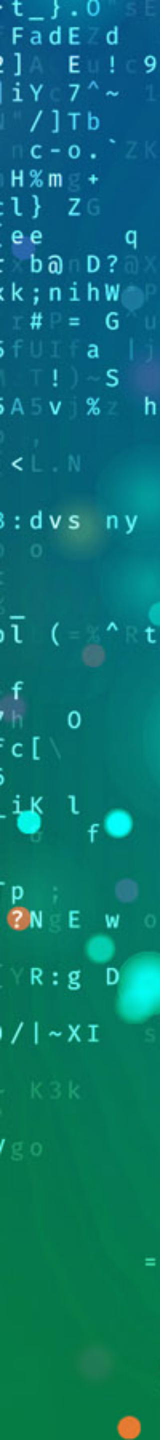


Pacific Northwest
NATIONAL LABORATORY

Reinforcement Learning for Controls

Malachi Schram
Data Science Architecture and AI Lead

On behalf of ExaLearn Control Pillar Team

and the Controls for HEP Accelerators project

U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy

Talking Points

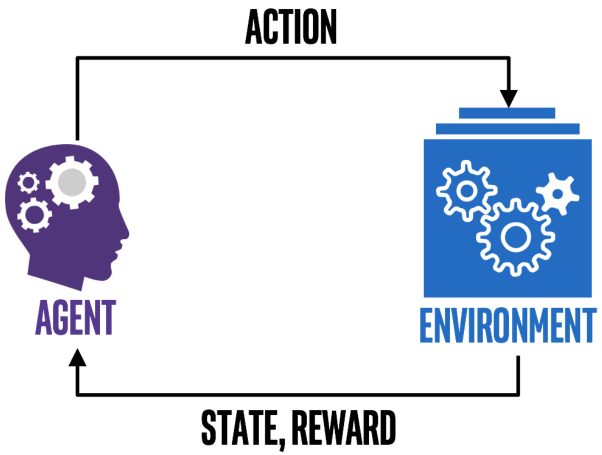
Controls for HEP accelerator

Control Application Pillar in ExaLearn

Pacific Northwest
NATIONAL LABORATORY

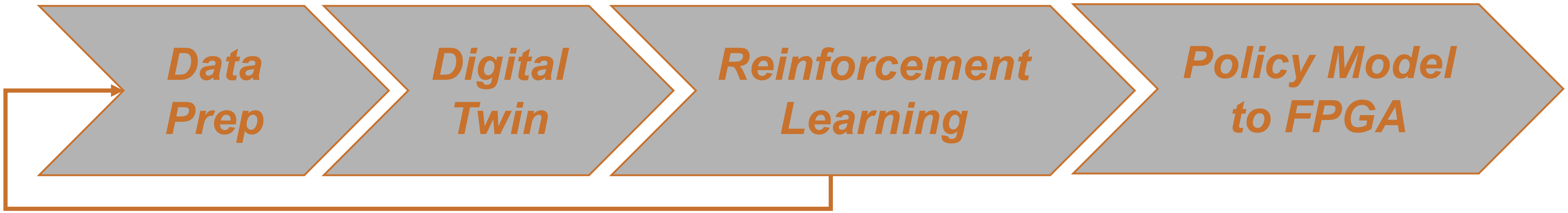
Reinforcement Learning

- “Reinforcement learning is learning what to do — how to map situations to actions—so as to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them.” - Barton & Sutton
- Key concepts to reinforcement learning:
  - Agent (controller – policy and sampling)
    - Action (control signal)
  - Environment (controlled system)
    - State (representation of environment)
    - Reward (numerical consequence of action)
- Sequence of experience and agent forms trajectory: S_0, A_0, R_0, S_1, A_1, R_1, …



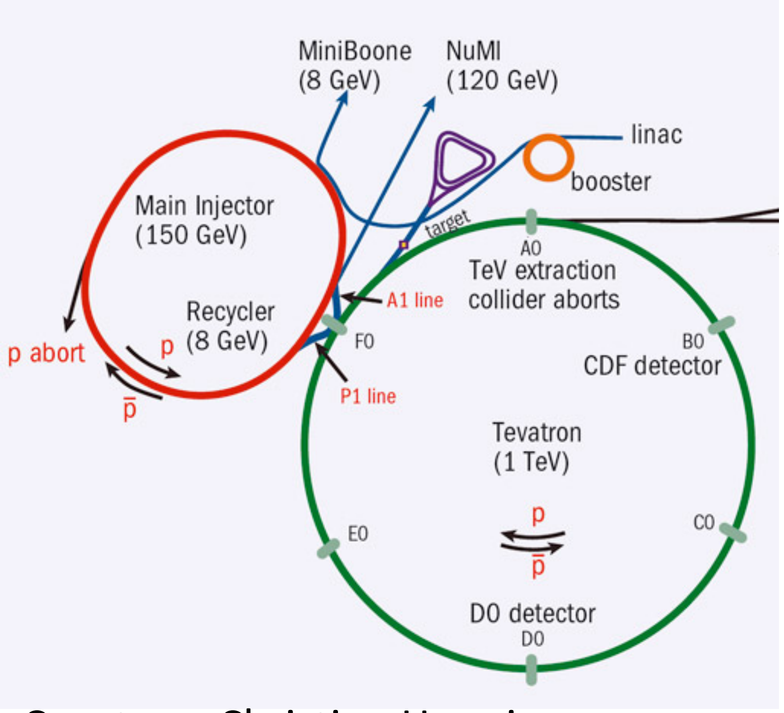
Controls for FNAL booster

- **Goals:** The goal is to reduce beam losses in the FNAL Booster by developing a machine learning (ML) model that provides optimal set of actions for accelerator controls:
  - Clean/prune data, resample, correlation analysis, etc.
  - Create digital twin of accelerator using historical and targeted data
  - Create reinforcement learning (RL) workflow
  - ML algorithm on a custom FPGA board to control the magnet power supplies (GMPS)

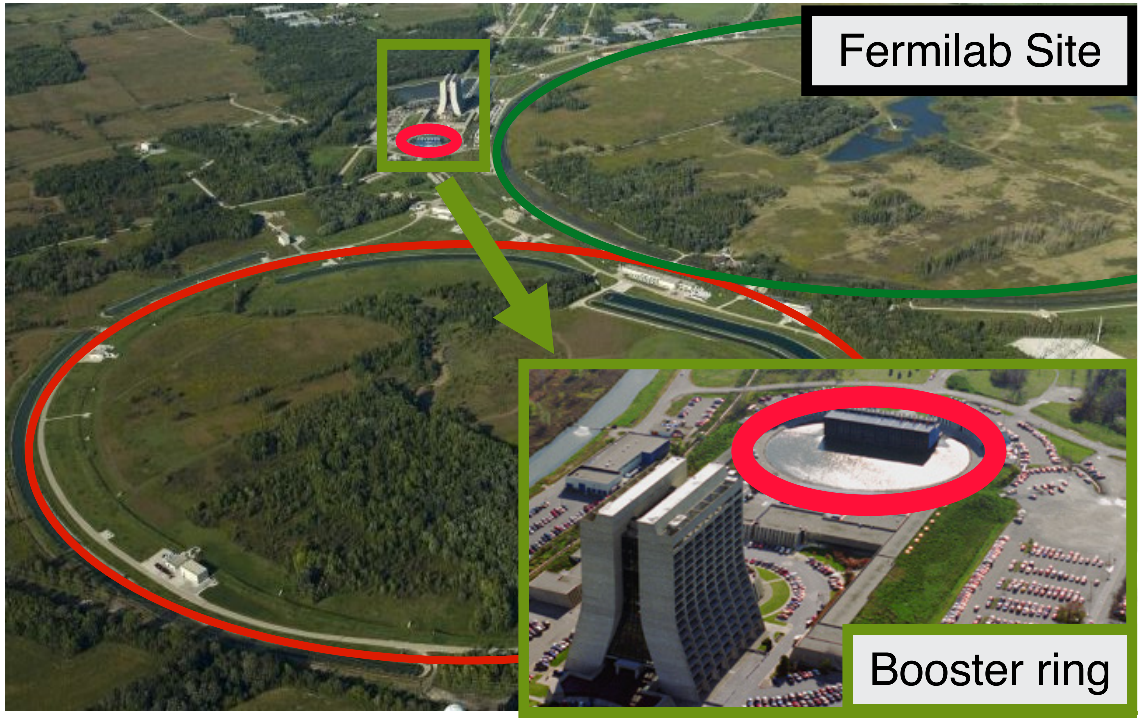


FNAL Accelerator Complex

- Booster synchrotron: 400 MeV H^- from Linac accelerated to 8 GeV protons for delivery to Main Injector, experiments
  - Batches delivered to MI/Recycler @15 hz ('rapid cycling')
- Efficient operation critical for PIP-II goal of MW beam



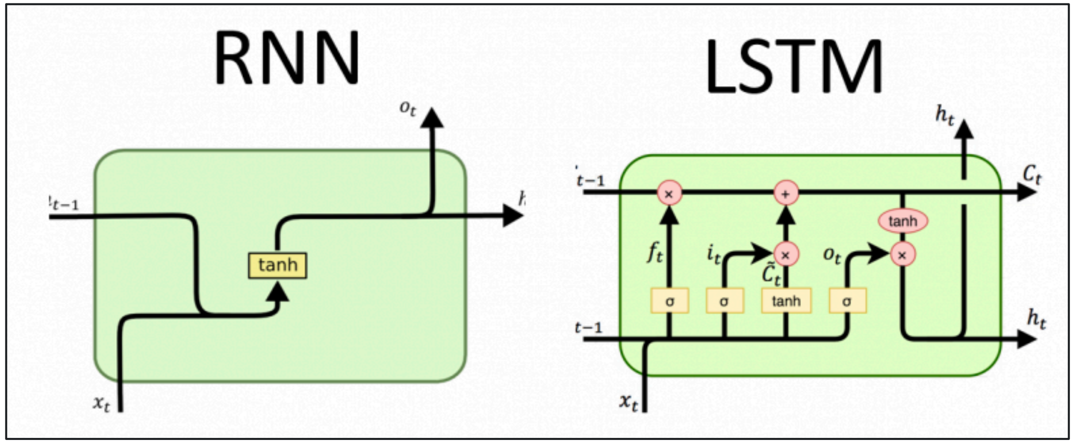
Courtesy: Christian Herwig

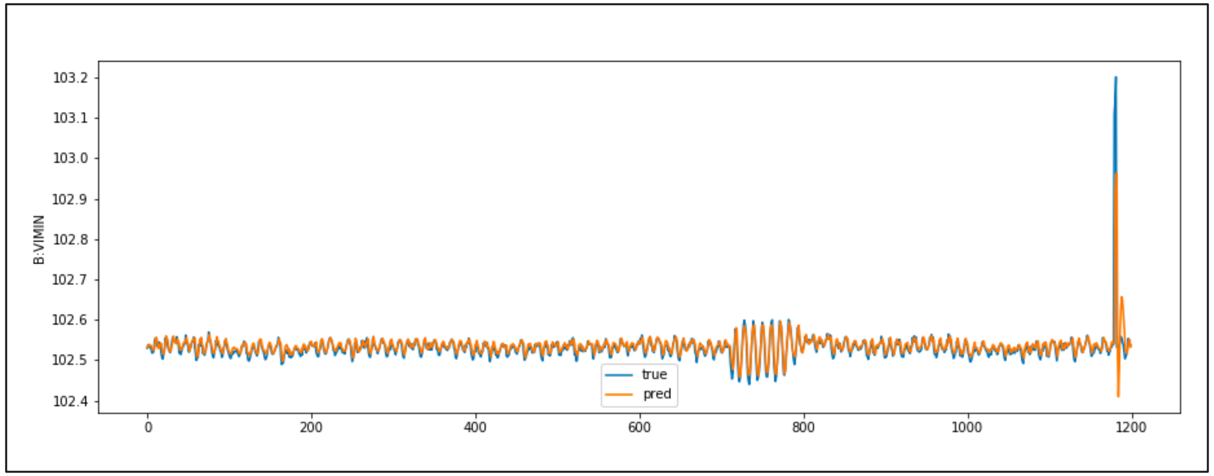


Pacific Northwest
NATIONAL LABORATORY

Digital Twin: Use to mimic the accelerator response

- **Goals:**
  - Develop a forward model to predicts how the accelerator responds to new setting provided by the RL agent
- **Key factors:**
  - Use variables identified during data prep stage
  - Initial model was developed using historical data
  - New data is required to study additional correlations, system lags and for better interpolation

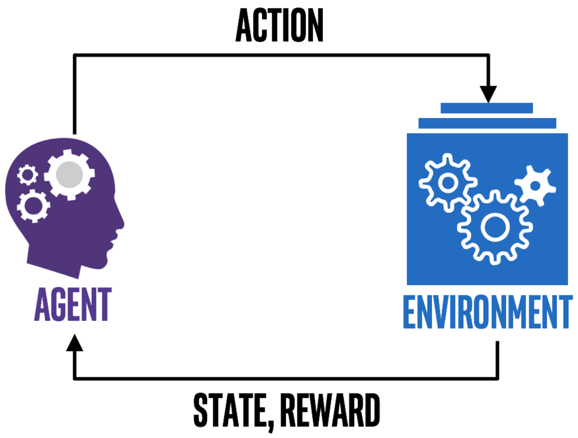




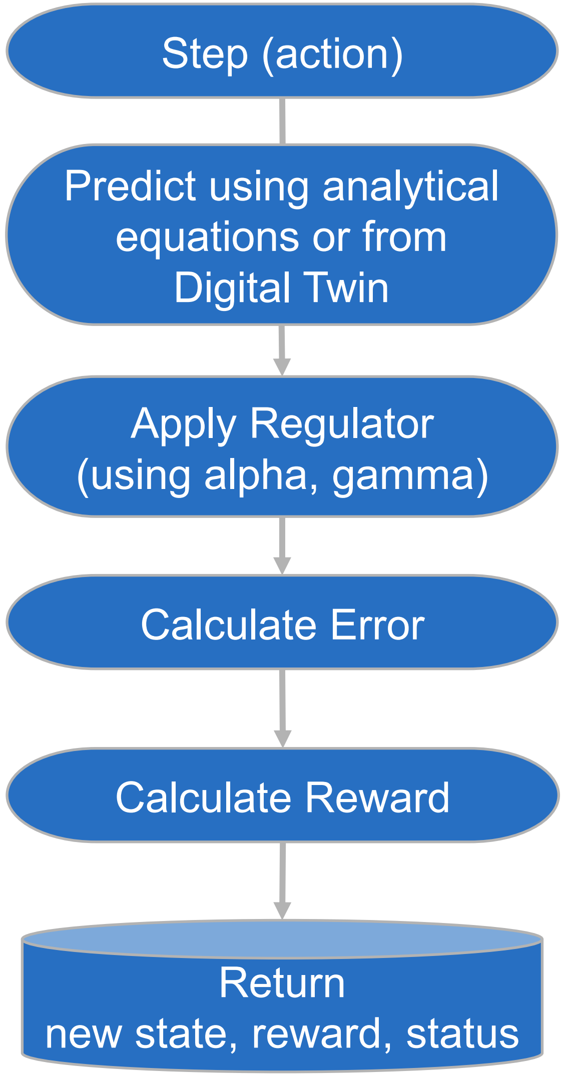
Comparison between ML predictions (orange) and real data (blue) shows good agreement

Reinforcement Learning Workflow

- **Goals:** Develop a RL workflow that provide a policy that creates a set of actions that optimizes reward
- **Setup:**
  - Discretizes action space
  - DQN policy model with greedy epsilon sampling
  - Observation space was a full cycle
  - Reward based on target error goal

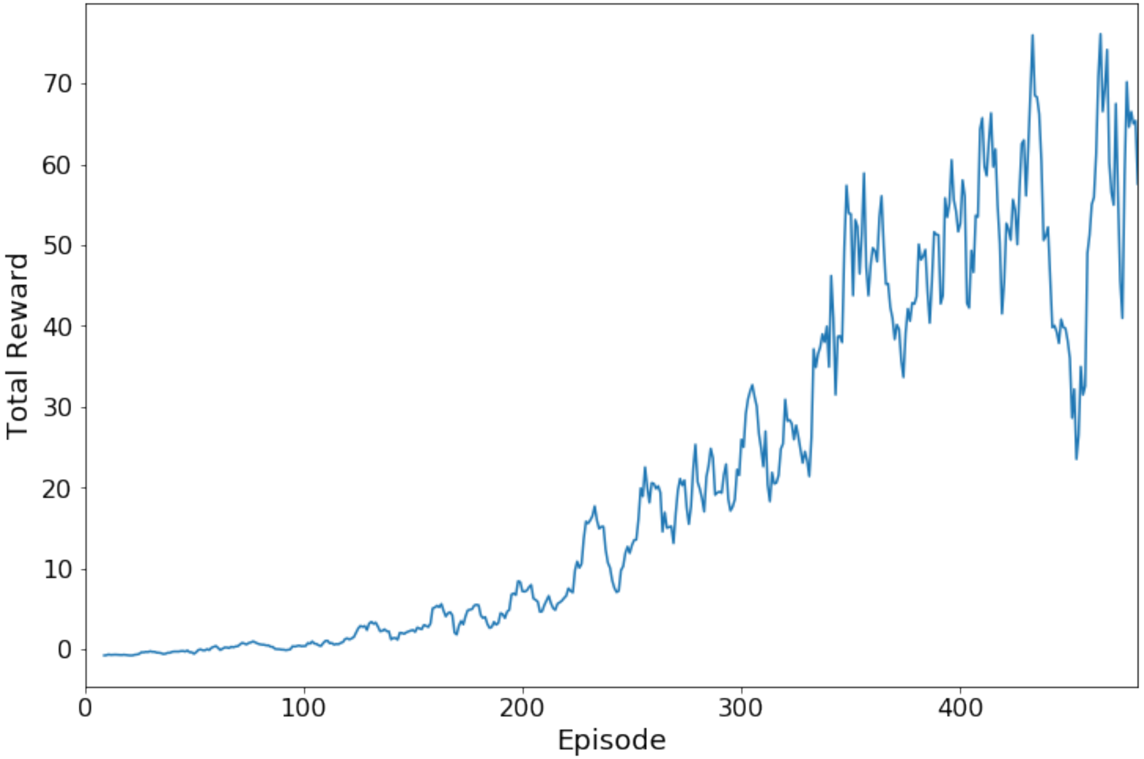


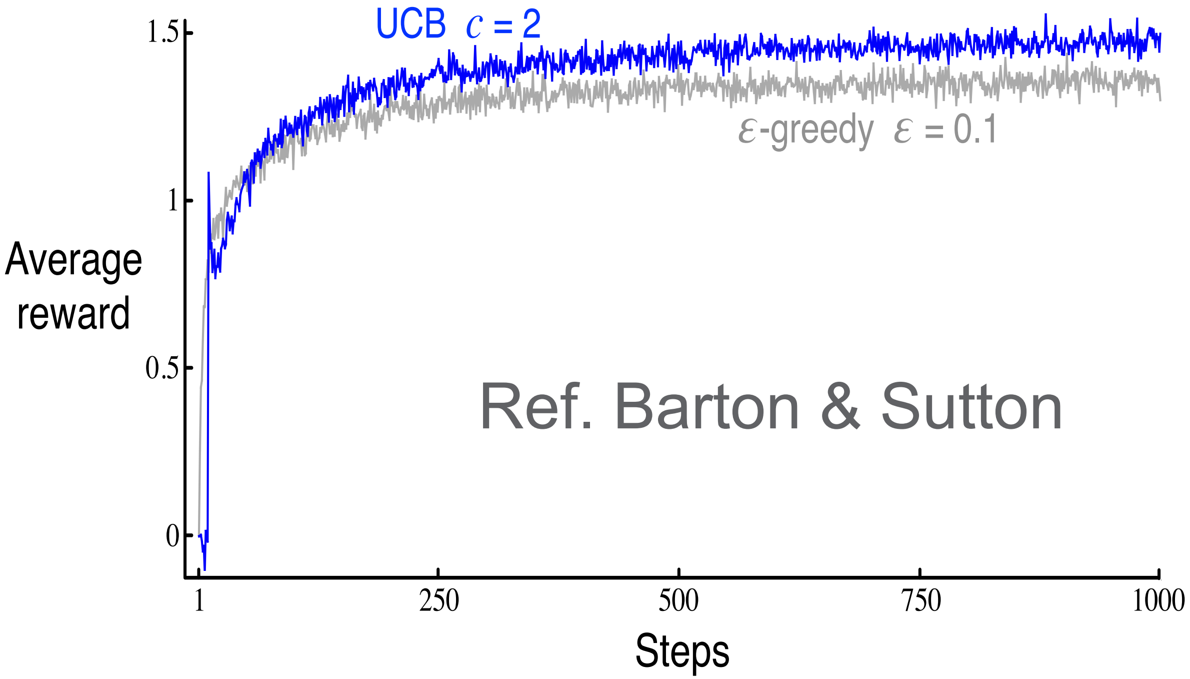
OpenAI


Step (action)
Predict using analytical equations or from Digital Twin
Apply Regulator (using alpha, gamma)
Calculate Error
Calculate Reward
Return new state, reward, status

Example of reward from DQN RL model using accelerator data

- Initial RL optimization show increasing total reward
- Identified optimal setting that requires validation and new data
- RL workflows have additional hyper-parameters that require optimization
- It will be important to optimization these hyper-parameters on DOE computing resources.





Ref. Barton & Sutton

Brief Overview of ExaLearn Control Pillar

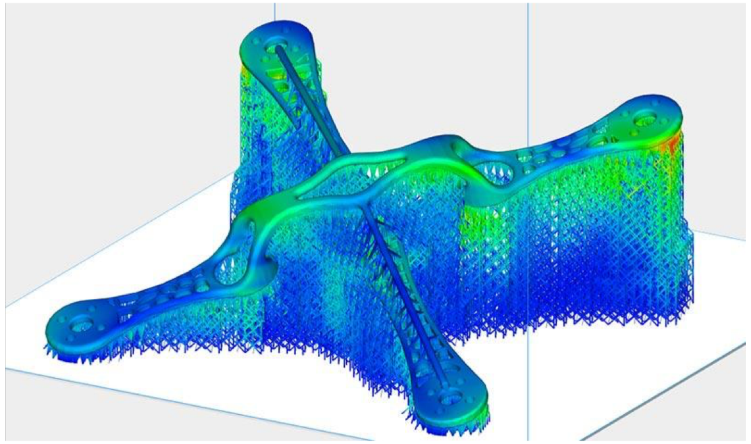
- **Goals**
  - Provide scalable control-related machine learning software for ECP applications
  - Implement use case applications for demonstration and testing
  - Run on exascale DOE leadership class computing facilities

- **Methods**
  - Initially using deep reinforcement learning, however, the workflow can be expanded to other methods
  - Science use case: RL for temperature control for block copolymer self-annealing in light source experiments
  - EXARL software framework for exascale reinforcement learning for science and benchmarking

Control Problems in Science

Simulation

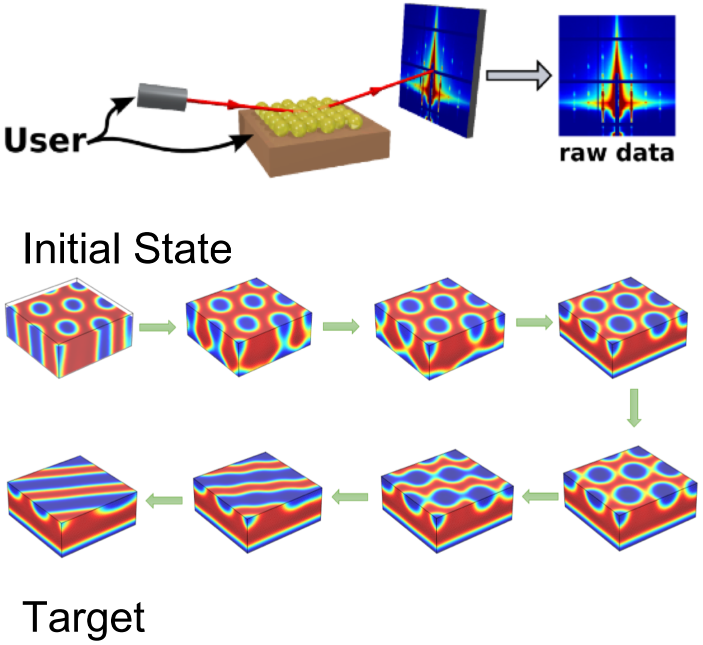
- Accelerate sampling in a simulation via search, to reduce computation required for solution



https://www.materialise.com/en/press-releases/materialise-brings-simulation-for-additive-manufacturing-to-production-floor

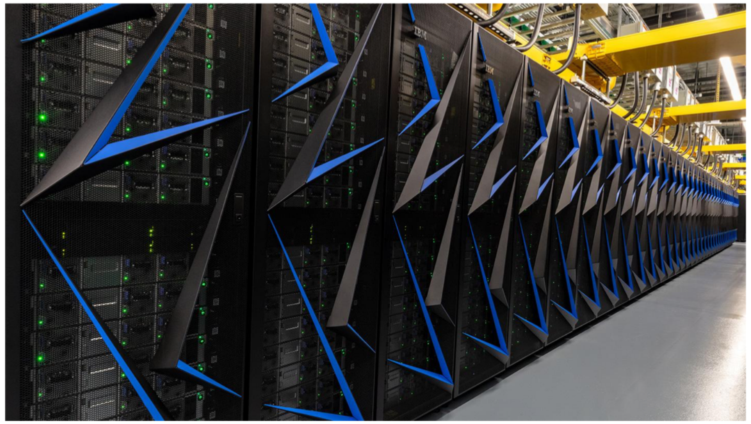
Experiment

- Guide scientific experiments - eg. block copolymer self-annealing



Operation

- Control HPC facility resource management
- Control experimental facilities (eg x-ray beam)
- Control air, land or space vehicles



https://www.olcf.ornl.gov/summit

Example of a Reinforcement Learning Algorithm Workflow

Deep Q-network (DQN)

DQN uses a deep neural network to estimate the value of taking a specific action at a certain state, also called the state-action value or Q-value.

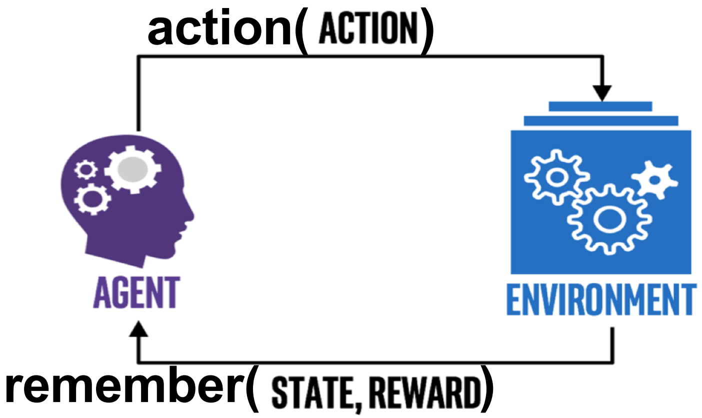
The DQN agent, once trained properly, suggests the action with the highest Q-value as its policy, and maximizes the total reward over the episode.

DQN can suggest optimal discrete actions.

- Policy Models (ML hyper-parameters)
- Exploit and explore (RL hyper-parameters)
- Soft update (RL hyper-parameters)
- Experiences buffer

train():
- Train active model

update():
- Soft update of the target model

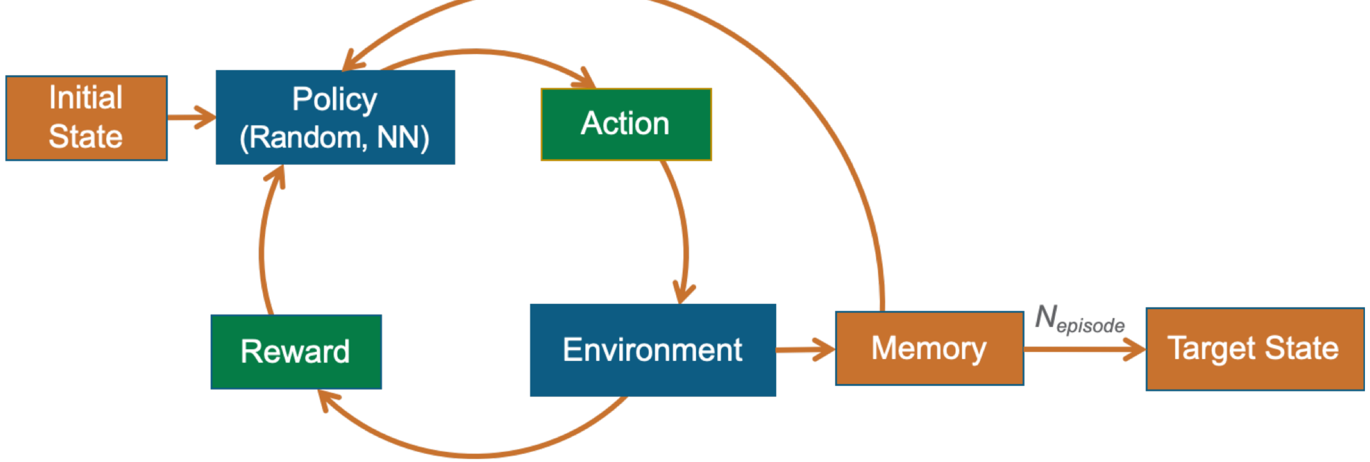


action(ACTION)

AGENT

ENVIRONMENT

remember(STATE, REWARD)



Initial State

Policy (Random, NN)

Action

Environment

Reward

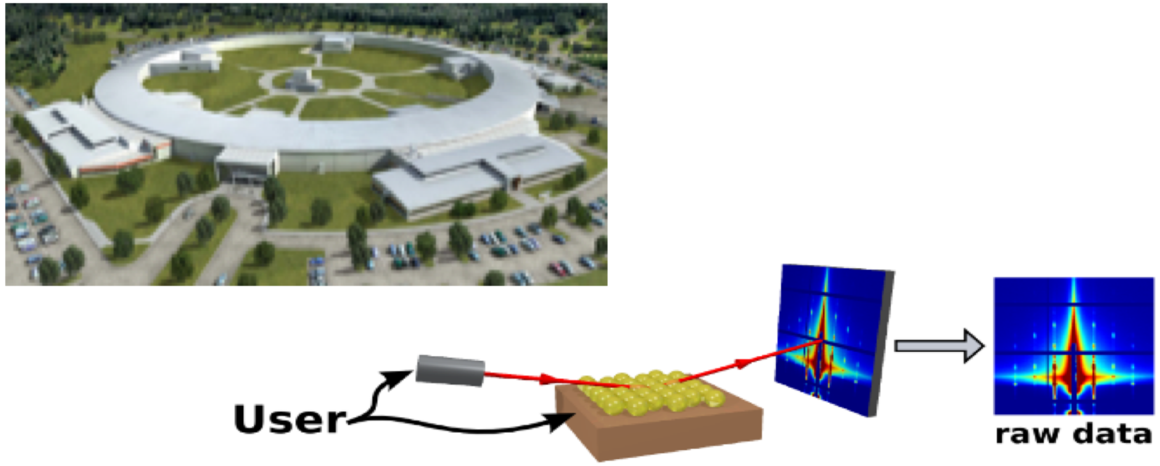
Memory

$N_{episode}$

Target State

Challenges for Block Copolymer Experiments (BCP)

- BCP experiments are performed at DOE light source user facilities.
- Temperature is adjusted to direct the formation of the block copolymers to a target morphology.
- Light source beam shining on sample at small grazing incidence angle produces a diffraction pattern
- The multi-dimensional energy landscape underlying directed block copolymer self-assembly requires engineering a convoluted pathway in order to obtain a target morphology.



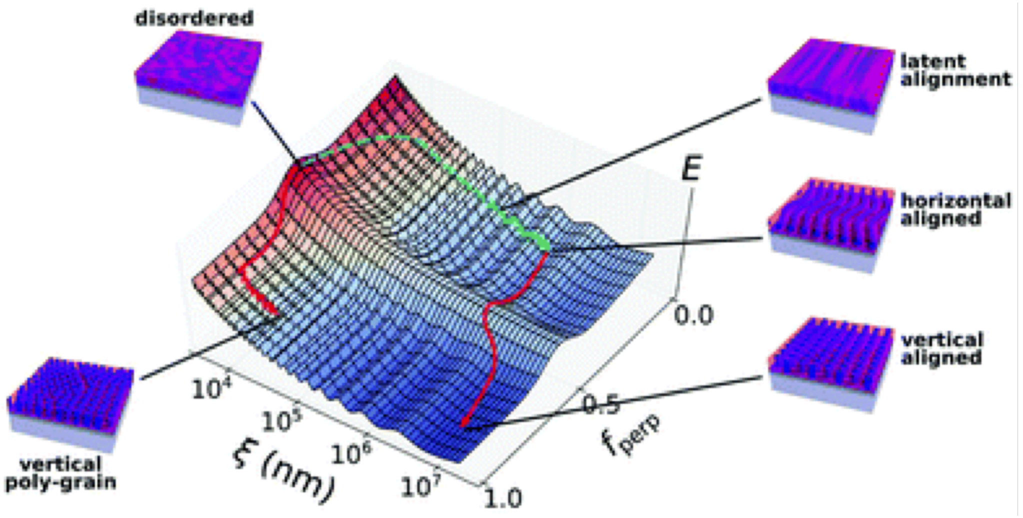
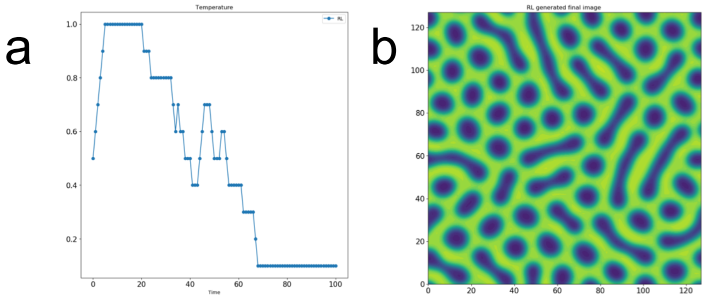


Image from Nanoscale, 2018, 10, 416. Choo, Majewski, Fukuto, Osuji and Yager.

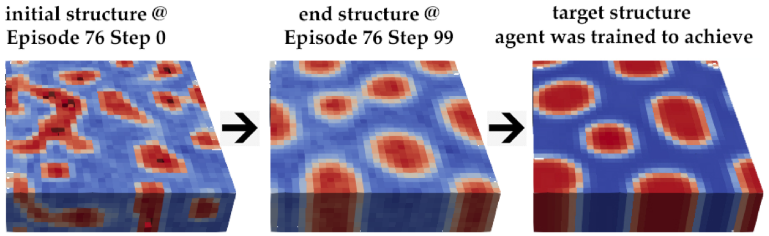
Block Copolymer Reinforcement Learning Challenges and Results

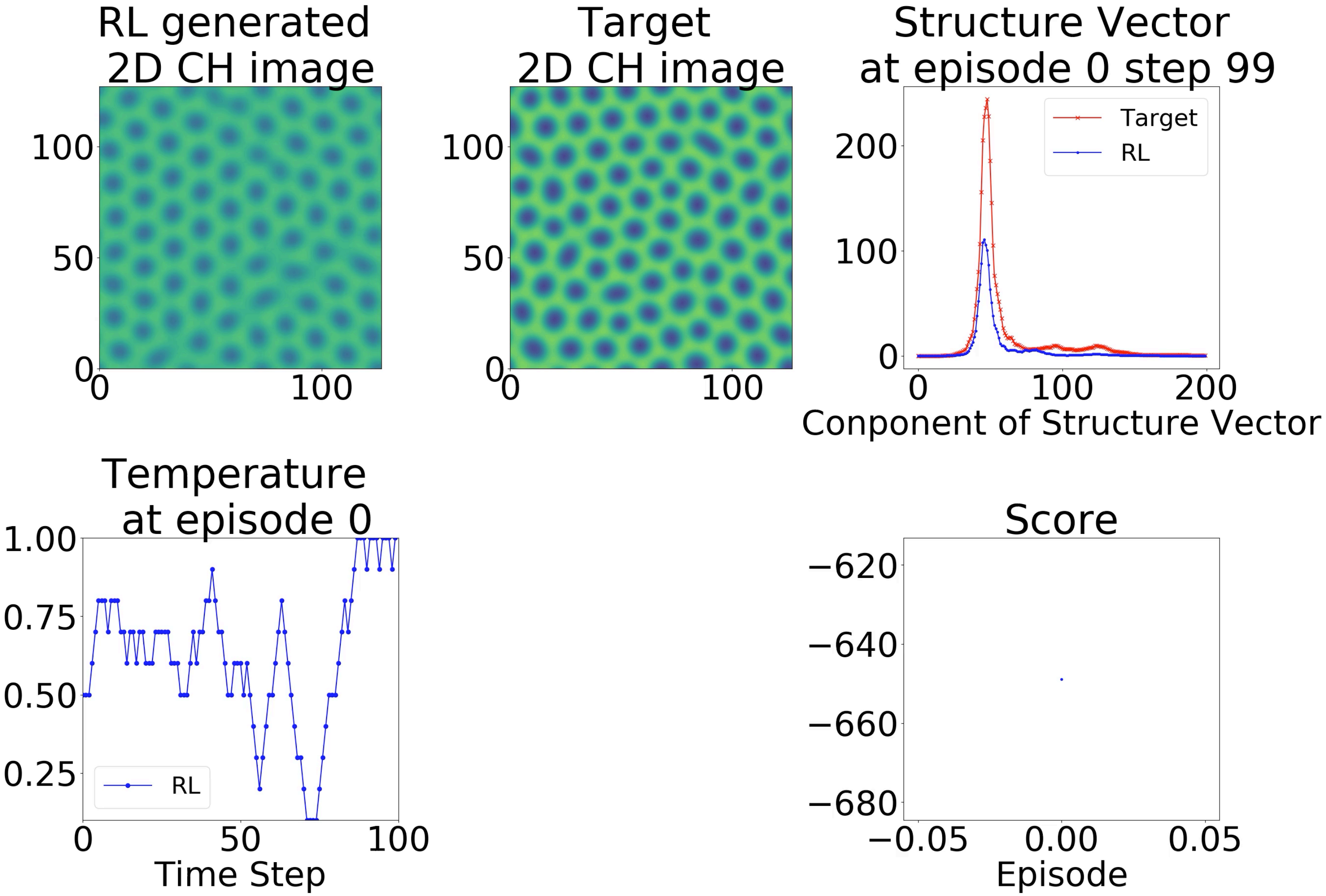
- Develop 2D/3D CPU and GPU surrogate models
- Structure vector to capture characteristics



RL algorithm develops policy that helps control temperature during self-annealing (a), which results in BCP morphology (b).

3D Block Co-polymer Reinforcement Learning Application





Motivation for Deep RL Learning

Complex problems like Go Game have almost infinite possibilities

Traditional Reinforcement Learning Approach

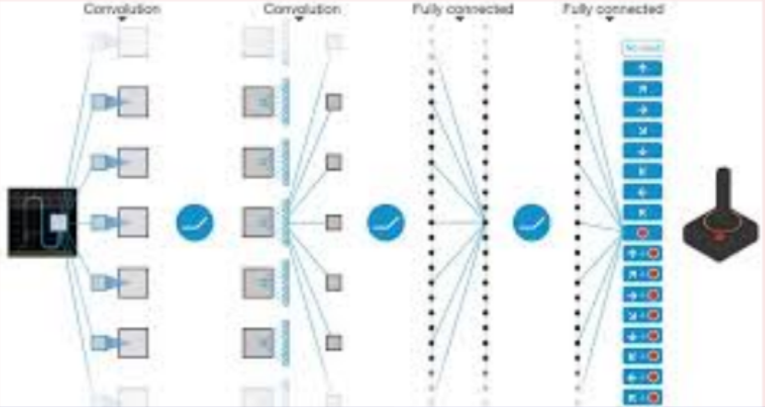
- Generally use a lookup table to decide an action

	Action 1	⋯	Action n
State 1			
⋮		⋮	
State m			

Deep Reinforcement Learning Approach

- A lookup table can be large
  ⇒ **Approximation** by Deep Learning method
- Deep learning generally performs well for this approximation with a big data

(Reference: `https://skymind.ai/wiki/deep-reinforcement-learning`)

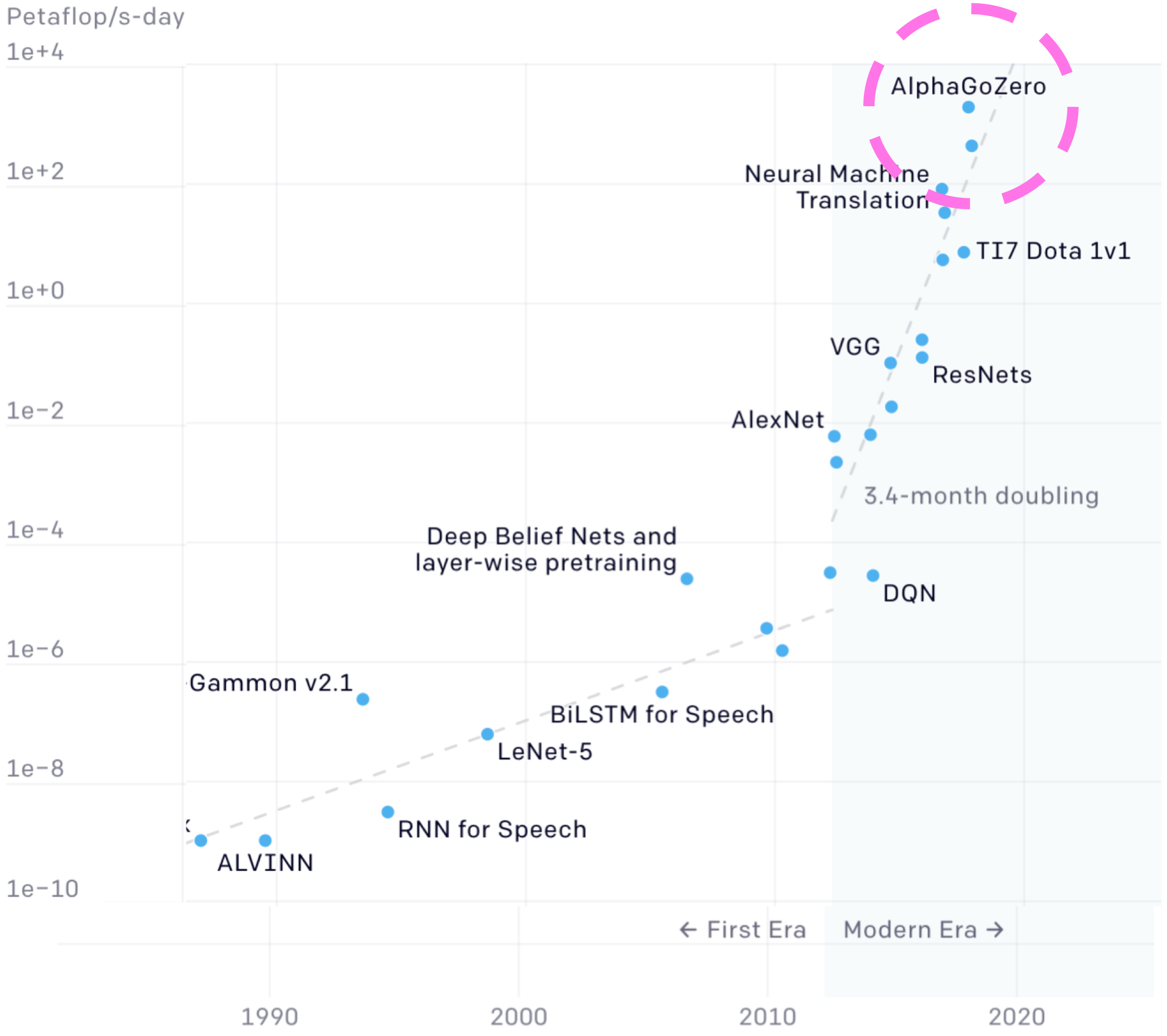


Why Reinforcement Learning at Scale?

Reasons RL needs to scale:

- Environments may use many computational resources (CPUs, GPUs, etc.)
- Function approximator for complex tasks will use deep ML models
- Large number of actions and/or states
- Policy network ML hyper-parameters
- New RL hyper-parameters will need to be studied

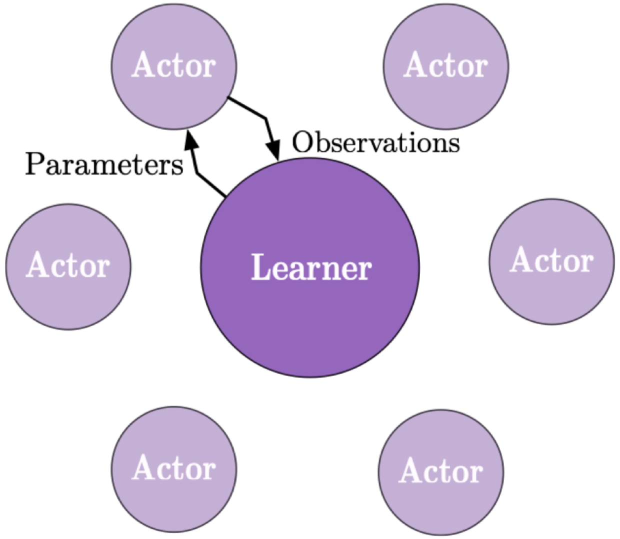
Two Distinct Eras of Compute Usage in Training AI Systems

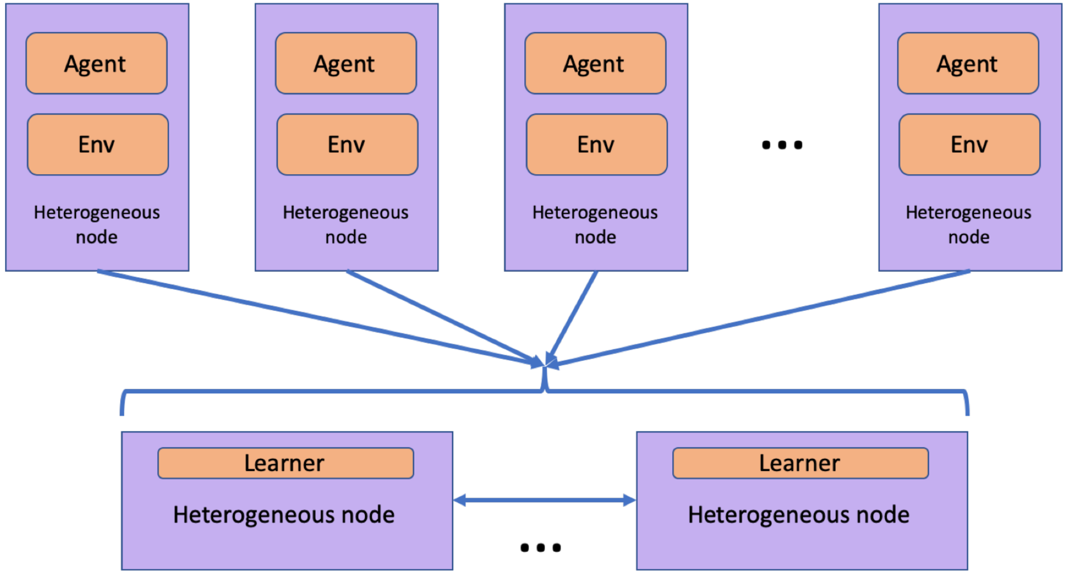


Excerpted from https://openai.com/blog/ai-and-compute

Scaling Challenges for Reinforcement Learning

- How many learners do we need?
- How many actors will saturate a learner?
- How to optimize multi-learners setup?
- How many experiences to send in batch to learner?
- How much learning is required to maximize the GPU efficiently?
- How to balance computation and memory usage for learning versus environment on a node?
- How to minimize policy model lag/stagnation
- Tuning ML & RL learning parameters
  - Extending ***CANDLE*** to incorporate RL workflows and use sophisticated resources management

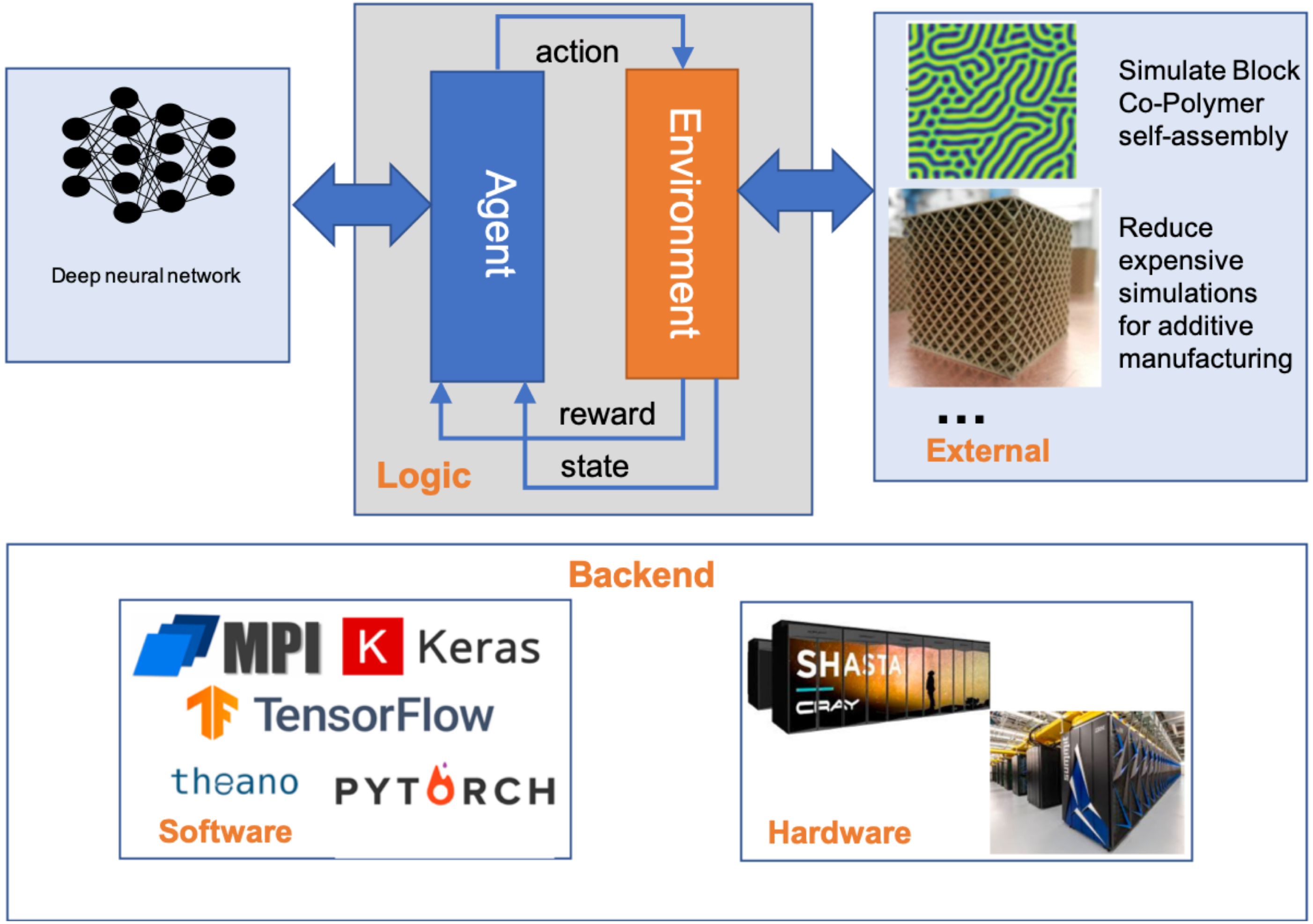




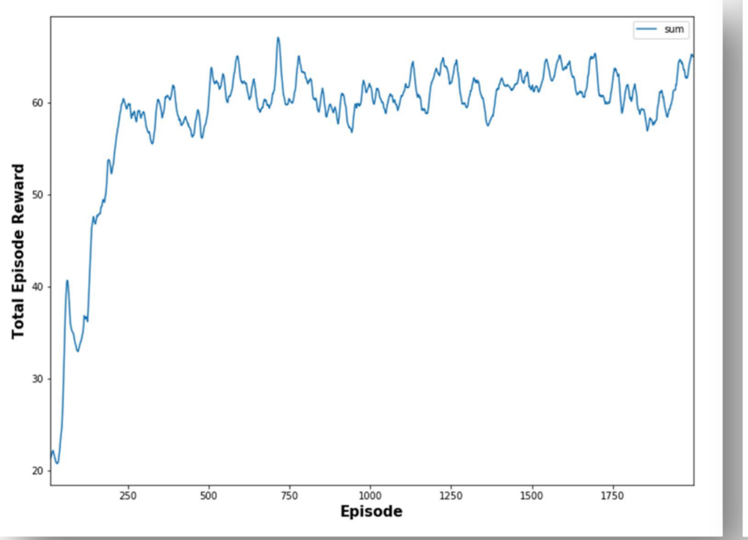
Take away: Many challenges to scaling RL so use EXARL!

Easily eXtendable Architecture for Reinforcement Learning (EXARL)

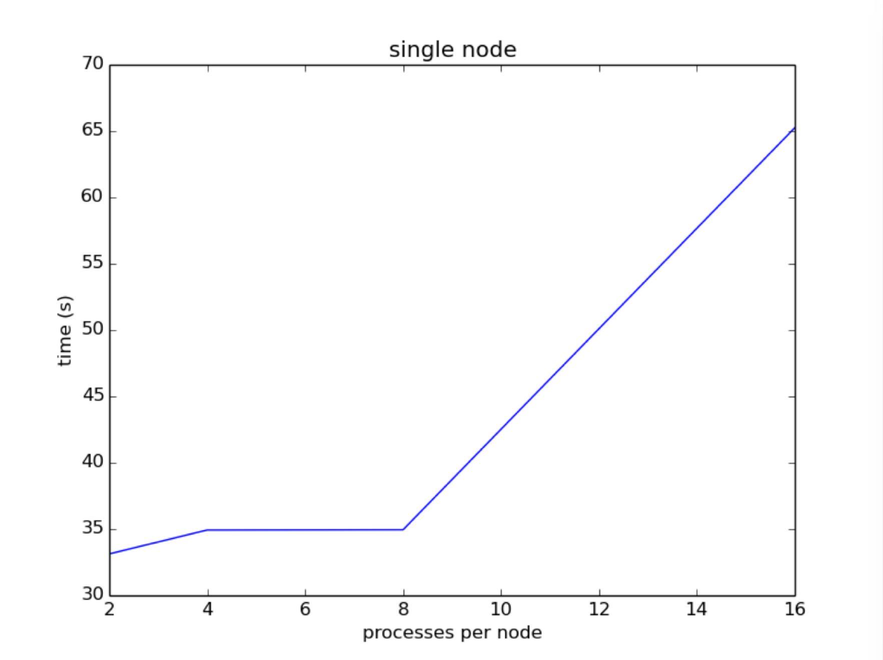
- EXARL: scalable RL framework for scientific environments
- Extends OpenAI Gym’s environment registry to agents
- Dynamic multi-node environments
- Abstract classes to mandate necessary functionality
- Easy to register new agents and environments
- Supports different hardware and software infrastructures
    - Use existing prevalent infrastructure

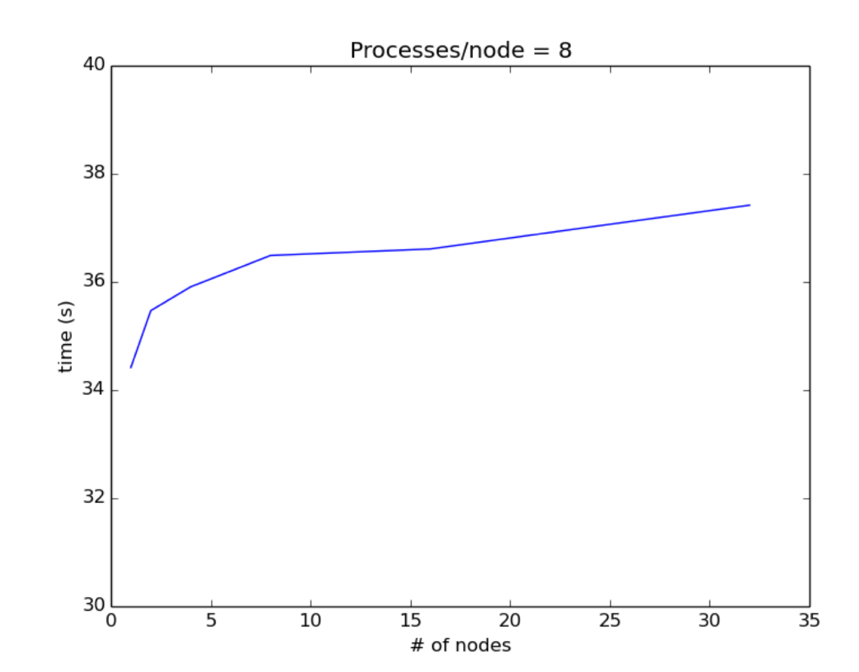


Performance of EXARL – Preliminary Results



Reinforcement learning with ExaCartpole environment within EXARL framework using DQN algorithm on one node of Summit.





Preliminary results from the EXARL framework proxy agents and proxy learners run on LANL Darwin cluster. These results show weak scaling of one learner and multiple agents (processes) during online reinforcement learning on Broadwell nodes (2 sockets, 18 cores each). Number parent processes are shown; each parent spawned 2 children.