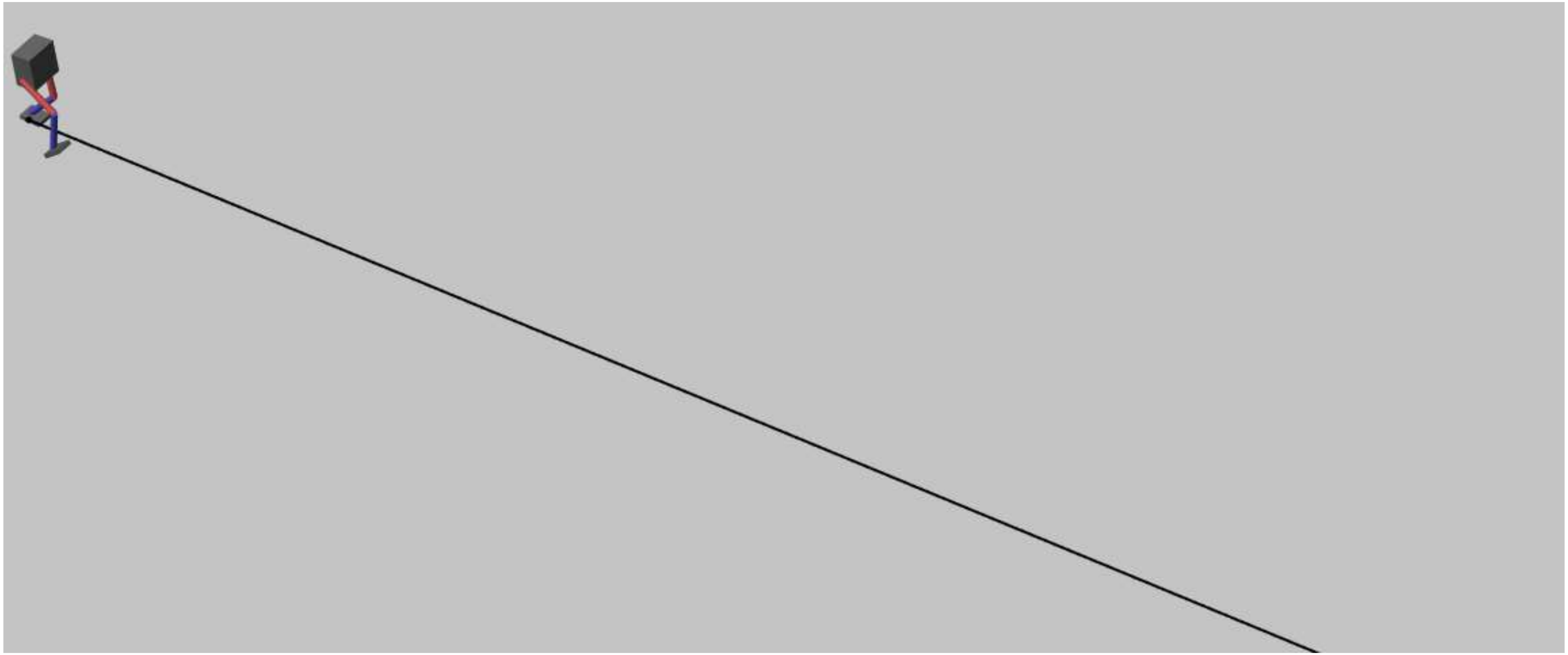
Reinforcement Learning: Leveraging Deep Learning for Controls

Aditya Baru

Goal: We hope you walk away knowing the answer to these questions

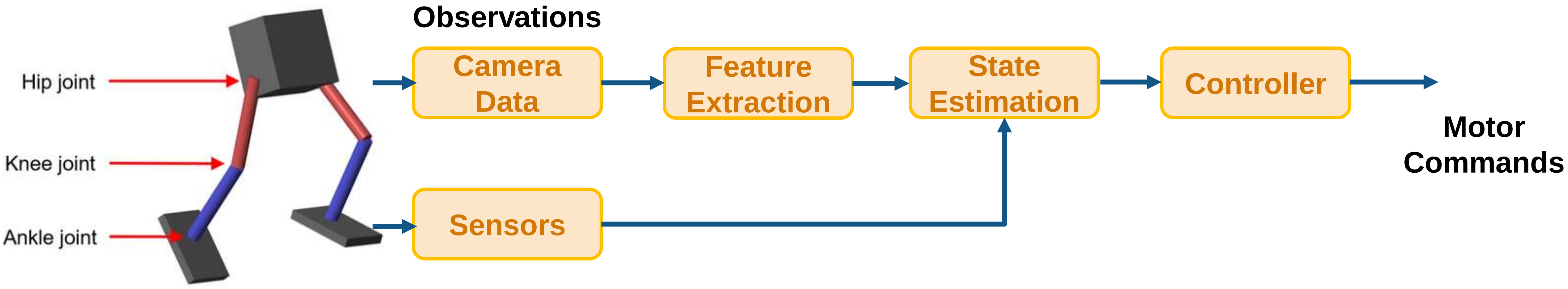
- What is reinforcement learning and why should I care about it?
- How do I set it up and solve it? [*from an engineer’s perspective*]
- What are some benefits and drawbacks?

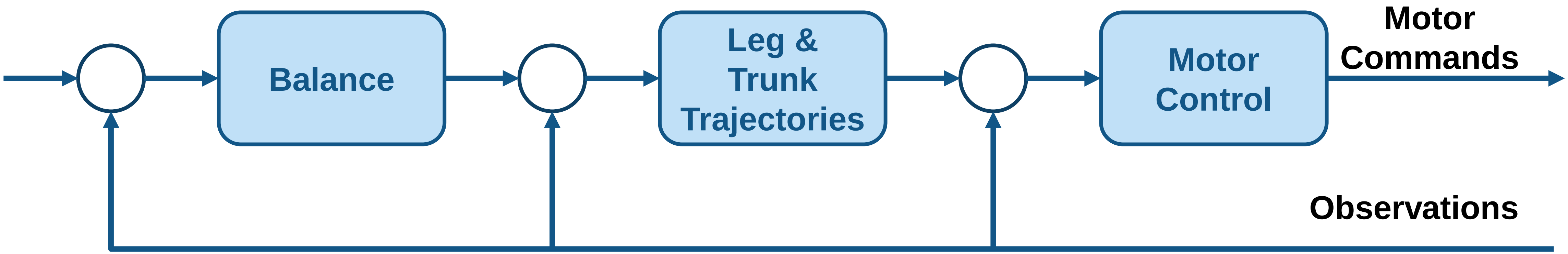
Why should you care about reinforcement learning?



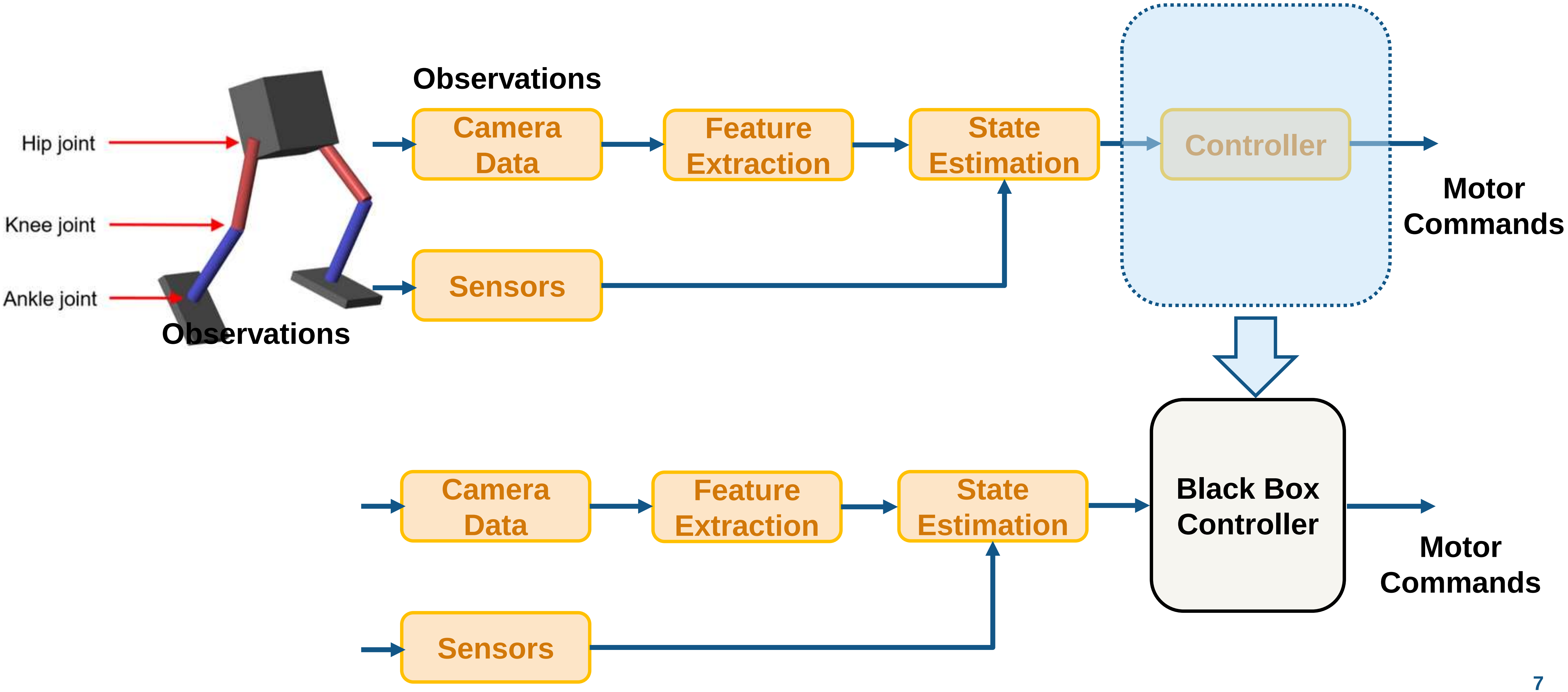
Teach a robot to follow a straight line using camera data

Let’s try to solve this problem the traditional way

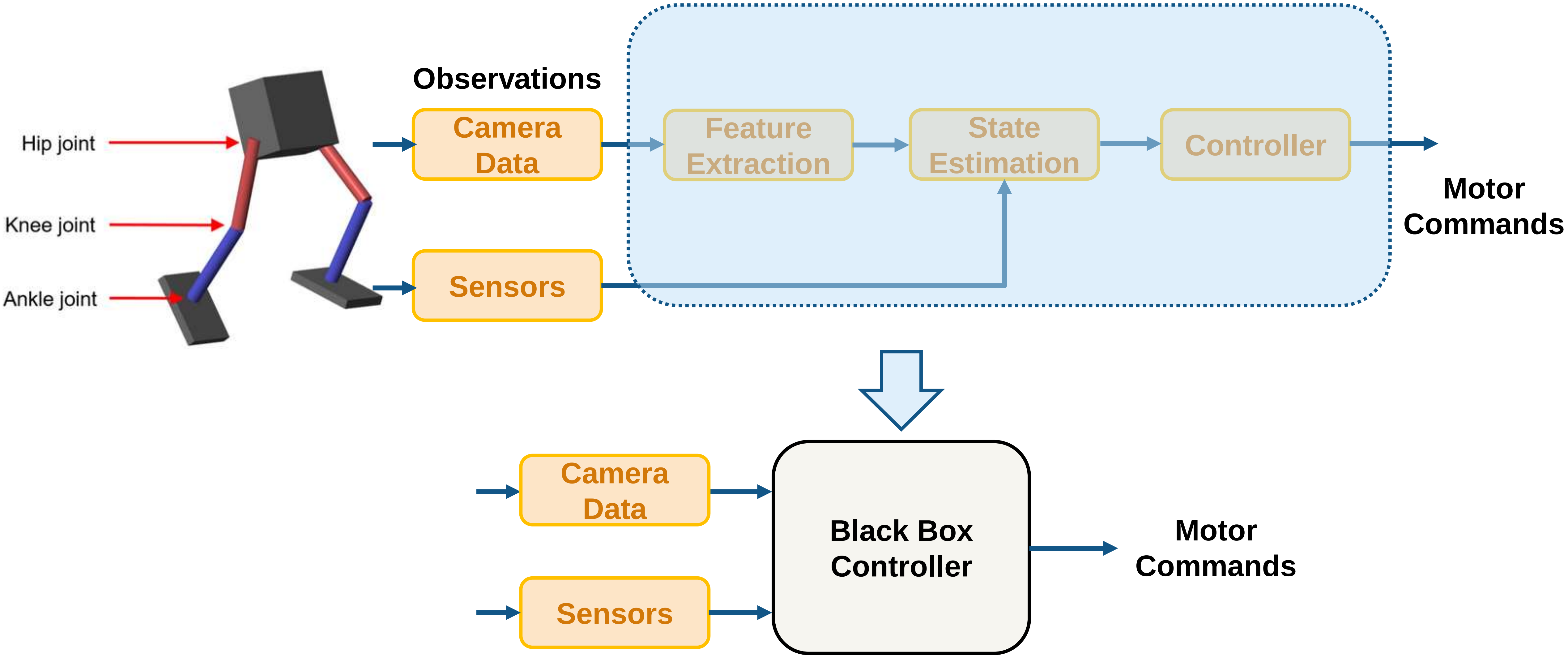




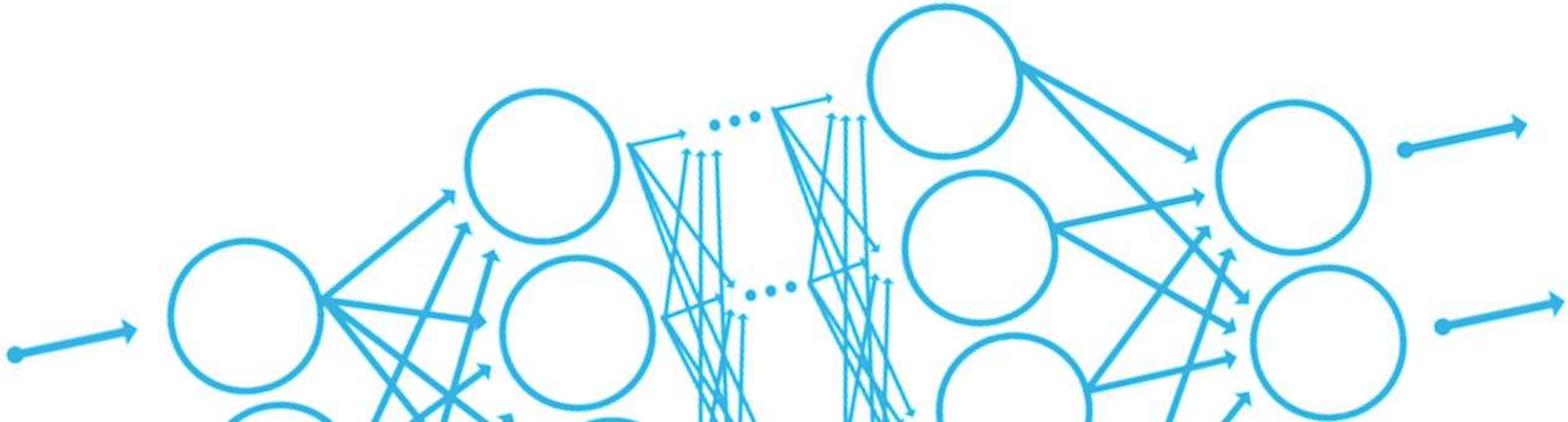
What is the alternative approach?



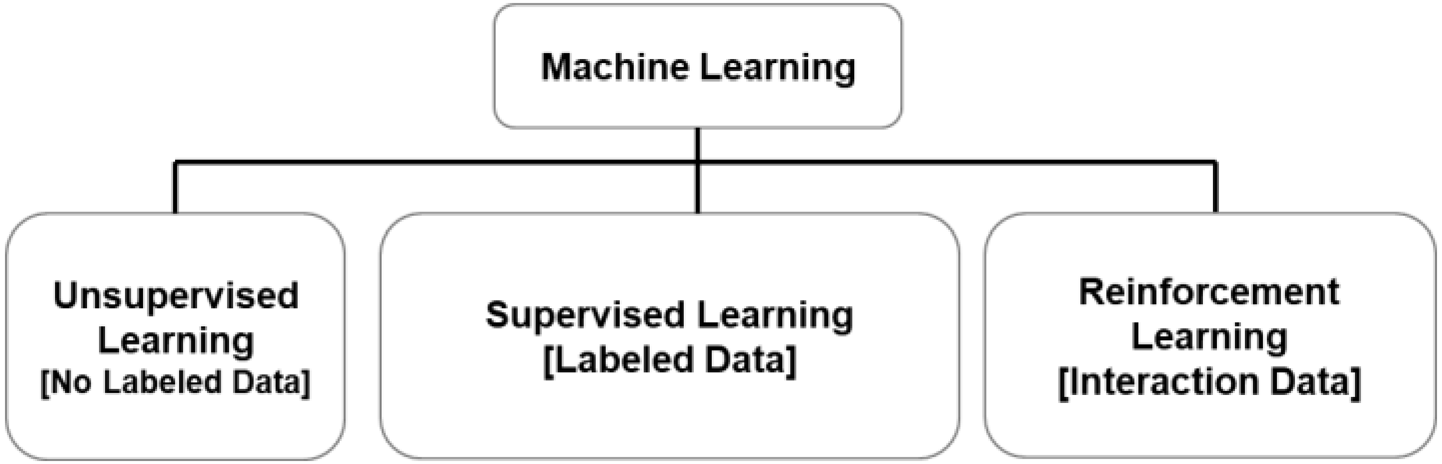
What is the alternative approach?



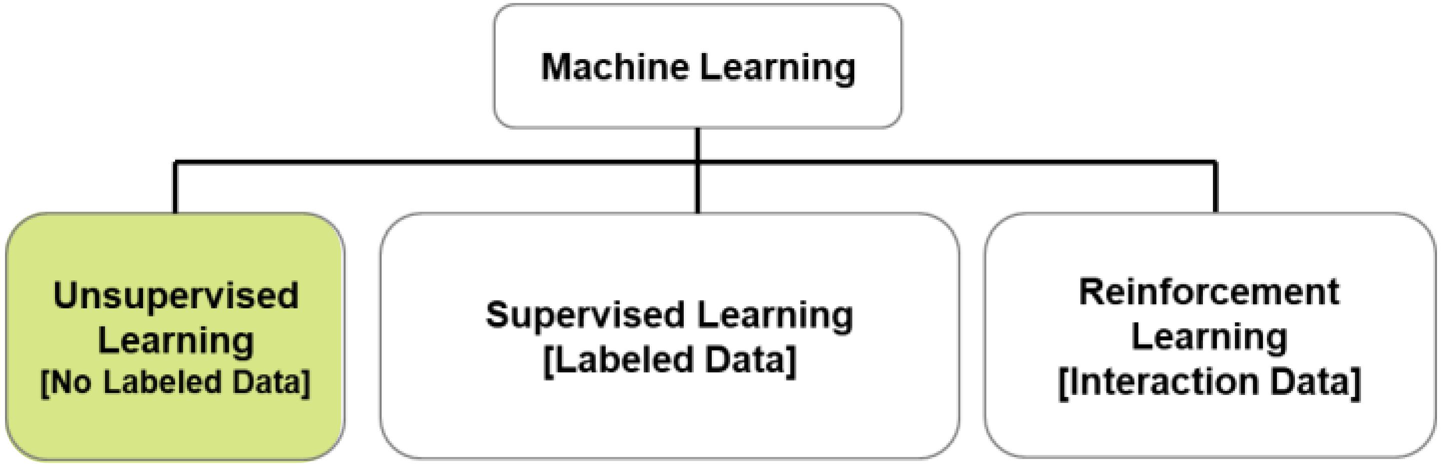
What is reinforcement learning?

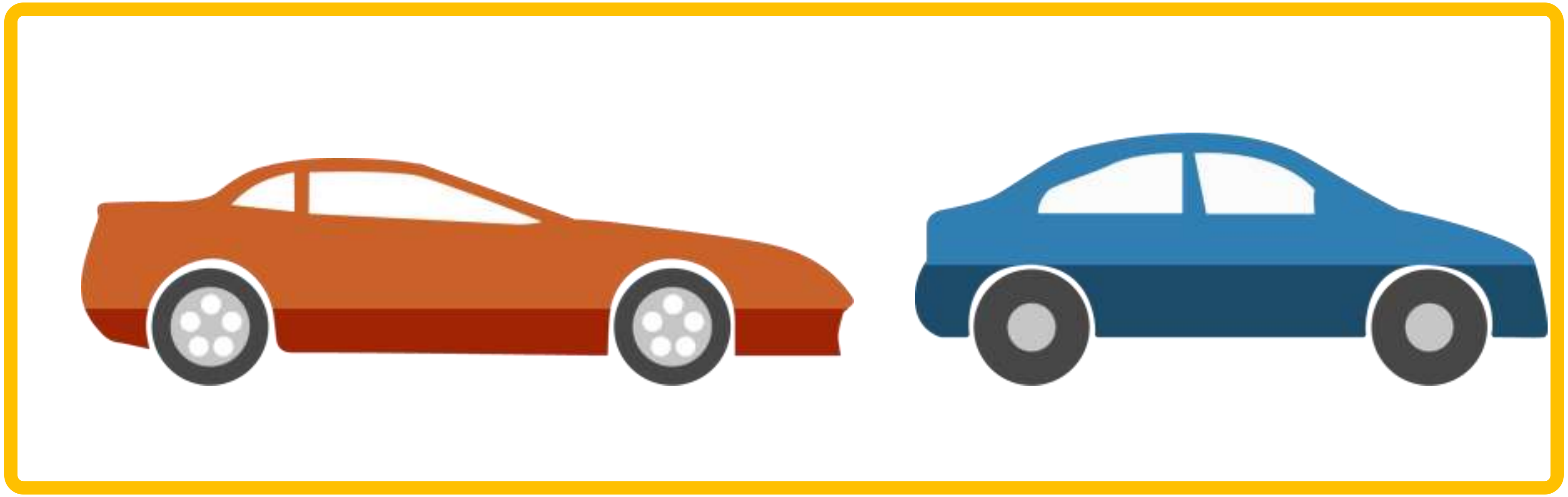


Reinforcement Learning vs Machine Learning vs Deep Learning

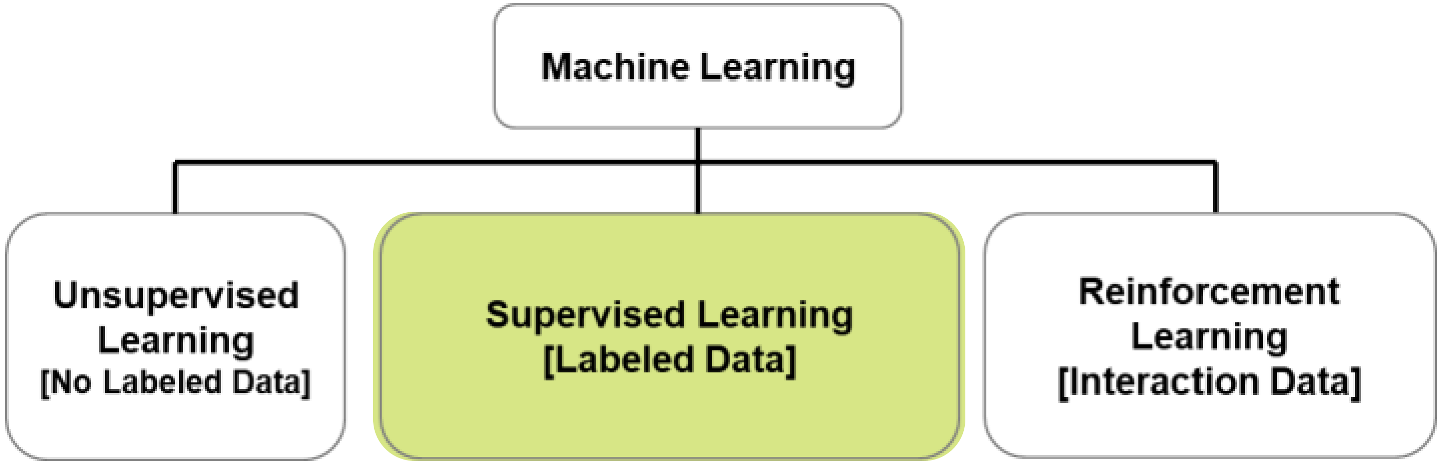


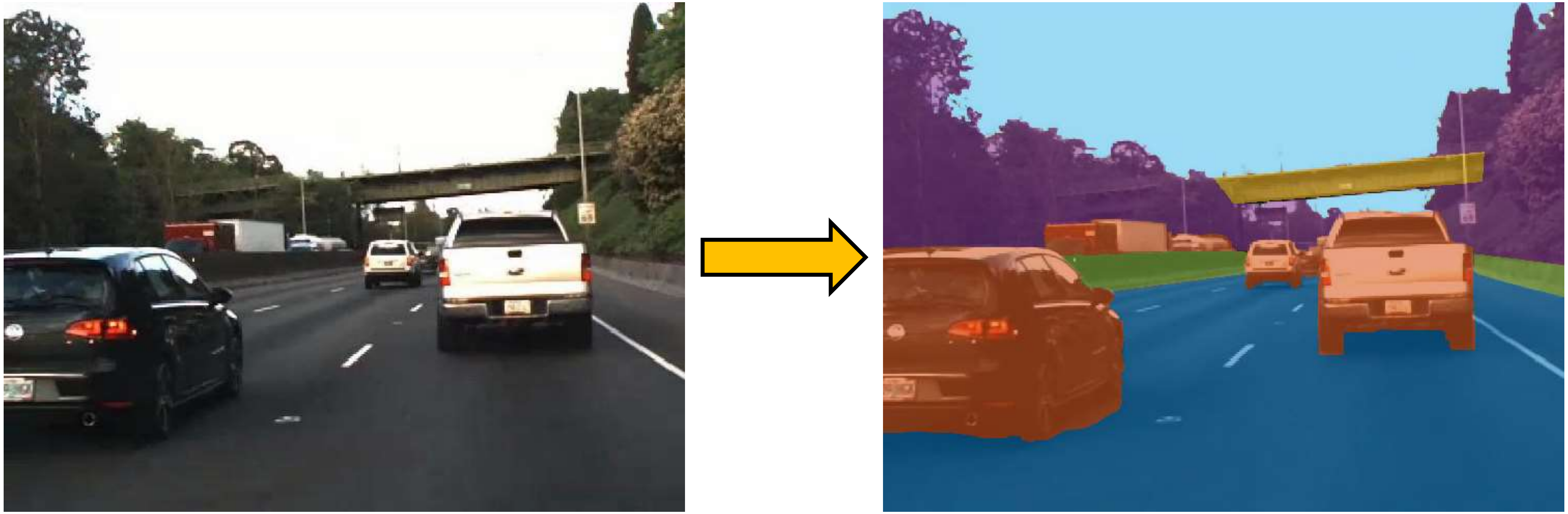
Reinforcement Learning vs Machine Learning vs Deep Learning





Reinforcement Learning vs Machine Learning vs Deep Learning



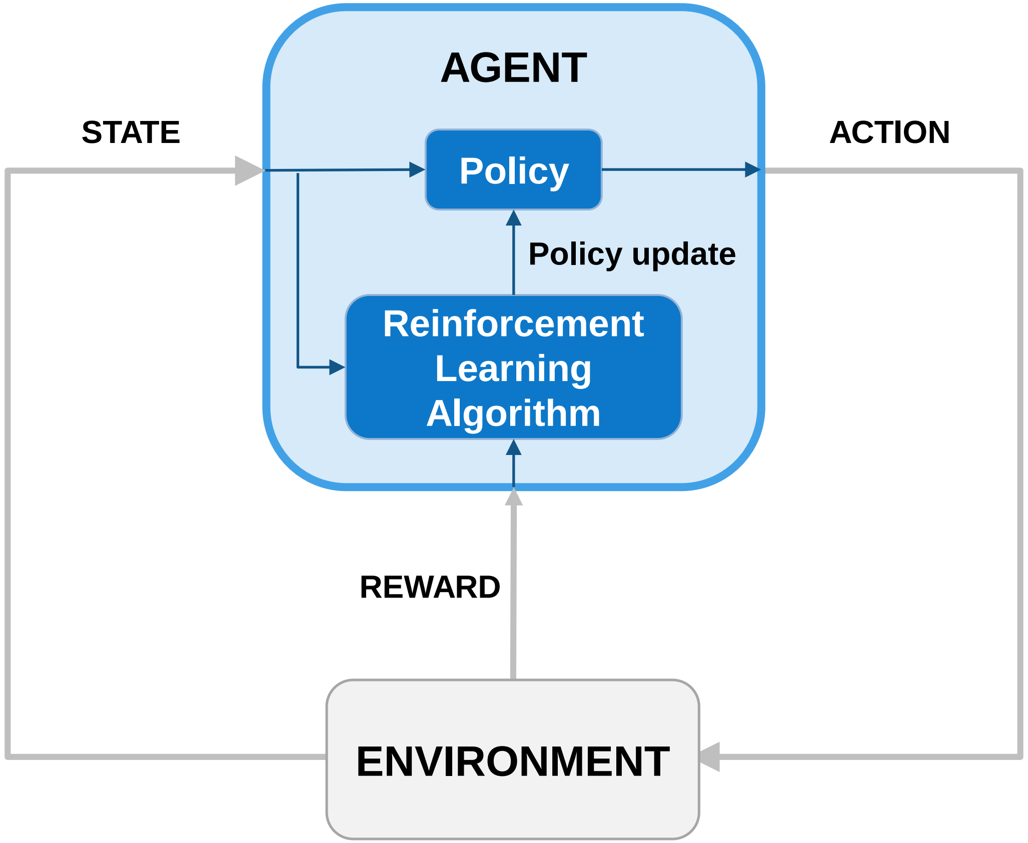


Reinforcement Learning vs Machine Learning vs Deep Learning

- **Machine Learning**
  - **Unsupervised Learning** [No Labeled Data]
  - **Supervised Learning** [Labeled Data]
  - **Reinforcement Learning** [Interaction Data]

A Practical Example of Reinforcement Learning

Training a Self-Driving Car

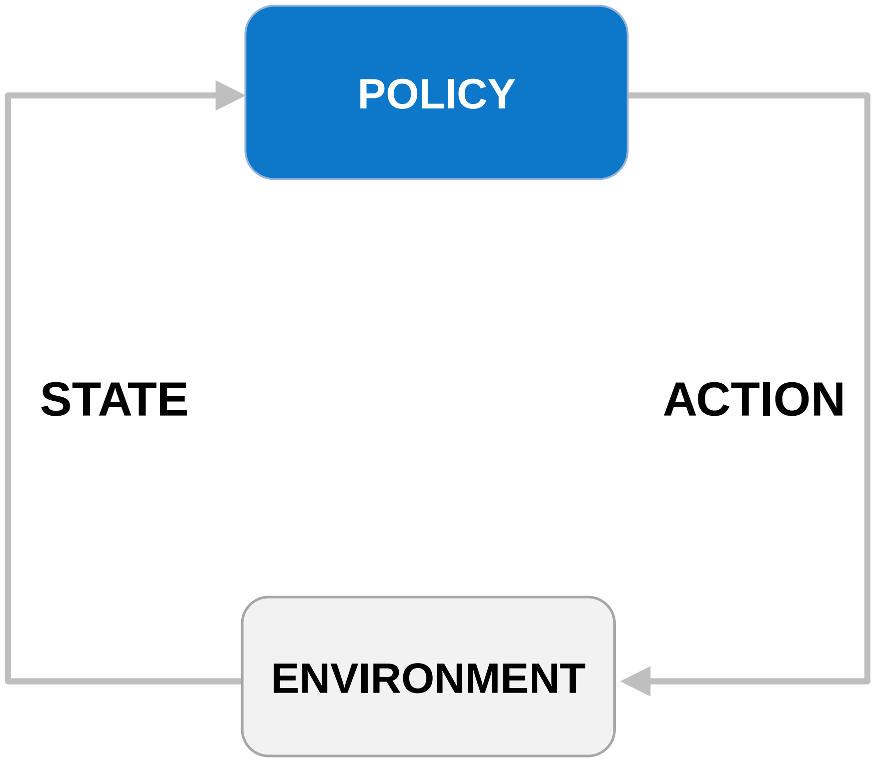


- Vehicle's computer learns how to drive… (**agent**)
- using sensor readings from LIDAR, cameras,… (**state**)
- that represent road conditions, vehicle position,… (**environment**)
- by generating steering, braking, throttle commands,… (**action**)
- based on an internal state-to-action mapping… (**policy**)
- that tries to optimize driver comfort & fuel efficiency… (**reward**).

- The policy is updated through repeated trial-and-error by a **reinforcement learning algorithm**

A Practical Example of Reinforcement Learning

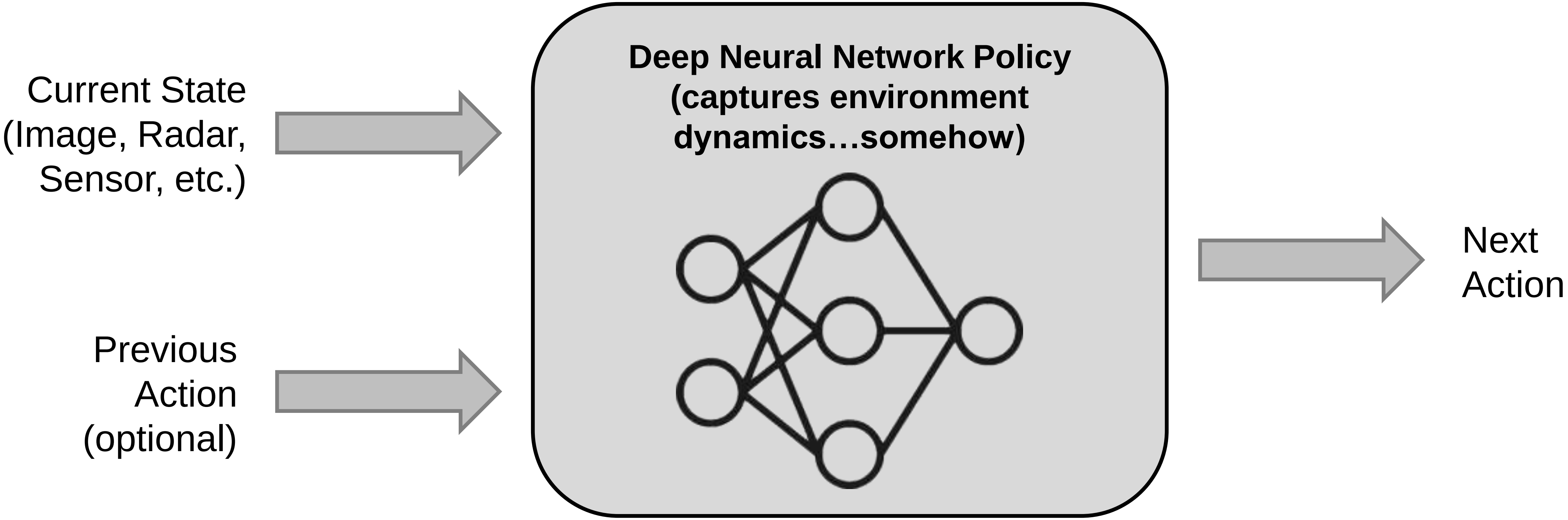
A Trained Self-Driving Car Only Needs A Policy To Operate



- Vehicle's computer uses the final state-to-action mapping… (**policy**)
- to generate steering, braking, throttle commands,… (**action**)
- based on sensor readings from LIDAR, cameras,… (**state**)
- that represent road conditions, vehicle position,… (**environment**)

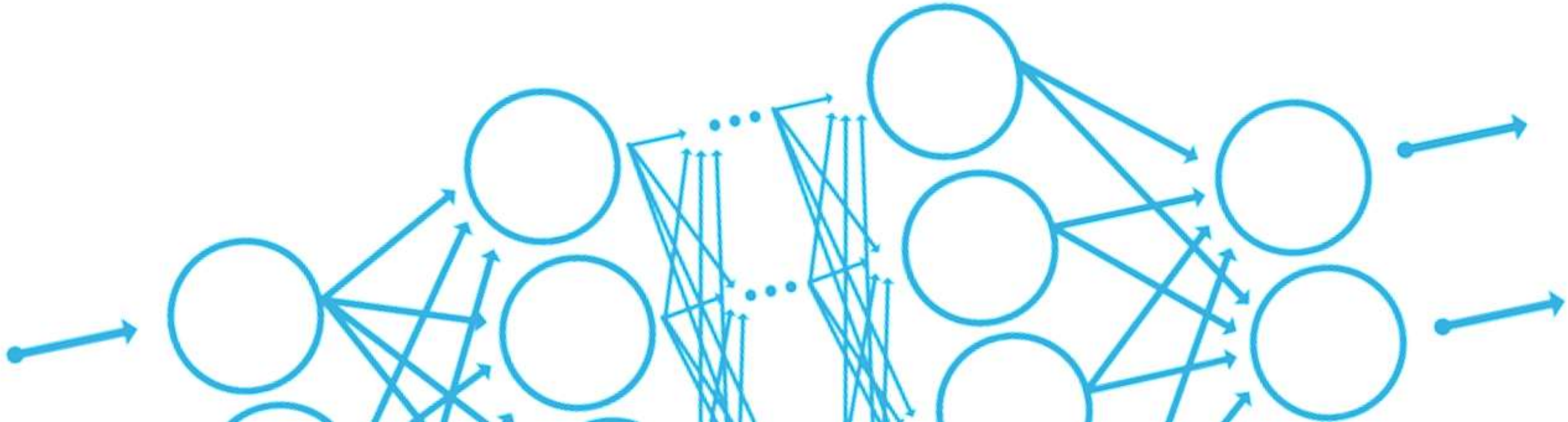
By definition, this trained policy is optimizing driver comfort & fuel efficiency

A deep neural network trained using reinforcement learning is a black-box model that determines the best possible action

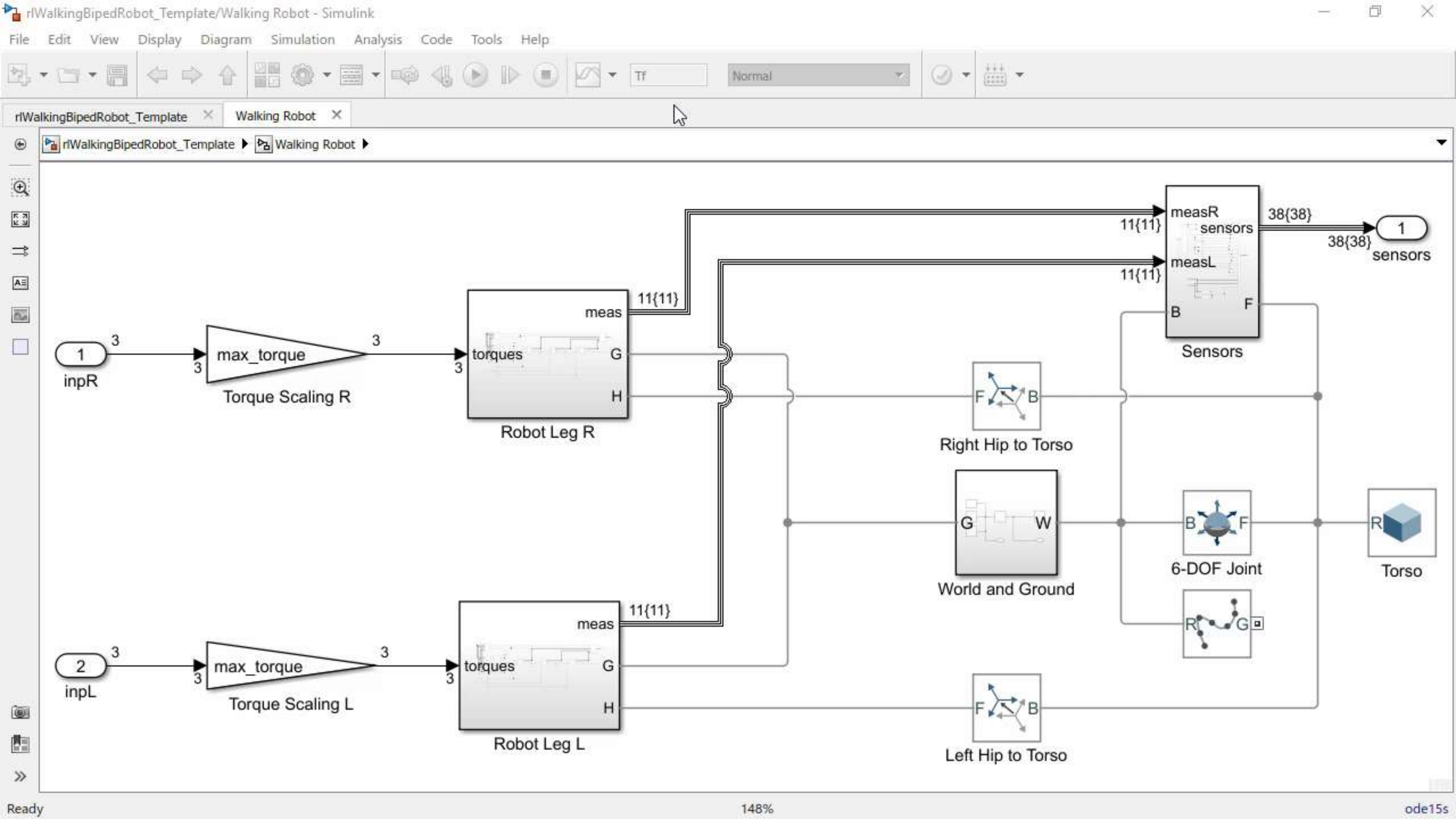


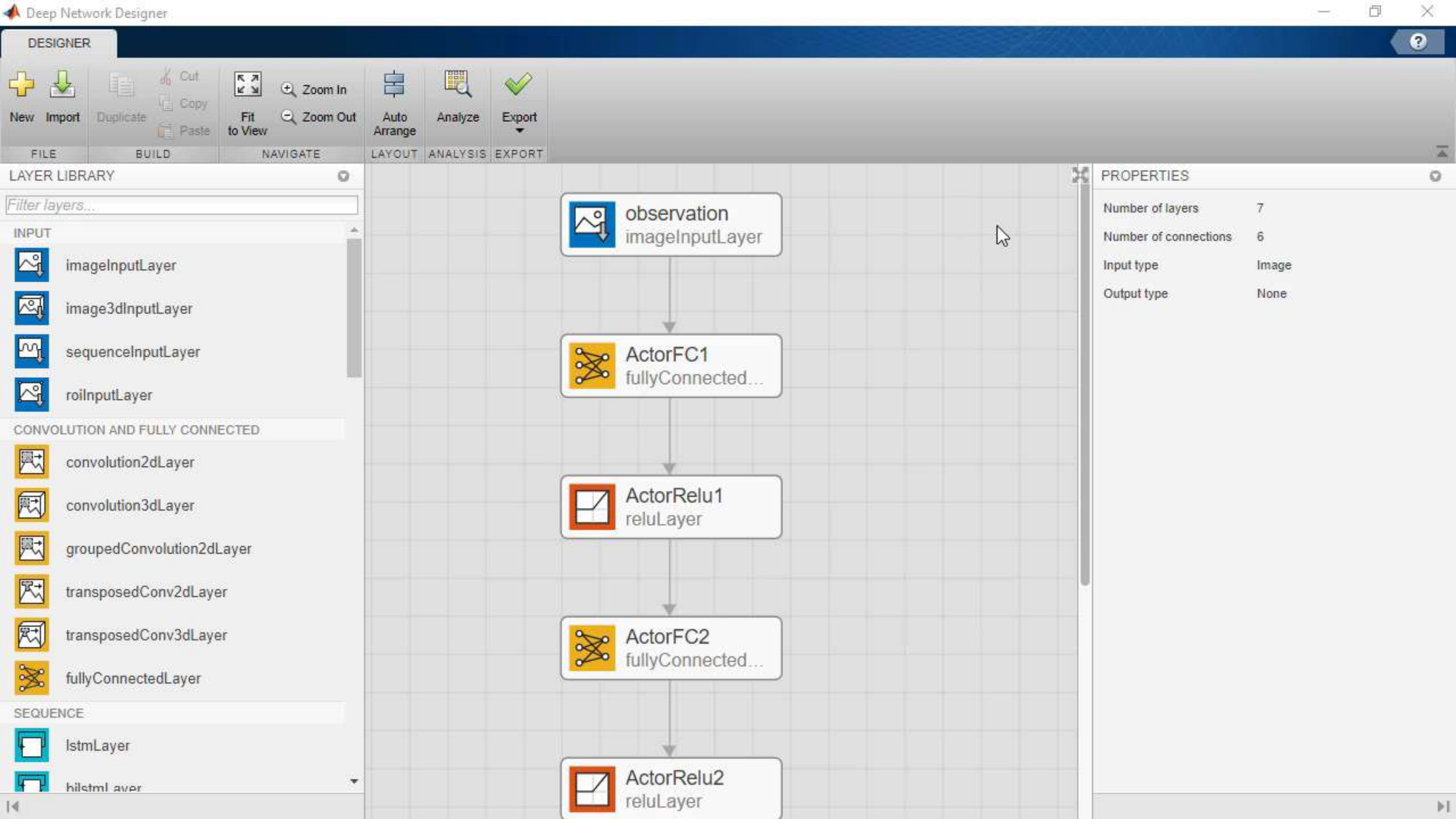
By representing policies using deep neural networks, we can solve problems for **complex, non-linear systems** (continuous or discrete) by directly using **data that traditional approaches cannot use easily**

How do I set it up and solve it?

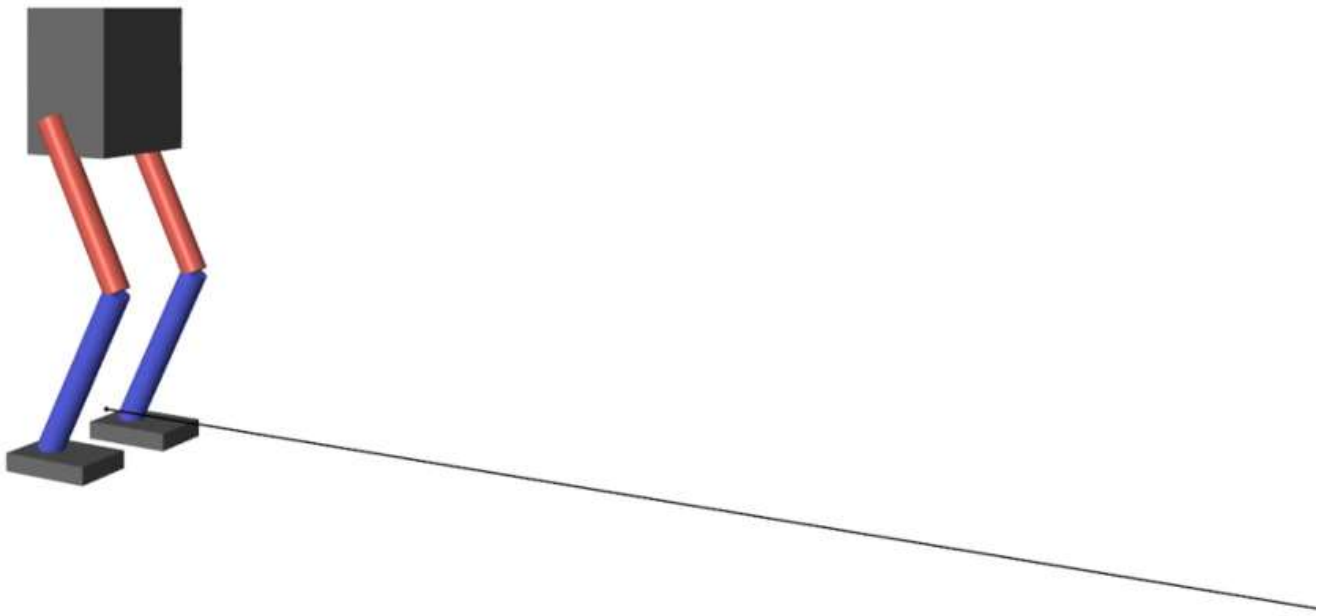


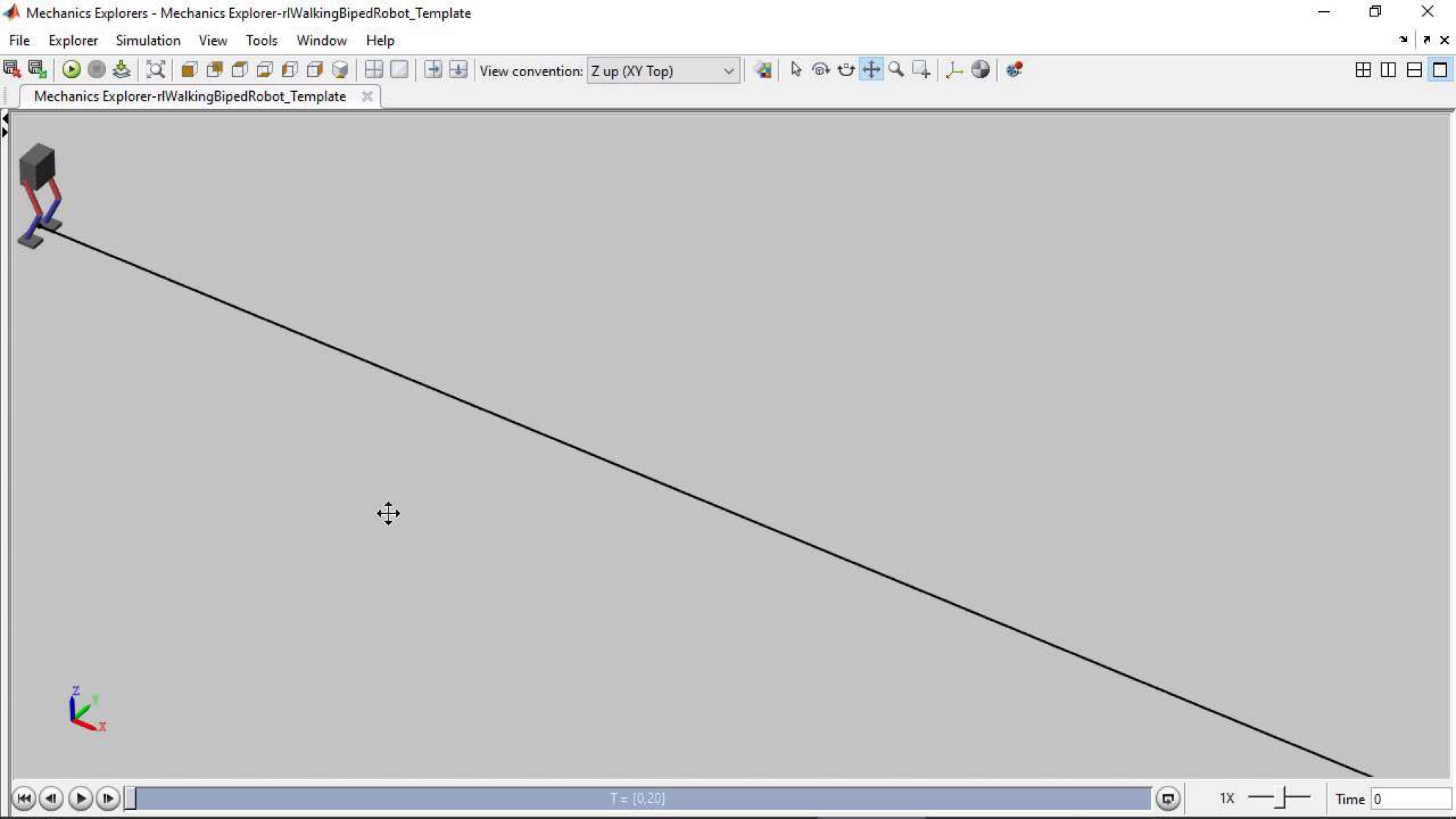
Reinforcement Learning Workflow


rlWalkingBipedRobot_Template/Walking Robot - Simulink
File
Edit
View
Display
Diagram
Simulation
Analysis
Code
Tools
Help
Tf
Normal
rlWalkingBipedRobot_Template
Walking Robot
rlWalkingBipedRobot_Template
Walking Robot
1
inpR
max_torque
Torque Scaling R
torques
meas
G
H
Robot Leg R
11{11}
2
inpL
max_torque
Torque Scaling L
torques
meas
G
H
Robot Leg L
11{11}
measR
sensors
measL
B
F
Sensors
11{11}
11{11}
38{38}
38{38}
1
sensors
F
B
Right Hip to Torso
G
W
World and Ground
B
F
6-DOF Joint
R
Torso
R
G
F
B
Left Hip to Torso
3
3
3
3
3
3
3
3
Ready
148%
ode15s


Deep Network Designer
DESIGNER
New
Import
Duplicate
Cut
Copy
Paste
Fit to View
Zoom In
Zoom Out
Auto Arrange
Analyze
Export
FILE
BUILD
NAVIGATE
LAYOUT
ANALYSIS
EXPORT
LAYER LIBRARY
Filter layers...
INPUT
imageInputLayer
image3dInputLayer
sequenceInputLayer
roiInputLayer
CONVOLUTION AND FULLY CONNECTED
convolution2dLayer
convolution3dLayer
groupedConvolution2dLayer
transposedConv2dLayer
transposedConv3dLayer
fullyConnectedLayer
SEQUENCE
lstmLayer
observation
imageInputLayer
ActorFC1
fullyConnected...
ActorRelu1
reluLayer
ActorFC2
fullyConnected...
ActorRelu2
reluLayer
PROPERTIES
Number of layers 7
Number of connections 6
Input type Image
Output type None

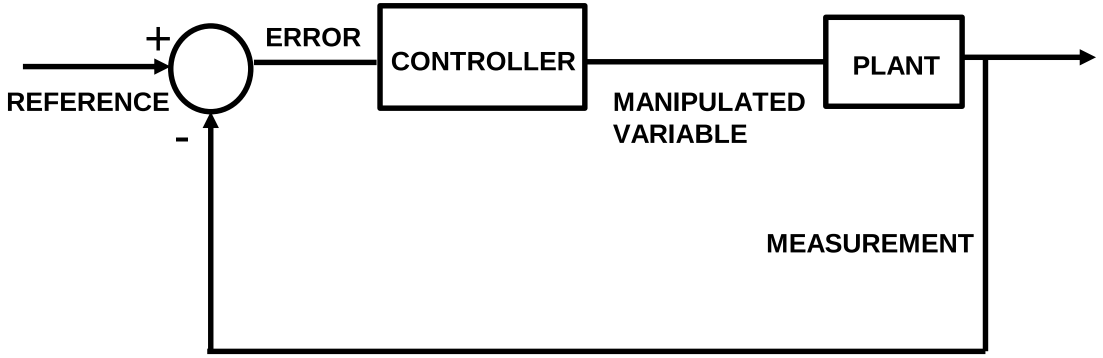
Steps in the Reinforcement Learning Workflow



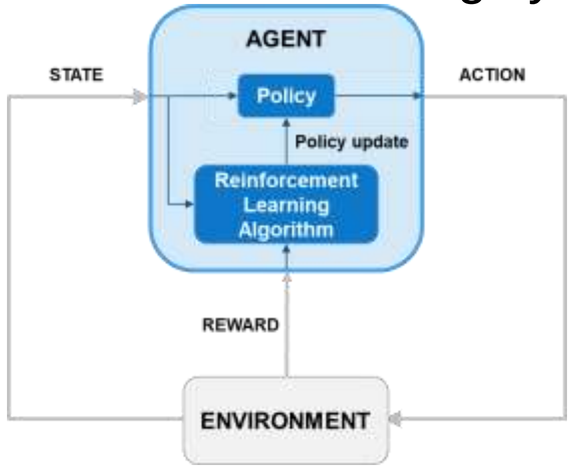

Mechanics Explorers - Mechanics Explorer-rlWalkingBipedRobot_Template
File Explorer Simulation View Tools Window Help
View convention: Z up (XY Top)
Mechanics Explorer-rlWalkingBipedRobot_Template
T = [0,20]
1X
Time 0

Reinforcement Learning vs Controls

Control system



Reinforcement learning system



Control system	Reinforcement learning system
Adaptation mechanism	RL Algorithm
Error/Cost function	Reward
Manipulated variable	Action
Measurement	Observation
Plant	Environment
Controller	Policy

Reinforcement learning has parallels to **control system design**

Pop Quiz: When would you use Reinforcement Learning?

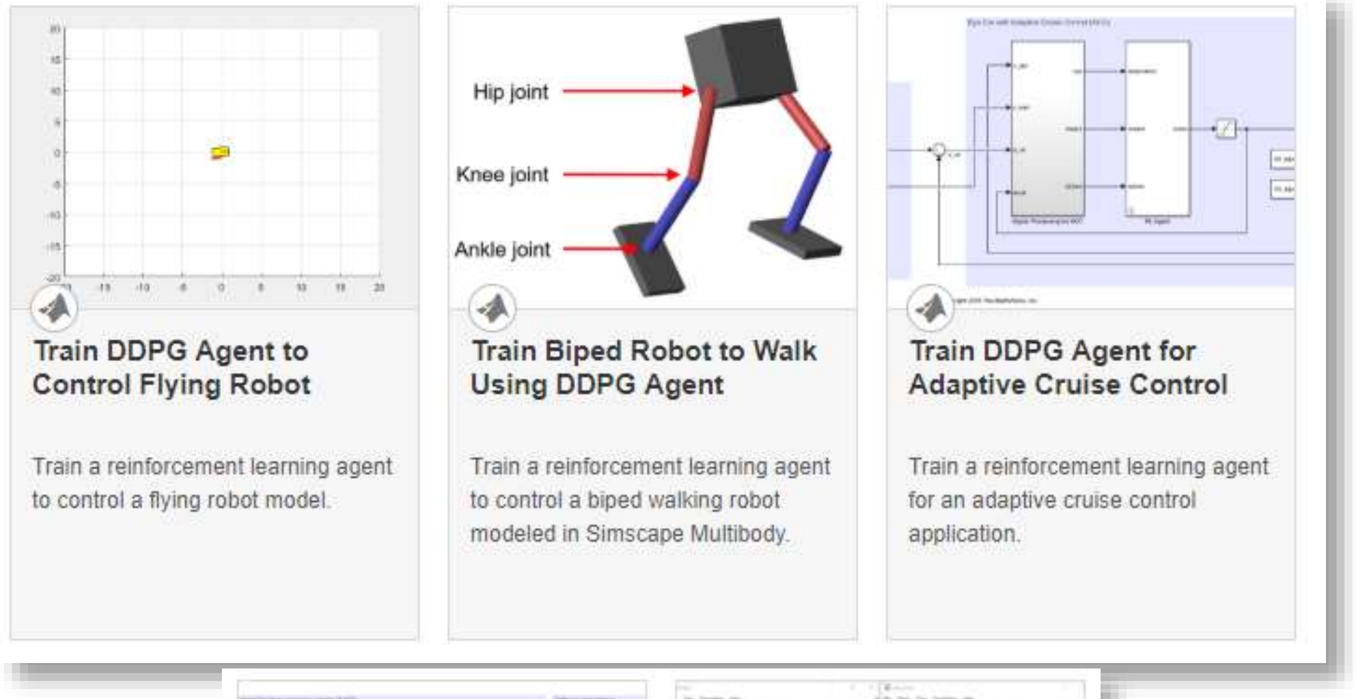
	Controller Capability	Computational Cost in Training/Tuning	Computational Cost in Deployment
PID	Low	Low	Low
Model Pred Control	High	Low	High
Reinforcement Learning	High	High	Medium

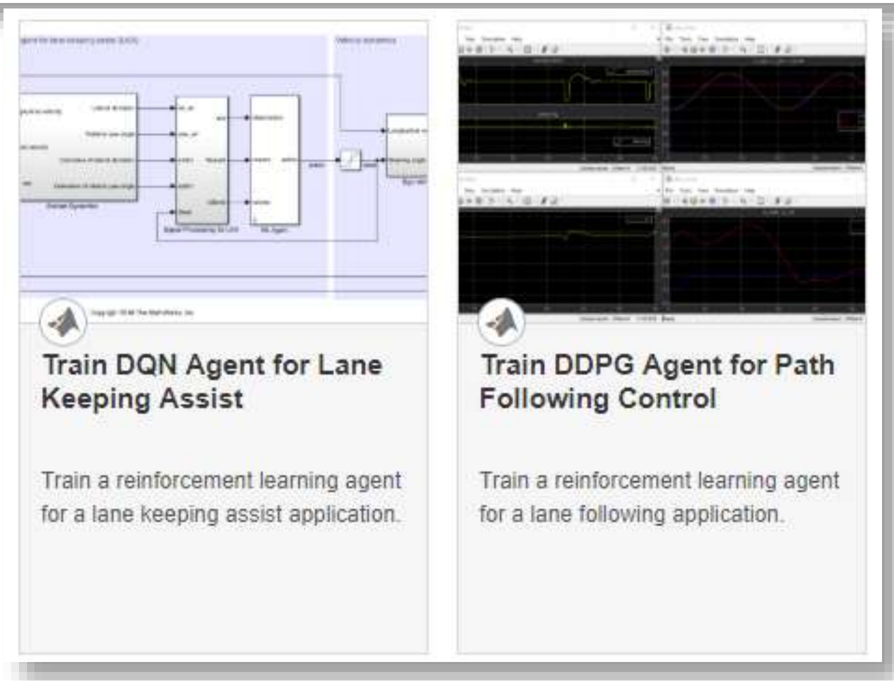
Reinforcement learning might be a good fit if

- An environment model is available (trial & error on hardware can be expensive), and
- Training/tuning time is **not** critical for the application, and
- Uncertain environments or nonlinear environments

Automotive Applications

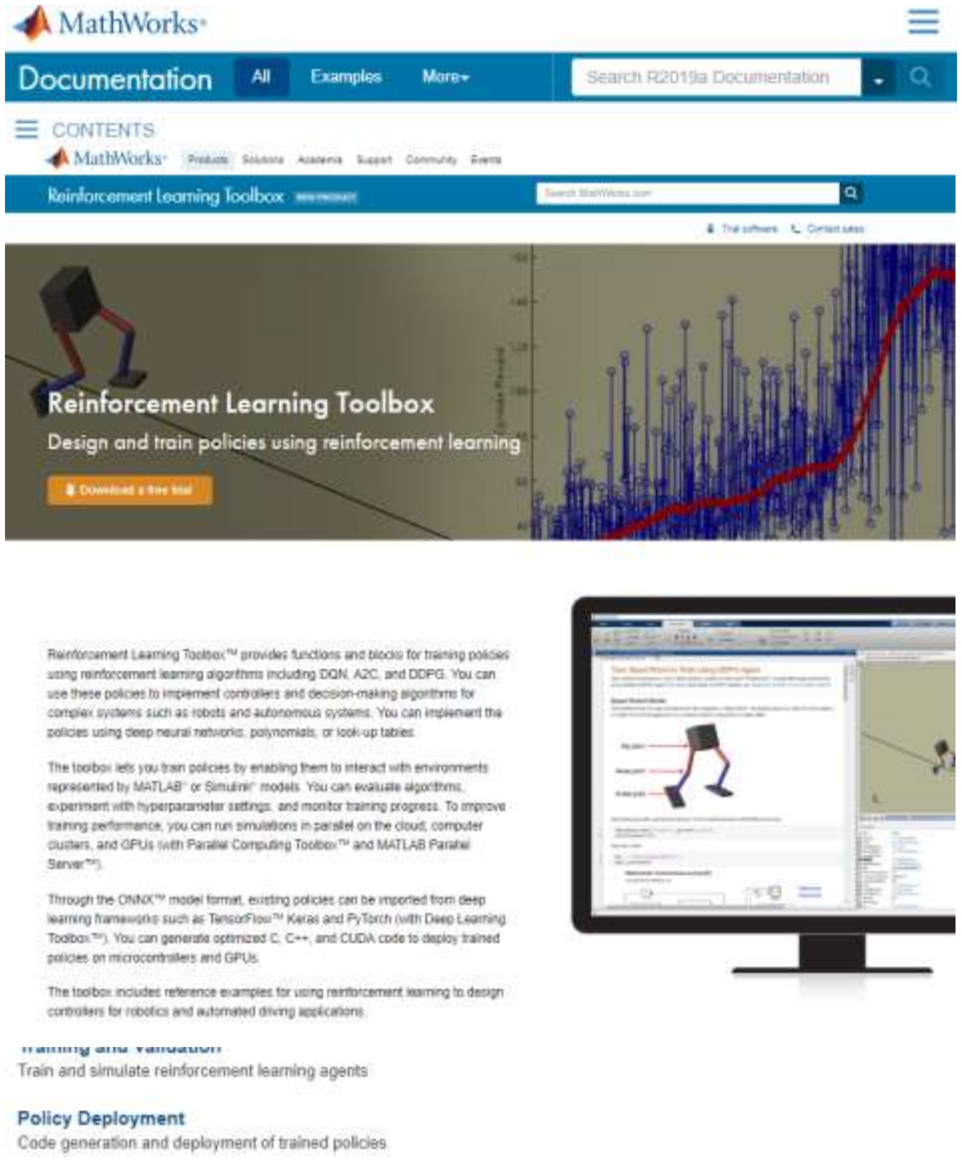
- Controller Design
- Lane Keep Assist
- Adaptive Cruise Control
- Path Following Control
- Trajectory Planning



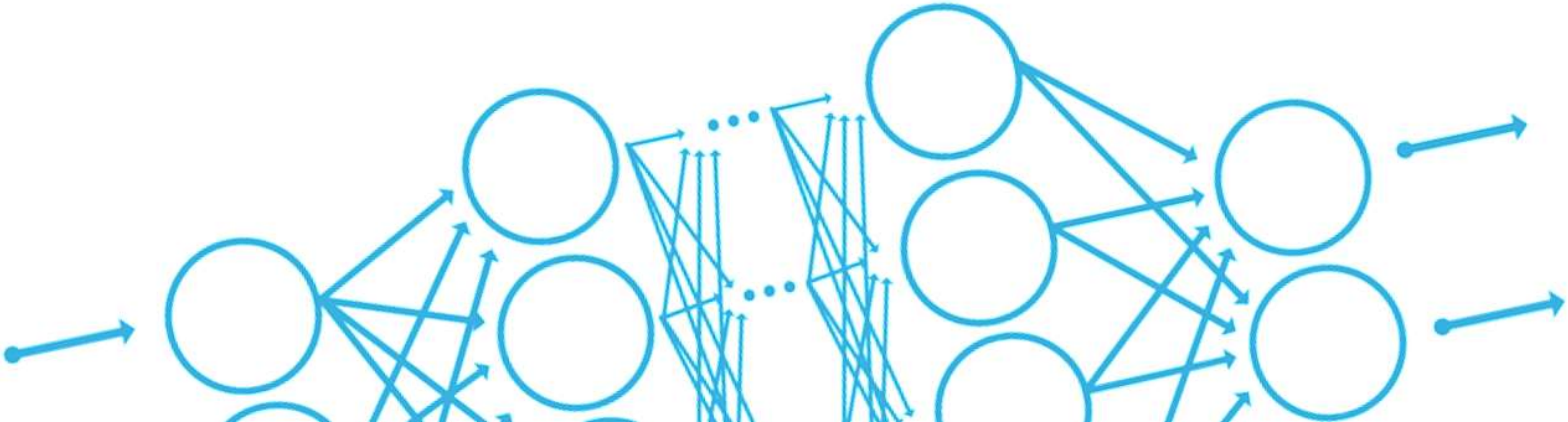


Reinforcement Learning Toolbox
New in R2019a

- Built-in and custom algorithms for reinforcement learning
- Environment modeling in MATLAB and Simulink
- Deep Learning Toolbox support for designing policies
- Training acceleration through GPUs and cloud resources
- Deployment to embedded devices and production systems
- Reference examples for getting started



Takeaways

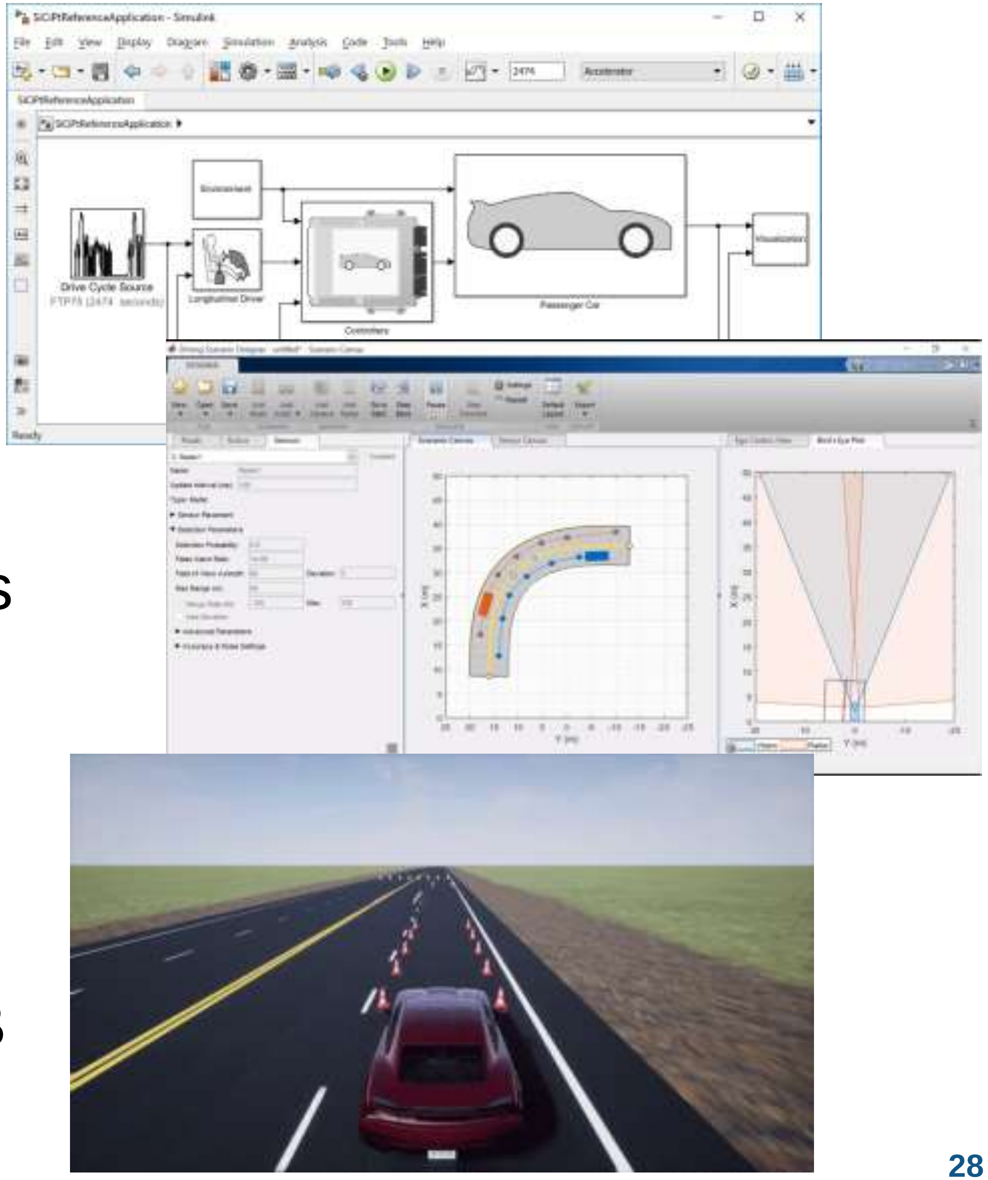


Simulation and Virtual Models are a Key Aspect of Reinforcement Learning

- Reinforcement learning needs **<u>a lot</u>** of data (*sample inefficient*)
  - Training on hardware can be prohibitively expensive and dangerous

- Virtual models allow you to simulate conditions hard to emulate in the real world
  - This can help develop a more robust solution

- Many of you have already developed MATLAB and Simulink models that can be reused

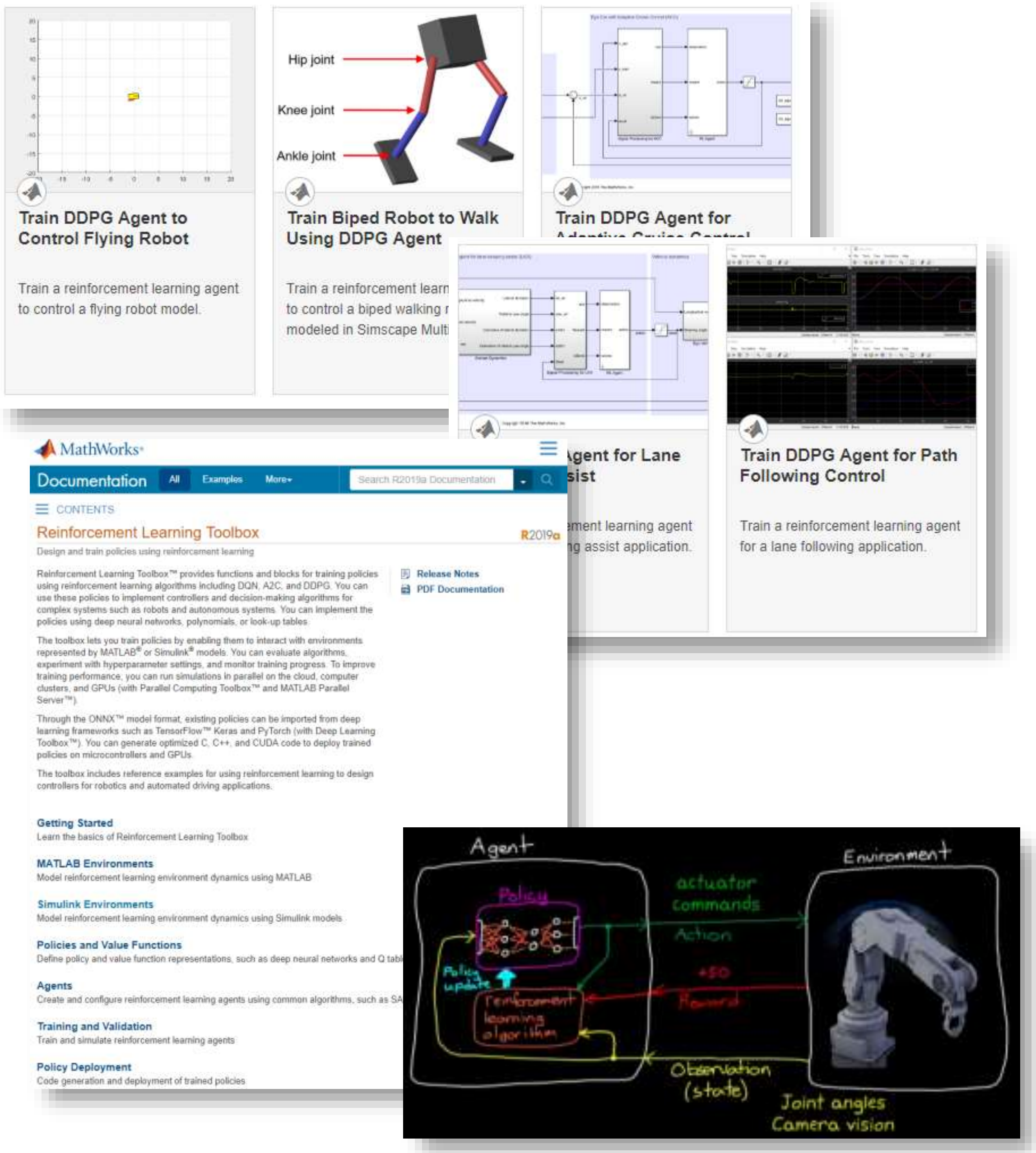


Pros & Cons of Reinforcement Learning

Pros	Cons
No need to collect data before training	A lot of simulation trials required
Opens up AI applications intractable today	Training may not converge
Complex end-to-end solutions can be developed (e.g. camera input→ car steering wheel)	Reward signal design, network layer structure & hyperparameter tuning can be challenging
Suitable for uncertain, nonlinear environments	No performance guarantees
Virtual models allow simulations of varying conditions and training parallelization	Further training might be necessary after deployment on real hardware
Everyone is excited about it as it appears to be a silver bullet for all problems	

Resources

- Examples for automotive and autonomous system applications
- Documentation written for engineers and domain experts
- Tech Talk video series on reinforcement learning concepts for engineers



Extra Slides