Carnegie Mellon

School of Computer Science

Deep Reinforcement Learning and Control

Introduction

Spring 2020, CMU 10-403

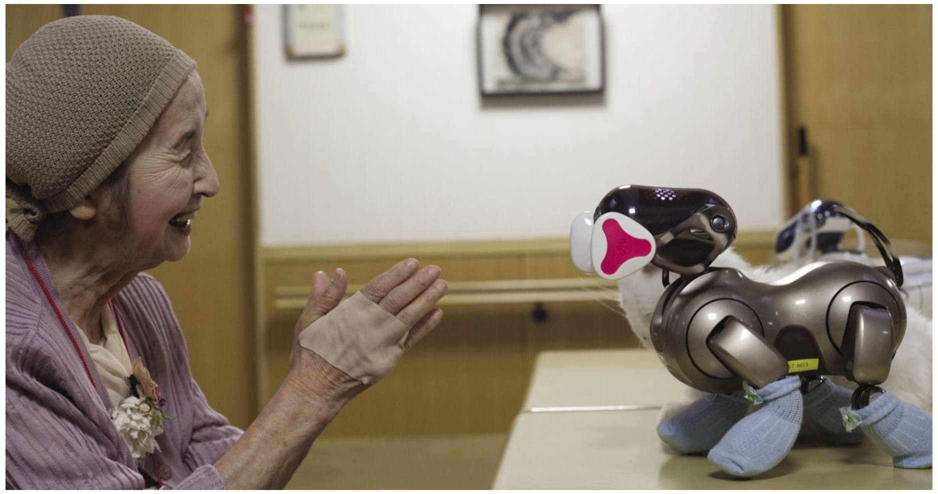
Katerina Fragkiadaki

Course Logistics

- Course website: https://cmudeeprl.github.io/Spring202010403website/ all you need to know
- Grading:
  - 5 Homework assignments: implementation and question/answering many optional and extra grade questions, teams of 2 people each- 60%
  - 2 quizzes - 40%
- Resources: AWS for those that do not have access to GPUs
- HW code for guidance will be in TensorFlow, you can use your favorite deep learning package
- People can audit the course, unless there are no seats left in class
- The readings on the schedule are required unless noted otherwise

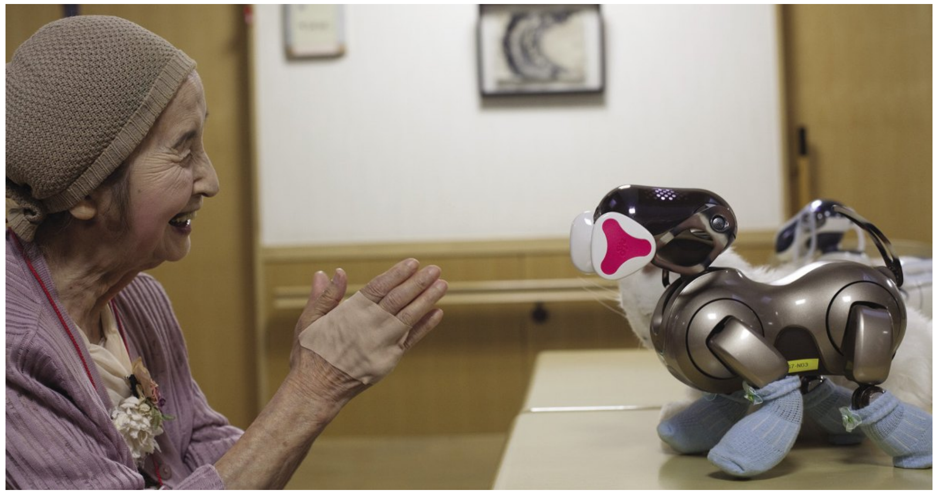
Goal of the course: Learning to act

Building agents that learn to act and accomplish goals in dynamic environments



Goal of the course: Learning to act

Building agents that learn to act and accomplish goals in dynamic environments

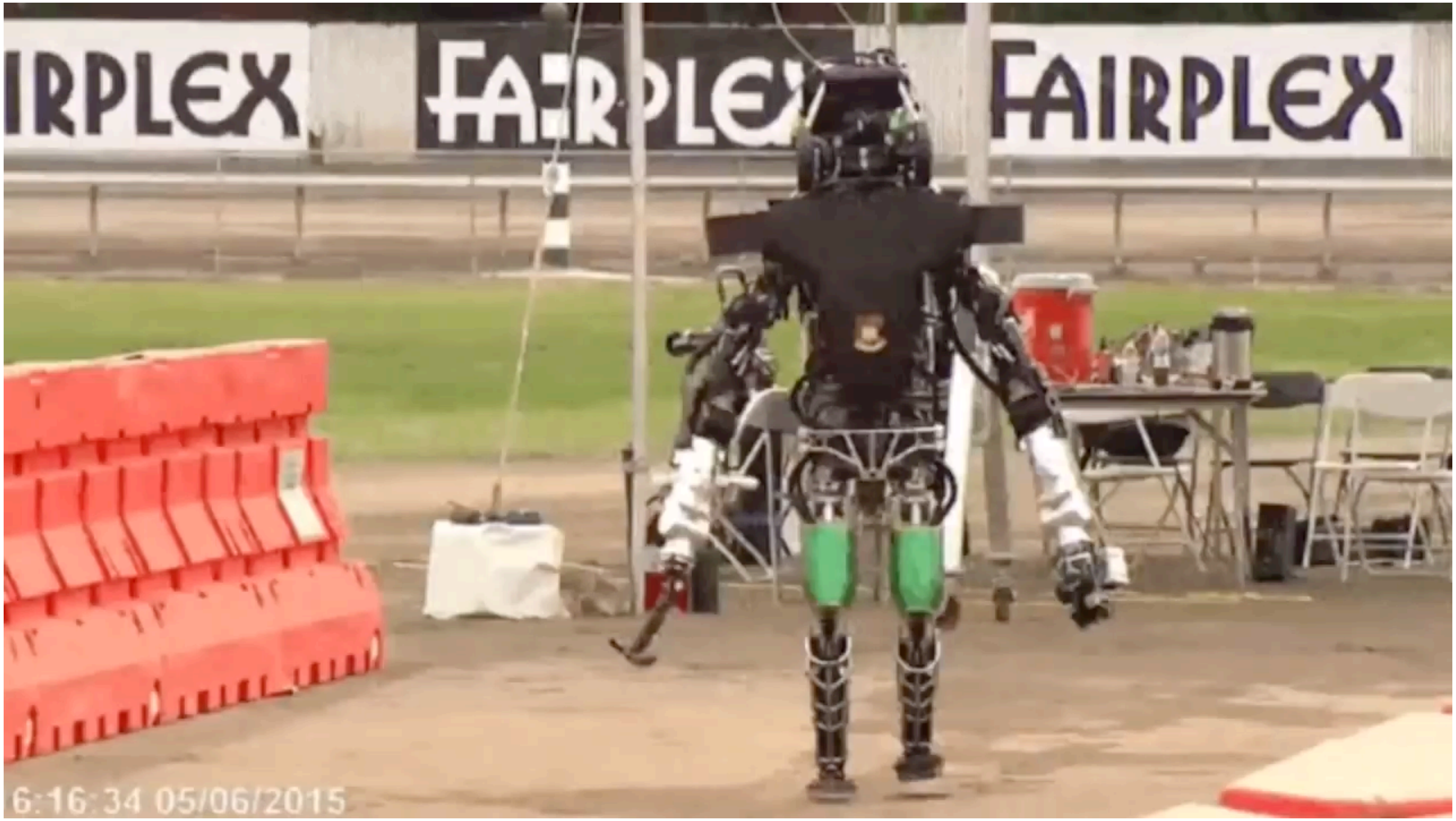


…as opposed to agents that execute pre-programmed behaviors in static environments…

How far are we?

Here the robot is teleoperated: it does not actually operate on its own.

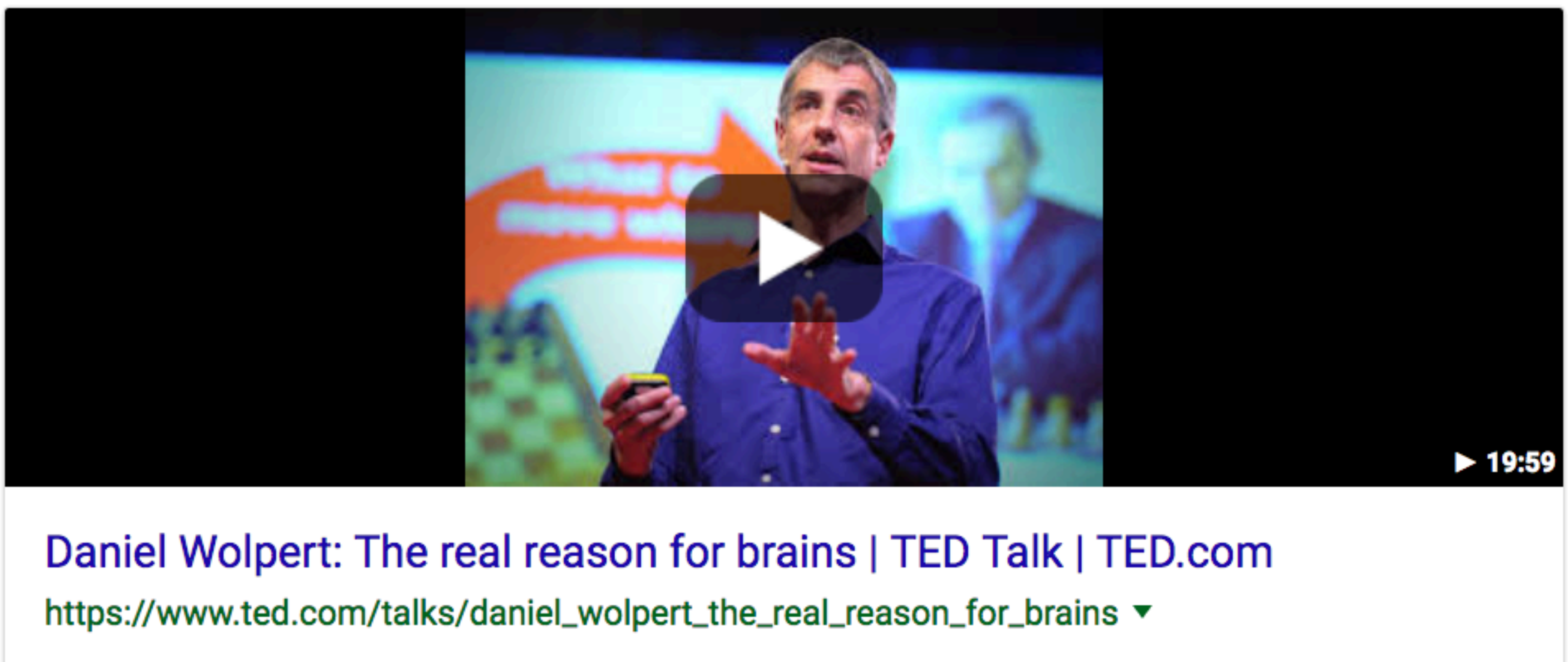
How far are we?



Here the robot operates on its own.
A: not very far.

Motor control is Important

“The brain evolved, not to think or feel, but to control movement.”
Daniel Wolpert



https://www.ted.com/talks/daniel_wolpert_the_real_reason_for_brains?language=en

Motor control is Important

“The brain evolved, not to think or feel, but to control movement.”
Daniel Wolpert

Sea squirts digest their own brain when they decide not to move anymore

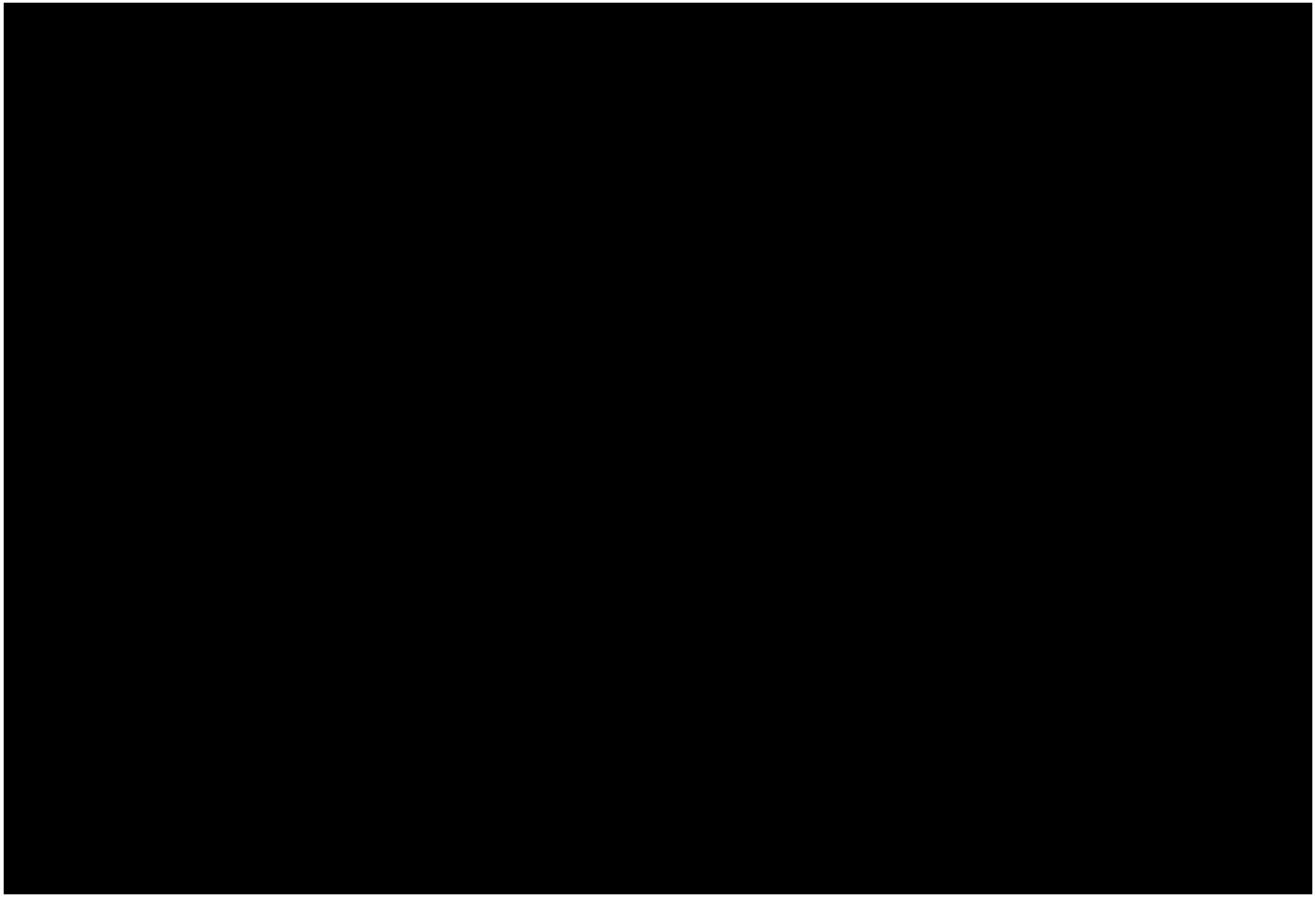
How are behaviors shaped?

Behavior is primarily shaped by reinforcement rather than free-will.

B.F. Skinner
1904-1990
Harvard psychology

- behaviors that result in praise/pleasure tend to repeat,
- behaviors that result in punishment/pain tend to become extinct.

How are behaviors shaped?



B.F. Skinner
1904-1990
Harvard psychology

https://www.youtube.com/watch?v=yhvaSEJtOV8

How are behaviors shaped?

B.F. Skinner
1904-1990
Harvard psychology

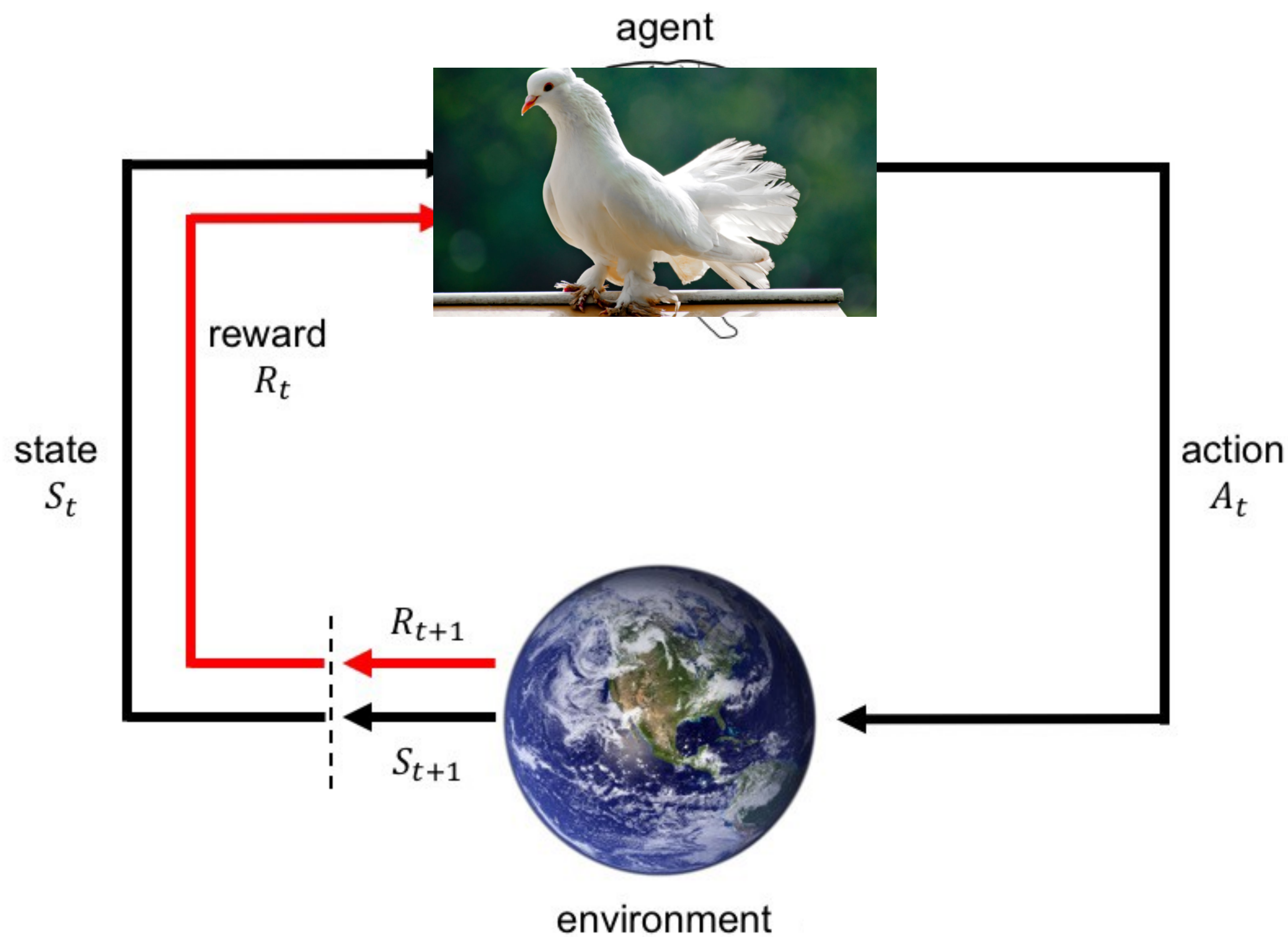
Behavior is primarily shaped by reinforcement rather than free-will.

- behaviors that result in praise/pleasure tend to repeat,
- behaviors that result in punishment/pain tend to become extinct.

Interesting additional finding: Variable ratio reward schedule: Pigeons become addicted to pecking under variable (non-consistent) rewarding

Reinforcement learning = trial-and-error

Learning policies that maximize a reward function by interacting with the world



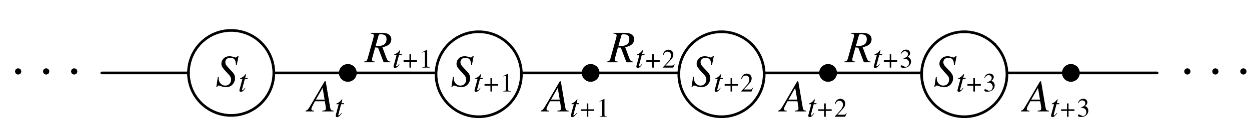
Agent and environment interact at discrete time steps: $t = 0, 1, 2, K$

Agent observes state at step t: $S_t \in \mathcal{S}$

produces action at step t : $A_t \in \mathcal{A}(S_t)$

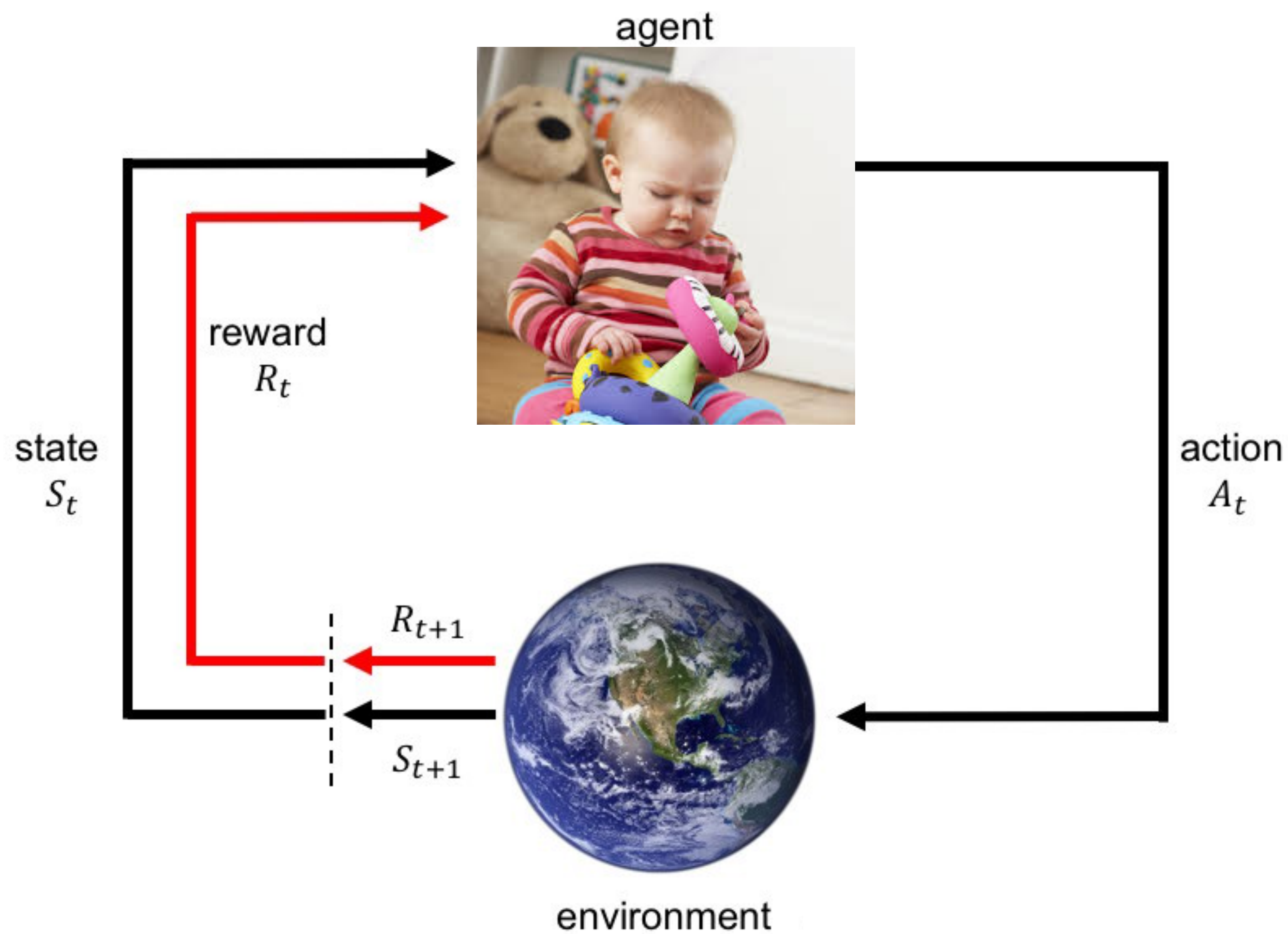
gets resulting reward: $R_{t+1} \in \mathbb{R}$

and resulting next state: $S_{t+1} \in \mathcal{S}^+$



Reinforcement learning

Rewards can be intrinsic, i.e., generated by the agent and guided by its curiosity as opposed to the external environment.



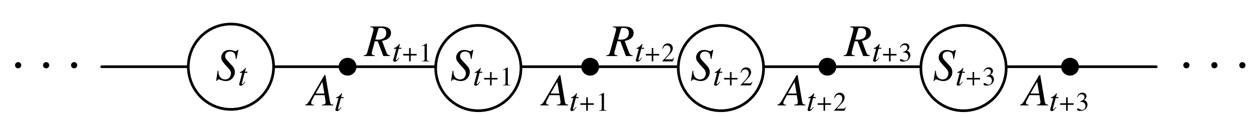
Agent and environment interact at discrete time steps: $t = 0, 1, 2, K$

Agent observes state at step t: $S_t \in \mathcal{S}$

produces action at step t: $A_t \in \mathcal{A}(S_t)$

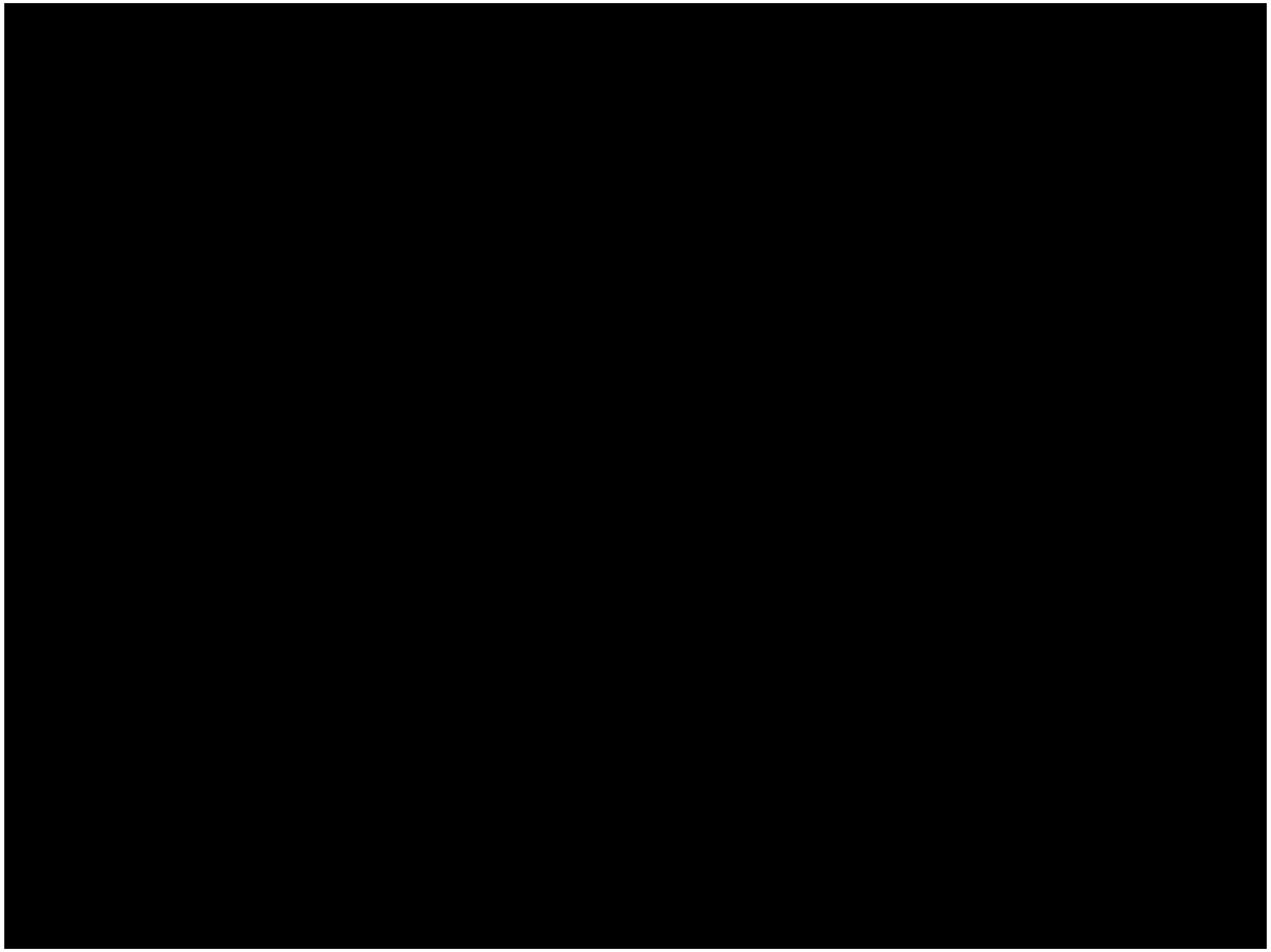
gets resulting reward: $R_{t+1} \in \mathbb{R}$

and resulting next state: $S_{t+1} \in \mathcal{S}^+$



Reinforcement learning

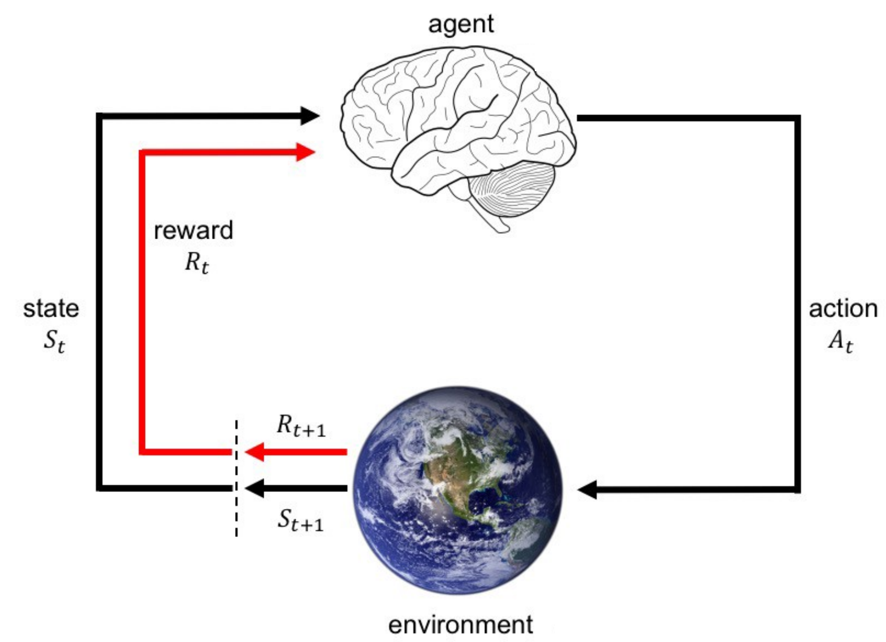
Rewards can be intrinsic, i.e., generated by the agent and guided by its curiosity as opposed to the external environment.



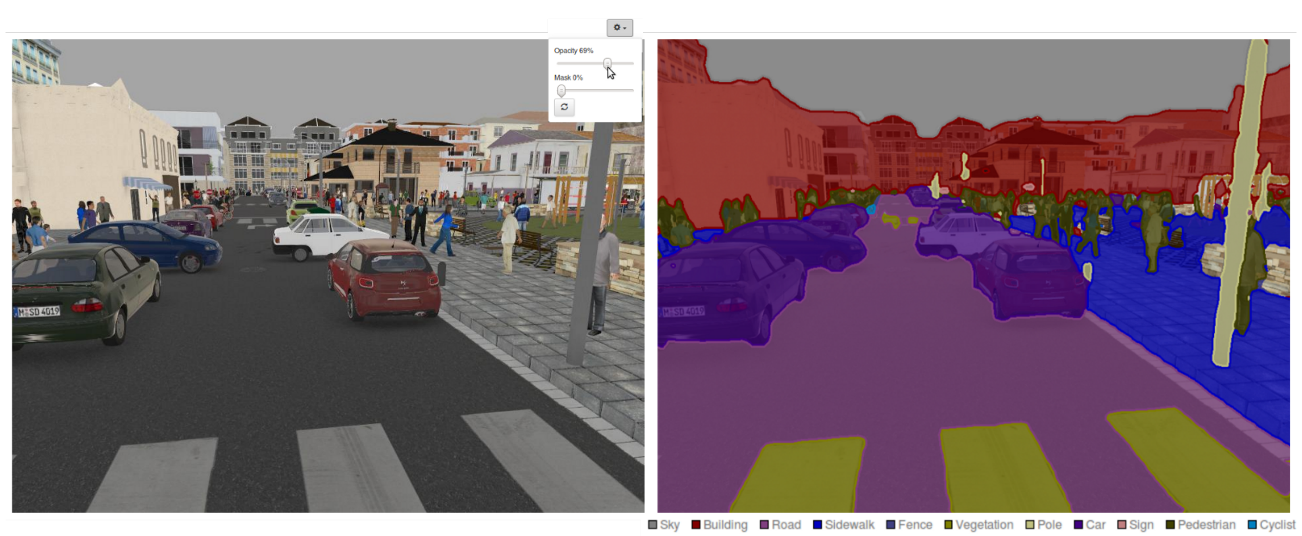
https://youtu.be/8vNxjwt2AqY

Reinforcement learning

- It is considered the most biologically plausible form of learning
- It addresses the full problem of making artificial agents that act in the world, so it is driven by the right end goal

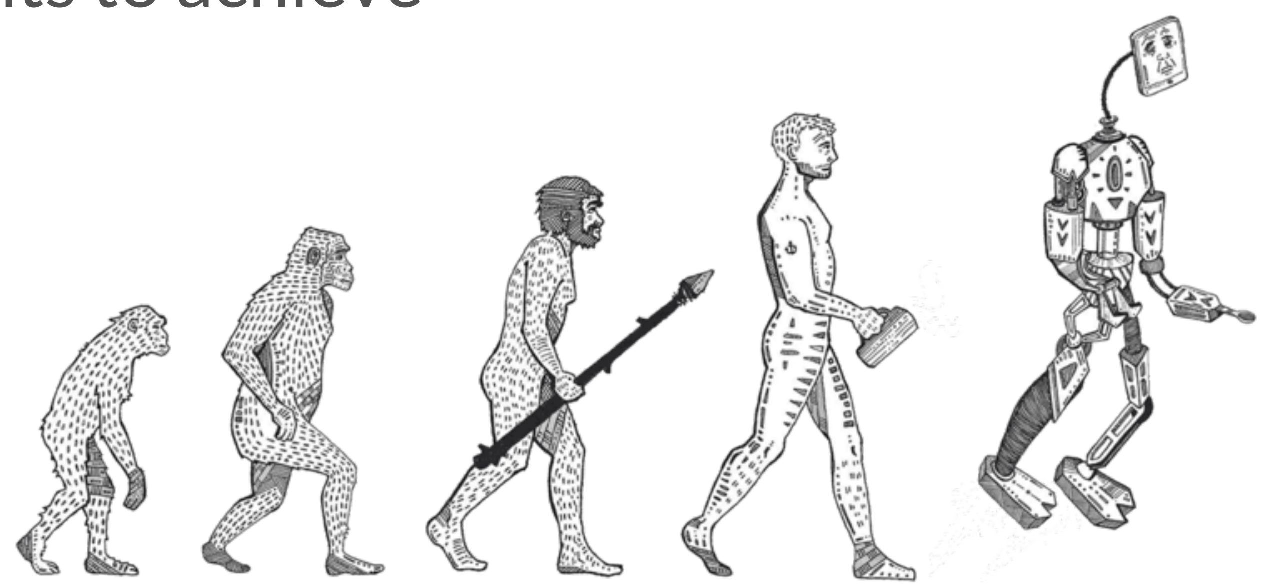


...in contrast to, for example, pixel labelling



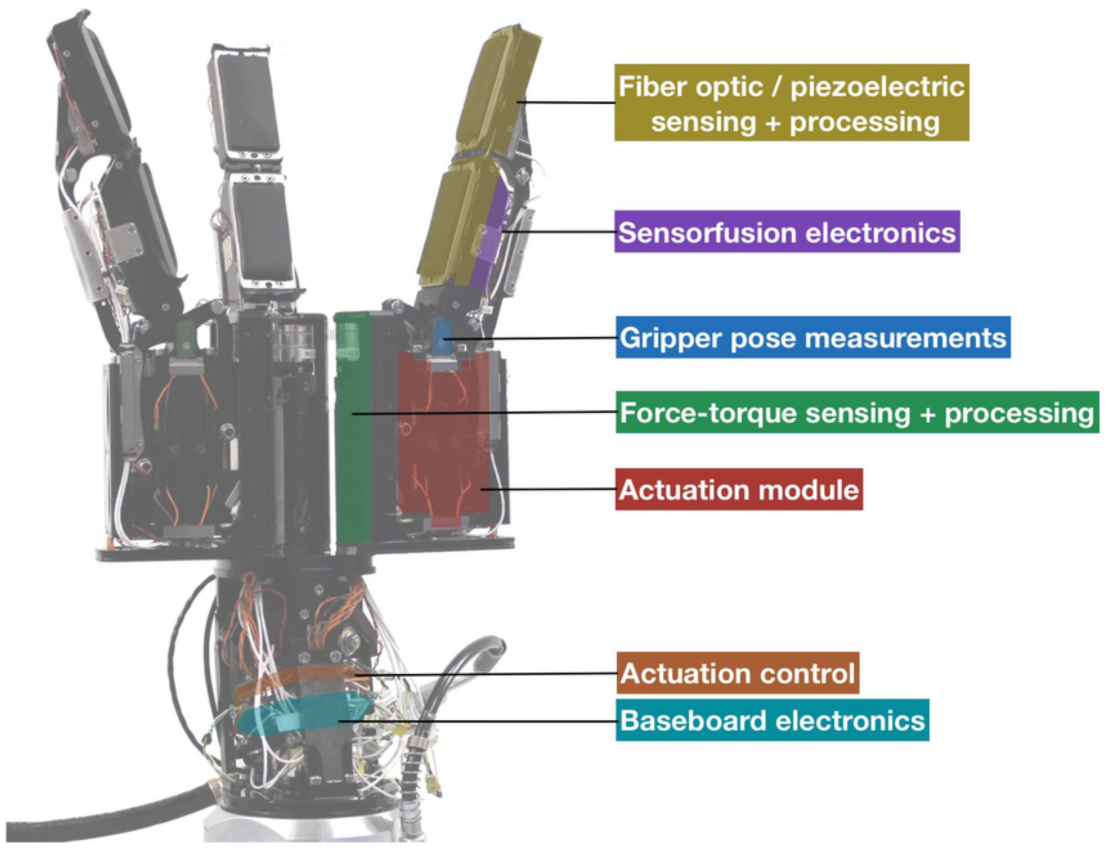
Agent

An entity that is equipped with
- sensors, in order to sense the environment,
- end-effectors in order to act in the environment, and
- goals that she wants to achieve



Actions

They are used by the agent to interact with the world. They can have many different temporal granularities and abstractions.



Actions can be defined to be

- The instantaneous torques applied on the gripper
- The instantaneous gripper translation, rotation, opening
- Instantaneous forces applied to the objects
- Short sequences of the above

State estimation: from observations to states

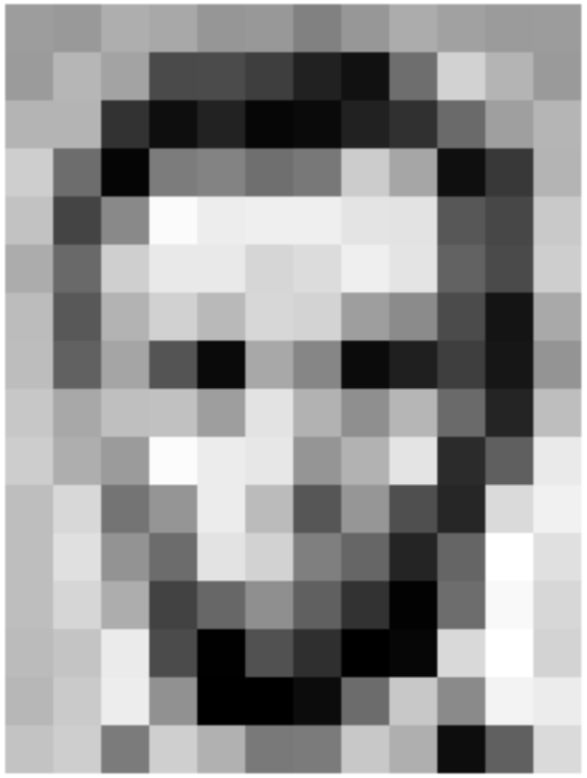
- An observation a.k.a. sensation: the (raw) input of the agent's sensors, images, tactile signal, waveforms, etc.
- A state captures whatever information is available to the agent at step t about its environment.
- The state can include immediate "sensations," highly processed sensations, and structures built up over time from sequences of sensations, memories etc.

Representation learning helps state estimation

- Representation learning: mapping raw sensory input to a feature space from which the mapping to actions or to semantic labels is easier to infer.

Representation learning helps state estimation

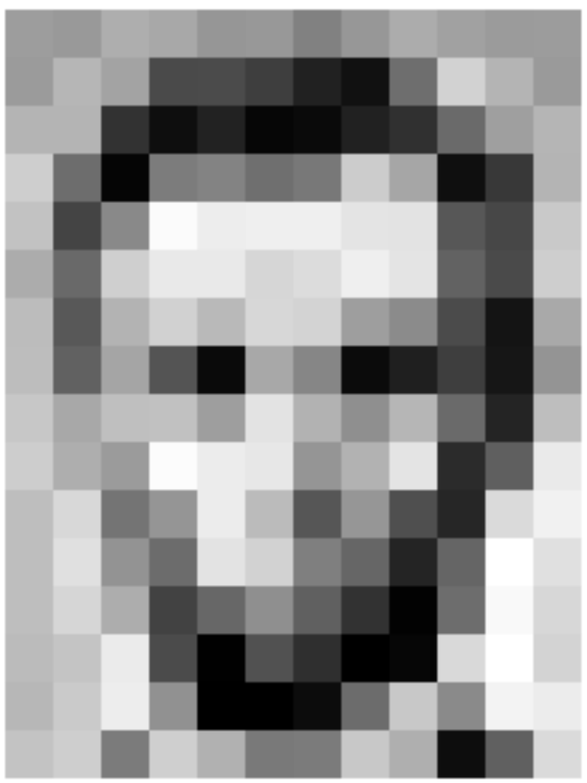
- Representation learning: mapping raw sensory input to a feature space from which the mapping to actions or to semantic labels is easier to infer.



- Remember what the computer sees

Representation learning helps state estimation

- Representation learning: mapping raw sensory input to a feature space from which the mapping to actions or to semantic labels is easier to infer.

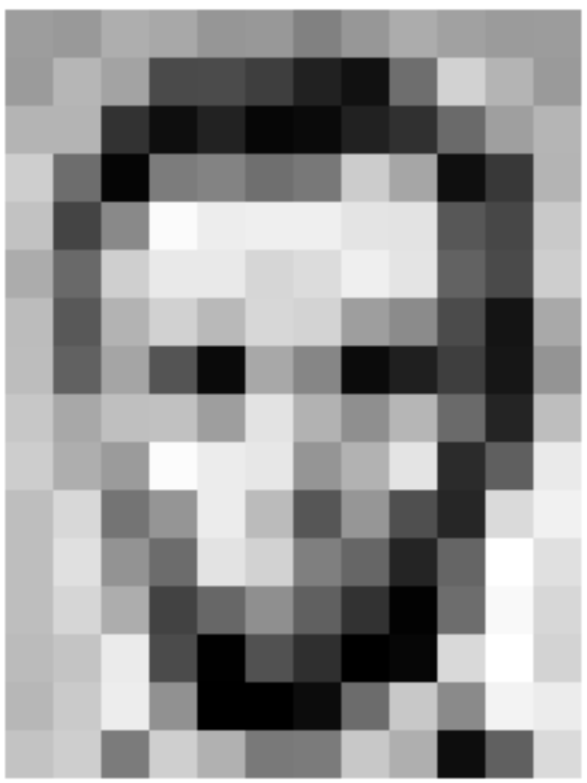


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

- Remember what the computer sees

Representation learning helps state estimation

- Representation learning: mapping raw sensory input to a feature space from which the mapping to actions or to semantic labels is easier to infer.



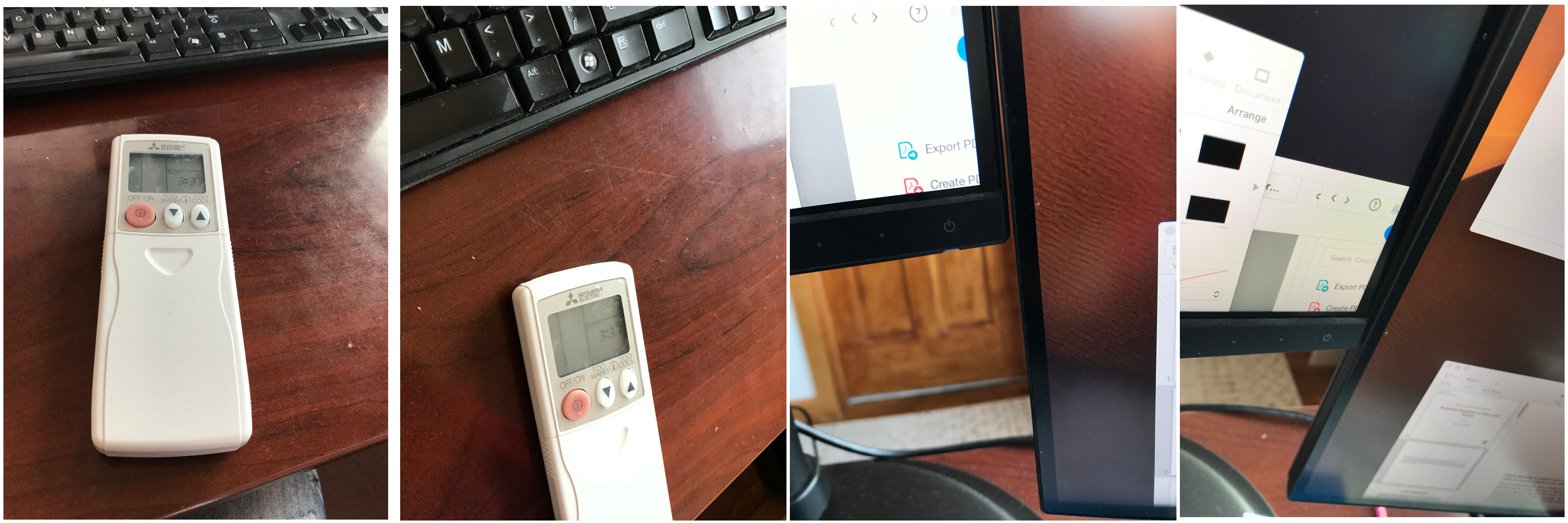
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

- Remember what the computer sees

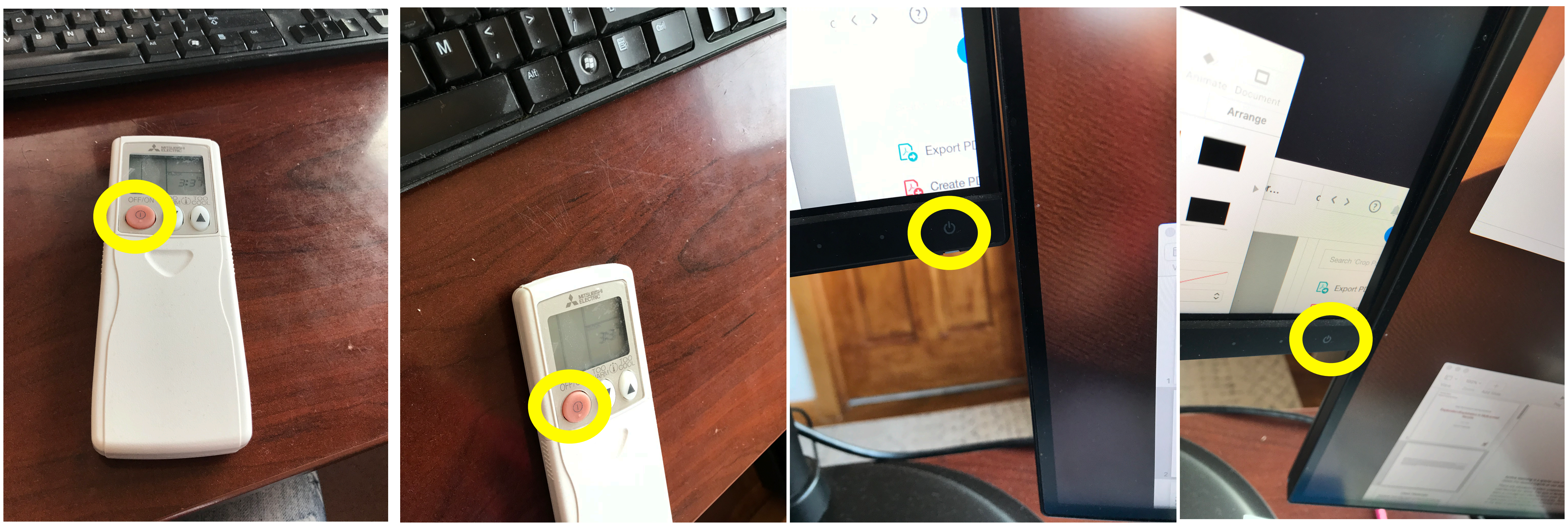
Representation learning helps state estimation

- Representation learning: mapping raw sensory input to a feature space from which the mapping to actions or to semantic labels is easier to infer.



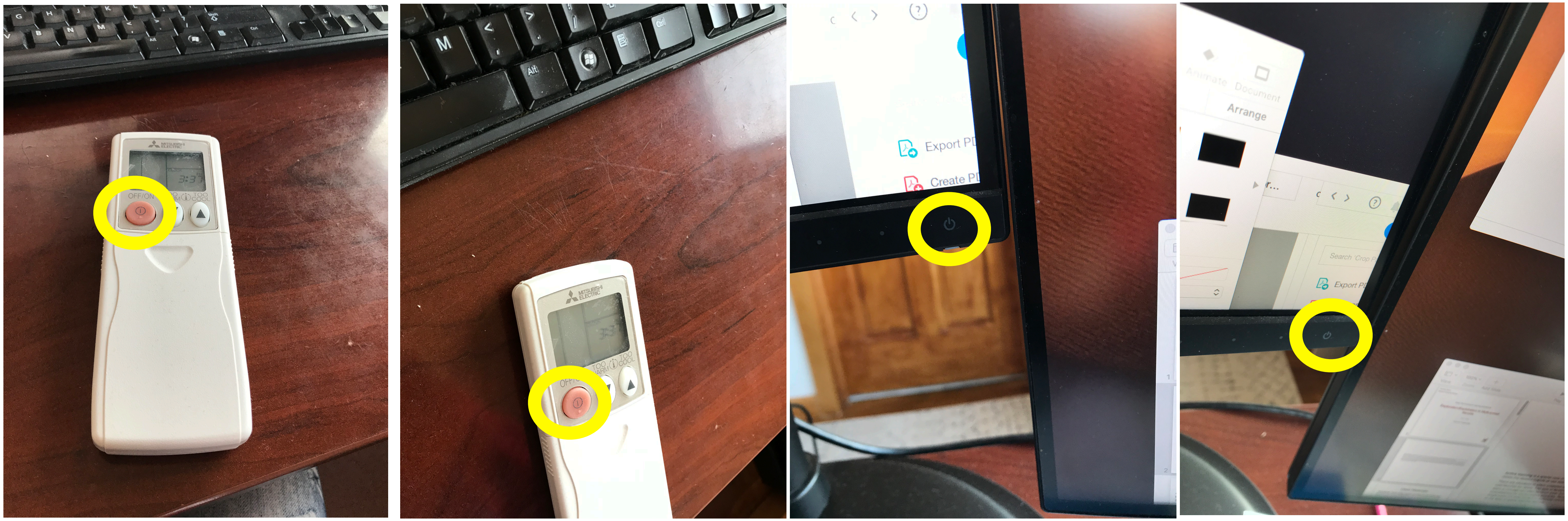
Representation learning helps state estimation

- Representation learning: mapping raw sensory input to a feature space from which the mapping to actions or to semantic labels is easier to infer.



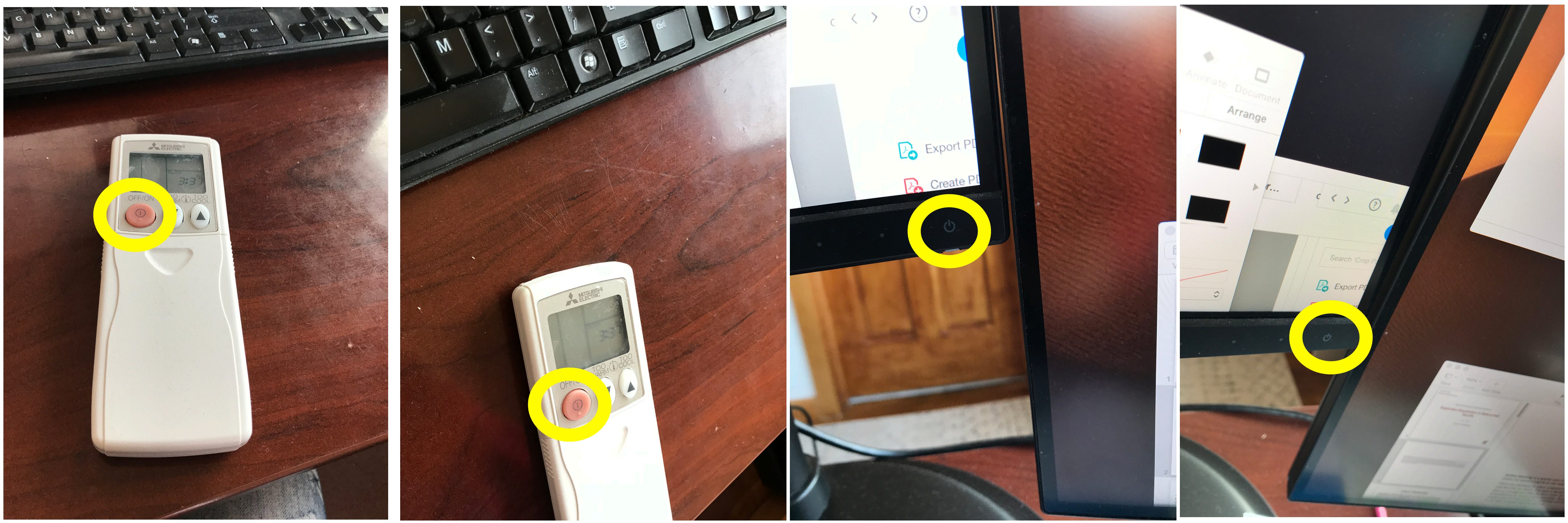
Representation learning helps state estimation

- Despite those images have very different pixel values, actions required to achieve the goal of switching on the device are similar. Visual perception is instrumental to learning to act, in transforming raw pixels to action-relevant feature vectors.



Representation learning helps state estimation

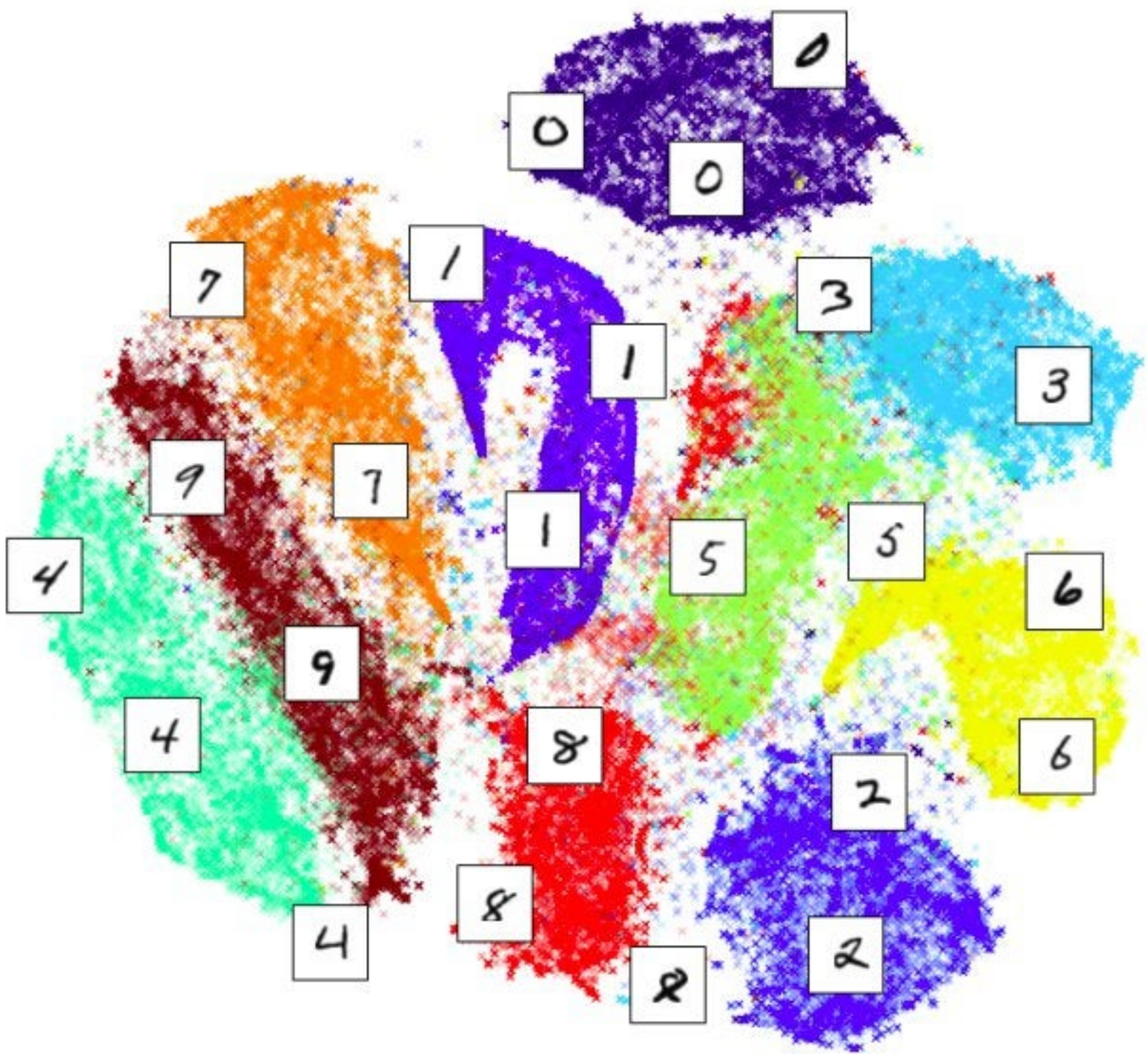
- Having pre-trained our visual representations with auxiliary tasks is likely to dramatically decrease the number of interactions with the environment we need to learn to press buttons.



- Q: What are reasonable auxiliary tasks?
    - Supervised: object detection, image classification, pixel labelling.
    - Unsupervised: open research problem

Representation learning helps state estimation

- Visual representation learning: mapping pixels to features vectors from which the mapping to actions or to semantic labels is easier to infer.



Representation learning helps state estimation

- Visual representation learning: mapping pixels to features vectors from which the mapping to actions or to semantic labels is easier to infer.

Representation learning helps state estimation

- Visual representation learning: mapping pixels to features vectors from which the mapping to actions or to semantic labels is easier to infer.

Policy

A mapping function from states to actions of the end effectors.

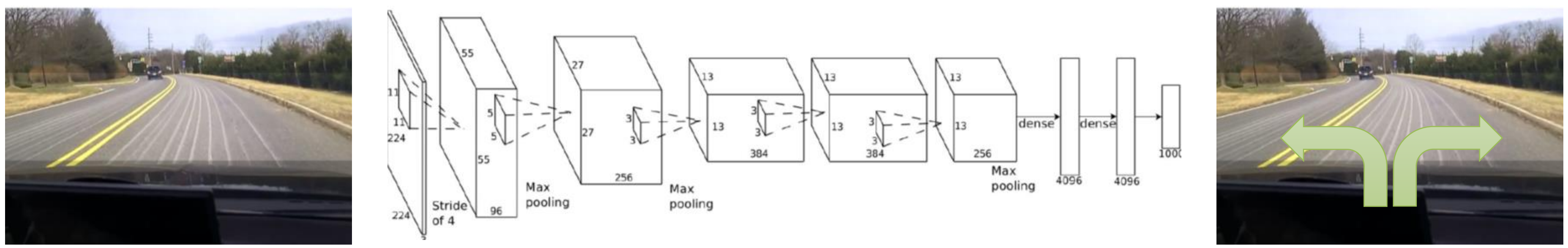
$$\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$$

Policy

A mapping function from states to actions of the end effectors.

$$\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$$

It can be a shallow or a deep function mapping

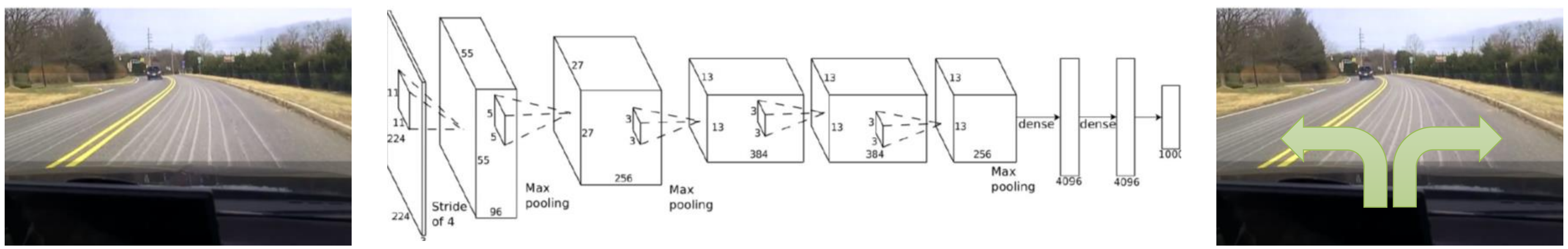


Policy

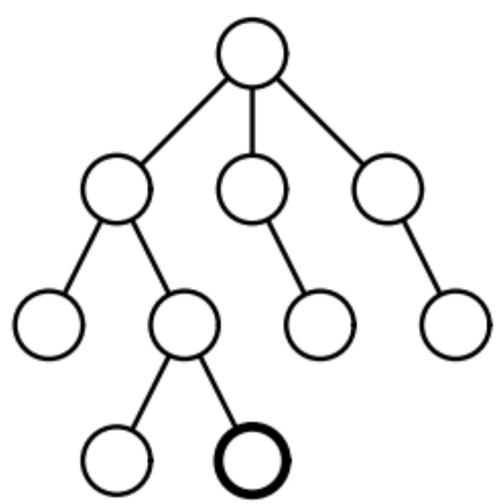
A mapping function from states to actions of the end effectors.

$$\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$$

It can be a shallow or a deep function mapping



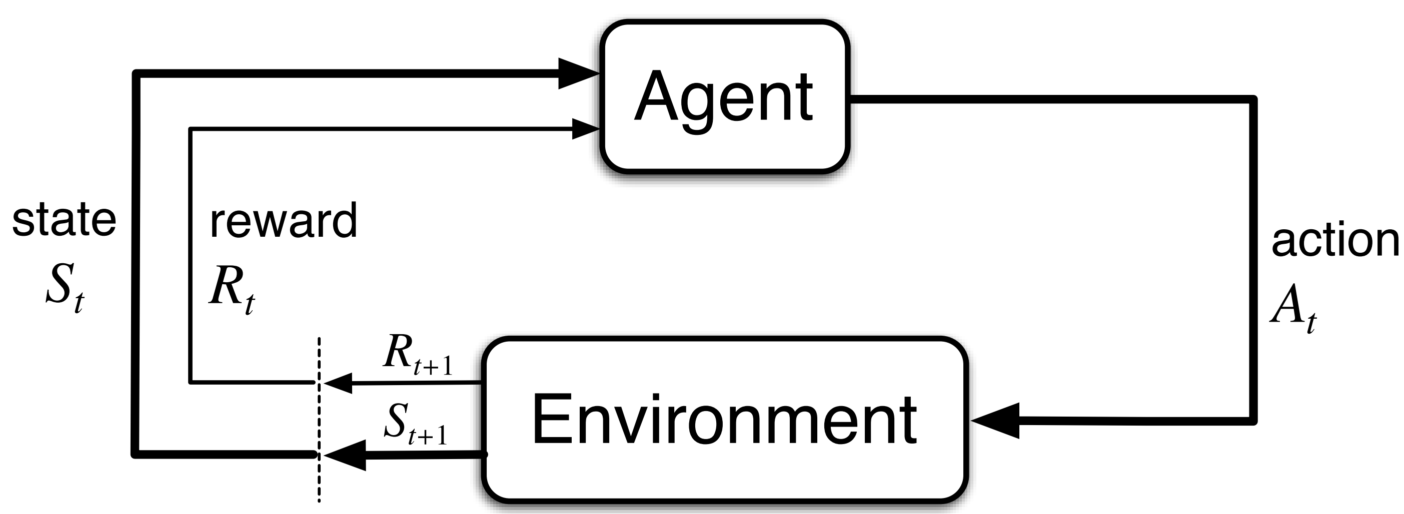
or it can be as complicated as involving a tree look-ahead search



Closed loop sensing and acting

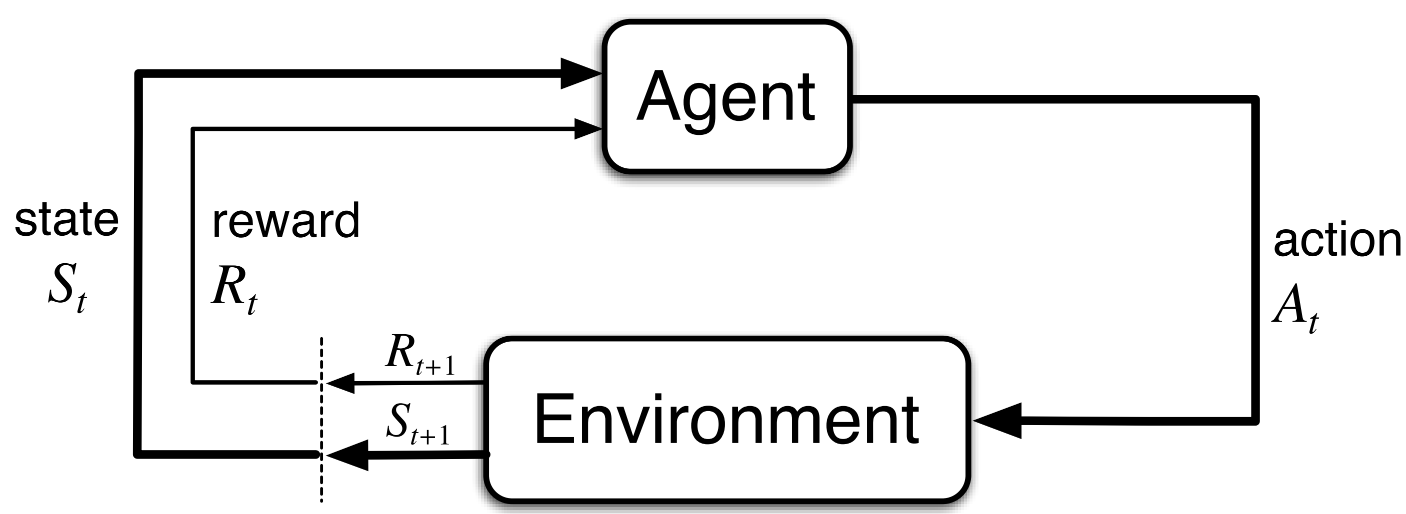
Imagine an agent that wants to pick up an object and has a policy that predicts what the actions should be for the next 2 secs ahead.

This means, for the next 2 secs we switch off the sensors, and just execute the predicted actions. In the next second, due to imperfect sensing, the object is about to fall!



Closed loop sensing and acting

Sensing is always imperfect. Our excellent motor skills are due to continuous sensing and updating of the actions, a.k.a. servoing. So the perception-action loop is in fact extremely short in time.



Rewards

They are scalar values provided provided to the agent that indicate whether goals have been achieved, e.g., 1 if goal is achieved, 0 otherwise, or -1 for overtime step the goal is not achieved

- Rewards specify what the agent needs to achieve, not how to achieve it.
- The simplest and cheapest form of supervision, and surprisingly general: All of what we mean by goals and purposes can be encoded mathematically as the maximization of the cumulative sum of a received scalar signal (reward)

Returns

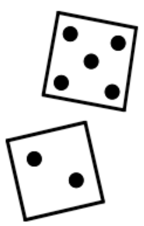
Goal-seeking behavior of an agent can be formalized as the behavior that seeks maximization of the expected value of the cumulative sum of (potentially time discounted) rewards, we call it return.

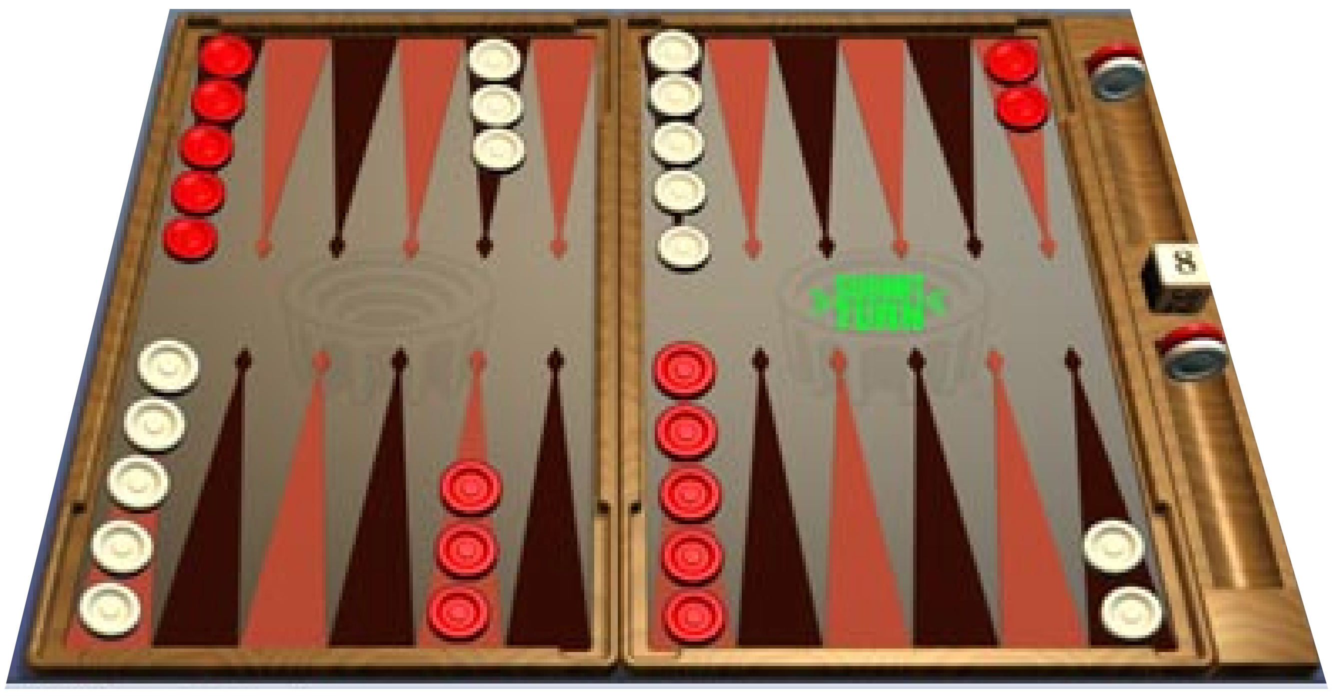
We want to maximize returns.

$$G_t = R_{t+1} + R_{t+2} + \cdots + R_T$$

Example: Backgammon

- States: Configurations of the playing board (≈1020)
- Actions: Moves
- Rewards:
  - win: +1
  - lose: –1
  - else: 0



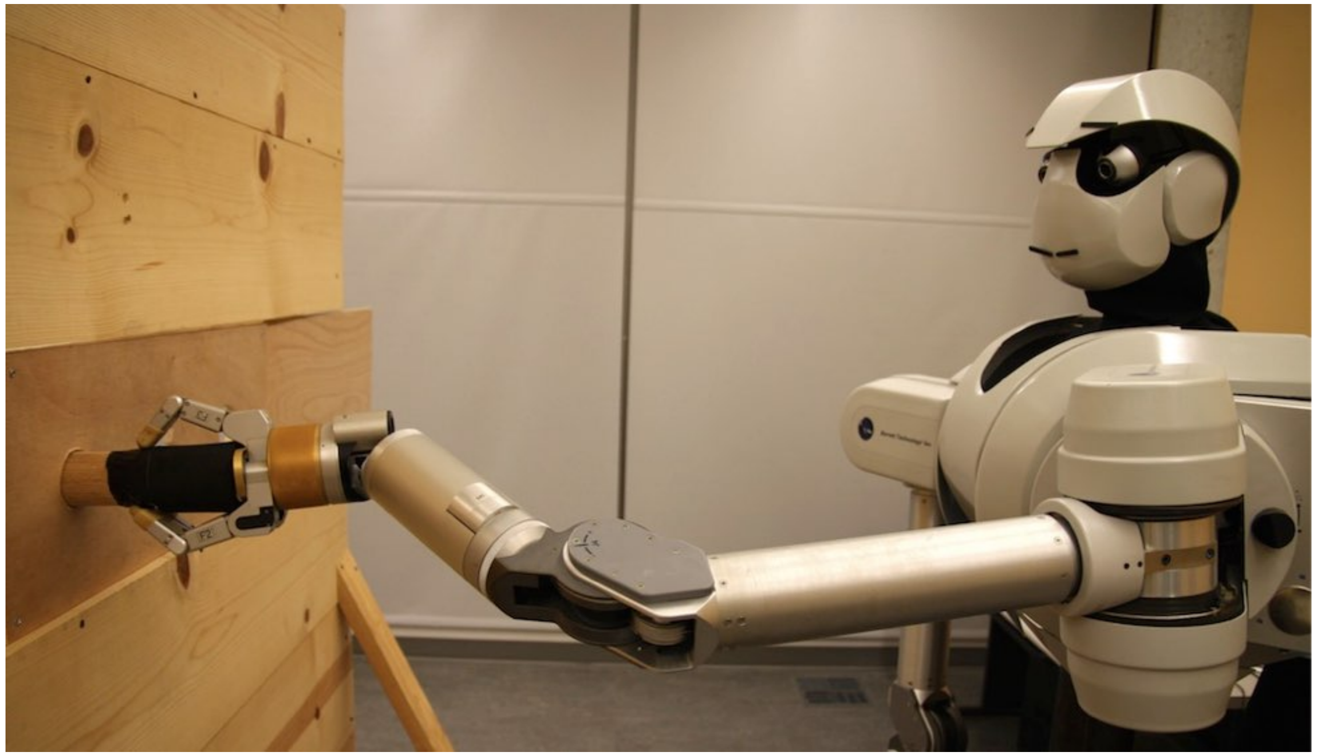


Example: Driving

- States: Road traffic, weather, time of day
- Actions: steering wheel, break
- Rewards:
  - +1 reaching goal not over-tired
  - -1: honking from surrounding drivers
  - -100: collision

Example: Peg in Hole Insertion

- States: Joint configurations (7DOF)
- Actions: Torques on joints
- Rewards: Penalize jerky motions, reaching target pose
- Q: why we do not simply use Euclidean distances between current and target poses/locations?



Dynamics a.k.a. the World Model

- Encodes the results of the actions of the agent.

- How the states and rewards change given the actions of the agent:

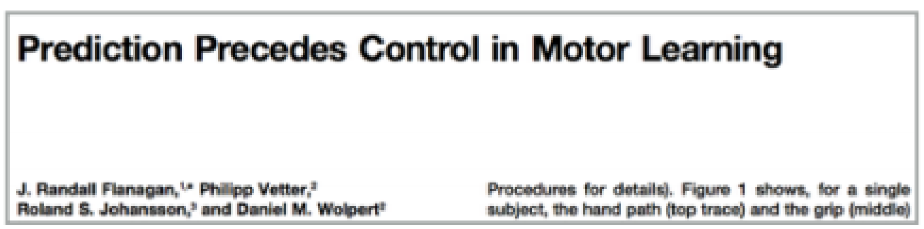
$$\mathrm{p}(s', r \mid s, a) = \mathbb{P}\{S_t = s', \mathrm{R_t} = r \mid \mathrm{S}_{t-1} = s, \mathrm{A}_{t-1} = a\}$$

- Transition function or next step function:

$$\mathrm{T}(s' \mid s, a) = \mathrm{p}(s' \mid s, a) = \mathbb{P}\{S_t = s' \mid \mathrm{S}_{t-1} = s, \mathrm{A}_{t-1} = a\} = \sum_{r \in \mathbb{R}} \mathrm{p}(s', r \mid s, a)$$

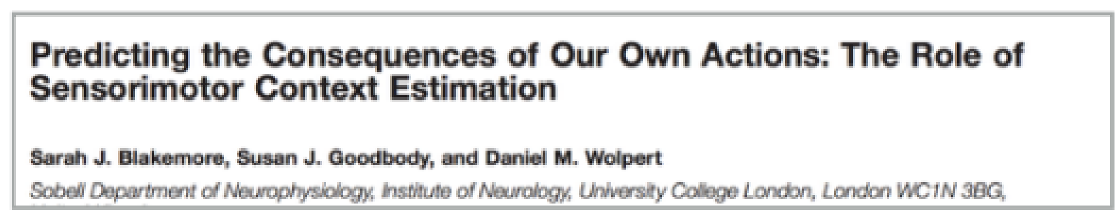
Dynamics a.k.a. the World Model

"the idea that we predict the consequences of our motor commands has emerged as an important theoretical concept in all aspects of sensorimotor control"


Prediction Precedes Control in Motor Learning

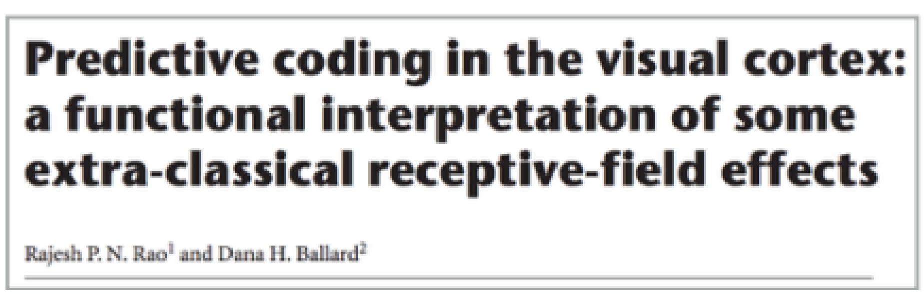
J. Randall Flanagan,[1*] Philipp Vetter,[2]
Roland S. Johansson,[3] and Daniel M. Wolpert[2]

Procedures for details). Figure 1 shows, for a single subject, the hand path (top trace) and the grip (middle)


Predicting the Consequences of Our Own Actions: The Role of Sensorimotor Context Estimation

Sarah J. Blakemore, Susan J. Goodbody, and Daniel M. Wolpert

Sobell Department of Neurophysiology, Institute of Neurology, University College London, London WC1N 3BG,


Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects

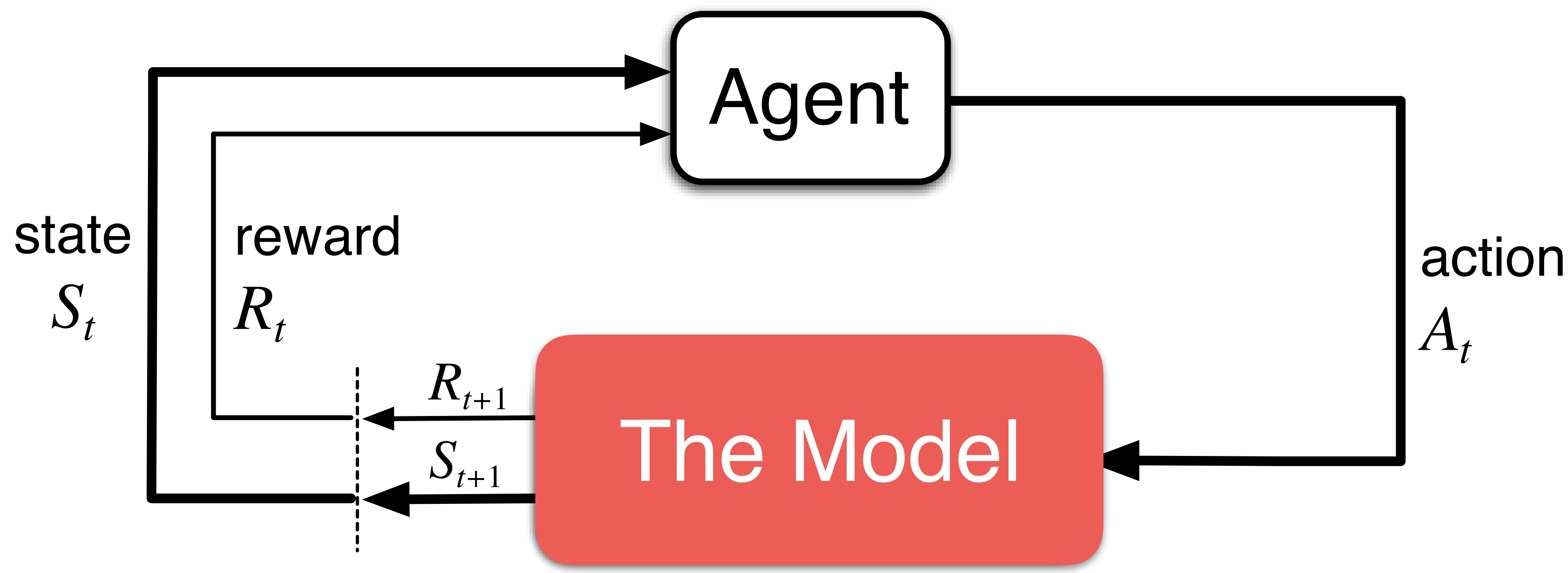
Rajesh P. N. Rao[1] and Dana H. Ballard[2]

slide borrowed from Sergey

Planning

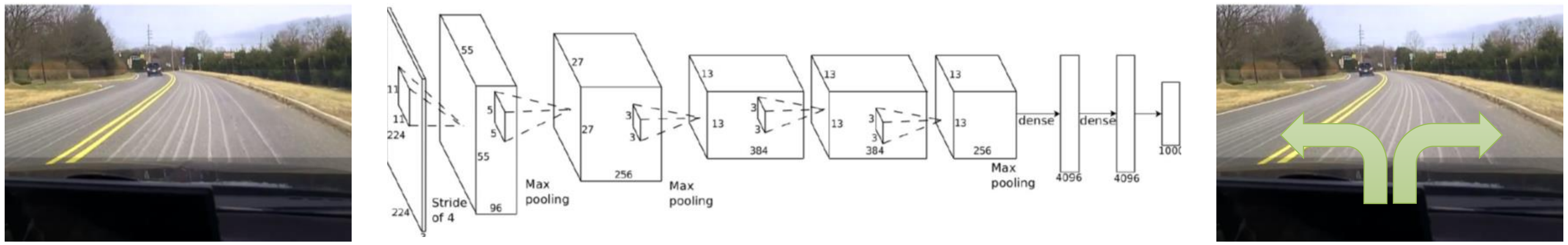
Planning: unrolling (querying) a model forward in time and selecting the best action sequence that satisfies a specific goal

Plan: a sequence of actions



Why deep reinforcement learning?

Because the policy, the model and the value functions (expected returns) will often be represented by some form of a deep neural network.



Limitations of Learning by Interaction

- Can we think of goal directed behavior learning problems that cannot be modeled or are not meaningful using the trial-and-error reinforcement learning framework?
- The agent should have the chance to try (and fail) enough times
- This is impossible if episode takes too long, e.g., reward="obtain a great Ph.D."
- This is impossible when safety is a concern: we can't learn to drive via reinforcement learning in the real world, failure cannot be tolerated

Q: what other forms of supervision humans use to learn to act in the world?

Other forms of supervision for learning behaviours?

1. Learning from rewards

2. Learning from demonstrations

3. Learning from specifications of optimal behavior

Behavior: High Jump

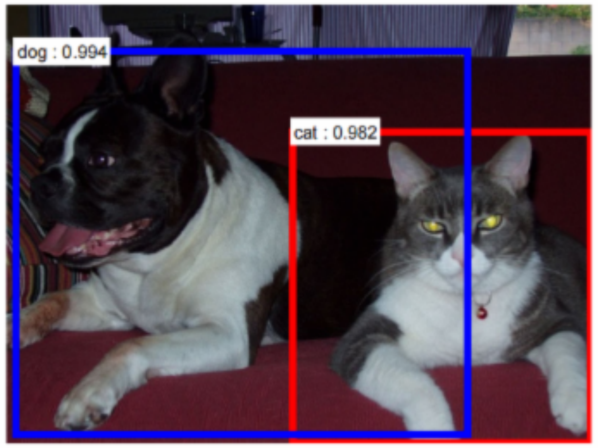
scissors

Fosbury flop

- Learning from rewards
  - Reward: jump as high as possible: It took years for athletes to find the right behavior to achieve this
- Learning from demonstrations
  - It was way easier for athletes to perfection the jump, once someone showed the right general trajectory
- Learning from specifications of optimal behavior
  - For novices, it is much easier to replicate a behavior if additional guidance is provided in natural language: where to place the foot, how to time yourself, etc. .

Reinforcement learning Versus supervised learning

- RL is a form of active learning:
  - the agent gets the chance to collect her own data by acting in the world, querying humans, and so on.
  - the data changes over time, it depends on the policy of the agent.
- Supervised learning is a form of passive learning:
  - the data does not depend on the agent in anyway, it is provided by external labellers.
  - the data is static throughout learning.



Reinforcement learning Versus supervised learning

- In RL, we often cannot use gradient-based optimization:
  - the agent does not know neither the model to unroll nor the reward function to maximize.
- In supervised learning, we usually can use gradient-based optimization:
  - E.g., we consider a parametric form for our regressor or classifier and optimize it via stochastic gradient descent (SGD).

Reinforcement learning Versus supervised learning

- RL is a form of active learning:
  - the agent gets the chance to collect her own data by acting in the world, querying humans, and so on.
  - the data changes over time, it depends on the policy of the agent.

Reinforcement learning Versus supervised learning

- RL is a form of active learning:
  - the agent gets the chance to collect her own data by acting in the world, querying humans, and so on.
  - the data changes over time, it depends on the policy of the agent.
  - To query the environment effectively, the agent needs to keep track of its uncertainty: what she knows and what she does not, and thus needs to explore next.

Reinforcement learning Versus supervised learning

- RL is a form of active learning:
  - the agent gets the chance to collect her own data by acting in the world, querying humans, and so on.
  - the data changes over time, it depends on the policy of the agent.
  - To query the environment effectively, the agent needs to keep track of its uncertainty: what she knows and what she does not, and thus needs to explore next.
  - We will explore many computational tools for representing uncertainty about how good/bad a particular state is (state value), how the model behaves, and so on.

Reinforcement learning Versus supervised learning

- In RL, we often cannot use gradient-based optimization:
  - the agent does not know neither the model to unroll nor the reward function to maximize.

Reinforcement learning Versus supervised learning

- In RL, we often cannot use gradient-based optimization:
  - the agent does not know neither the model to unroll nor the reward function to maximize.
  - We will explore many gradient estimators and gradient-free optimization methods for learning behaviours

Goal: Minimizing interactions while achieving our goal

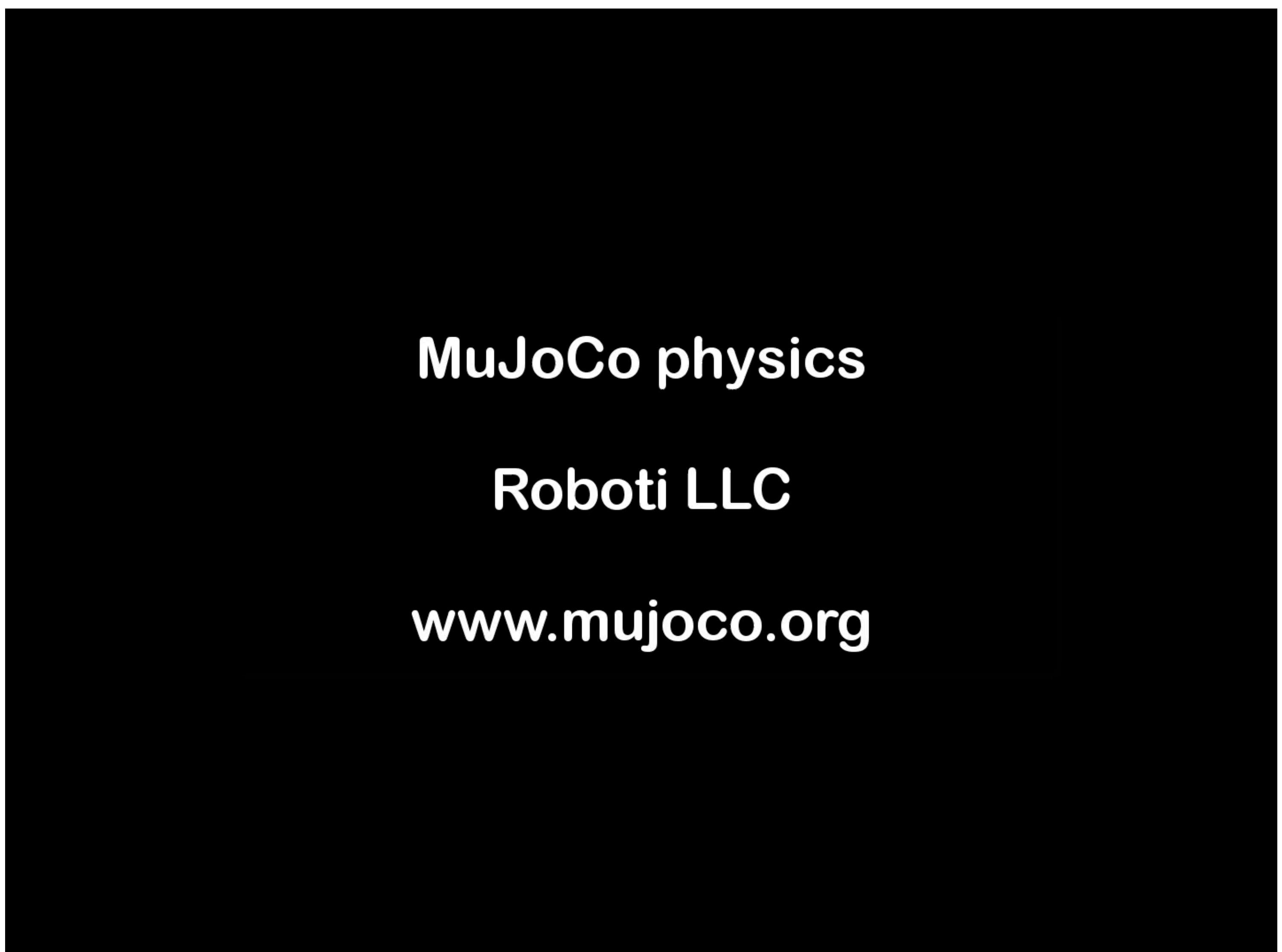
- The agent collects her own data. Actions take time to carry out in the real world, i.e., each interaction has a non-negligible cost. Our goal is the agent to minimize the amount of interactions with the environment while succeeding in the task.

Goal: Minimizing interactions while achieving our goal

- The agent collects her own data. Actions take time to carry out in the real world, i.e., each interaction has a non-negligible cost. Our goal is the agent to minimize the amount of interactions with the environment while succeeding in the task.
- We can use simulated experience and tackle the sim2real transfer

Goal: Minimizing interactions while achieving our goal

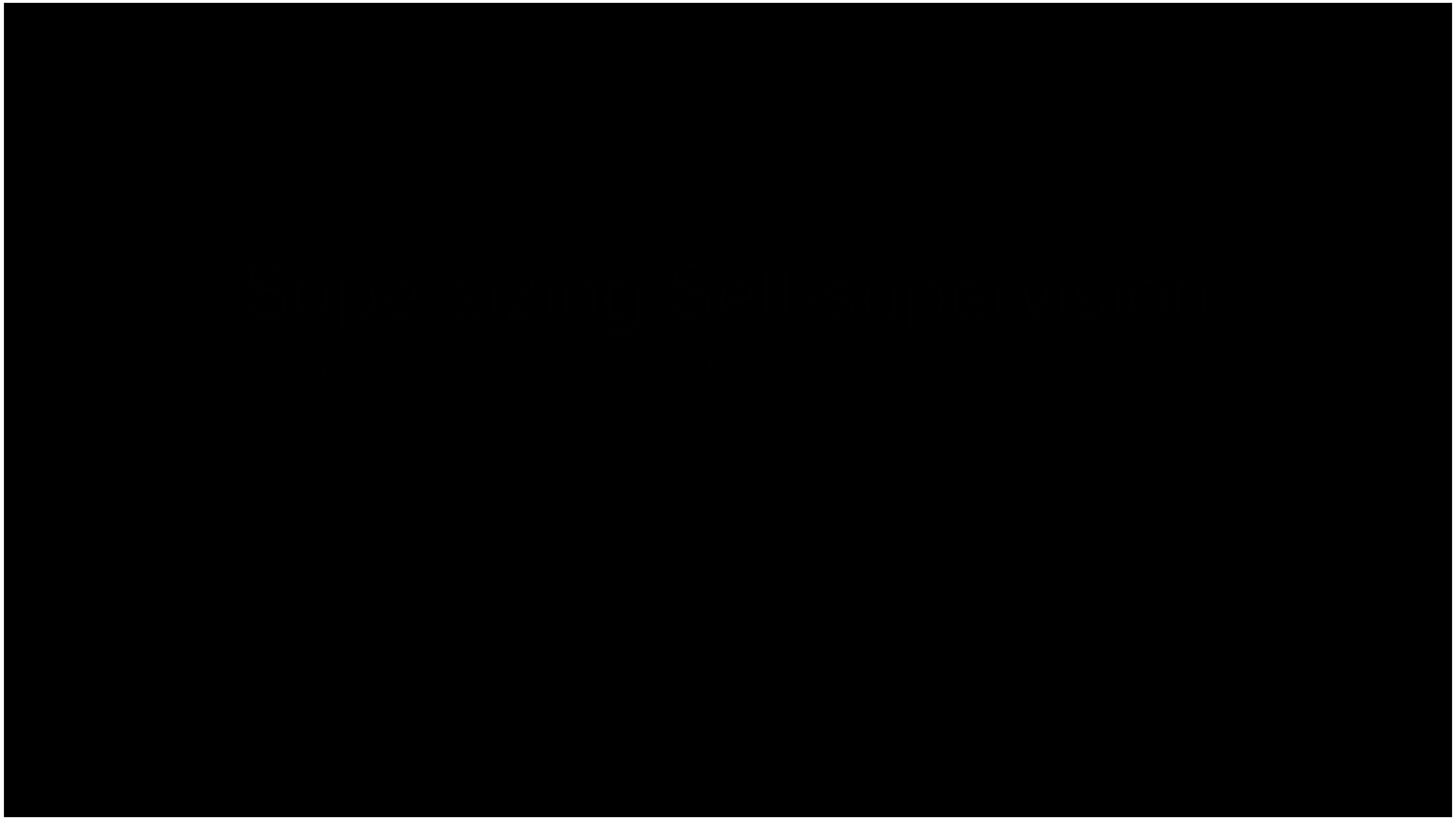
- The agent collects her own data. Actions take time to carry out in the real world, i.e., each query has a non-negligible cost. Our goal is the agent to minimize the amount of interactions with the environment while succeeding in the task.
- We can use simulated experience and tackle the sim2real transfer



Goal: Minimizing interactions while achieving our goal

- The agent collects her own data. Actions take time to carry out in the real world, i.e., each query has a non-negligible cost. Our goal is the agent to minimize the amount of interactions with the environment while succeeding in the task.
- We can use simulated experience and tackle the sim2real transfer
- We can have robots working 24/7

Supersizing Self-Supervision



Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours, Pinto and Gupta

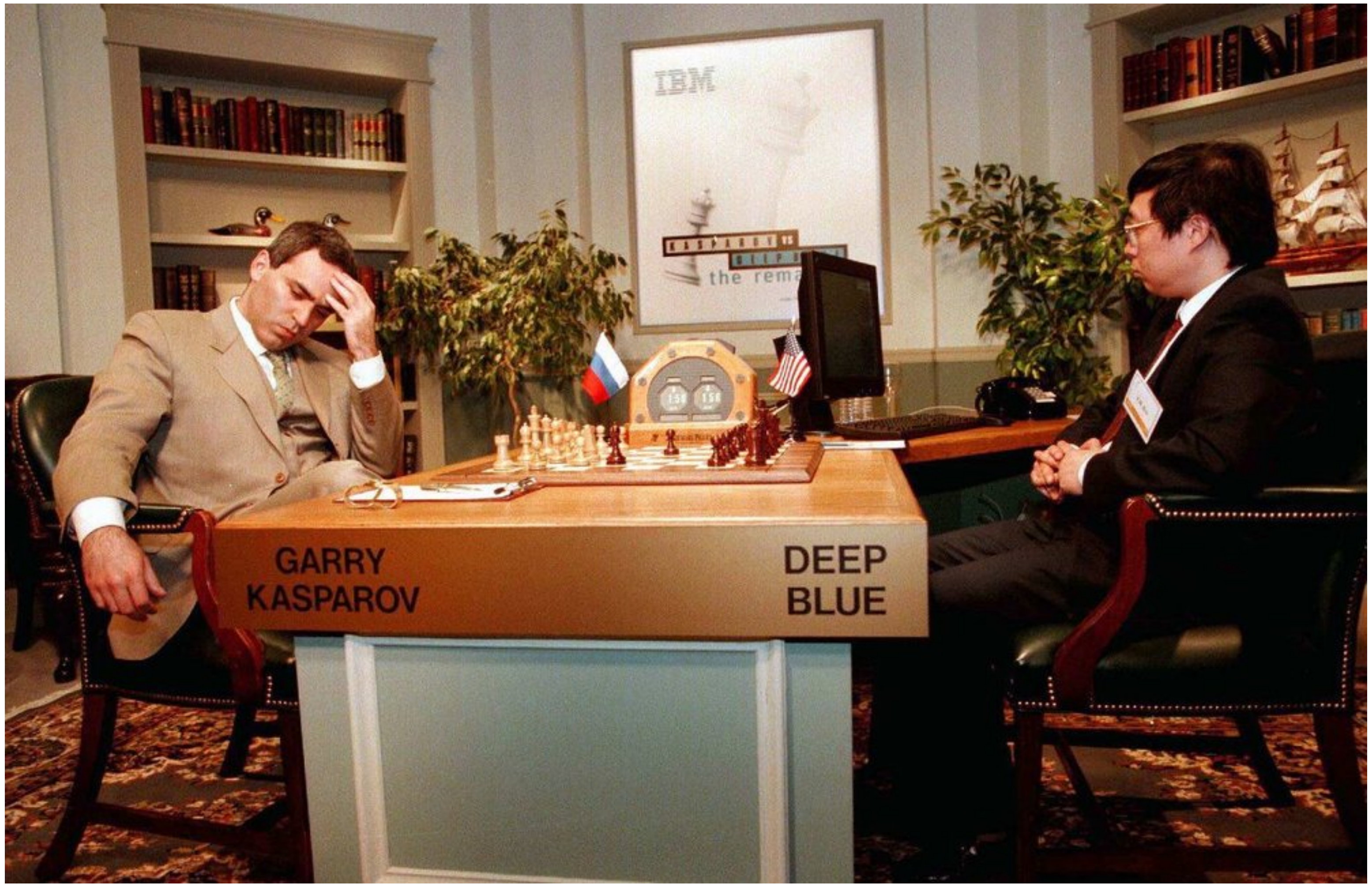
Reinforcement learning Versus supervised learning

- The agent collects her own data. Actions take time to carry out in the real world, i.e., each query has a non-negligible cost. Our goal is the agent to minimize the amount of interactions with the environment while succeeding in the task.
- We can use simulated experience and tackle the sim2real transfer
- We can have robots working 24/7
- We can buy many robots

Google's Robot Farm

Some successes so far

Deep Blue



Q1: Is this a machine learning achievement?

Q2: What is machine learning / artificial intelligence?

A2: The discipline that develops agents that learn and improve with experience (Tom Mitchell)

A1: No, it is not. Brute-force manual development of a board evaluation function

Backgammon

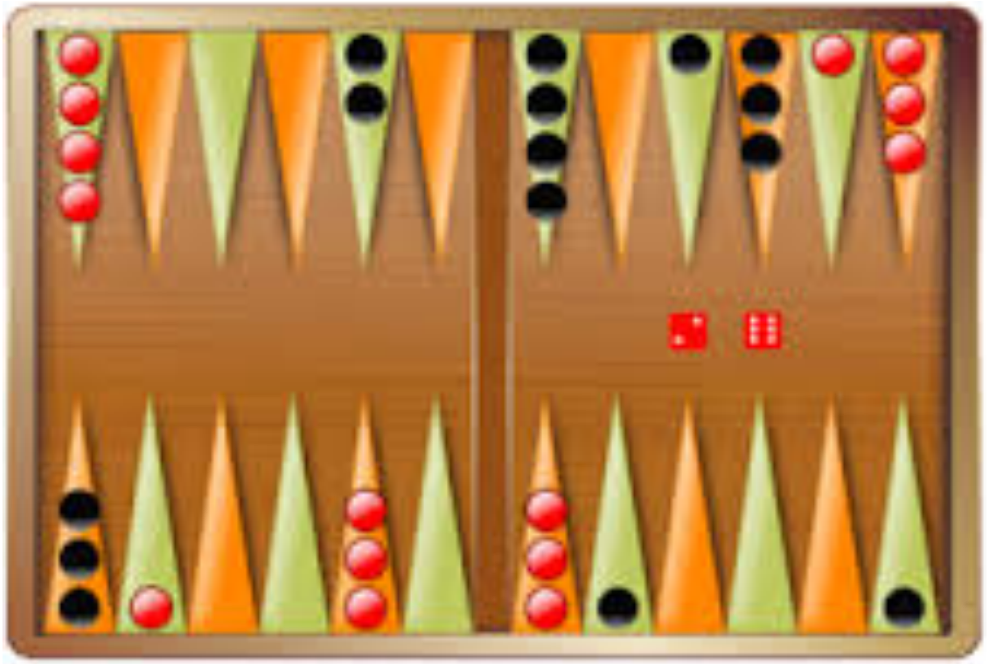
Backgammon

How is it different than chess?

Backgammon

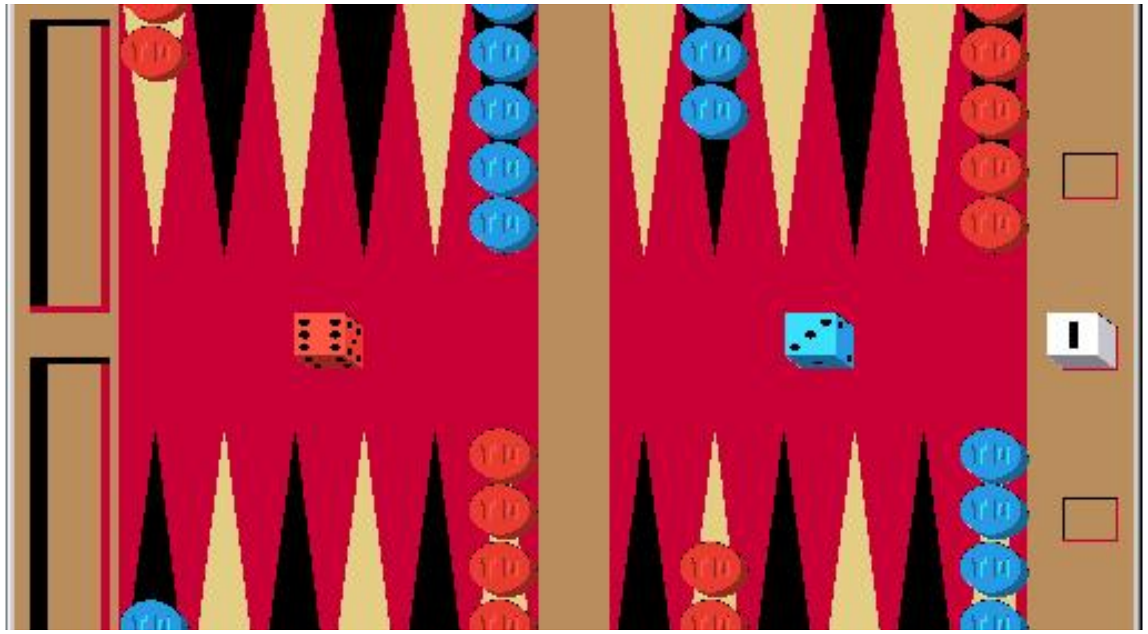
High branching factor due to dice roll prohibits brute force deep searches such as in chess

Neuro-Gammon



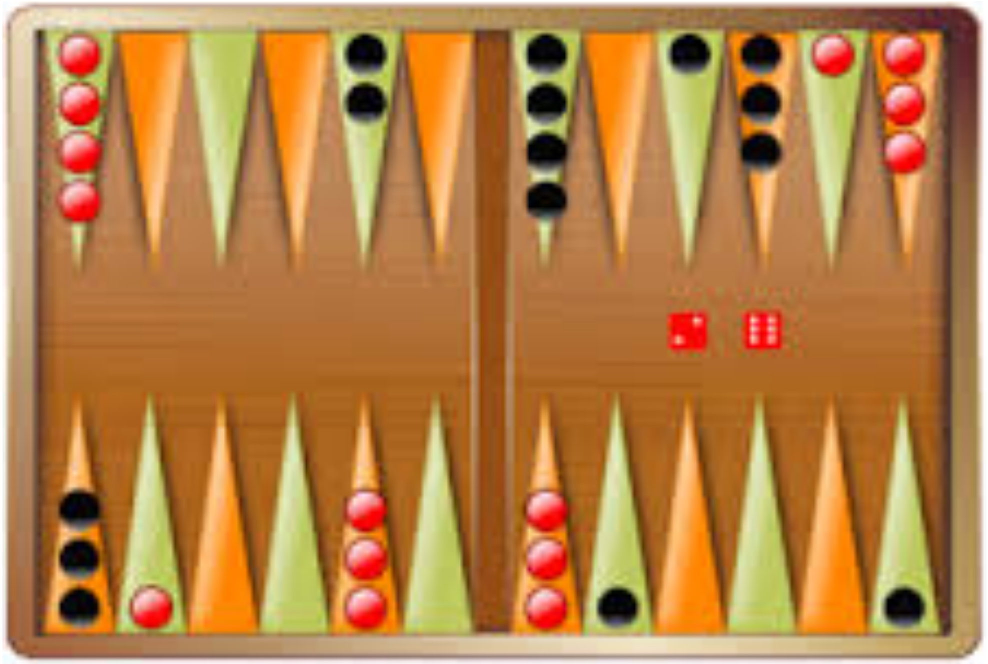
- Developed by Gerald Tesauro in 1989 in IBM’s research center
- Trained to mimic expert demonstrations using supervised learning
- Achieved intermediate-level human player

TD-Gammon



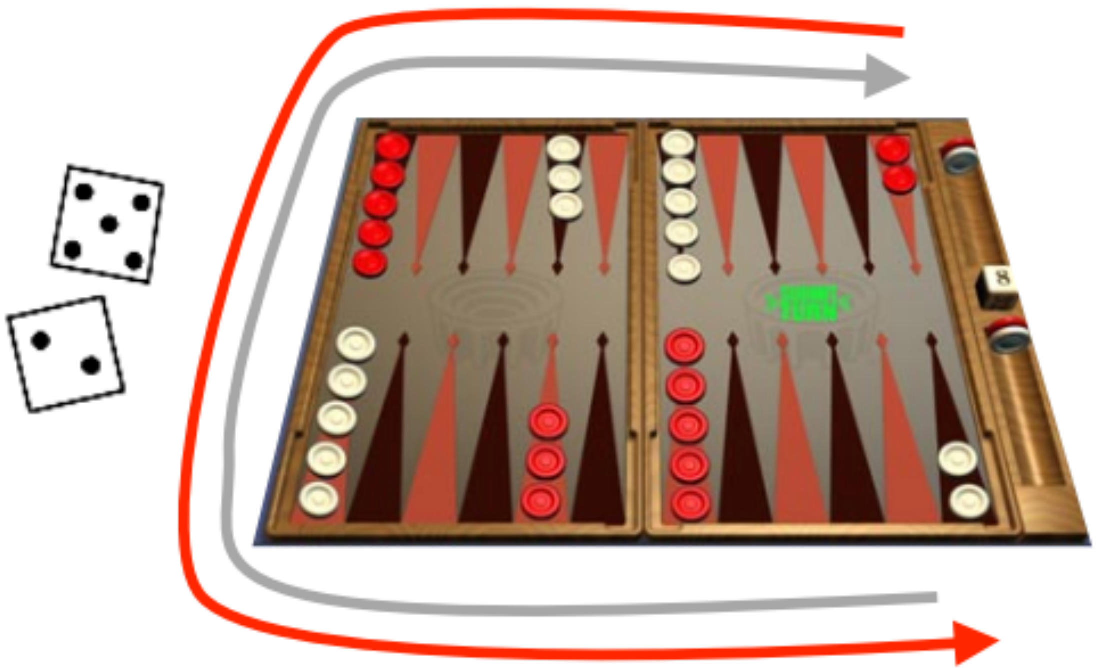
- Developed by Gerald Tesauro in 1992 in IBM's research center
- A neural network that trains itself to be an evaluation function by playing against itself starting from random weights
- Achieved performance close to top human players of its time

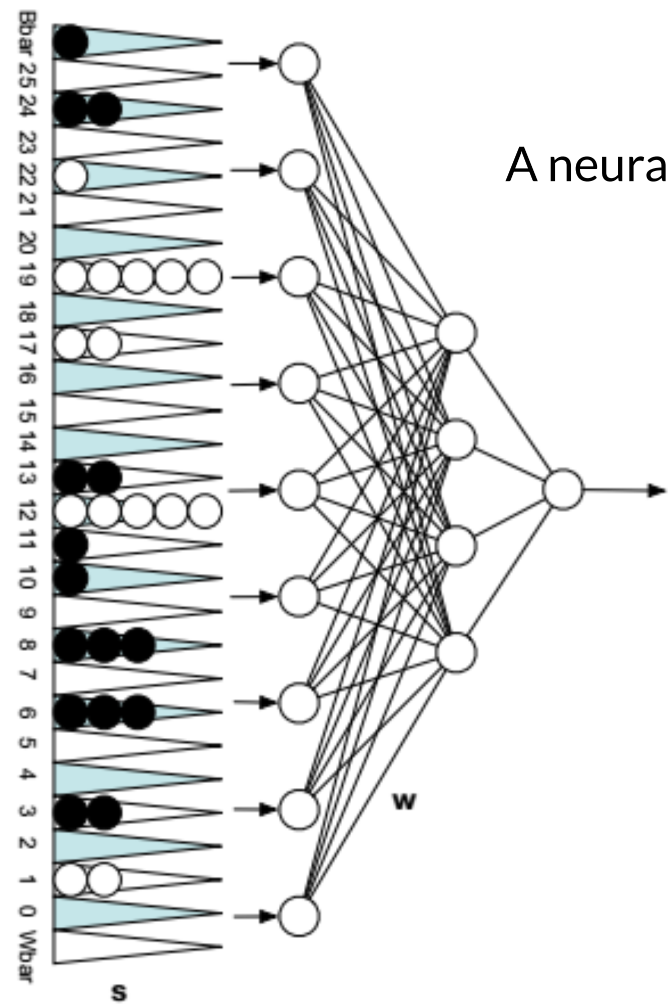
Neuro-Gammon



- Developed by Gerald Tesauro in 1989 in IBM's research center
- Trained to mimic expert demonstrations using supervised learning
- Achieved intermediate-level human player

Evaluation function

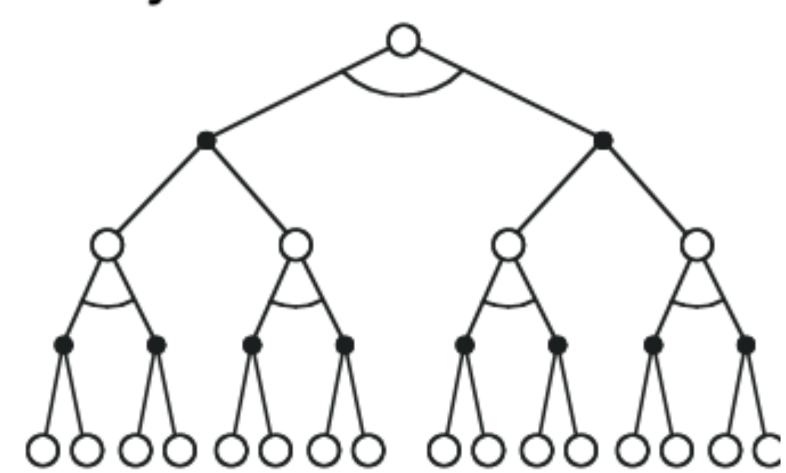




A neural net with only 80 hidden units..

estimated state value
(≈ prob of winning)

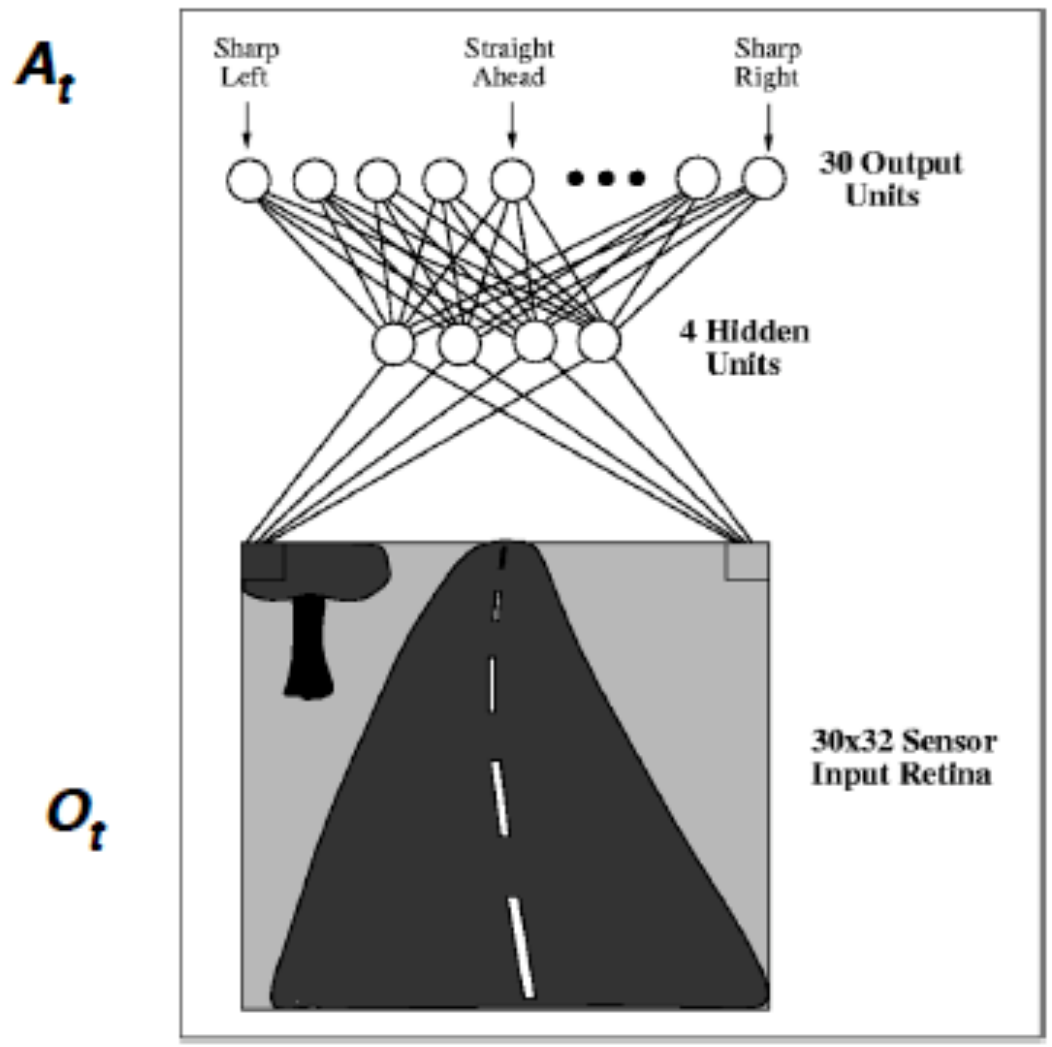
Action selection
by a shallow search



Self-Driving Cars

Self-Driving Cars

Policy network π: mapping observations to actions



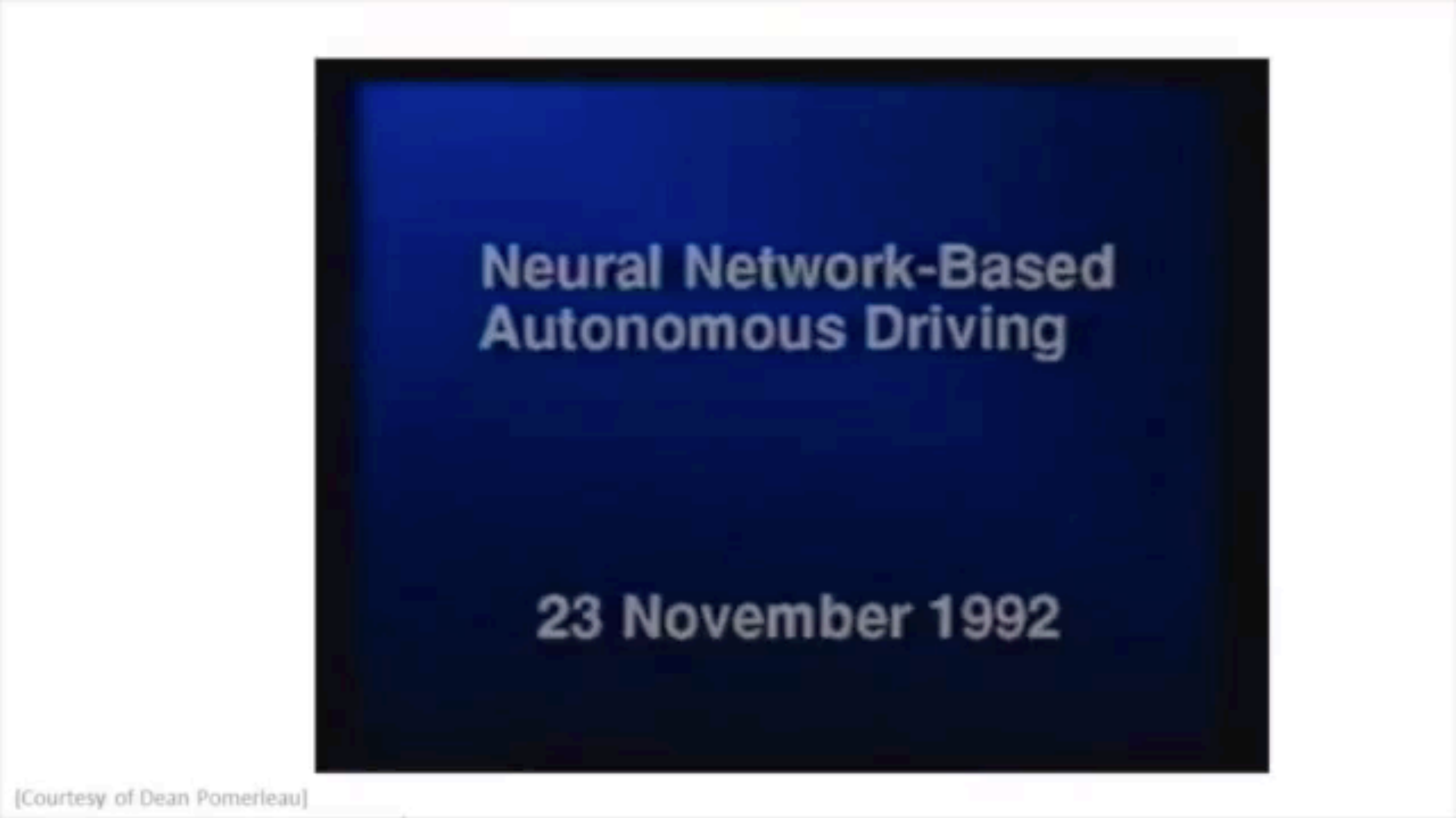
1989

ALVINN, an autonomous land vehicle in a neural network

Dean A. Pomerleau
Carnegie Mellon University

Self-Driving Cars

- [ALVINN video]



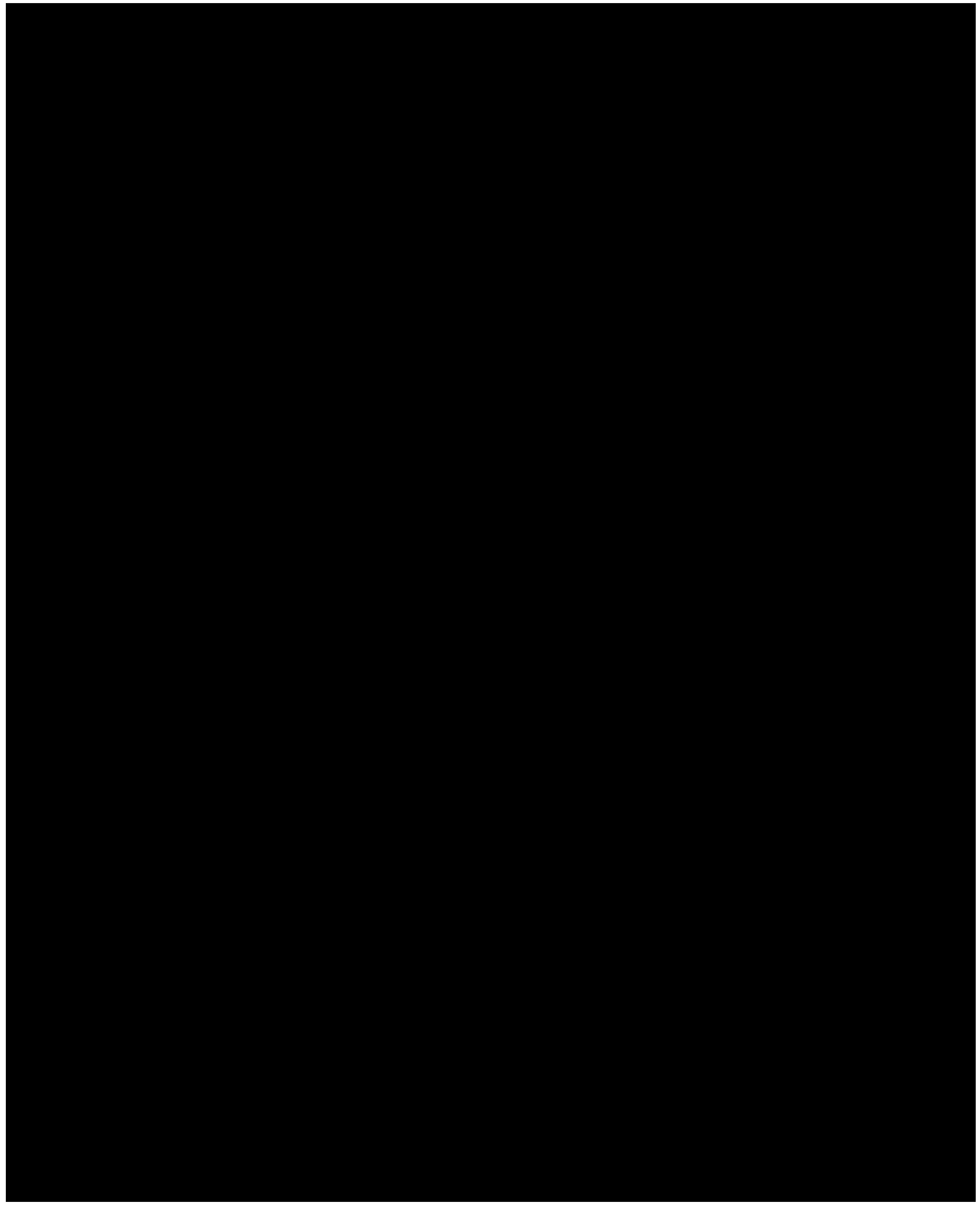
- behavior cloning- learning from the human driver
- data augmentation to deal with compounding errors

ALVINN (Autonomous Land Vehicle In a Neural Network), Efficient Training of Artificial Neural Networks for Autonomous Navigation, Pomerleau 1991

Self-Driving Cars

- Currently: much better computer vision front end: object detection, trajectory forecasting etc.
- Open problem: learning reward functions from humans on how to behave on intersections, crowds, traffic jams, etc. .

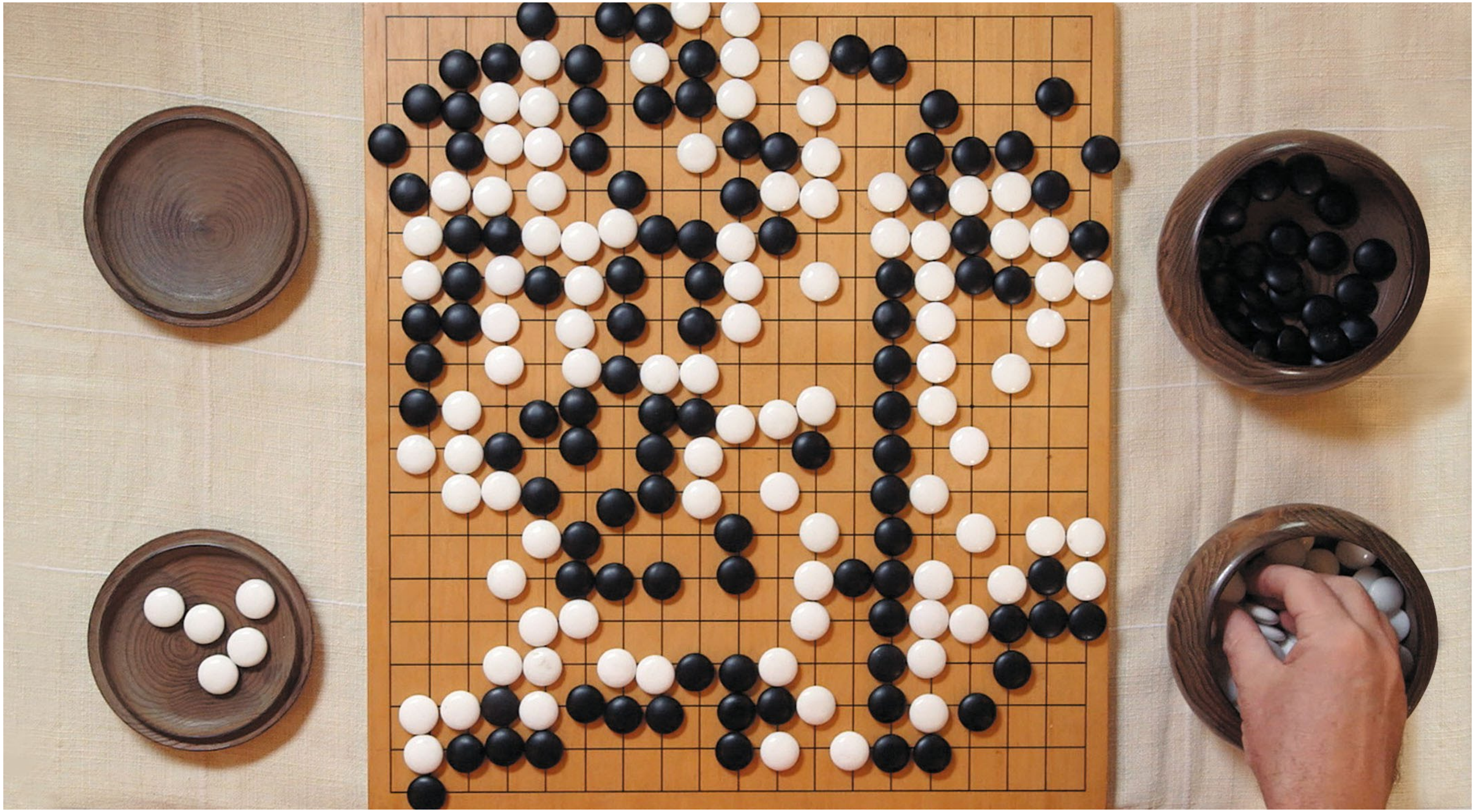
Atari



Deep Q learning

Deep Mind
2014+

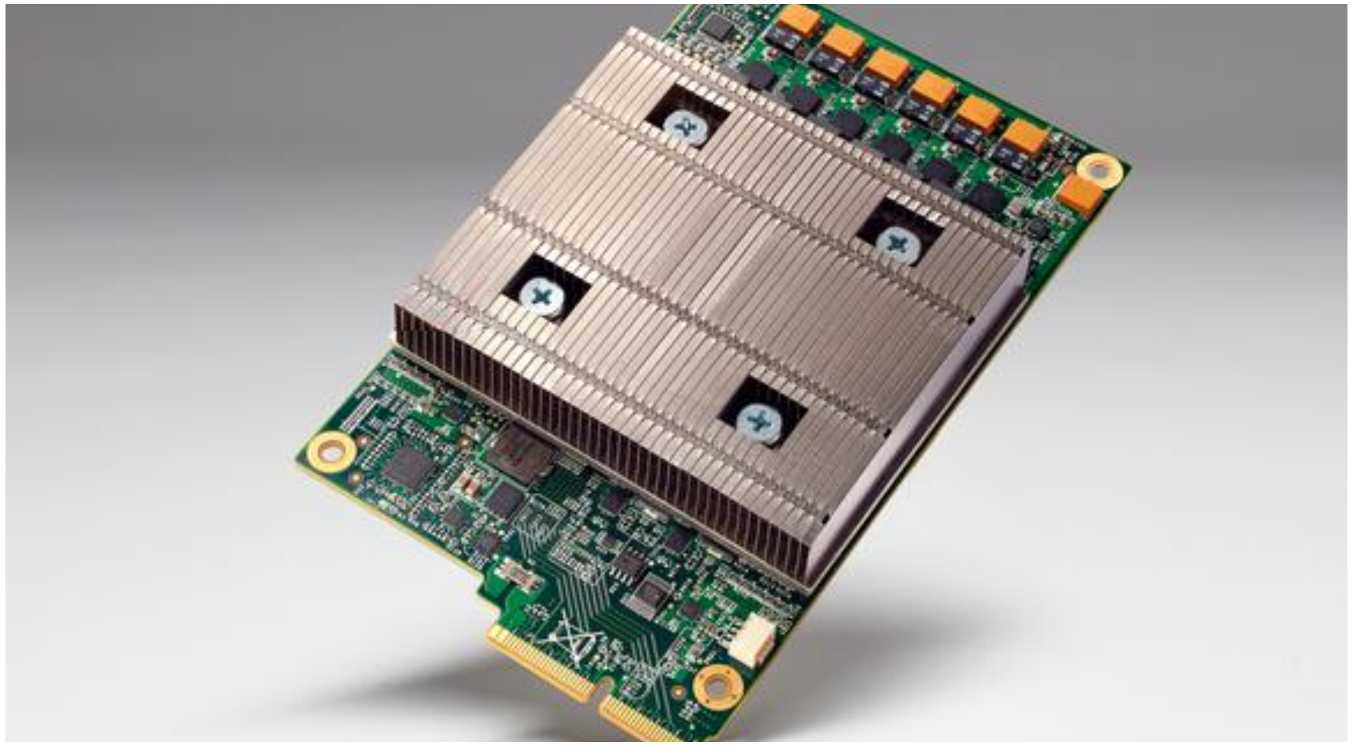
GO



AlphaGo

- Monte Carlo Tree Search with neural nets
- expert demonstrations
- self play

AlphaGo



Tensor Processing Unit from Google

Go Versus the real world

How the world of Alpha Go is different than the real world?

1. Known environment (known entities and dynamics) Vs Unknown environment (unknown entities and dynamics).

2. Need for behaviors to transfer across environmental variations since the real world is very diverse

3. Discrete Vs Continuous actions

4. One goal Vs many goals

5. Rewards are provided automatically by an oracle environment VS rewards need themselves to be detected

Alpha Go Versus the real world

How the world of Alpha Go is different than the real world?

1. Known environment (known entities and dynamics) Vs Unknown environment (unknown entities and dynamics).

2. Need for behaviors to transfer across environmental variations since the real world is very diverse

Alpha Go Versus the real world

How the world of Alpha Go is different than the real world?

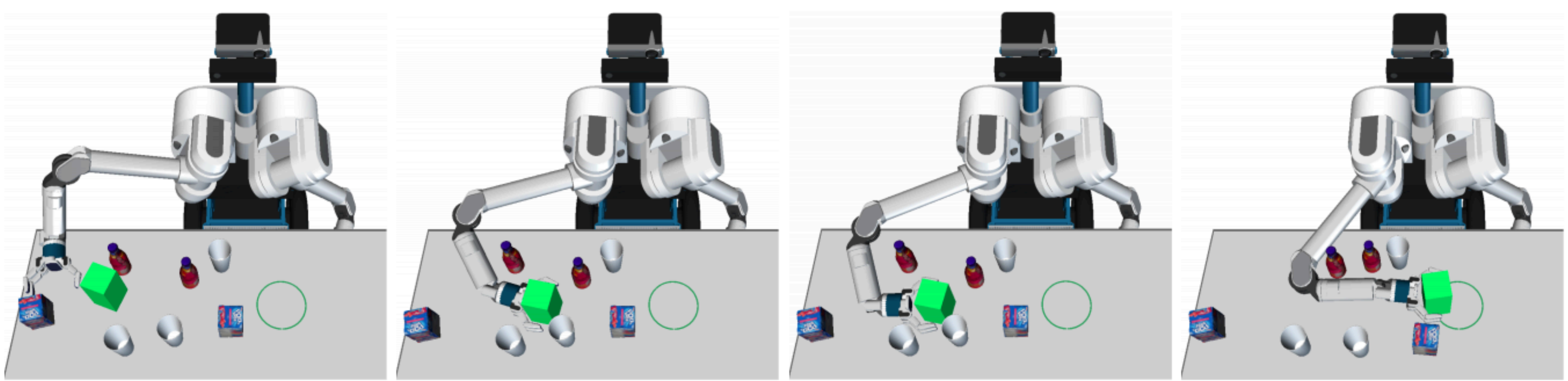
1. Known environment (known entities and dynamics) Vs Unknown environment (unknown entities and dynamics).

2. Need for behaviors to transfer across environmental variations since the real world is very diverse

State estimation: To be able to act you need first to be able to see, detect the objects that you interact with, detect whether you achieved your goal

State estimation - Two extremes

- Assuming we know everything about the objects (object locations, 3D shapes, physical properties). Use planners to search for the action sequence to achieve a desired goal.

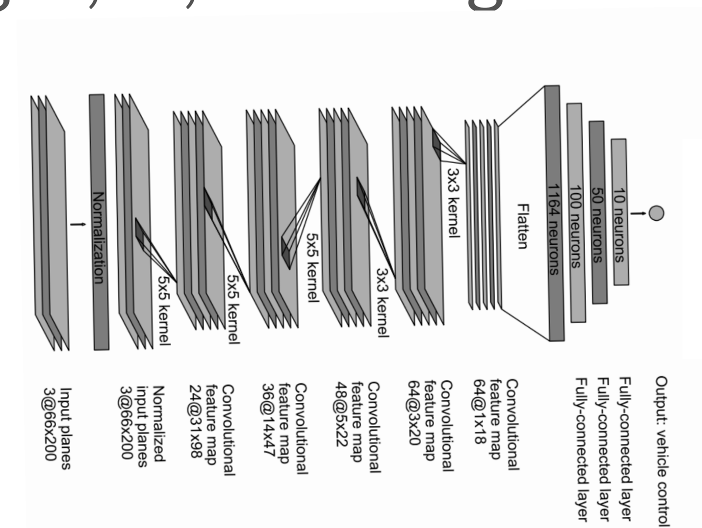


Rearrangement Planning via Heuristic Search, Jennifer E. King, Siddhartha S. Srinivasa

State estimation - Two extremes

- Assuming we know everything about the objects (object locations, 3D shapes, physical properties). Use planners to search for the action sequence to achieve a desired goal.

- Assuming we know nothing about the objects. Learn to map pixels directly to actions while optimizing for your end task, i.e., not crashing and obeying the traffic signs, or, imitating human demonstrations.



End-to-End Learning for Self-Driving Cars, NVIDIA

Go Versus the real world

How the world of Go is different than the real world?

1. Known environment (known entities and dynamics) Vs Unknown environment (unknown entities and dynamics).

2. Need for behaviors to transfer across environmental variations since the real world is very diverse

3. Discrete Vs Continuous actions

4. One goal Vs many goals

5. Rewards automatic VS rewards need themselves to be detected

Go Versus the real world

How the world of Go is different than the real world?

1. Known environment (known entities and dynamics) Vs Unknown environment (unknown entities and dynamics).

2. Need for behaviors to transfer across environmental variations since the real world is very diverse

3. Discrete Vs Continuous actions

4. One goal Vs many goals

5. Rewards automatic VS rewards need themselves to be detected

Go Versus the real world

How the world of Go is different than the real world?

1. Known environment (known entities and dynamics) Vs Unknown environment (unknown entities and dynamics).

2. Need for behaviors to transfer across environmental variations since the real world is very diverse

3. Discrete Vs Continuous actions (curriculum learning, progressively add degrees of freedom)

4. One goal Vs many goals (generalized policies parametrized by the goal, Hindsight Experience Replay)

5. Rewards automatic VS rewards need themselves to be detected

Alpha Go Versus the real world

How the world of Go is different than the real world?

1. Known environment (known entities and dynamics) Vs Unknown environment (unknown entities and dynamics).

2. Need for behaviors to transfer across environmental variations since the real world is very diverse

3. Discrete Vs Continuous actions (curriculum learning, progressively add degrees of freedom)

4. One goal Vs many goals (generalized policies parametrized by the goal, Hindsight Experience Replay)

5. Rewards automatic VS rewards need themselves to be detected (learning perceptual rewards, use Computer Vision to detect success)

AI’s paradox

Go Versus the real world

Beating the world champion is easier than moving the Go stones.

AI's paradox

Hans Moravec

"it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility"

AI’s paradox

Marvin Minsky

"we're more aware of simple processes that don't work well than of complex ones that work flawlessly"

Evolutionary explanation

Hans Moravec

"We should expect the difficulty of reverse-engineering any human skill to be roughly proportional to the amount of time that skill has been evolving in animals.

The oldest human skills are largely unconscious and so appear to us to be effortless.

Therefore, we should expect skills that appear effortless to be difficult to reverse-engineer, but skills that require effort may not necessarily be difficult to engineer at all."

AI’s paradox

Intelligence was "best characterized as the things that highly educated male scientists found challenging", such as chess, symbolic integration, proving mathematical theorems and solving complicated word algebra problems.

Rodney Brooks

AI’s paradox

Intelligence was "best characterized as the things that highly educated male scientists found challenging", such as chess, symbolic integration, proving mathematical theorems and solving complicated word algebra problems.

"The things that children of four or five years could do effortlessly, such as visually distinguishing between a coffee cup and a chair, or walking around on two legs, or finding their way from their bedroom to the living room were not thought of as activities requiring intelligence.”

Rodney Brooks

AI’s paradox

Rodney Brooks

Intelligence was "best characterized as the things that highly educated male scientists found challenging", such as chess, symbolic integration, proving mathematical theorems and solving complicated word algebra problems.

"The things that children of four or five years could do effortles

No cognition. Just sensing and action

coffee cup and a chair, or walking around on two legs, or finding their way from their bedroom to the living room were not thought of as activities requiring intelligence.”

Learning from Babies

- Be multi-modal
- Be incremental
- Be physical
- Explore
- Be social
- Learn a language

The Development of Embodied Cognition: Six Lessons from Babies. Linda Smith, Michael Gasser