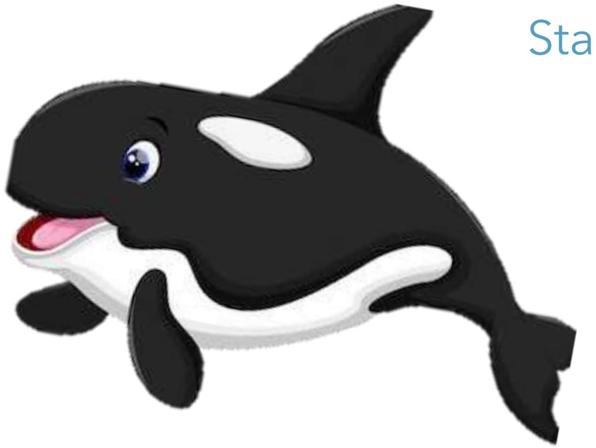
Can I Train a Robot Like my Dog?

Maximilian Du

Stanford '24 | CS Major (AI) | Creative Writing Minor (Prose)



The Training Game

The simple rules

1. There are two roles: **trainer** and **trainee**
2. Trainee is motivated by a **reward** (clicker or whistle)
3. Trainer wants to get trainee to do a predefined task

Volunteers?

Walk close to trainer

Spin in circles

Jump up and down

Stand on table

Ch 0

Should I care about training?

Max Du (he/him)

Class of 2024

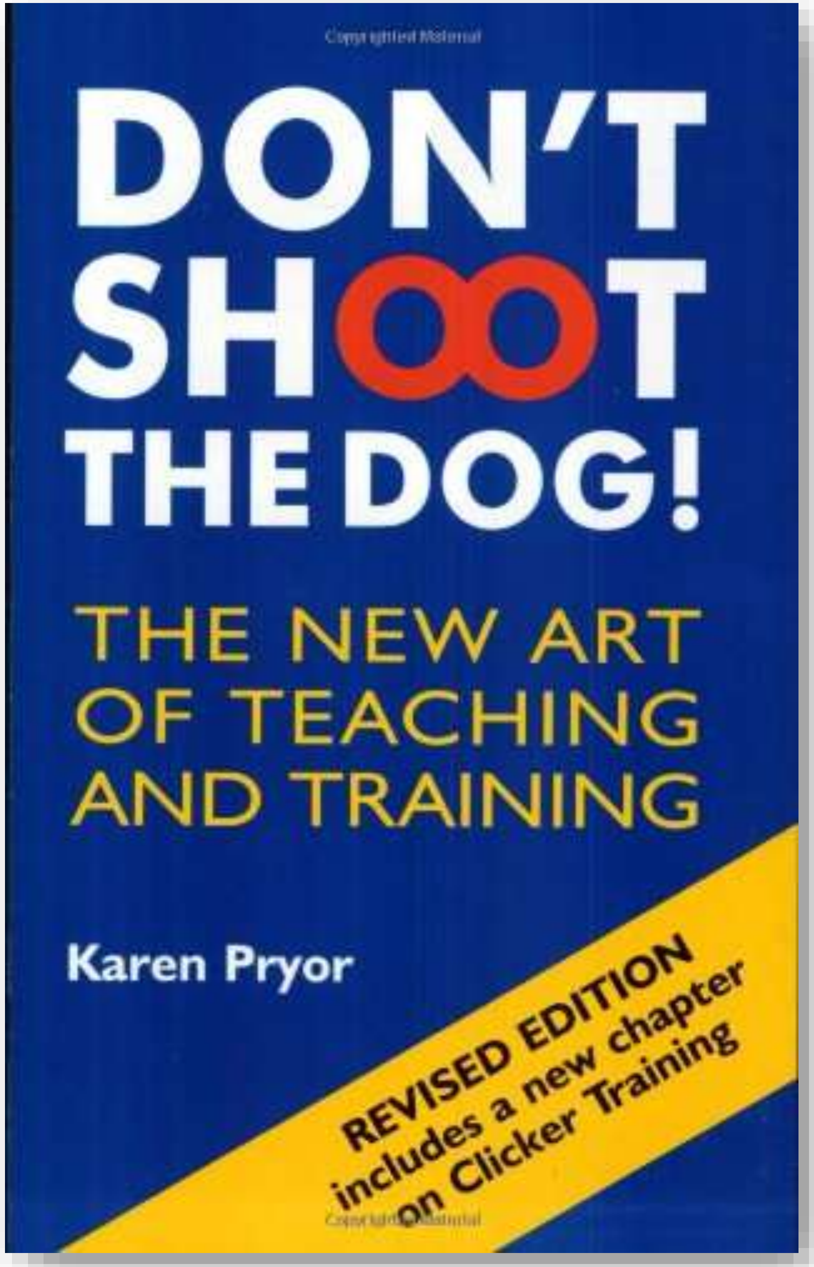
Computer Science BS Candidate
Creative Writing Minor Candidate
Psychology Minor Candidate


Lyndsey Schemm

Animal training developed from the Behavioral Science foundations of **B.F. Skinner**

His main idea: every living creature learns everything from **rewards**

The **first dolphin trainer**, Karen Pryor, brought the theory into a set of **best practices** for marine mammals



Out of BF Skinner's ideas also came **Reinforcement Learning**

Reinforcement learning is how we can get **computers** to learn from **rewards**

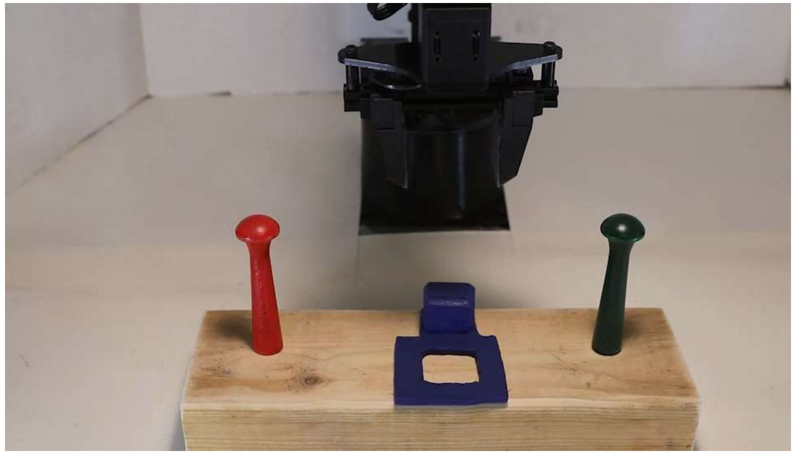
Main Research Interest

Getting robots to develop **broadly intelligent** behavior through **learning** and **interaction.**

Can robots use **audio** and **vision**?

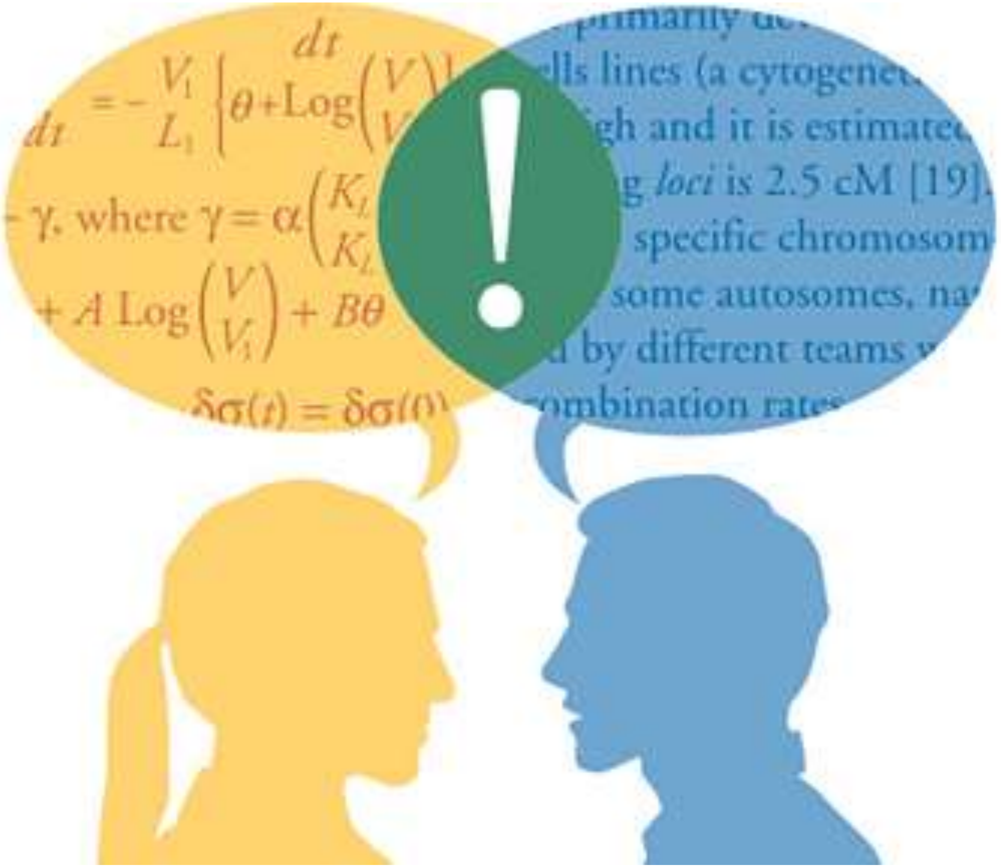
Can robots learn from **mixed-quality** data?

Can robots try **different strategies**?



Whale trainers don't talk to robot researchers.

Can we fix this?



The *Deadly Sins* of Learning

1. Context Shift
2. Superstition
3. Under-exploration

The Malfunctioning Dolphin

Ch 1

Context Shift

The *Deadly Sins* of Learning

1. **Context Shift**
2. Superstition
3. Under-exploration

Much of machine learning is pushing an **approximation** close to a **target**

$$\overbrace{\arg\min_{\theta}}^{\text{Pushing things together}} D(f_\theta, y)$$

Things you tune

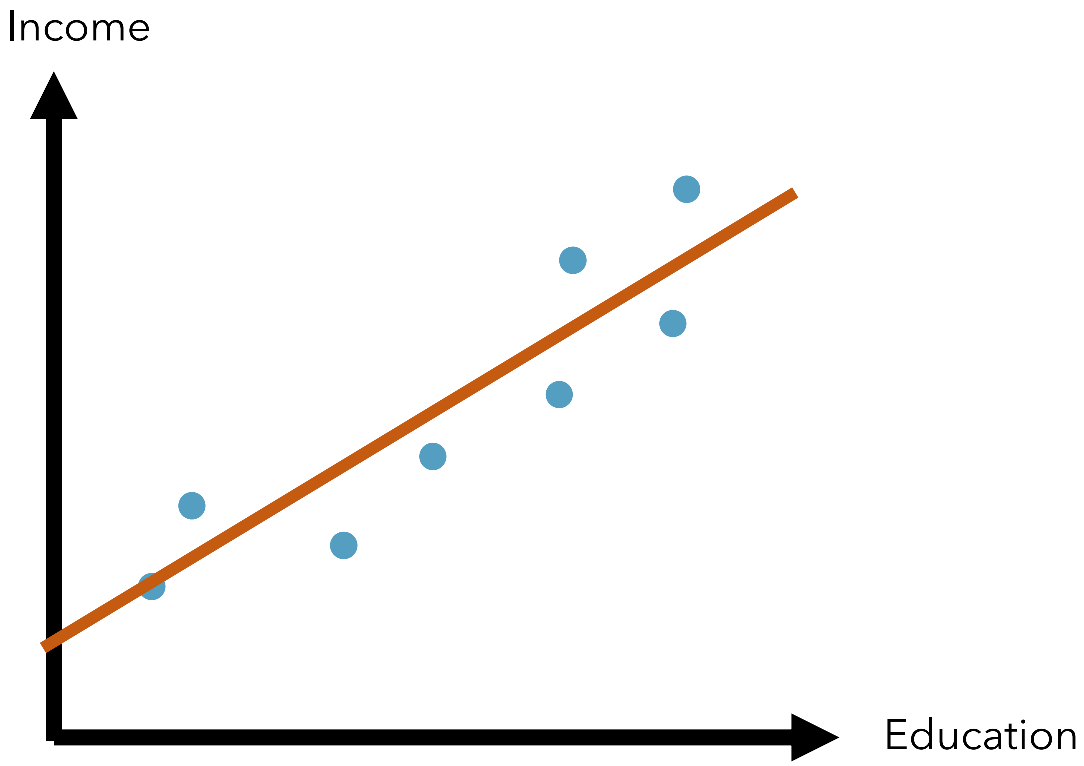
Target

Case study: linear regression

$$\underbrace{\arg\min_{\theta}}^{\text{Pushing things together}} D(f_\theta, y)$$

Things you tune

Target



The **model** is

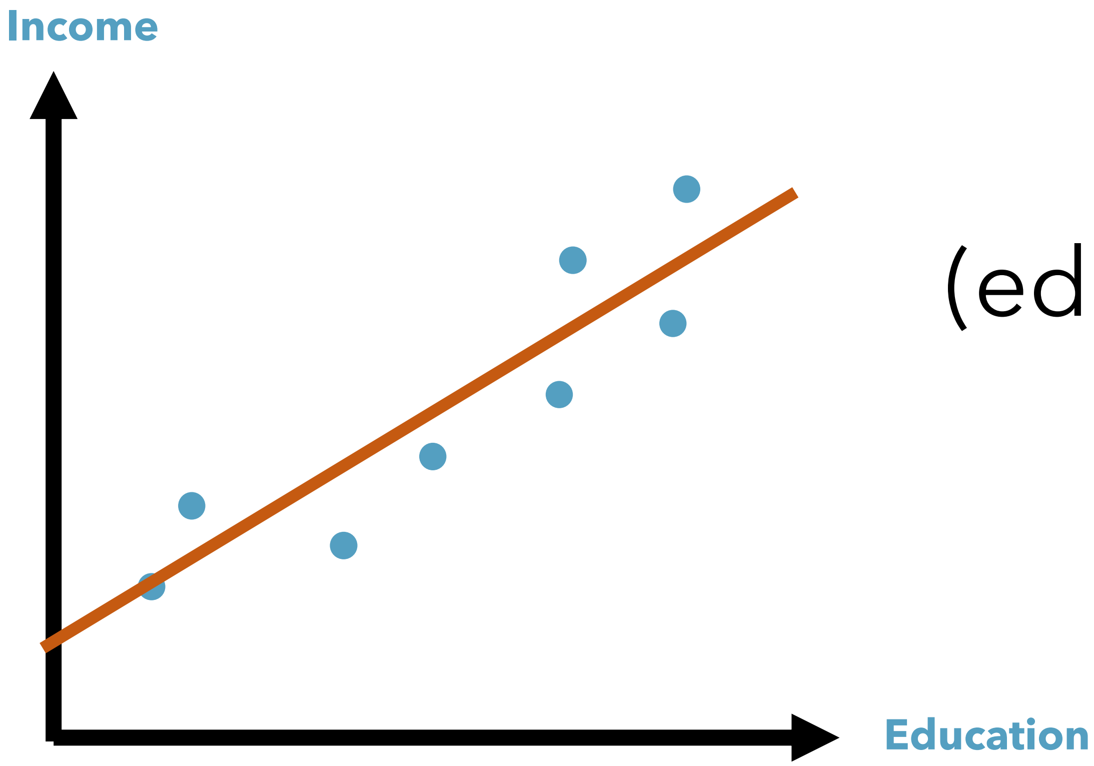
$$f_\theta(x) = \theta_1 x + \theta_2$$

Case study: linear regression

$$\overbrace{\arg\min_{\theta}}^{\text{Pushing things together}} D(f_\theta, y)$$

Things you tune

Target



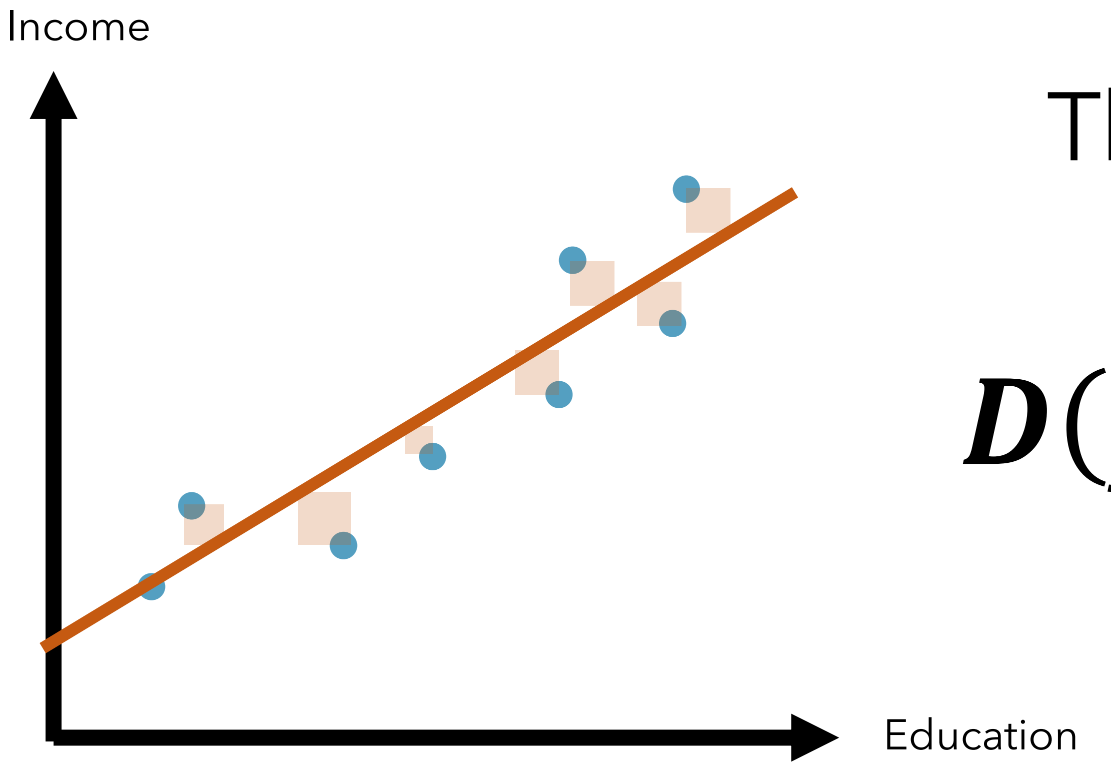
We are given **inputs** x (education) and want to predict **outputs** y (income)

Case study: linear regression

$$\overbrace{\arg\min_{\theta}}^{\text{Pushing things together}} D(f_\theta, y)$$

Things you tune

Target



The D is **squared distance**

$$\boldsymbol{D(f_\theta, y) = (f_\theta(x) - y)^2}$$

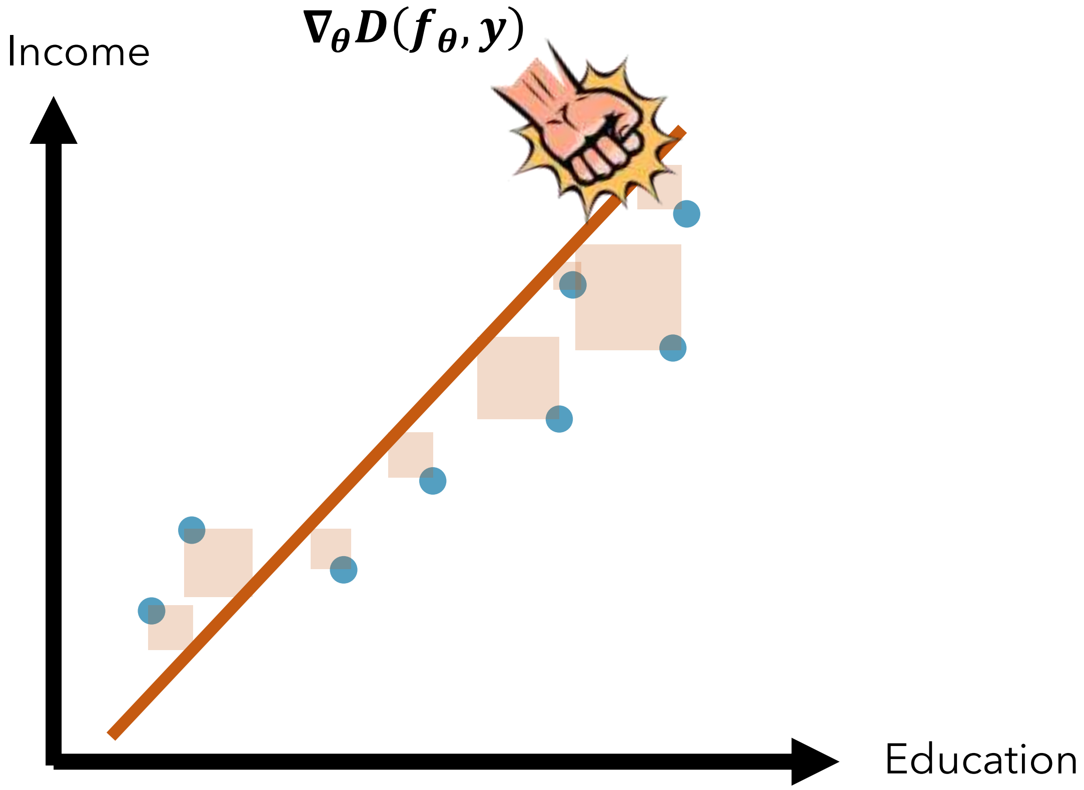
Case study: linear regression

Pushing things together

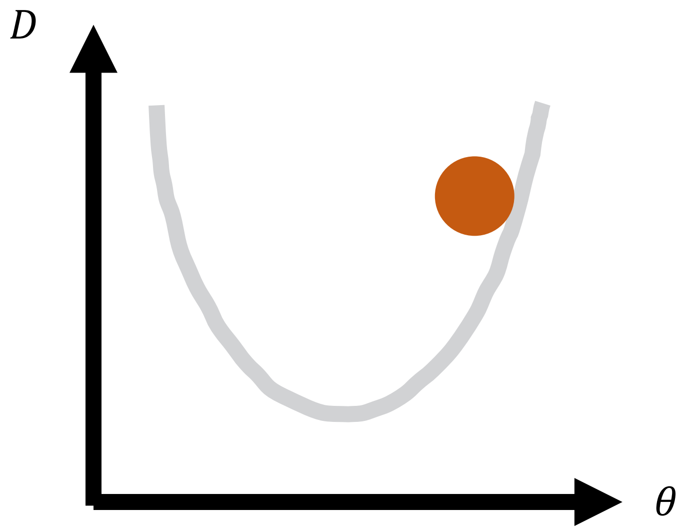
$$\overbrace{\arg\min_{\theta}}^{\text{Pushing things together}} D(f_\theta, y)$$

Things you tune

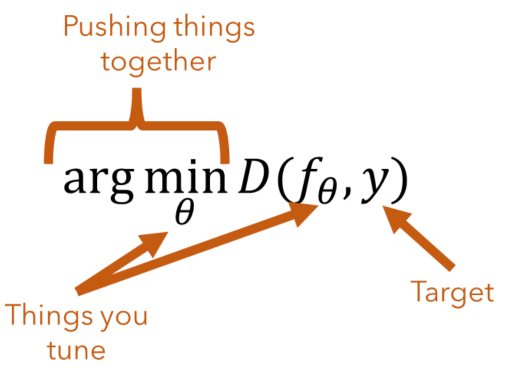
Target



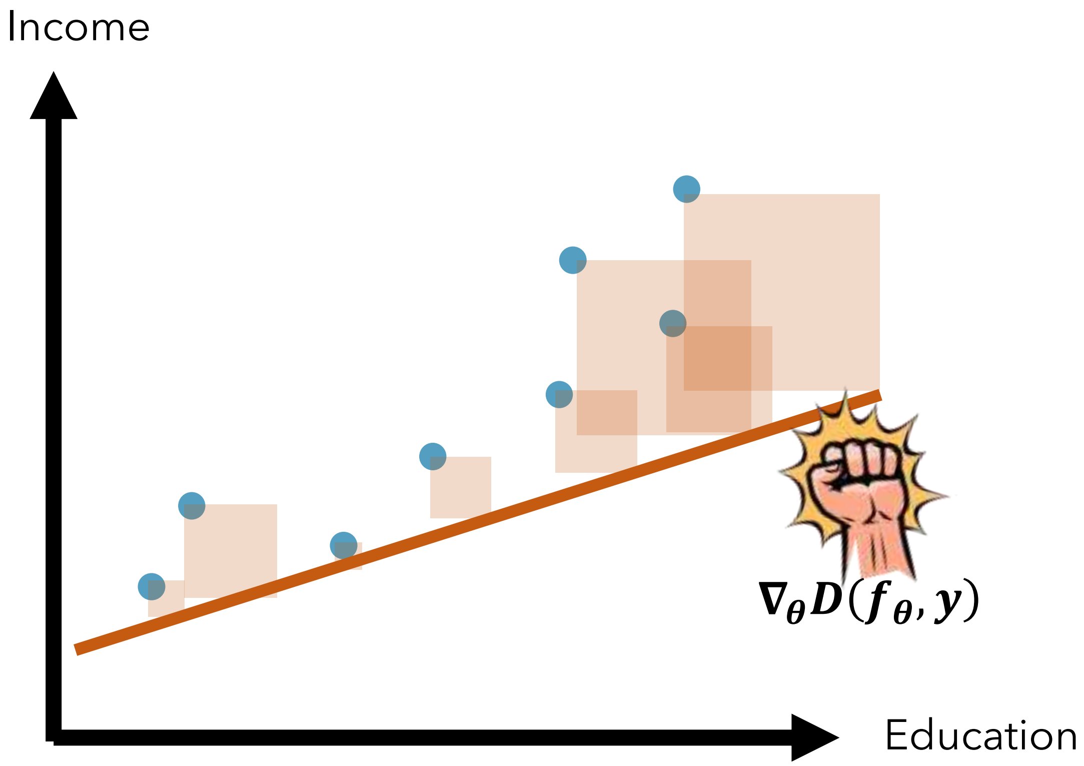
There is a natural place of **best fit**

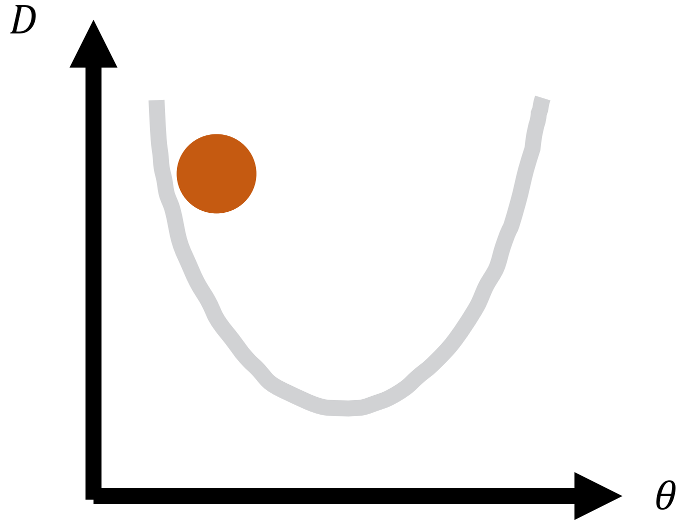


Case study: linear regression



There is a natural place of **best fit**





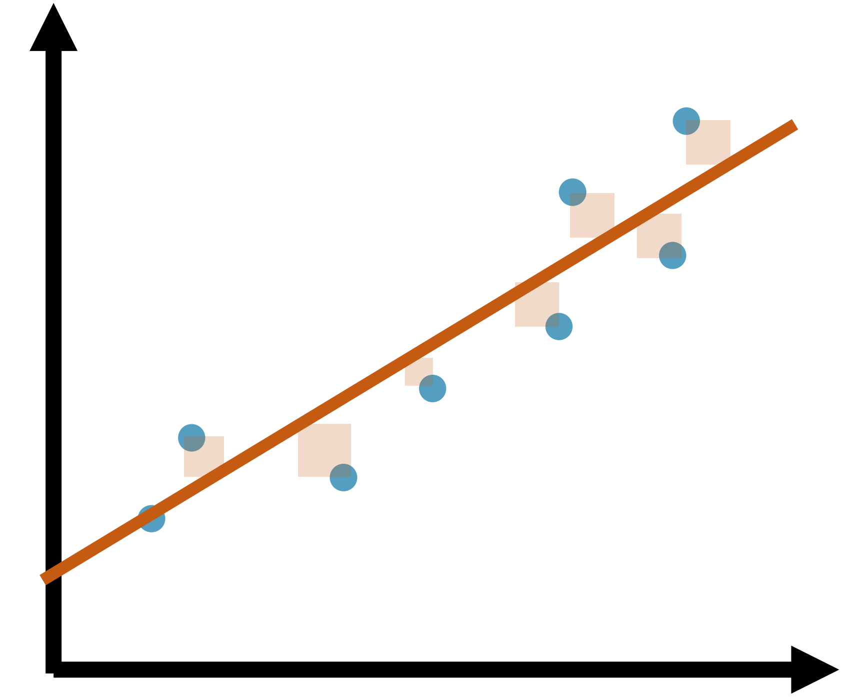
Case study: linear regression

Pushing things together

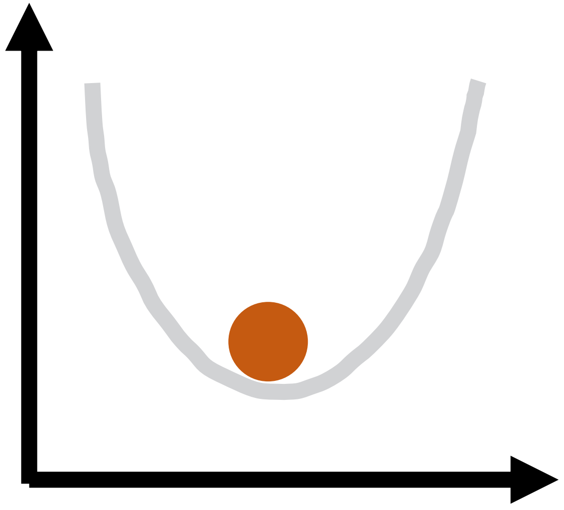
$$\underbrace{\arg\min_\theta}_{\text{Pushing things together}} D(f_\theta, y)$$

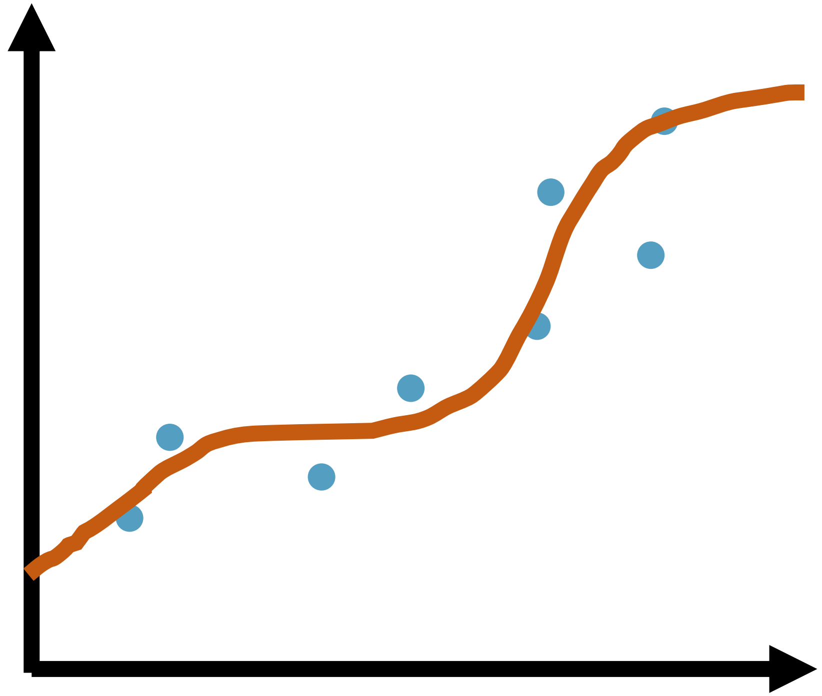
Things you tune

Target

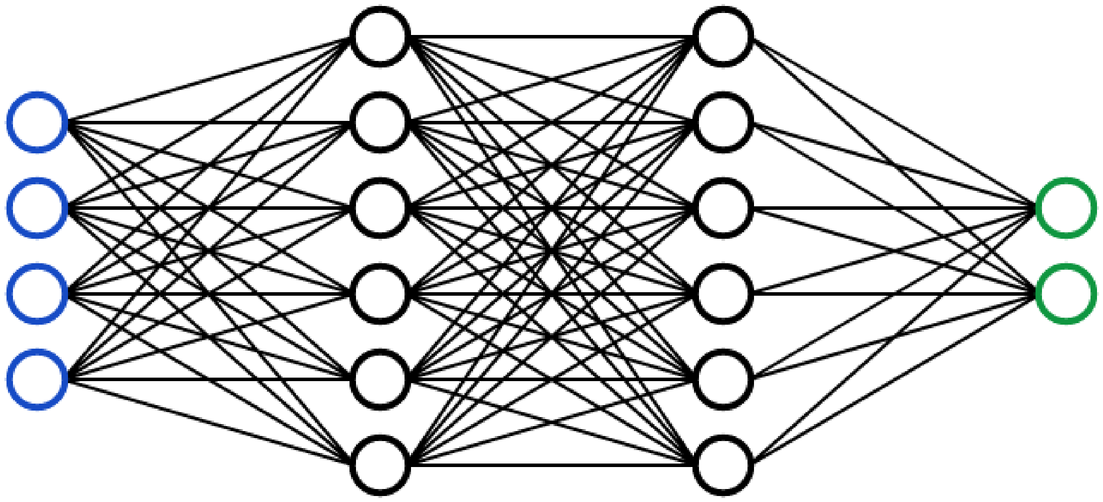


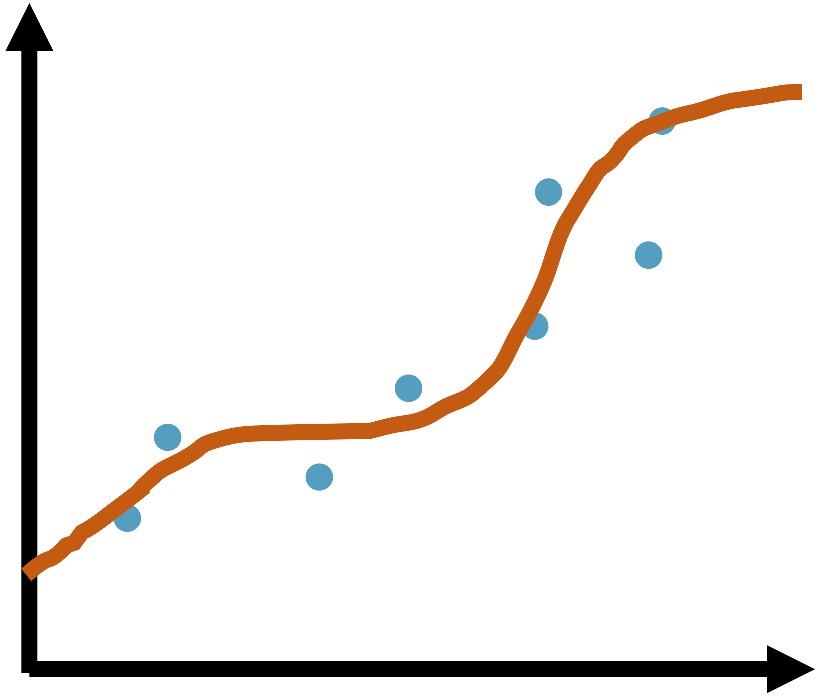
There exists one solution where the D is as small as can be





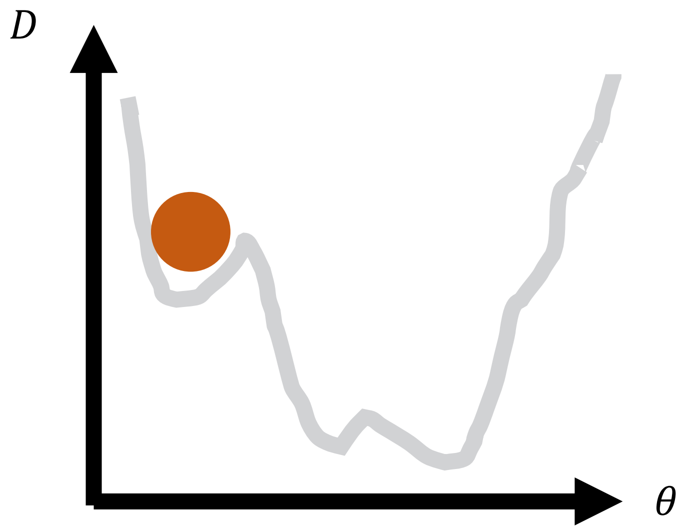
We can use θ to represent some parameters of a "neural network"

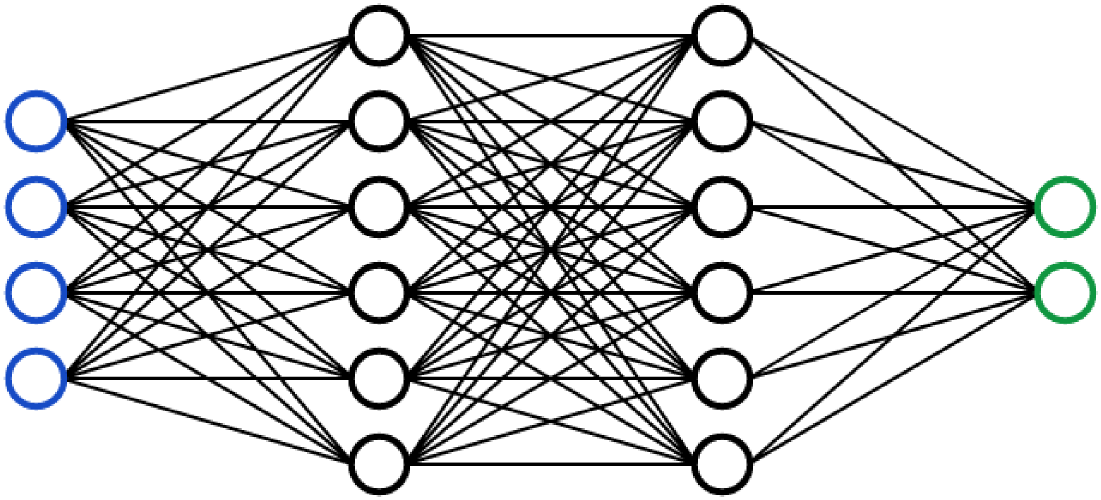




We can use θ to represent some parameters of a "neural network"

This is hard to optimize!





More complicated f_θ means…

+ More expressivity

- More difficulty in getting right

Questions so far?

We can complicate the system further by changing x, y

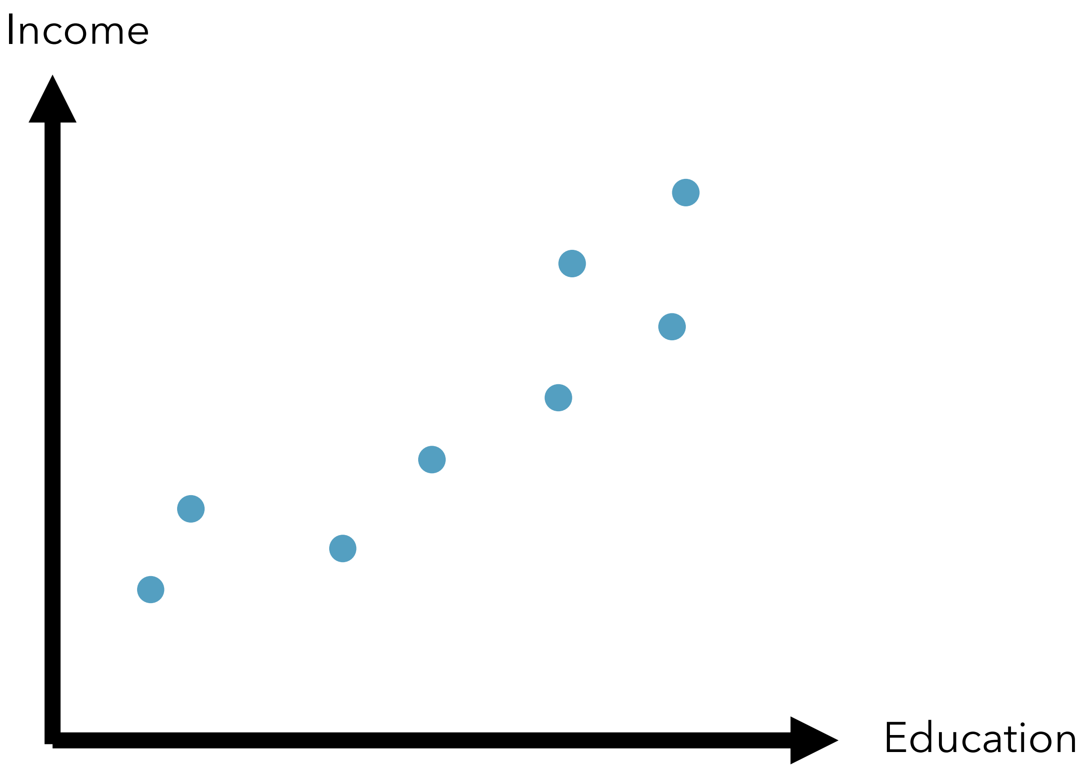
$$\overbrace{\arg\min_{\theta}}^{\text{Pushing things together}} D(f_{\theta}, y)$$

Things you tune

Target

Input x — Output y

Number — Number

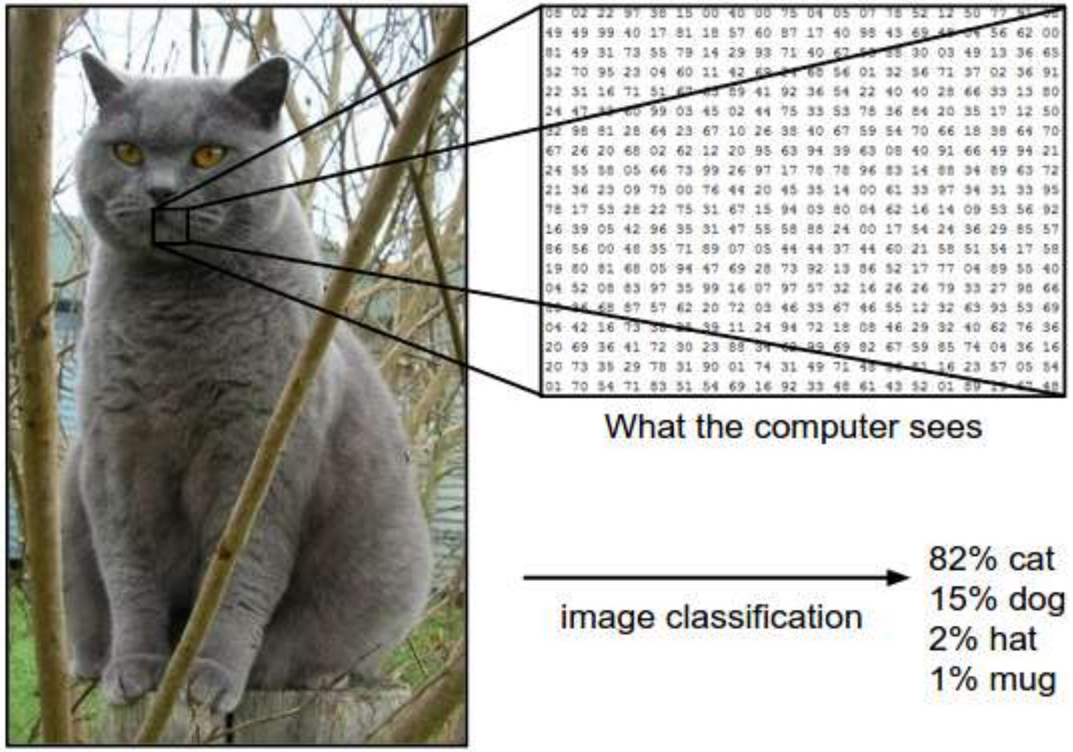


Input x

Output y

Images

Vector (prediction)

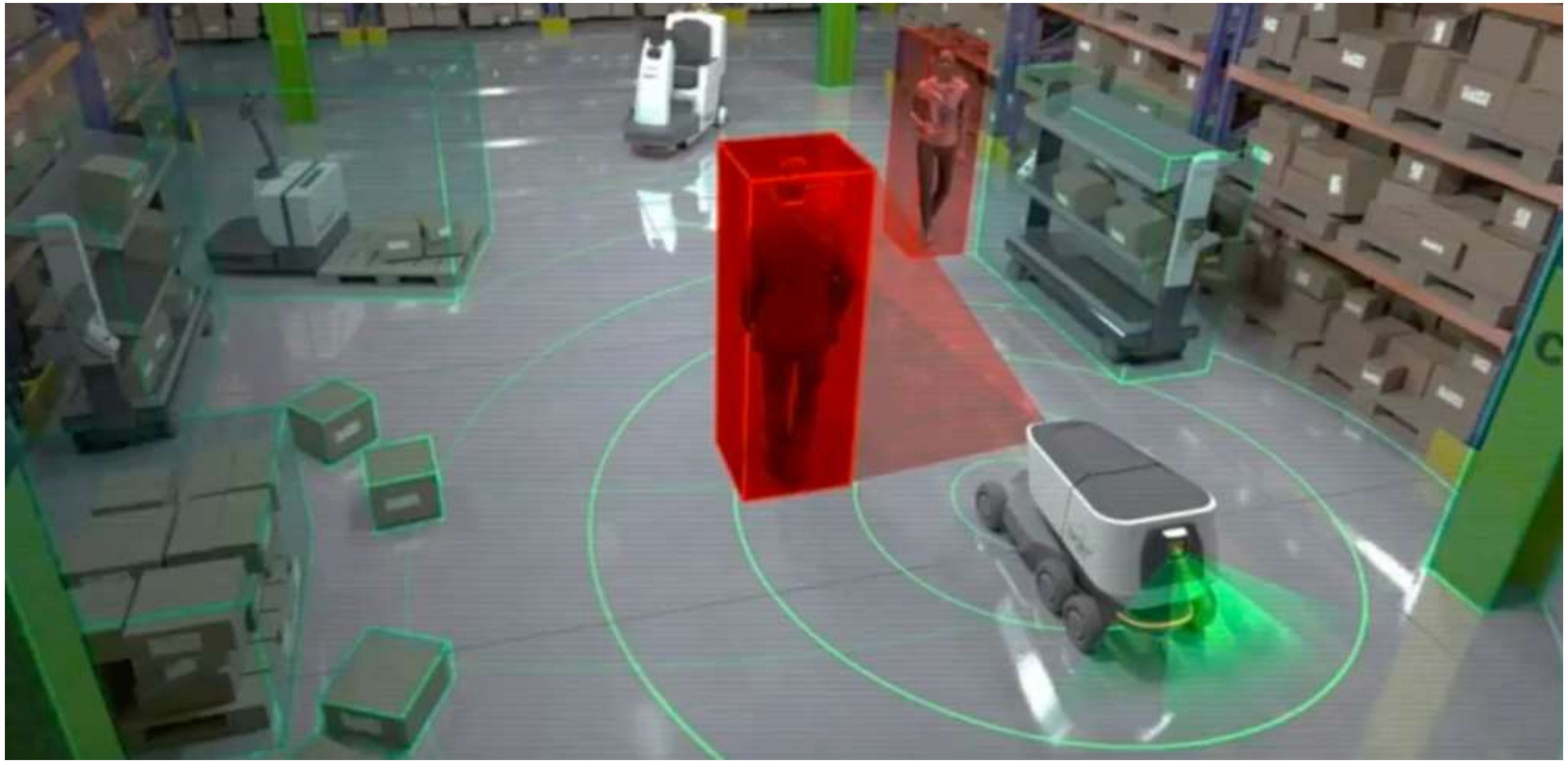


Input x

Observation (image)

Output y

Action (vector)

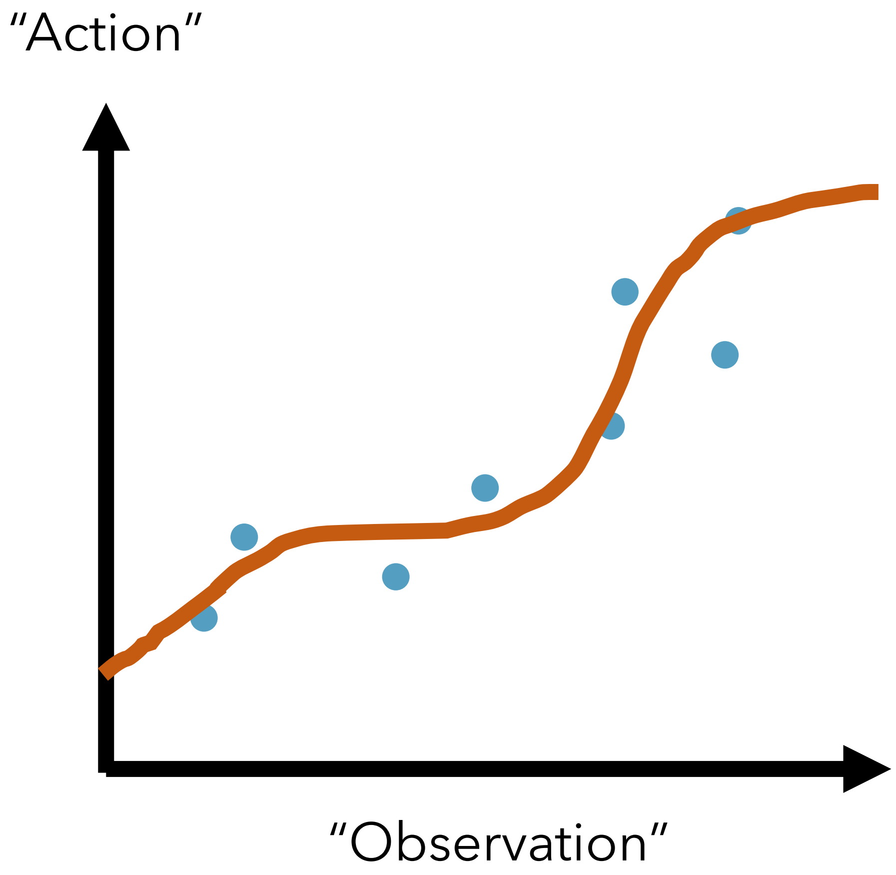


Behavior cloning is the mapping of observations to actions

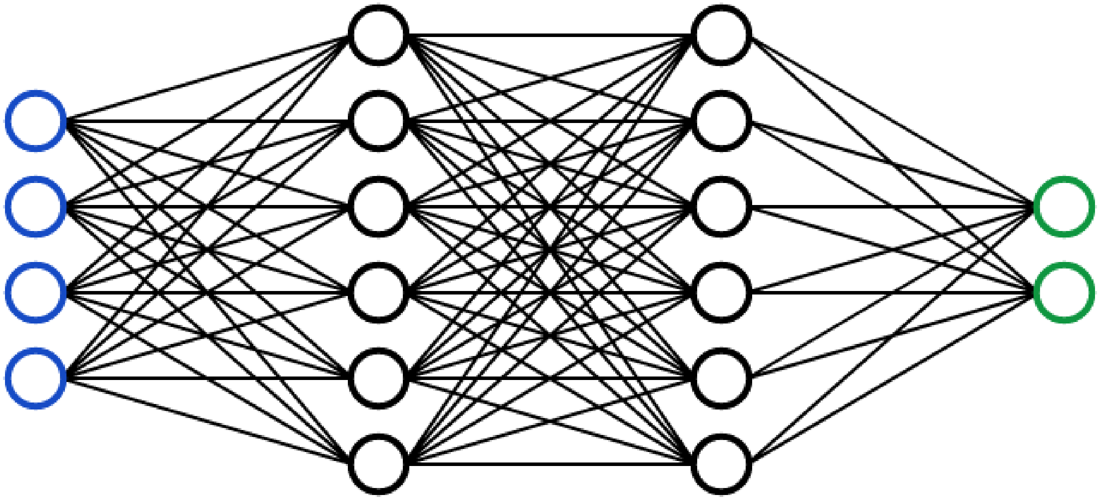
x Observation

`[0.1, -0.2, 1]` *"close door"*

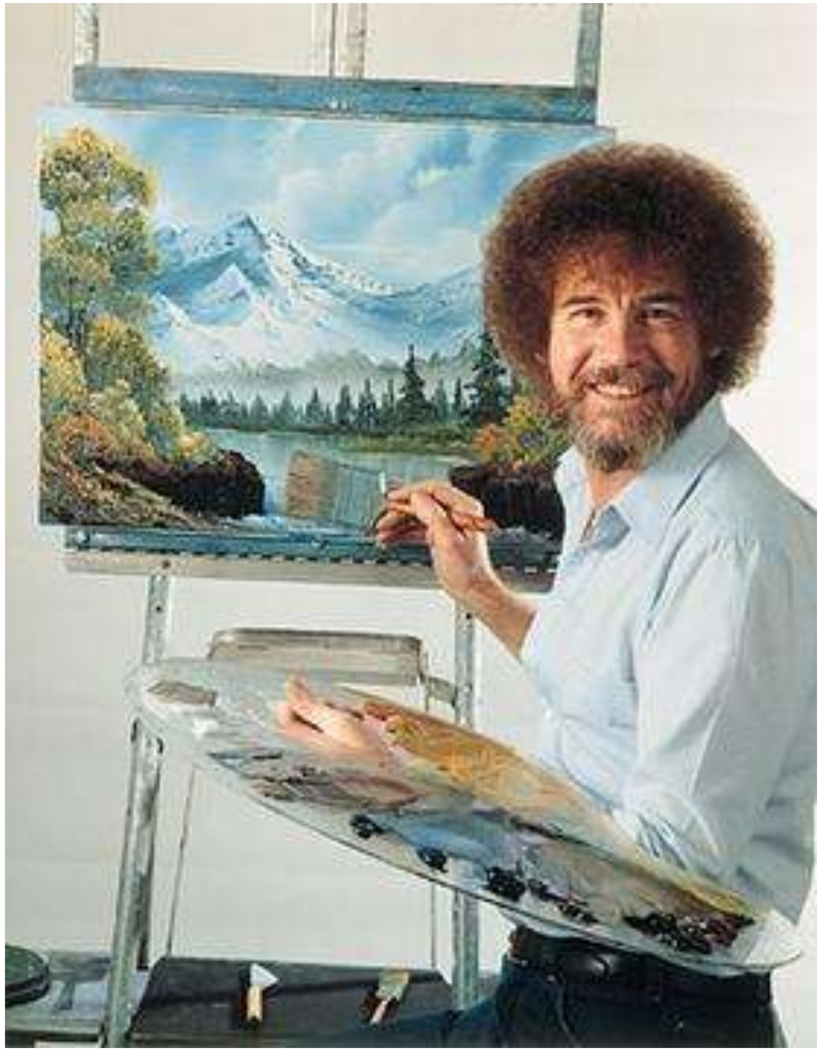
y Action



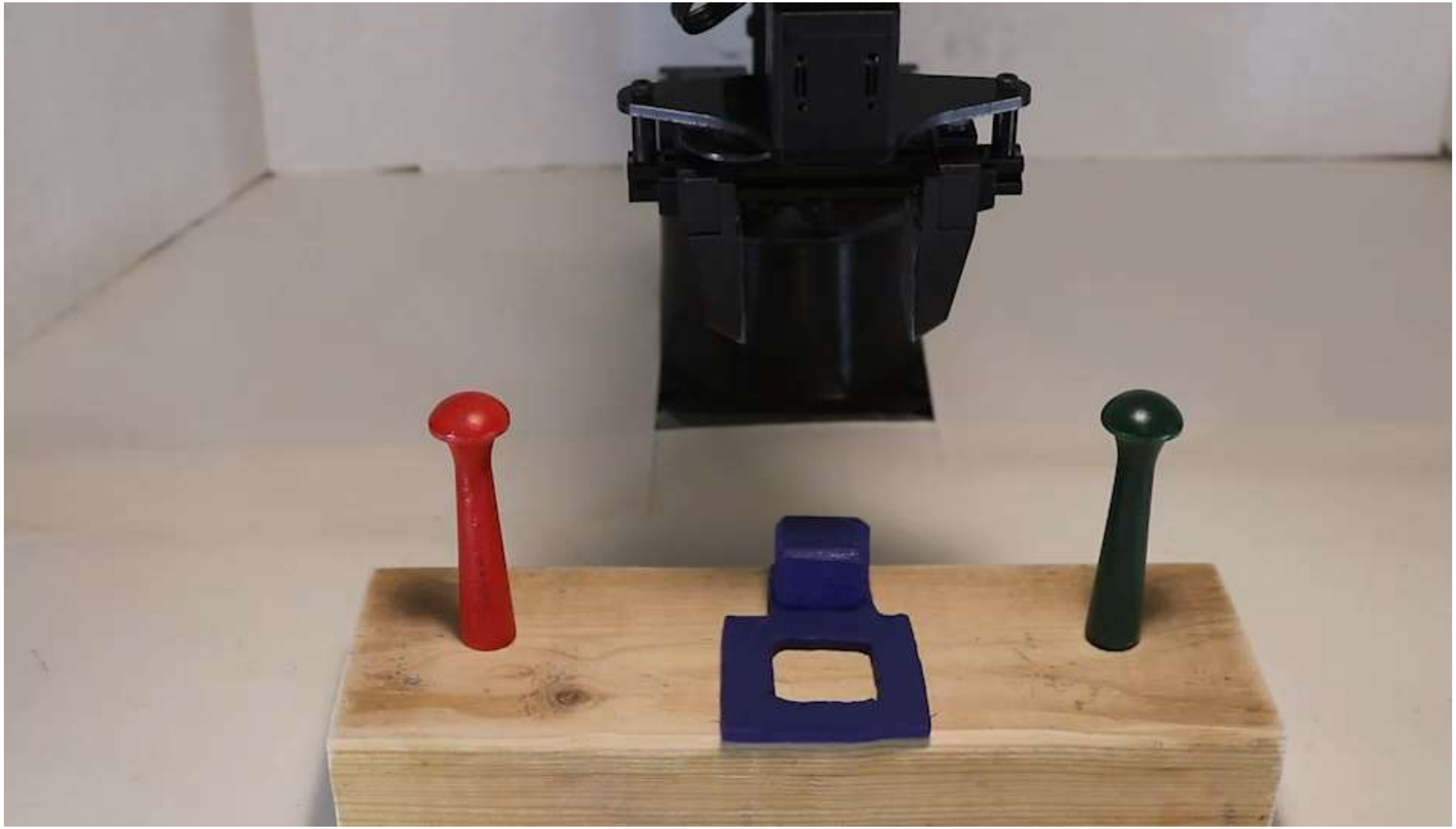
We can use **similar techniques** to regression: calculus and a **complicated model (neural network)**



Behavior cloning (imitation) is the **easiest way** of learning **complicated behaviors**



This robot learned through Behavior Cloning



Wikie the talking orca

Hmmmm

Dank_Smirk 3 years ago

Scientists: "Speak."
Wikie: "HEWW🅾️! *OwO"*
Scientists: "By god, what have we done?

411 Reply

12 replies

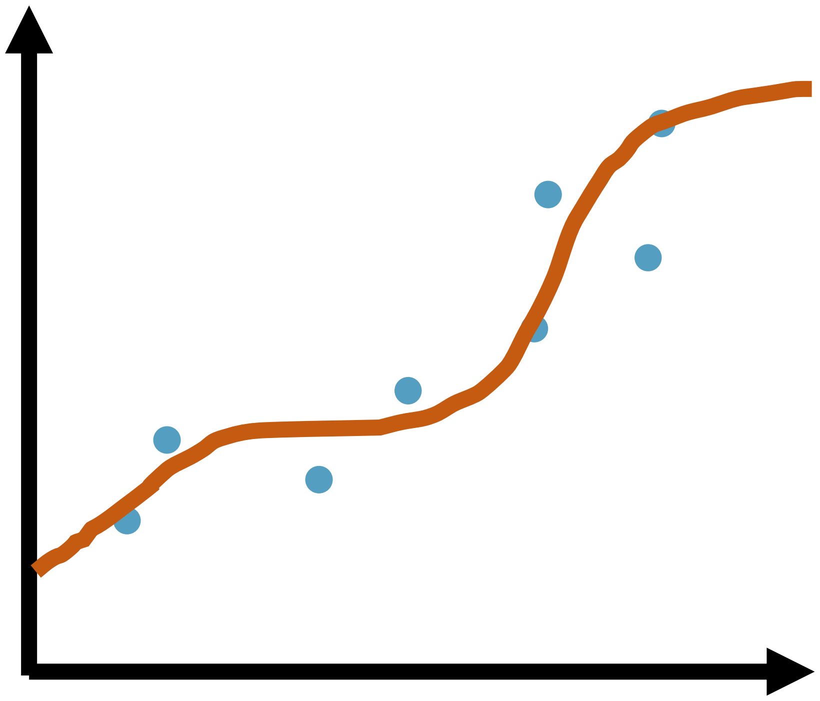
James D 4 years ago

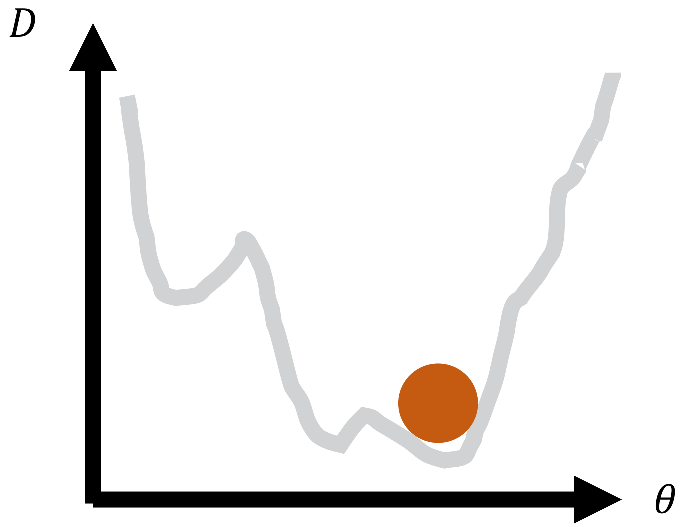
Orca- RAWWWEEEKKKERRRR
Human- awwwww he said my name!

1.9K Reply

24 replies

Hmmm



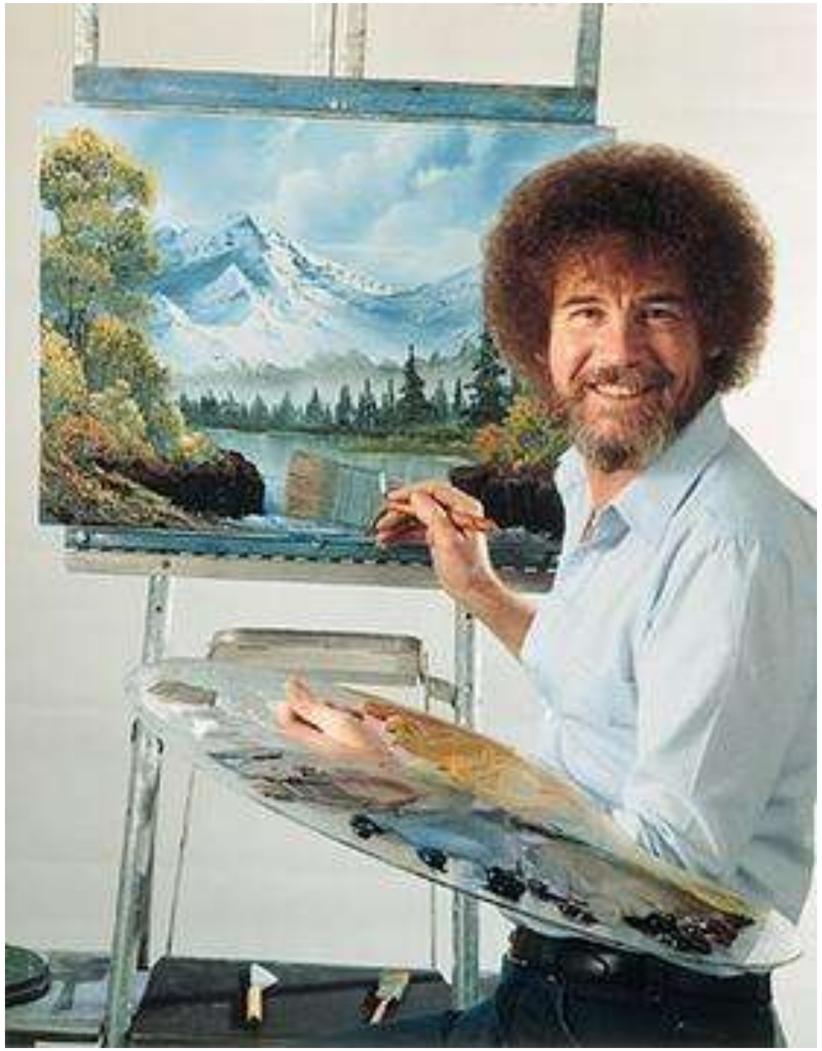


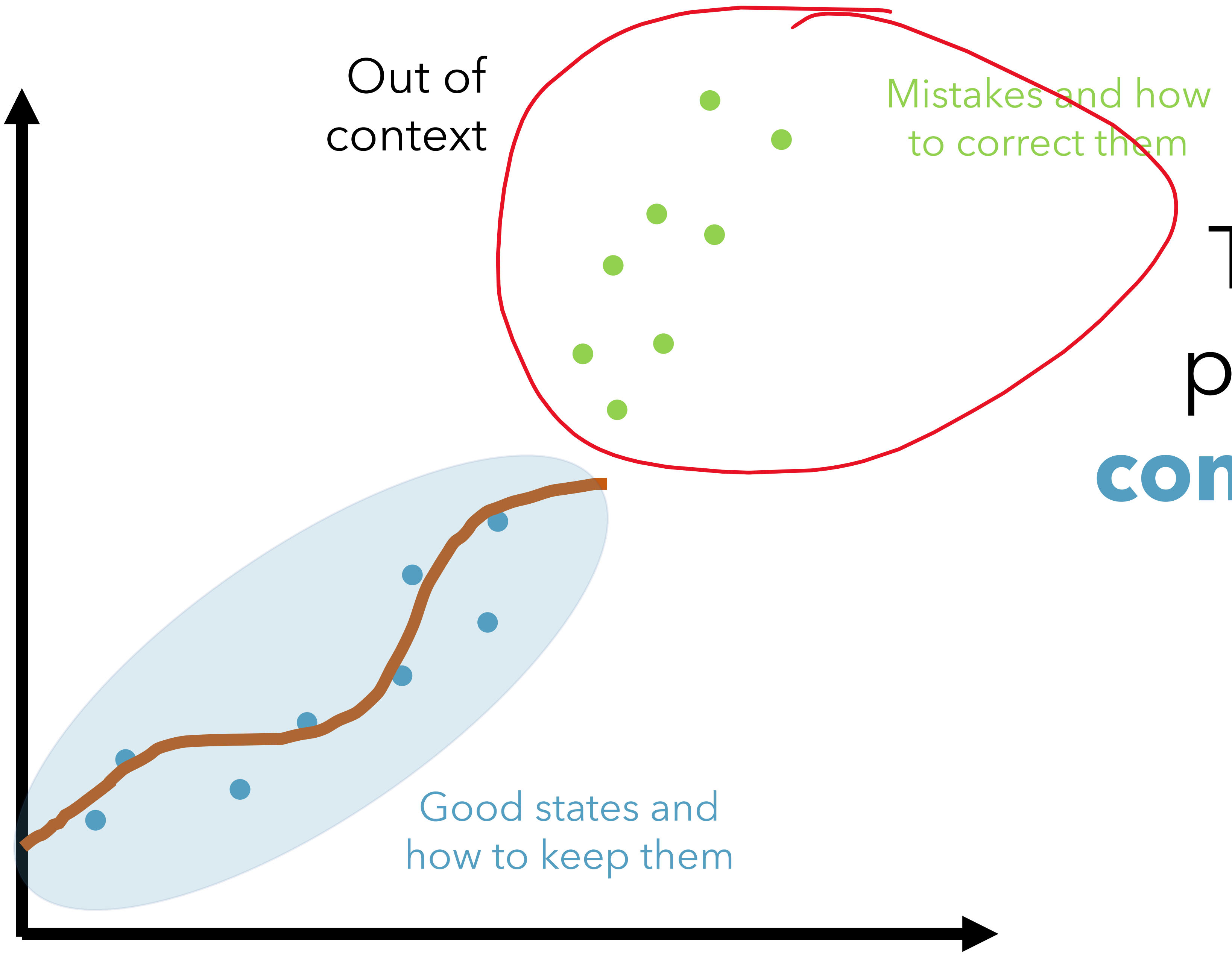
This problem can happen with the **cleanest** expert data and the **best fit** model

Pair up and discuss: What could be wrong wih behavior cloning?

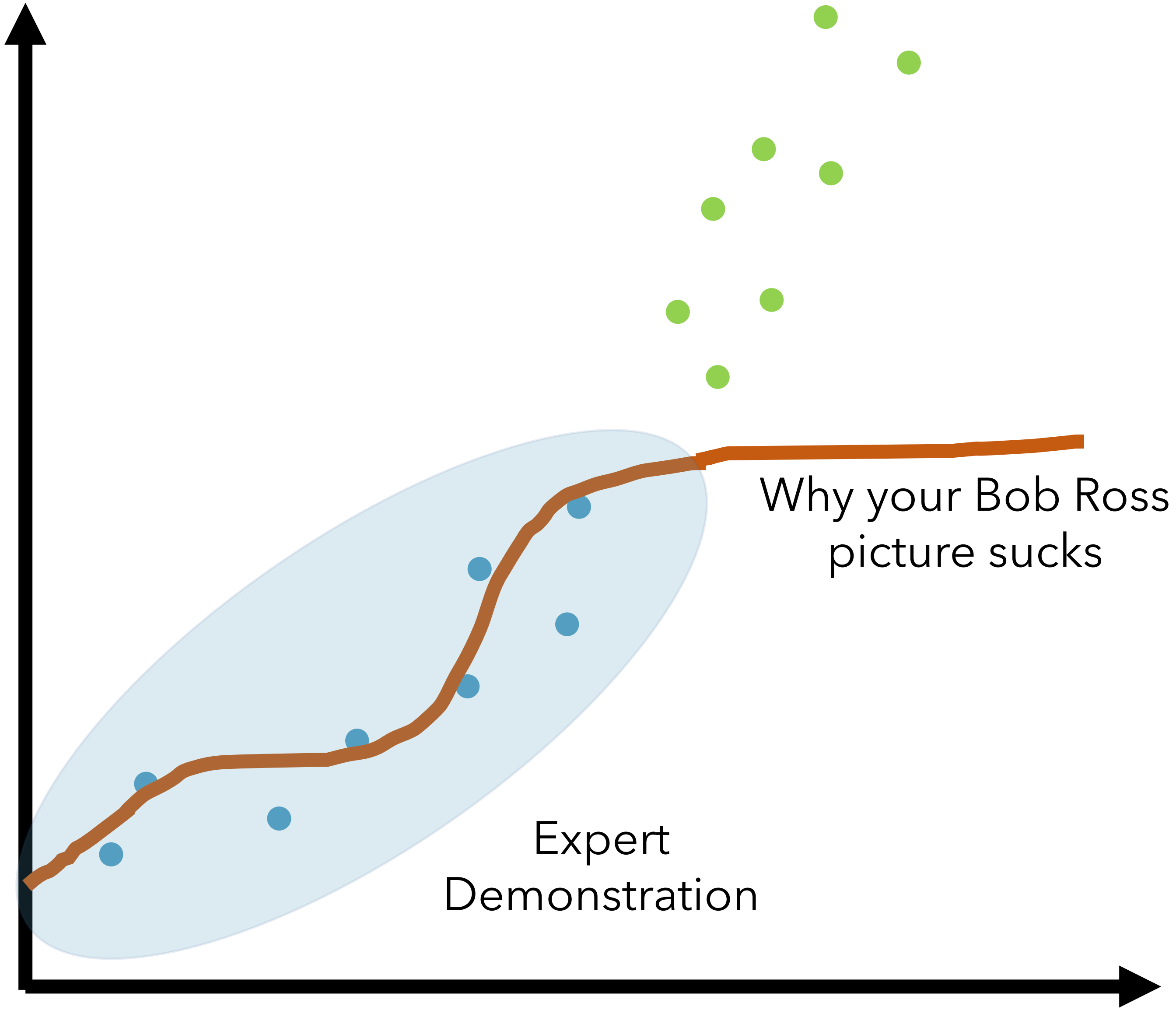
1. Collect expert data
2. Mimic expert data

Experts demonstrate what is **good**, but seldom how to **recover** from what is **bad**



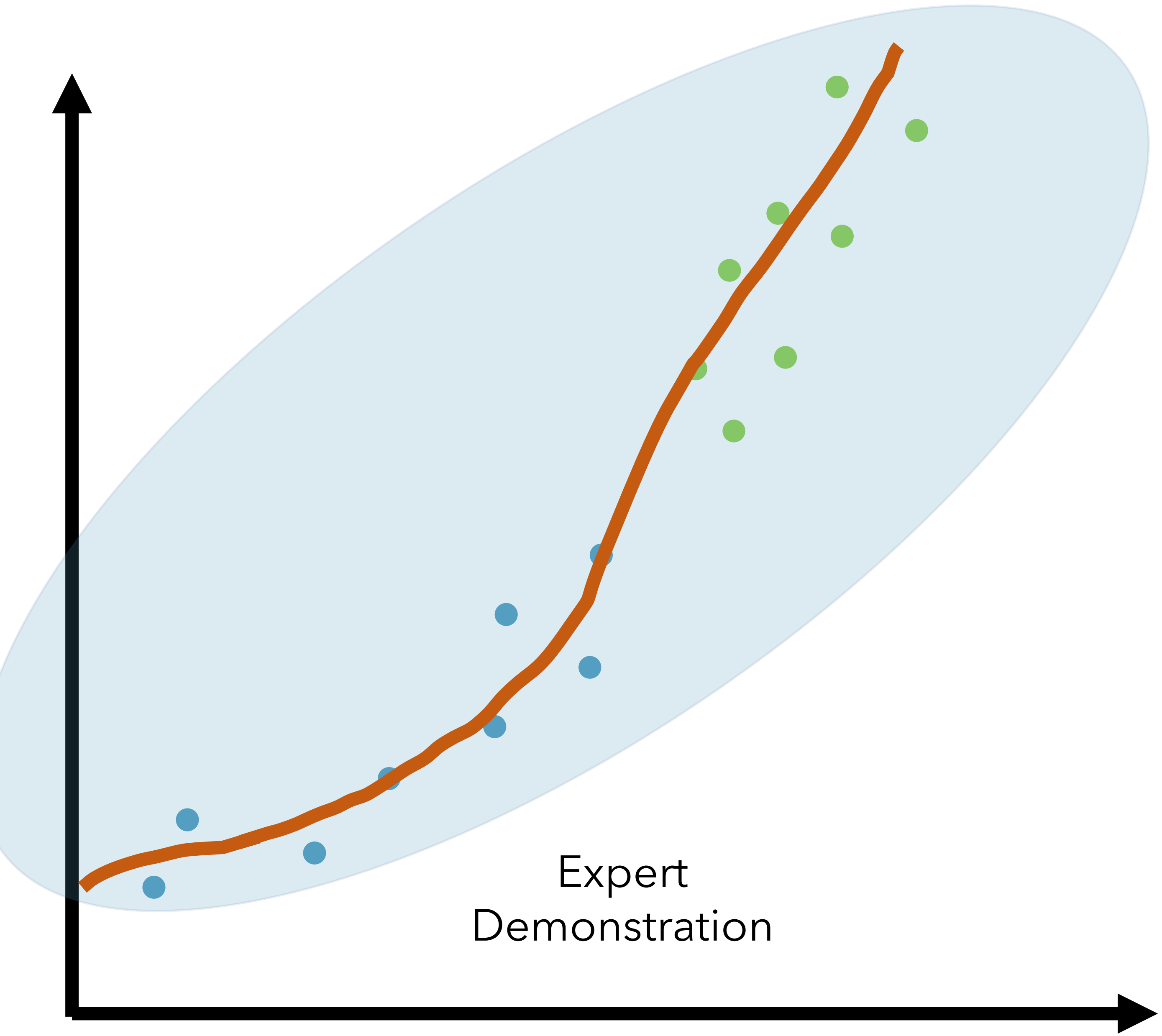


This is the problem of **context shift** in machine learning



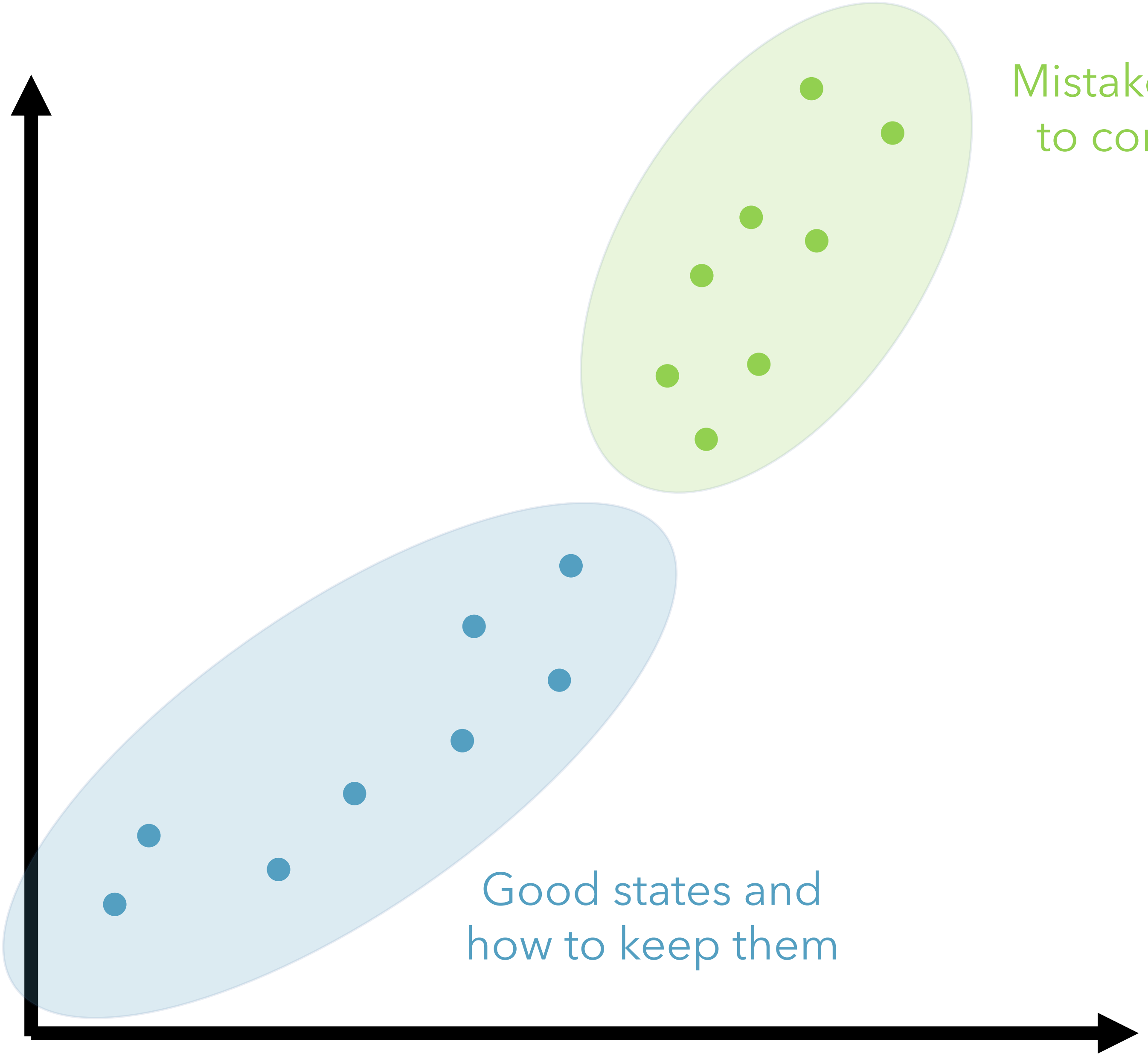
The sin of Context Shift

Outside a learned **context**, there is no guarantees of meaningful function



Sin Resolution

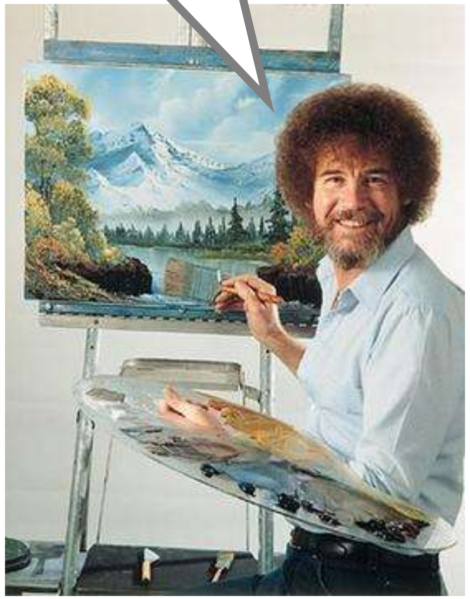
We can **expand** the context (expert demonstrations)



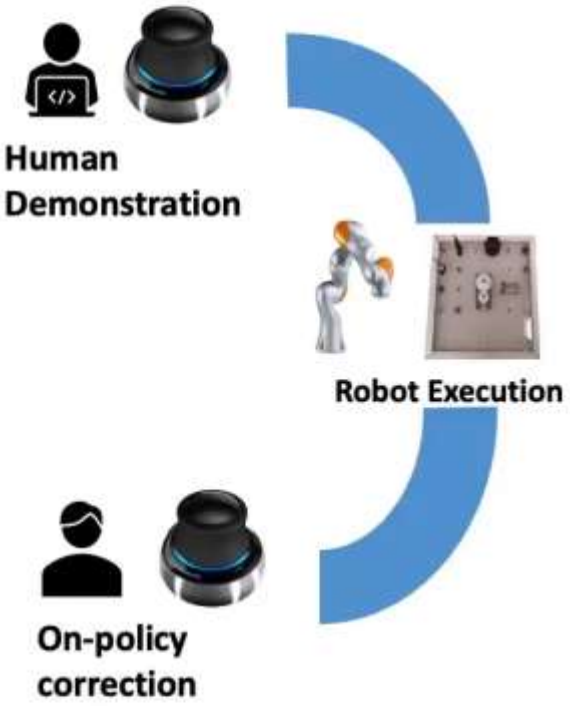
It may be hard to **proactively** find mistakes.

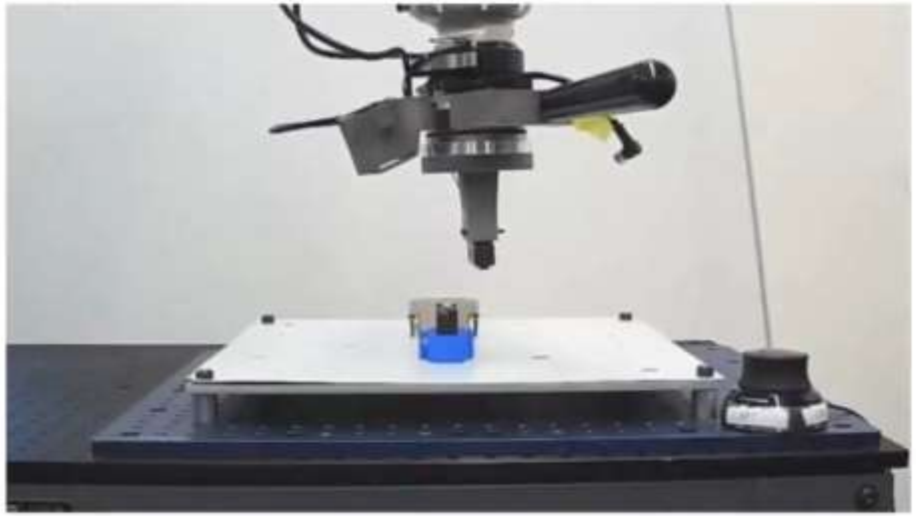
Any ideas?

Let the **trained agent** make the mistakes, and then **correct them!**

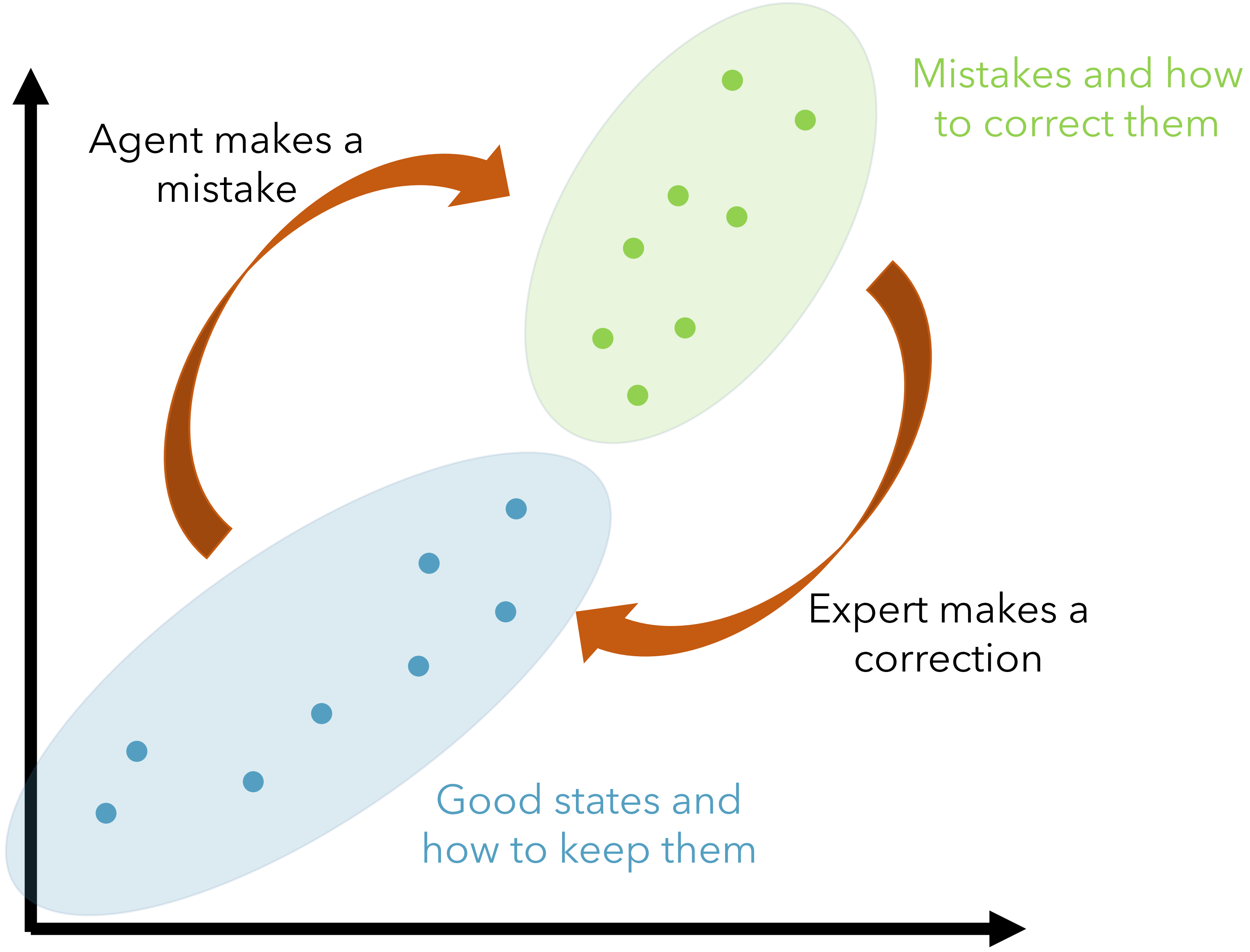


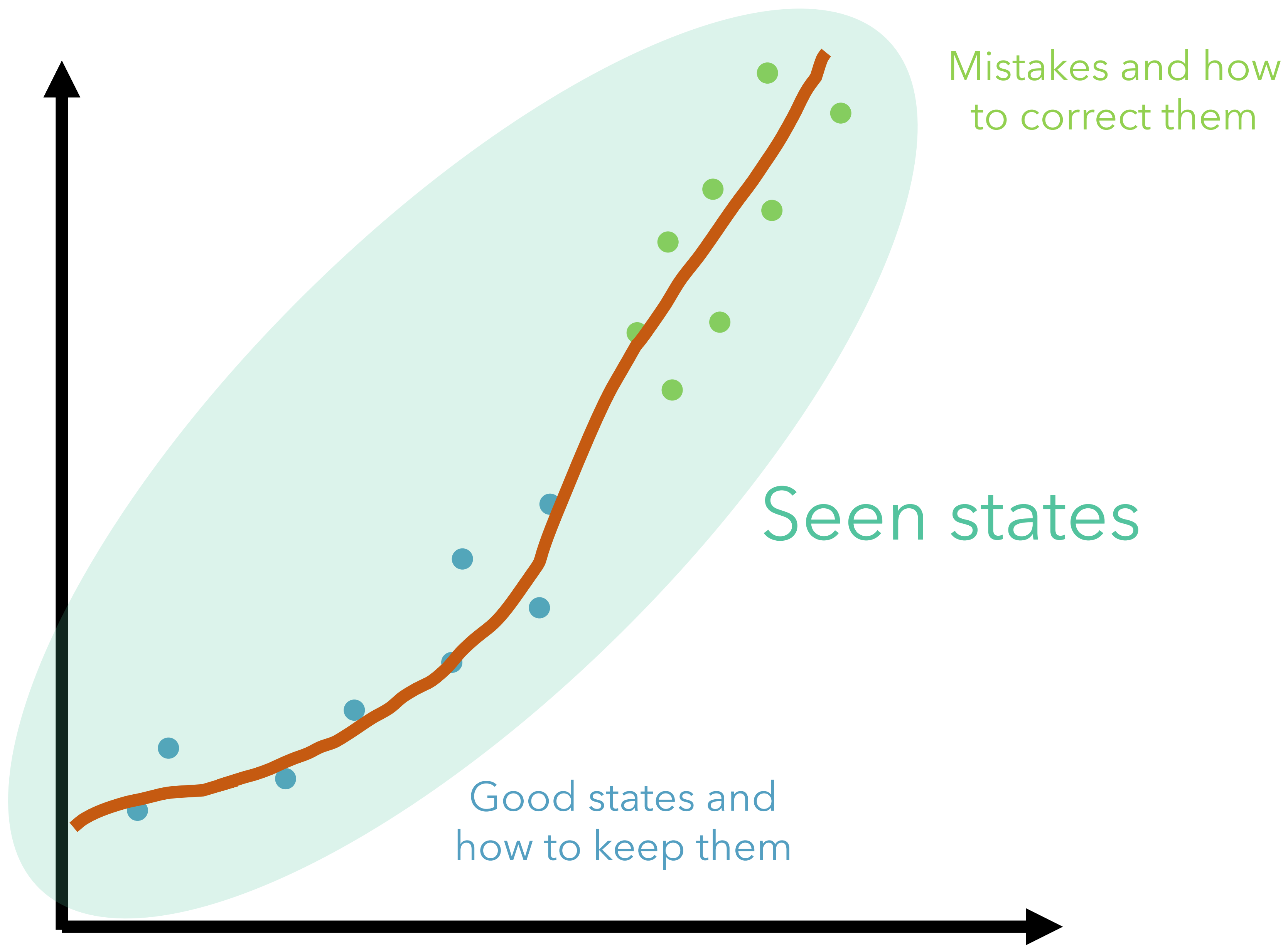
During robot execution, we perform corrections if necessary





Robust Multi-Modal Policies for Industrial Assembly via Reinforcement Learning and Demonstrations: A Large-Scale Study, Jang et al

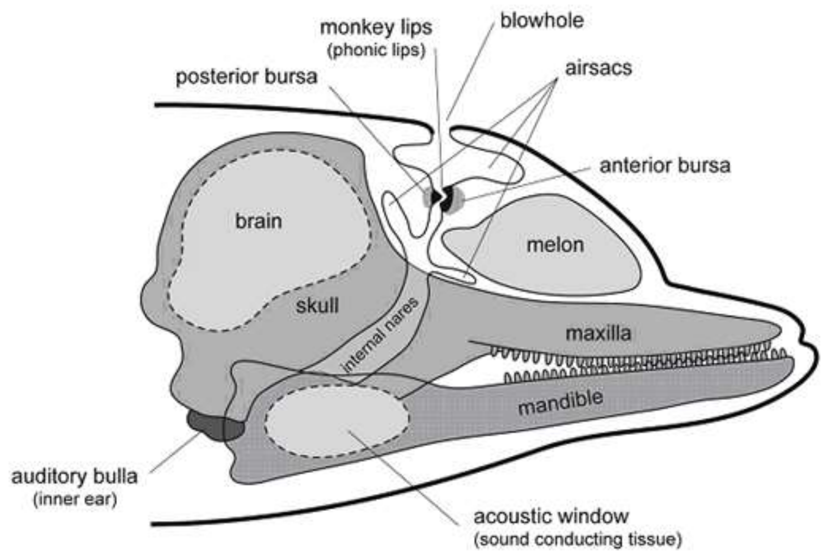

Mistakes and how to correct them
Agent makes a mistake
Expert makes a correction
Good states and how to keep them


Mistakes and how
to correct them
Seen states
Good states and
how to keep them

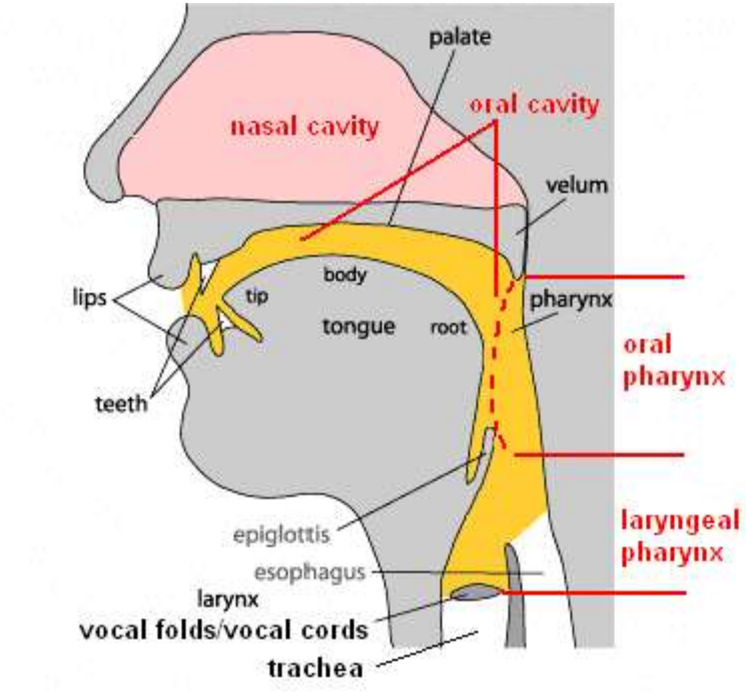
Questions so far?

Sometimes the context shift happens with the **hardware**

Schematic illustration of a dolphin's head anatomy

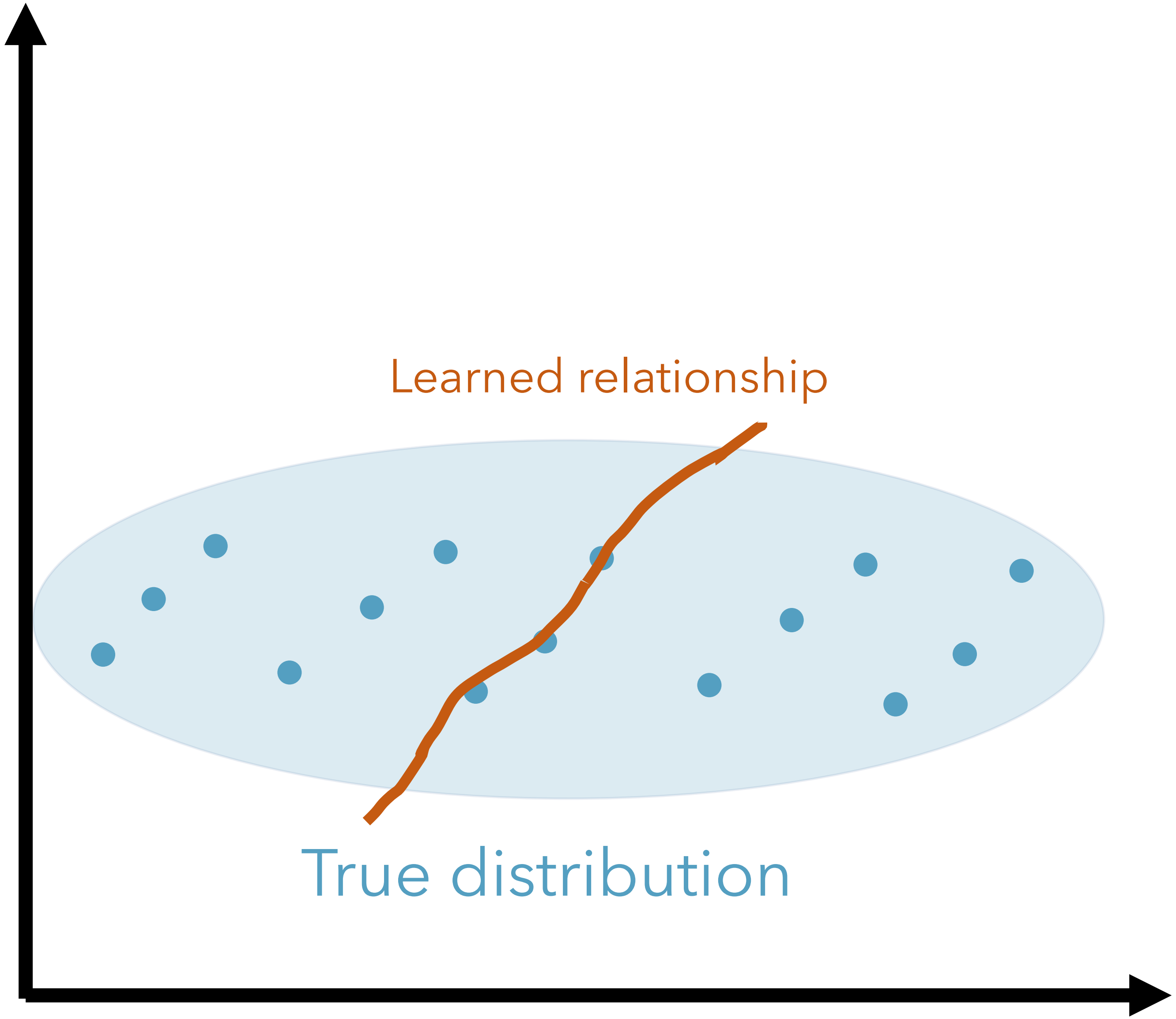


Sound generator: The Monkey Lips/Dorsal Bursae Complex (MLDB)



Sometimes the context shift happens with the **hardware**

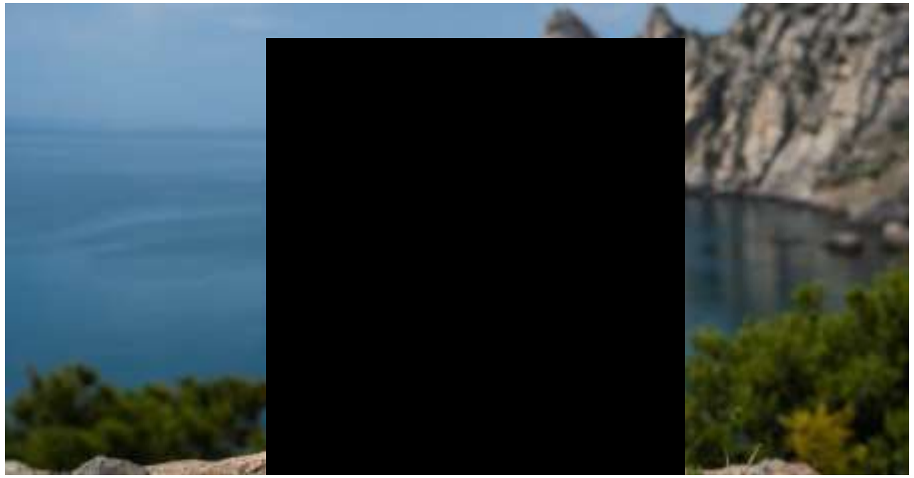
Questions so far?



Bonus Sin:

The model might pick up on the **wrong context**

In fact, sometimes **degenerate** distributions are **mathematically easier to learn**



The Malfunctioning Dolphin

The dolphin picked up on an **irrelevant context**, leading to catastrophic failure upon seeing the new boots

*this is a pedagogically-twisted example. In reality, dolphins are neophobic, and the shiny red object may have triggered this phobia

Humans are
equally susceptible

Sin Resolution

We can **consciously combat** these problems, usually with **more data** and/or **special fitting approaches**

Questions so far?

Ch 2

Superstition

Head-tapping Trua

The *Deadly Sins* of Learning

1. Context Shift
2. **Superstition**
3. Under-exploration

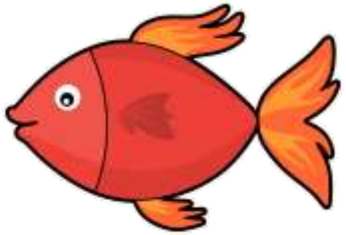
There are many reasons why we may **not** have an **expert correcting** our mistakes


BOB ROSS
OCT. 29, 1942 – JULY 4, 1995

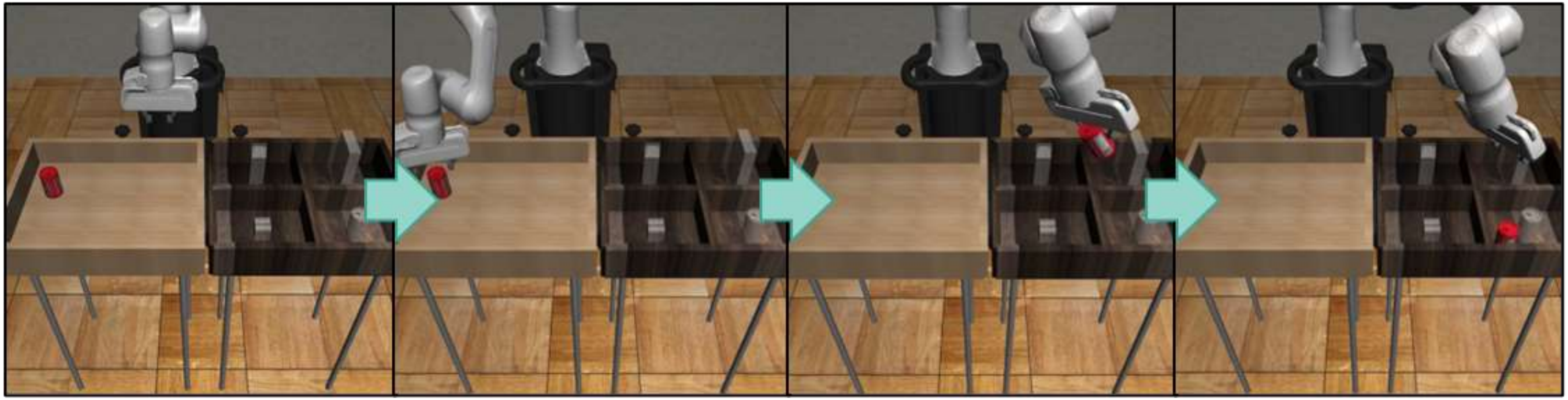
We may **not** even have an **expert** to start with

Does the lack of an expert stop animals from **learning?**

The **environment** can give notions of what's **good** and **bad**



The **environment** can give notions of what’s **good** and **bad**



Bad Good Better Best!

Expert Signal

+ Very easy to fit using standard ML methods

+ Can get good complex behaviors quickly

- Tedious to get large amounts of data

- Overdetermined and can lead to problems with distribution shift

Environment Signal

+ Easy to get (sometimes already given, other times easy to program)

+ Encourages agent to “figure things out”

- Underdetermined (can lead to poor performance)

Let's make a new objective:

"Get as much rewards r as possible during a lifetime T"

The RL Objective™

$$\max \underbrace{\sum_{t=0}^{T} \overbrace{r(s_t, a_t)}^{\text{Rewards given by environment}}}_{\text{Across time}}$$

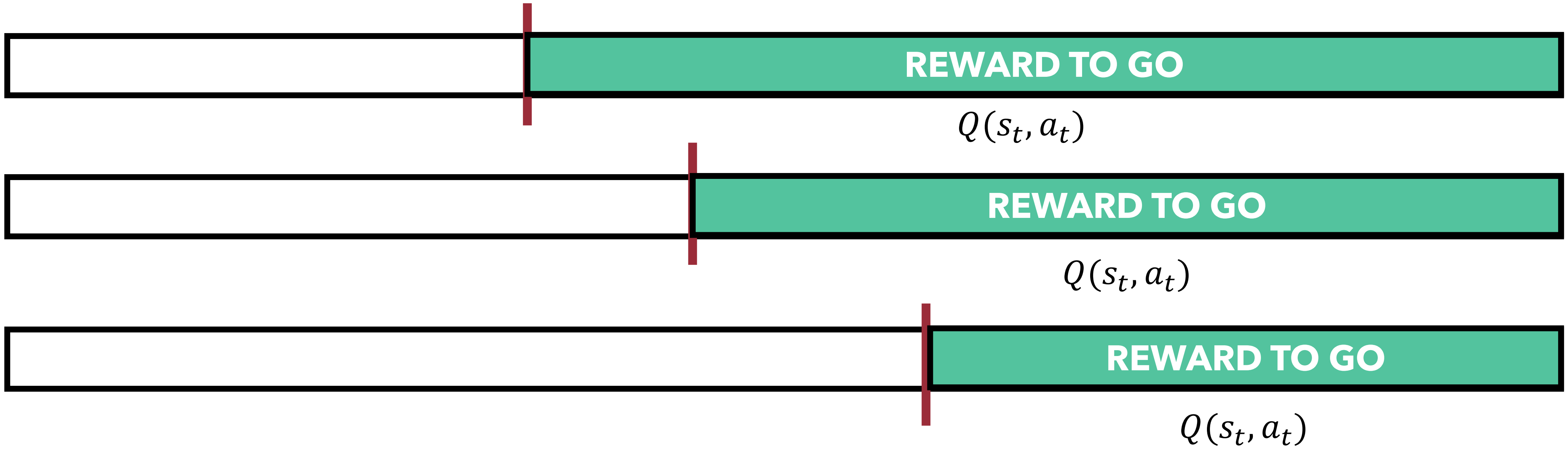
Maximization

Reward to Go

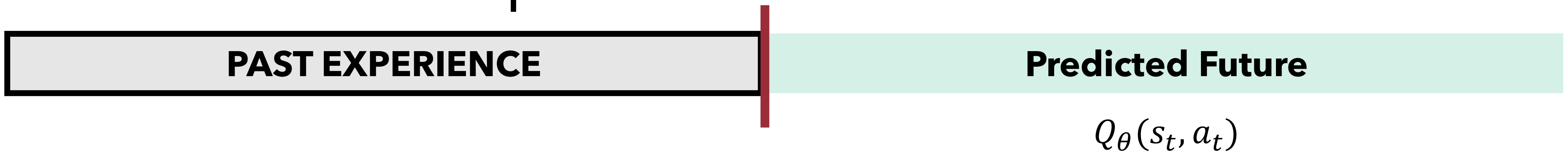
Let's define Q, which is the **rewards** you will get for the **rest of your life.**

$$Q(s_t, a_t) = \sum_{l=t}^{T} r(s_l, a_l)$$

Q represents a sum of future rewards (easily accessible from past experience)



If we have a model of Q, we can predict the future!



How do you **solve the RL objective** using your model of Q?

$$\arg\max_{a_t} Q_\theta(s_t, a_t)$$

Why can't we just optimize over all $a_1, \ldots a_T$ at once?

Reward to Go

$$Q_\theta(s_t, a_t) := \sum_{l=t}^{T} r(s_l, a_l)$$

Optimizing over Q solves the RL Objective

$$\arg\max_{a_t} Q_\theta(s_t, a_t)$$

Yesterday is history
We don't consider the past in this objective

Tomorrow is a mystery
we can't change future actions directly

Today is a gift
All we can do is execute an action in the present

Questions so far?

Input x

Current state and current action

(s_t, a_t)

Output y

Reward to go (Q)

$$\sum_{l=t}^{T} r(s_l, a_l)$$

Model:

$Q_\theta(s_t, a_t)$

First attempt: if we use a neural network as Q_θ, we can **set things up** like we did **before!**

$$\arg\min_\theta D(\underbrace{Q_\theta(\overbrace{s_t, a_t}^{\text{Input}})}_{\text{Prediction}}, \underbrace{\sum_{l=t}^{T} r(s_l, a_l)}_{\text{Real value}})$$

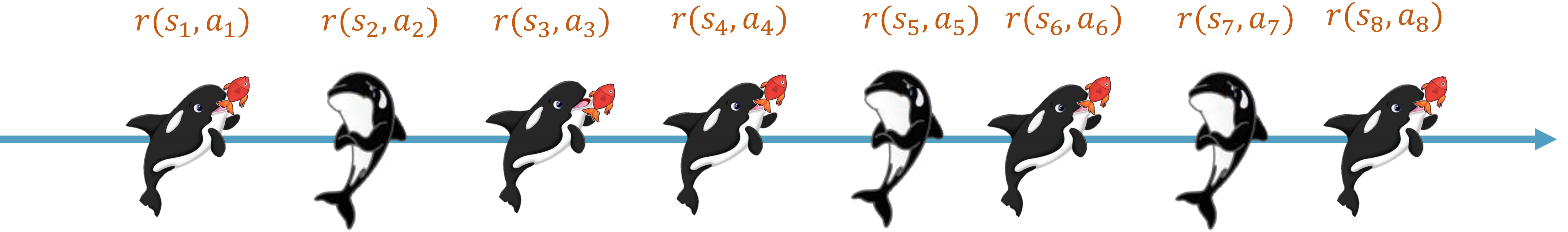
This is very **inefficient!** We need to collect **many, many** lifetimes ("trajectories") to get a good Q!

$$\arg\min_{\theta} D(\underbrace{Q_\theta(\overbrace{s_t, a_t}^{\text{Input}})}_{\text{Prediction}}, \underbrace{\sum_{l=t}^{T} r(s_l, a_l)}_{\text{Real value}})$$

When we **conformed** the objective to our old setup, we **destroyed the meaning** of Q

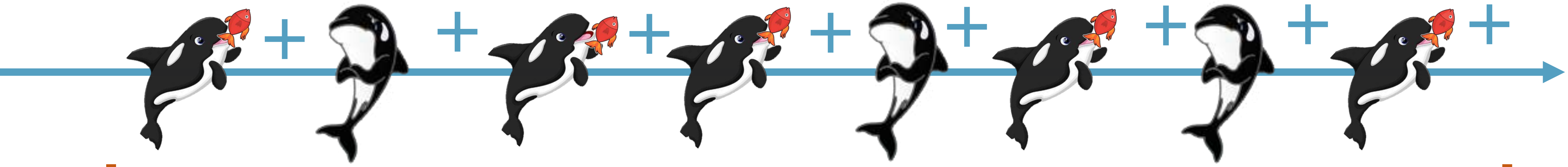
$$\arg\min_\theta D(\underbrace{Q_\theta(\overbrace{s_t, a_t}^{\text{Input}})}_{\text{Prediction}}, \underbrace{\sum_{l=t}^{T} r(s_l, a_l)}_{\text{Real value}})$$

In search of better Q



In search of better Q

$r(s_1, a_1)$ $r(s_2, a_2)$ $r(s_3, a_3)$ $r(s_4, a_4)$ $r(s_5, a_5)$ $r(s_6, a_6)$ $r(s_7, a_7)$ $r(s_8, a_8)$



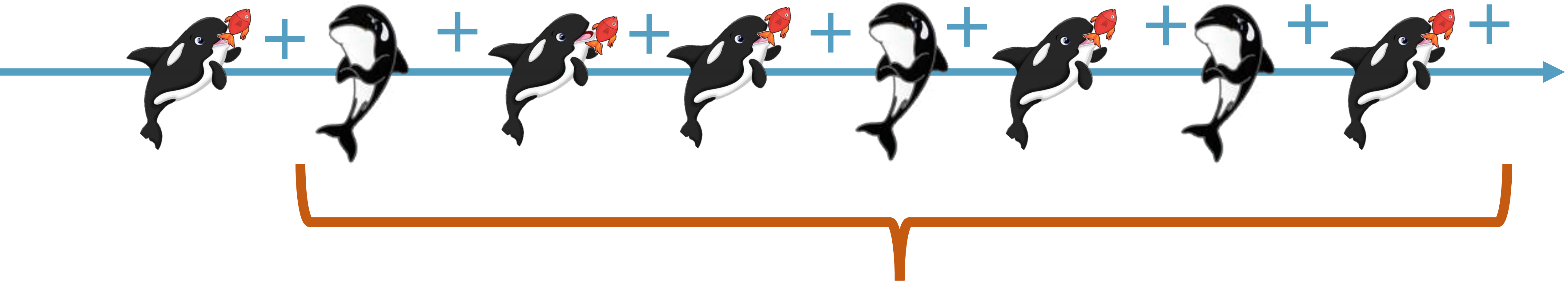
$$\sum_{t=1} r(s_t, a_t)$$

Assuming that
Q is accurate…

$$Q(s_1, a_1)$$

In search of better Q

$r(s_1, a_1)$ $r(s_2, a_2)$ $r(s_3, a_3)$ $r(s_4, a_4)$ $r(s_5, a_5)$ $r(s_6, a_6)$ $r(s_7, a_7)$ $r(s_8, a_8)$

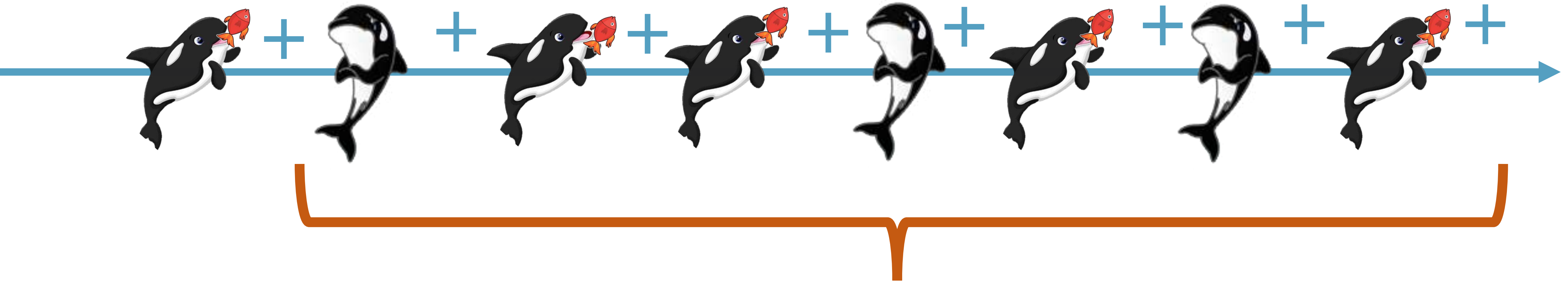


$$\sum_{t=2} r(s_t, a_t)$$

$$Q(s_2, a_2)$$

In search of better Q

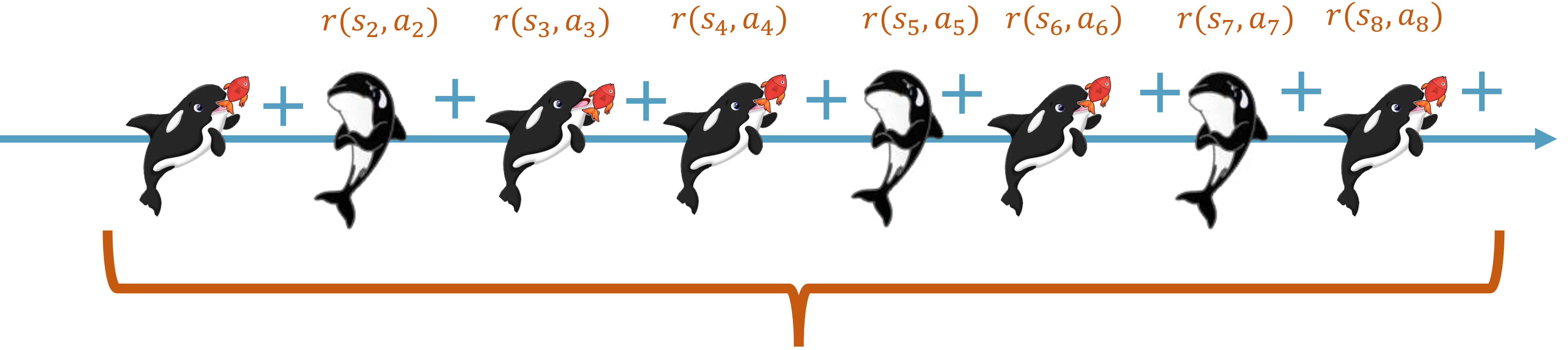
$r(s_2, a_2)$ $r(s_3, a_3)$ $r(s_4, a_4)$ $r(s_5, a_5)$ $r(s_6, a_6)$ $r(s_7, a_7)$ $r(s_8, a_8)$



$$\sum_{t=2} r(s_t, a_t)$$

$$r(s_1, a_1) + Q(s_2, a_2)$$

In search of better Q



$$\sum_{t=1} r(s_t, a_t)$$

$$Q(s_1, a_1) = \quad r(s_1, a_1) + Q(s_2, a_2)$$

We can define the “reward to go” in **terms of itself!**

$$Q(s_t, a_t) = r(s_t, a_t) + Q(s_{t+1}, a_{t+1})$$

“The value for the rest of your life is the current reward plus the value of the rest of your life in the next time step”

So, why don't we constrain the **model of Q** like this?

$$\underbrace{Q_\theta(s_t, a_t)}_{\text{Prediction}} := \underbrace{r(s_t, a_t) + Q_\theta(s_{t+1}, a_{t+1})}_{\text{Target}}$$

Using this observation, we make our **new objective**!

$$\arg\min_{\theta} D(\underbrace{Q_\theta(s_t, a_t)}_{\text{Estimated life value}}, \overbrace{r(s_t, a_t)}^{\text{Current reward}} + \underbrace{Q_\theta(s_{t+1}, a_{t+1})}_{\text{Estimated life value one step from now}})$$

Questions so far?

$$\arg\min_{\theta} D(\underbrace{Q_\theta(s_t, a_t)}_{\text{Estimated life value}}, \overbrace{r(s_t, a_t)}^{\text{Current reward}} + \underbrace{Q_\theta(s_{t+1}, a_{t+1})}_{\text{Estimated life value one step from now}})$$

This is known as a **Bellman Backup.** It is one of the fundamental equations of **reinforcement learning!**

But…why is it called a **Backup??**

Pro-tip: whenever something is weird, find a stupid example and see what happens.

State-Actions	Reward
s_1, a_2	0
s_2, a_2	0
s_3, a_3	0
s_4, a_4	0
s_5, a_5	1

But…why is it called a **Backup??**

We also assume that we can write down $Q(s, a)$ explicitly on the same table

State-Actions	Reward	Q value
s_1, a_1	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	0

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_1	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	0

$$Q(s,a) \leftarrow r + Q(s',a')$$

To train, we need (s, a, r, s', a')

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_1	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	0

$$\boxed{Q(s_1, a_1)} \leftarrow \boxed{r_1} + \boxed{Q(s_2, a_2)}$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_1	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	0

$$Q(s_2, a_2) \leftarrow r_2 + Q(s_3, a_3)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_1	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	0

$$Q(s_3, a_3) \leftarrow r_3 + Q(s_4, a_4)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_1	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	0

$$\boxed{Q(s_4, a_4)} \leftarrow \boxed{r_4} + \boxed{Q(s_5, a_5)}$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_1	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	1

$$Q(s_4, a_4) \leftarrow r_4 + 0 \text{ (you're dead)}$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	1

This is what happens when we run the data once. What happens if we **run it again**?

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	1

$$Q(s_1, a_1) \leftarrow r_1 + Q(s_2, a_2)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	1

$$Q(s_2, a_2) \leftarrow r_2 + Q(s_3, a_3)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	0
s_5, a_5	1	1

$$Q(s_3, a_3) \leftarrow r_3 + Q(s_4, a_4)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	1
s_5, a_5	1	1

$$Q(s_4, a_4) \leftarrow r_4 + Q(s_5, a_5)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	1
s_5, a_5	1	1

$$Q(s_4, a_4) \leftarrow r_4 + 0 \text{ (you're dead)}$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	1
s_5, a_5	1	1

Interesting! Again!

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	1
s_5, a_5	1	1

$$Q(s_1, a_1) \leftarrow r_1 + Q(s_2, a_2)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	0
s_4, a_4	0	1
s_5, a_5	1	1

$$Q(s_2, a_2) \leftarrow r_2 + Q(s_3, a_3)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

$$Q(s_3, a_3) \leftarrow r_3 + Q(s_4, a_4)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

$$Q(s_4, a_4) \leftarrow r_4 + Q(s_5, a_5)$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

$$Q(s_4, a_4) \leftarrow r_4 + 0 \text{ (you're dead)}$$

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	0
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

See a pattern?

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	0
s_2, a_2	0	1
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

After another run through the data

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	1
s_2, a_2	0	1
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

And after another run…

But…why is it called a **Backup??**

State-Actions	Reward	Q value
s_1, a_2	0	1
s_2, a_2	0	1
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

And after another run…

But…why is it called a **Backup??**

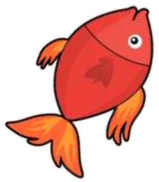
State-Actions	Reward	Q value
s_1, a_2	0	1
s_2, a_2	0	1
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

Hmm…looks like we've stopped changing things

The Bellman Backup operation **pushes** a reward signal into the **past**, allowing us to **plan** for the future

State-Actions	Reward	Q value
s_1, a_2	0	1
s_2, a_2	0	1
s_3, a_3	0	1
s_4, a_4	0	1
s_5, a_5	1	1

The Bellman Backup operation **pushes** a reward signal into the **past**, allowing us to associate actions to rewards



Sidenote: in reality, we add a **discount factor γ** to keep us more focused on present rewards

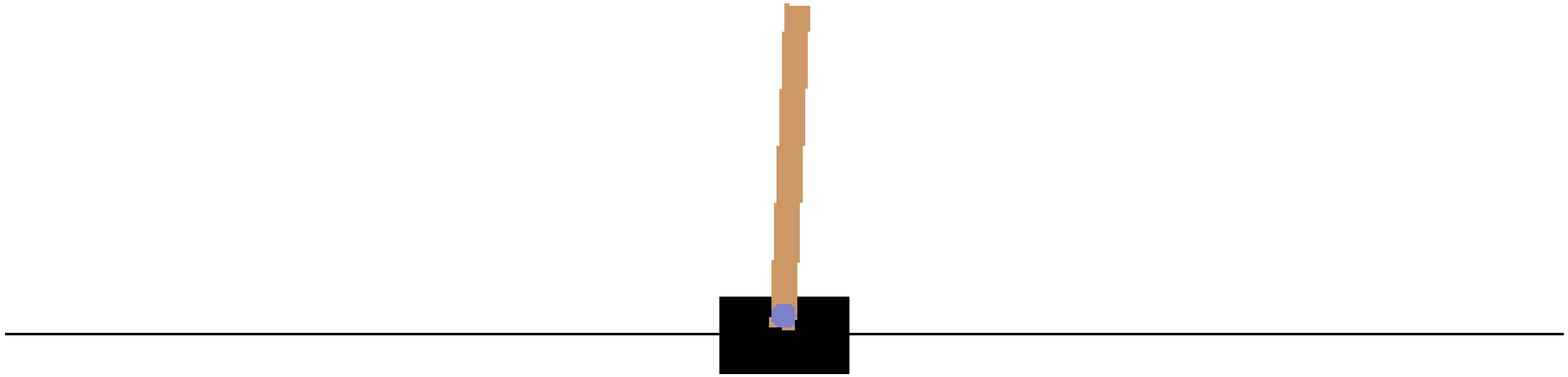
State-Actions	Reward	Q value
s_1, a_2	0	0.9606
s_2, a_2	0	0.9703
s_3, a_3	0	0.9801
s_4, a_4	0	0.99
s_5, a_5	1	1

$$Q(s, a) \leftarrow r + \gamma Q(s', a')$$

Here, we use $\gamma = 0.99$

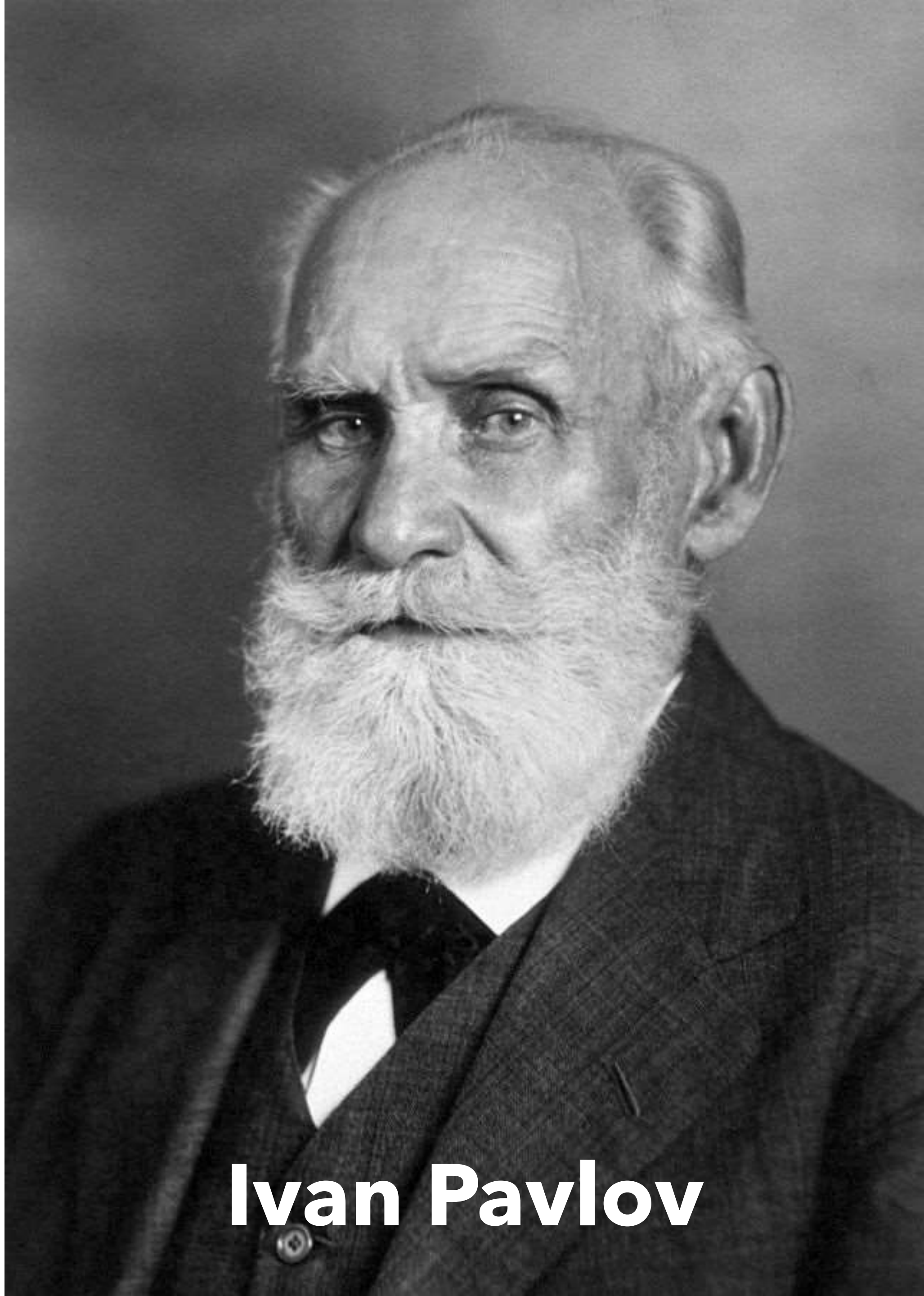
Questions so far?

Once you have a Q, just find the **best action** at each step ($\max_a Q(s, a)$)



You've just done RL!

Bellman backups happen all the time in **real life**—it is how humans and animals **learn** as well!

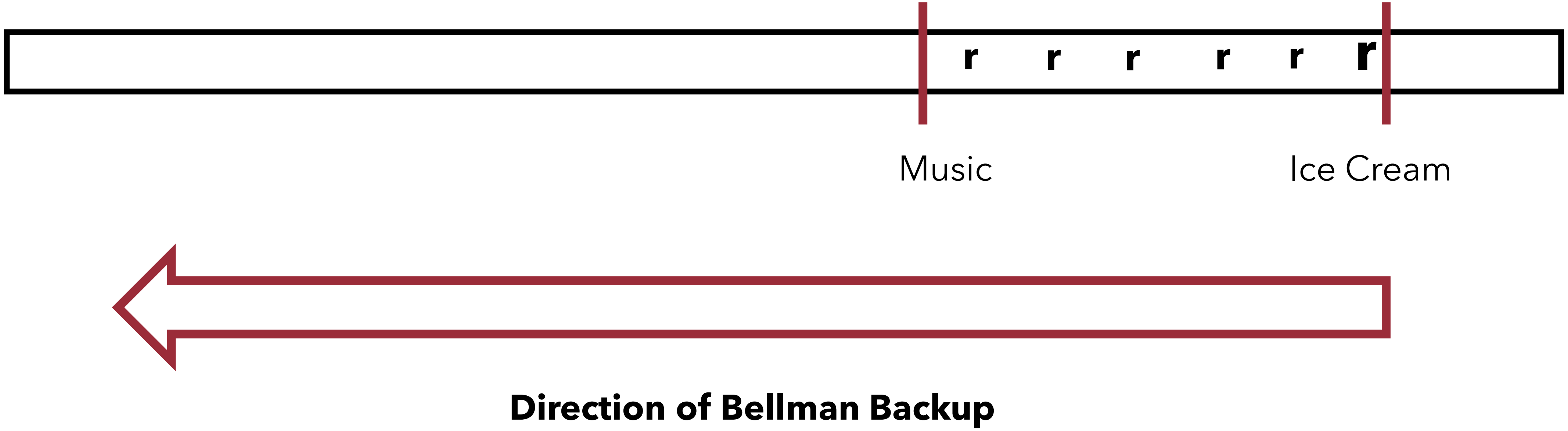


Food is a naturally-occurring reinforcement. This is a **primary reinforcer** (r = nonzero)

Music is normally a **neutral stimulus** (r = 0)

We run to the sound of ice cream truck **music,** and we might even **salivate**

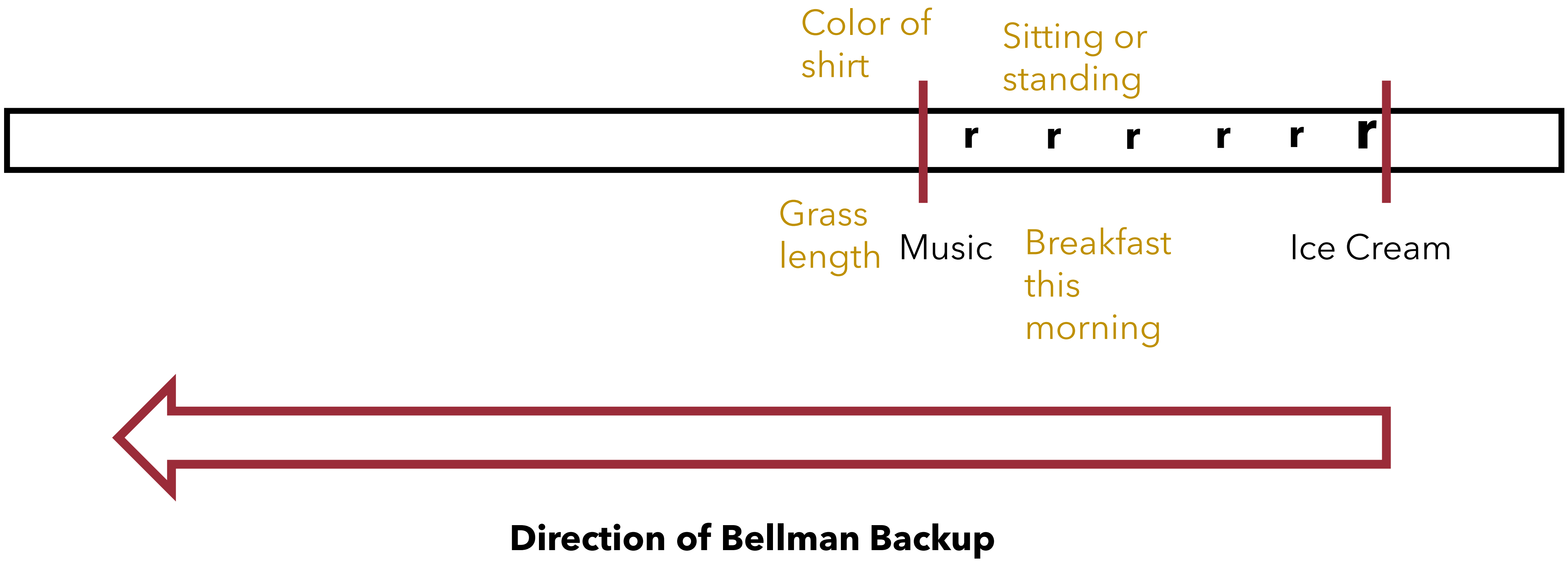
This is because we **propagated** a non-zero reward (ice cream) to a zero-reward (music) using the **Bellman Backup!**



Question: does the reward stop at the music?

The music is **associated** with ice cream through the Bellman Backup

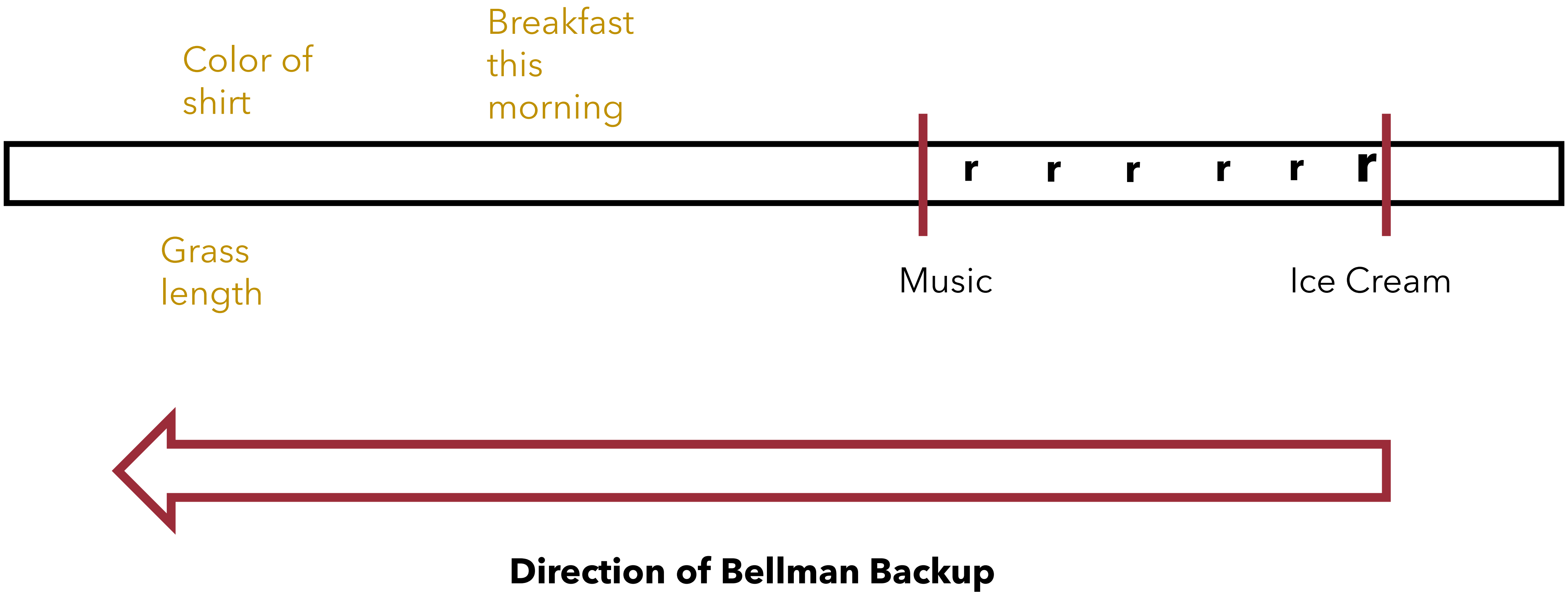
Why don't you associate the color of your mom's jacket, the length of grass outside, the price of gas, with the ice cream truck?



For every reward, there are myriad of **potential causes**. To learn properly, we need to assign the **right credit.**

Potential Causes of Ice Cream

- Length of your grass
- Ice cream truck proximity
- Breakfast this morning
- Currently sitting / standing
- The children running towards the ice cream truck (i.e. the children create the ice cream)



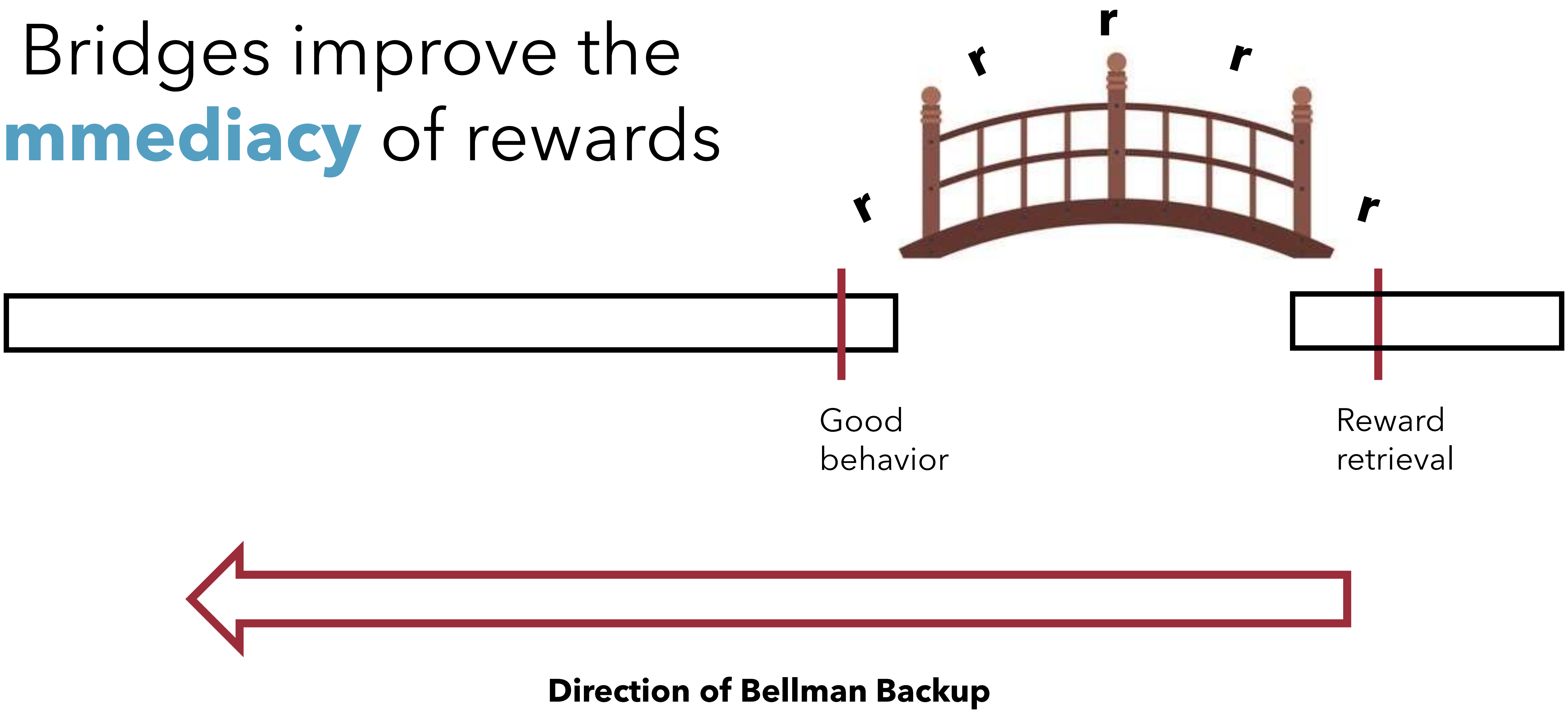
Observation #1: **Immediacy** of signal-reward pair is helpful

It is impossible to reward a whale at the **exact time** it does something good.

Trainers recognize importance of **reward immediacy,** so they use a sound signal to as a reward **stand-in.**

This is called a **"Bridge"**

Bridges improve the **immediacy** of rewards



When immediacy isn't adequate

**A car stops near a pedestrian.
What happened?**

- A pedestrian force field?
- The driver pushing on the brakes?

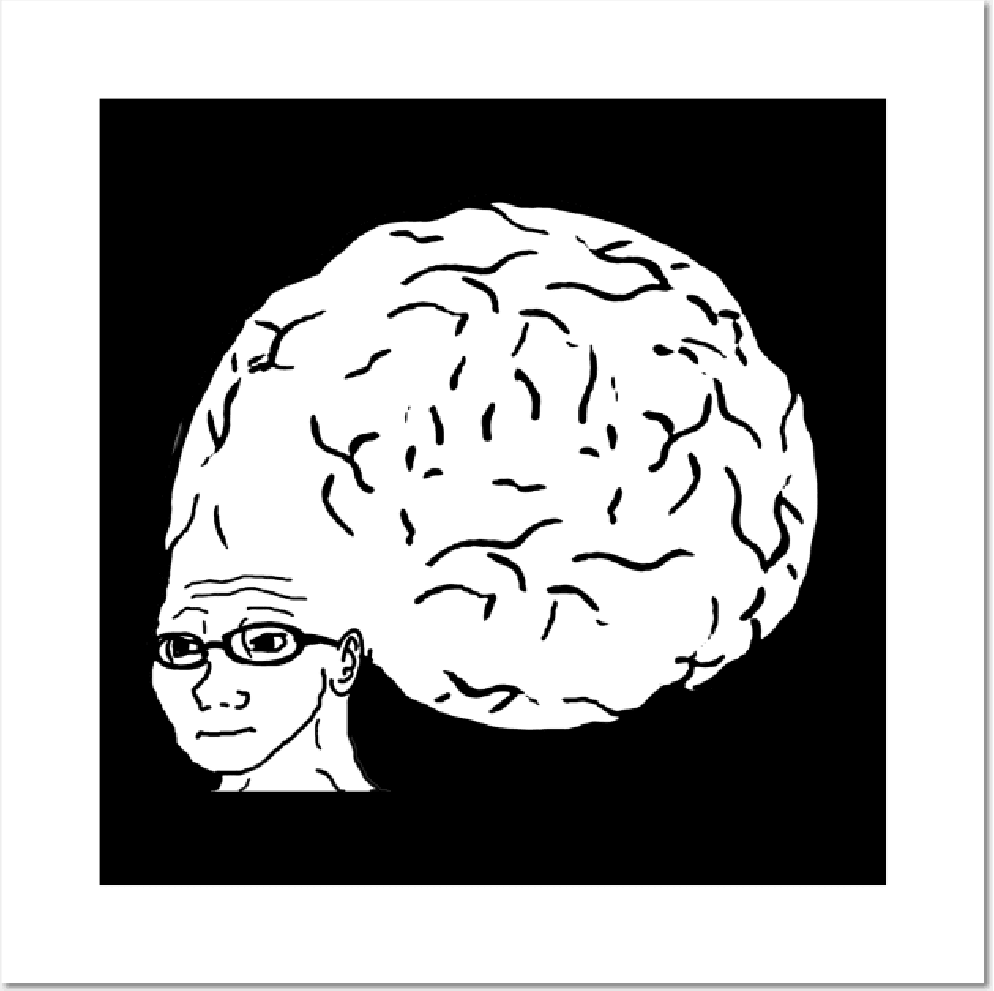
Potential Causes of Ice Cream

- ~~Length of your grass~~
- Ice cream truck proximity
- ~~Breakfast this morning~~
- Currently sitting / standing
- The children running towards the ice cream truck (i.e. the children create the ice cream)

But what about these?

As humans, we have an intricate knowledge of the world around us.

We siphon from this **world-knowledge** to perform **highly accurate** credit assignment



From a dolphin’s point of view: you’ve just been given a bridge + fish. What was the cause?

- Head movement
- Tail movement
- The pattern I swam in the pool
- Standing still
- Nothing (purely random)

Not so easy, is it?

The sin of superstition

Trua's head-tapping was a **superstition**: he thought that tapping was a necessary part of the behavior

Superstitions are a failure in credit assignment

$$\arg\min_{\theta} D(\underbrace{Q_\theta(s_t, a_t)}_{\text{Estimated life value}}, \overbrace{r(s_t, a_t)}^{\text{Current reward}} + \underbrace{Q_\theta(s_{t+1}, a_{t+1})}_{\text{Estimated life value one step from now}})$$

Bad credit-assignment can arise during the fitting of Q_θ, leading to robots creating **absurd superstitions**

Sin Resolution

Make rewards **closer** to the cause (immediacy)

Add **world knowledge** to your robot / dolphin

Ch 3

Under-Exploration

Can you train this?

The *Deadly Sins* of Learning

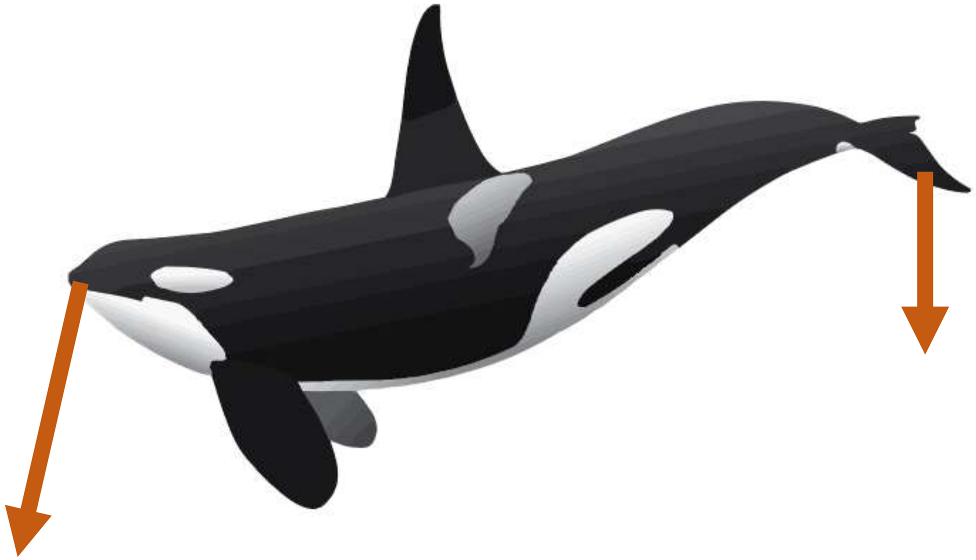
1. Context Shift
2. Superstition
3. **Under-exploration**

What we know so far

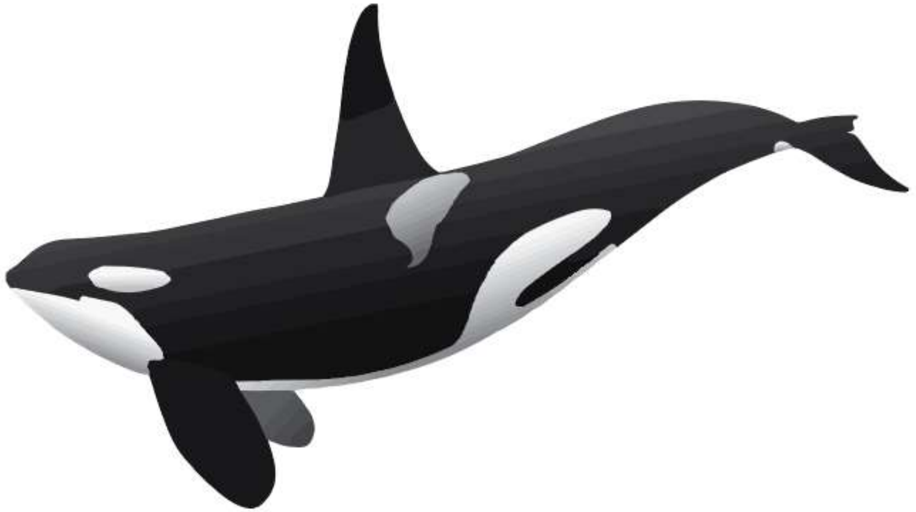
Given rewards, we can perform the best behavior

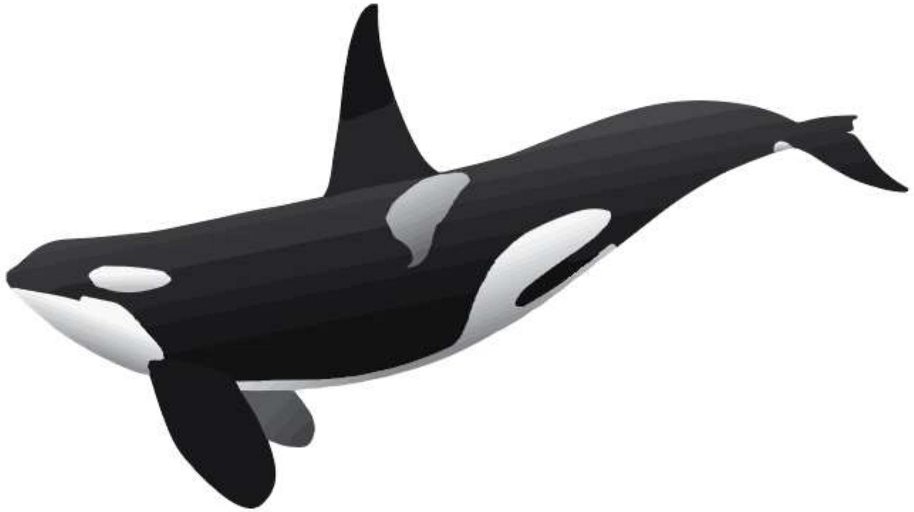
How to give the rewards to get desired behavior

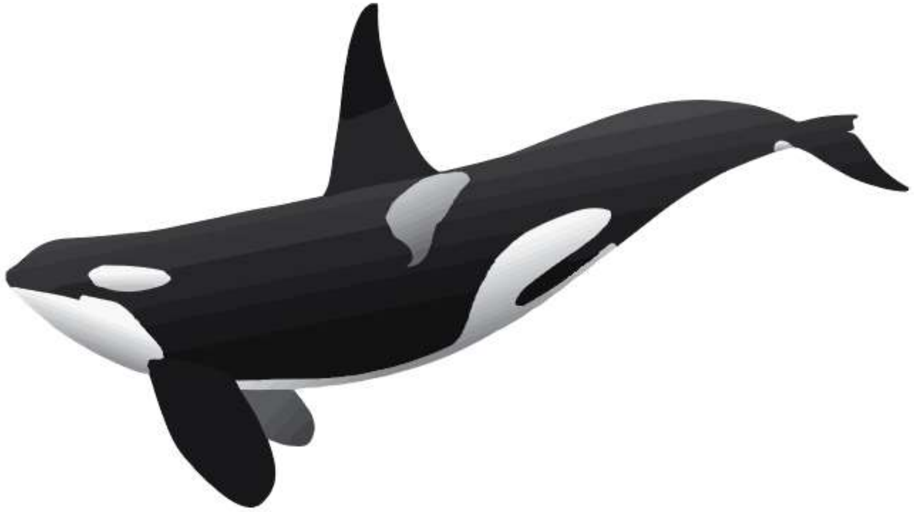
When and how do you **reward?**

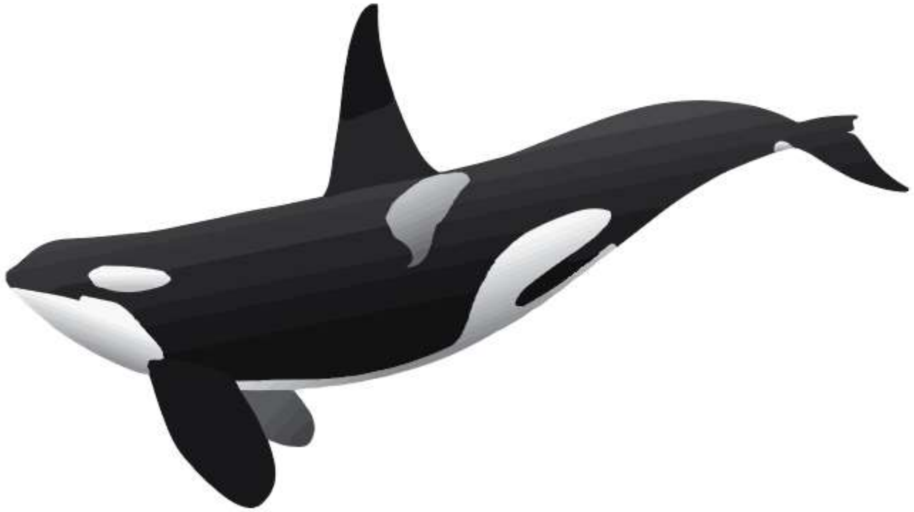


What could happen?

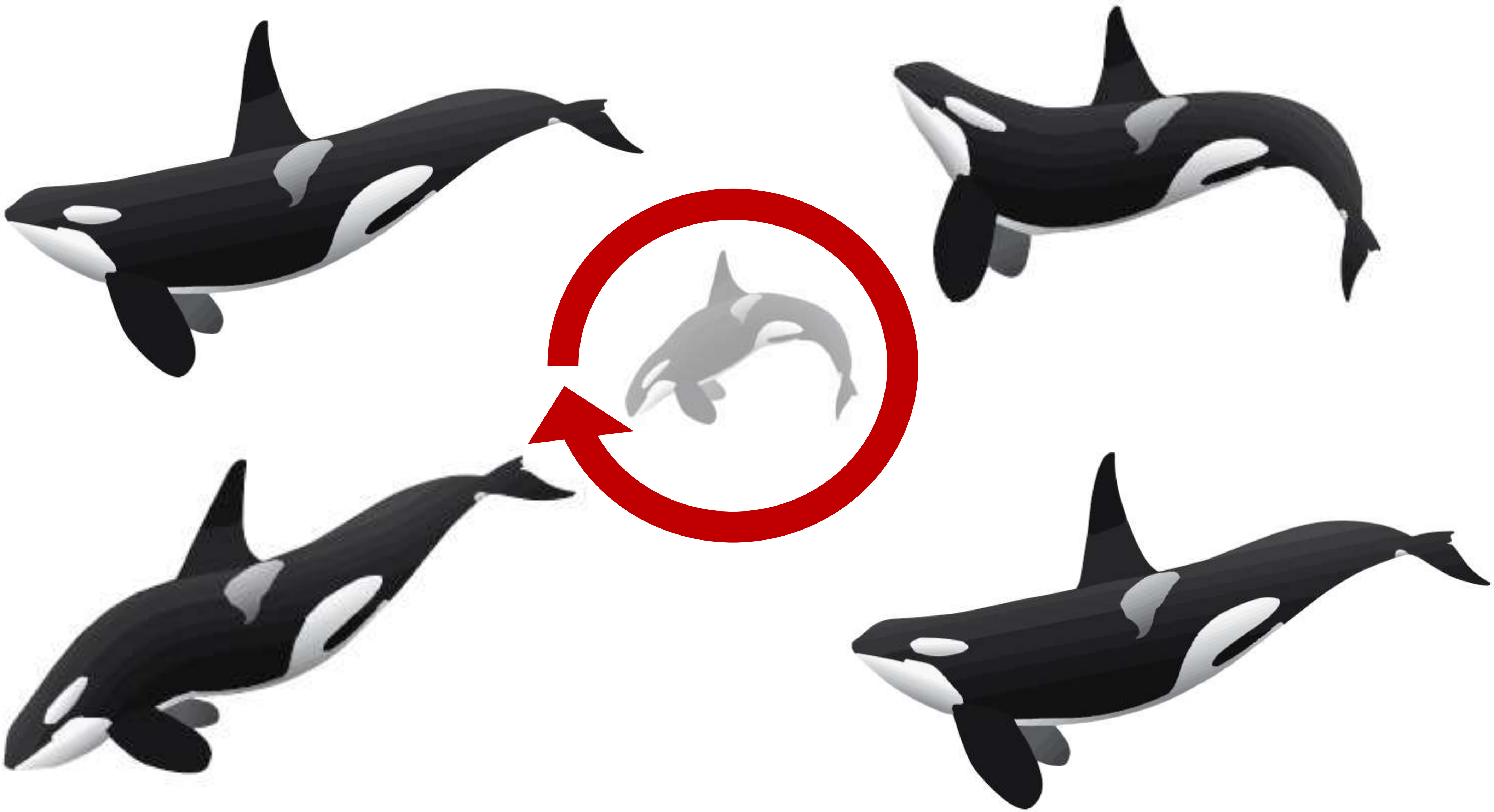






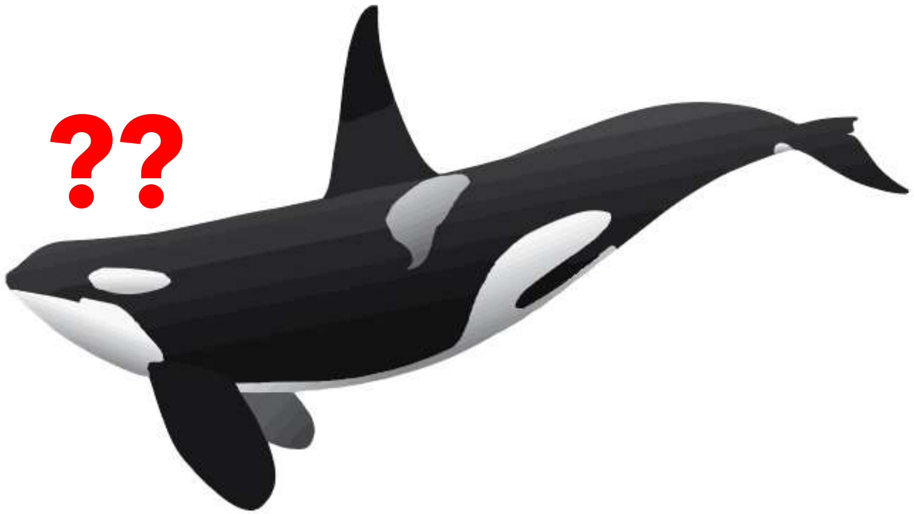


What could happen?



The sin of under-exploration

With a **"sparse"** reward like the whistle, we transmit very **little information**, which may lead to very little progress.



With a **"dense"** reward, we can convey **more information** and succeed

Similarity-O-Meter

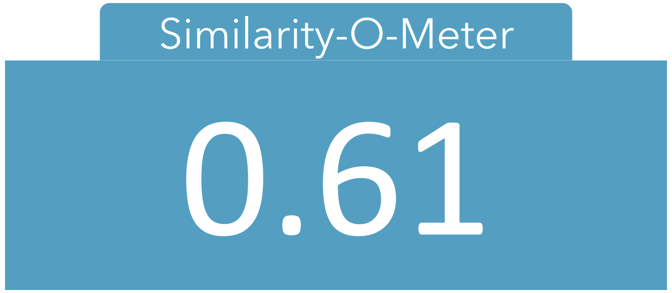
0.63

With a **"dense"** reward, we can convey **more information** and succeed

Similarity-O-Meter

0.69

With a **"dense"** reward, we can convey **more information** and succeed



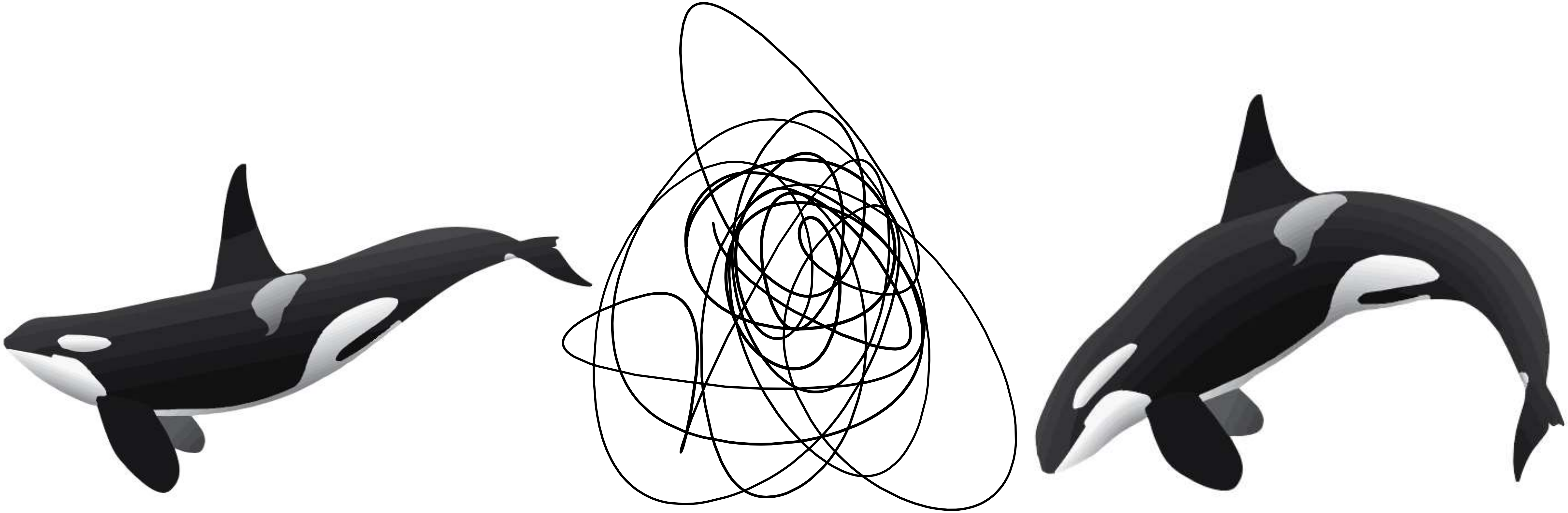
With a **"dense"** reward, we can convey **more information** and succeed

Similarity-O-Meter

0.99

Big
problem...whales
can't read.

Maybe we're too ambitious...



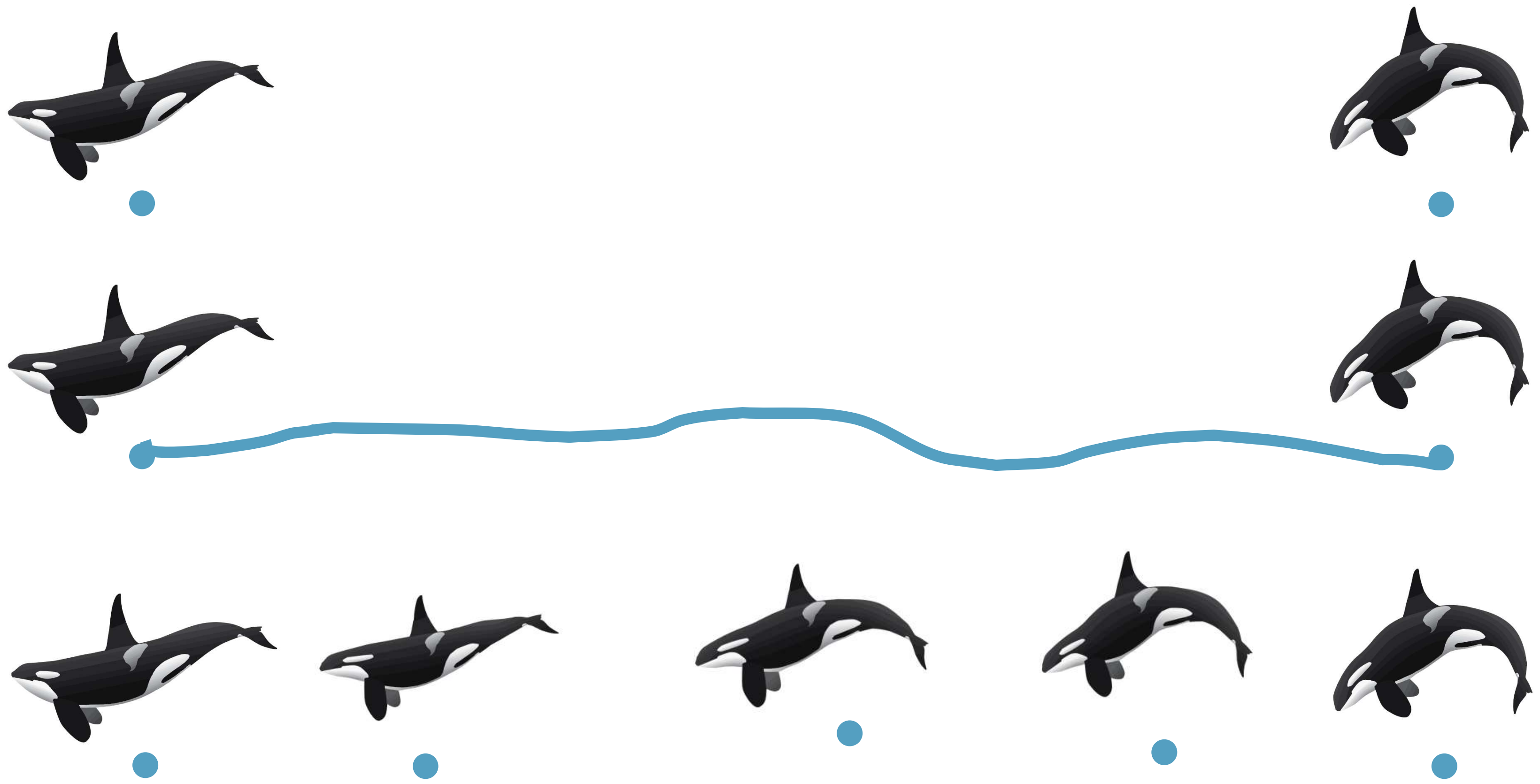
Provide sparse rewards at every **small change.** This is known as creating **successive approximations.**

You approximate a dense reward with a sequence of sparse rewards

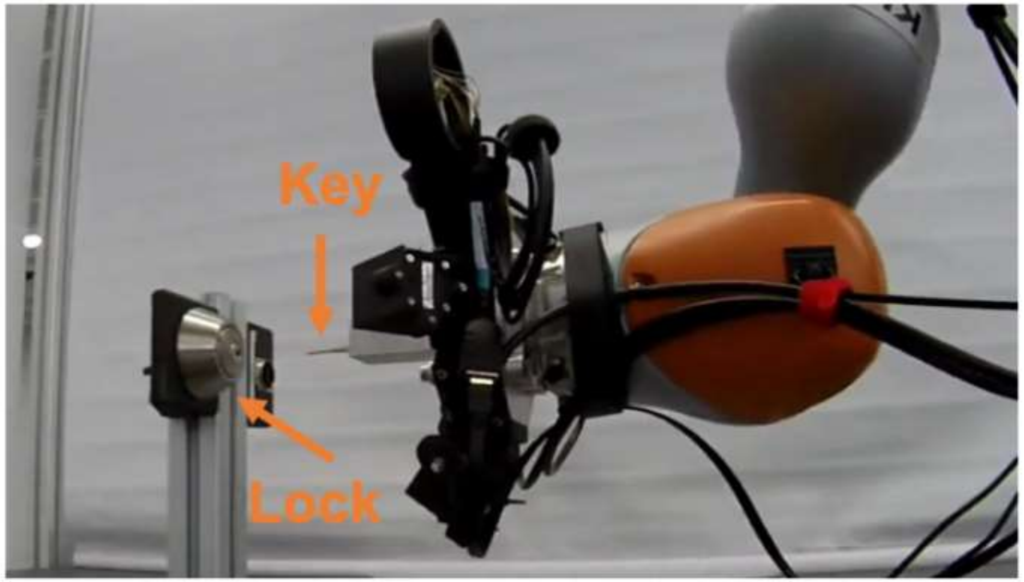
Sparse (naïve trainer)

Shaped (impossible / difficult ideal)

Successive approximations (compromise)



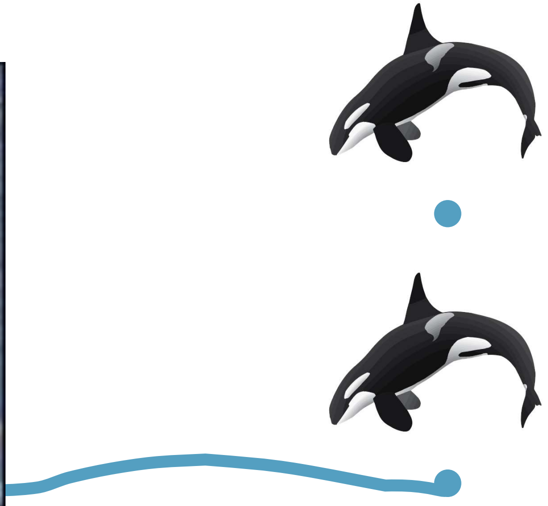
In robot learning, we call successive approximations a **"curriculum"**



Learning Robust Multi-Modal Polices for Industrial Robotic Tasks via Reinforcement Learning and Demonstrations, Luo et al

You approximate a dense reward with a sequence of sparse rewards

Sparse (naïve trainer)



Shaped (impossible / difficult ideal)

Successive approximations (compromise)

But how do you jump this gap?

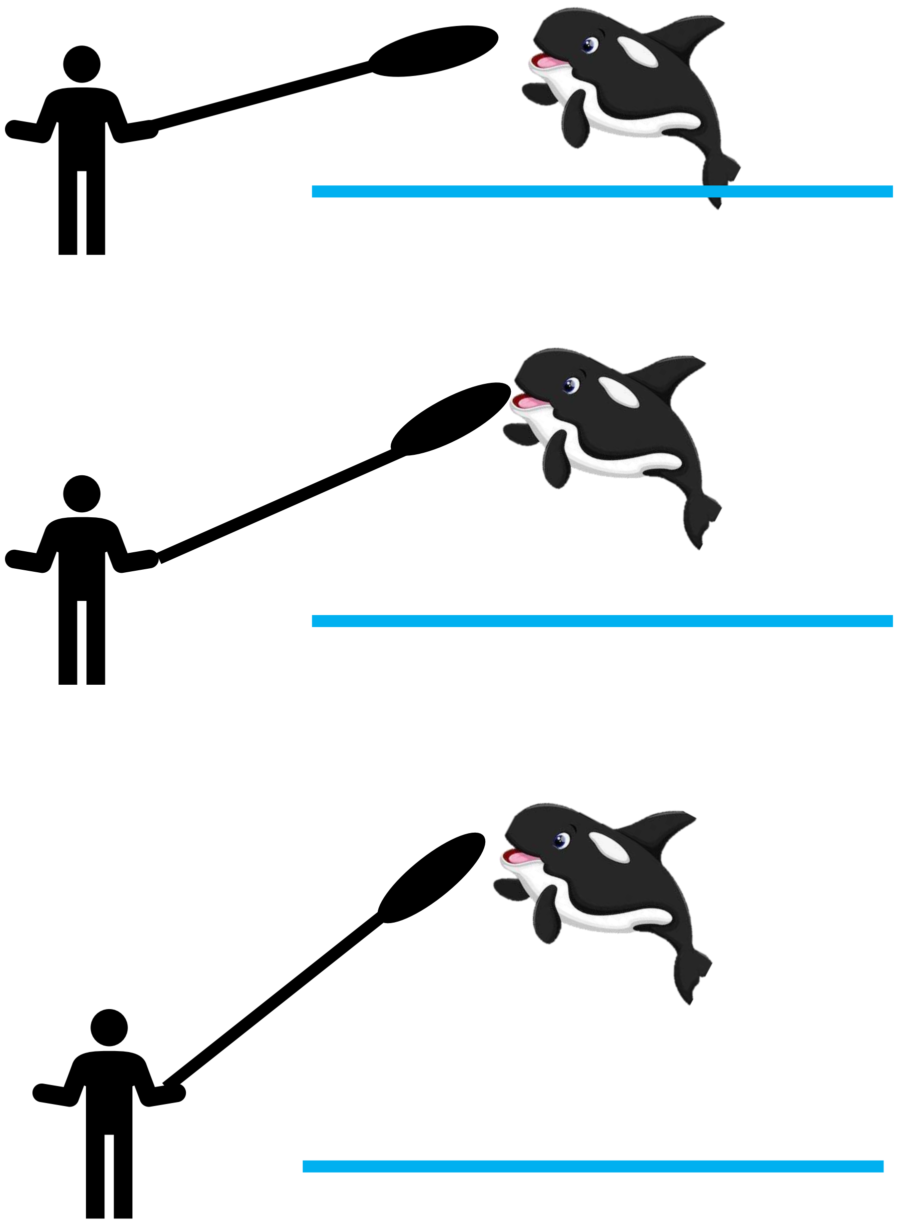
Questions so far?

How do you solve this?

(live demo)

***this demo has a high rate of failure…**

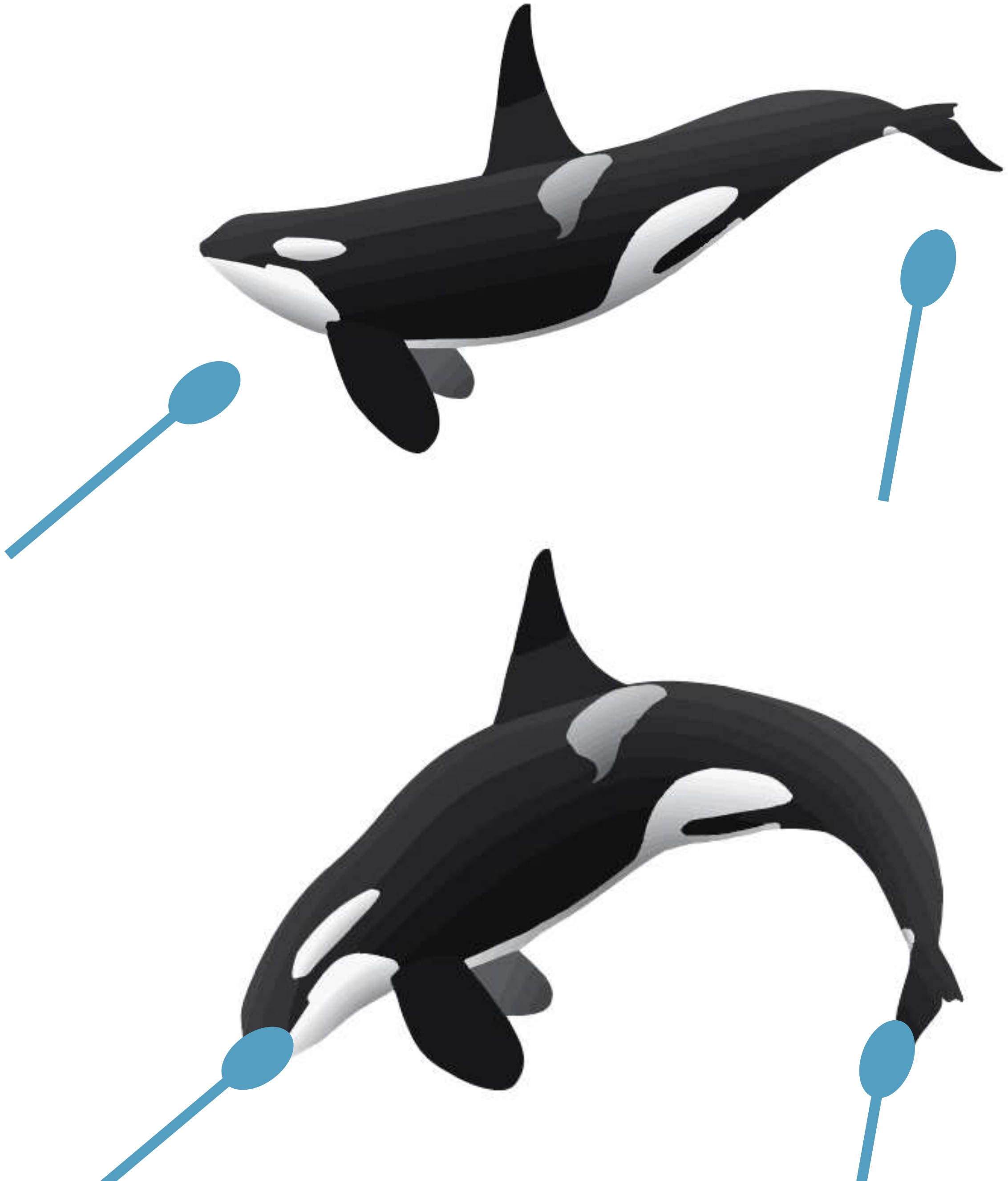
The target rod turns **any hard task** into a **reaching task** (which uses existing, primitive skills)



Successive Approximation Demonstration

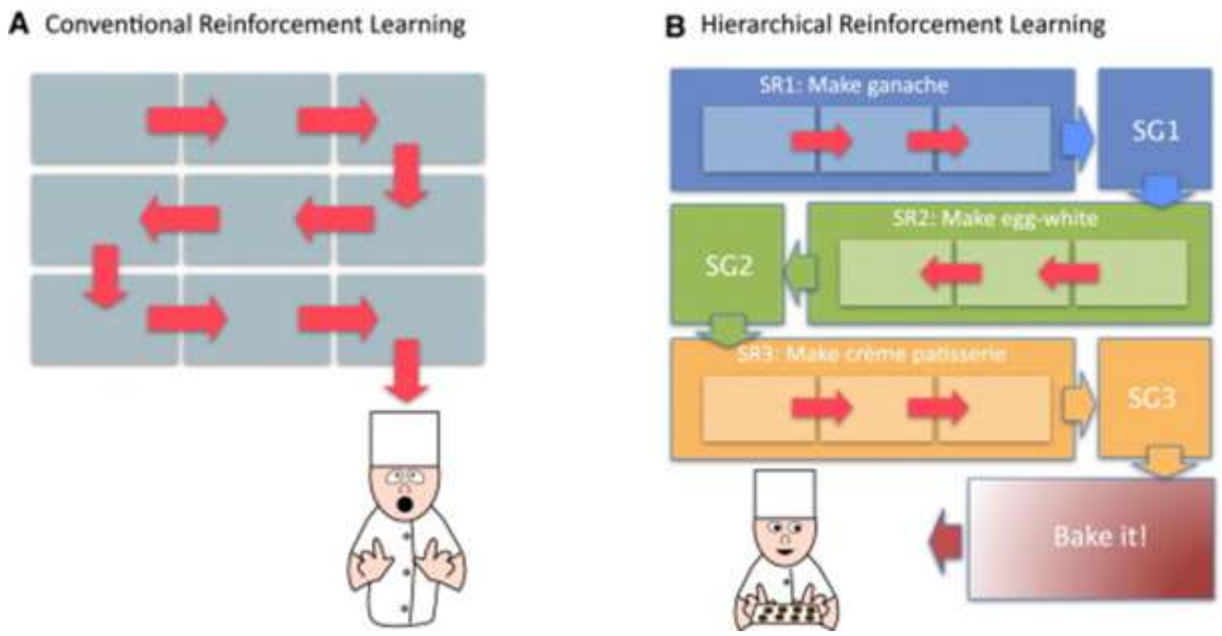
IMATA Workshops 2022
Orkid "Cartwheel"

Use **multiple targets** to solve difficult problems



Use **multiple targets** to solve difficult problems

The **same trick** is used in robot learning to **simplify** complicated tasks



What we know so far

Given rewards, we can perform the best behavior

Sort of...

How to give the rewards to get desired behavior

Sin Resolution

Make the **problem simpler** to solve

Or...improve the ability to **explore** (active question)

To help with the learning process, we (trainers, roboticists) can use **successive approximations** on a task

If we aren't using punishments / physical restraints, we run into a problem of…**things not happening.**

Questions so far?

The *Deadly Sins* of Learning

1. Context Shift
2. Superstition
3. Under-exploration

Sin of Context Shift

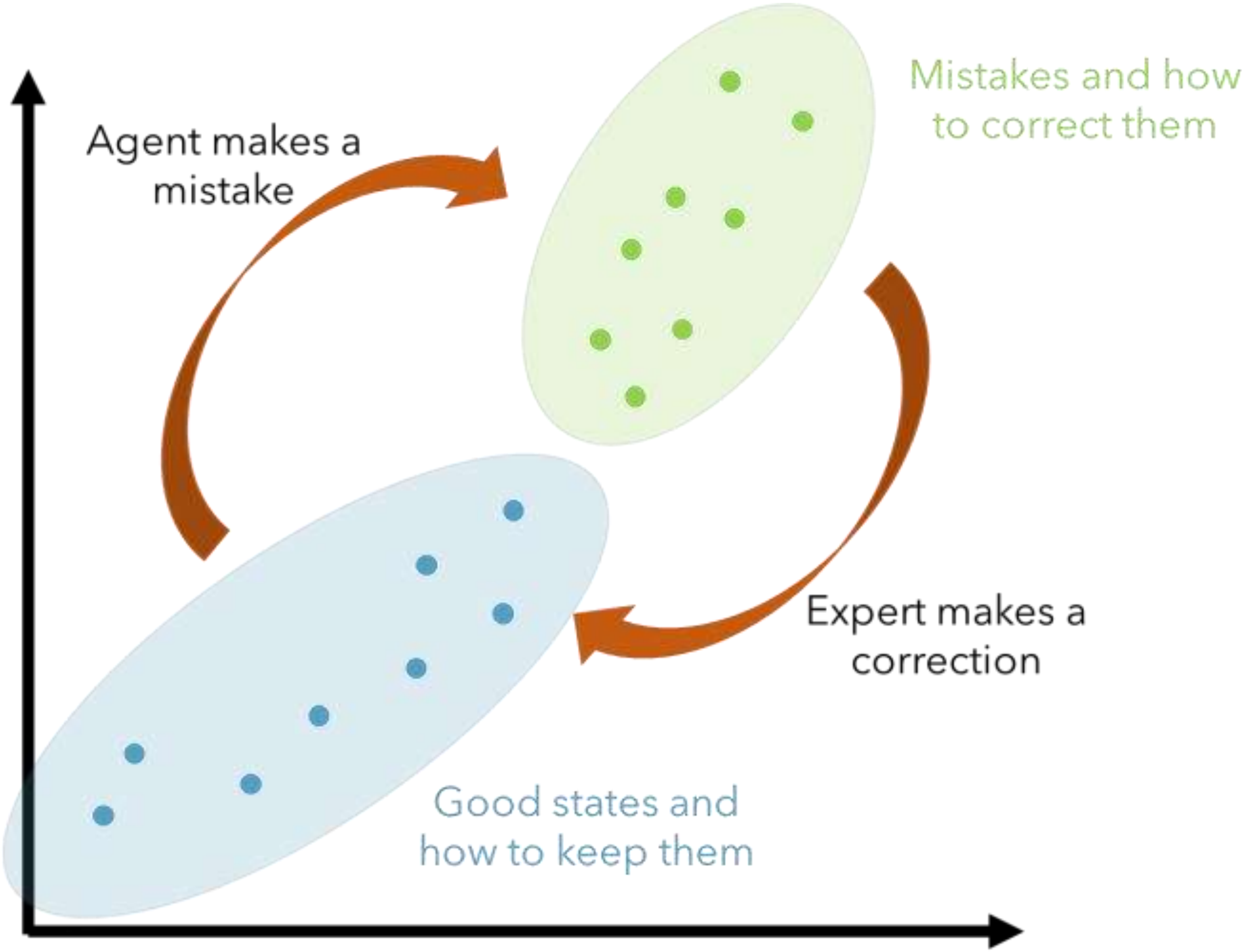
Our existing knowledge may not be well equipped for the real world

Sin Resolution

Add more experiences, strategically

Sin Resolution

Add more experiences, strategically



The *Deadly Sins* of Learning

1. Context Shift
2. Superstition
3. Under-exploration

Sin of Superstition

The laws of cause-and-effect are harder than we make it look

Head-tapping is an important part of this behavior

Sin Resolution

Make rewards more immediate, and teach your animal/robots the rules of the human world

The *Deadly Sins* of Learning

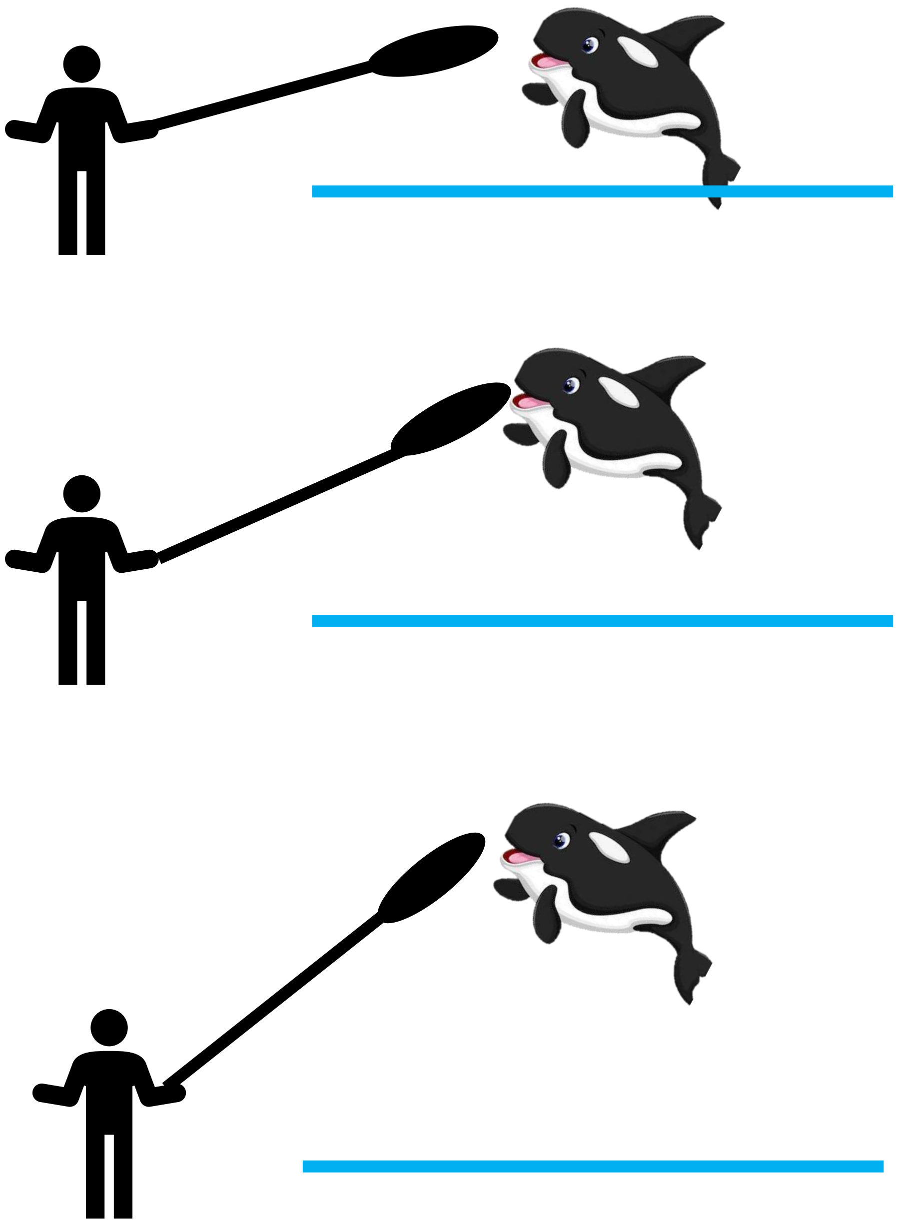
1. Context Shift
2. Superstition
3. Under-exploration

Sin of Under-Exploration

There are many ways of doing things wrong; only a few ways of doing it right

Sin Resolution

Break the task into simpler steps. Shape complex behaviors gradually.



The *Deadly Sins* of Learning

1. Context Shift
2. Superstition
3. Under-exploration

These are all challenges faced while trying to understand the world. They are not technical problems, but rather deep problems of all life.

Ch 7

Epilogue

“We shall not cease from exploration, and the end of all our exploring will be to arrive where we started and know the place for the first time”

- T.S. Eliot


STANFORD


Final Questions?