Robot Learning
Spring 2022

Sae Na Na
DUV Reticle Stage MTD

April 26, 2024
Wilton CT

I took a Robot Learning class this spring
An introduction to machine learning in robotics

- Supervised and unsupervised learning

- Deep learning
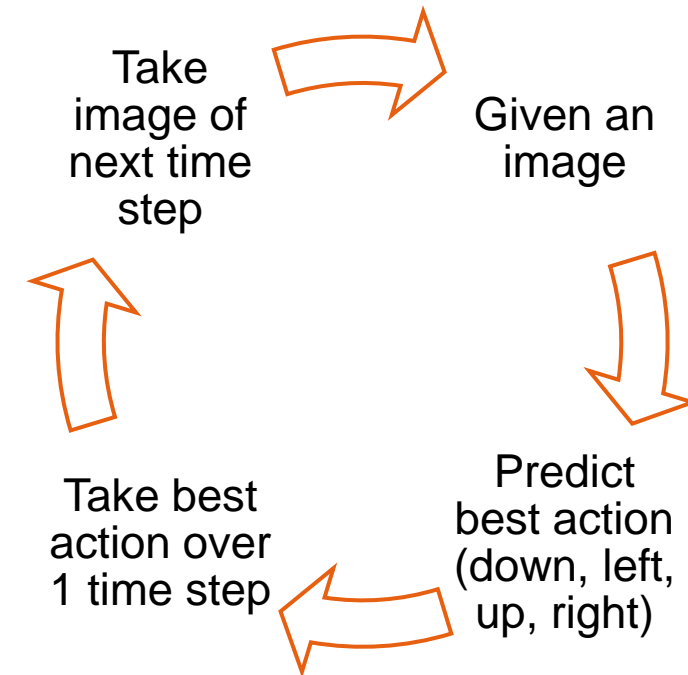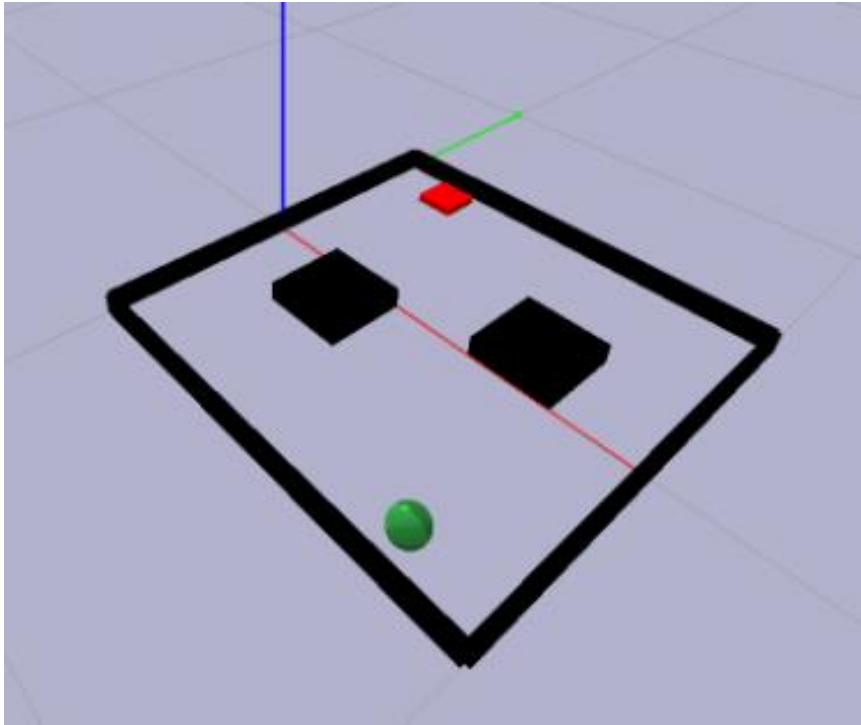
- Forward models

- Reinforcement learning

Structure of the class

- lectures on theory + recent paper discussions
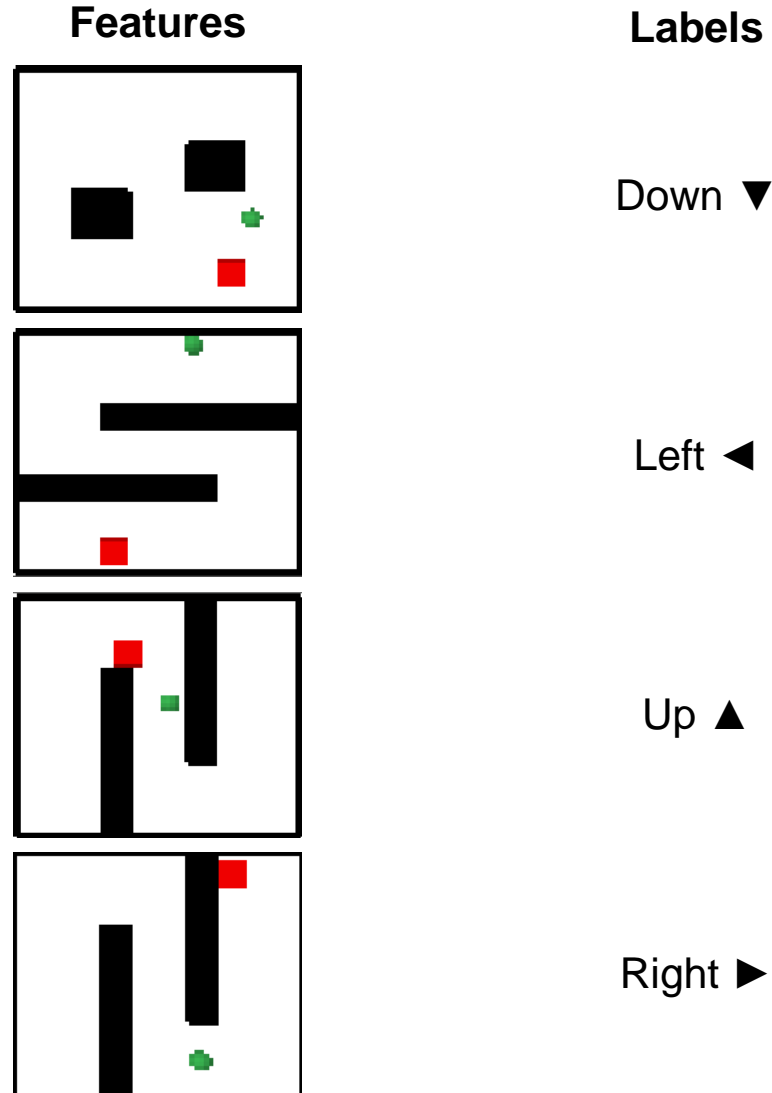
- 5 programming projects in Python

Behavioral cloning

Behavioral cloning
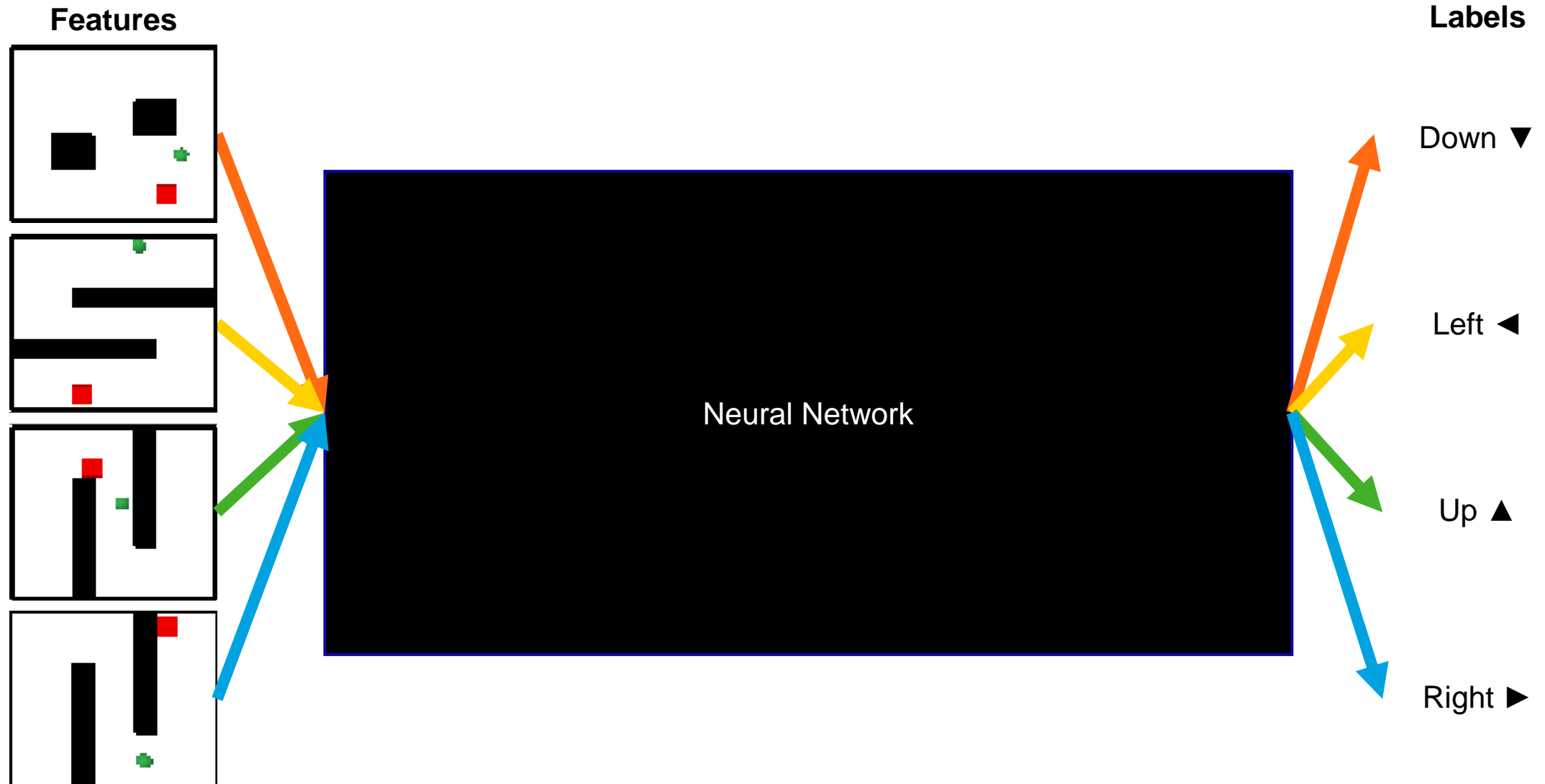## Given an RGB image of the world, choose the best action to reach the goal

Take image of next time step → Given an image → Predict best action (down, left, up, right) → Take best action over 1 time step →

Behavioral cloning
Given an RGB image of the world, choose the best action to reach the goal

Features

Labels

Down ▼

Left ◄

A dataset with 12000 feature-label pairs is *provided* to us

Up ▲

Right ▶

April 26, 2024

Behavioral cloning
Given an RGB image of the world, choose the best action to reach the goal

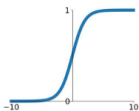**Features**

Down ▼

Neural Network

Left ◄

Up ▲

Right ▶

Neural Networks
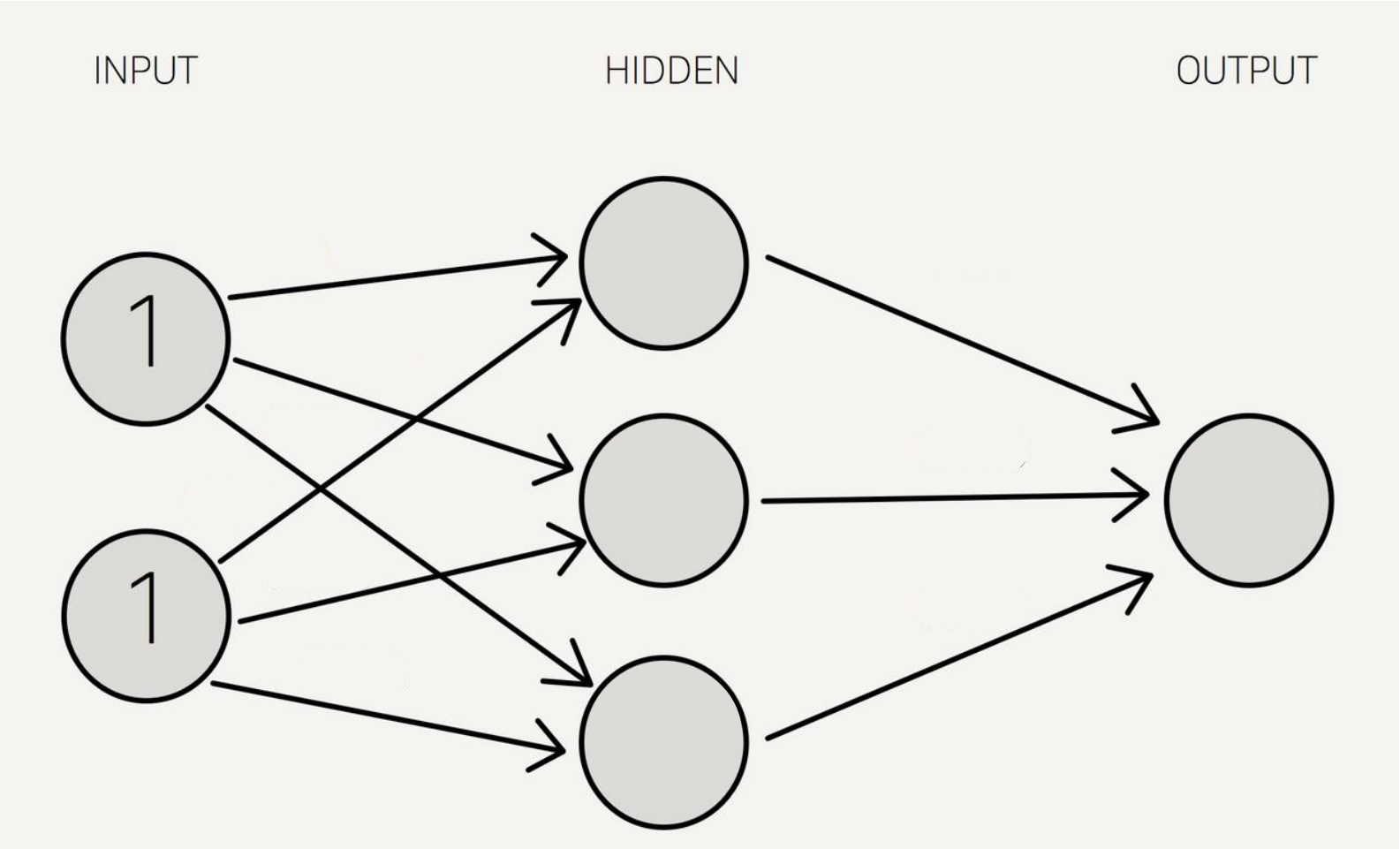
Activation Functions
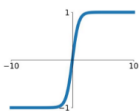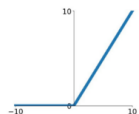
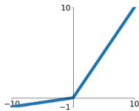**Sigmoid**
$\sigma(x) = \frac{1}{1+e^{-x}}$

tanh
$\tanh(x)$

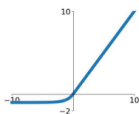**ReLU**
$\max(0, x)$

Leaky ReLU
$\max(0.1x, x)$

Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
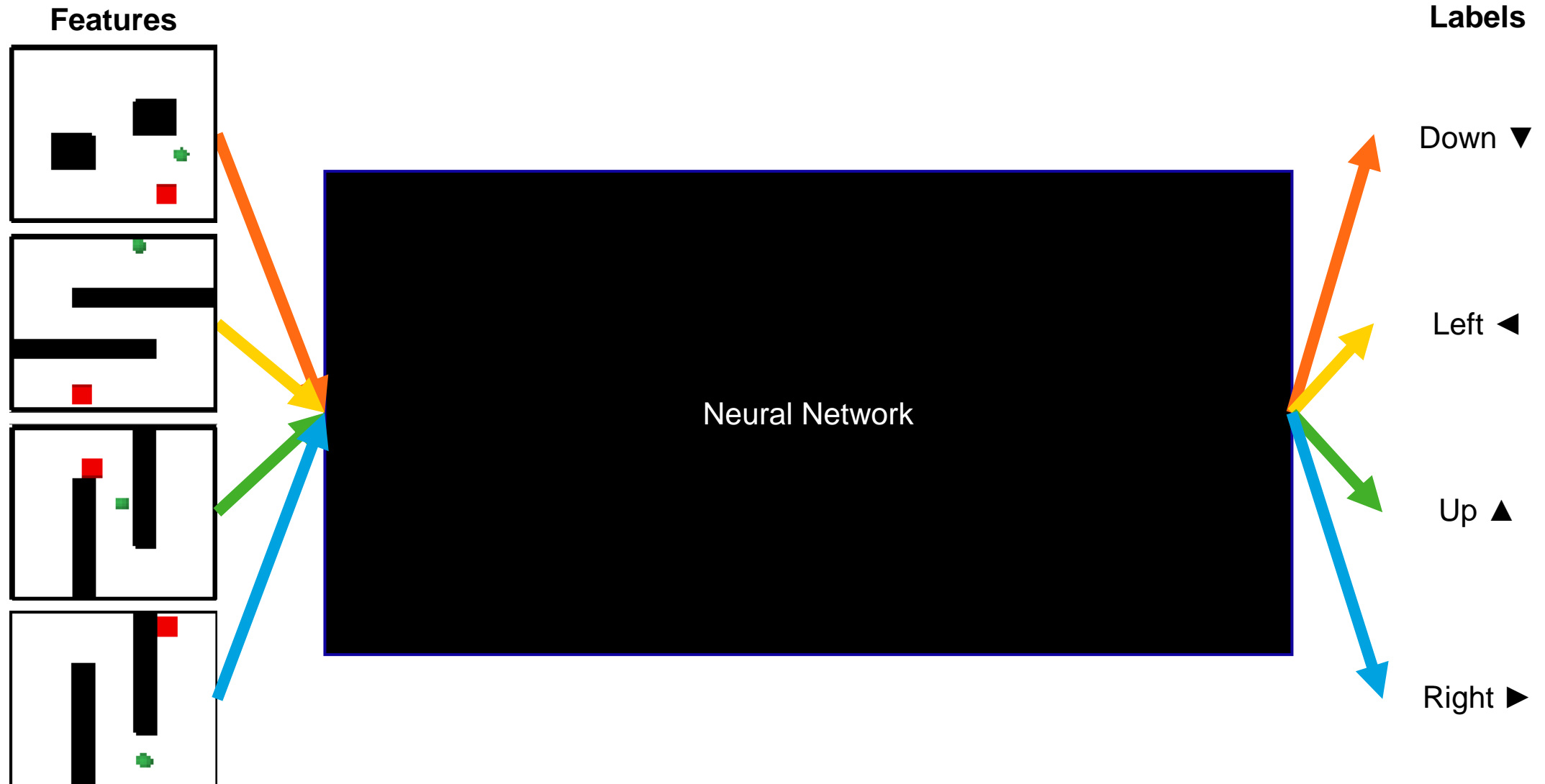$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

Forward propagation

Backpropagation

Behavioral cloning
Given an RGB image of the world, choose the best action to reach the goal

Features

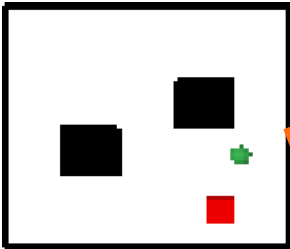**Labels**

Neural Network

Down ▼

Left ◄

Up ▲

Right ▶

ASML

Behavioral cloning

Given an RGB image of the world, choose the best action to reach the goal

Features

64x64x3

Labels

Down ▼

Neural Network

$\mathbb{R}^{64x64x3}$ → $\mathbb{R}^{60x60x6}$ → $\mathbb{R}^{56x56x1}$ → $\mathbb{R}^{3136}$ → $\mathbb{R}^{512}$ → $\mathbb{R}^{128}$ → $\mathbb{R}^{4}$
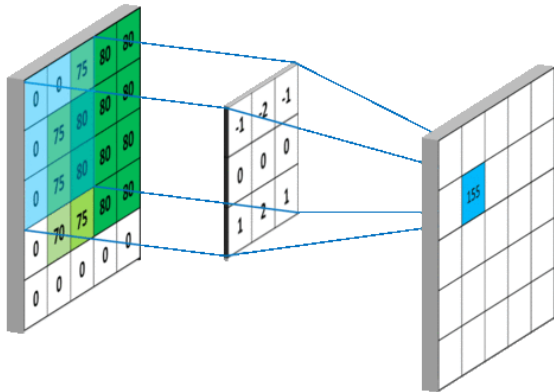
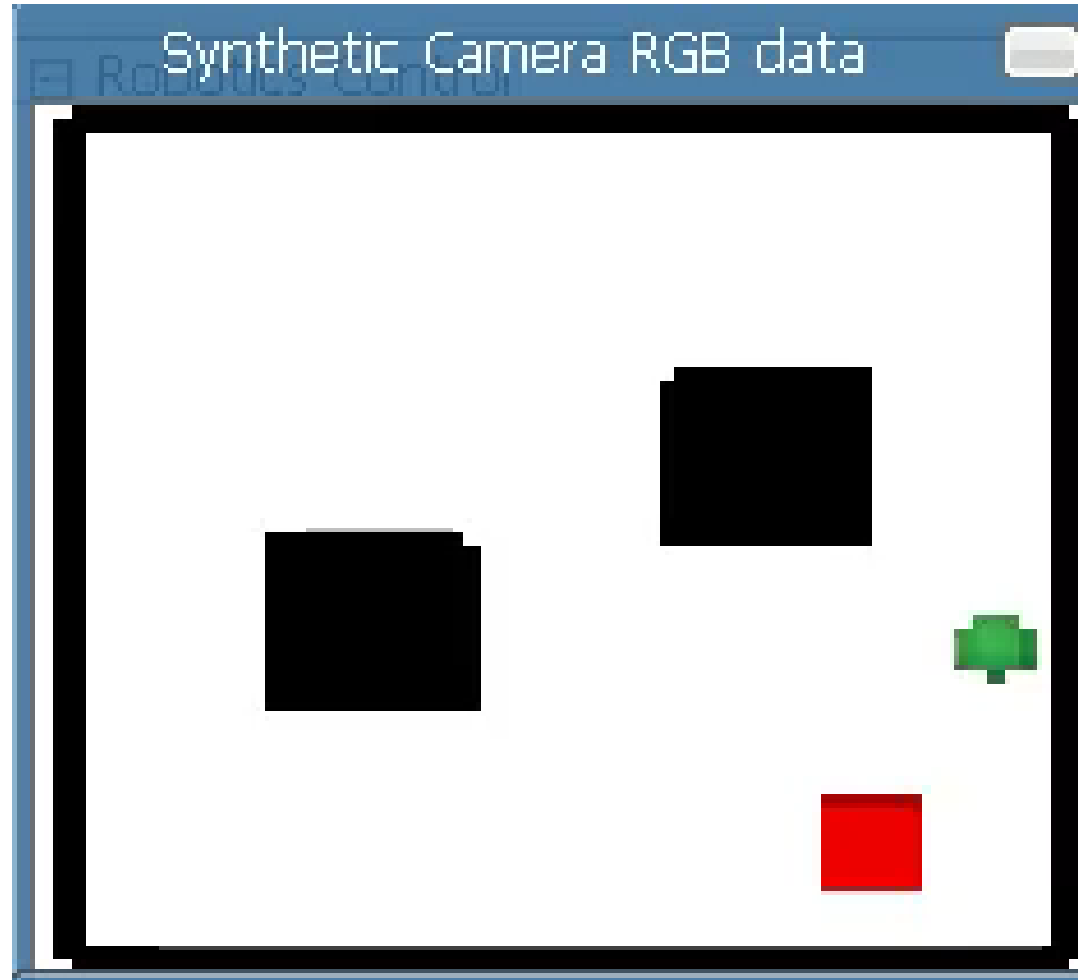1,671,680 weights to optimize
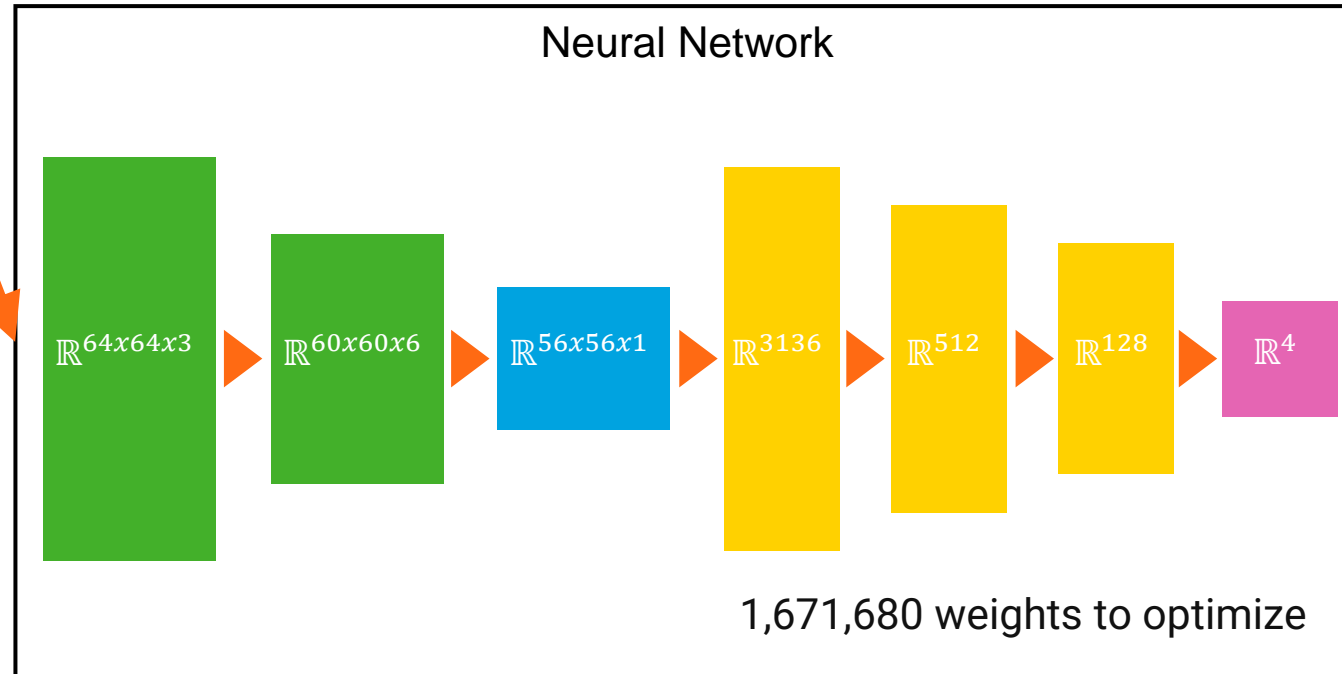
Convolution + ReLU + Pooling   Flatten   Fully connected + ReLU   Argmax
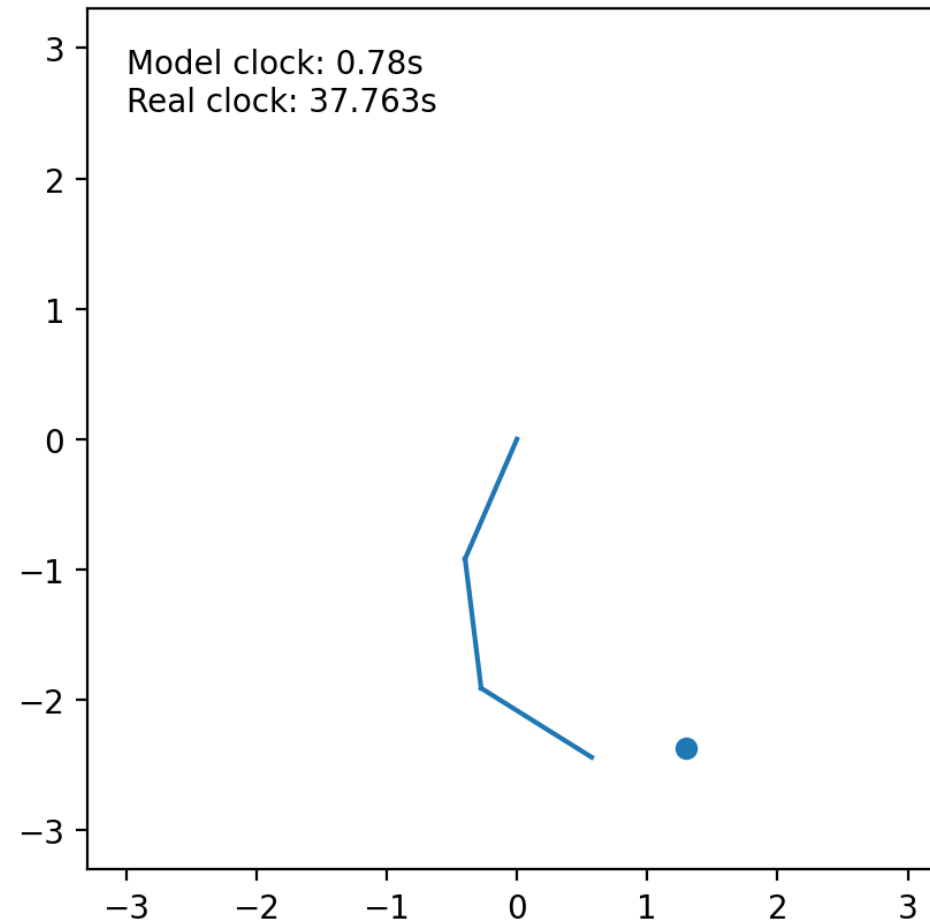
ASML

Behavioral cloning
Given an RGB image of the world, choose the best action to reach the goal

April 26, 2024

Model Predictive Control,
and learning a forward model

MPC and learning a forward model
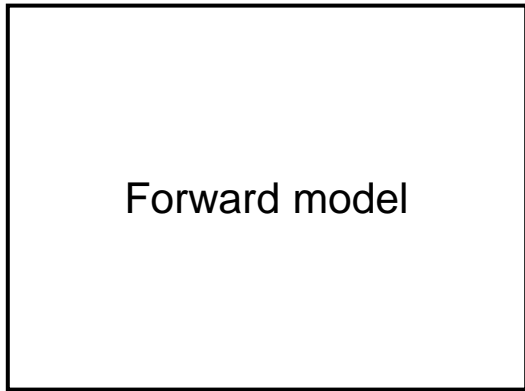## Get the end-effector of the 1-, 2-, and 3-link pendulum to reach the goal within a threshold

Model clock: 0.78s
Real clock: 37.763s

MPC and learning a forward model
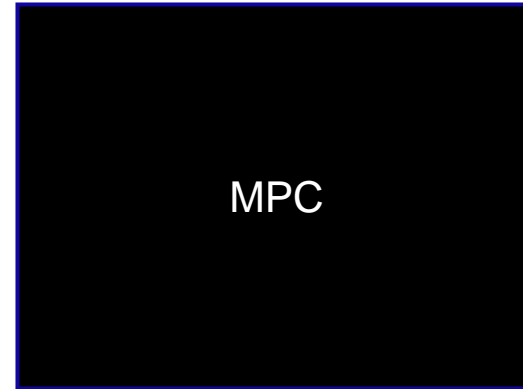## Get the end-effector of the 1-, 2-, and 3-link pendulum to reach the goal within a threshold

State and action

$$x_k = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$u_k = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Forward model

Next state

$$x_{k+1} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$
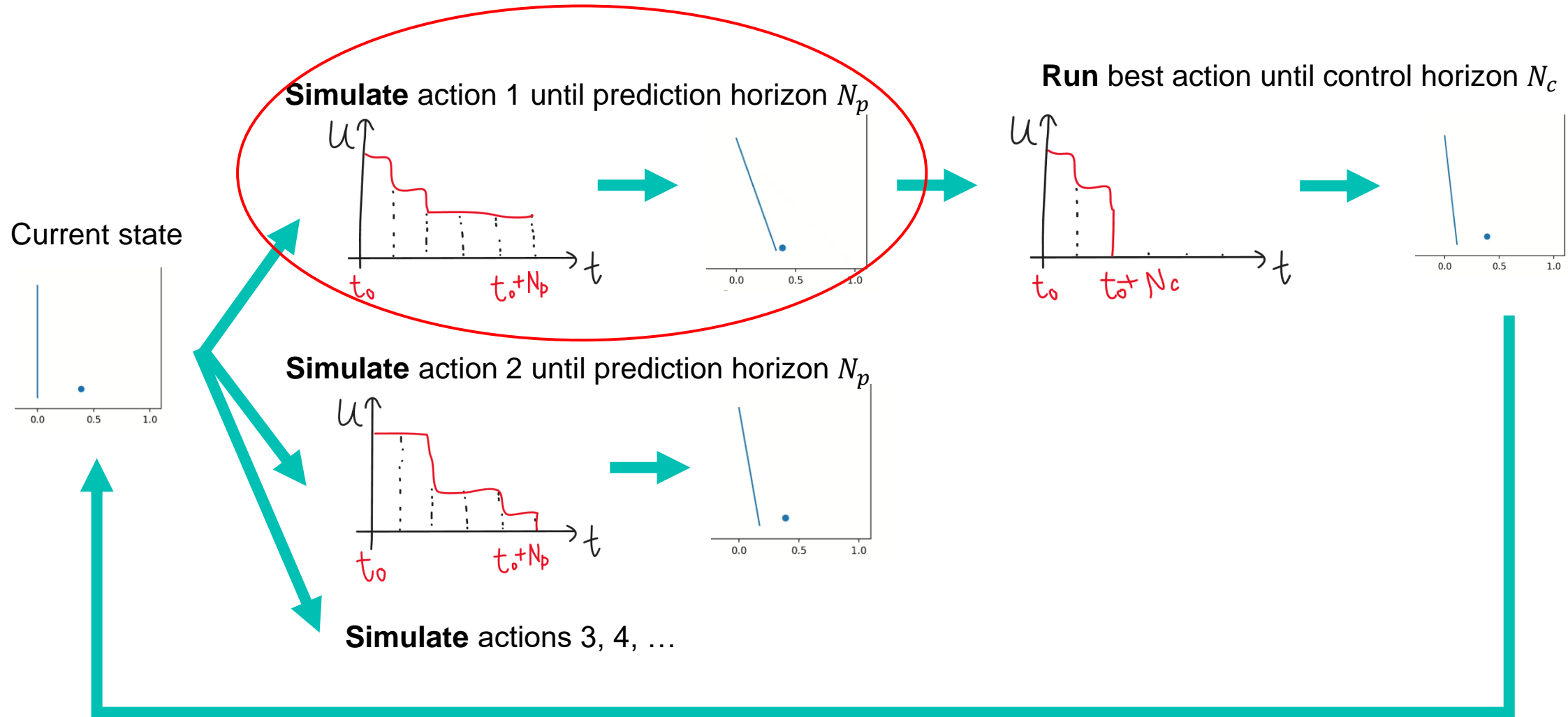
MPC

Best action

$$u_{k+1} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Model Predictive Control (MPC)

Can be MIMO, considers future states, needs knowledge of forward model

Current state

Simulate action 1 until prediction horizon N_p

Run best action until control horizon N_c

Simulate action 2 until prediction horizon N_p

Simulate actions 3, 4, …

MPC and learning a forward model
Get the end-effector of the 1-, 2-, and 3-link pendulum to reach the goal within a threshold

Model clock: 0.02s
Real clock: 1.216s

# links	Prediction horizon	Control horizon	Delta u
1	50	10	0.1
2	40	2	0.1
3	90	10	0.2, 0.1

State and action

$$x_k = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$u_k = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Forward model

Next state

$$x_{k+1} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

MPC

Best action

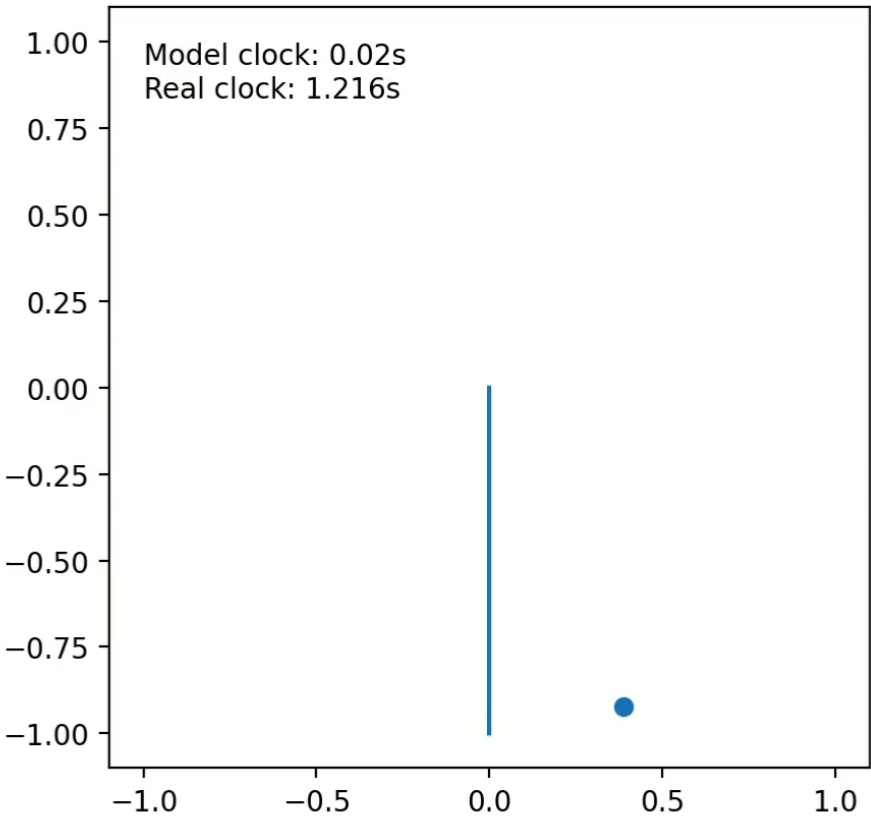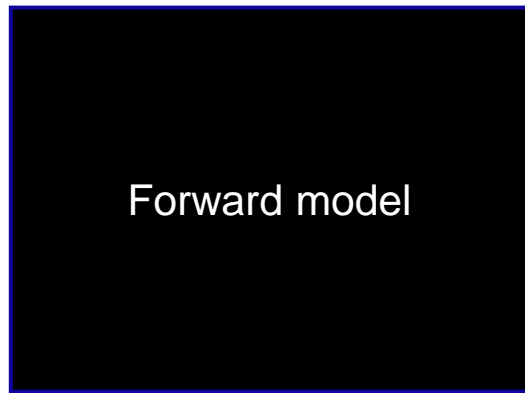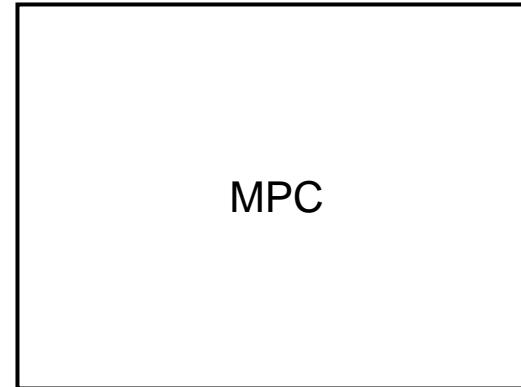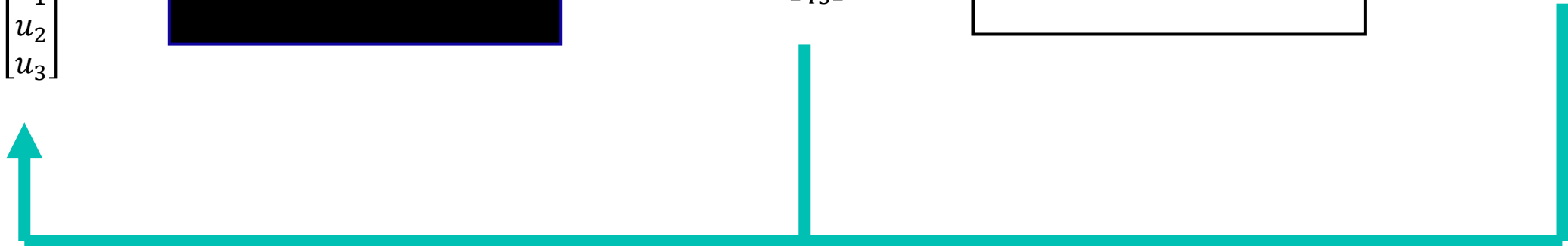$$u_{k+1} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$
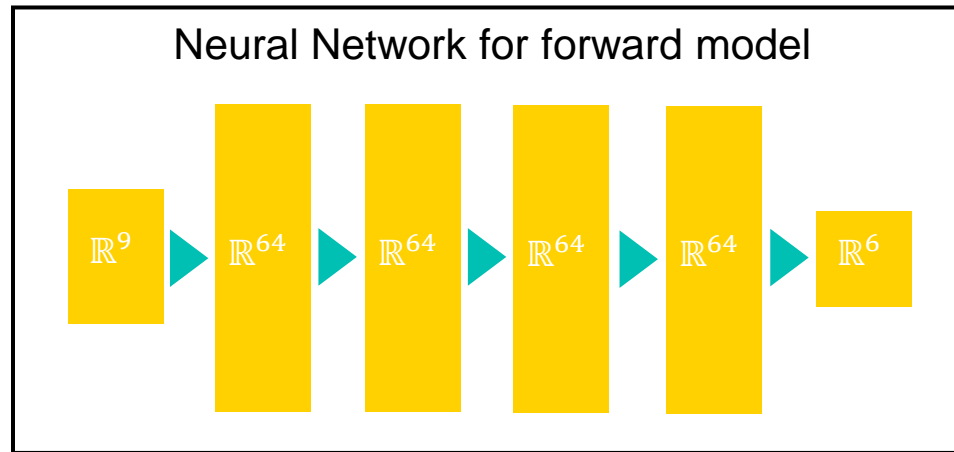
Forward model

Assume Markovian

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$$

- $\mathbf{x}_k$ - state of the system at time step k
- $\mathbf{u}_k$ - action / command given to the robot at time step k
- $\mathbf{x}_{k+1}$ - state of the system at the next time step, k+1
- Using discretized time - size of time step Δt not explicitly part of forward model, but often very important

State and action

$$x_k = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$u_k = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Neural Network for forward model

$\mathbb{R}^9$ → $\mathbb{R}^{64}$ → $\mathbb{R}^{64}$ → $\mathbb{R}^{64}$ → $\mathbb{R}^{64}$ → $\mathbb{R}^6$
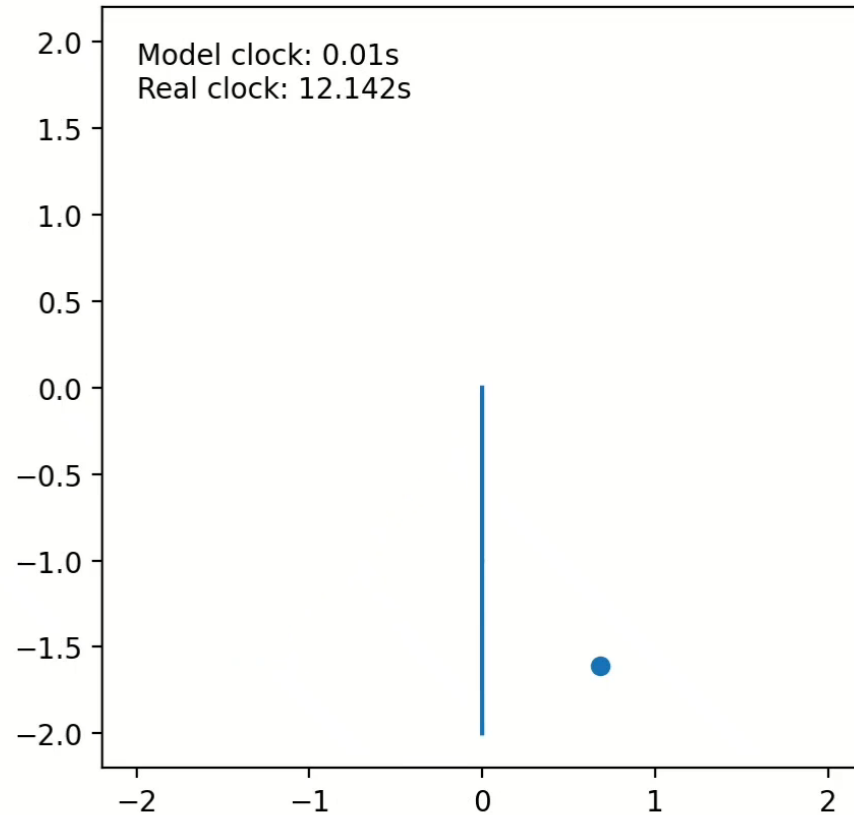
Fully connected + ReLU

Next state

$$x_{k+1} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$
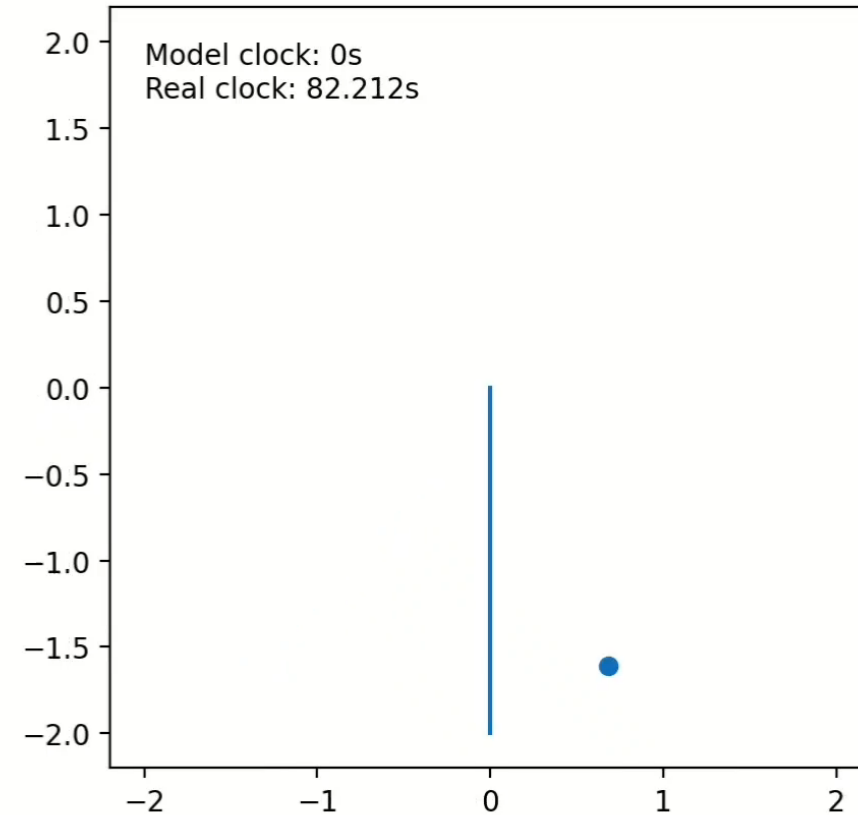
MPC and learning a forward model
Get the end-effector of the 1-, 2-, and 3-link pendulum to reach the goal within a threshold

Analytical forward model

Neural network forward model

Model clock: 0.01s
Real clock: 12.142s

Model clock: 0s
Real clock: 82.212s

MPC and learning a forward model
Get the end-effector of the 1-, 2-, and 3-link pendulum to reach the goal within a threshold
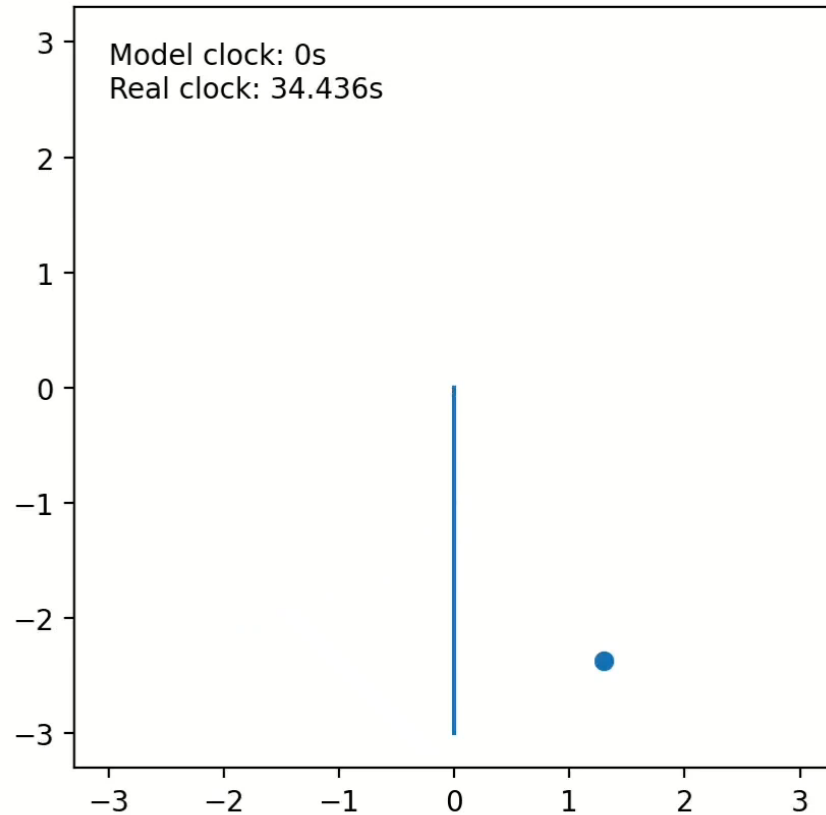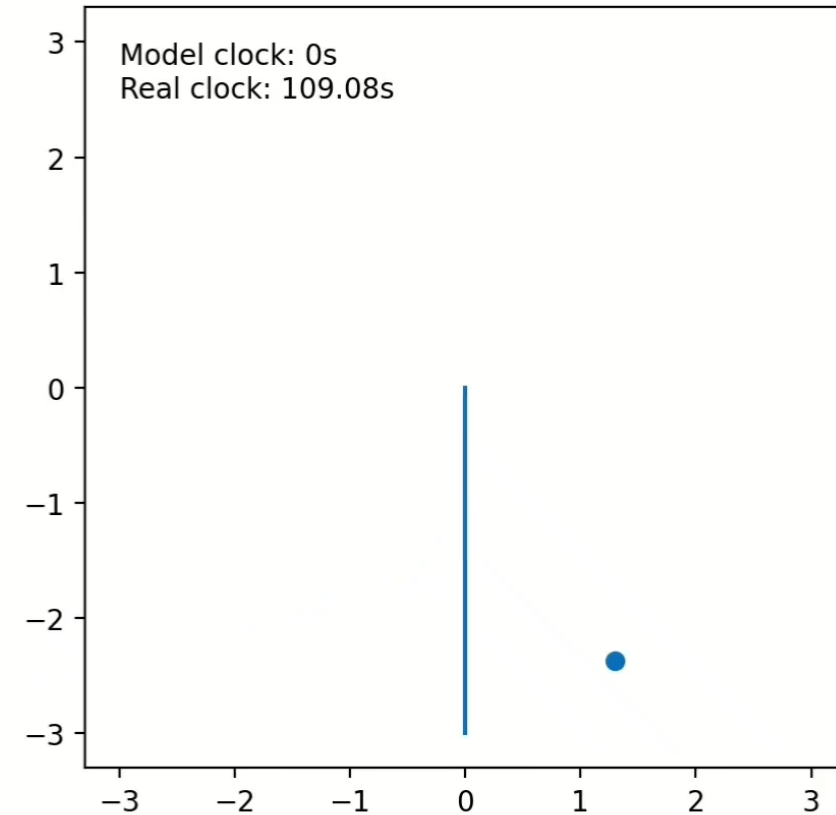
Analytical forward model

Neural network forward model

ASML

Thanks

Sae Na Na
DUV Reticle Stage MTD

Wilton CT