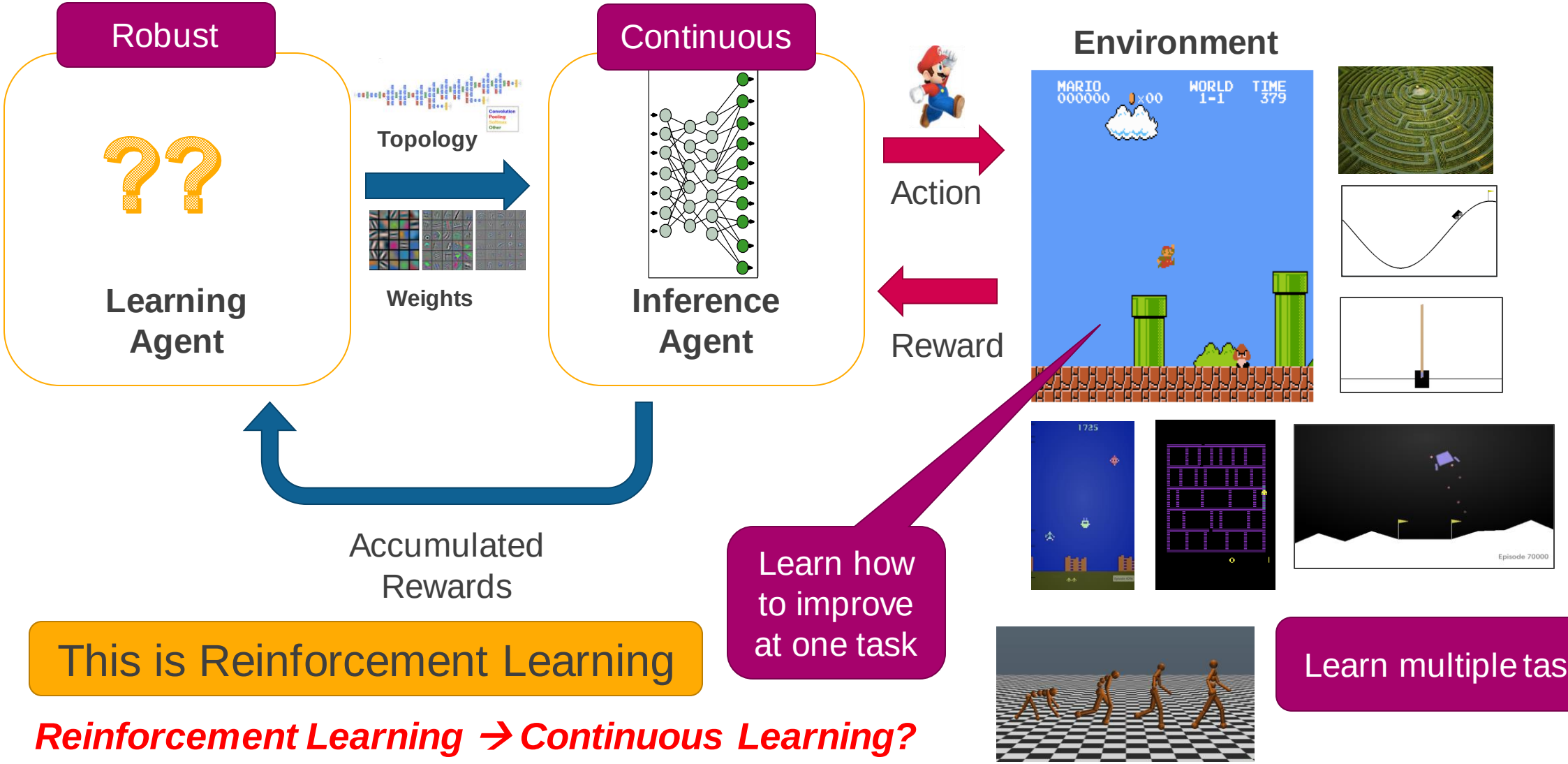
Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna,
GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware,
In Proc of 51st Annual IEEE/ACM International Symposium on Microarchitecture(MICRO), 2018

Continuous Learning in Brains



Constant synapse formation and pruning

Template for Continuous Learning



This is Reinforcement Learning

Reinforcement Learning → Continuous Learning?

Candidates for Continuous Learning

	Data	Hyper-param Tuning	DNN Plasticity	Compute	Memory
Supervised DL	Labeled ✗	Manual ✗	Designed for one problem ✗	Backprop (gradients) ✗	Backprop (gradients) ✗
Reinforcement Learning	Unlabeled ✓	Manual ✗	Reward Function ✓	Backprop (gradients) ✗	Backprop (gradients) ✗

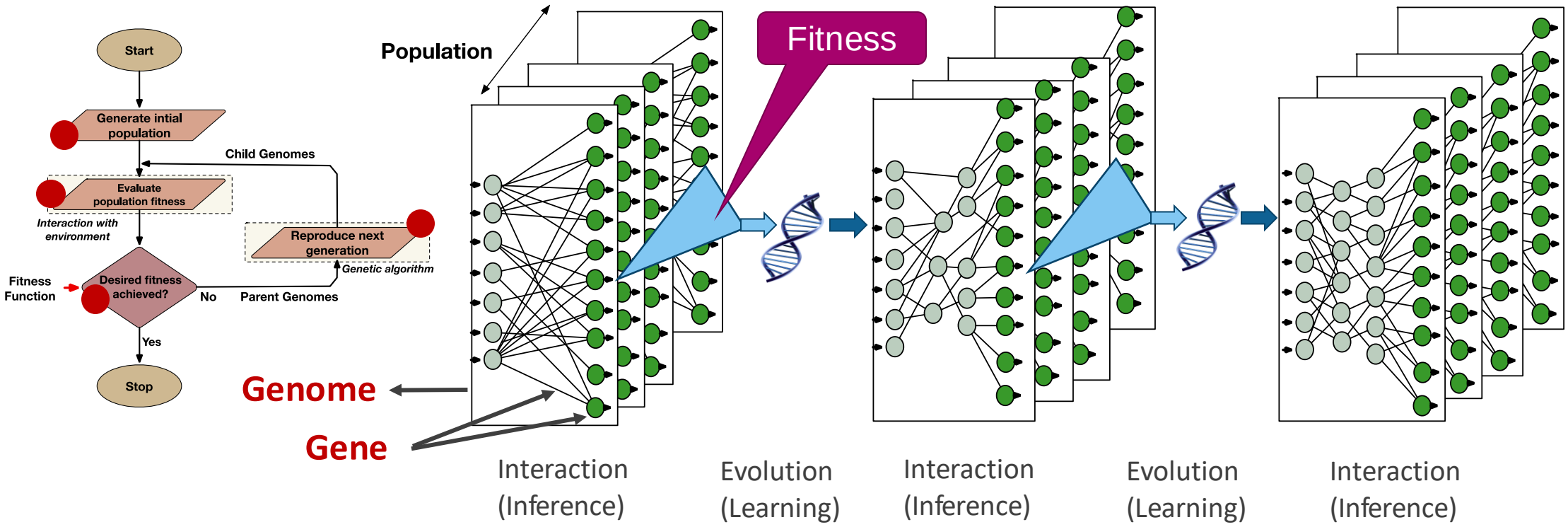
RL is not viable for continuous learning on the edge

Outline of Talk

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

Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna,
GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware,
In Proc of 51st Annual IEEE/ACM International Symposium on Microarchitecture(MICRO), 2018

Neuro-Evolutionary (NE) Algorithm



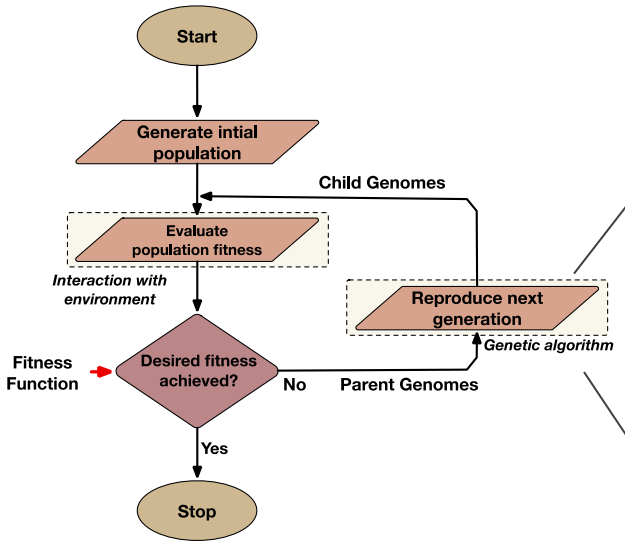
Neural Network (NN) expressed as a graph

Gene: Vertex or Edge in the graph

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

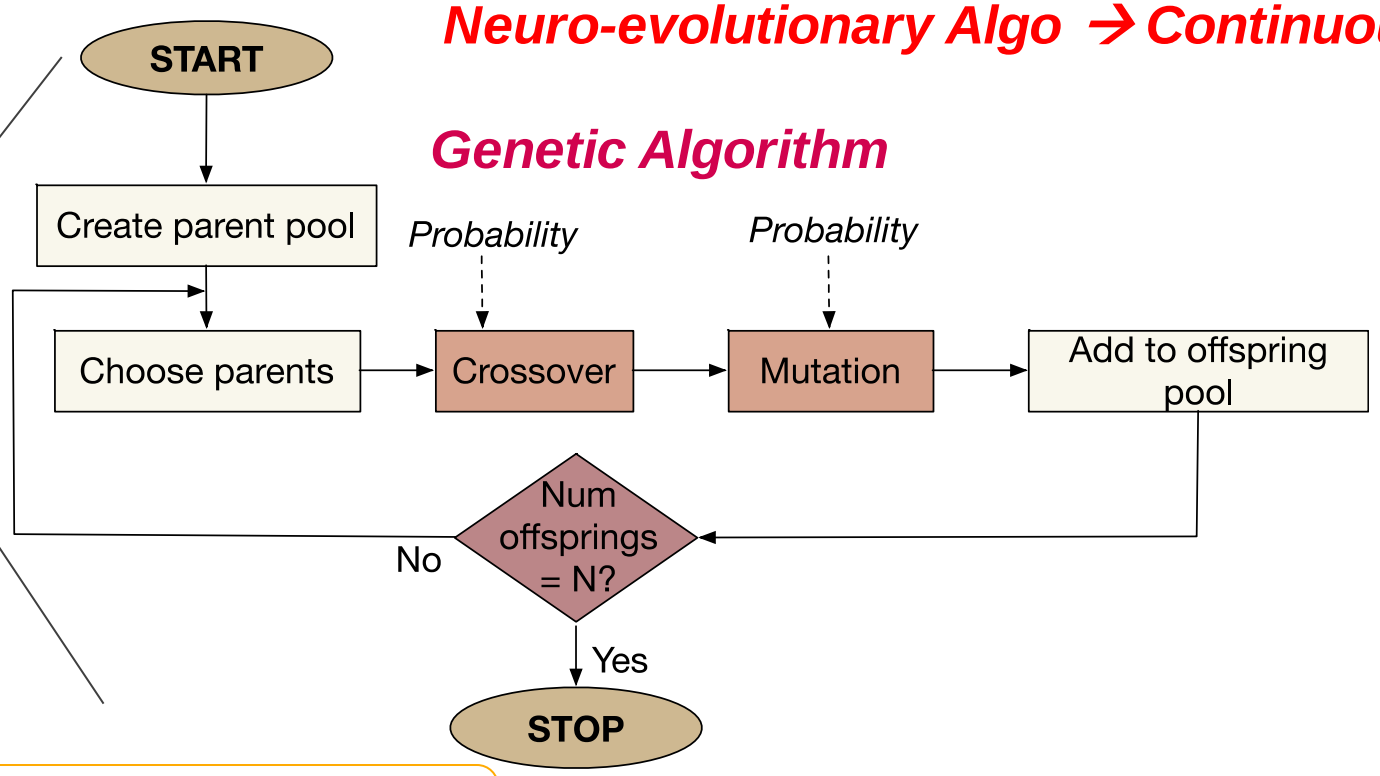
[1] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, *10*(2), 99-127.

Neuro-Evolutionary (NE) Algorithm



Neuro-evolutionary Algo → Continuous Learning?

Genetic Algorithm



Neural Network (NN) expressed as a graph

Gene: Vertex or Edge in the graph

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

NeuroEvolution of Augmented Topologies (NEAT) [1]

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

Candidates for Continuous Learning

	Data	Hyper-param Tuning	DNN Plasticity	Compute	Memory
Supervised DL	Labeled ✗	Manual ✗	Designed for one problem ✗	Backprop (gradients) ✗	Backprop (gradients) ✗
Reinforcement Learning	Unlabeled ✓	Manual ✗	Reward Function ✓	Backprop (gradients) ✗	Backprop (gradients) ✗
Evolutionary	Unlabeled ✓	Automated ✓	Reward Function ✓	??	??

Outline of Talk

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

Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna,
GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware,
In Proc of 51st Annual IEEE/ACM International Symposium on Microarchitecture(MICRO), 2018

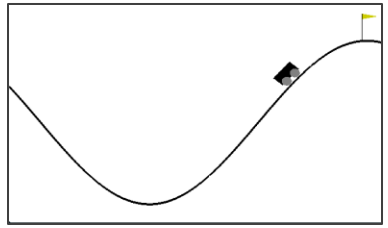
Characterization of NEAT

NEAT - Python

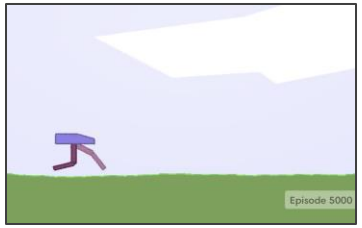
Codebase

OpenAI Gym

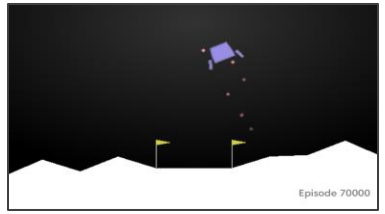
Environments



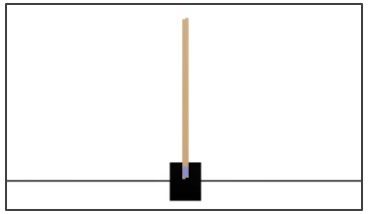
Mountain car



Bipedal



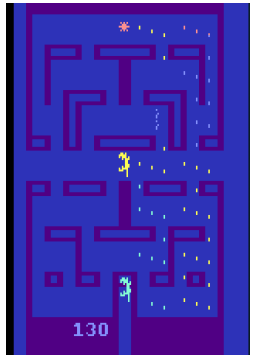
Lunar Lander



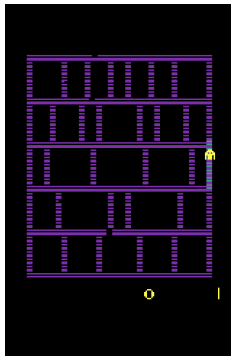
Cart pole

Ran each environment till convergence, multiple times

Airraid-RAM



Alien-RAM



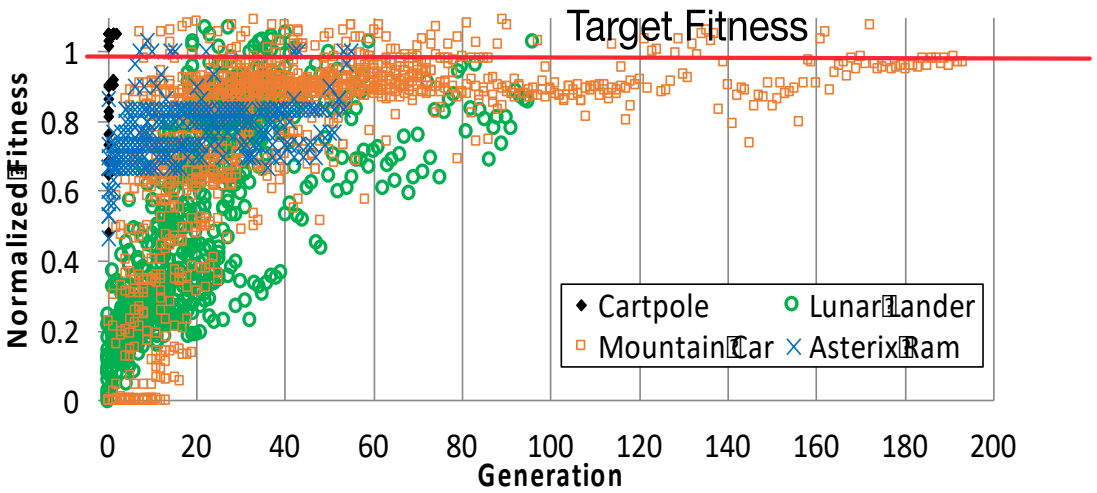
Amidar-RAM

Only changed fitness function between workloads

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

Characterization of NEAT

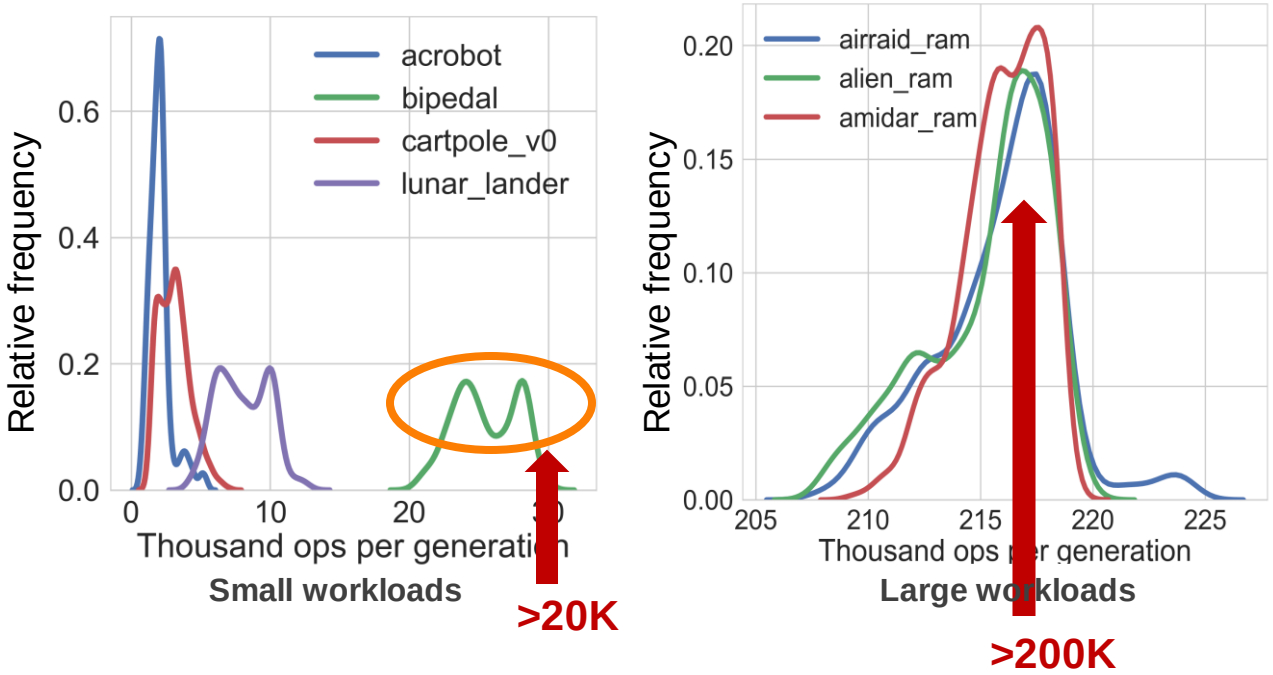
Computations



Inference:
Population level parallelism (PLP)

Evolution:
Gene level parallelism (GLP)

Distribution of Operations/Generation



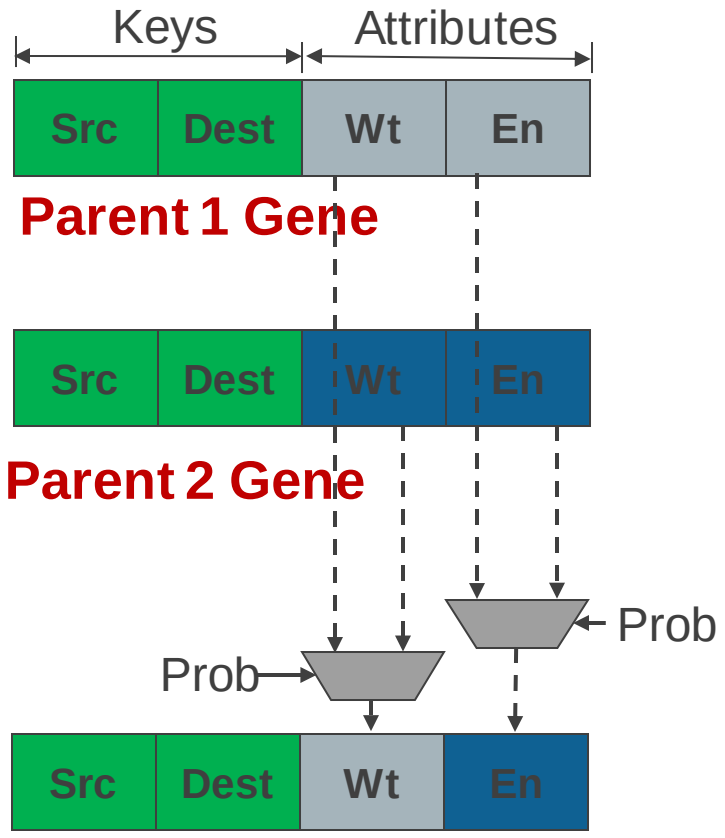
All operations are independent

Large operation level Parallelism

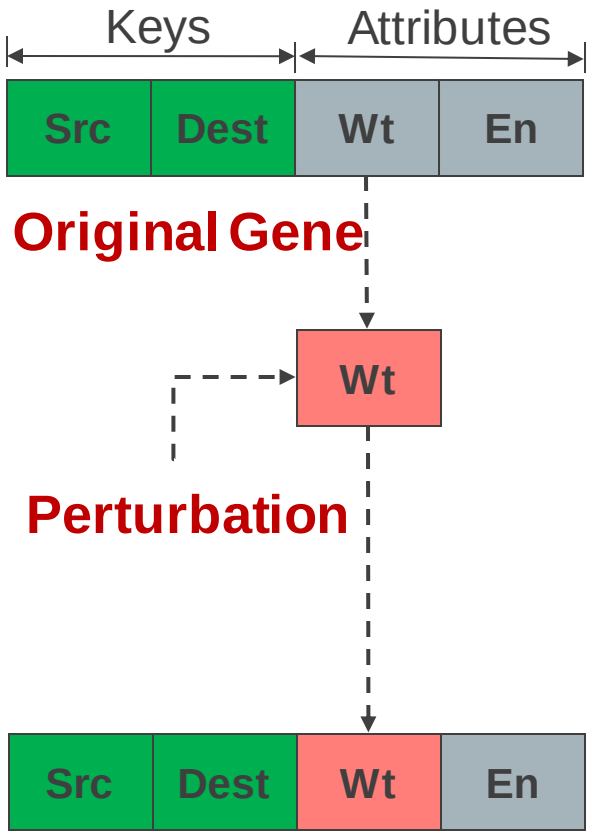
Operations in NEAT

Evolution

Crossover



Mutation



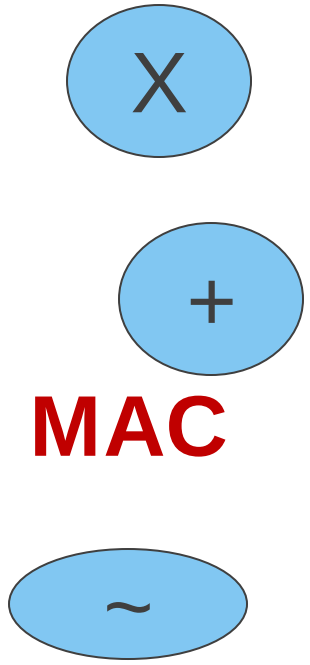
Addition mutation

- *Add new node*
- *Add new connection*

Deletion mutation

- *Delete connection*
- *Delete node*

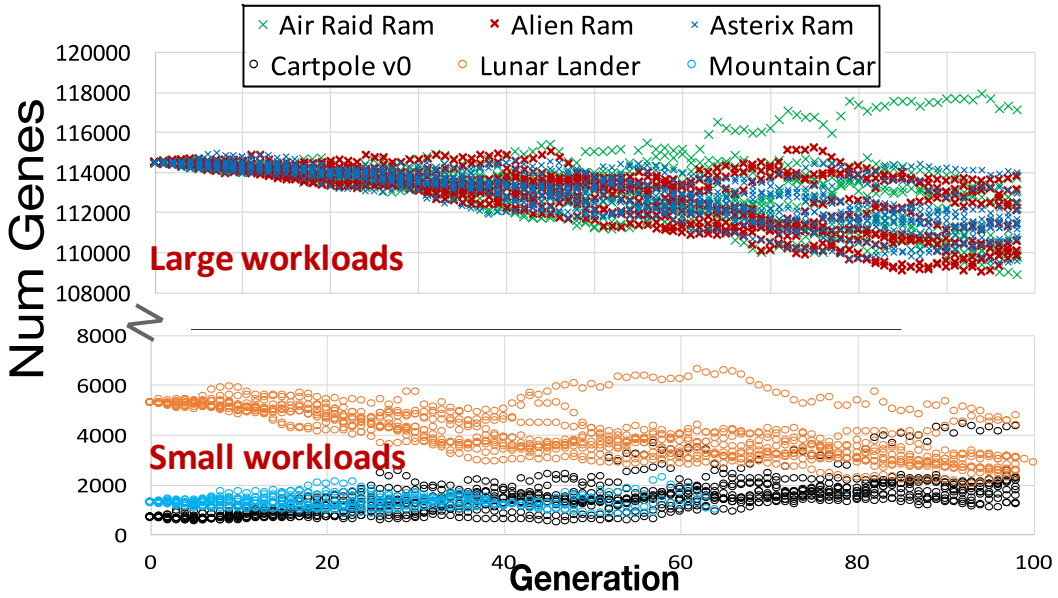
Inference

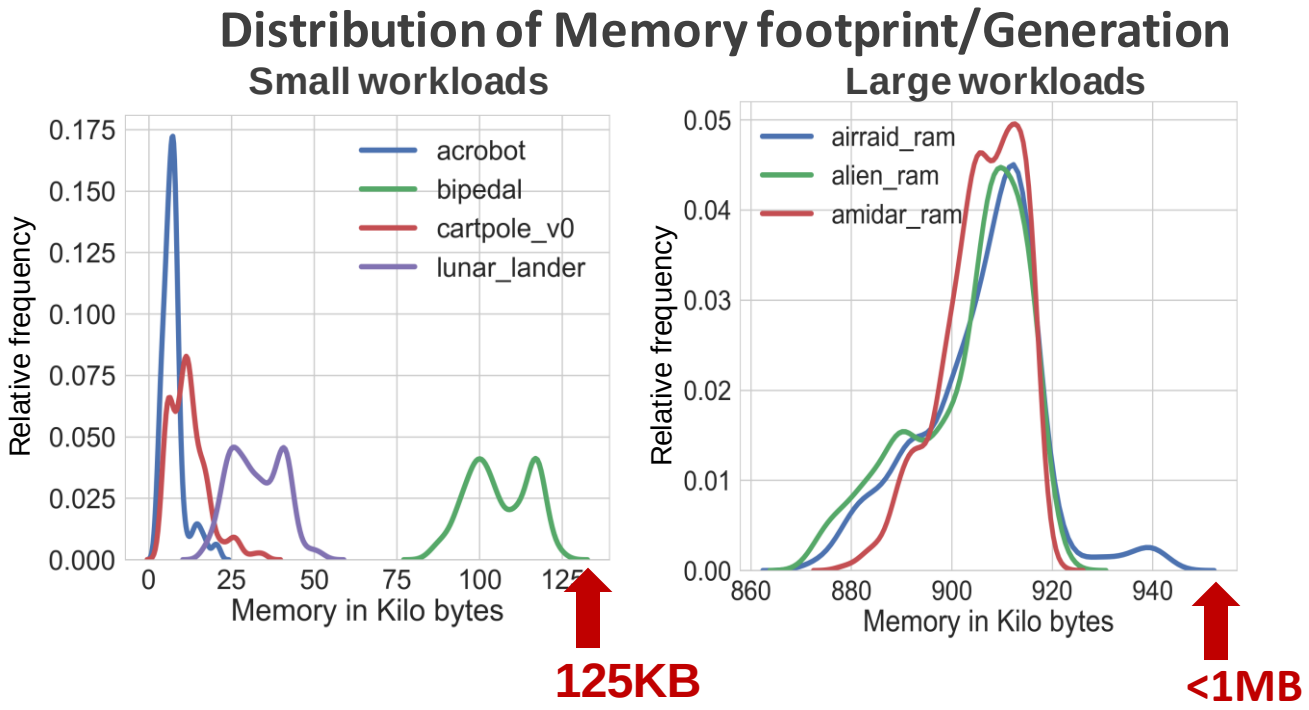


Simple operations

Characterization of NEAT

Memory





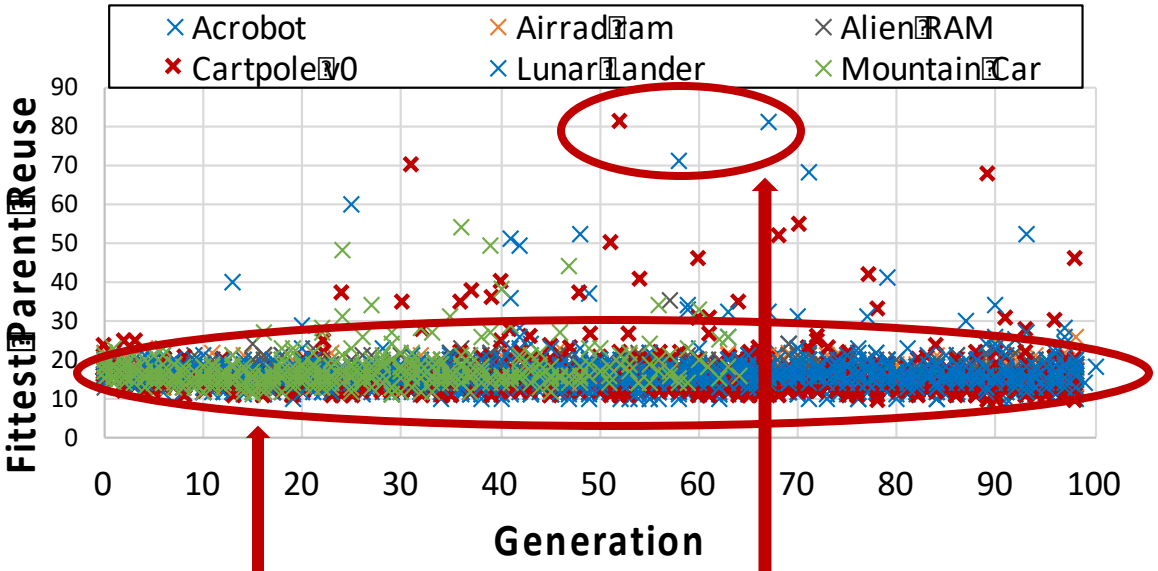
Entire population can fit on-chip

Only need to store the weights and node info

Characterization of NEAT

Memory

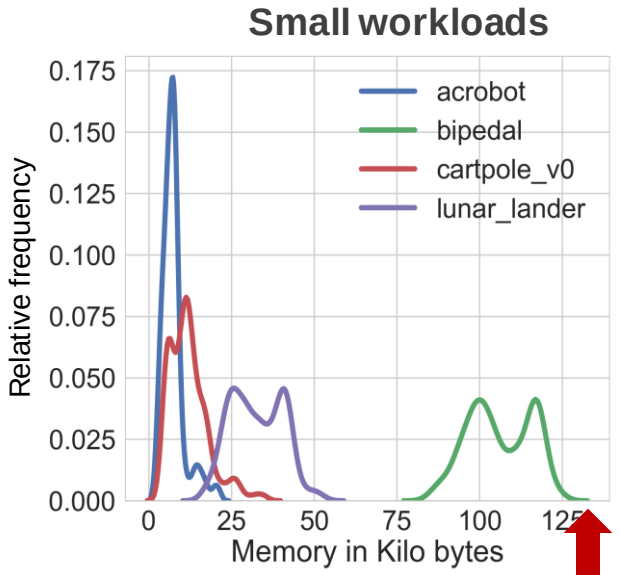
Opportunity for Reuse



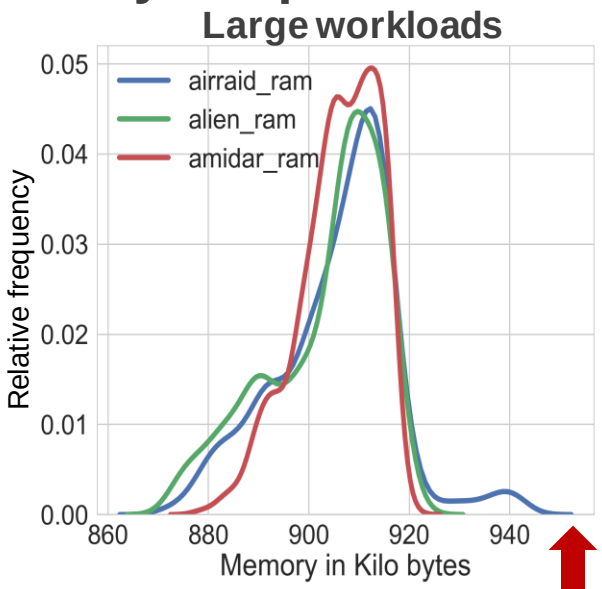
Fittest parent genome is used about ~10-20 times each generation

Even higher in certain cases

Distribution of Memory footprint/Generation



125KB



<1MB

Entire population can fit on-chip

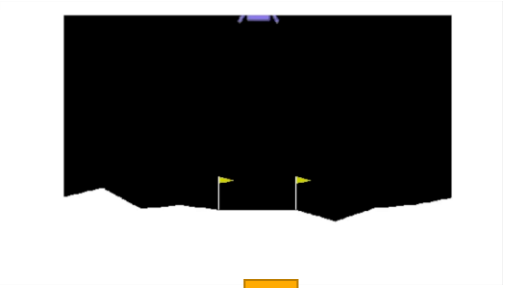
Only need to store the weights and node info

Properties of NE algorithms

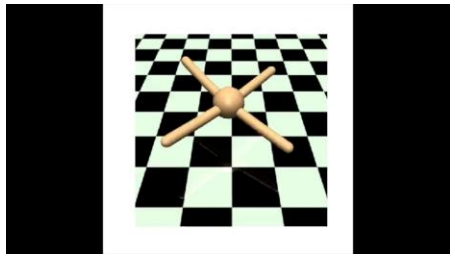
Algorithmic

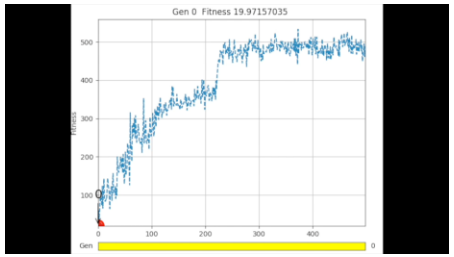
No Explicit Training

No Manual Tuning



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

Candidates for Continuous Learning

	Data	Hyper-param Tuning	DNN Plasticity	Compute	Memory
Supervised DL	Labeled ✗	Manual ✗	Designed for one problem ✗	Backprop (gradients) ✗	Backprop (gradients) ✗
Reinforcement Learning	Unlabeled ✓	Manual ✗	Reward Function ✓	Backprop (gradients) ✗	Backprop (gradients) ✗
Evolutionary	Unlabeled ✓	Automated ✓	Reward Function ✓	Massive Parallelism ✓	Only store model ✓

Motivation for HW-SW Co-Design

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

Gene and Population level parallelism

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

Simple HW friendly operations

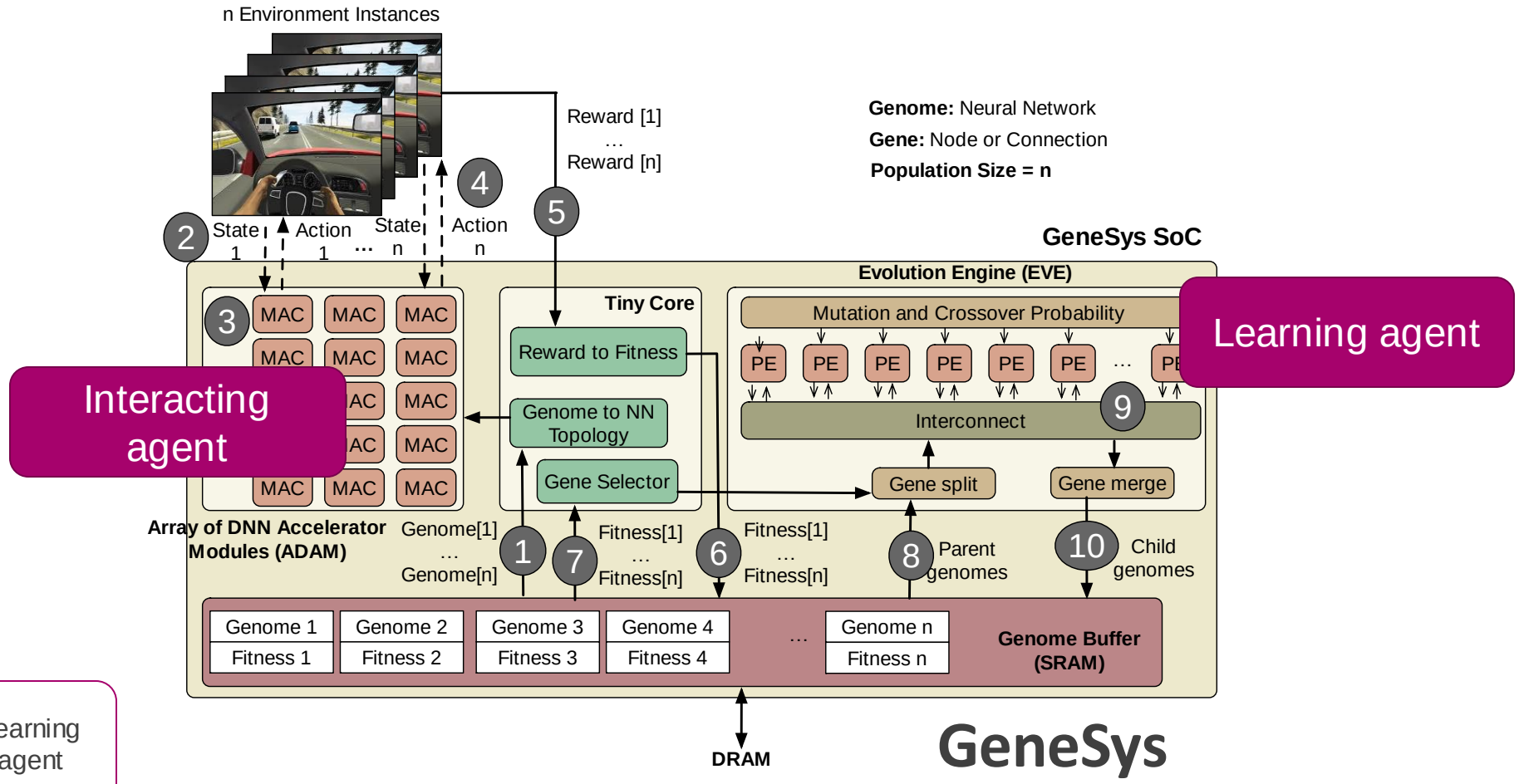
Hardware-Software co-design of NE makes them viable for continuous learning

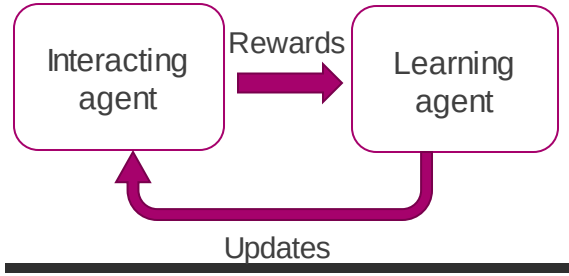
Outline of Talk

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

Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna,
GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware,
In Proc of 51st Annual IEEE/ACM International Symposium on Microarchitecture(MICRO), 2018

GeneSys SoC

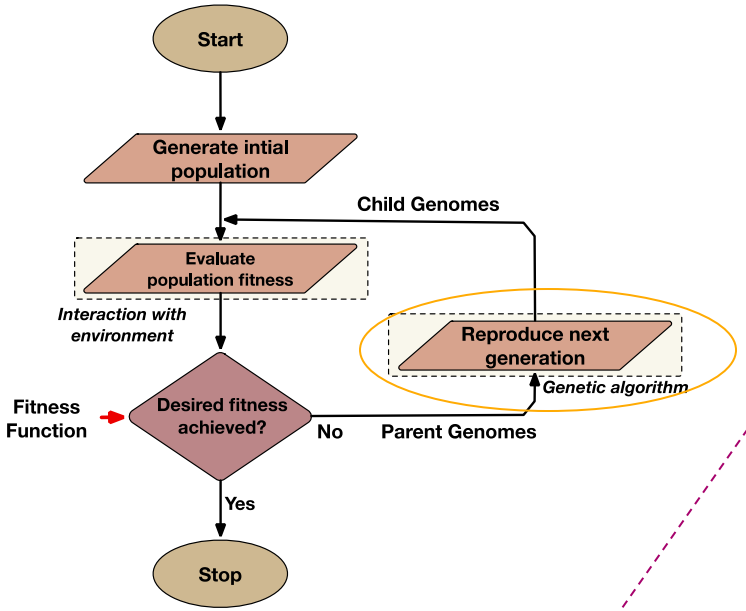


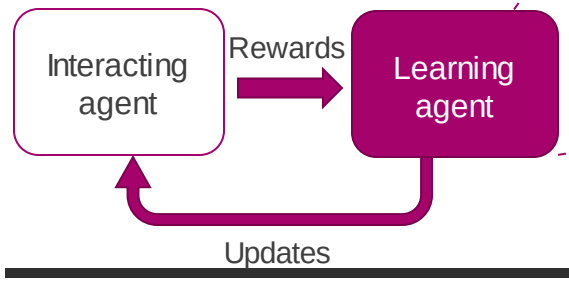


Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna, ***GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware****, MICRO 2018*

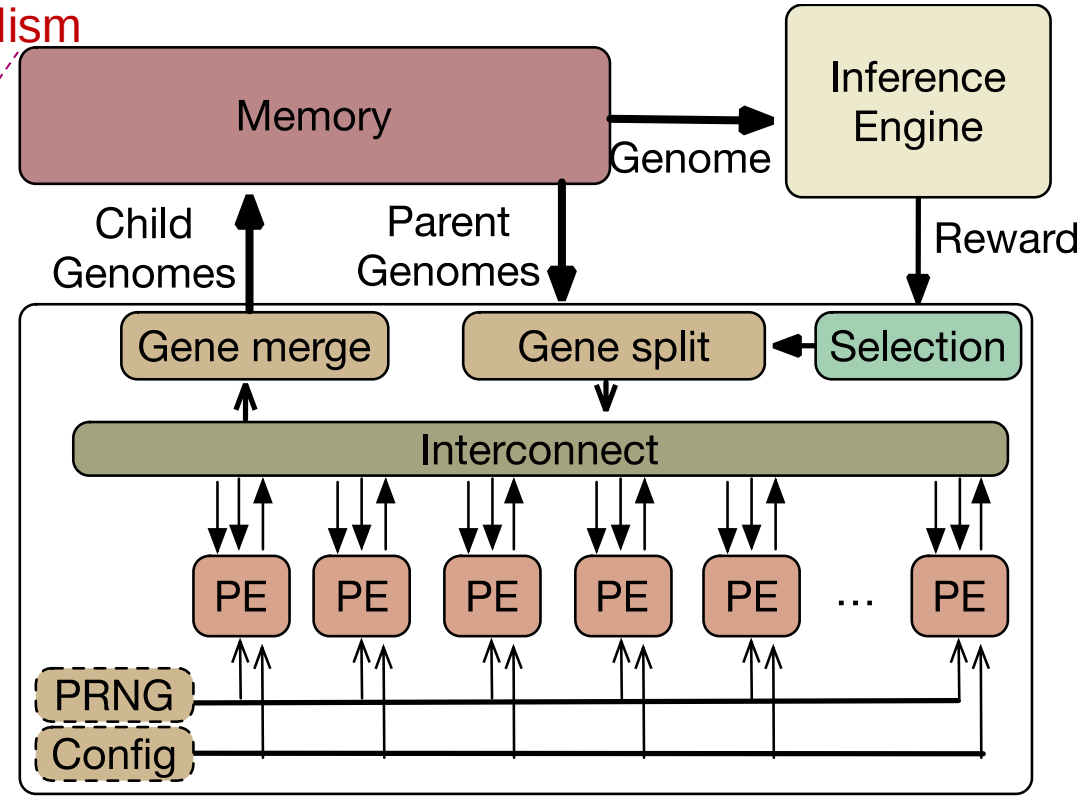
Evolution Engine: EvE Microarchitecture

GeneSys



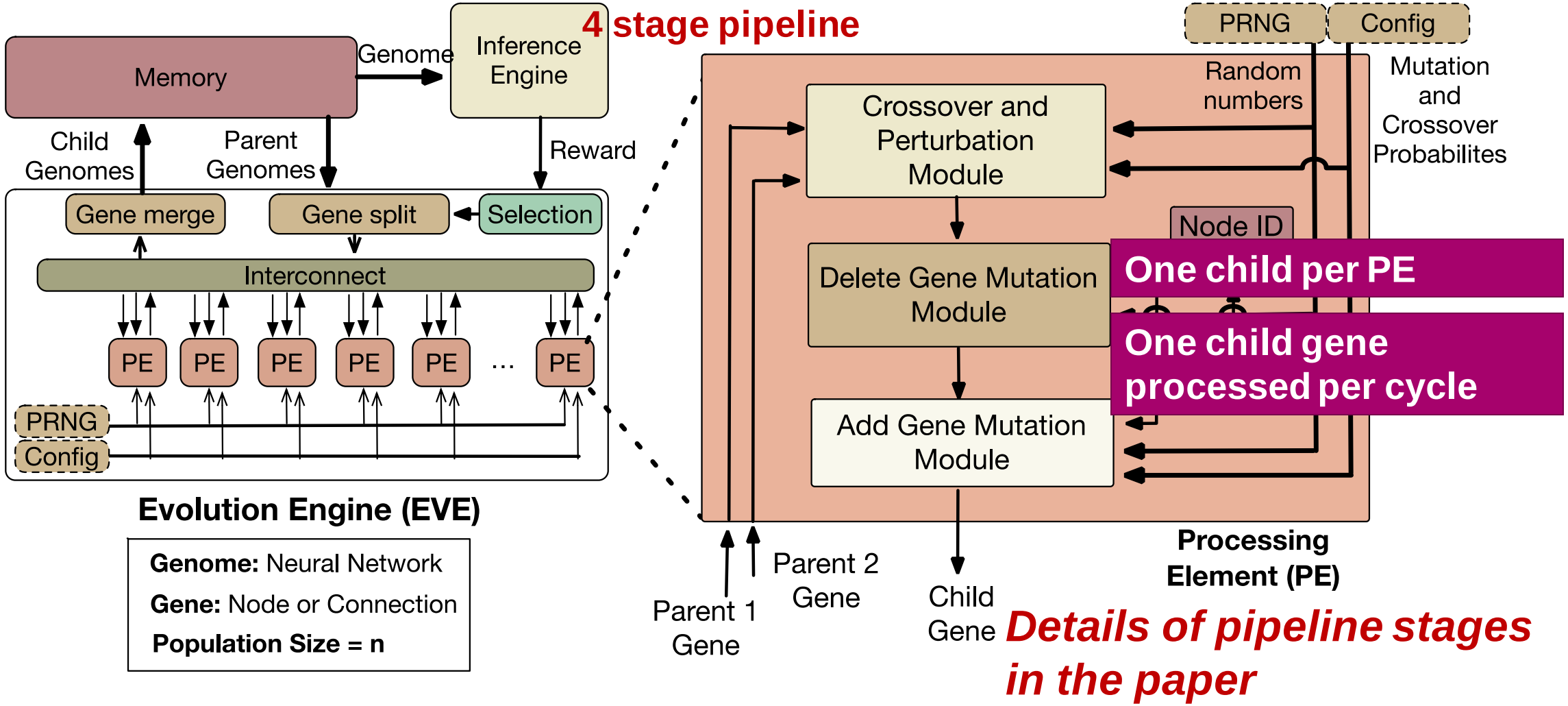


Large number of PE to exploit parallelism

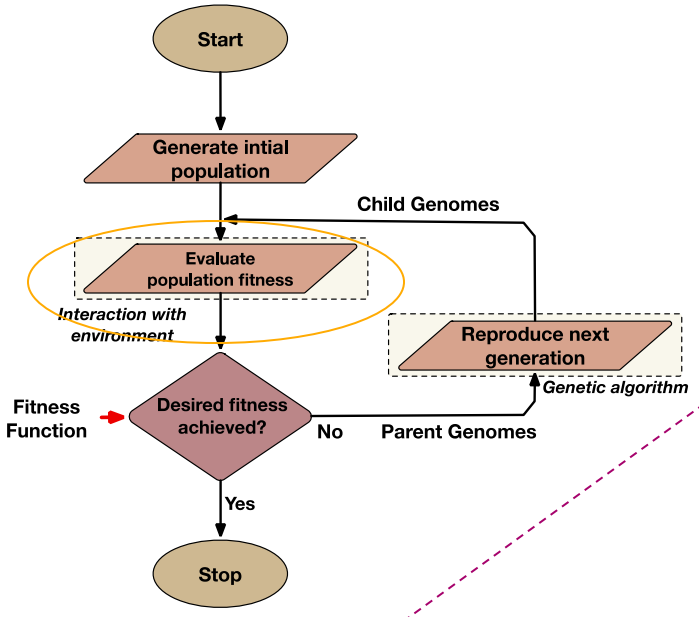


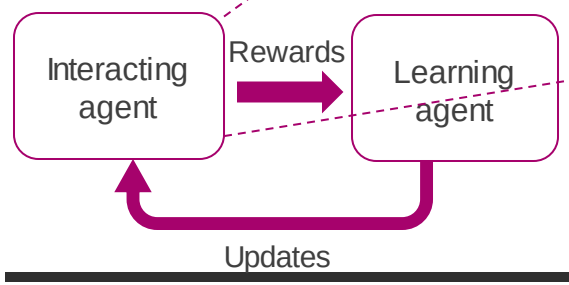
Evolution Engine (EVE)

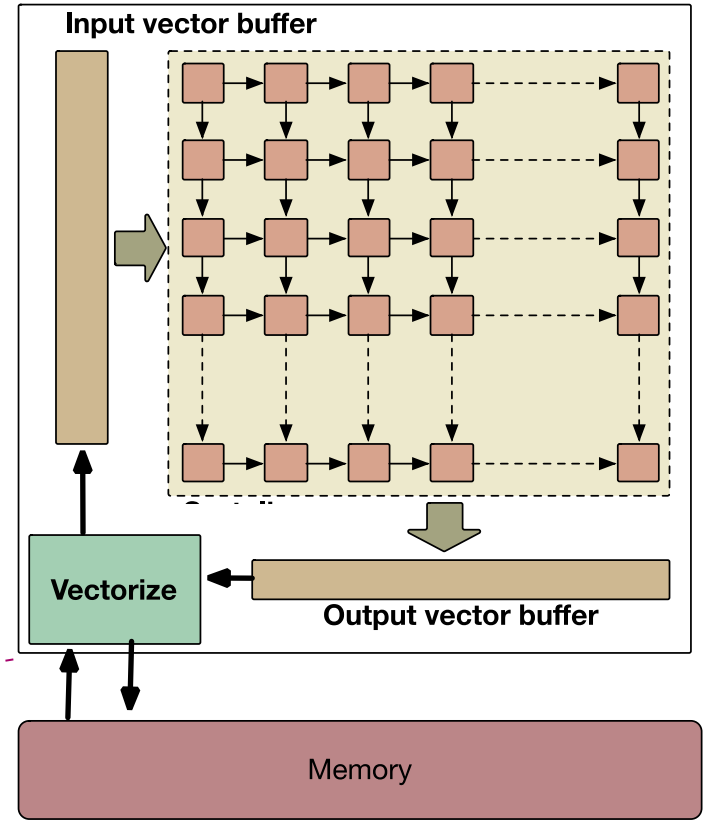
PE Microarchitecture



Inference Engine: ADAM Microarchitecture





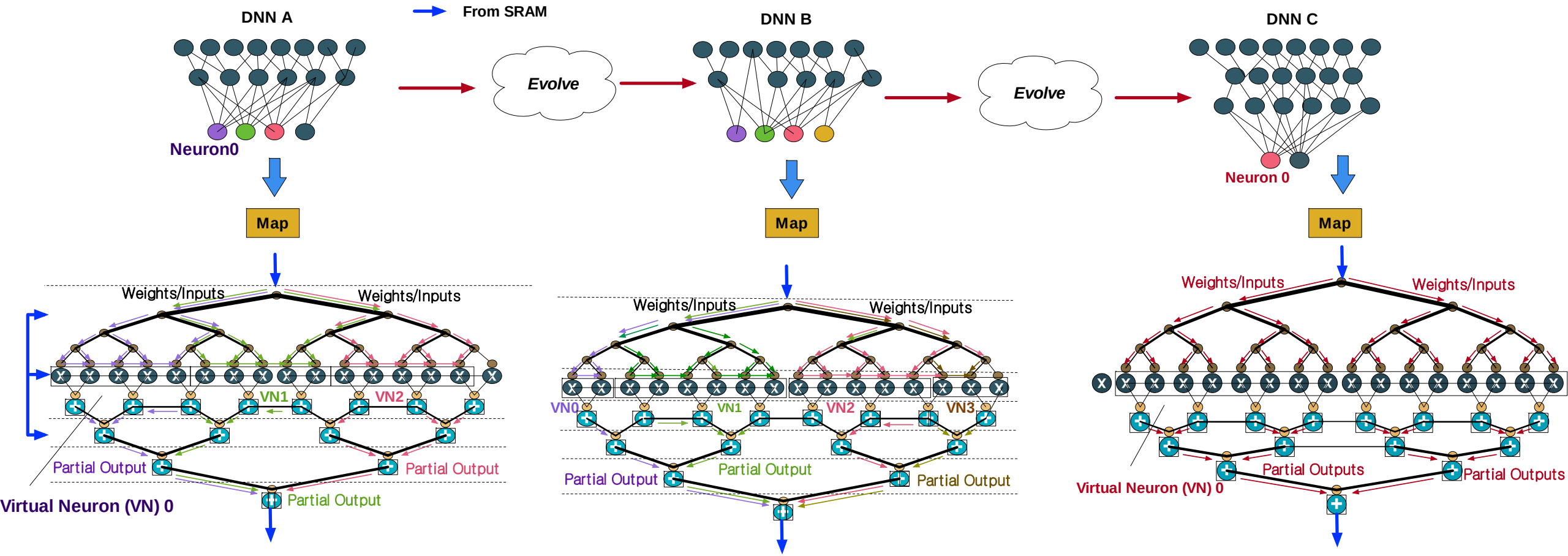


Conventional DNN Inference Accelerator

Exploit Population Level Parallelism

Networks generated by NEAT are irregular (thus sparse)

Mapping Evolving DNNs over ADAM



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***

Outline of Talk

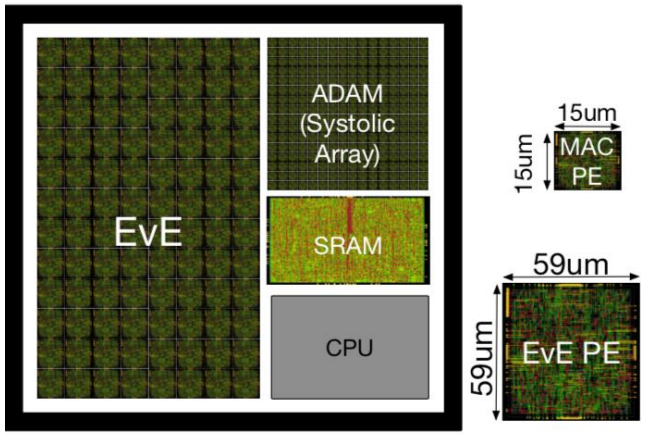
- Continuous Learning Template
- Neuro-Evolutionary Algorithms
  - Algorithm Description
  - Characterizing NEAT
- Microarchitecture
- Evaluations

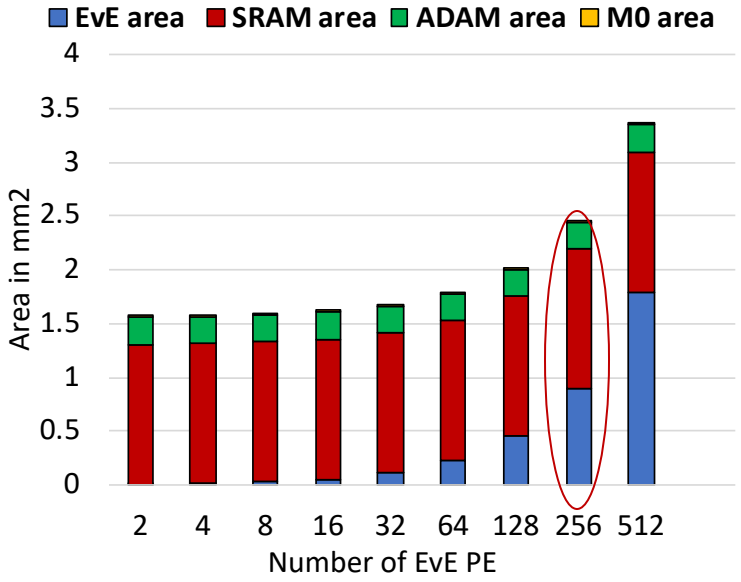
Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna,
GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware,
In Proc of 51st Annual IEEE/ACM International Symposium on Microarchitecture(MICRO), 2018

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



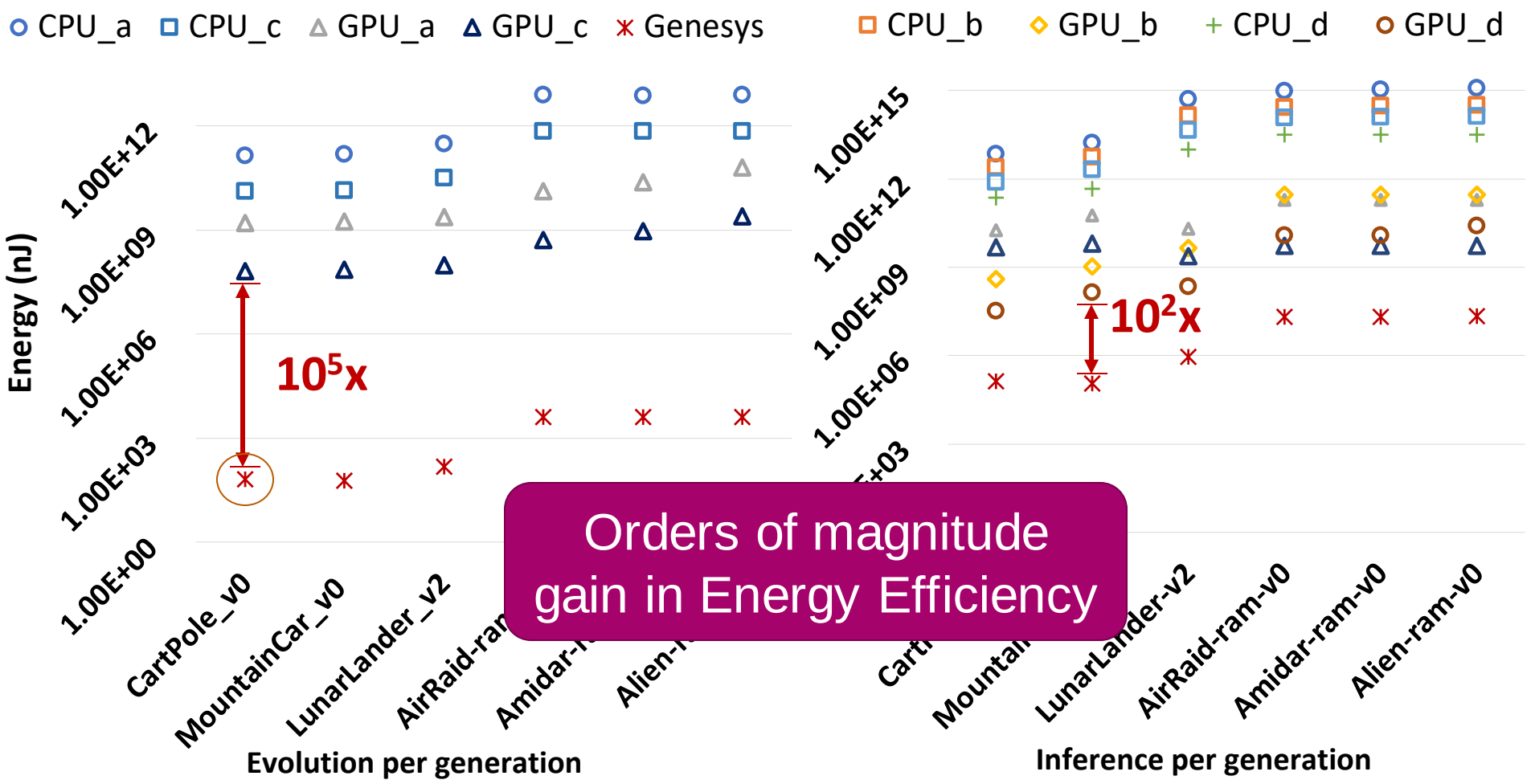


Evaluations

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

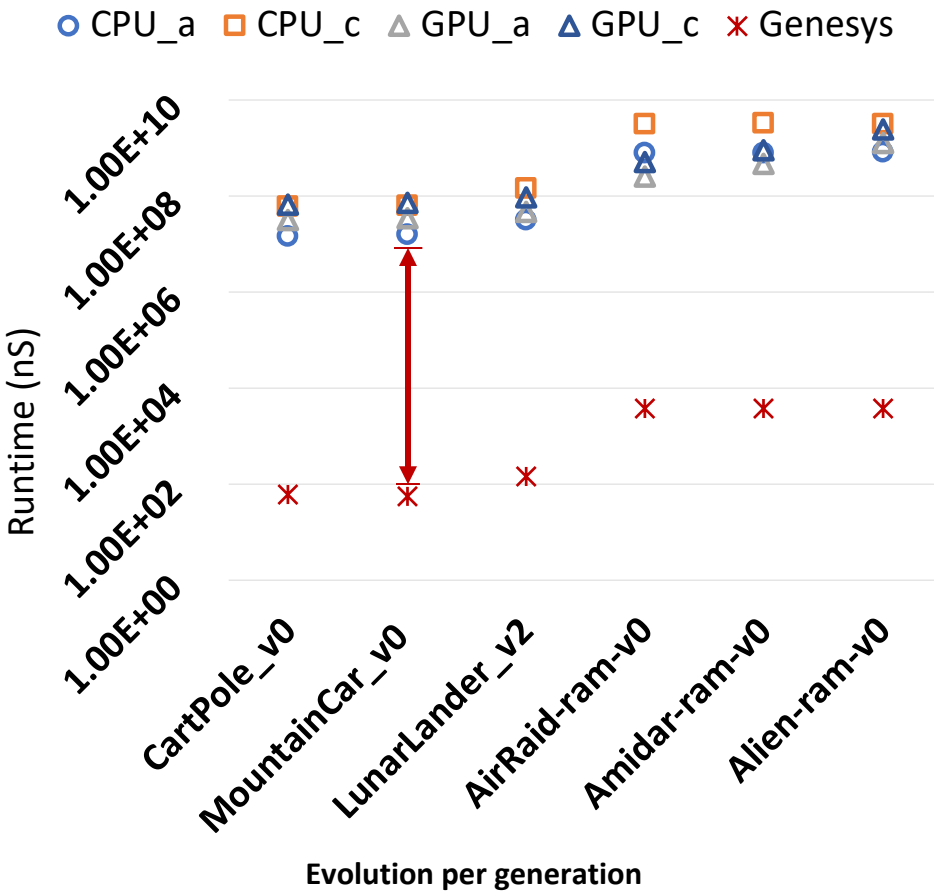
PLP (GLP) - Population (Gene) Level Parallelism
BSP - Bulk Synchronous Parallelism (GPU)

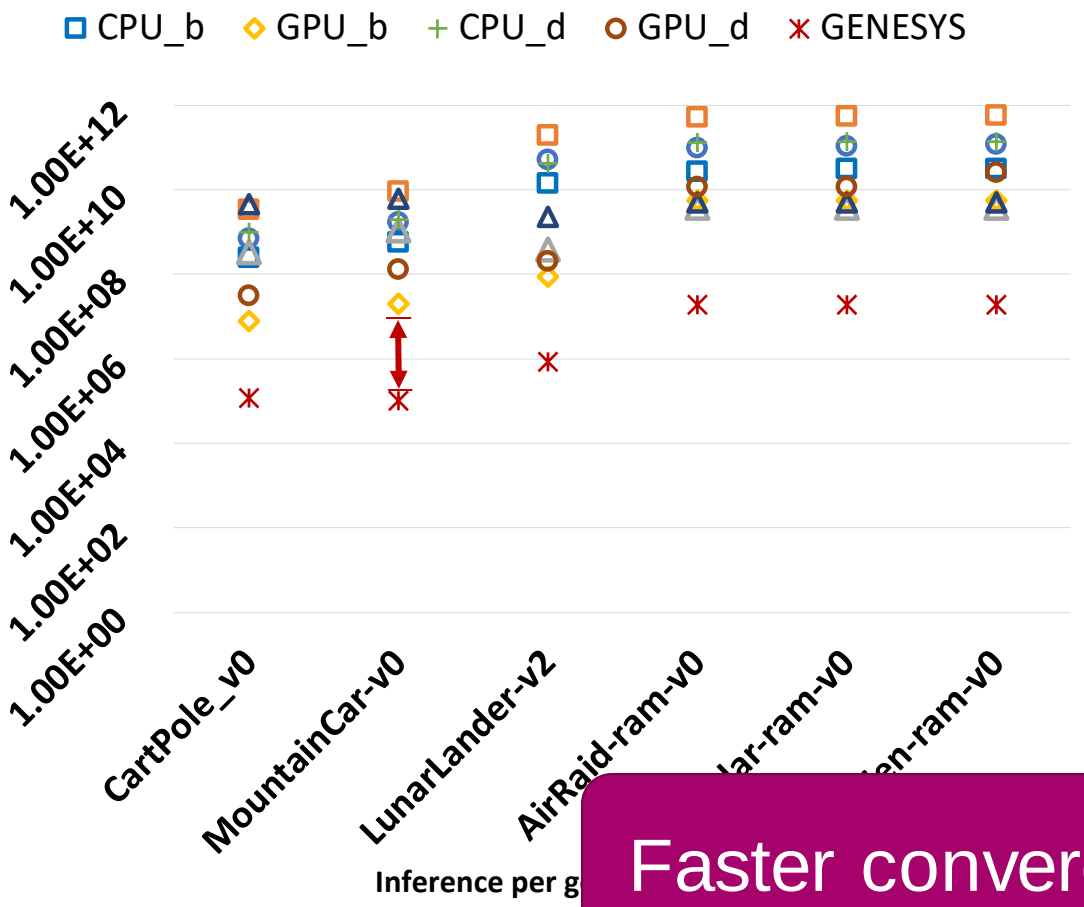
Evaluations: Energy



Orders of magnitude gain in Energy Efficiency

Evaluations: Runtime



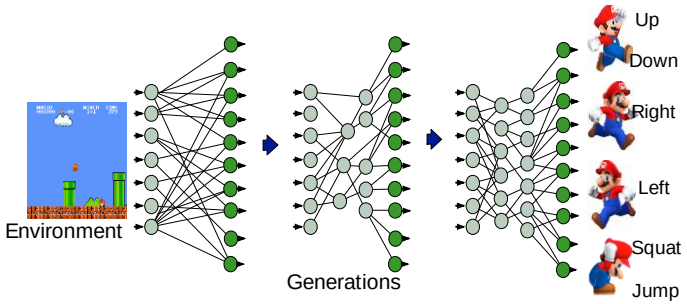


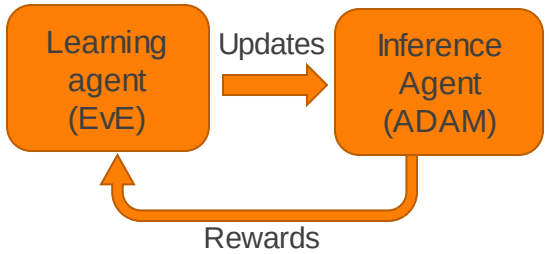
Faster convergence

Conclusion

- **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

Thank you!

http://synergy.ece.gatech.edu