Reinforcement Learning

C. Karen Liu
(Adapted from Wenhao Yu's slides)

Go


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO

Video games

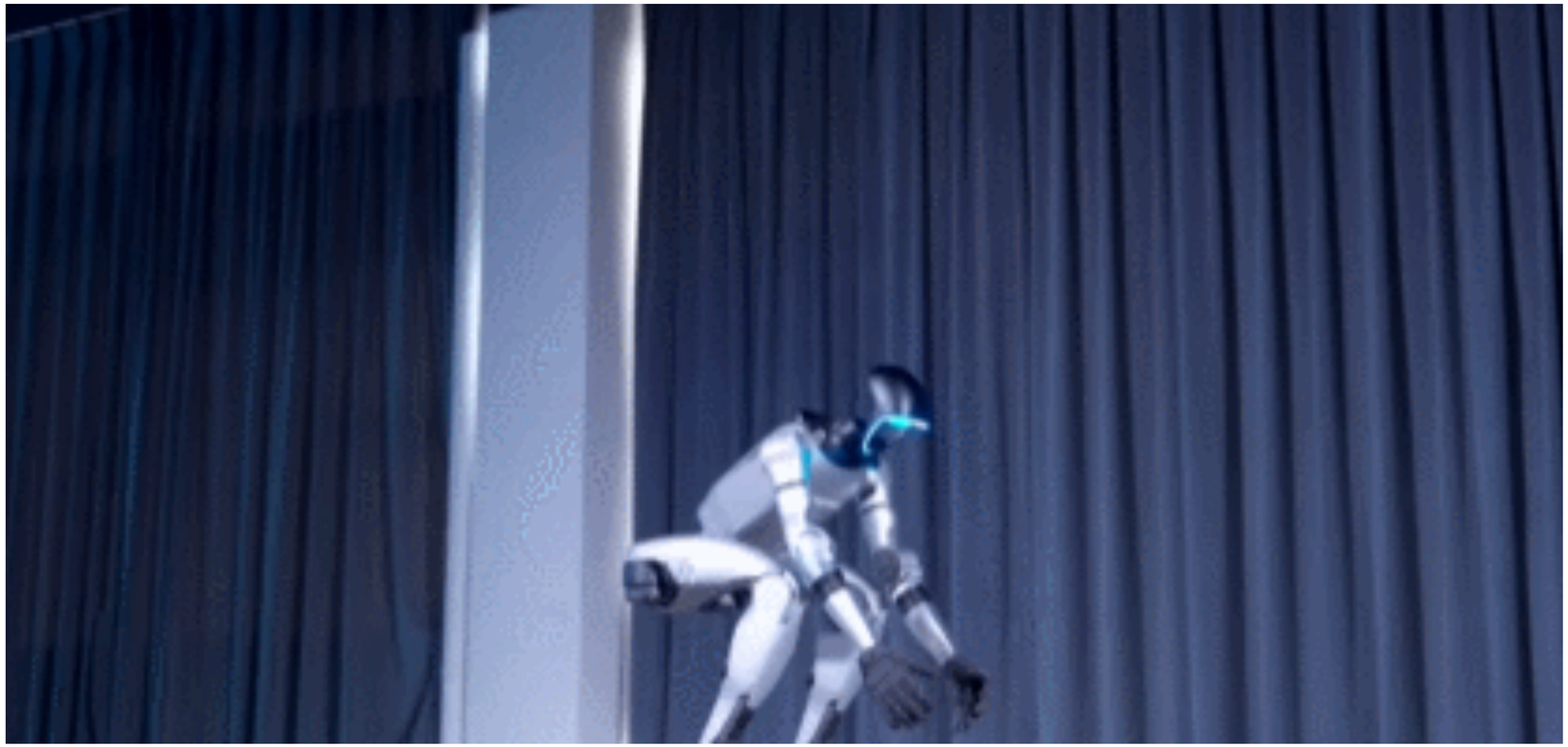

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
LEARNING CURVE
Self-taught AI software attains human-level performance in video games
PAGES 406 & 529


OpenAI-5 (Bot)
OpenAI-4 (Bot)
OpenAI-3 (Bot)


Workers: 3/3
Workers: 25/18
Workers: 3/3
AlphaStar
177 /200
SUPPLY
945 +2015
MINERALS
758 +873
GAS
64
WORKERS
113
ARMY
940
APM
LiquidTLO
147 /172
335 +1595
442 +1030
61
86
1377

Robotics


The international journal of science / 31 August 2023
nature
DRONE RACER
AI pilot beats human champions in aerial contest
Offset agreement
Overhaul pricing of carbon credits to help fund climate projects
Dining companions
Corals devour algal partners when food supplies run low
Sentence synthesis
Brain implants show promise in rendering speech from thoughts



LLM

What we will learn today

- What is RL?

What we will learn today

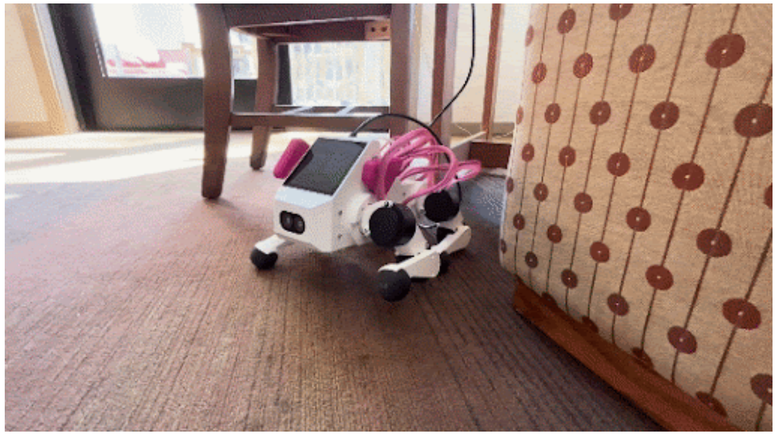
- What is RL?
- How to formulate a RL problem

What we will learn today

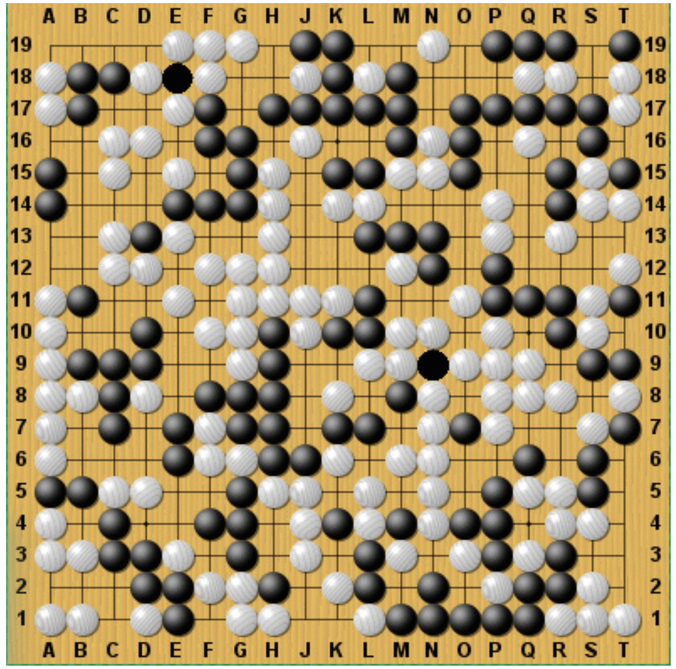
- What is RL?
- How to formulate a RL problem
- How to solve a RL problem

What we will learn today

- What is RL?
- How to formulate a RL problem?
- How to solve a RL problem?
- How to teach Pupper to walk using RL?



What is RL?

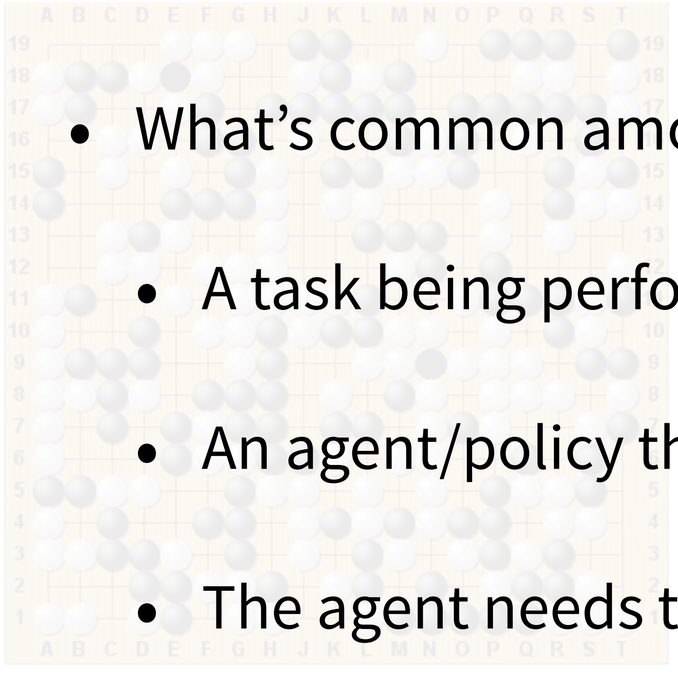


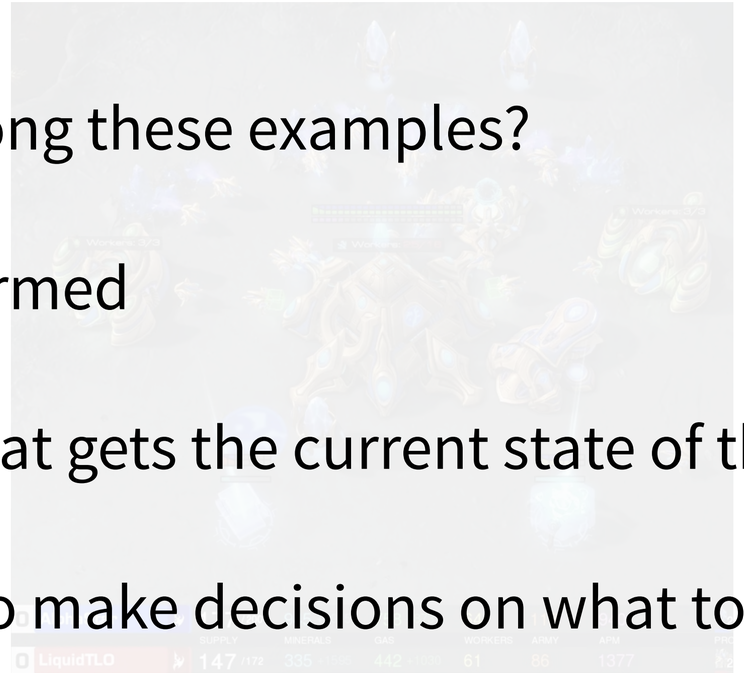
play go

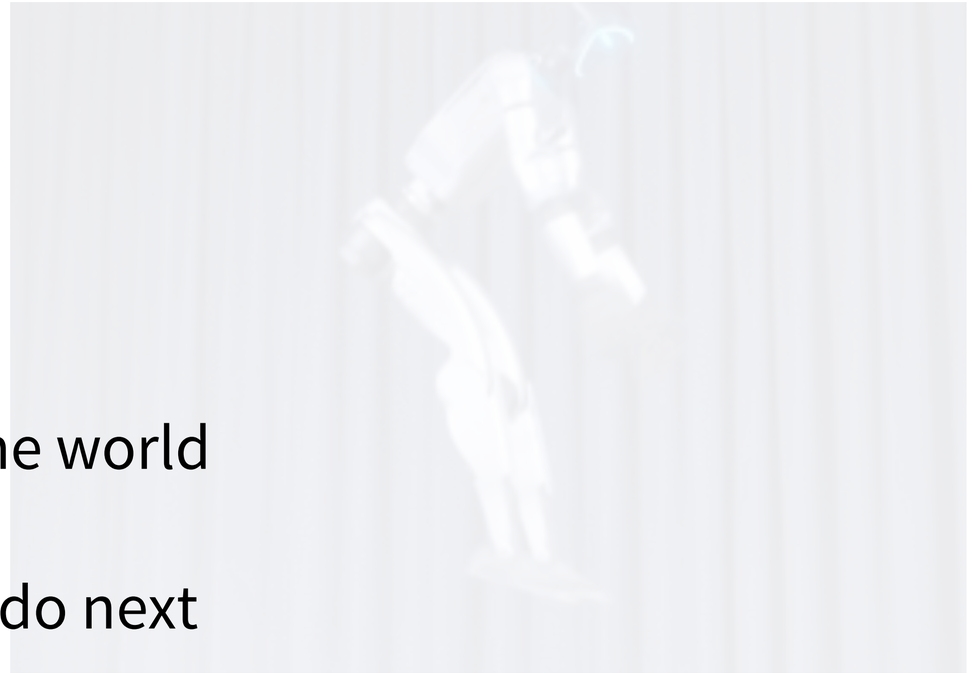
play Starcraft

control robot

What is RL?







play go

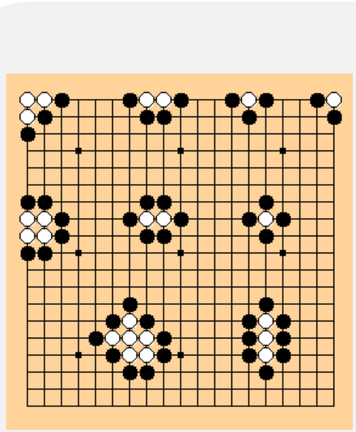
play Starcraft

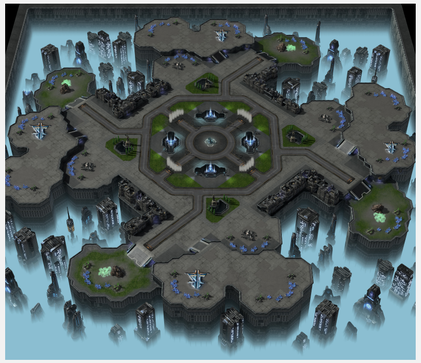
control robot

- What's common among these examples?
  - A task being performed
  - An agent/policy that gets the current state of the world
  - The agent needs to make decisions on what to do next
  - No clear answer for individual actions, but a reward for overall correctness

Markov Decision Process

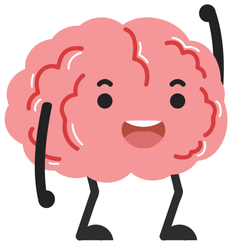
Markov Decision Process





environment

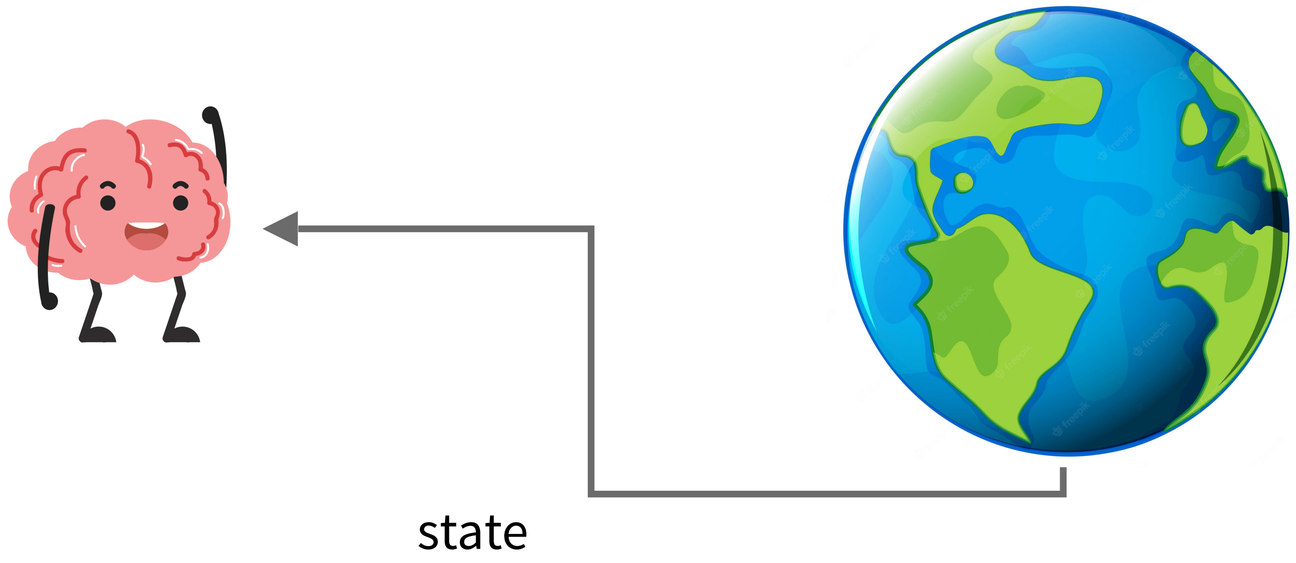
Markov Decision Process



environment

agent/policy

Markov Decision Process



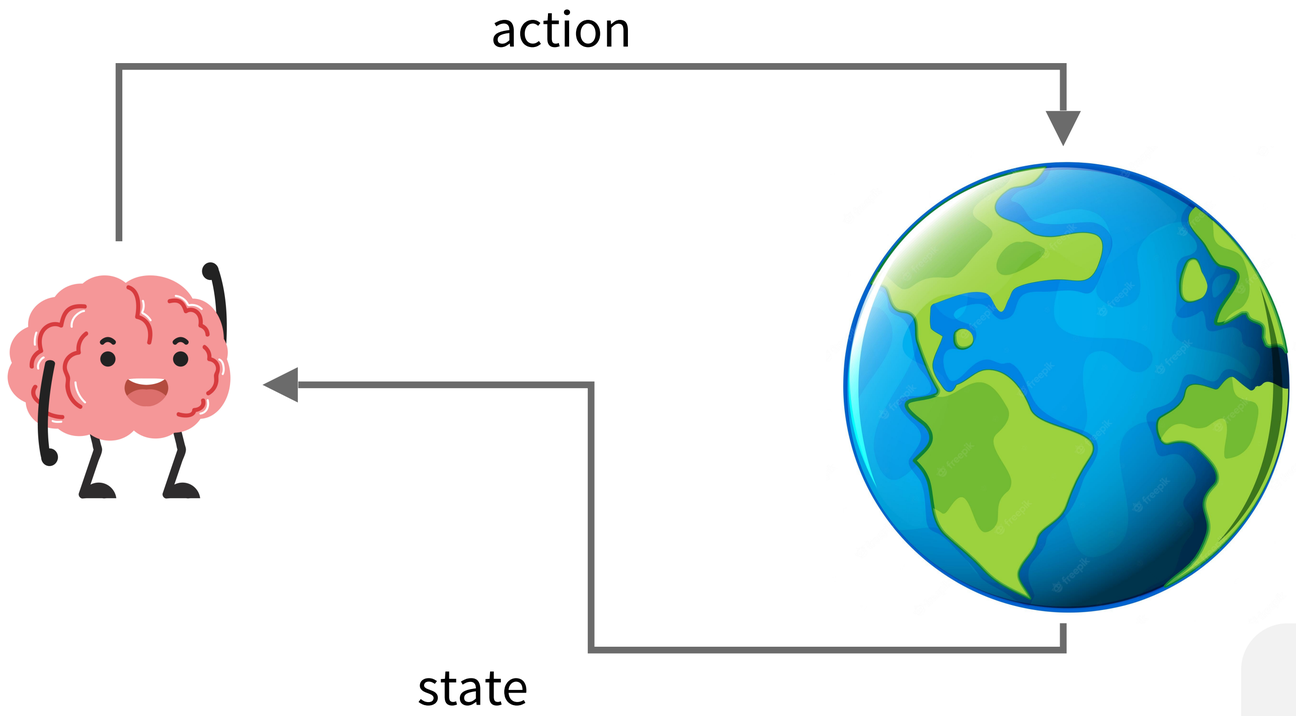
environment

agent/policy

state

e.g. Piece positions on the board

Markov Decision Process



environment

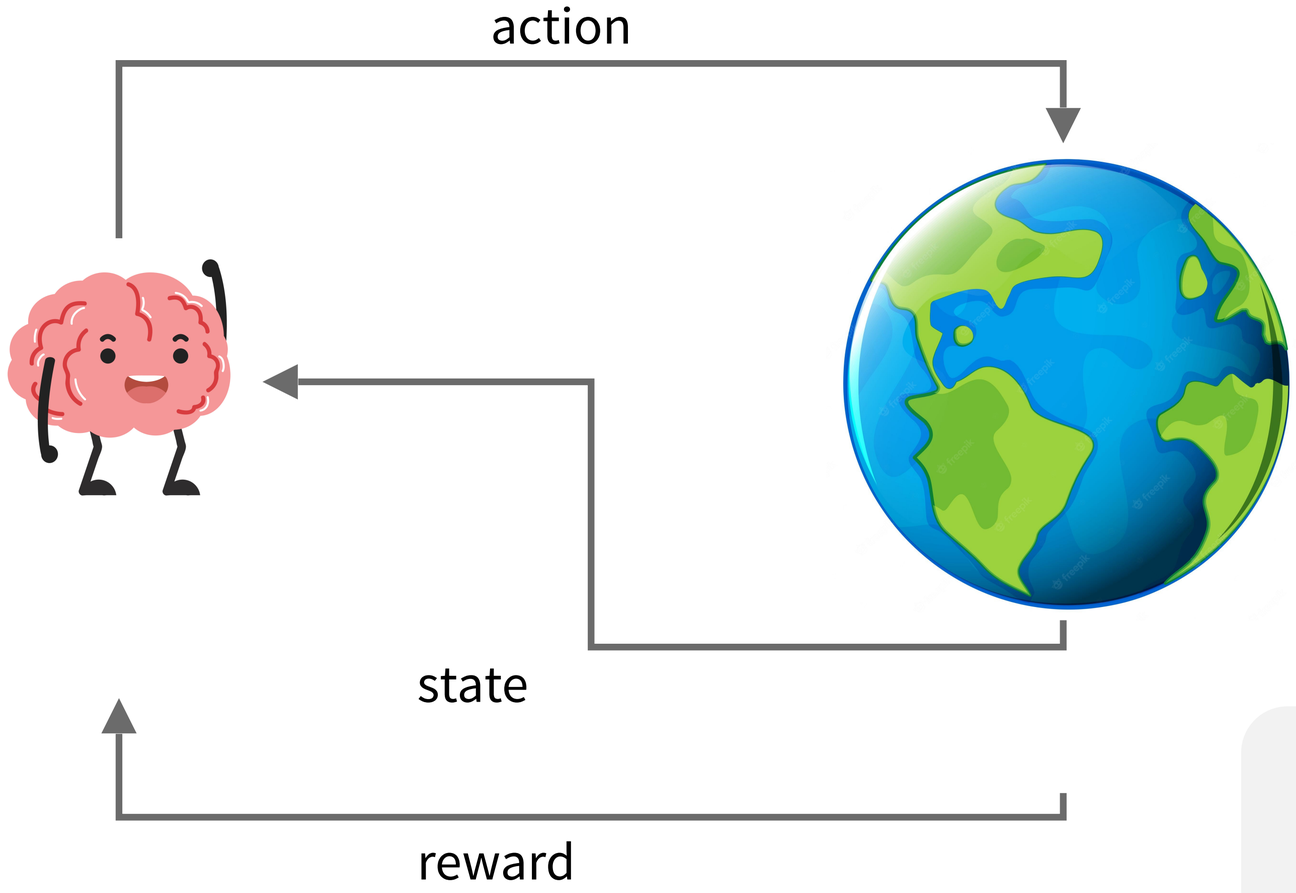
agent/policy

state

action

e.g. place the next piece

Markov Decision Process



environment

agent/policy

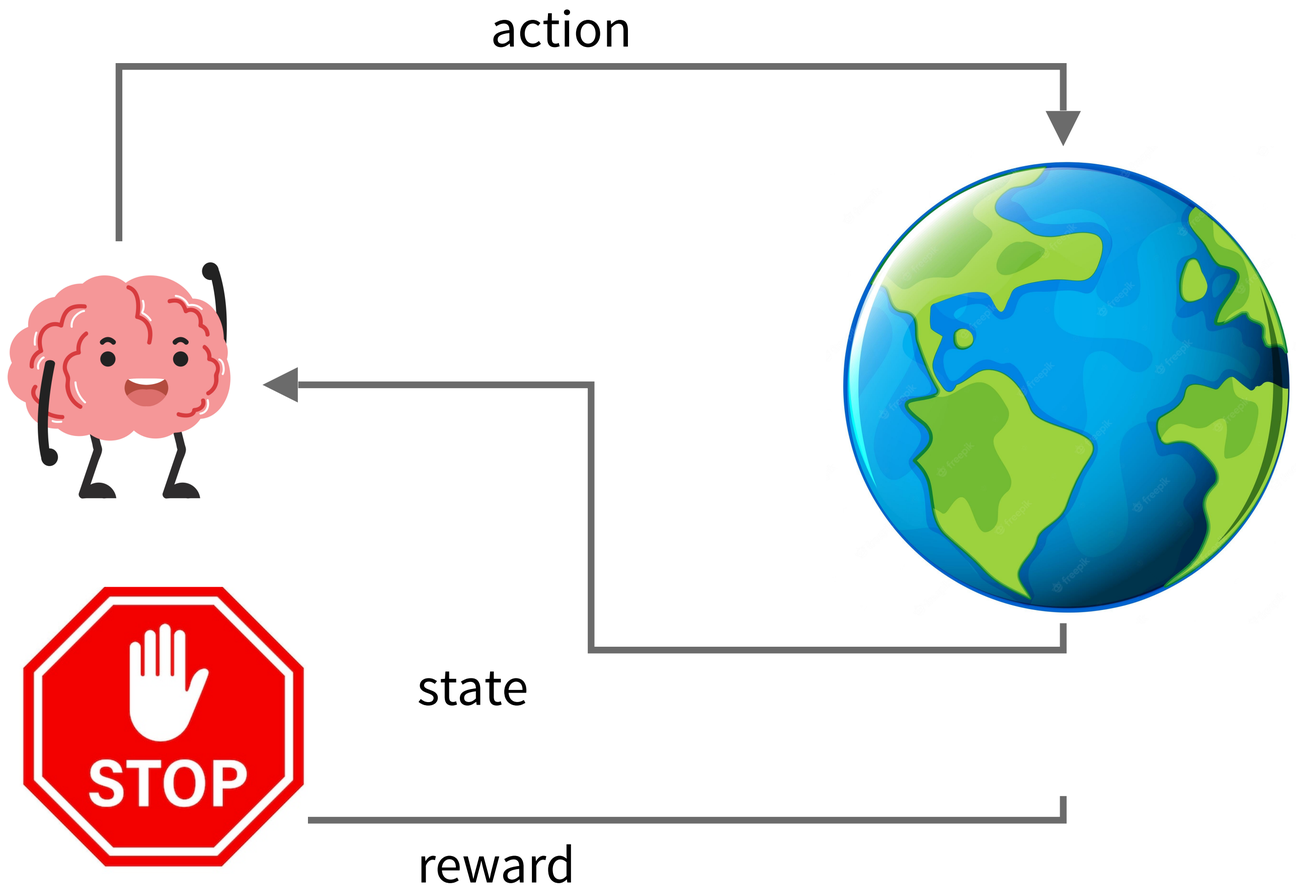
state

action

reward

e.g. +1 point if win the game

Markov Decision Process



environment

agent/policy

state

action

reward

termination

Markov Decision Process

Problem:
How can an agent take actions in an environment so as to maximize long-term reward?

Markov Decision Process (MDP)

- state $s \in S$
- action $a \in A$
- reward $\mathrm{r}(s,\ a)$
- environment $T(s,\ a) \mapsto s'$
- termination

Solution:
The agent/policy: $\pi(s) \mapsto a$

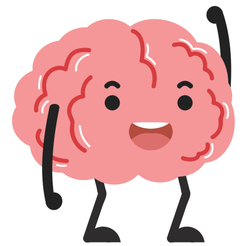
Quiz

If we want to create an AI that's good at Super Mario

- What's the environment?
- What's the agent observing?
- What actions can the agent take?
- What's the reward?
- When is the trajectory terminated?

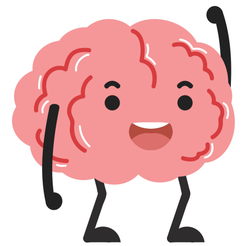
How to formulate a RL problem

- Given a MDP, RL aims to find the smartest policy/agent that gets the most reward
- But… how?



How to solve RL?

- Given a MDP, RL aims to find the smartest policy/agent that gets the most reward
- But… how?



How to solve RL? Optimization!

maximize reward

policy/agent

How to solve RL? Optimization!

maximize reward

policy/agent

Optimization argument
(The thing we want to get)

How to solve RL? Optimization!

maximize reward

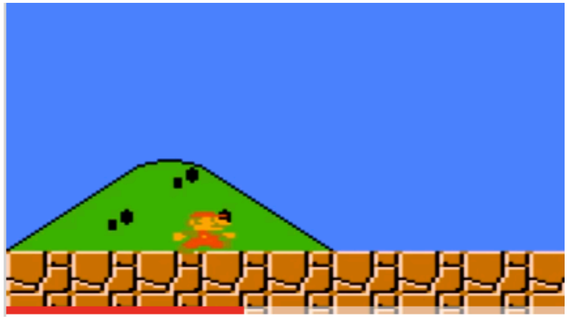
policy/agent

Objective Function
(The metric we want to optimize)

How to solve RL? Optimization!

maximize reward

policy/agent

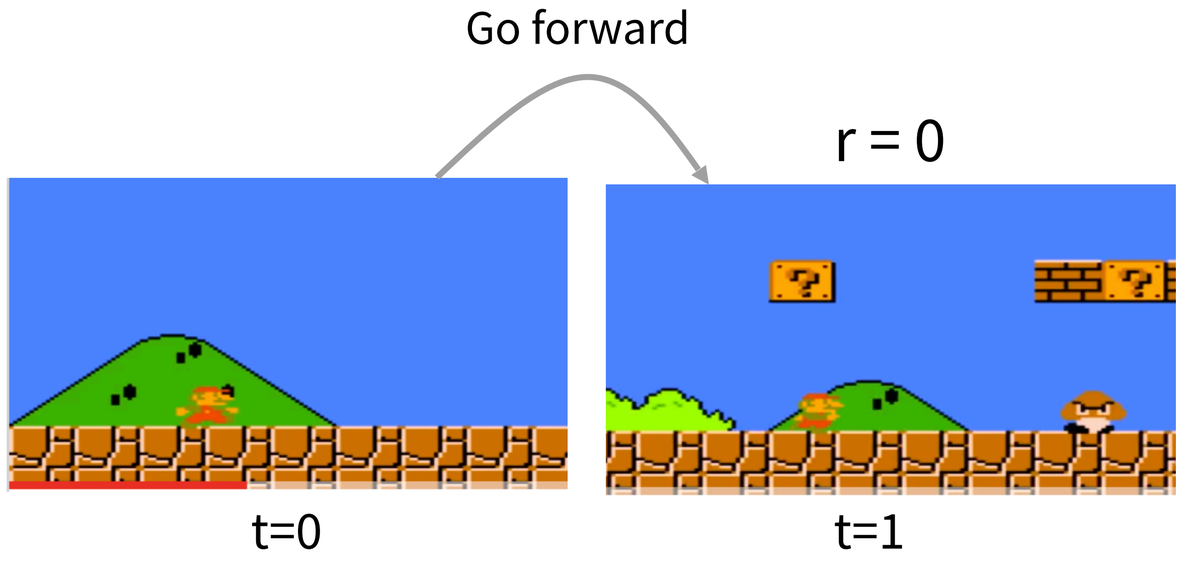


t=0

How to solve RL? Optimization!

maximize reward

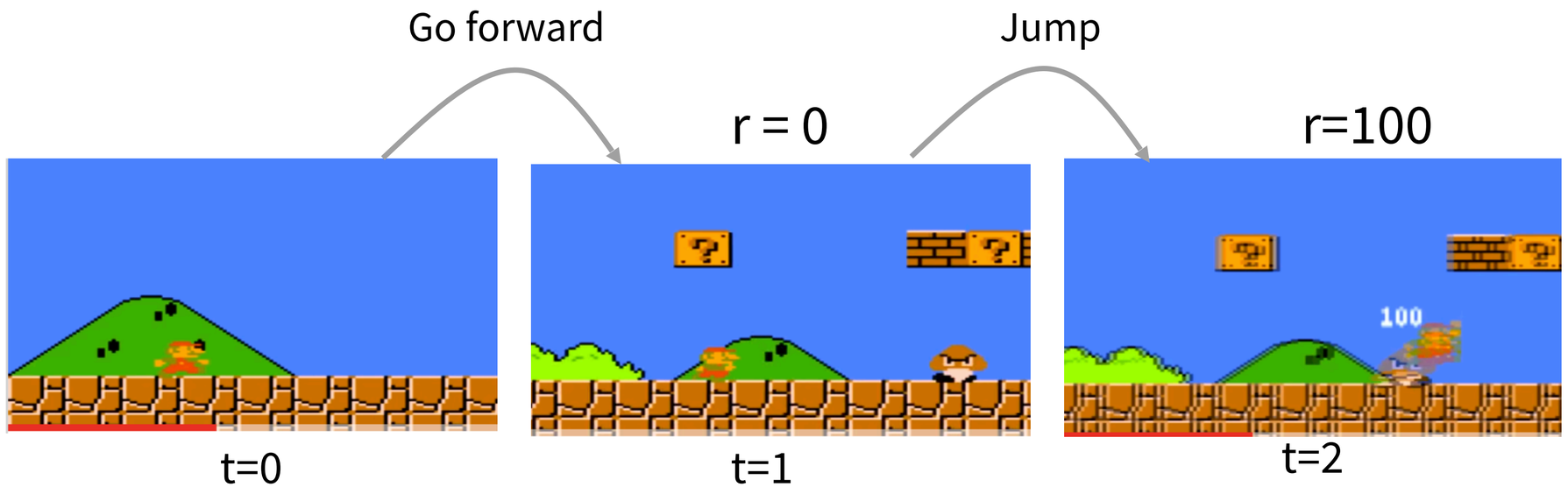
policy/agent



How to solve RL? Optimization!

maximize reward

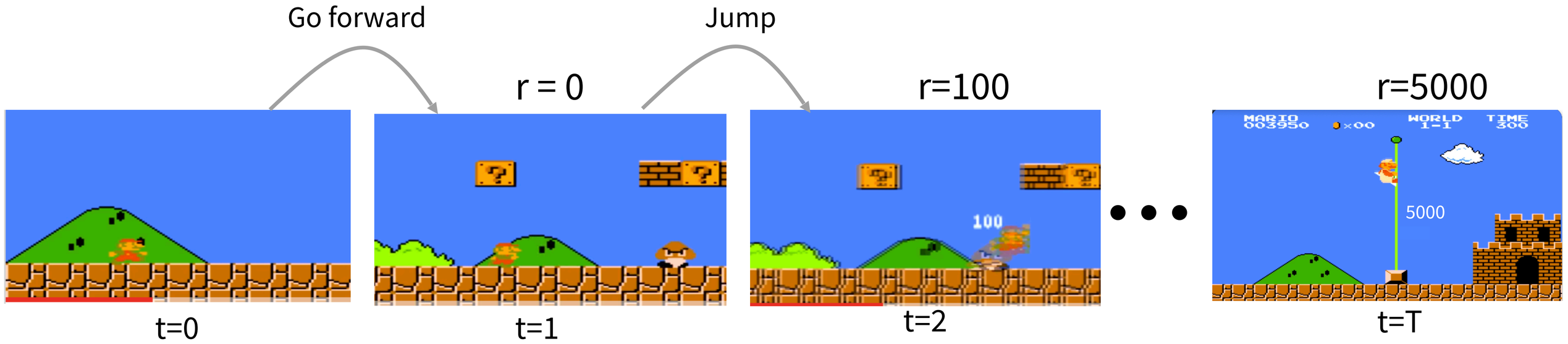
policy/agent



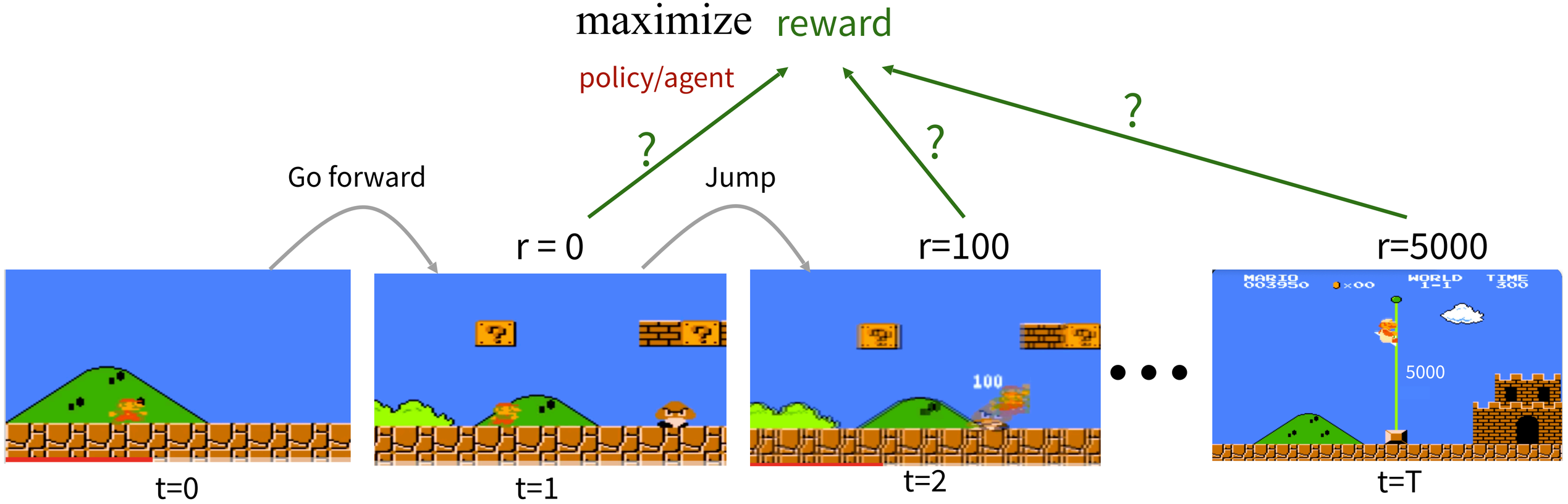
How to solve RL? Optimization!

maximize reward

policy/agent



How to solve RL? Optimization!



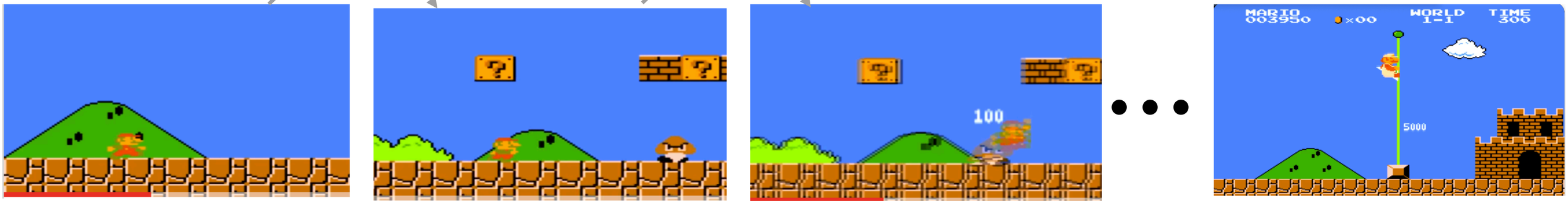
How to solve RL? Optimization!

maximize reward

policy/agent

Go forward

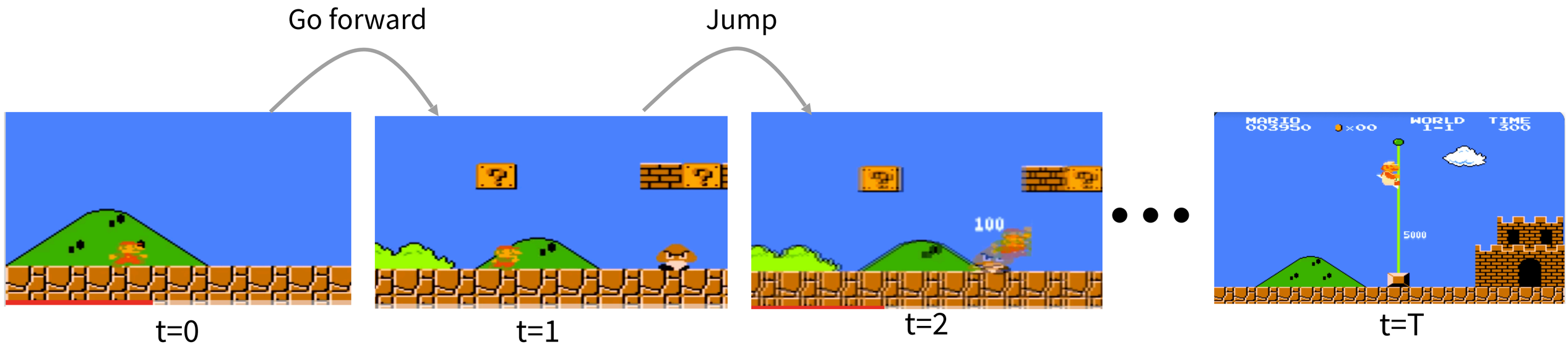
Jump



t=0 t=1 t=2 t=T

How to solve RL? Optimization!

$$\underset{\text{policy/agent}}{\text{maximize}} \sum_{t=0}^{T} r(s_t, a_t)$$



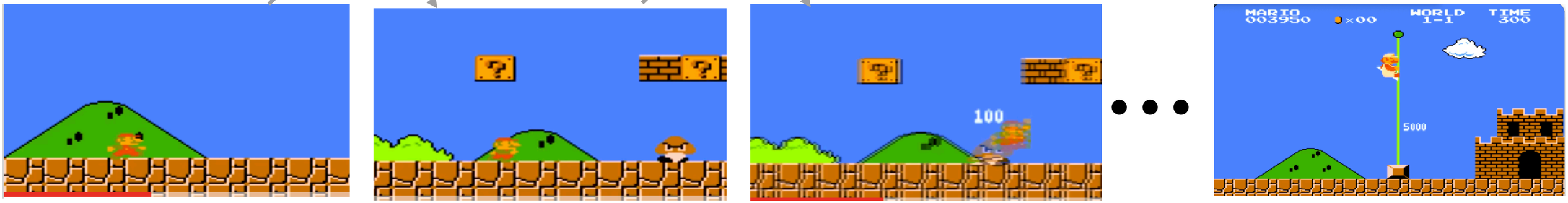
How to solve RL? Optimization!

$$\text{maximize} \sum_{t=0}^{T} r(s_t, a_t)$$

? policy/agent

Go forward

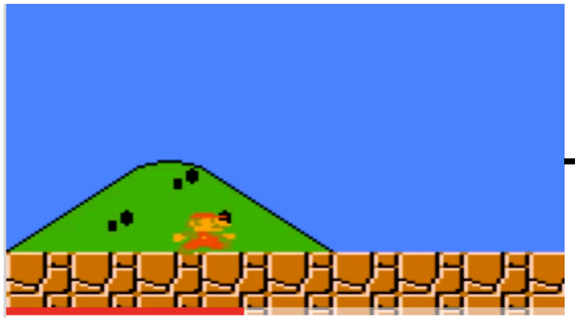
Jump



t=0 t=1 t=2 t=T

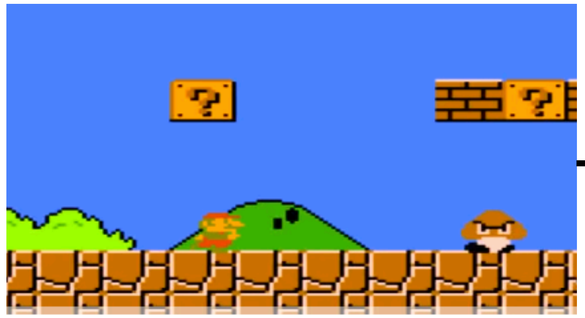
How to solve RL? Optimization!

$$\text{maximize} \sum_{t=0}^{T} r(s_t, a_t)$$



t=0

policy/agent → Go forward

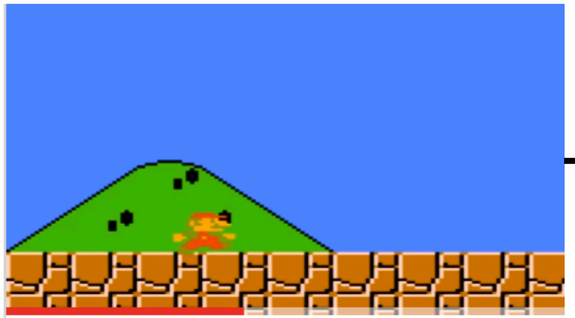


t=1

policy/agent → Jump

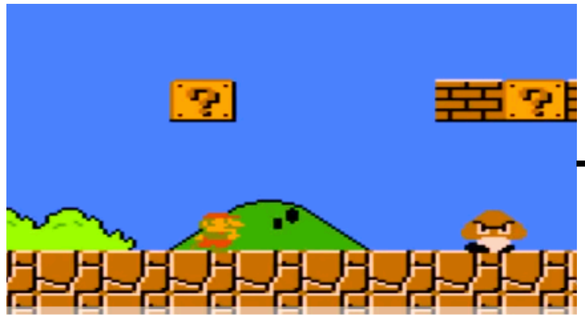
How to solve RL? Optimization!

$$\text{maximize} \sum_{t=0}^{T} r(s_t, a_t)$$



t=0

$\pi(s)$ → Go forward



t=1

$\pi(s)$ → Jump

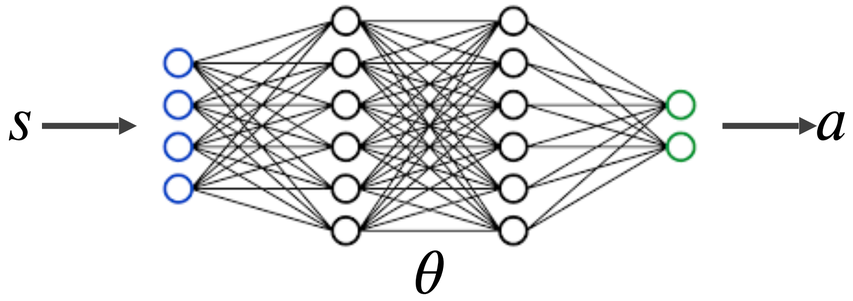
How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \quad \sum_{t=0}^{T} r(s_t,\ a_t)$$

How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Neural network parameters



How to solve RL? Optimization!

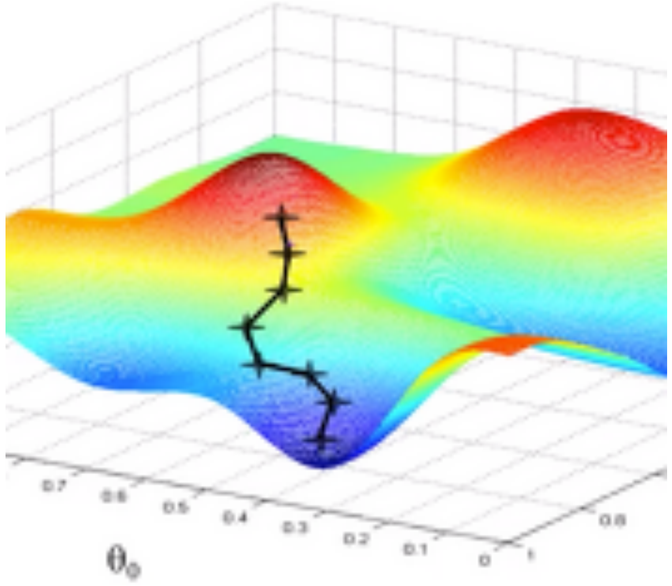
$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

How to solve this optimization problem? Hint: lab3

How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Gradient descent!



How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Step 1: Initialize θ_0

How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Step 1: Initialize θ_0

Step 2: Compute gradient: $g = \dfrac{\partial \sum_{t=0}^{T} r(s_t,\ a_t)}{\partial \theta_0}$

How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Step 1: Initialize θ_0

Step 2: Compute gradient: $g = \dfrac{\partial \sum_{t=0}^{T} r(s_t,\ a_t)}{\partial \theta_0}$

Step 3: Update agent: $\theta_1 = \theta_0 + g * \alpha$

How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Step 1: Initialize θ_0

Step 2: Compute gradient: $g = \dfrac{\partial \sum_{t=0}^{T} r(s_t,\ a_t)}{\partial \theta_0}$

Step 3: Update agent: $\theta_1 = \theta_0 + g * \alpha$

Step 4: repeat Step 2 and 3

How to solve RL? Optimization!

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Step 1: Initialize θ_0

Step 2: Compute gradient: $g = \frac{\partial \sum_{t=0}^{T} r(s_t,\ a_t)}{\partial \theta_0}$

Step 3: Update agent: $\theta_1 = \theta_0 + g * \alpha$

Step 4: repeat Step 2 and 3

Why is this gradient not easy to compute?

$$g = \frac{\partial \sum_{t=0}^{T} r(s_t,\ a_t)}{\partial \theta}$$

Why is this gradient not easy to compute?

$$g = \frac{\partial \sum_{t=0}^{T} r(s_t,\ a_t)}{\partial \theta} = \frac{\partial r(s_1, a_1)}{\partial \theta} + \frac{\partial r(s_2, a_2)}{\partial \theta} + \cdots \frac{\partial r(s_T, a_T)}{\partial \theta}$$

Why is this gradient not easy to compute?

$$g = \frac{\partial \sum_{t=0}^{T} r(s_t,\ a_t)}{\partial \theta} = \frac{\partial r(s_1, a_1)}{\partial \theta} + \frac{\partial r(s_2, a_2)}{\partial \theta} + \cdots \frac{\partial r(s_T, a_T)}{\partial \theta}$$

Why is this gradient not easy to compute?

$$\frac{\partial r(s_2, a_2)}{\partial \theta} = \frac{\partial r}{\partial s_2}\frac{\partial s_2}{\partial \theta} + \frac{\partial r}{\partial a_2}\frac{\partial a_2}{\partial \theta}$$

Why is this gradient not easy to compute?

$$\frac{\partial r(s_2, a_2)}{\partial \theta} = \boxed{\frac{\partial r}{\partial s_2}} \frac{\partial s_2}{\partial \theta} + \boxed{\frac{\partial r}{\partial a_2}} \frac{\partial a_2}{\partial \theta}$$

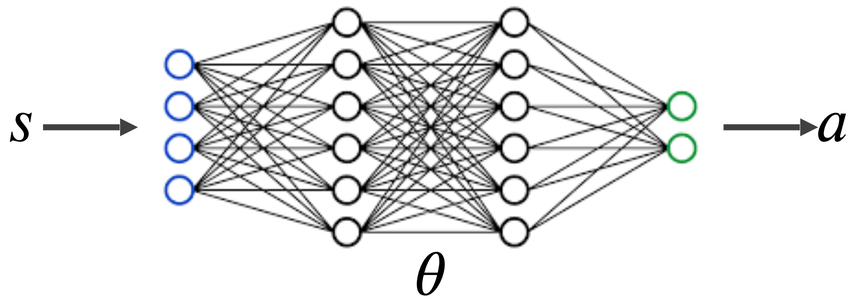
1. Requires reward to be differentiable

Why is this gradient not easy to compute?

$$\frac{\partial r(s_2, a_2)}{\partial \theta} = \frac{\partial r}{\partial s_2}\frac{\partial s_2}{\partial \theta} + \frac{\partial r}{\partial a_2}\frac{\partial a_2}{\partial \theta}$$

$$\frac{\partial s_2}{\partial a_1}\frac{\partial a_1}{\partial \theta} + \dots$$

1. Requires reward to be differentiable
2. Requires dynamics to be differentiable

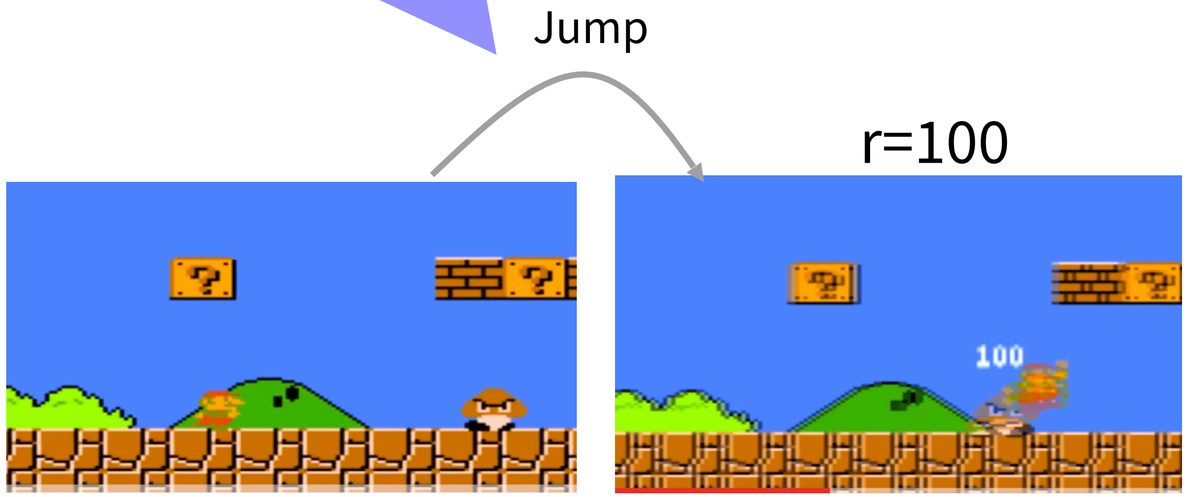


Why is this gradient not easy to compute?

$$\frac{\partial r(s_2, a_2)}{\partial \theta} = \frac{\partial r}{\partial s_2}\frac{\partial s_2}{\partial \theta} + \frac{\partial r}{\partial a_2}\frac{\partial a_2}{\partial \theta}$$

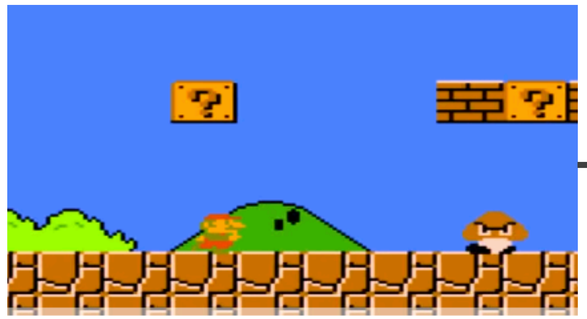
How does a pixel change in image when we change action slightly?

$$\frac{\partial s_2}{\partial a_1}\frac{\partial a_1}{\partial \theta} + \cdots$$



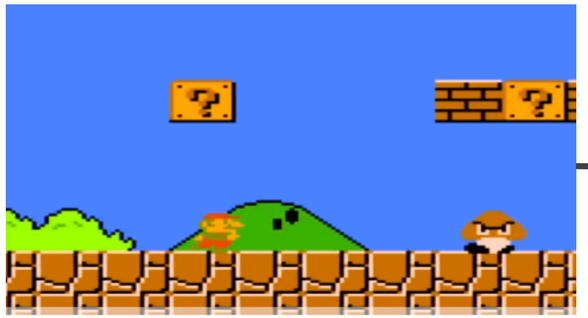
Stochastic Policy come to rescue

Deterministic policy: $a = \pi_\theta(s)$



$\pi_\theta(s)$ → jump

Stochastic policy: $a \sim \pi_\theta(a \mid s)$



$\pi_\theta(s)$ →

forward jump

Stochastic Policy come to rescue

Deterministic policy: $a = \pi_\theta(s)$

Deterministic environment: $T(s,\ a) \mapsto s'$

Stochastic policy: $a \sim \pi_\theta\left(a \mid s\right)$

Stochastic environment: $p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$

Stochastic Policy come to rescue

Deterministic formulation:

$$\underset{\pi_\theta(s)}{\text{maximize}} \sum_{t=0}^{T} r(s_t,\ a_t)$$

Stochastic formulation:

Stochastic Policy come to rescue

Deterministic formulation:

$$\underset{\pi_\theta(s)}{\text{maximize}} \quad \sum_{t=0}^{T} r(s_t,\ a_t)$$

Stochastic formulation:

$$\underset{\pi_\theta(s)}{\text{maximize}} \quad E_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} r\left(s_t,\ a_t\right)\right]$$

Sampled trajectories/rollouts

Quiz

- What is the probability of a rollout under a given dynamic transition and a given policy, assuming the initial state probability is $p(\mathbf{s}_1)$?

  $\pi(\mathbf{a}|\mathbf{s})$: policy

  $p(\mathbf{s}'|\mathbf{s},\mathbf{a})$: dynamic transition

  $p_\theta(\boldsymbol{\tau}) = ?$

Policy gradient derivation

$$J(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} r(s_t,\ a_t)\right] = E_\tau\Big[R(\tau)\Big] = \int_\tau p_\theta(\tau) R(\tau) d\tau \ \text{, where } R(\tau) = \sum_t^T r(\mathbf{s}_t, \mathbf{a}_t)$$

$$\nabla_\theta J = \int_\tau \nabla_\theta p_\theta(\tau) R(\tau) d\tau$$

$$\nabla_\theta J = \int_\tau p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) R(\tau) d\tau = E_\tau\Big[\nabla_\theta \log p_\theta(\tau) R(\tau)\Big]$$

$$\nabla_\theta \log p_\theta(\tau) = \nabla_\theta \log \left(p_1(\mathbf{s}_1) \prod_t^{T-1} \pi(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\right)$$

$$= \nabla_\theta \Big(\cancel{\log p_1(\mathbf{s}_1)} + \sum_t^{T-1} \log \pi(\mathbf{a}_t|\mathbf{s}_t) + \sum_t^{T-1} \cancel{\log p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} \Big)$$

$$= \sum_t^{T-1} \nabla_\theta \log \pi(\mathbf{a}_t|\mathbf{s}_t)$$

Likelihood-ratio identity:

$$\nabla_x \log f(x) = \frac{\nabla_x f(x)}{f(x)}$$

$$\nabla_x f(x) = f(x) \nabla_x \log f(x)$$

replace f with p and x with θ

$$\nabla_\theta p_\theta = p_\theta \nabla_\theta \log p_\theta$$

Policy gradient derivation

$$\nabla_\theta J = E_\tau \Big[\nabla_\theta \log p_\theta(\tau) R(\tau) \Big]$$

$$= E_\tau \Big[\sum_t^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t | \mathbf{s}_t) \sum_t^{T} r(\mathbf{s}_t, \mathbf{a}_t) \Big]$$

$$\approx \frac{1}{N} \sum_i^N \big(\sum_t^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i | \mathbf{s}_t^i) \sum_t^{T} r(\mathbf{s}_t^i, \mathbf{a}_t^i) \big)$$

- Sample N rollouts by executing the current policy.
- Evaluate rewards of each rollout and sum them up: $\sum_t^T r(\mathbf{s}_t^i, \mathbf{a}_t^i)$
- Evaluate $\nabla_\theta \log \pi_\theta$ at each step in the rollout and sum them up: $\sum_t^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i | \mathbf{s}_t^i)$

Policy Gradient

$$g(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t \mid s_t)}{\partial \theta} \sum_{t=0}^{T} r\left(s_t,\ a_t\right)\right]$$

Policy Gradient

$$g(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t \mid s_t)}{\partial \theta} \sum_{t=0}^{T} r\left(s_t,\ a_t\right)\right]$$

How to make a trajectory more likely

Policy Gradient

$$g(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t \mid s_t)}{\partial \theta} \sum_{t=0}^{T} r\left(s_t,\ a_t\right)\right]$$

How to make a trajectory more likely

x

Total reward in a trajectory

Policy Gradient

$$g(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t \mid s_t)}{\partial \theta} \sum_{t=0}^{T} r(s_t, a_t)\right]$$

Average over many trajectories

(How to make a trajectory more likely x Total reward in a trajectory)

Policy Gradient Algorithm

1. Collect many trajectories using π_θ

2. Compute gradient:

$$g(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t \mid s_t)}{\partial \theta} \sum_{t=0}^{T} r(s_t,\ a_t)\right]$$

Average over many trajectories

(How to make a trajectory more likely X Total reward in a trajectory)

3. Update $\theta' = \theta + g(\theta)\ \delta$ (make good trajectories more likely!)

Problem with simple Policy Gradient

1. Require a lot of samples to get good gradient.
2. Old sample becomes useless once policy changes.

Problem with simple Policy Gradient

1. Require a lot of samples to get good gradient.
2. Old sample becomes useless once policy changes.

What if …

1. we can use fewer samples to get good gradient?

2. we can re-use old samples form previous policies?

Problem with simple Policy Gradient

1. Require a lot of samples to get good gradient.
2. Old sample becomes useless once policy changes.

What if …

1. we can use fewer samples to get good gradient?

   Generalized Advantage Estimator (GAE)!

2. we can re-use old samples form previous policies?

Problem with simple Policy Gradient

1. Require a lot of samples to get good gradient.
2. Old sample becomes useless once policy changes.

What if …

1. we can use fewer samples to get good gradient?

Generalized Advantage Estimator (GAE)!

2. we can re-use old samples form previous policies?

Trust-Region Policy Optimization (TRPO)!
Proximal Policy Optimization (PPO)!

What have we learned today?

- What is RL
    - Use Markov Decision Process (MDP) to formulate RL problems
- Solve the formulated RL problem using optimization
    - Why is it difficult to solve it using deterministic policy
    - Policy Gradient algorithm using stochastic policy

A side note

- There are many good ways to interpret and understand RL.
- Best way to understand it is to learn it from different perspectives.
- We learn one of them today, that hopefully augments other materials.
- Other useful resources:
  - https://spinningup.openai.com/en/latest/
  - https://cs224r.stanford.edu/slides/cs224r-actor-critic-split.pdf
  - …

Lab 5 preliminary

- Sim2real and Accelerator-based sim

Sim2real and Accelerator-based sim

Zhuang et al, 2023

Cheng et al, 2023

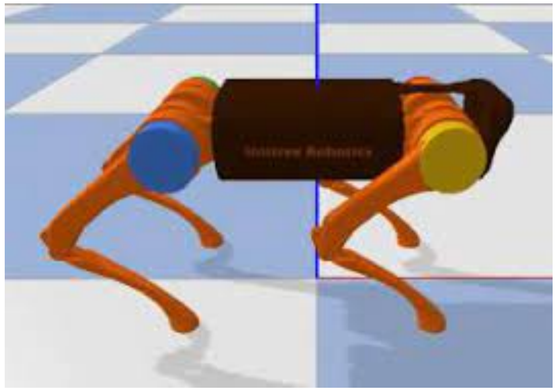
Sim2real and Accelerator-based sim

Zhuang et al, 2023

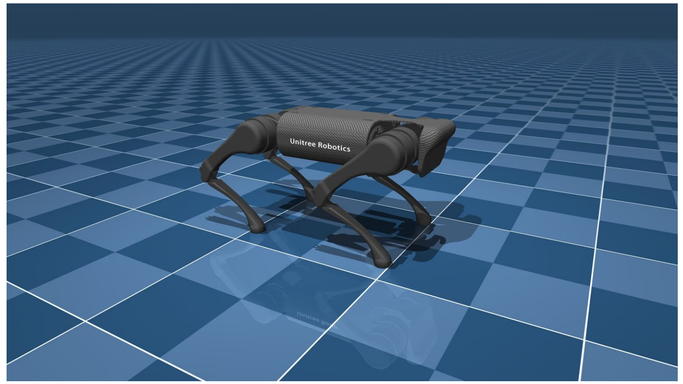
Cheng et al, 2023

~3 Billion control steps in training!
750 days!

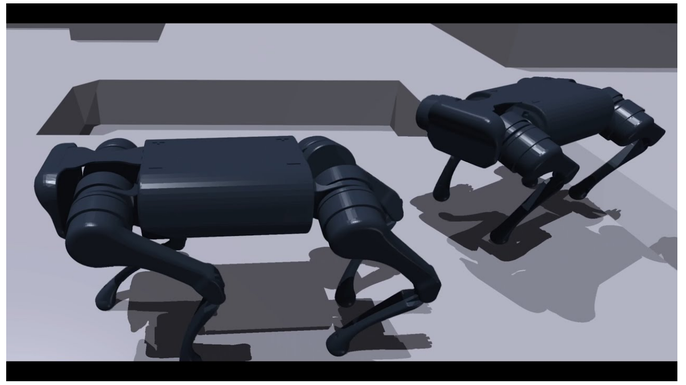
Sim2real and Accelerator-based sim



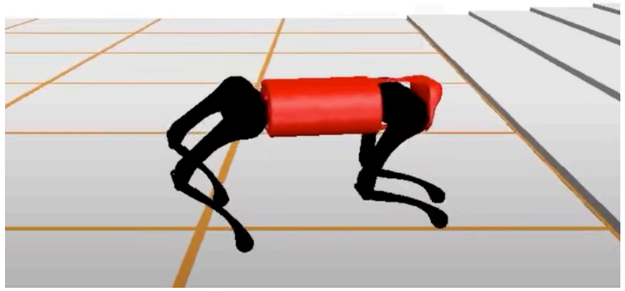
PyBullet



MuJoCo/MJX



IsaacGym

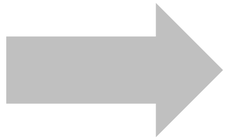


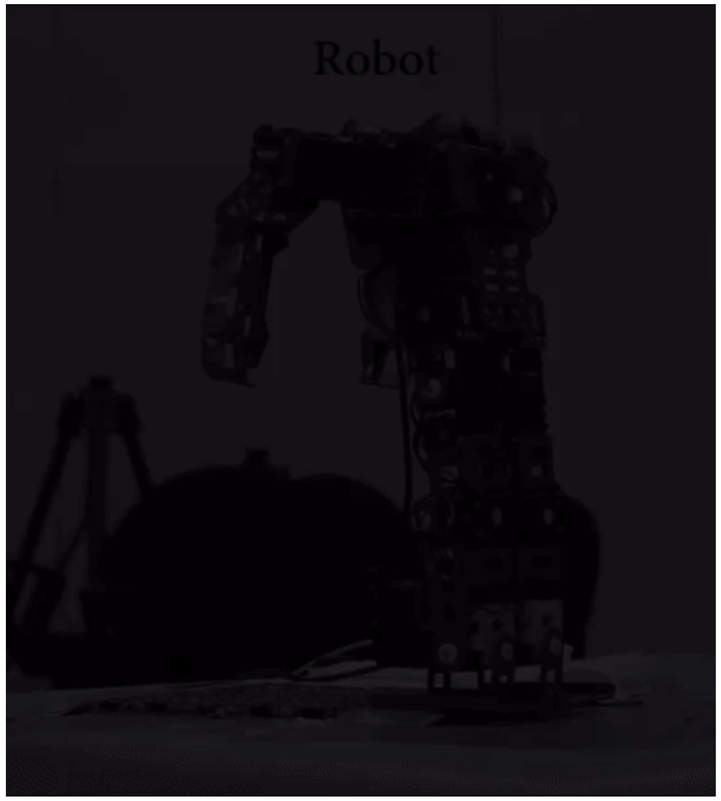
Dart

RaiSim

Sim2Real

Sim2Real





Sim2Real

- Sim2Real gap:

Sim2Real

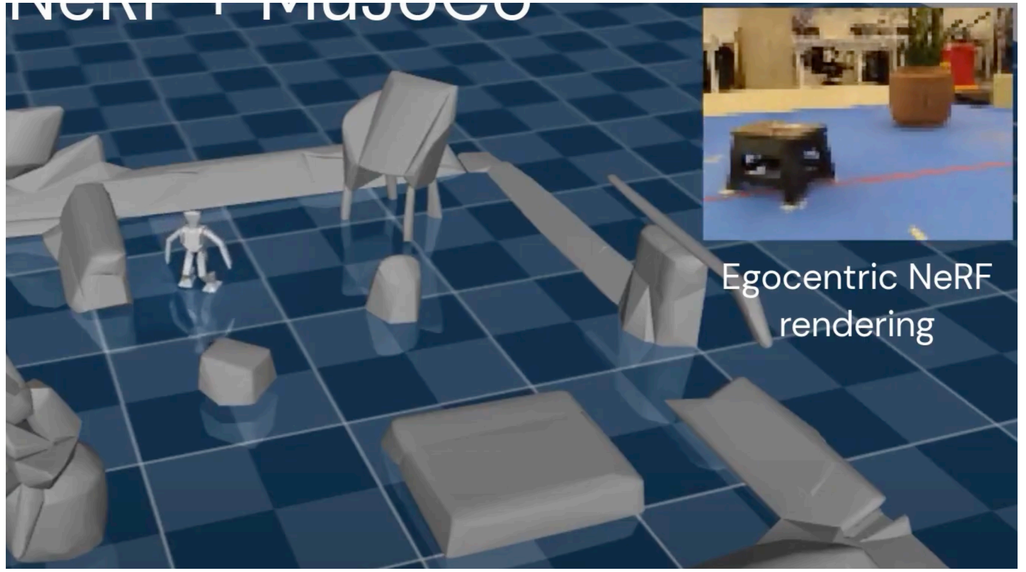
- Sim2Real gap:
    - Image looks different
    - Dynamics is different

Sim2Real

- Sim2Real gap:
    - Can we make simulation the same as real-world?

Sim2Real

- Sim2Real gap:
  - Can we make simulation the same as real-world?
    - Yes, to some extent.



https://sites.google.com/corp/view/nerf2real/home

Sim2Real – Domain Randomization

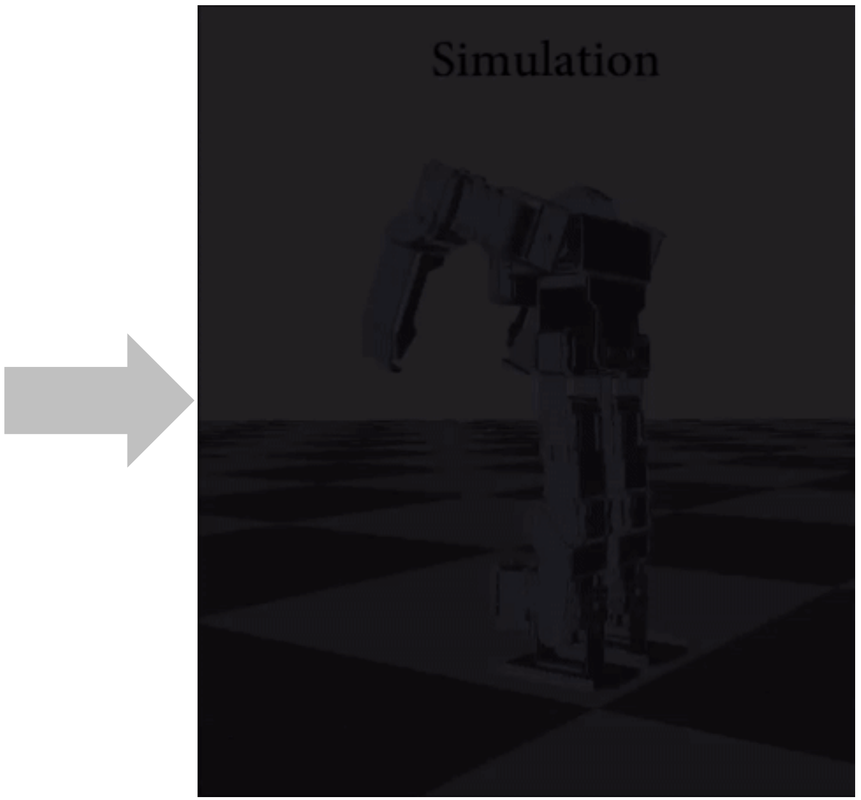
Train with different:

- Mass
- Friction
- Motor strengths
- Latency
- …

Sim2Real – Domain Randomization

Train with different:

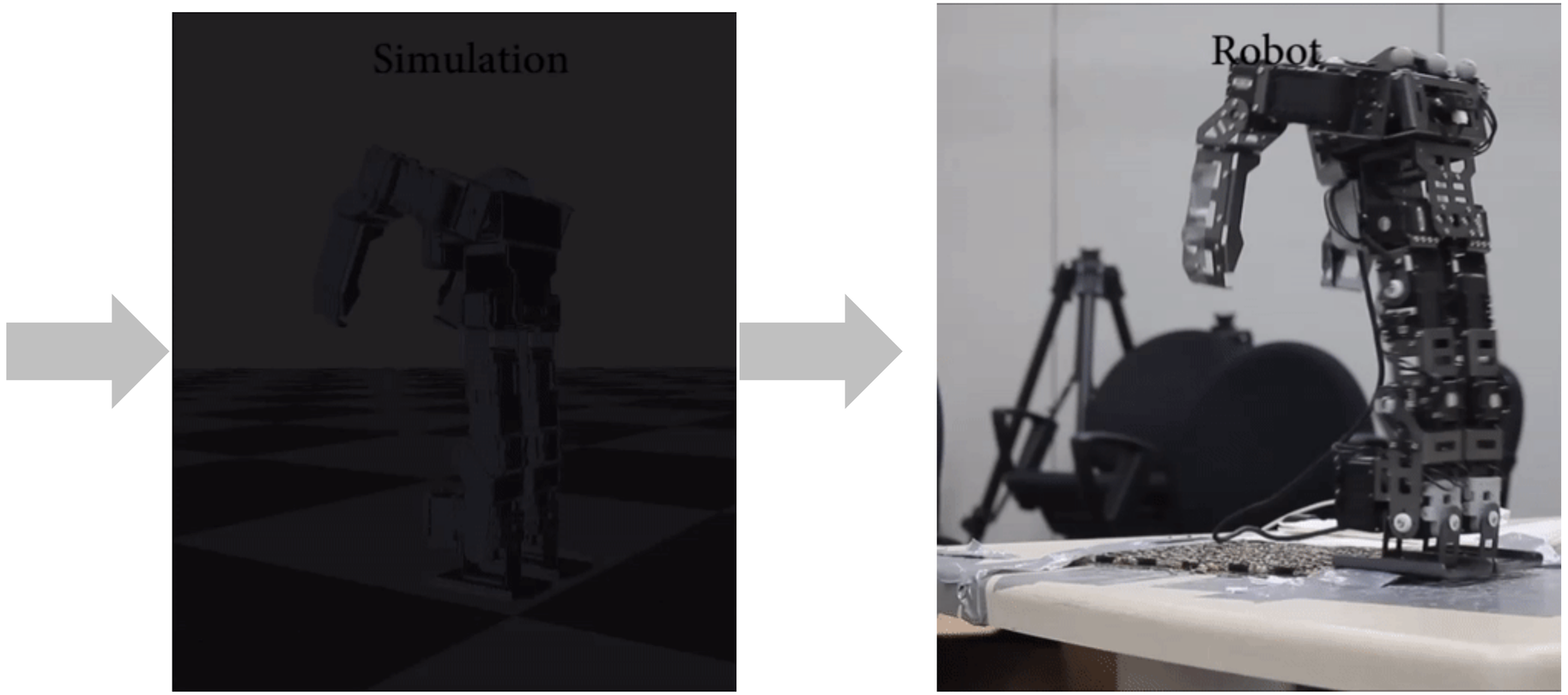
- Mass
- Friction
- Motor strengths
- Latency
- …



Sim2Real – Domain Randomization

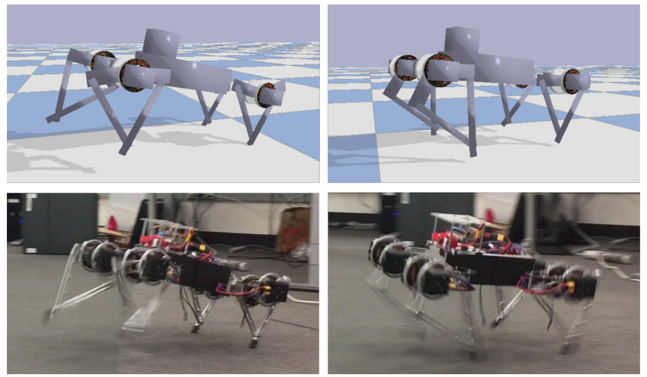
Train with different:

- Mass
- Friction
- Motor strengths
- Latency
- …



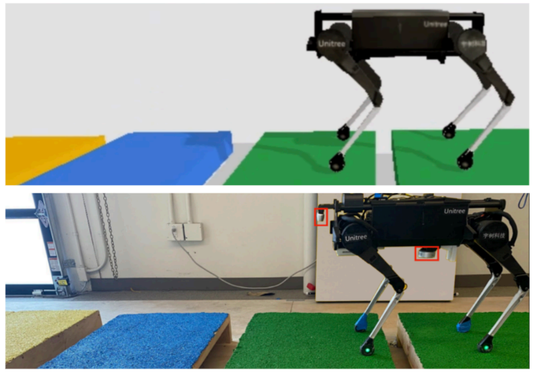
Sim2Real for Locomotion

0.5 sim day



Sim-to-real: Learning agile locomotion for quadruped robots

120 sim day



Visual-Locomotion: Learning to Walk on Complex Terrains with Vision

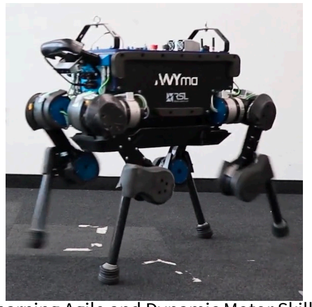
2018

2019

2021

2023

9 sim day

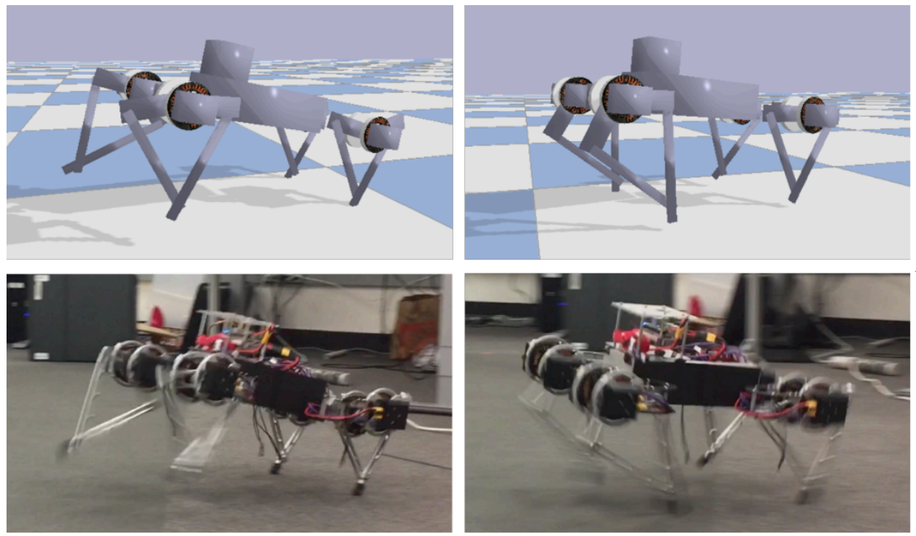


Learning Agile and Dynamic Motor Skills for Legged Robots

750 sim day

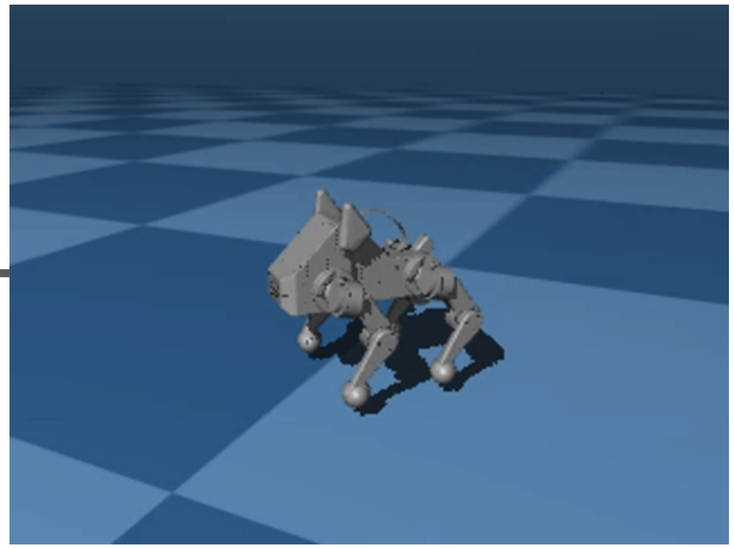
Robot Parkour Learning

Accelerator-based Parallel-simulation



CPU-based

500x faster



GPU-based