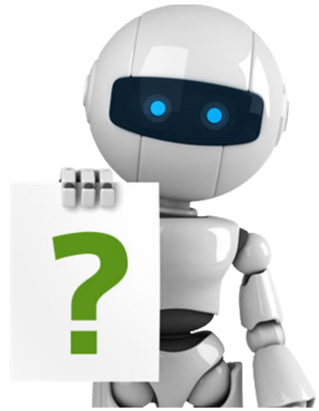
Introduction of Deep Reinforcement Learning (RL

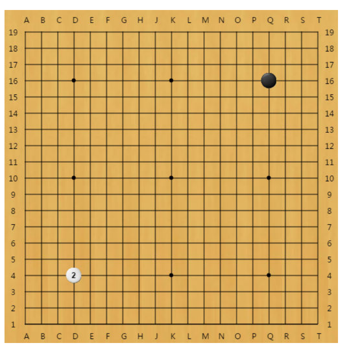
Hung-yi Lee

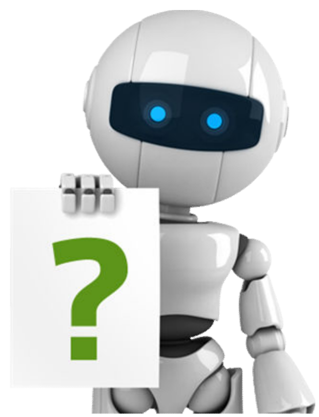
Supervised Learning → RL



? ◄···► Human labe

Cat





? ◄···► Human labe

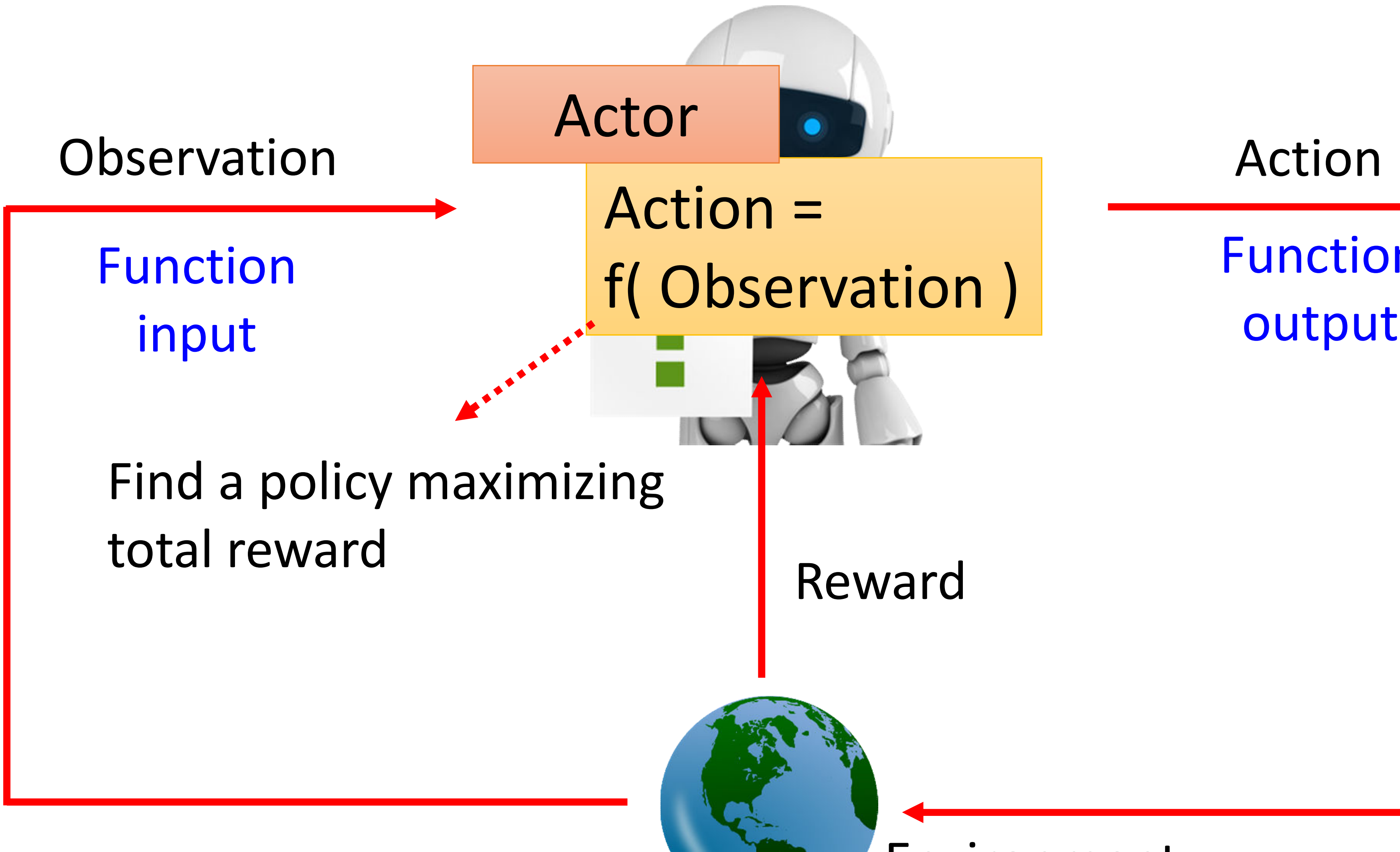
“3-3”?

It is challenging to label data in some tasks.

Outline

- What is RL? (Three steps in ML)
- Policy Gradient
- Actor-Critic
- Reward Shaping
- No Reward: Learning from Demonstration

Machine Learning
≈ Looking for a Function



Example: Playing Video Game

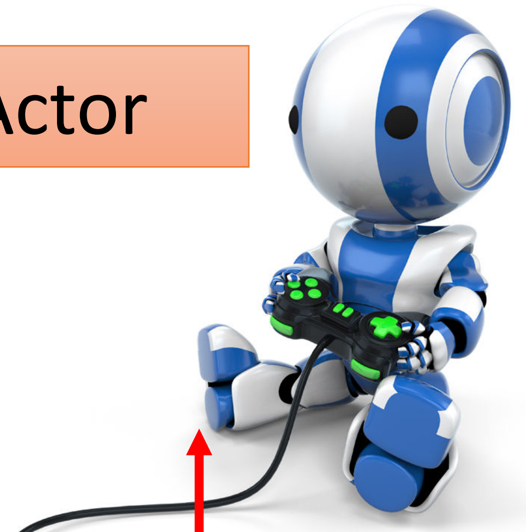
- Space invader

Termination: all the aliens are ki
or your spaceship is destroyed.

Example: Playing Video Game

Actor

Observation



Action

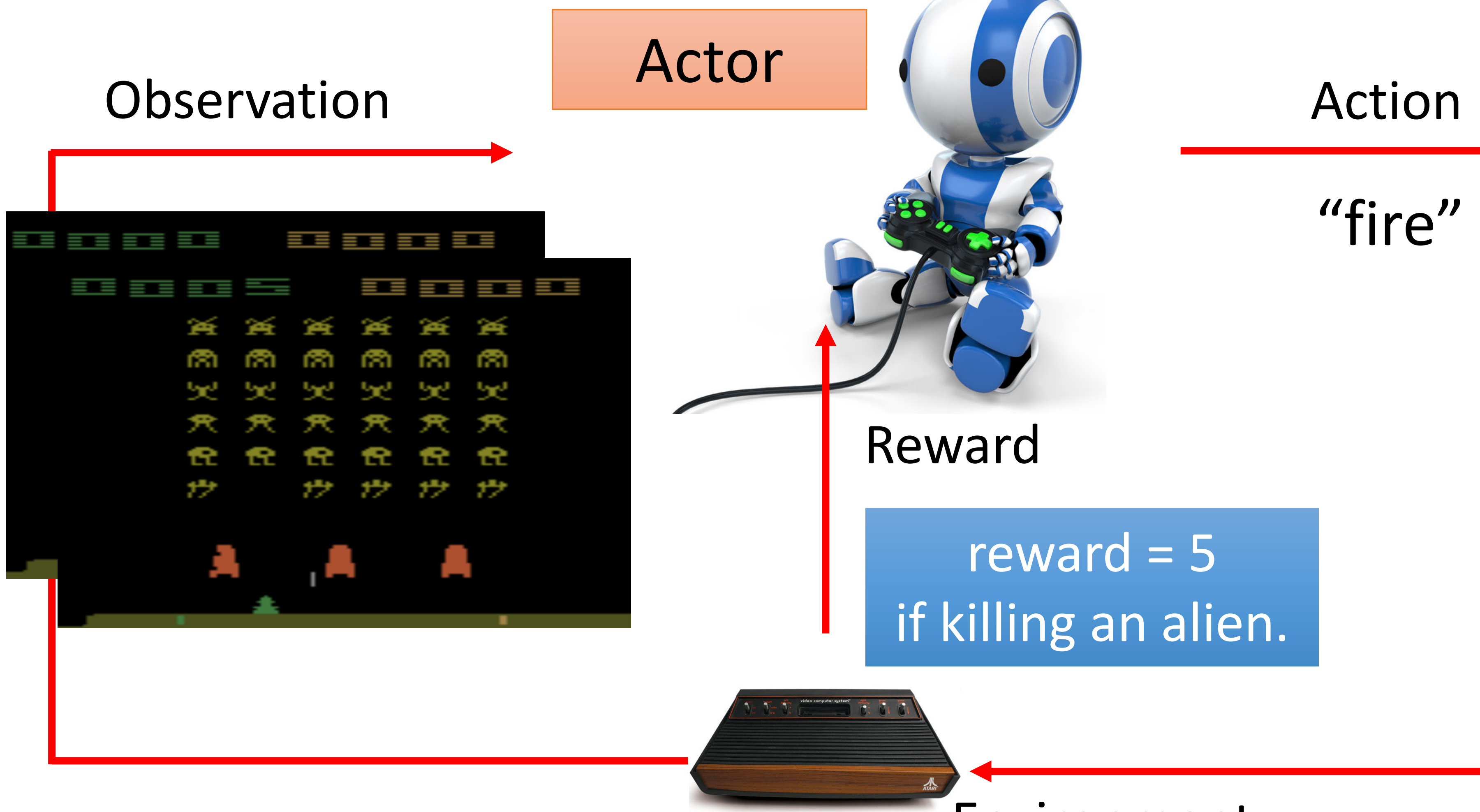
“right”

Reward

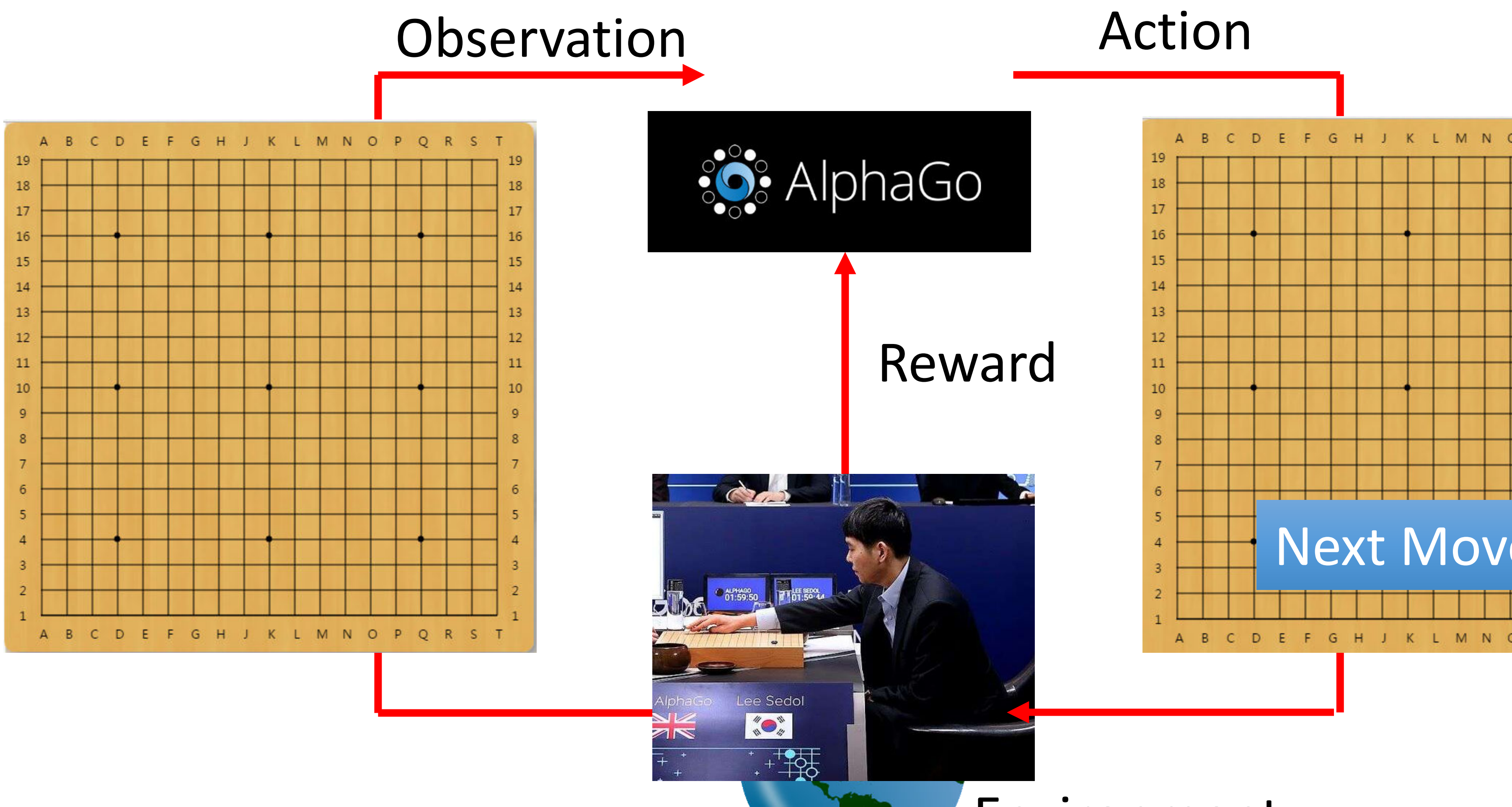
reward = 0

Example: Playing Video Game

Find an actor maximizing expected reward.

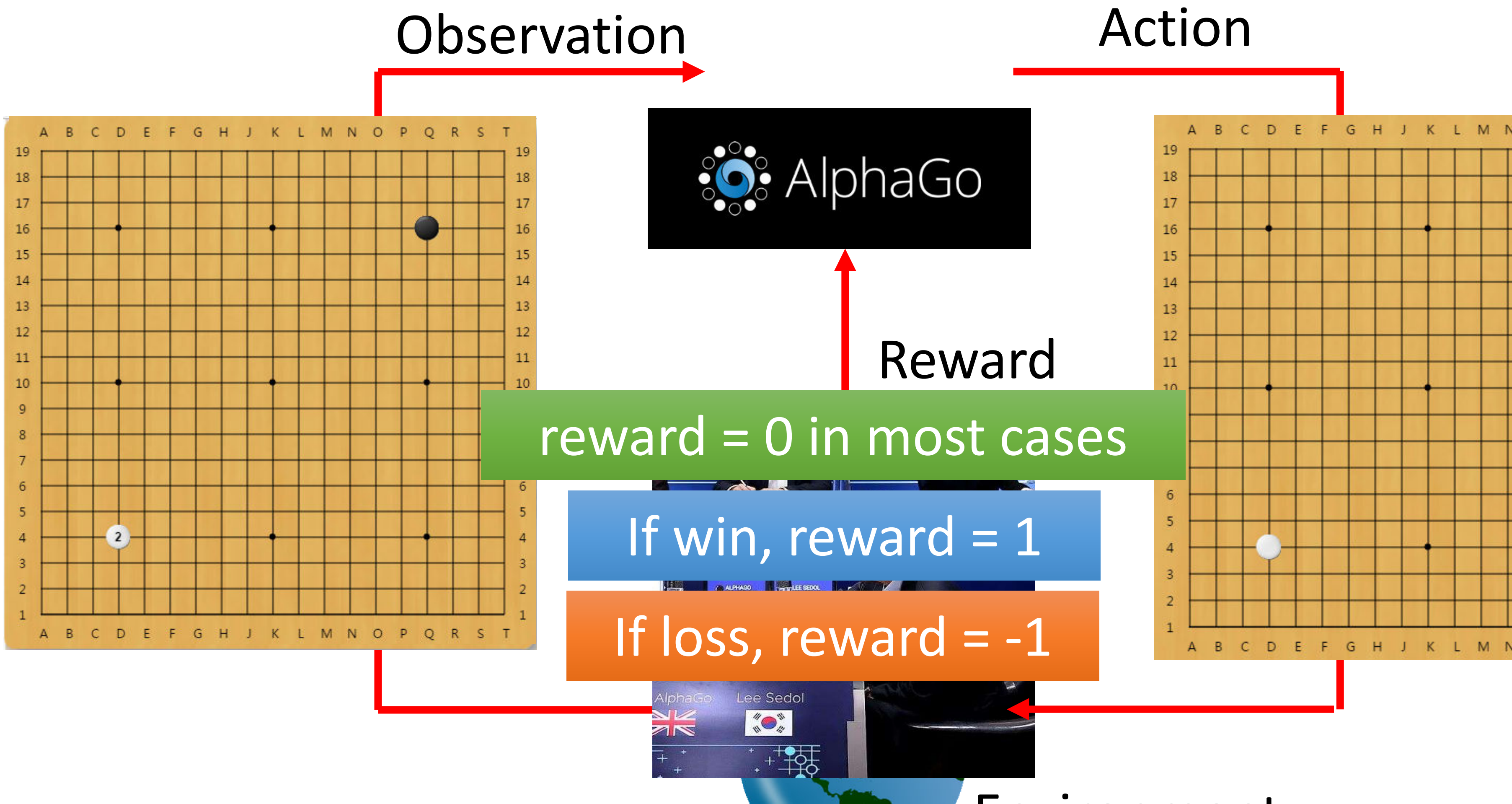


Example: Learning to play Go



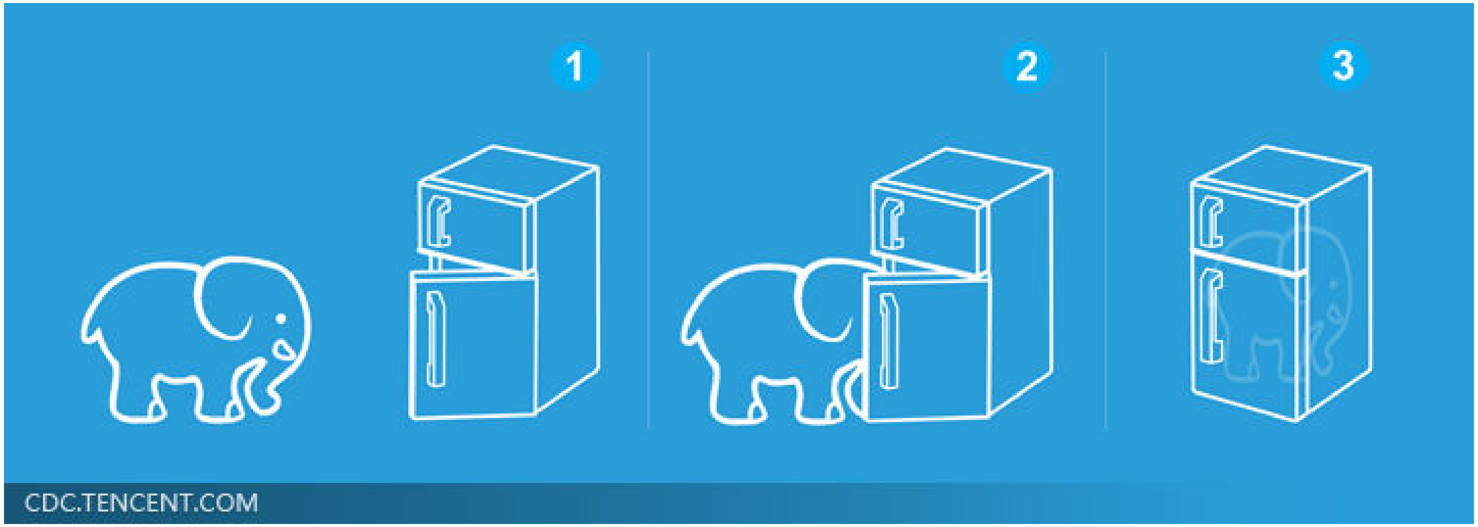
Example: Learning to play Go

Find an actor maximizing expected reward.

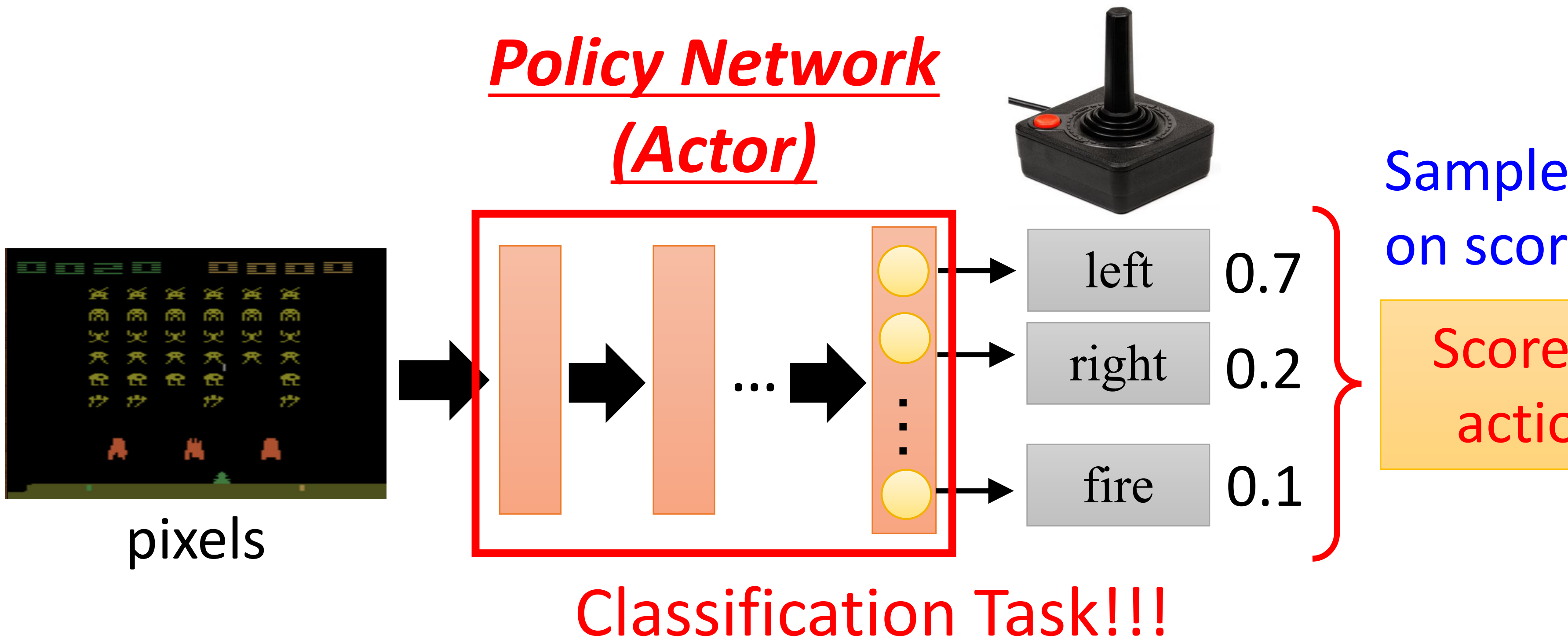


Machine Learning is so simple ……

Step 1: function with unknown → Step 2: define loss from training data → Step 3: optimizatio

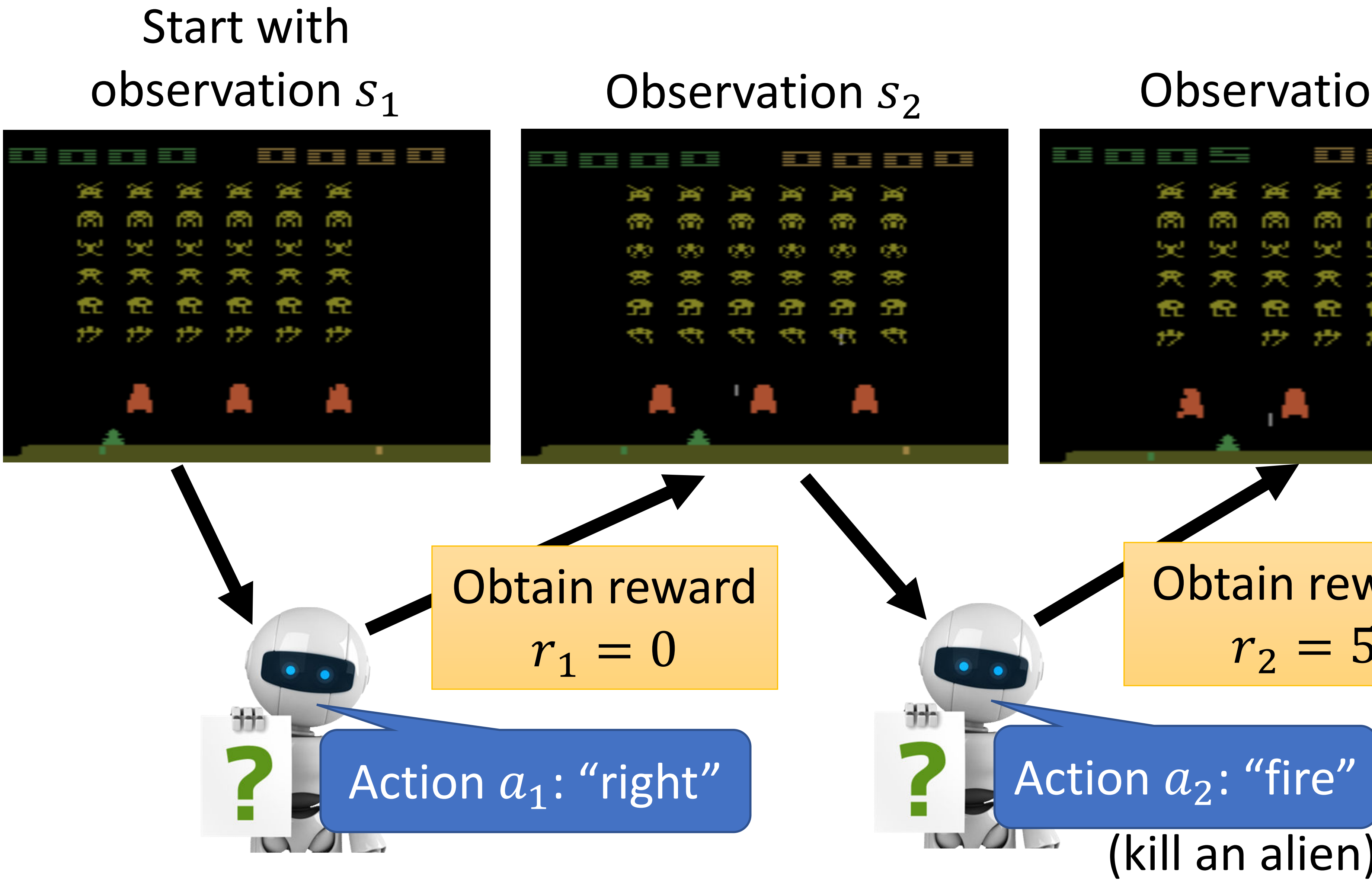


Step 1: Function with Unknown



- Input of neural network: the observation of machine represented as a vector or a matrix
- Output neural network : each action corresponds to a

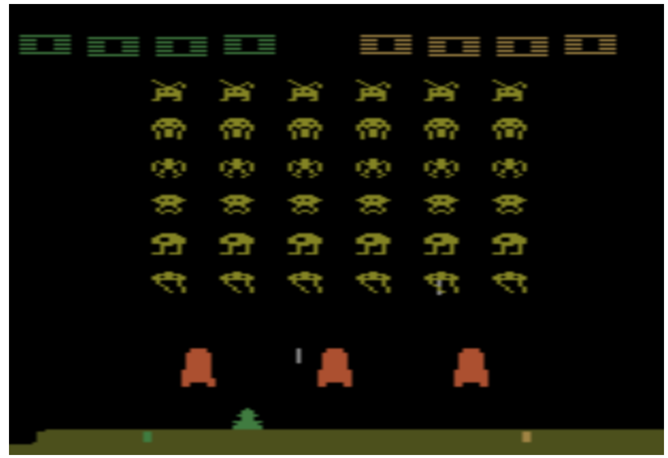
Step 2: Define "Loss"



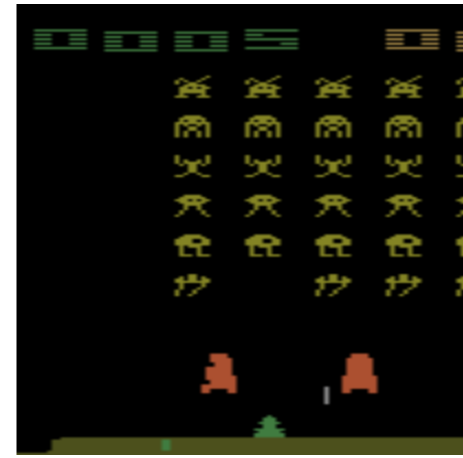
Step 2: Define "Loss"

Start with observation s_1

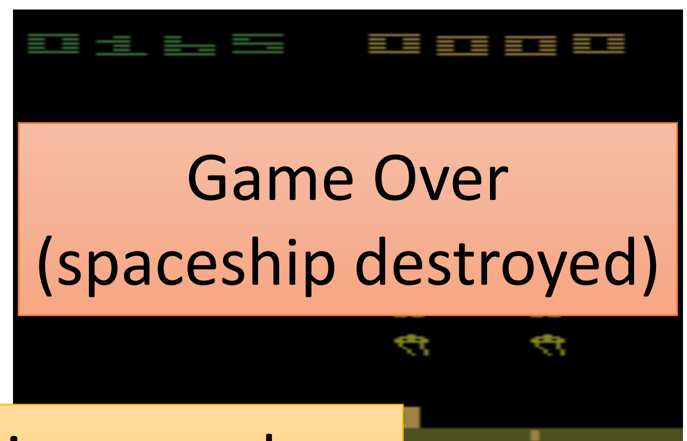
Observation s_2



Observatio



After many turns



Obtain reward r_T

This is an ***epi***

Total reward
(*return*):

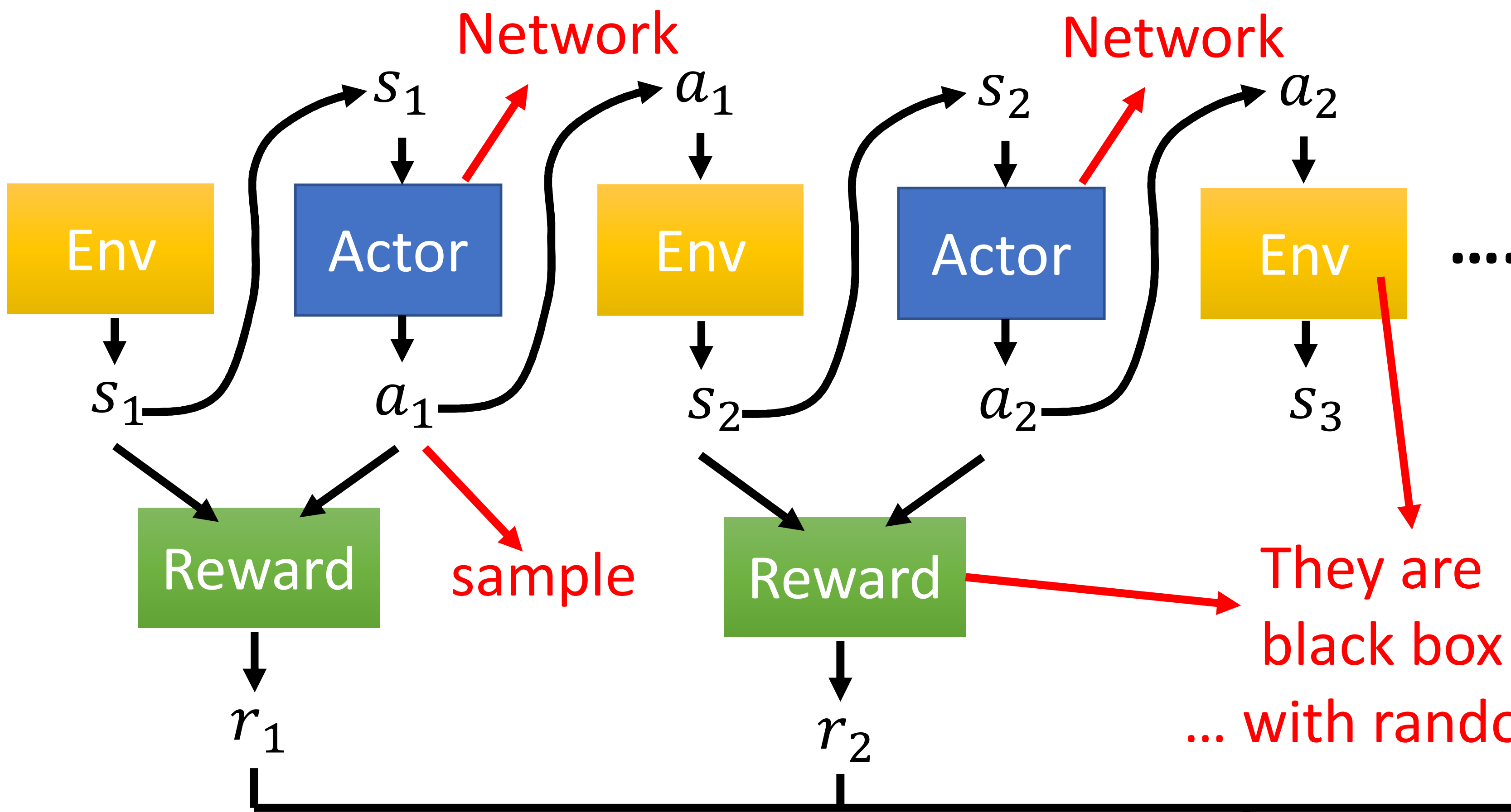
$R =$

What we w

Step 3: Optimization

Trajectory

$\tau = \{s_1, a_1, s_2, a_2,$



$R(\tau) = \sum^{T} r_t$

How to do the optimization here is

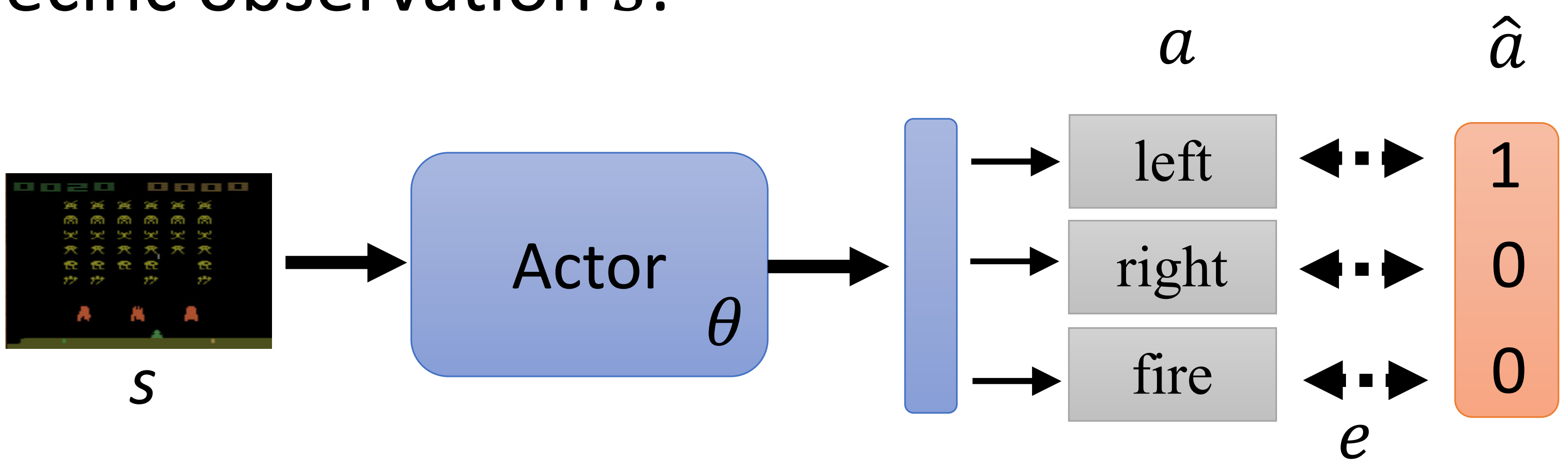
Outline

To learn more about policy gradien
https://youtu.be/W8XF3ME8G2I

- What is RL? (Three steps in ML)
- Policy Gradient
- Actor-Critic
- Reward Shaping
- No Reward: Learning from Demonstration

How to control your actor

- Make it take (or don't take) a specific action $\hat{a}$ giv specific observation s.



Take action $\hat{a}$

$$L = e$$

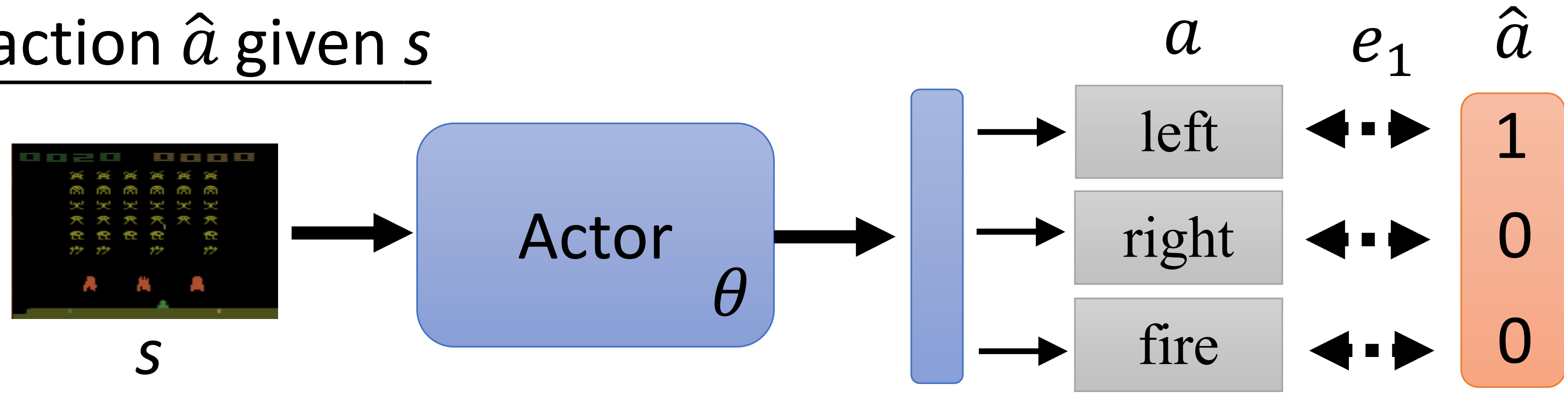
$$\theta^* = \arg\min_{\theta} L$$

Don't take action $\hat{a}$

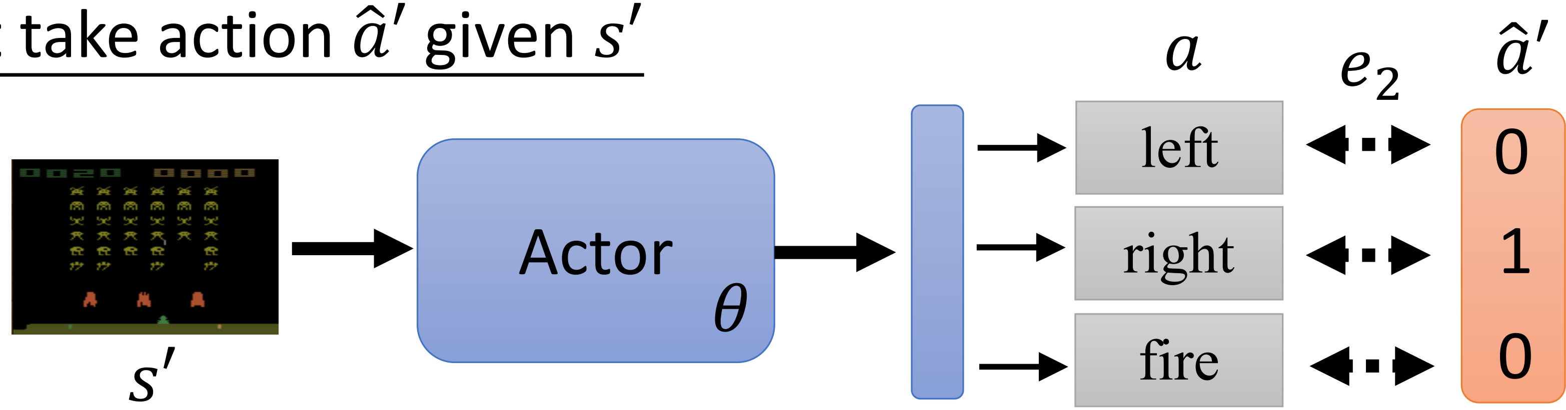
$$L = -e$$

How to control your actor

Take action $\hat{a}$ given s



Don't take action $\hat{a}'$ given s'



$L = e_1 - e_2 \qquad \theta^* = \arg\min L$

How to control your actor

Training Data

$\{s_1, \hat{a}_1\}$	+1	Yes
$\{s_2, \hat{a}_2\}$	-1	No
$\{s_3, \hat{a}_3\}$	+1	Yes
⋮	⋮	
$\{s_N, \hat{a}_N\}$	-1	No

$$L = + e_1 - e_2 + e_3 \cdots -$$

$$\theta^* = arg\min_{\theta} L$$

How to control your actor

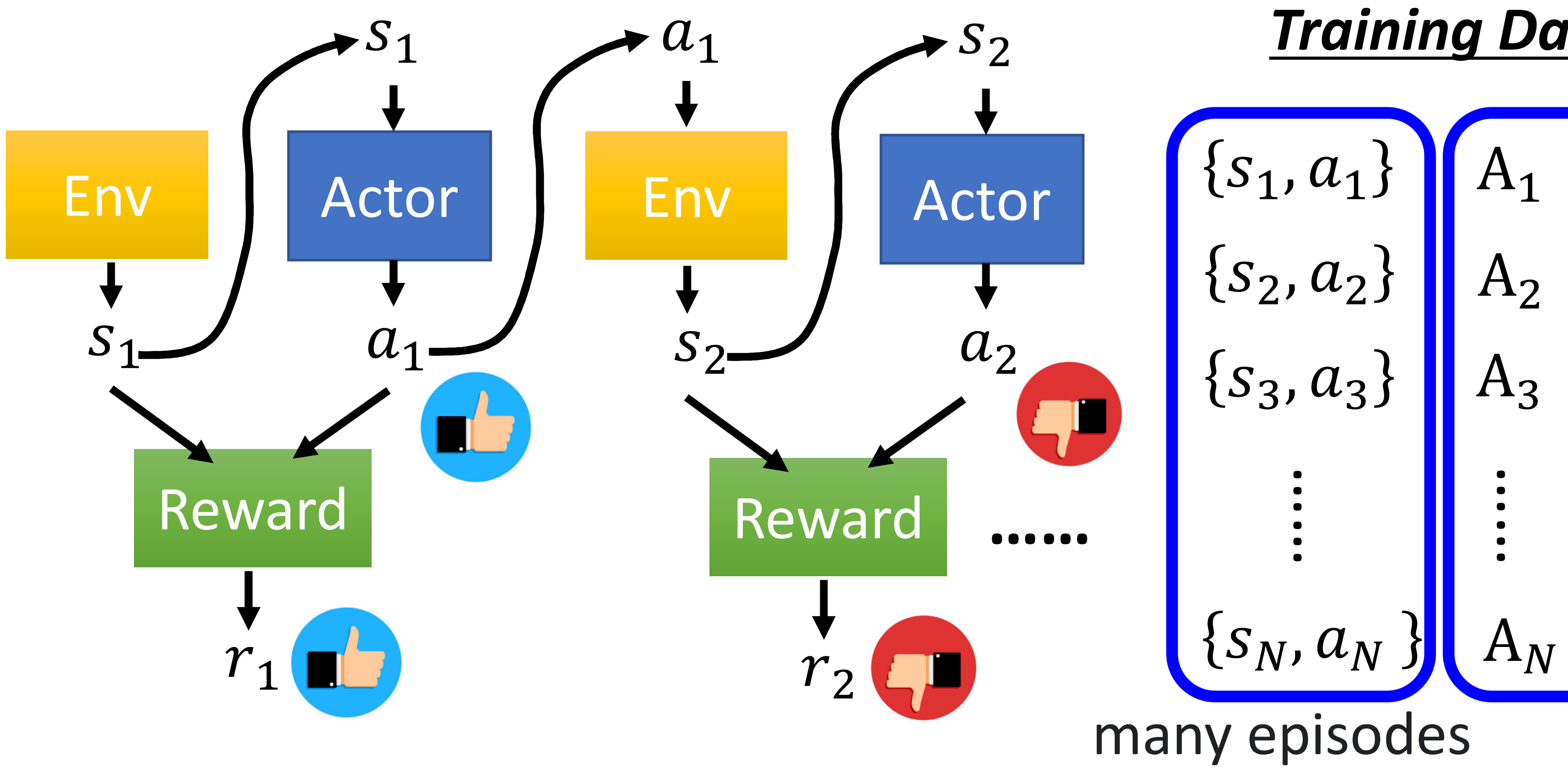
Training Data

$\{s_1, \hat{a}_1\}$	A_1	+1.5
$\{s_2, \hat{a}_2\}$	A_2	- 0.5
$\{s_3, \hat{a}_3\}$	A_3	+0.5
⋮	⋮	
$\{s_N, \hat{a}_N\}$	A_N	-10
?	?	

$$L = \sum A_n e_n$$

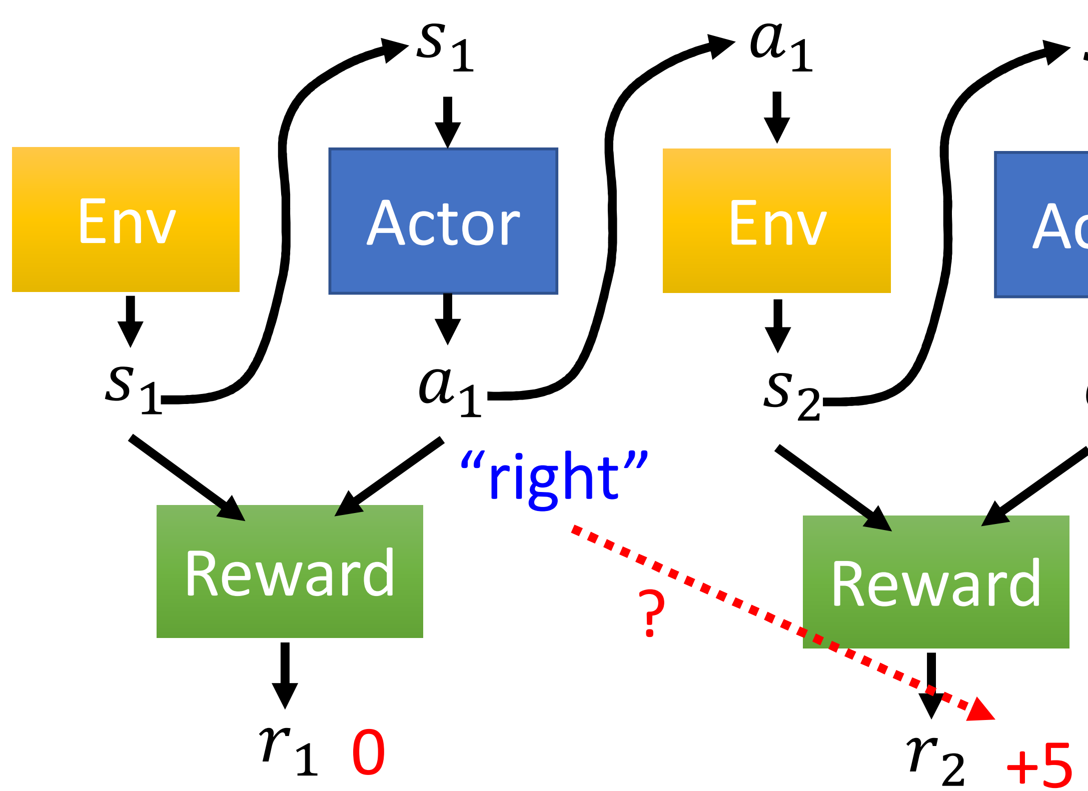
$$\theta^* = arg\min_{\theta} L$$

Version 0



Short-sighted Version!

Version 0



- An action affects the subsequent observations and thus subsequent rewards.
- *Reward delay*: Actor has to sacrifice immediate reward to gain more long-term reward.
- In space invader, only "fire" yields positive reward, so visi

Version 1

Training Data

$$\begin{array}{ccccc} s_1 & s_2 & s_3 & & s_N \\ \boxed{a_1} & \boxed{a_2} & \boxed{a_3} & \ldots\ldots & a_N \\ r_1 & r_2 & r_3 & & r_N \end{array}$$

$$\{s_1, a_1\} \quad A_1 =$$
$$\{s_2, a_2\} \quad A_2 =$$
$$\{s_3, a_3\} \quad A_3 =$$
$$\vdots \qquad \vdots$$
$$\{s_N, a_N\} \quad A_N =$$

$$G_1 = r_1 + r_2 + r_3 + \ldots\ldots + r_N$$

$$G_2 = r_2 + r_3 + \ldots\ldots + r_N$$

$$G_3 = r_3 + \ldots\ldots + r_N$$

$$G_t = \sum^{N} r$$

Version 2

Training Data

$$
\begin{array}{cccccc}
s_1 & s_2 & s_3 & & s_N \\
\boxed{a_1} & \boxed{a_2} & \boxed{a_3} & \cdots\cdots & a_N \\
r_1 & r_2 & r_3 & & r_N
\end{array}
$$

Also the credit of a_1?

$$G_1 = r_1 + r_2 + r_3 + \ldots\ldots + \boxed{r_N}$$

$$G'_1 = r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots\ldots$$

Discount factor $\gamma < 1$

$\{s_1, a_1\}$ $\mathrm{A}_1 =$

$\{s_2, a_2\}$ $\mathrm{A}_2 =$

$\{s_3, a_3\}$ $\mathrm{A}_3 =$

$\vdots$ $\vdots$

$\{s_N, a_N\}$ $\mathrm{A}_N =$

$$G'_t = \sum_{n=t}^{N} \gamma^{n-}$$

Version 3

s_1 s_2 s_3 s_N

$\boxed{a_1}$ $\boxed{a_2}$ $\boxed{a_3}$ …… a_N

r_1 r_2 r_3 r_N

Good or bad reward is "relative"

If all the $r_n \geq 10$

$r_n = 10$ is negative …

Minus by a baseline b ???

Training Data

$\{s_1, a_1\}$ $A_1 =$

$\{s_2, a_2\}$ $A_2 =$

$\{s_3, a_3\}$ $A_3 =$

⋮ ⋮

$\{s_N, a_N\}$ $A_N =$

$$G'_t = \sum_{n=t}^{N} \gamma^{n-t}$$

Policy Gradient

- Initialize actor network parameters θ^0
- For training iteration $i = 1$ to T
  - Using actor θ^{i-1} to interact
  - Obtain data $\{s_1, a_1\}, \{s_2, a_2\}, \ldots, \{s_N, a_N\}$
  - Compute $A_1, A_2, \ldots, A_N$
  - Compute loss L
  - $\theta^i \leftarrow \theta^{i-1} - \eta \nabla L$

Data collection is in the
loop" of training iteratio

Policy Gradient

Training Data

$\{s_1, a_1\}$	A_1
$\{s_2, a_2\}$	A_2
$\{s_3, a_3\}$	A_3
$\vdots$	$\vdots$
$\{s_N, a_N\}$	A_N

$s \longrightarrow$ Actor θ $\longrightarrow a$

$$L = \sum A_n e_n$$

$$\theta^i \leftarrow \theta^{i-1} - \eta \nabla L$$

only update onc

Each time you update the model parameters, you need t

Policy Gradient

- Initialize actor network parameters θ^0
- For training iteration $i = 1$ to T
  - Using actor $\boxed{\theta^{i-1}}$ to interact
  - Obtain data $\{s_1, a_1\}, \{s_2, a_2\}, \ldots, \{s_N, a_N\}$
  - Compute $A_1, A_2, \ldots, A_N$
  - Compute loss L
  - $\boxed{\theta^i} \leftarrow \boxed{\theta^{i-1}} - \eta \nabla L$

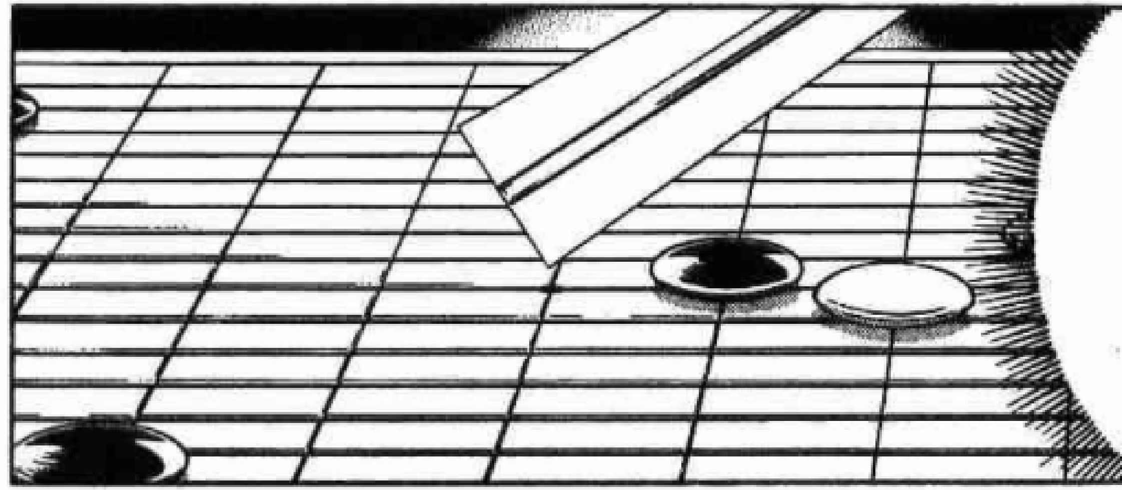
Experience of

May not be good for θ^i

棋魂第八集

阿光，這時候不要下小馬步飛，而是要大馬步飛。
※ 小馬步飛：跟將棋一樣，將棋子放在斜一格；大馬步飛則是放在斜好幾格。
大馬？
我覺得小馬也不錯啊！

棋魂第八集

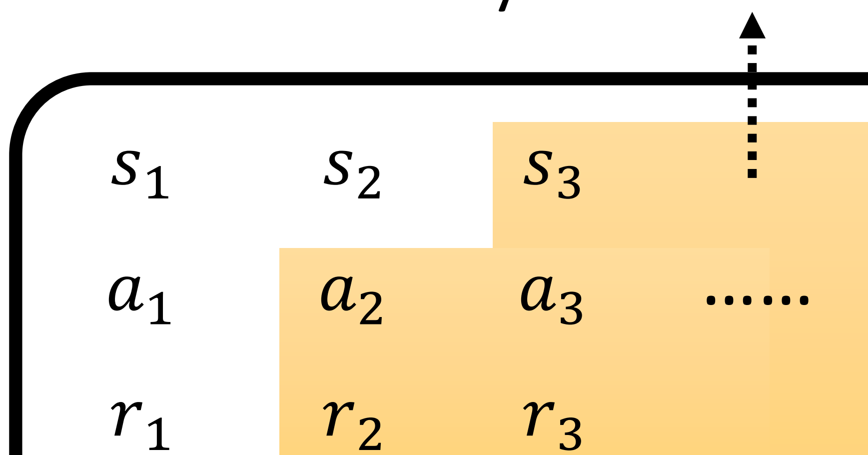


Policy Gradient

- Initialize actor network parameters θ^0
- For training iteration $i = 1$ to T
  - Using actor θ^{i-1} to interact
  - Obtain data $\{s_1, a_1\}, \{s_2, a_2\}, \dots, \{s_N, a_N\}$
  - Compute $A_1, A_2, \dots, A_N$
  - Compute loss L
  - $\theta^i \leftarrow \theta^{i-1} - \eta \nabla L$

May not observe

Trajectory of θ^{i-1}



On-policy v.s. Off-policy

- The actor to train and the actor for interacting is the same. → On-policy
- Can the actor to train and the actor for interactin be different? → Off-policy

Trajectory of θ^{i-1}

s_1	s_2	s_3		s_N
a_1	a_2	a_3	……	a_N
r_1	r_2	r_3		r_N

➡ To tr

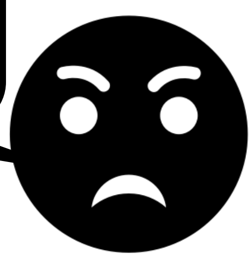
In this way, we do not have to collection data after each u

Off-policy → Proximal Policy Optimization (PPO)

- The actor to train has to know its difference from the actor to interact.

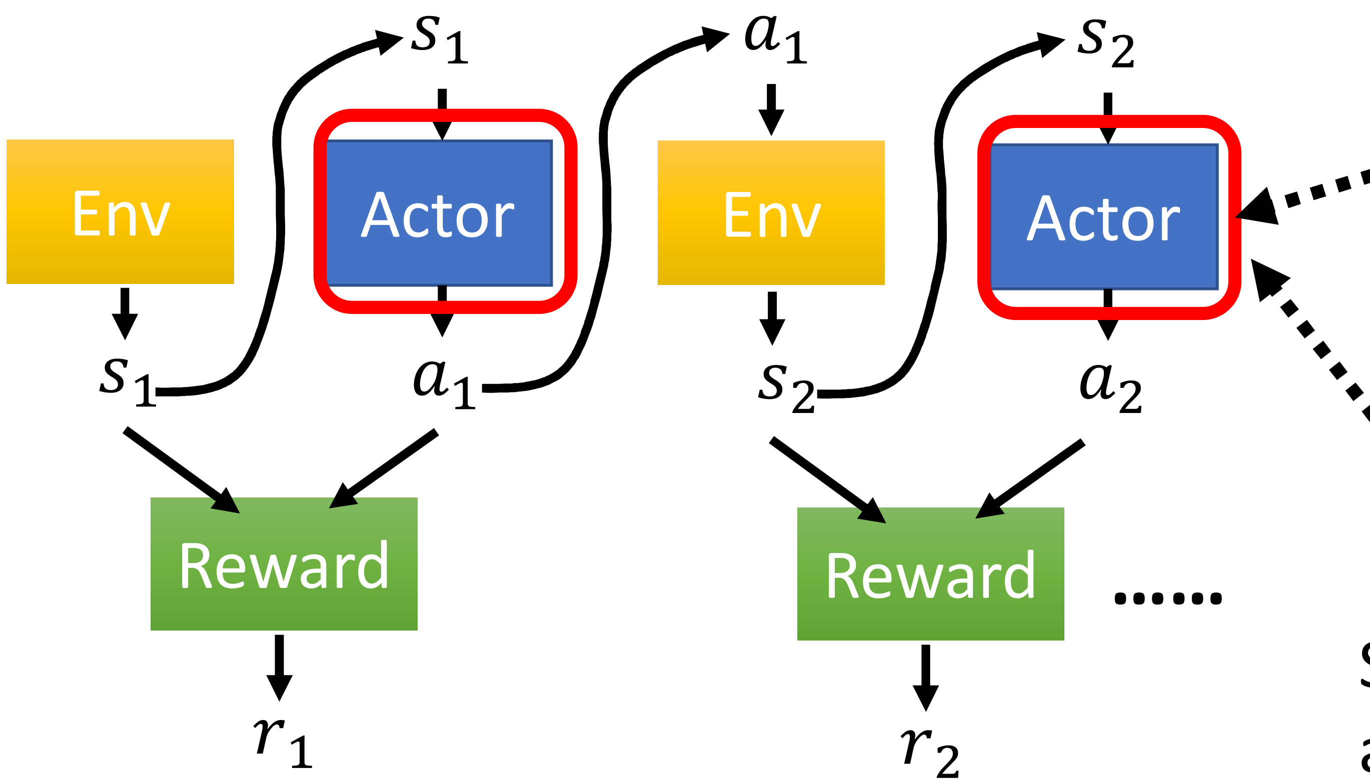
video: https://youtu.be/OAKAZhFmYoI

Not apply to everyone



the actor to train

Collection Training Data: **Exploration**



Enlarge out
entropy

Add noises
parameters

Suppose your
always takes "

We never kno
what would h
if taking "fire"

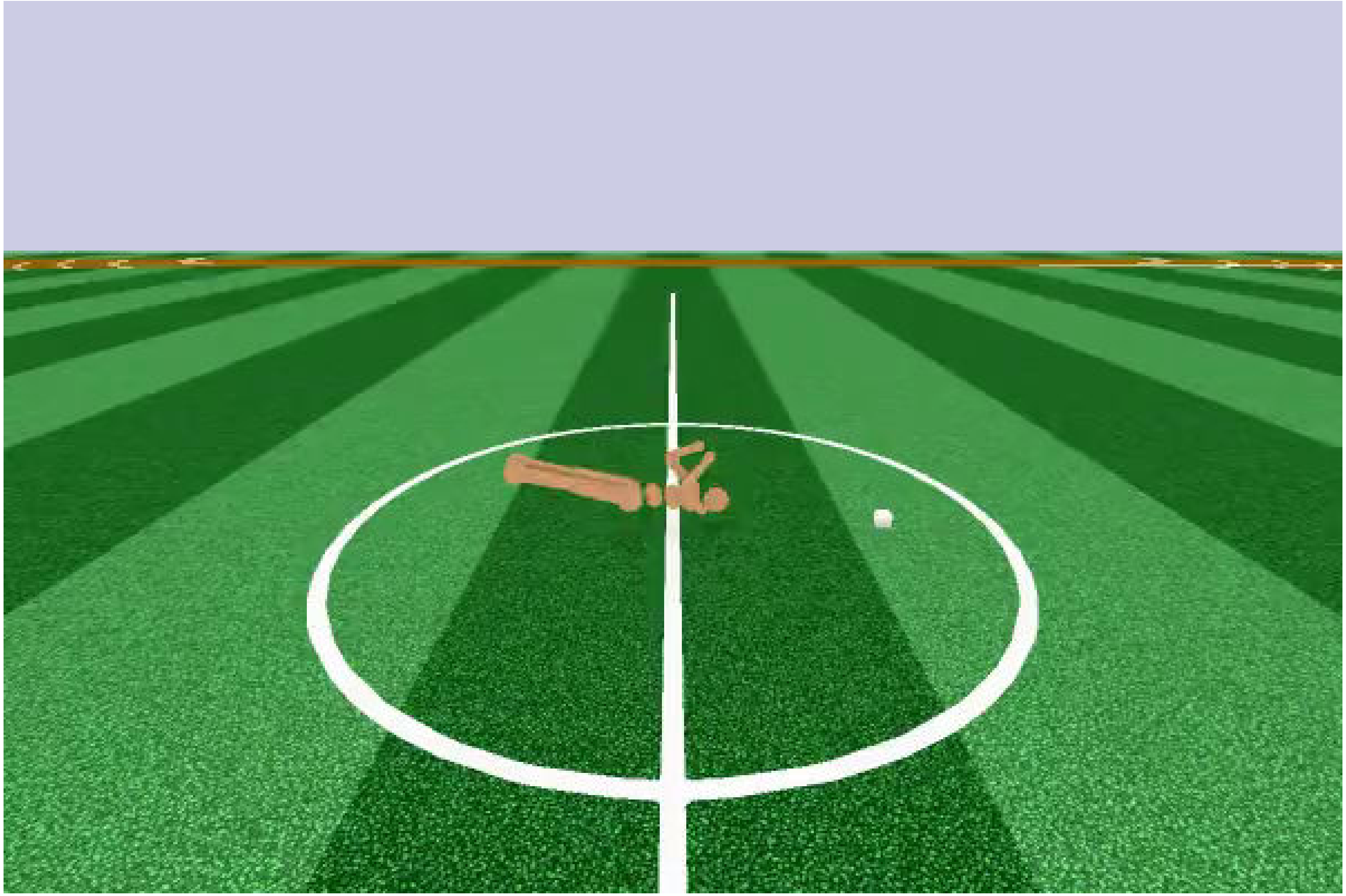
The actor needs to have randomness during data collection.

DeepMind - PPO

https://youtu.be/gn4nRCC9T

OpenAI - PPO

https://blog.openai.cor
penai-baselines-ppo/



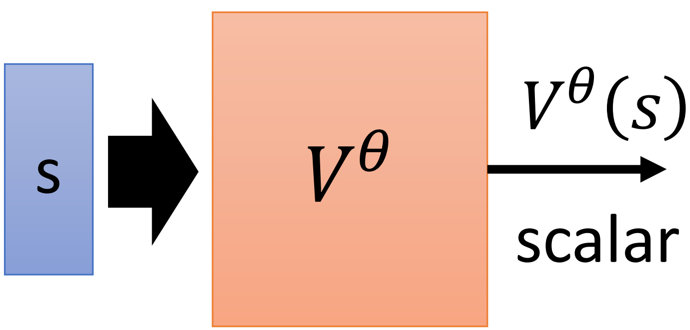
Outline

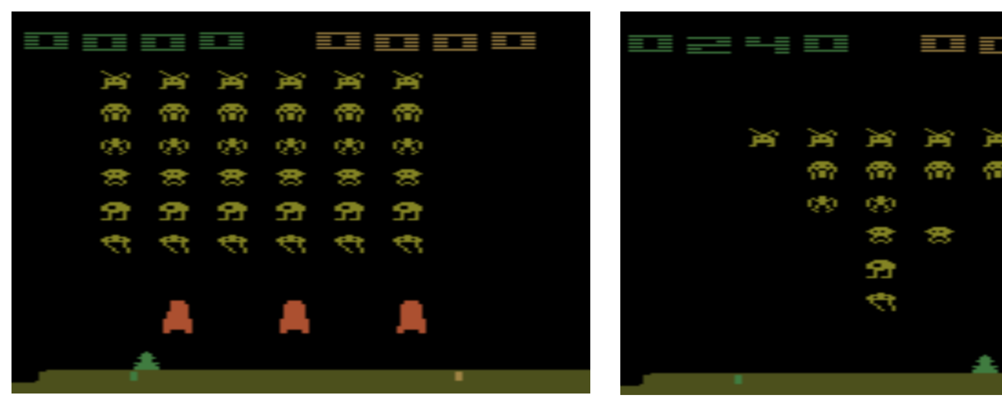
- What is RL? (Three steps in ML)
- Policy Gradient
- Actor-Critic
- Reward Shaping
- No Reward: Learning from Demonstration

Critic

$$G'_1 = r_1 + \gamma r_2 + \gamma^2 r_3 +$$

- Critic: Given actor θ, how good it is when observing s (and taking action a)
- Value function $V^{\boxed{\theta}}(s)$: When using actor θ, the discounted *cumulated* reward expects to be obtained after seeing s





$V^\theta(s)$ is large

$V^\theta(s)$ is sm

The output values of a critic depend on the actor evaluat

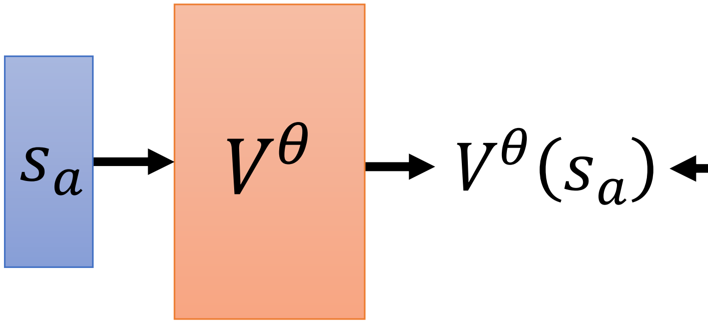
How to estimate $V^{\theta}(s)$

- **Monte-Carlo (MC) based approach**

The critic watches actor θ to interact with the environmen

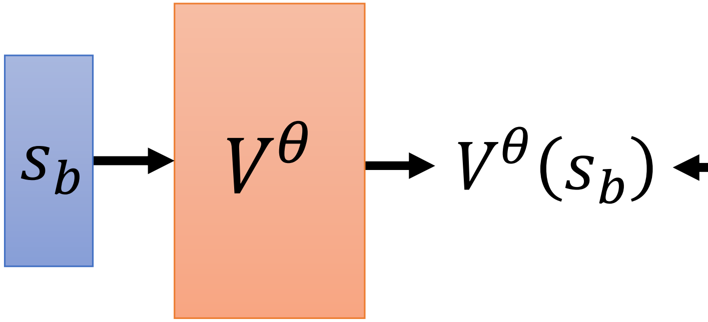
After seeing s_a,

Until the end of the episode, the cumulated reward is G'_a



After seeing s_b,

Until the end of the episode, the cumulated reward is G'_b



How to estimate $V^{\pi}(s)$

- **Temporal-difference (TD) approach**

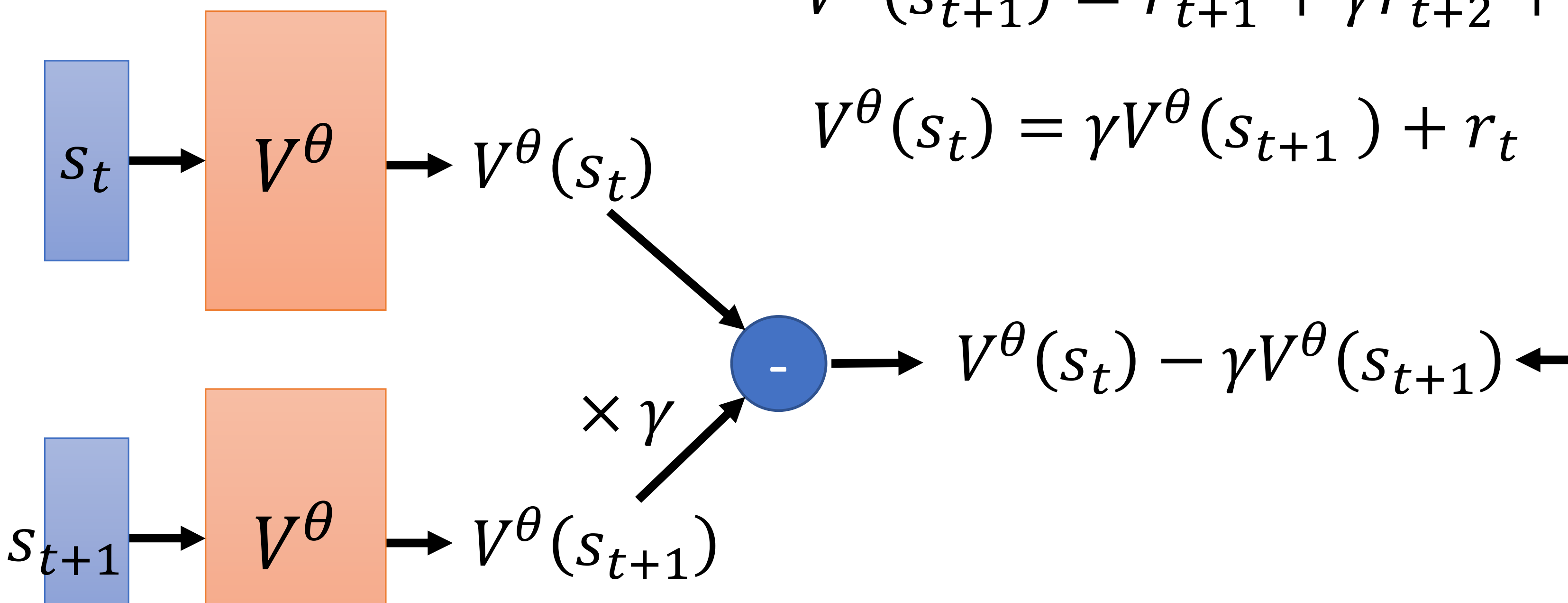
$$\cdots s_t, a_t, r_t, s_{t+1} \cdots$$

(ignore the expectati

$$V^{\theta}(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_t$$

$$V^{\theta}(s_{t+1}) = r_{t+1} + \gamma r_{t+2} +$$

$$V^{\theta}(s_t) = \gamma V^{\theta}(s_{t+1}) + r_t$$



MC v.s. TD

[Sutton, v
Example (

- The critic has observed the following 8 episodes
  - $s_a, r = 0, s_b, r = 0$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 0$, END

$$V^\theta(s_b) = 3/4$$

$$V^\theta(s_a) = ?$$ 0? 3/4?

Monte-Carlo: $V^\theta(s_a) = 0$

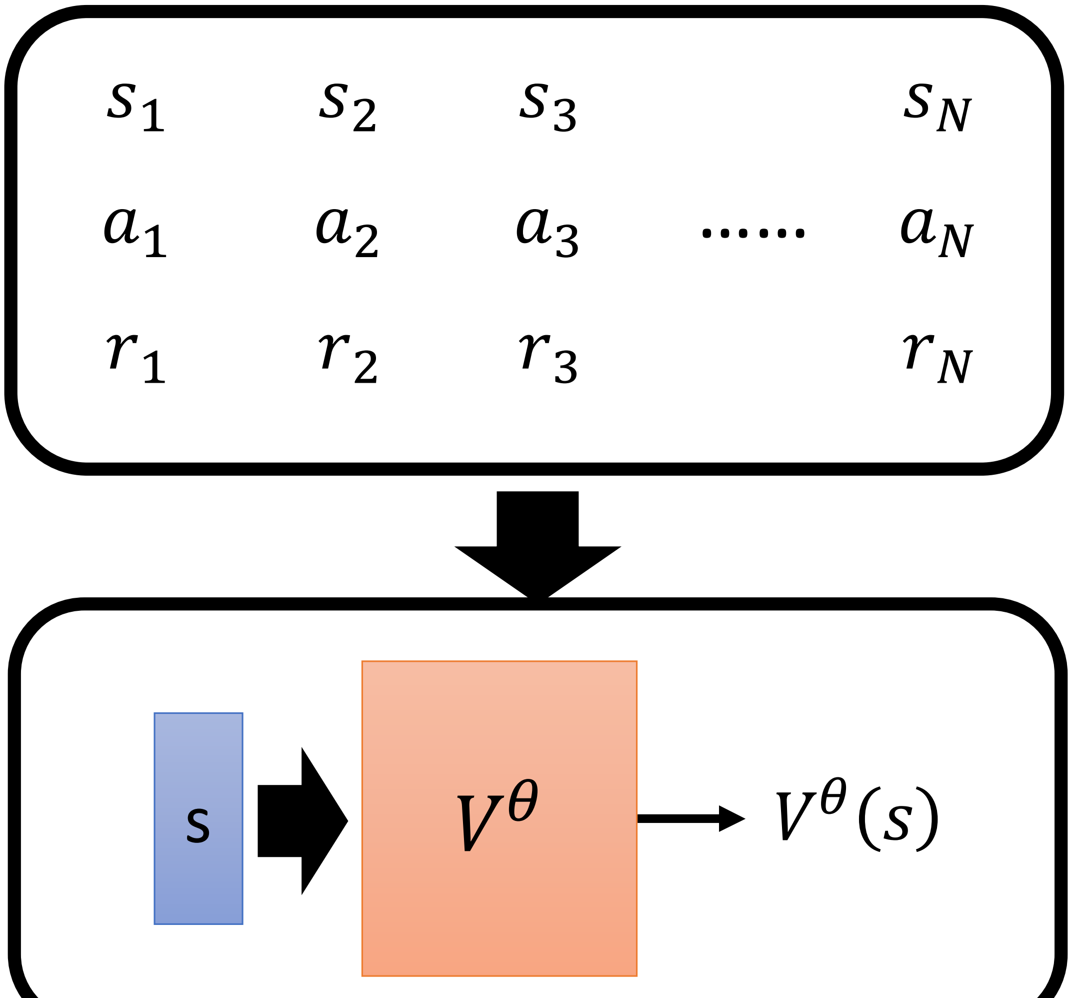
Temporal-difference:

$$V^\theta(s_a) = V^\theta(s_b) + r$$

3/4 3/4 0

(Assume γ = 1, and the
actions are ignored here.)

Version 3.5



Training Data

$\{s_1, a_1\}$ $A_1 = G'_1 -$

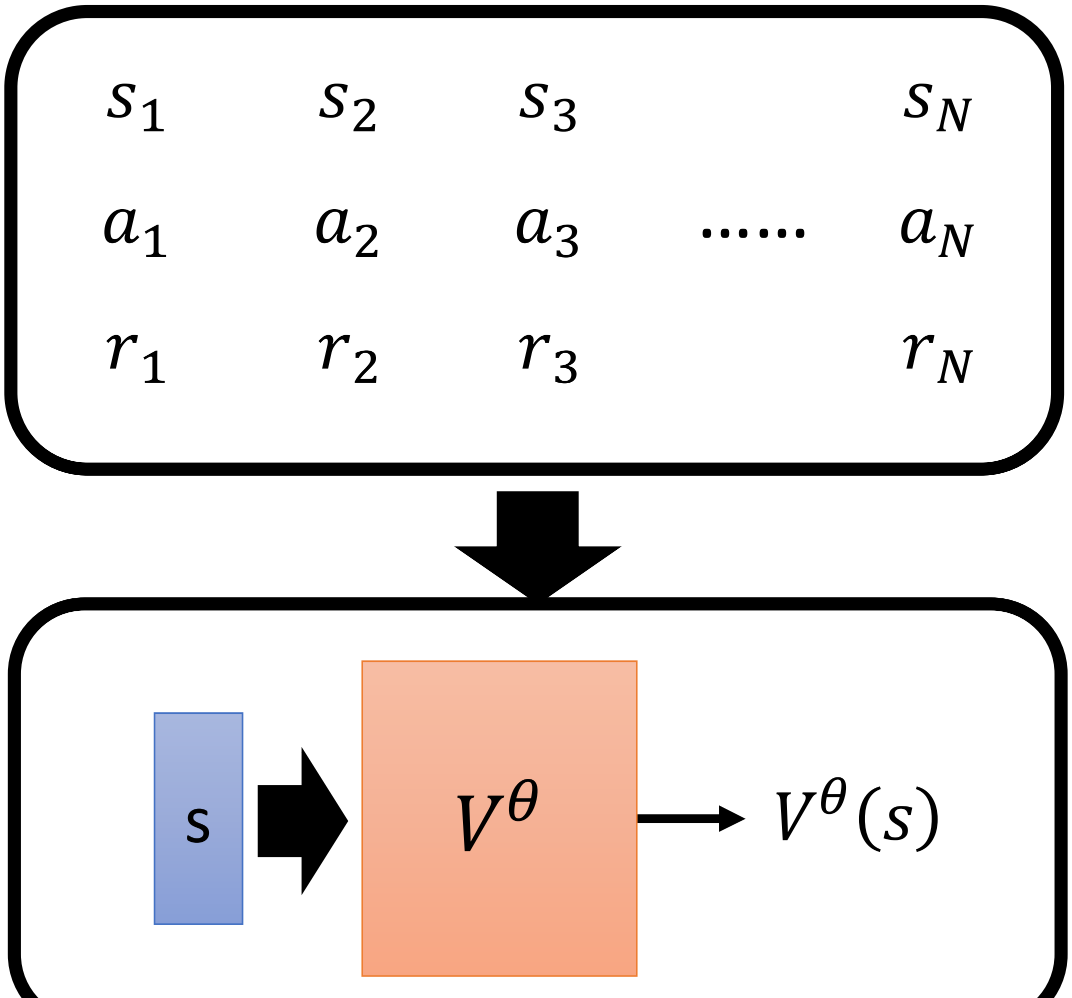
$\{s_2, a_2\}$ $A_2 = G'_2 -$

$\{s_3, a_3\}$ $A_3 = G'_3 -$

$\vdots$ $\vdots$

$\{s_N, a_N\}$ $A_N = G'_N -$

Version 3.5



<u>Training Data</u>

$\{s_1, a_1\}$ $\quad A_1 = G'_1 -$

$\{s_2, a_2\}$ $\quad A_2 = G'_2 -$

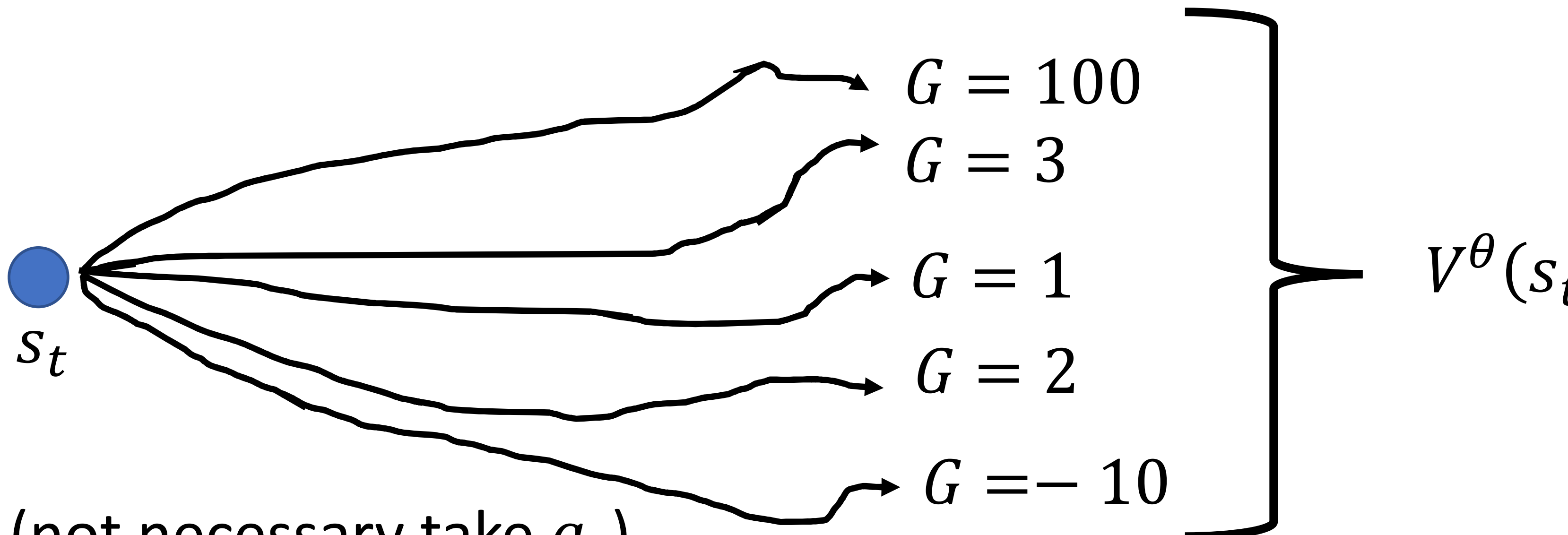
$\{s_3, a_3\}$ $\quad A_3 = G'_3 -$

⋮ ⋮

$\{s_N, a_N\}$ $\quad A_N = G'_N -$

Version 3.5

$\{s_t, a_t\}$ $\quad \mathrm{A}_t = G'_t - V^{\theta}(s_t)$

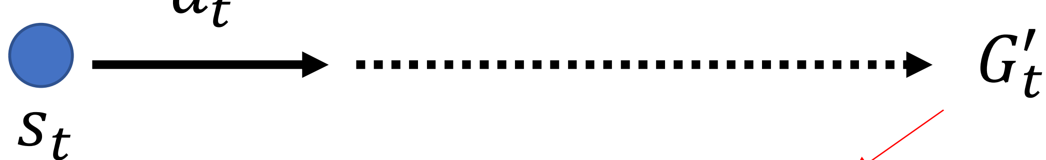


(not necessary take a_t)

(You sample the actions based on a distribution)

$\mathrm{A}_t > 0$

a_t is better than a



$\mathrm{A}_t < 0$

a_t is worse than a

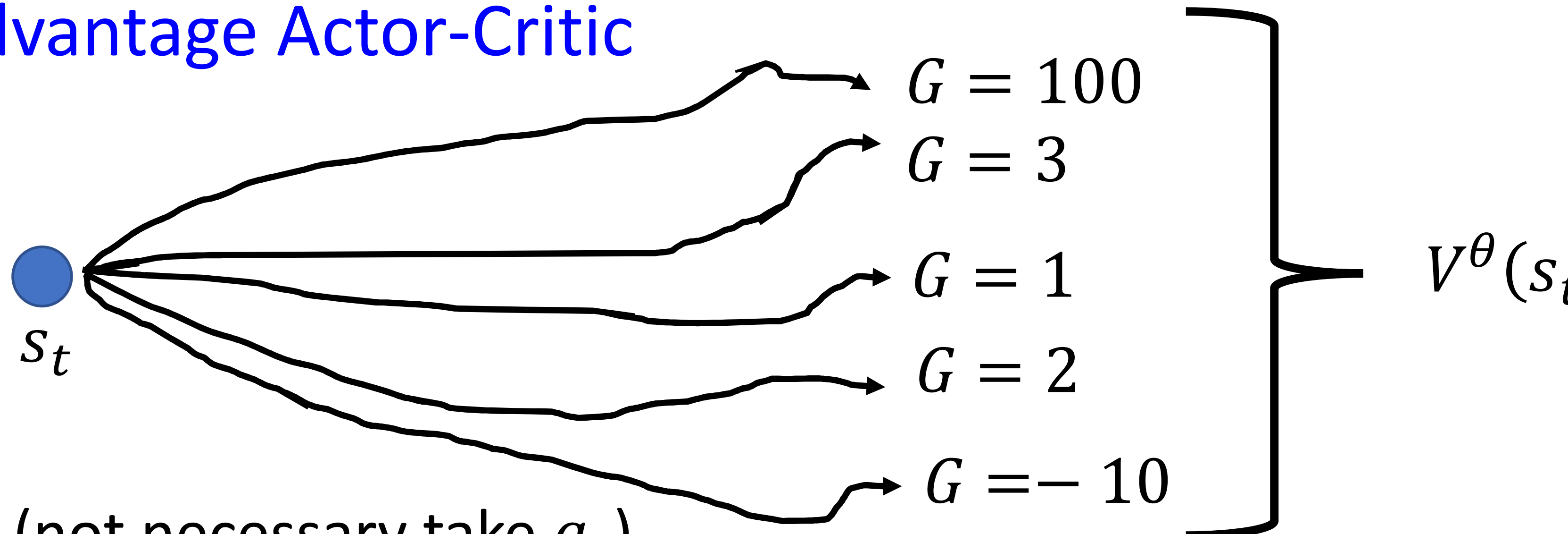
Just a sample

$$r_t + V^{\theta}(s_{t+1}) - V^{\theta}$$

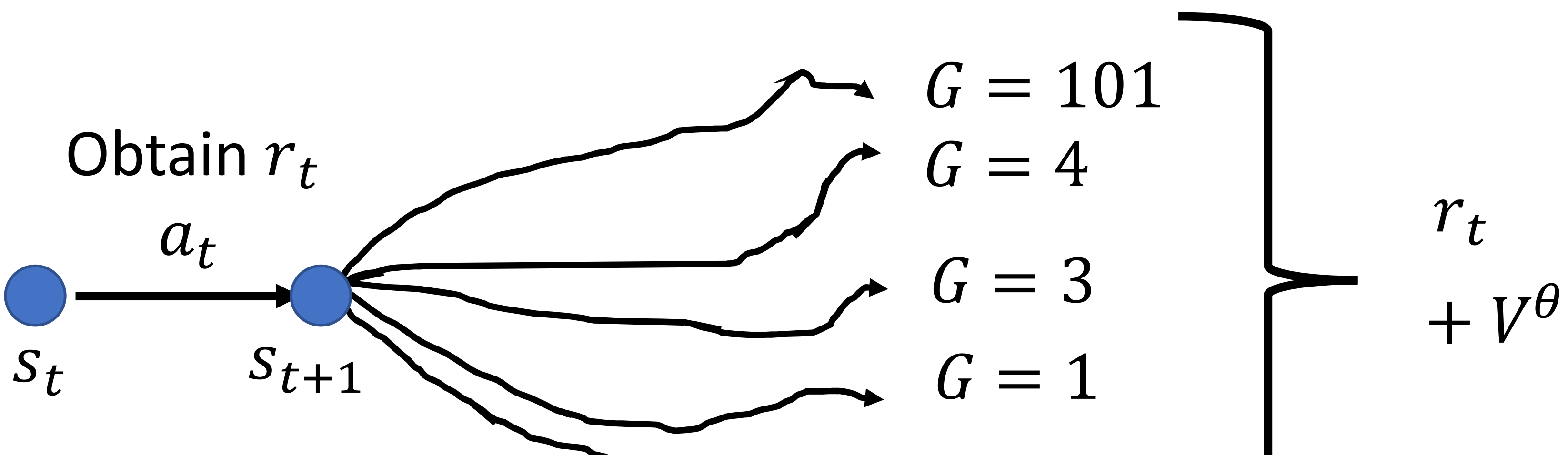
Version 4

$\{s_t, a_t\}$ $\quad$ $A_t = \cancel{G'_t - V^{\theta}(s_t)}$

Advantage Actor-Critic

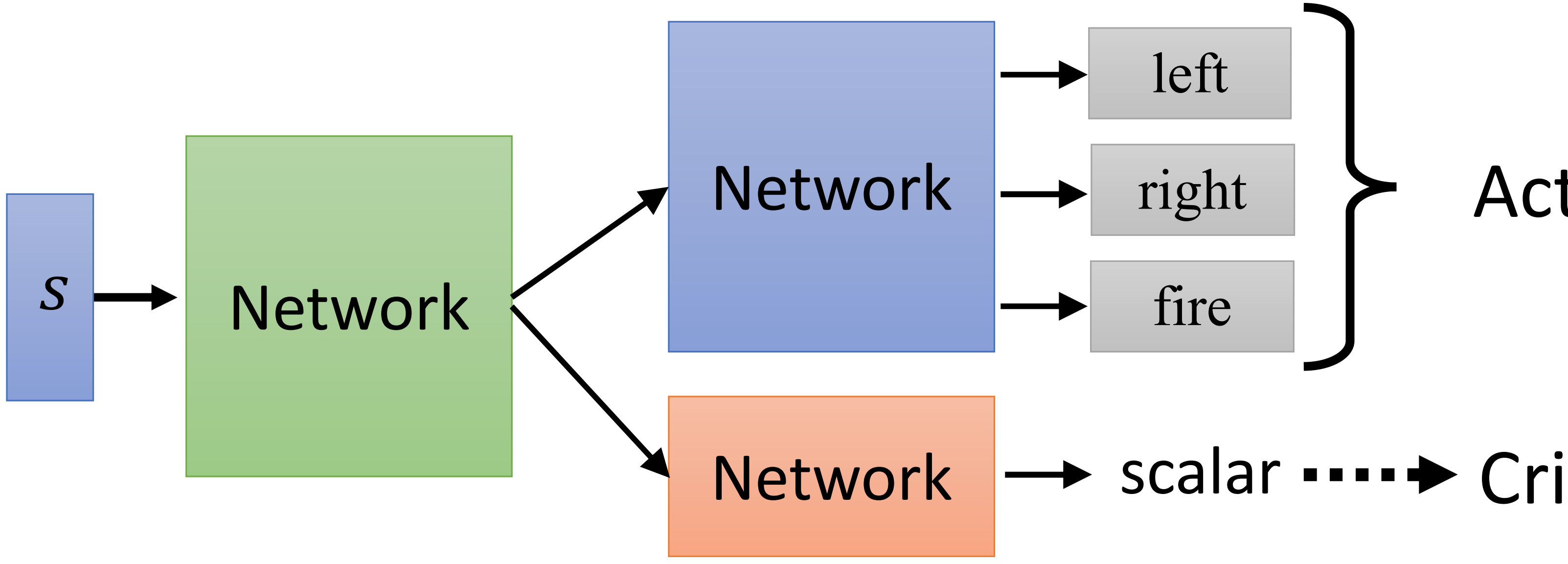


(not necessary take a_t)



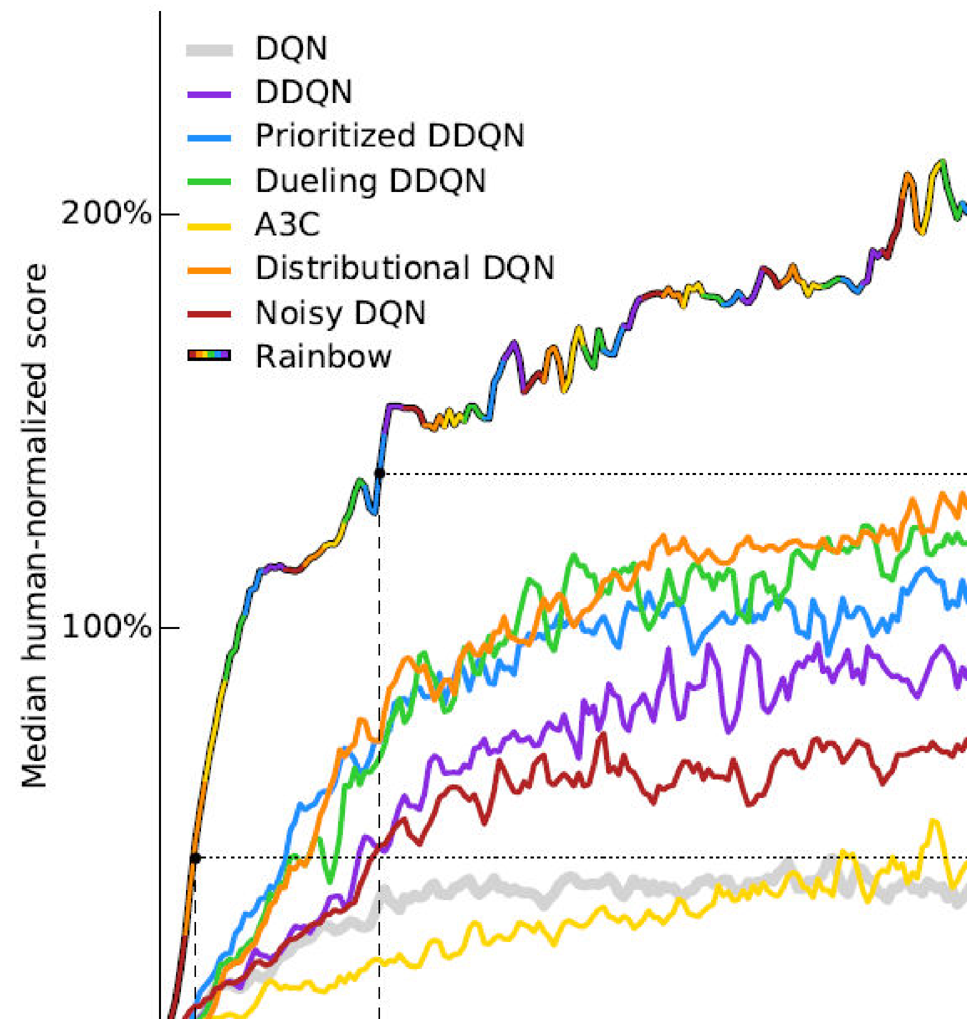
Tip of Actor-Critic

- The parameters of actor and critic can be shared.



Outlook: Deep Q Network (DQN)

Video:
https://youtu.be/o_g9JUMw1Oc
https://youtu.be/2-zGCx4iv_k



Outline

- What is RL? (Three steps in ML)
- Policy Gradient
- Actor-Critic
- Reward Shaping
- No Reward: Learning from Demonstration

Sparse Reward

$$A_t = r_t + V^{\theta}(s_{t+1}) - V$$

s_1	s_2	s_3		s_N
a_1	a_2	a_3	……	a_N
r_1	r_2	r_3		r_N

Training Data

$\{s_1, a_1\}$	A_1
$\{s_2, a_2\}$	A_2
$\{s_3, a_3\}$	A_3
$\vdots$	$\vdots$
$\{s_N, a_N\}$	A_N

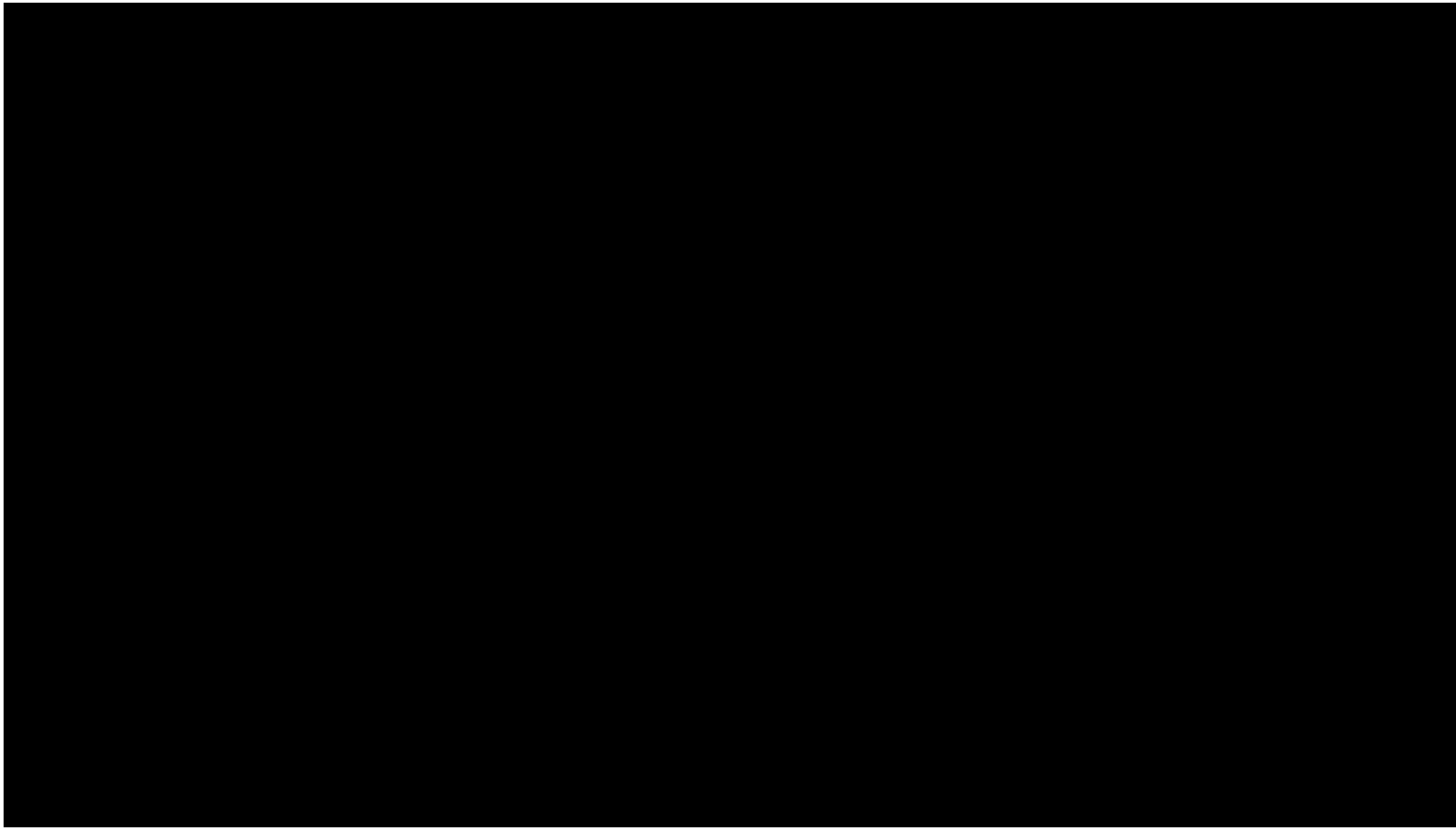
If $r_t = 0$ in most cases ➡ We don't know act are good or bad.

e.g., robot arm to bolt on the screws

The developers define extra ➡

Reward Shaping

VizDoom https://openreview.net/forum?id=Hk3mPK5gg&noteId=Hk3mPK5gg

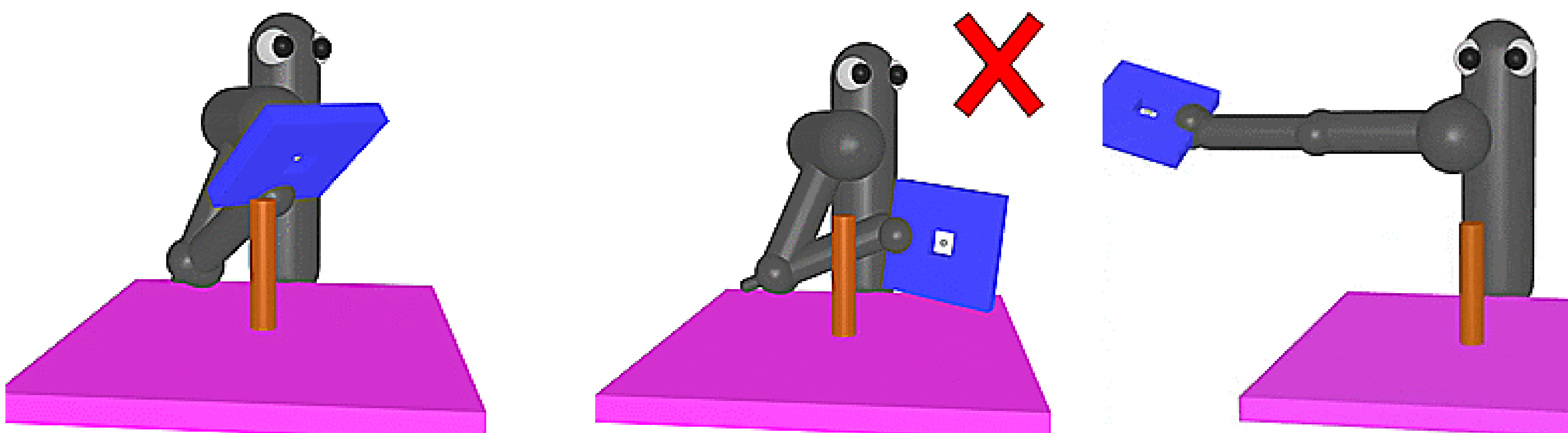


Visual Doom AI Competition @ CIG 2016

Reward Shaping

VizDoom https://openreview.net/forum?id=Hk3mPK5gg&noteId=Hk3mPK5gg

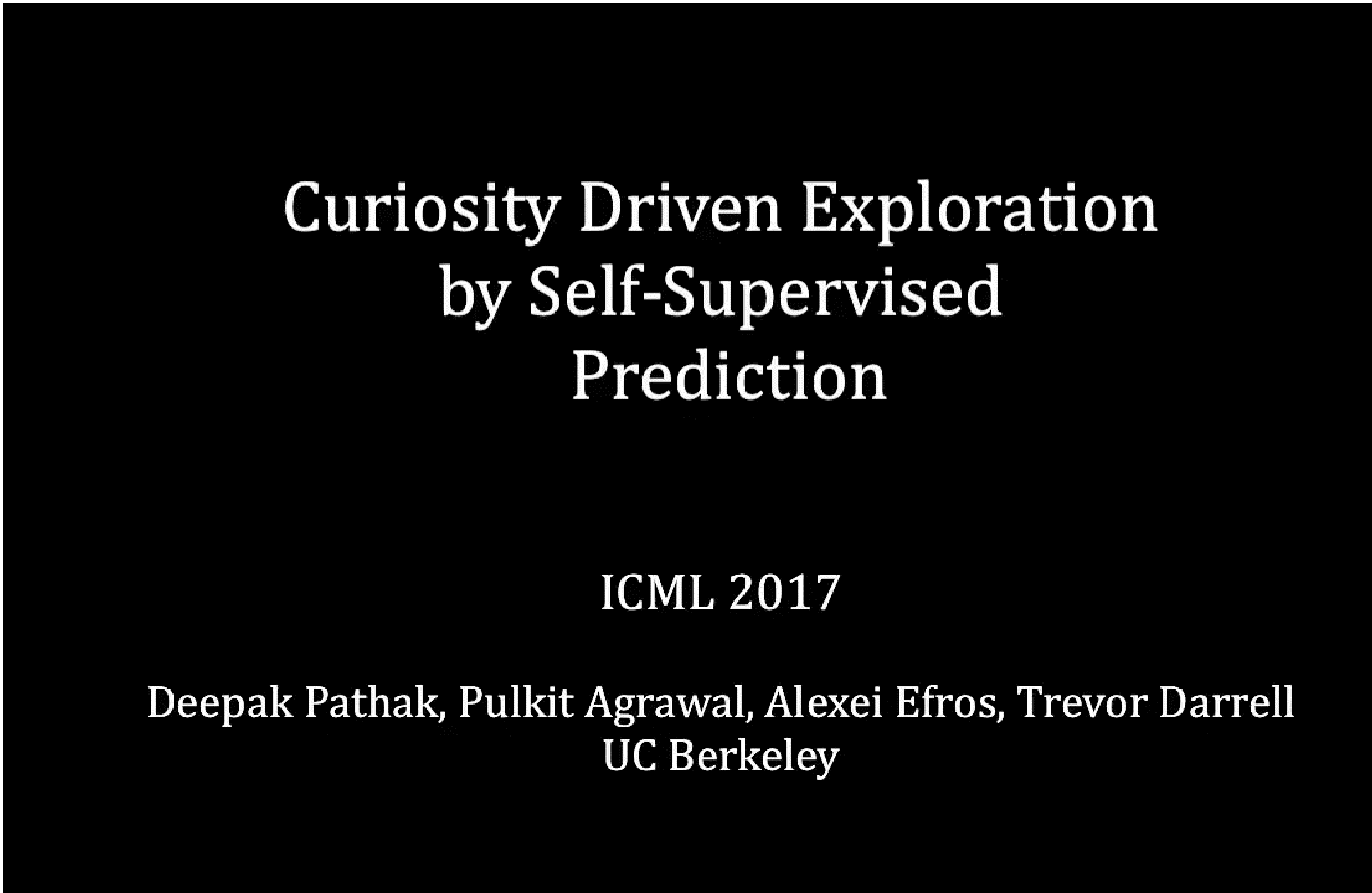
Parameters	Description	FlatMap	CIG
living	Penalize agent who just lives	-0.008 / acti	
health_loss	Penalize health decrement	-0.05 / uni	
ammo_loss	Penalize ammunition decrement	-0.04 / uni	
health_pickup	Reward for medkit pickup	0.04 / unit	
ammo_pickup	Reward for ammunition pickup	0.15 / unit	
dist_penalty	Penalize the agent when it stays	-0.03 / actio	
dist_reward	Reward the agent when it moves	9e-5 / unit dist	



Reward Shaping - Curiosity

https://arxiv.org/abs/17

Obtaining extra reward when the agent sees something new (but meaningful).



Outline

- What is RL? (Three steps in ML)
- Policy Gradient
- Actor-Critic
- Reward Shaping
- No Reward: Learning from Demonstration

Motivation

- Even define reward can be challenging in some tasks.
- Hand-crafted rewards can lead to uncontrolled behavior.

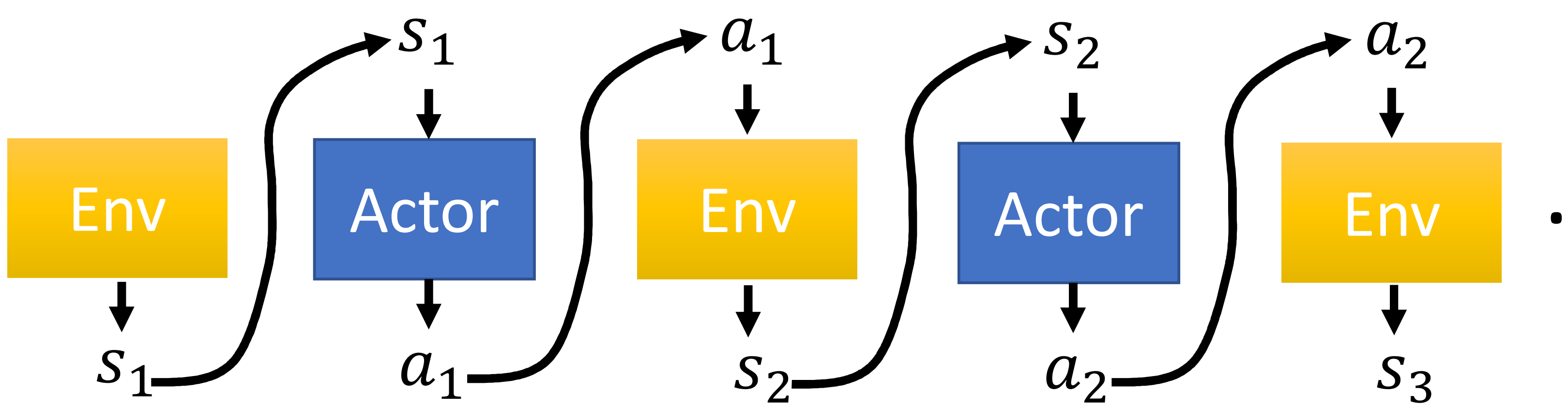
Three Laws of Robotics:

1. A robot may not injure a human being or, thro
inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by huma
except where such orders would conflict with the
3. A robot must protect its own existence as long
protection does not conflict with the First or Sec

?????

restraining individual human behavior and sacrifi

Imitation Learning



Actor can interact with the environment, but reward function is not available

We have demonstration of the expert.

$\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_K\}$

Each $\hat{\tau}$ is a trajectory of the export.

Self driving: r
human driver

Robot: grab th

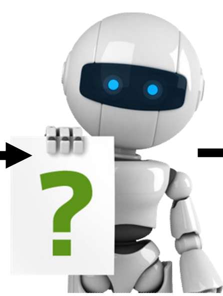
Isn’t it Supervised Learning?

- Self-driving cars as example

$$\hat{\tau} = \{s_1, \hat{a}_1, \ s_2, \hat{a}_2, \ \cdots\}$$

forward

$s_i \longrightarrow$

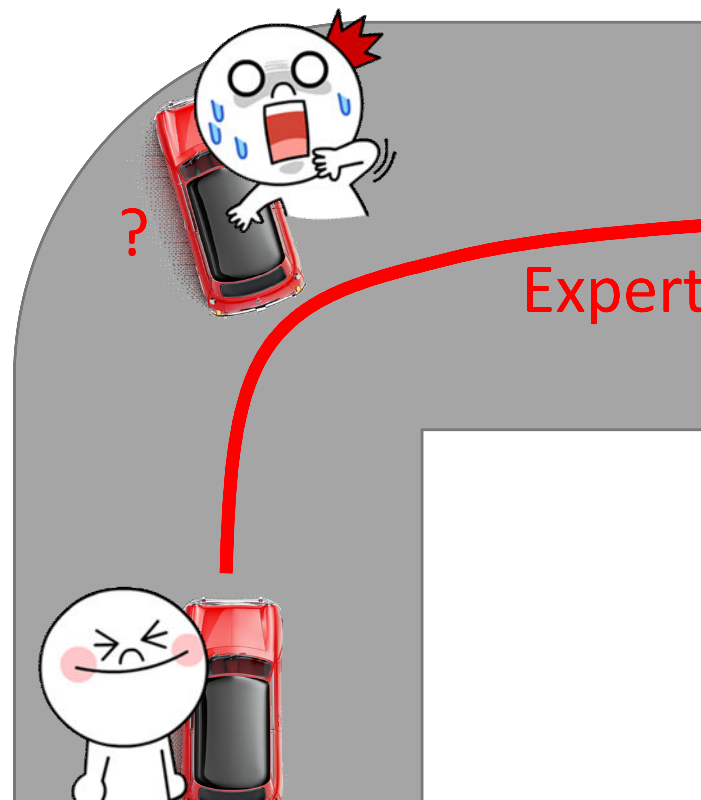


$\longrightarrow \ a_i \longleftrightarrow \hat{a}_i$

Problem: The experts only

Yes, also known a

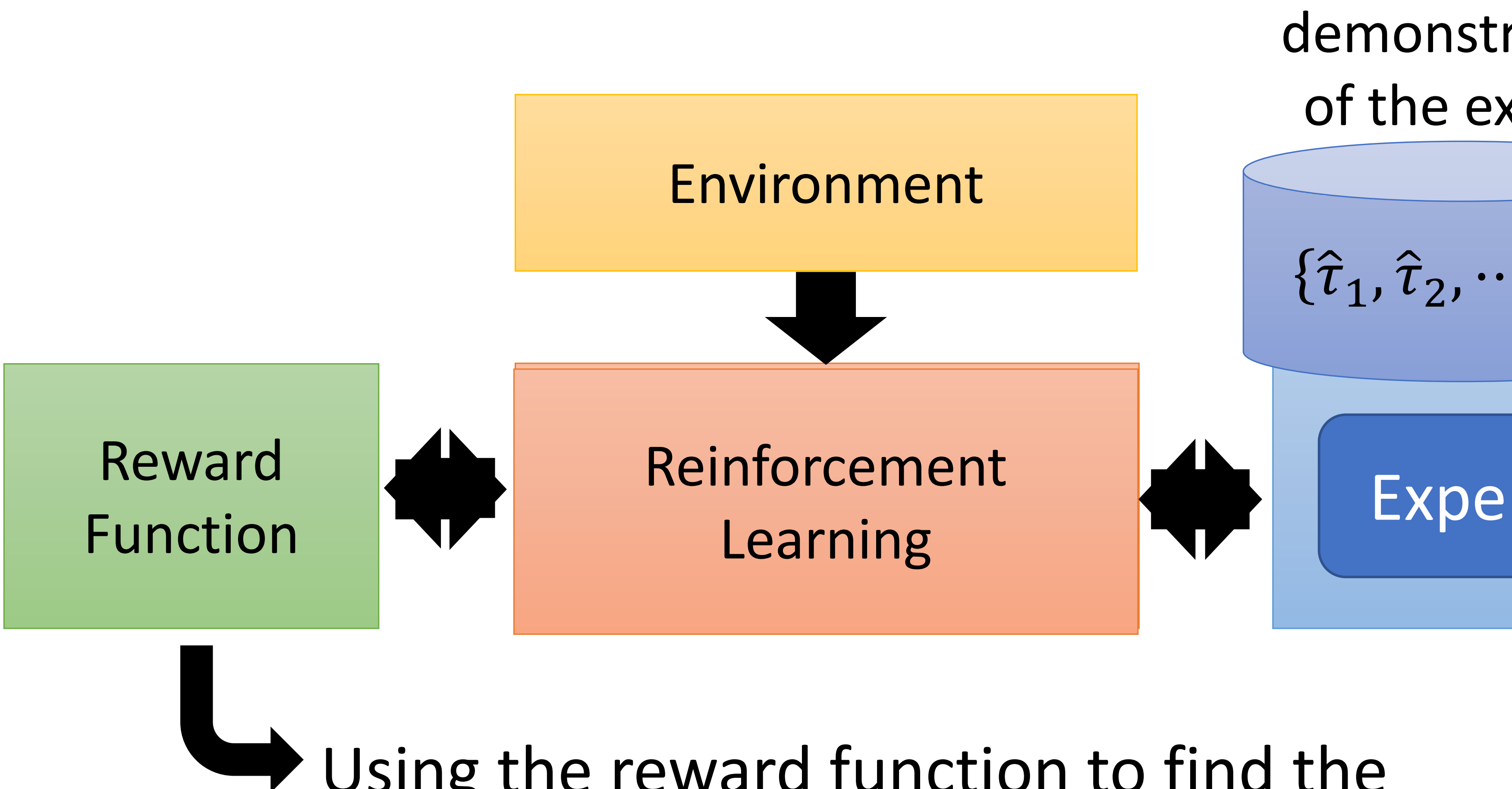
Behavior Cloning



More problem ……

The agent will co
every behavior,
irrelevant action

Inverse Reinforcement Learning

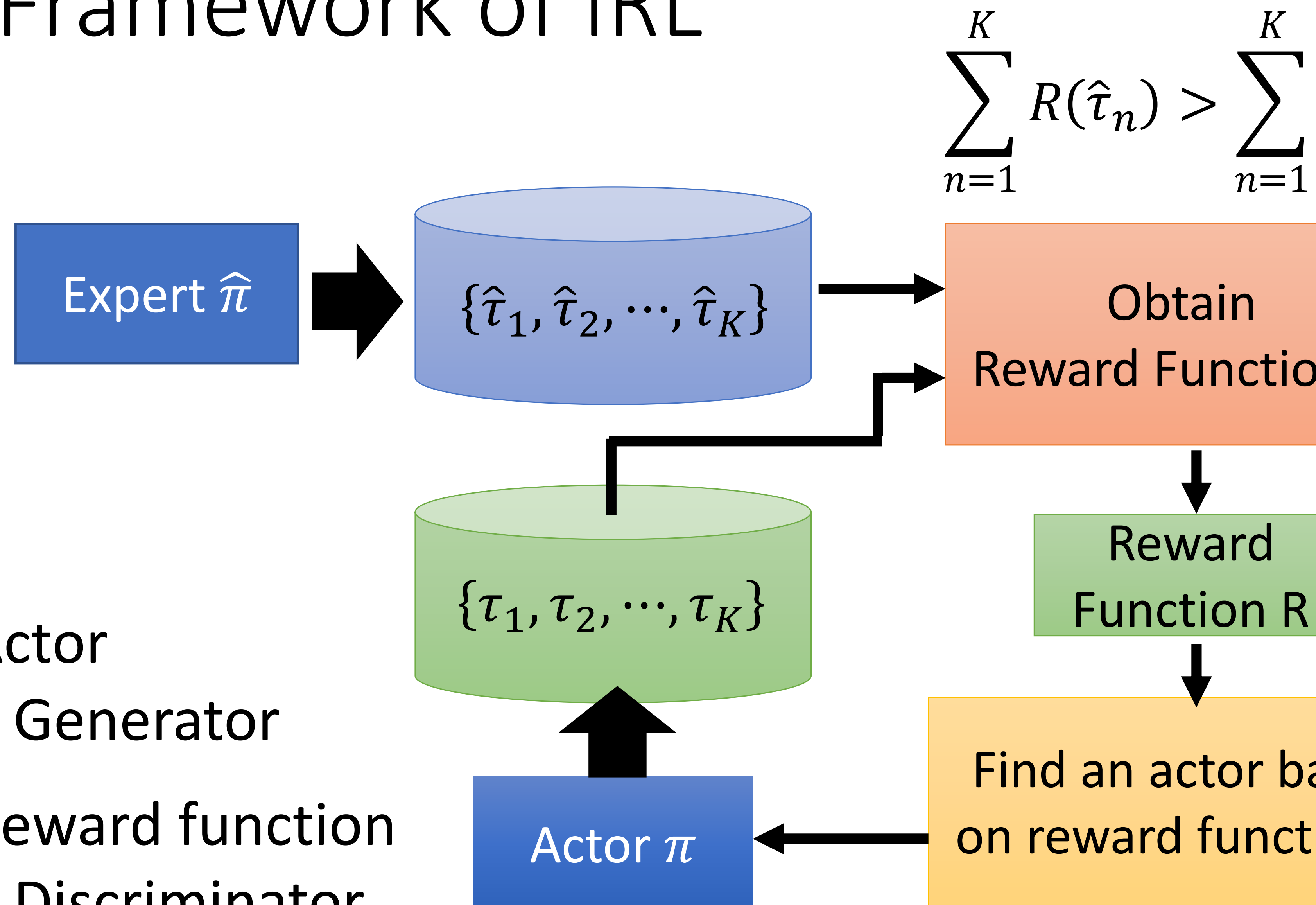


Using the reward function to find the ***optimal actor***.

Inverse Reinforcement Learning

- Principle: ***<u>The teacher is always the best</u>***.
- Basic idea:
  - Initialize an actor
  - In each iteration
    - The actor interacts with the environments to obta some trajectories.
    - Define a reward function, which makes the trajectories of the teacher better than the actor.
    - The actor learns to maximize the reward based on the new reward function.
  - Output the reward function and the actor learned fro

Framework of IRL

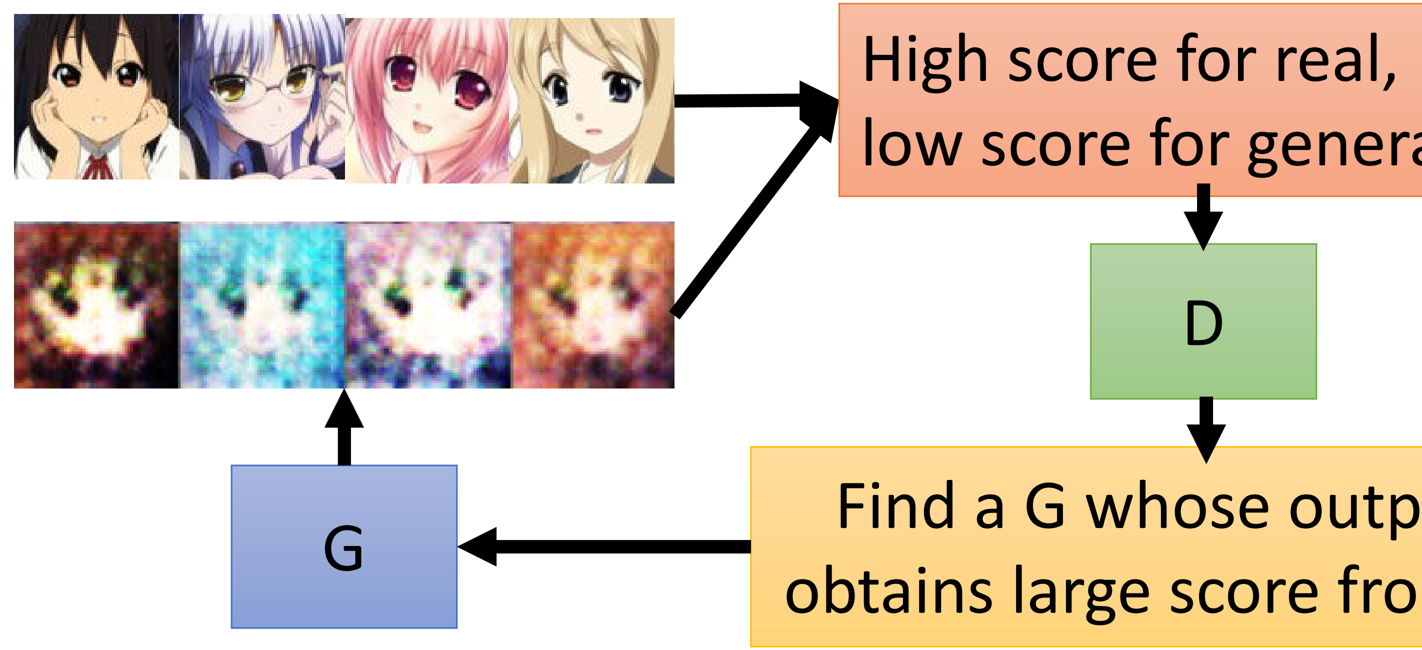


Actor
= Generator

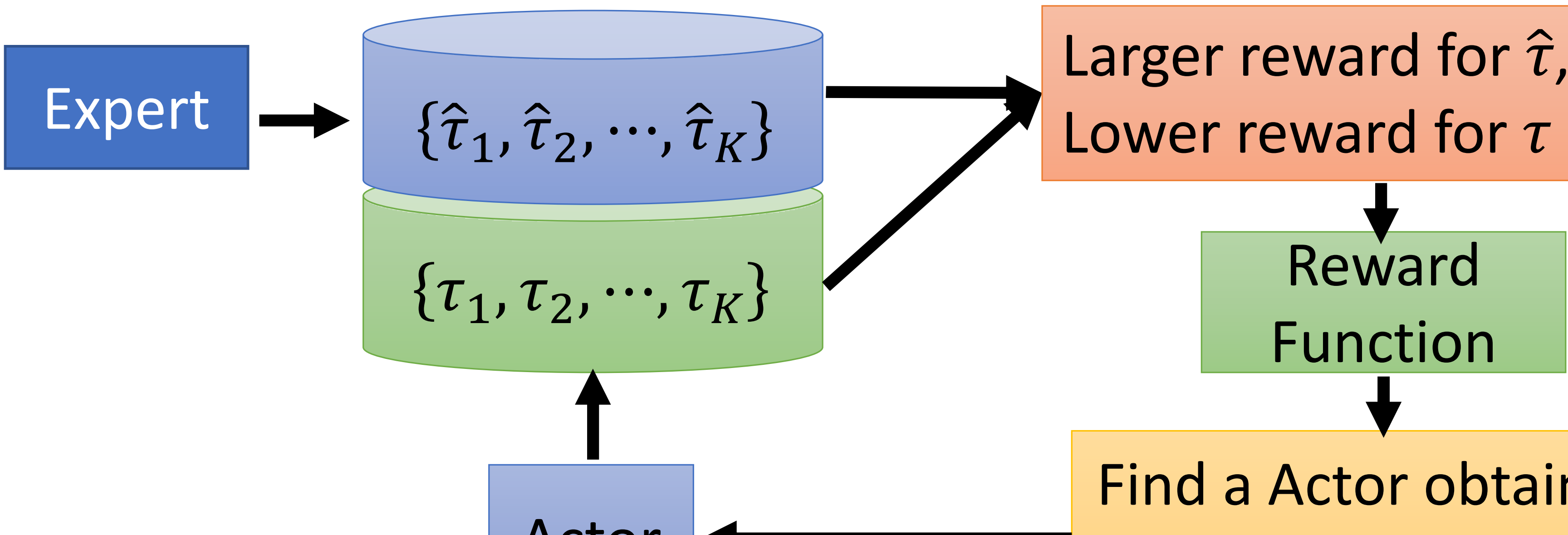
Reward function
= Discriminator

By Reinforcement lea

GAN


High score for real,
low score for genera
D
Find a G whose outp
obtains large score fro
G

IRL

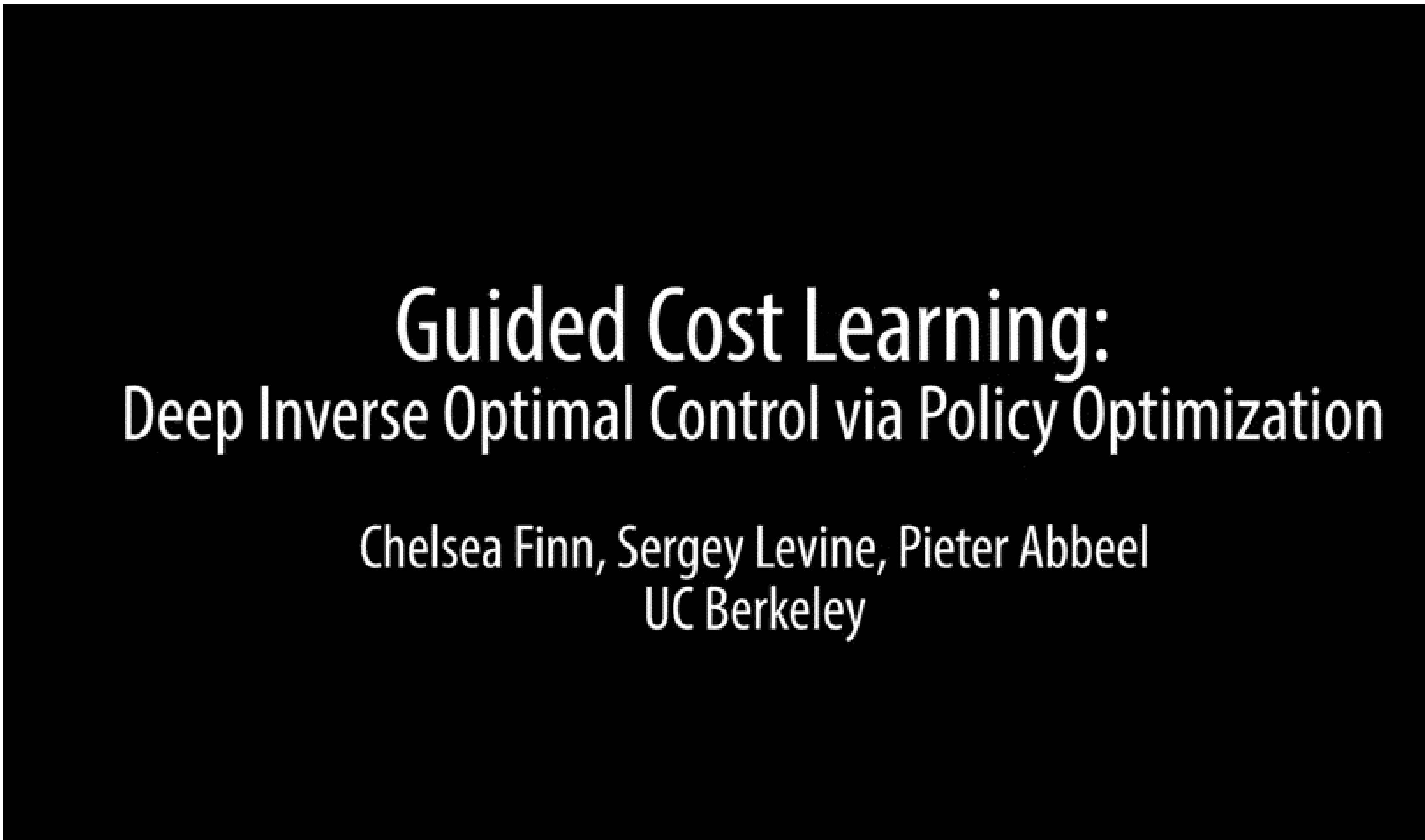

Expert
$\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_K\}$
$\{\tau_1, \tau_2, \cdots, \tau_K\}$
Larger reward for $\hat{\tau}$,
Lower reward for τ
Reward
Function
Find a Actor obtai
Actor

Robot

- How to teach robots? https://www.youtube.com/watch?v=DEGbtj

Robot

Chelsea Finn, Sergey Levine, Piet
Abbeel, Guided Cost Learning: Dee
Inverse Optimal Control via Polic
Optimization, ICML, 2016 http://rll.berkeley.e



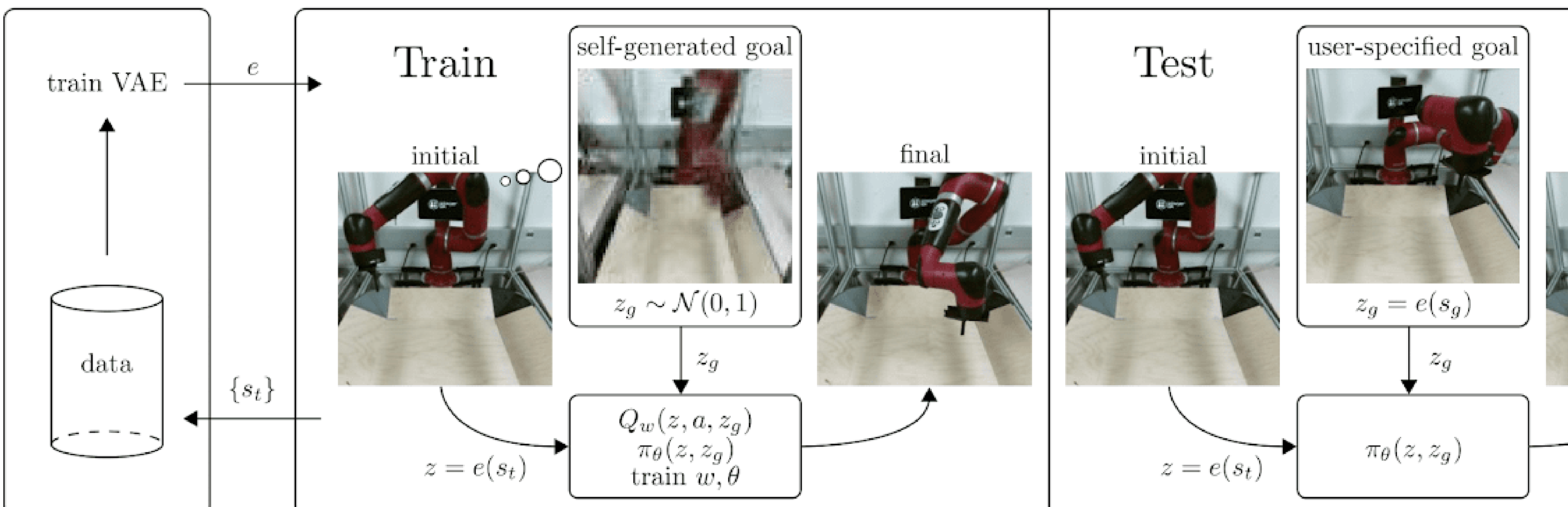
To Learn More …

Visual Reinforcement Learning with Imagined Goals, NIPS 201

https://arxiv.org/abs/180

Skew-Fit: State-Covering Self-Supervised Reinforcement Learn
ICML 2020

https://arxiv.org/abs/190



Concluding Remarks

- What is RL? (Three steps in ML)
- Policy Gradient
- Actor-Critic
- Sparse Reward
- No Reward: Learning from Demonstration