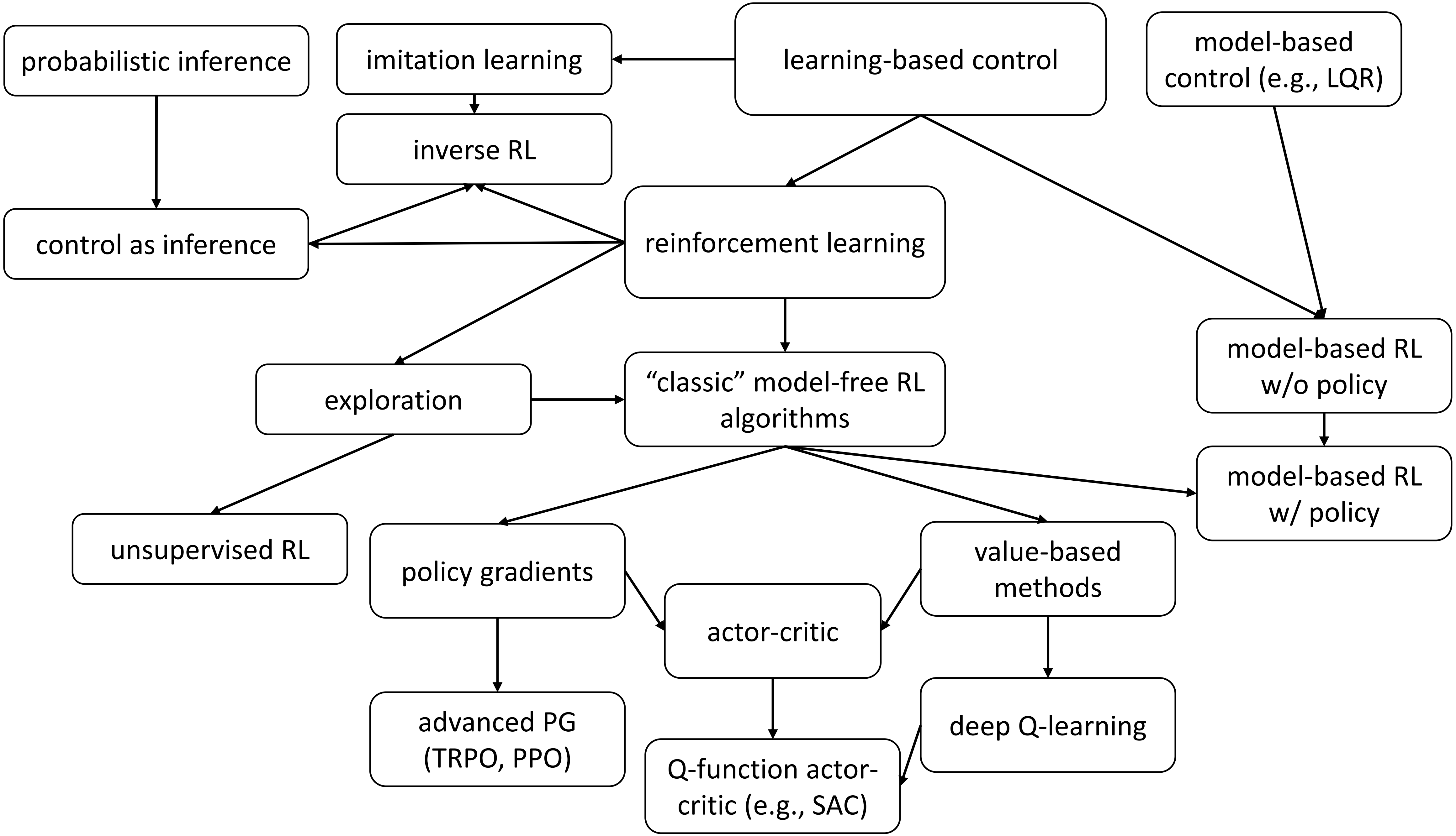
Challenges and Open Problems

CS 285

Instructor: Sergey Levine
UC Berkeley

A Brief Review



probabilistic inference
imitation learning
learning-based control
model-based control (e.g., LQR)
inverse RL
control as inference
reinforcement learning
model-based RL w/o policy
exploration
"classic" model-free RL algorithms
model-based RL w/ policy
unsupervised RL
policy gradients
value-based methods
actor-critic
advanced PG (TRPO, PPO)
deep Q-learning
Q-function actor-critic (e.g., SAC)

Challenges in Deep Reinforcement Learning

What’s the problem?

Challenges with **core algorithms**:

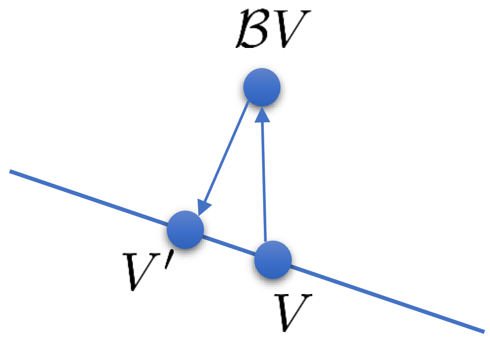
- Stability: does your algorithm converge?
- Efficiency: how long does it take to converge? (how many samples)
- Generalization: after it converges, does it generalize?

Challenges with **assumptions**:

- Is this even the right problem formulation?
- What is the source of *supervision*?

Stability and hyperparameter tuning

- Devising stable RL algorithms is very hard
- Q-learning/value function estimation
    - Fitted Q/fitted value methods with deep network function estimators are typically not contractions, hence no guarantee of convergence
    - Lots of parameters for stability: target network delay, replay buffer size, clipping, sensitivity to learning rates, etc.
- Policy gradient/likelihood ratio/REINFORCE
    - Very high variance gradient estimator
    - Lots of samples, complex baselines, etc.
    - Parameters: batch size, learning rate, design of baseline
- Model-based RL algorithms
    - Model class and fitting method
    - Optimizing policy w.r.t. model non-trivial due to backpropagation through time
    - More subtle issue: policy tends to *exploit* the model



gradient-free methods
(e.g. NES, CMA, etc.)

10x

fully online methods
(e.g. A3C)

10x

policy gradient methods
(e.g. TRPO)

10x

replay buffer value estimation methods
(Q-learning, DDPG, NAF, SAC, etc.)

10x

model-based deep RL
(e.g. PETS, guided policy search)

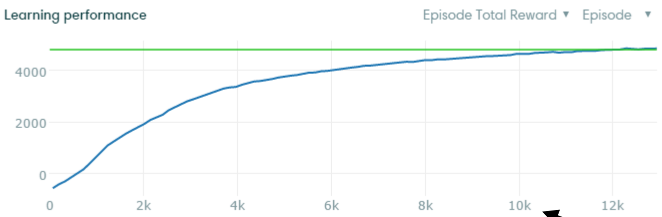
10x

model-based "shallow" RL
(e.g. PILCO)

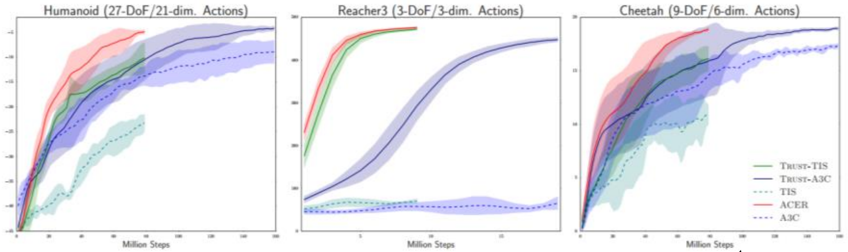
Evolution Strategies as a Scalable Alternative to Reinforcement Learning

Tim Salimans[1] Jonathan Ho[1] Xi Chen[1] Ilya Sutskever[1]

half-cheetah (slightly different version)



TRPO+GAE (Schulman et al. '16)

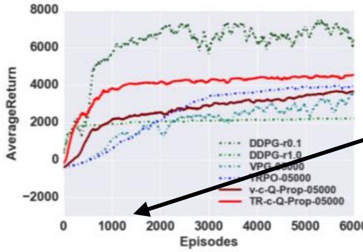


Wang et al. '17

100,000,000 steps
(100,000 episodes)
(~ 15 days real time)

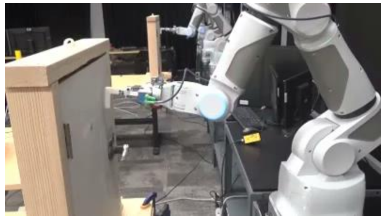
10,000,000 steps
(10,000 episodes)
(~ 1.5 days real time)

half-cheetah

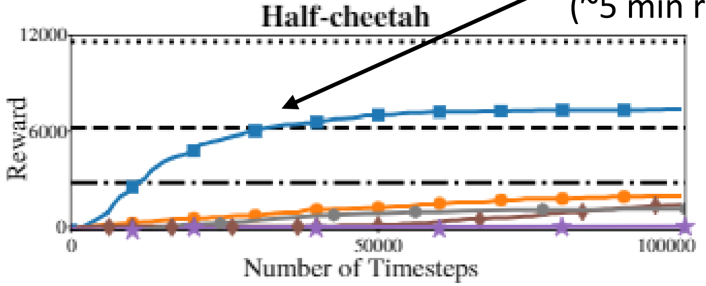


Gu et al. '16

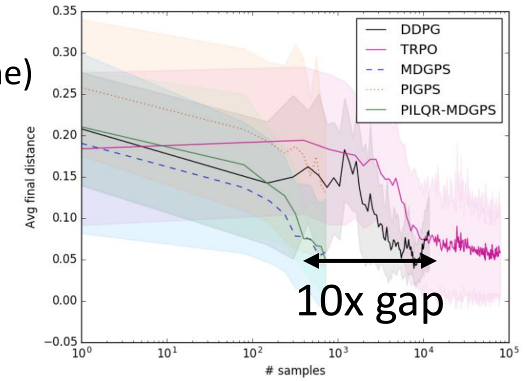
1,000,000 steps
(1,000 episodes)
(~3 hours real time)



30,000 steps
(30 episodes)
(~5 min real time)



Chua et a. '18: Deep Reinforcement Learning in a Handful of Trials

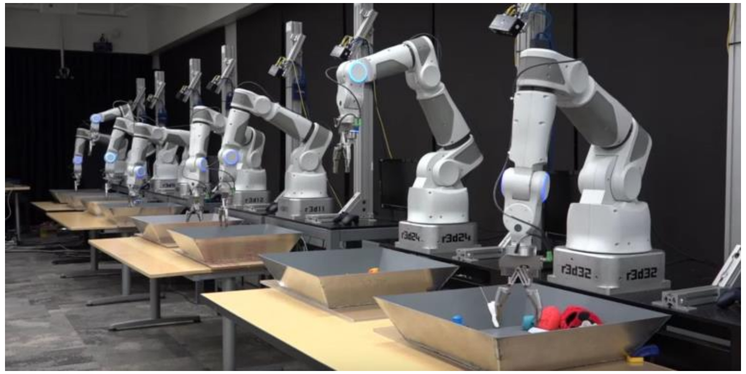


Chebotar et al. '17 (note log scale)

about 20 minutes of experience on a real robot

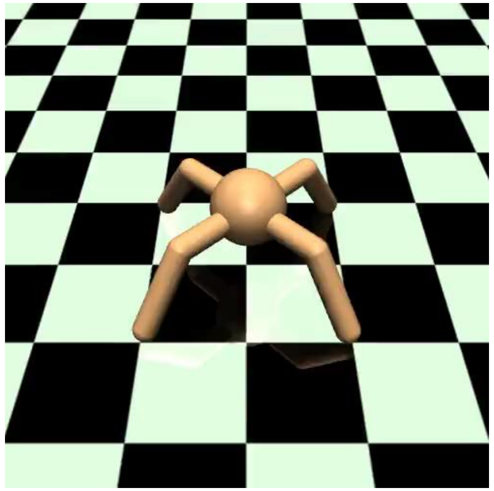
The challenge with sample complexity

- Need to wait for a long time for your homework to finish running
- Real-world learning becomes difficult or impractical
- Precludes the use of expensive, high-fidelity simulators
- Limits applicability to real-world problems



Scaling up deep RL & generalization

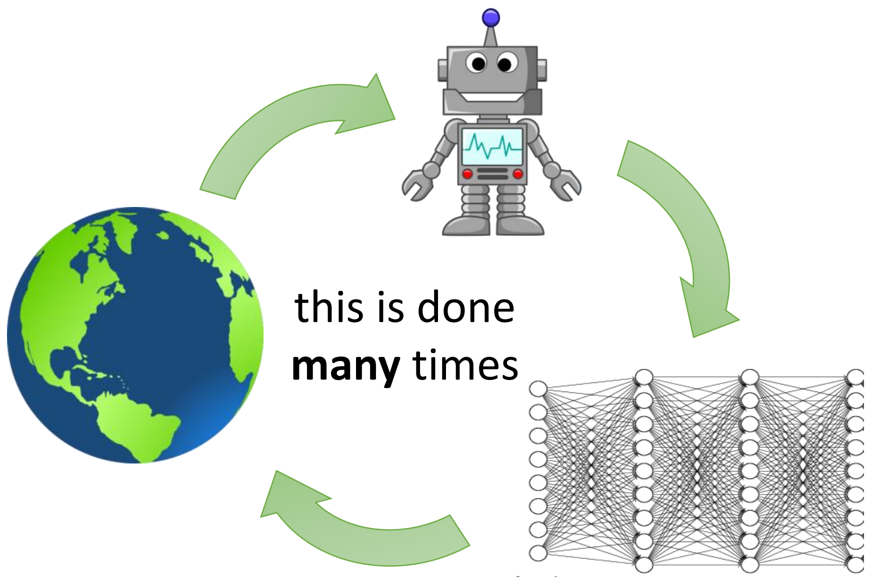
- Large-scale
- Emphasizes diversity
- Evaluated on generalization



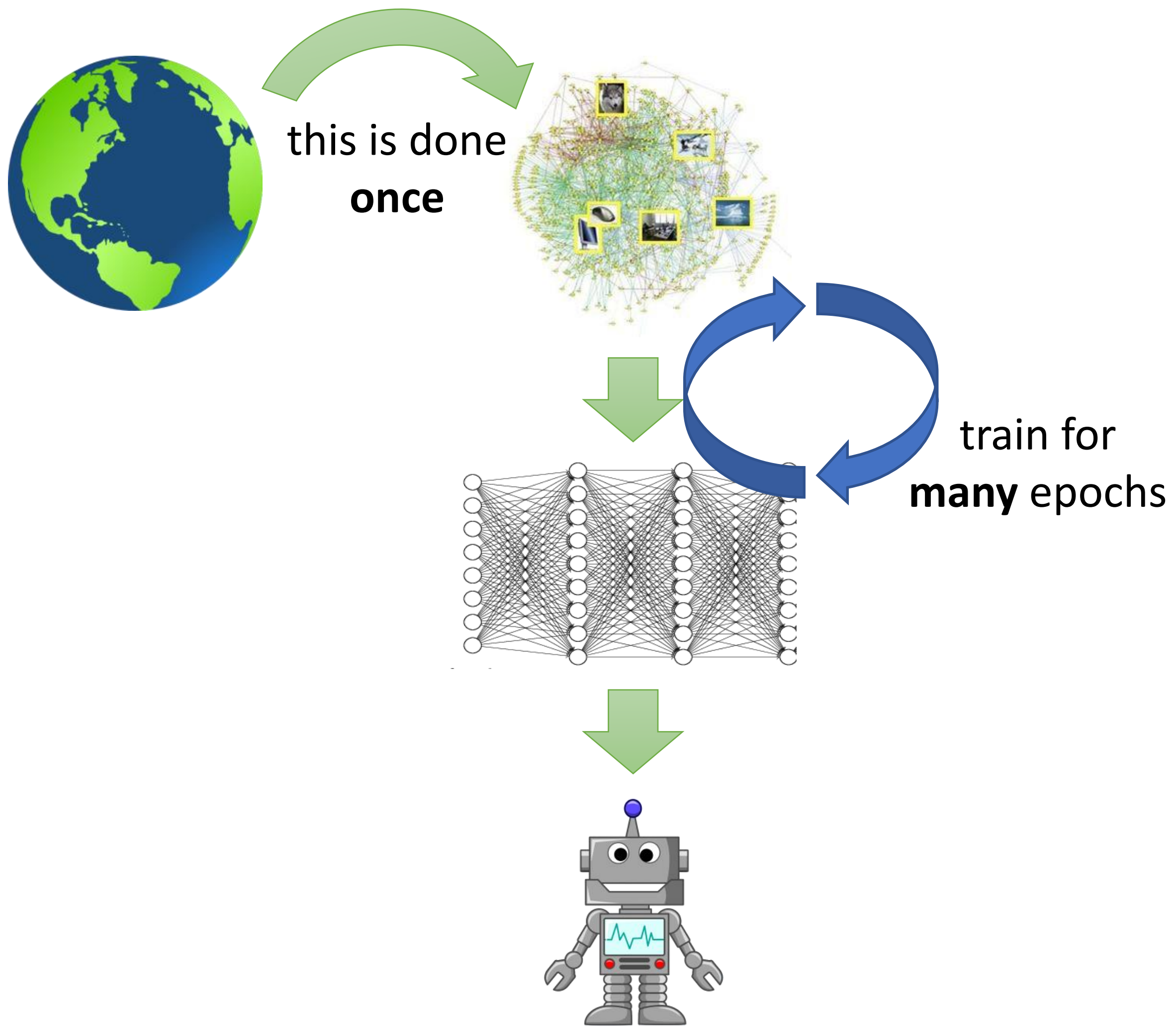
- Small-scale
- Emphasizes mastery
- Evaluated on performance
- Where is the generalization?

RL has a **big** problem

reinforcement learning

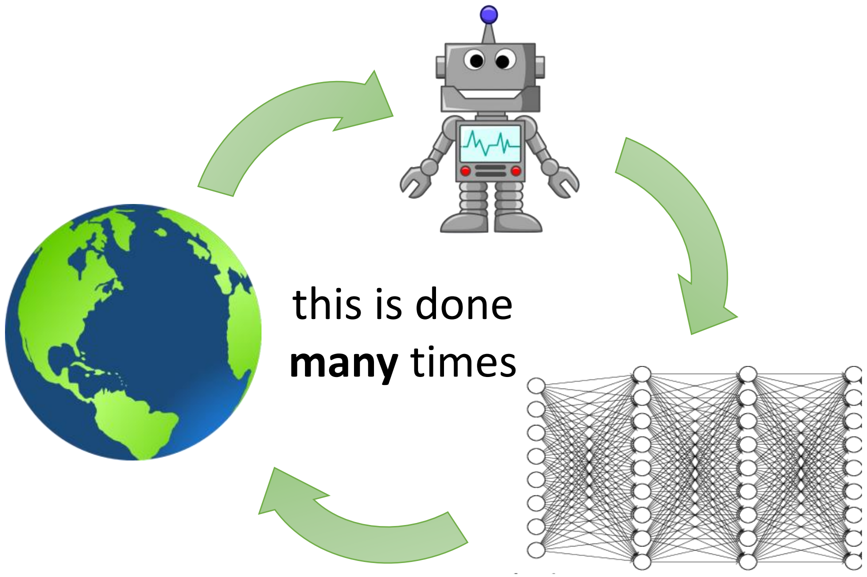


supervised machine learning

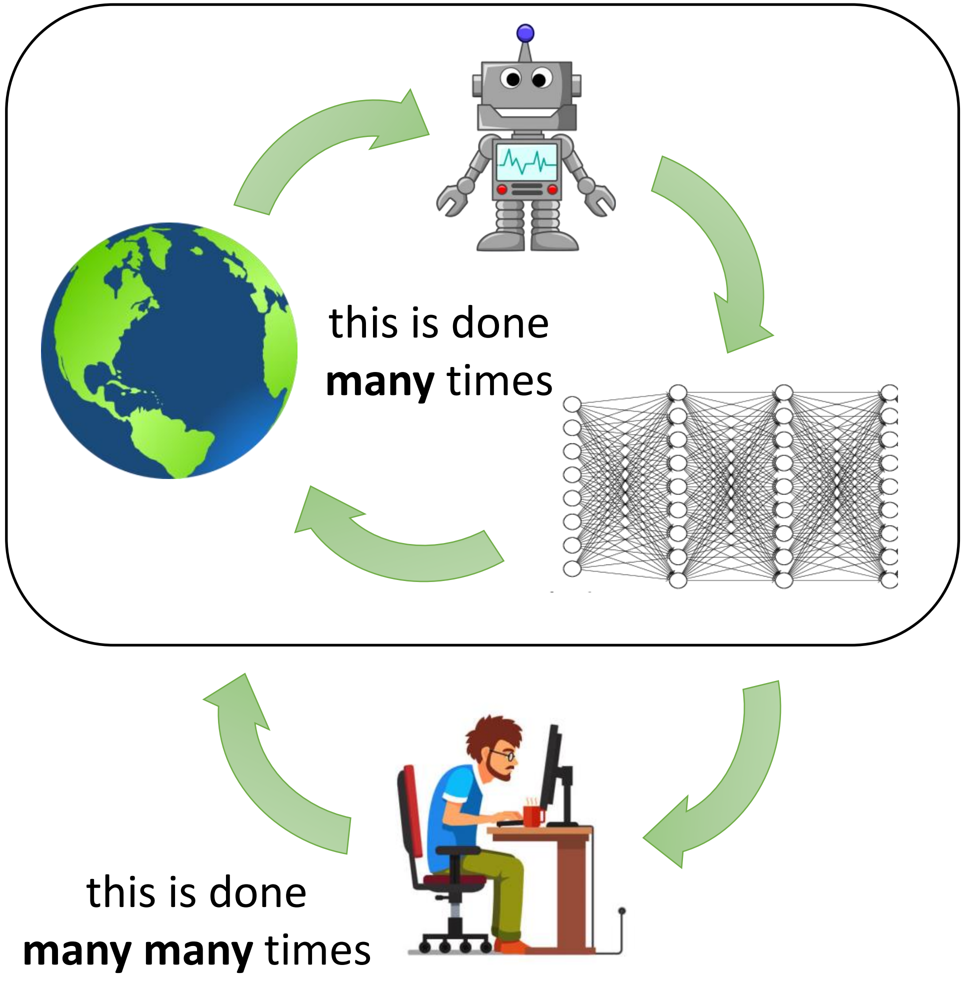


RL has a **big** problem

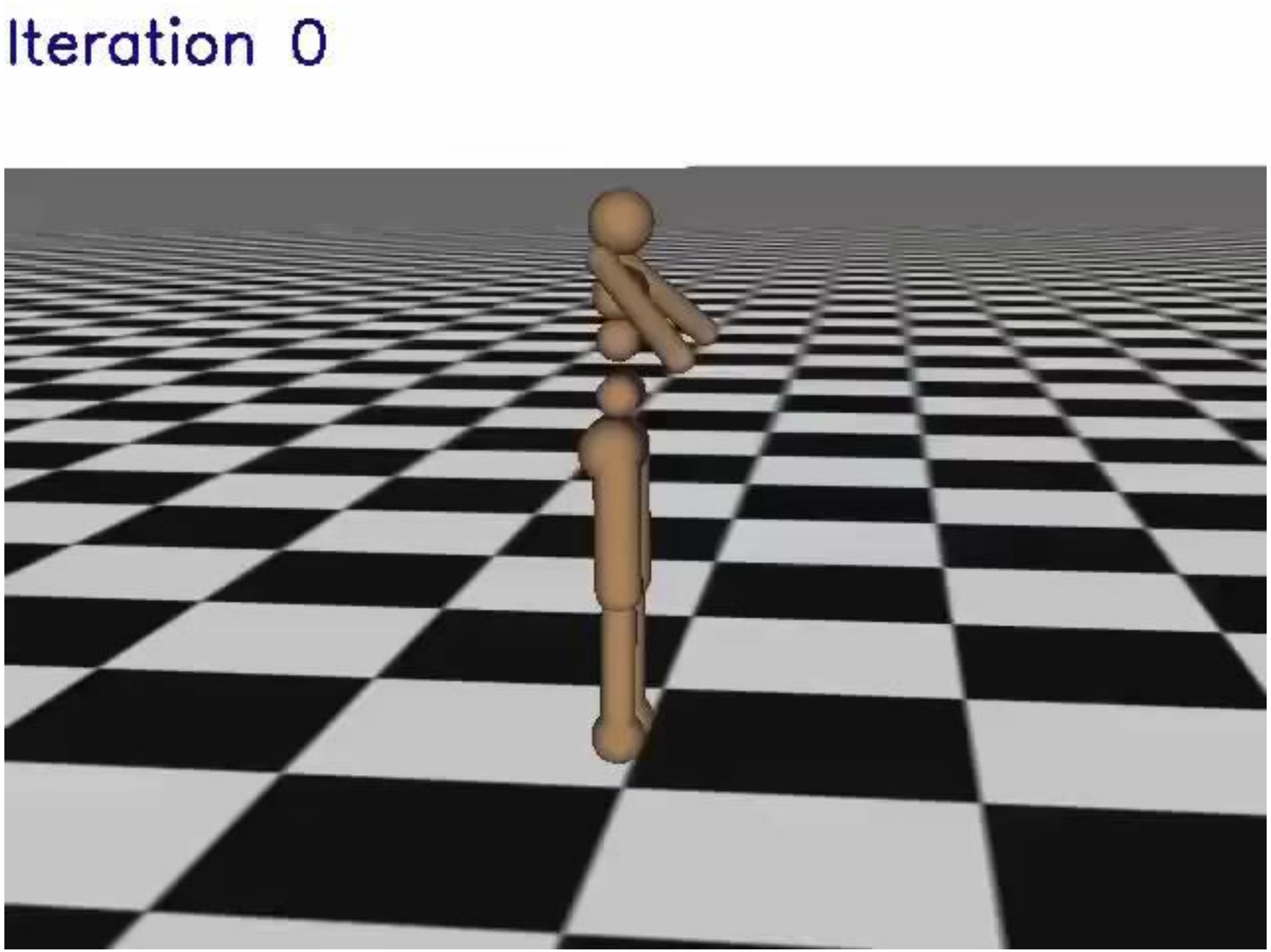
reinforcement learning



actual reinforcement learning



How bad is it?



- This is quite cool
- It takes 6 days of real time (if it was real time)
- …to run on an infinite flat plane

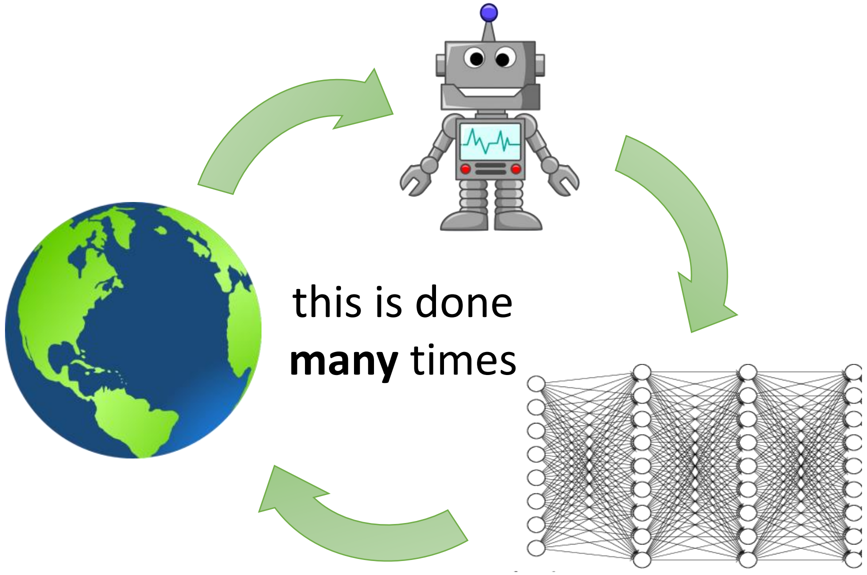



The real world is not so simple!

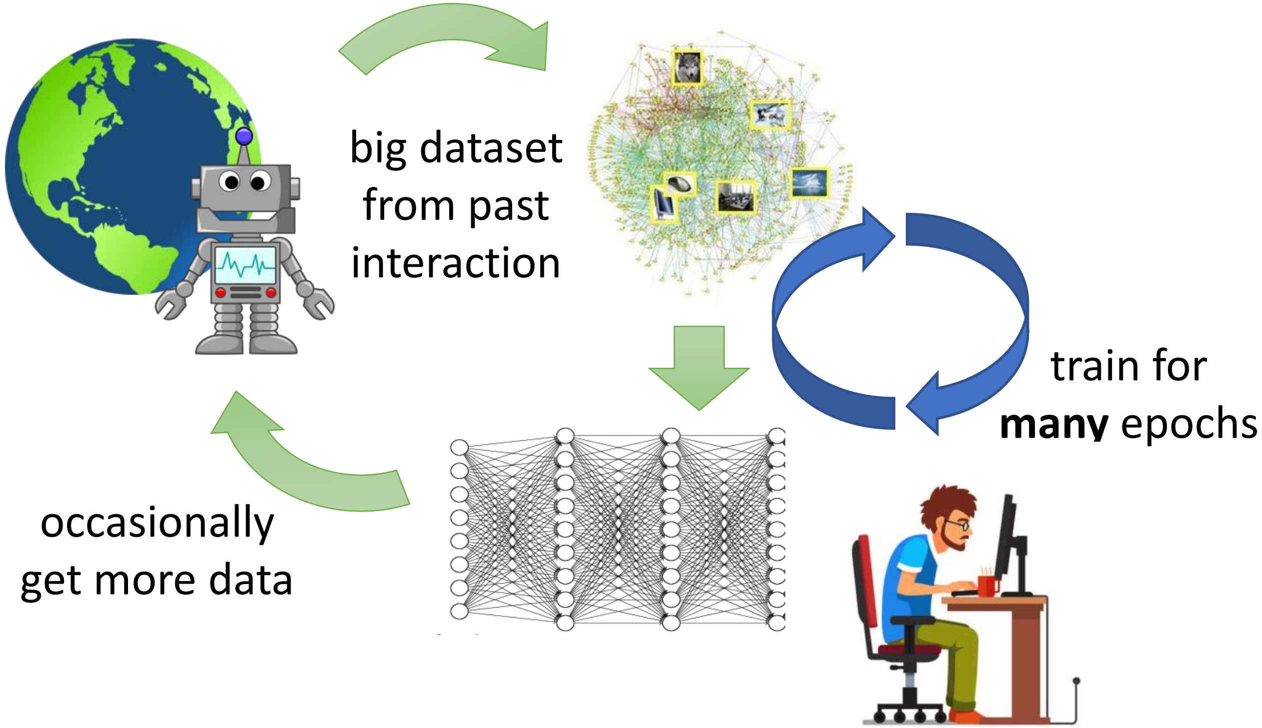
Schulman, Moritz, L., Jordan, Abbeel ’16

Off-policy RL?

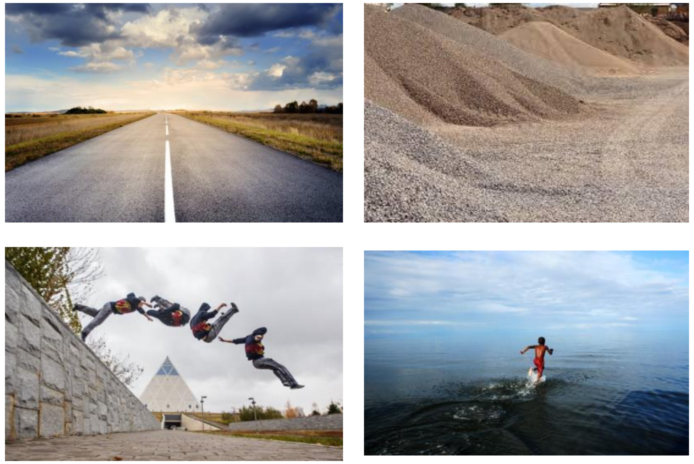
reinforcement learning



off-policy reinforcement learning



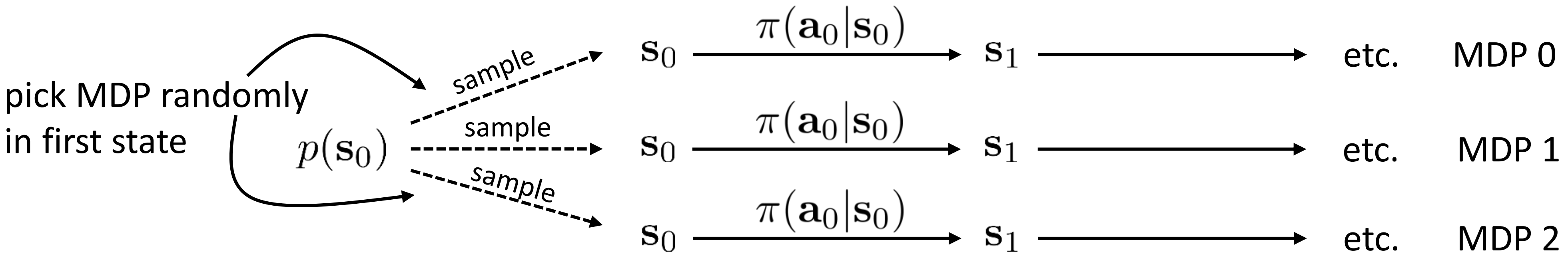
Single task or multi-task?



The real world is not so simple!

this is where generalization can come from…

maybe doesn’t require any new assumption, but might merit additional treatment



Where does the **supervision** come from?

- If you want to learn from many different tasks, you need to get those tasks somewhere!
- Learn objectives/rewards from demonstration (inverse reinforcement learning)
- Generate objectives automatically?

reward

Mnih et al. '15

reinforcement learning agent

what is the reward?

Other sources of supervision

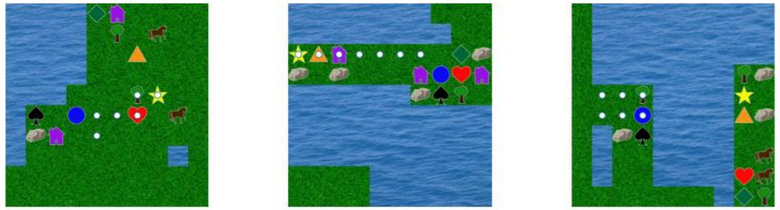
Should supervision tell us **what** to do or **how** to do it?

- Demonstrations
  - Muelling, K et al. (2013). Learning to Select and Generalize Striking Movements in Robot Table Tennis

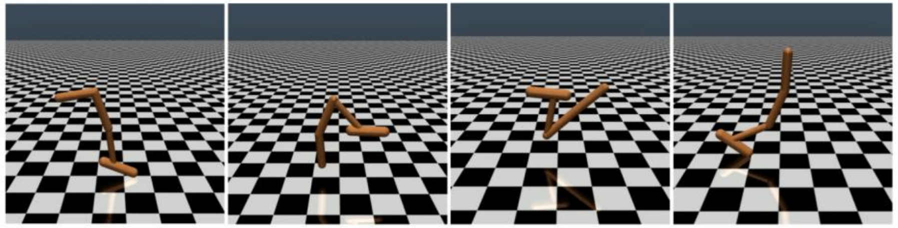
- Language
  - Andreas et al. (2018). Learning with latent language

Human description:
`move to the star`

Inferred description:
`reach the star cell`



- Human preferences
  - Christiano et al. (2017). Deep reinforcement learning from human preferences



Rethinking the Problem Formulation

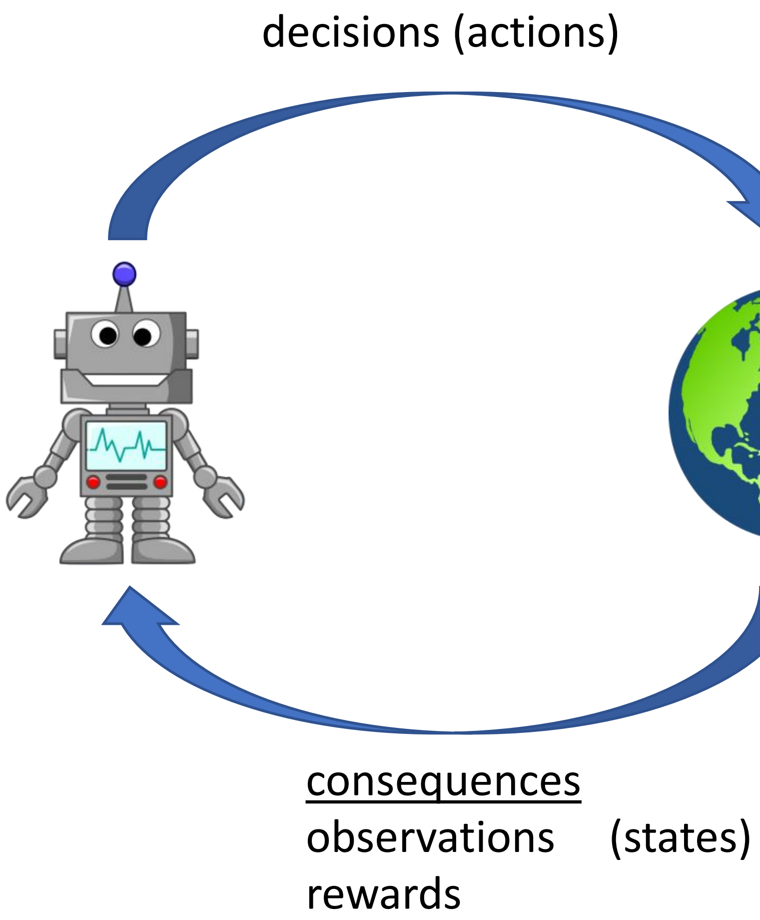
- How should we define a *control* problem?
  - What is the data?
  - What is the goal?
  - What is the supervision?
    - may not be the same as the goal…
- Think about the assumptions that fit your problem setting!
- Don’t assume that the basic RL problem is set in stone

Some perspectives…

Reinforcement Learning as an Engineering Tool
Reinforcement Learning and the Real World
Reinforcement Learning as “Universal” Learning

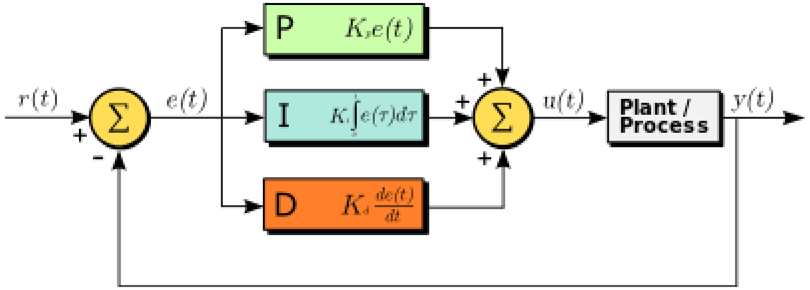
Reinforcement Learning as an Engineering Tool

What we think RL is…

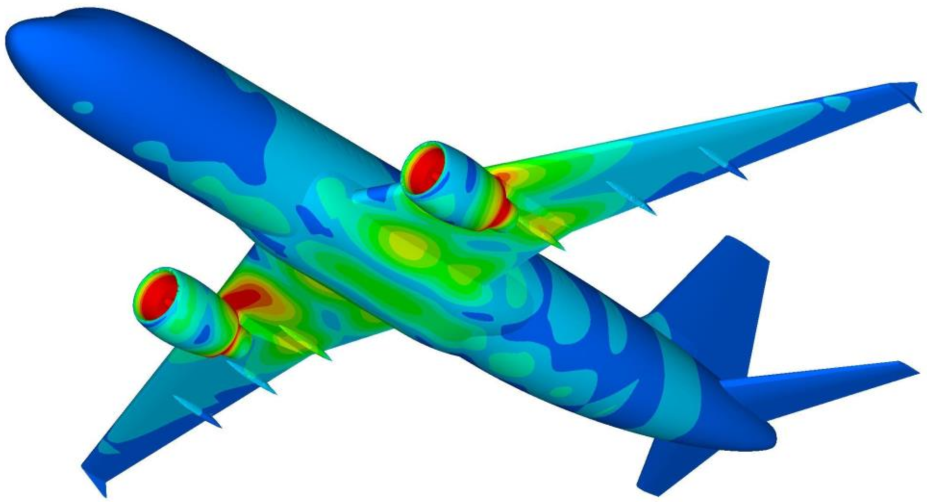


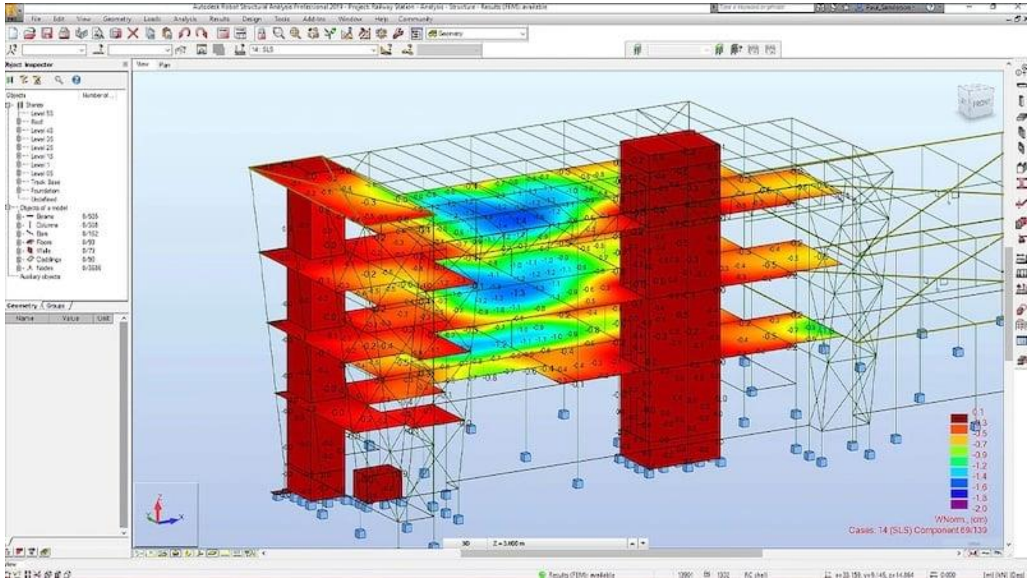
Engineering a control system

$$
\begin{aligned}
\mathbf{r} &= \mathbf{r}(t) = r\hat{\mathbf{e}}_r \\
\mathbf{v} &= v\hat{\mathbf{e}}_r + r\frac{d\theta}{dt}\hat{\mathbf{e}}_\theta + r\frac{d\varphi}{dt}\sin\theta\hat{\mathbf{e}}_\varphi \\
\mathbf{a} &= \left(a - r\left(\frac{d\theta}{dt}\right)^2 - r\left(\frac{d\varphi}{dt}\right)^2\sin^2\theta\right)\hat{\mathbf{e}}_r \\
&\quad + \left(r\frac{d^2\theta}{dt^2} + 2v\frac{d\theta}{dt} - r\left(\frac{d\varphi}{dt}\right)^2\sin\theta\cos\theta\right)\hat{\mathbf{e}}_\theta \\
&\quad + \left(r\frac{d^2\varphi}{dt^2}\sin\theta + 2v\frac{d\varphi}{dt}\sin\theta + 2r\frac{d\theta}{dt}\frac{d\varphi}{dt}\cos\theta\right)\hat{\mathbf{e}}_\varphi
\end{aligned}
$$

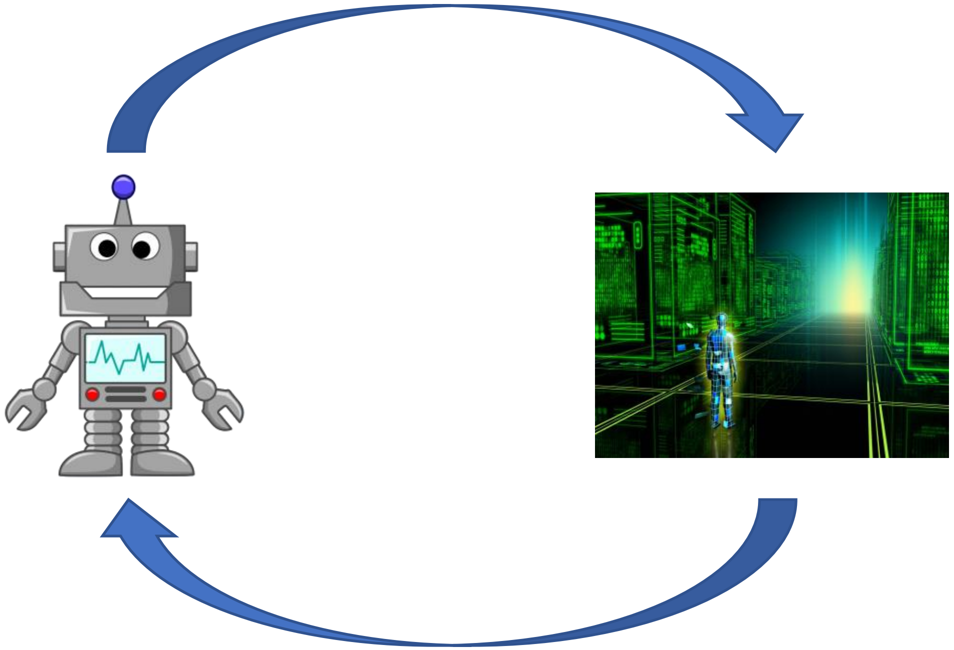


Characterization and simulation…





RL: anything you can *simulate* you can *control*



- Provides a powerful engineering tool
- Now *that* different from conventional engineering approach!
  - Before: characterize, simulate, control
  - Now: characterize, simulate, run RL
- Main role: powerful *inversion* engine
- Main weakness: still need to simulate!

Reinforcement Learning and the Real World

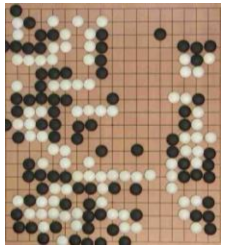

Google DeepMind
Challenge Match
8 - 15 March 2016
AlphaGo
AlphaGo
Lee Sedol
WWW.NEWS.CN

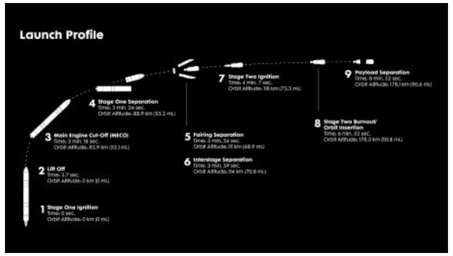
Moravec’s paradox

Moravec’s paradox seems like a statement about AI

but it is actually a statement about the physical universe

“easy” universes





Why?

“hard” universes

We are all prodigious olympians in perceptual and motor areas, so good that we make the difficult look easy. Abstract thought, though, is a new trick, perhaps less than 100 thousand years old. We have not yet mastered it. It is not all that intrinsically difficult; it just seems so when we do it.

- Hans Moravec

The main lesson of thirty-five years of AI research is that the hard problems are easy and the easy problems are hard. The mental abilities of a four-year-old that we take for granted – recognizing a face, lifting a pencil, walking across a room, answering a question – in fact solve some of the hardest engineering problems ever conceived.

- Steven Pinker


DETROIT DIESEL
92
POWER


PACIFIC GLORY
TUG

What does this have to do with RL?

How do we engineer a system that can deal with the unexpected?

- Minimal external supervision about what to do
- Unexpected situations that require adaptation
- Must discover solutions autonomously
- Must “stay alive” long enough to discover them!

- Humans are extremely good at this
- Current AI systems are extremely bad at this
- RL *in principle* can do this, and nothing else can

So what’s the problem?

➢ RL *in principle* can do this, and nothing else can

But we rarely study this kind of setting in RL research!

RL *should* be really good in the “hard” universes!

“easy” universes

success = high reward (“optimal control”)

closed world, rules are known

lots of simulation

Main question: can RL algorithms **optimize really well**

“hard” universes

success = “survival” (“good enough control”)

open world, everything must come from data

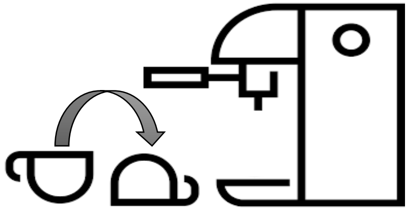
no simulation (because rules are unknown)

Main question: can RL **generalize** and **adapt**

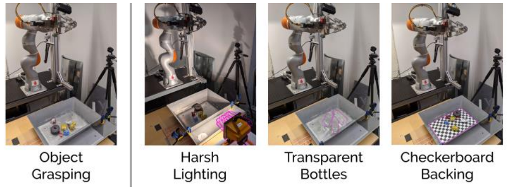
Some questions that come up in the real world

How do we tell RL agents **what we want them to do**?

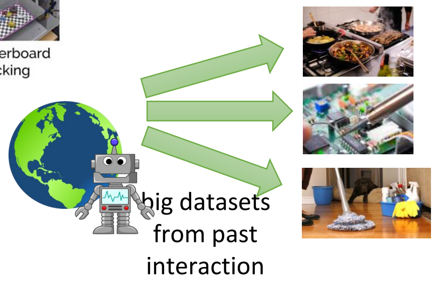
How can we learn **fully autonomously** in continual environments?



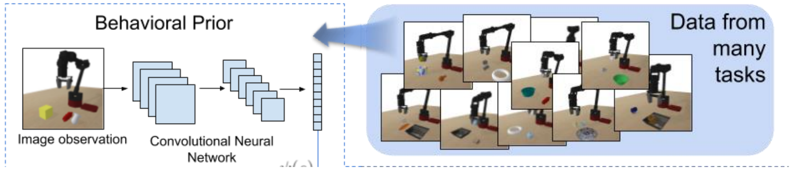
How do remain **robust** as the environment changes around us?



What is the right way to **generalize** using **experience** & **prior data**?

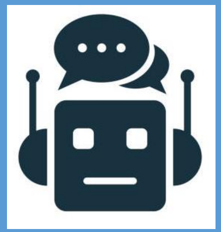


What is the right way to **bootstrap exploration** with **prior experience**?

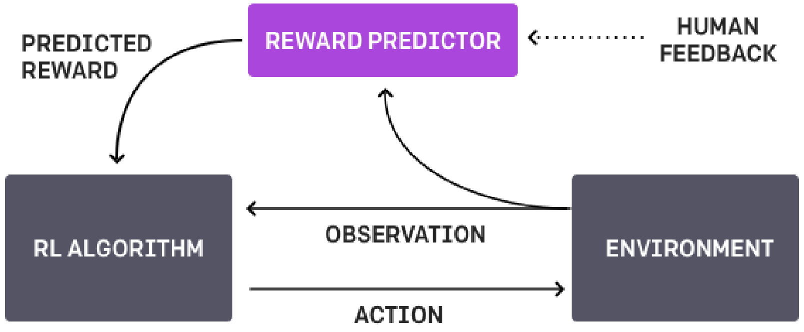


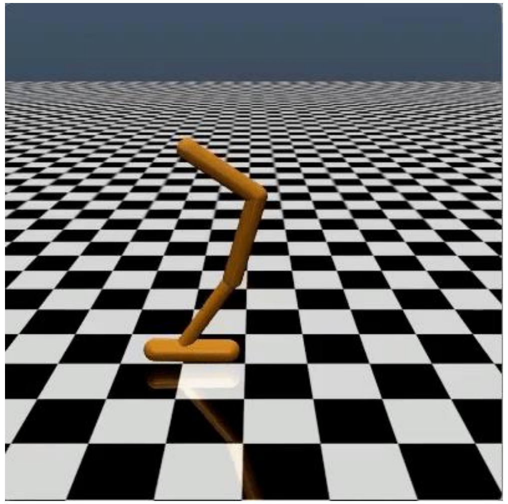
This is not about robots

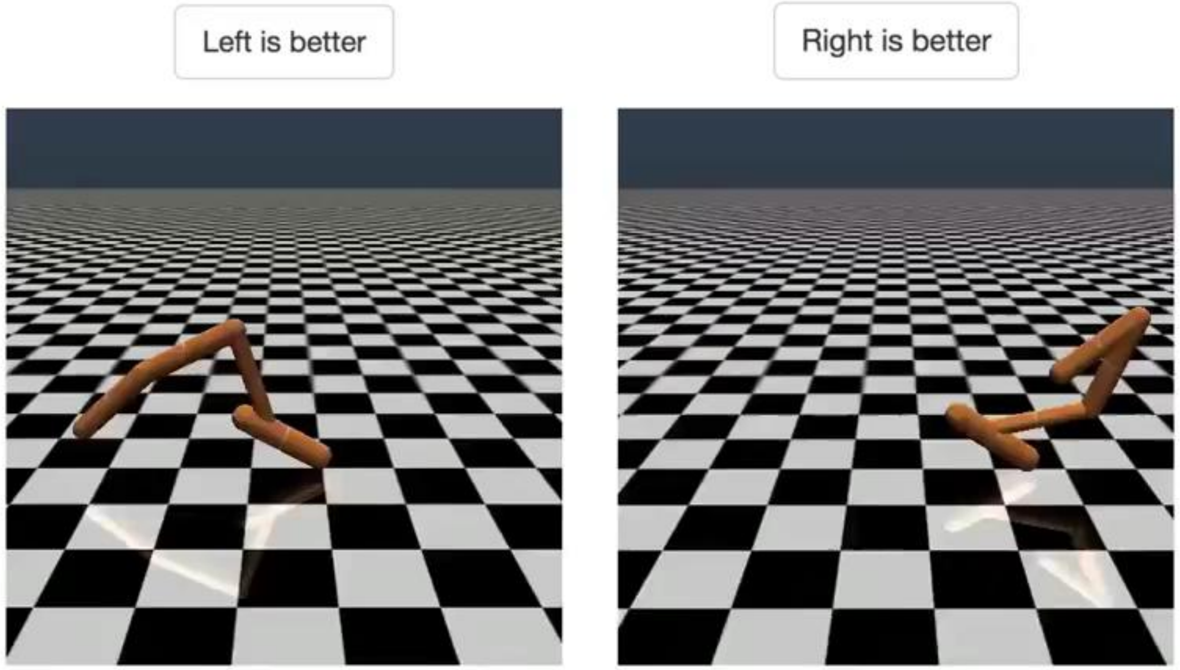
Robots are the most natural for us to think about, because they are embodied like we are



Other ways to **communicate objectives**?

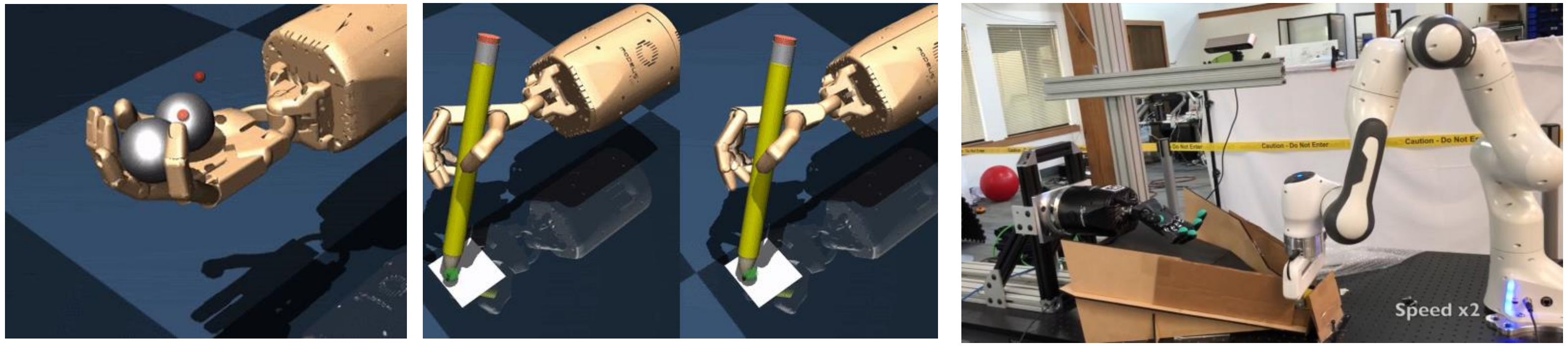




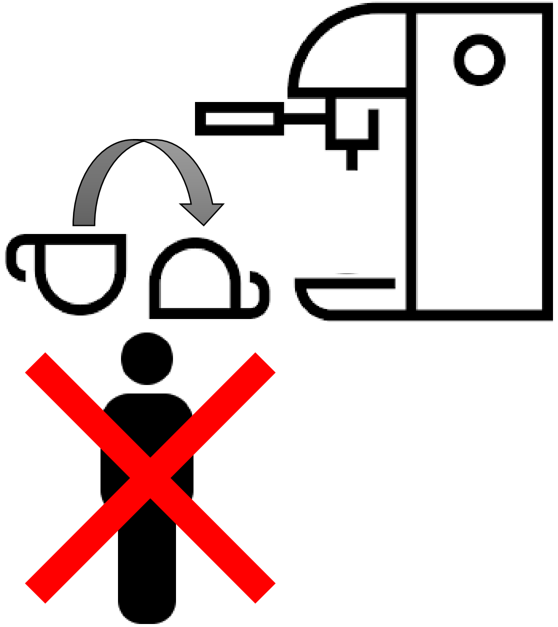


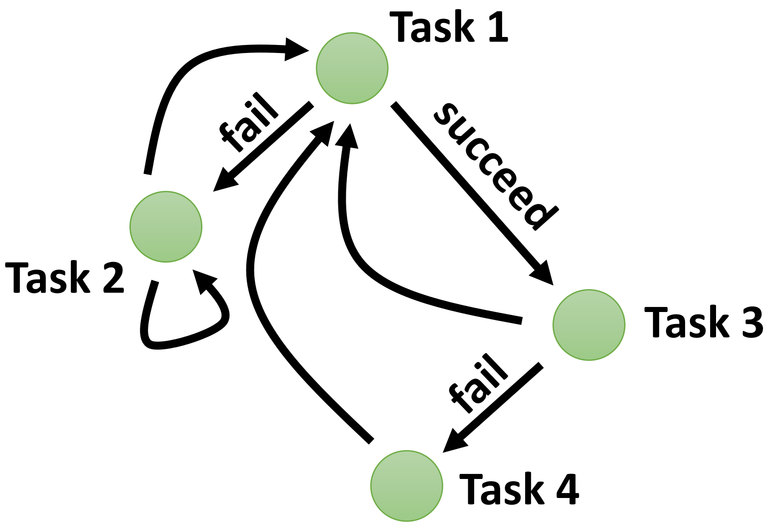
Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, Dario Amodei. **Deep reinforcement learning from human preferences**. 2017.

How can we learn **fully autonomously?**



Nagabandi, Konolige, Levine, Kumar. **Deep Dynamics Models for Learning Dexterous Manipulation.** CoRL 2019.





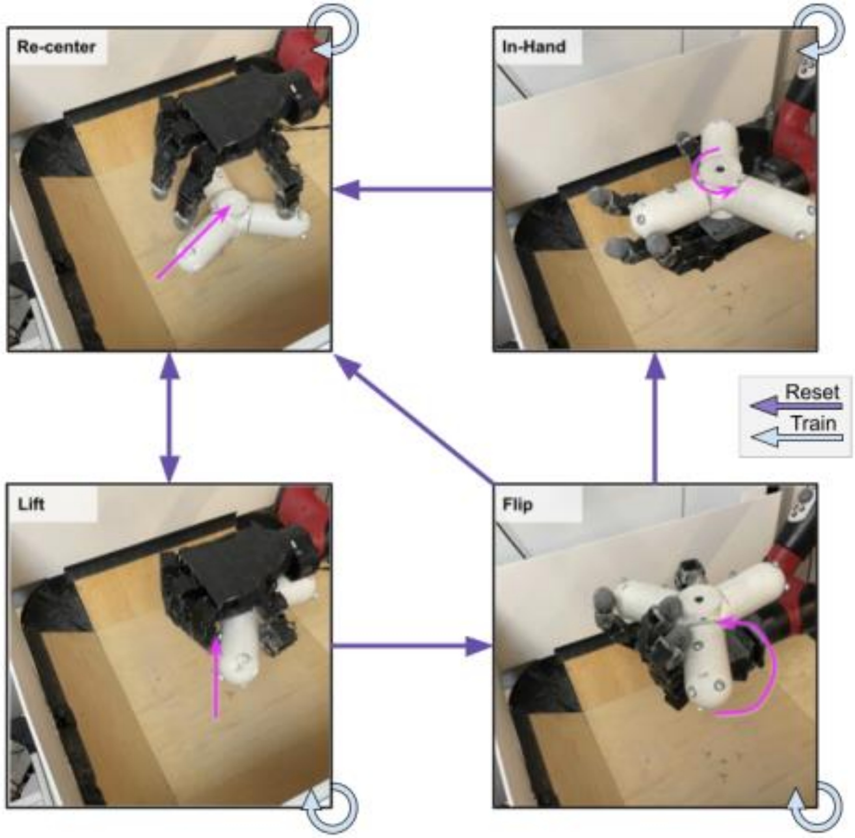
Task 1: put cup in coffee machine

Task 2: pick up cup

Task 3: replace cup

Task 4: clean up spill from cup...

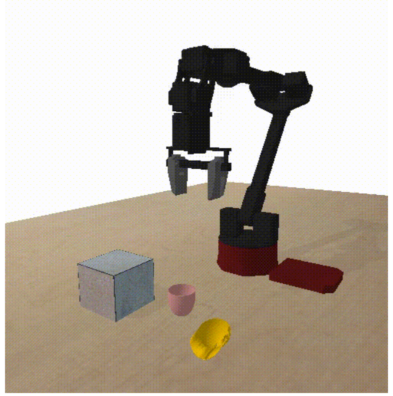
How can we learn **fully autonomously?**



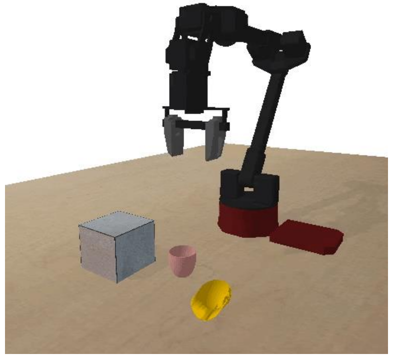
Gupta, Yu, Zhao, Kumar, Rovinsky, Xu, Devlin, Levine. **Reset-Free Reinforcement Learning via Multi-Task Learning: Learning Dexterous Manipulation Behaviors without Human Intervention**. 2021.

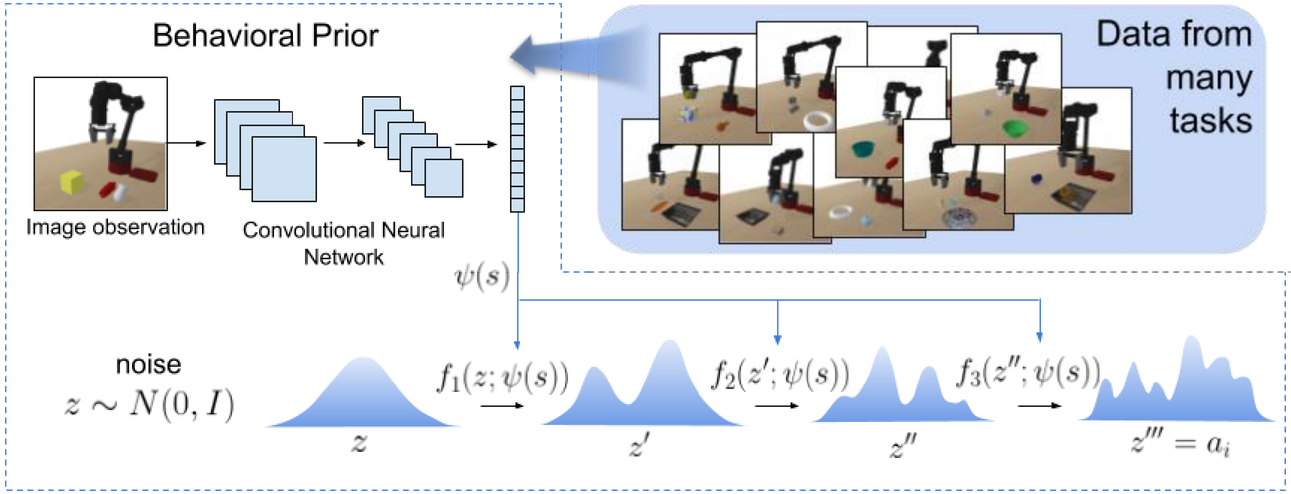
How to **bootstrap exploration** from experience?

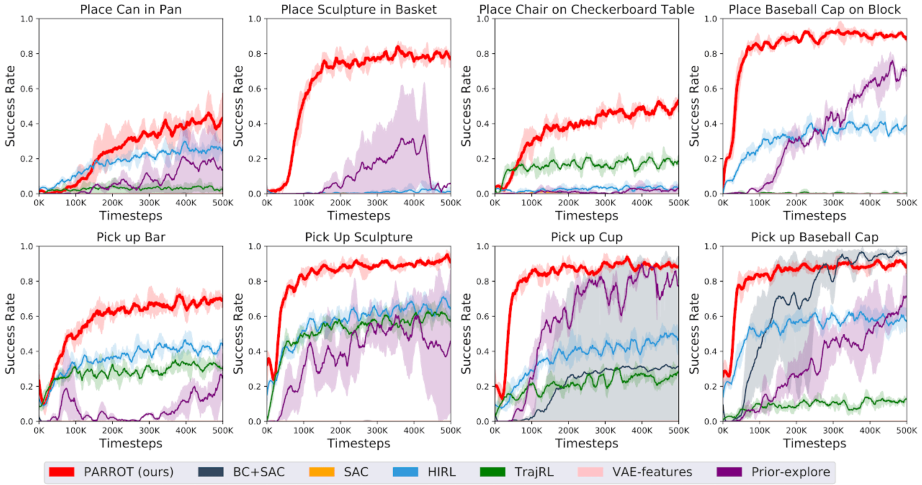
exploring from scratch

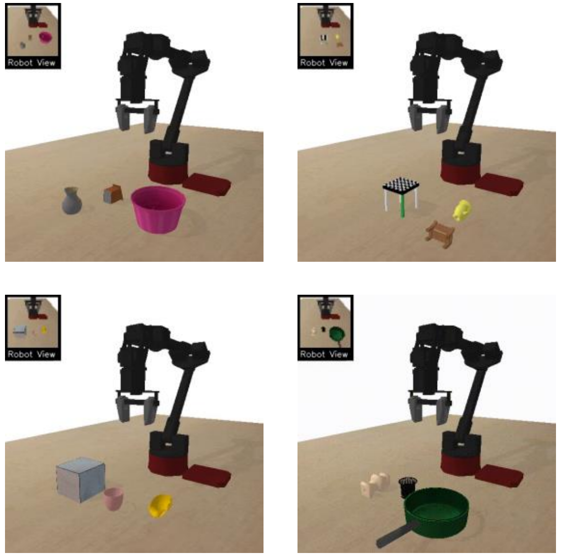


exploring from behavioral prior









Singh*, Hui*, Zhou, Yu, Rhinehart, Levine. **Parrot: Data-driven behavioral priors for reinforcement learning**. 2020.

This all seems really hard, what's the point?

Why is this interesting?

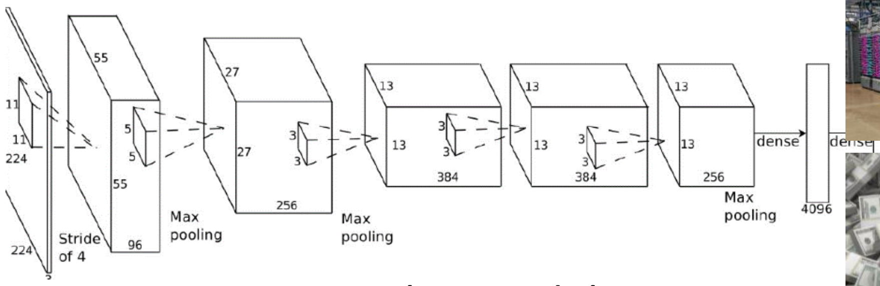
- It's exciting to see what solutions intelligent agents come up with
- Most exciting if they come up with something we don't expect
- This requires the world they inhabit to admit novel solutions
- This means that world must be complicated enough!

- To see interesting emergent behavior, we must train our systems in environments that actually require interesting emergent behavior!
- RL in the real world may be difficult, but it is also rewarding

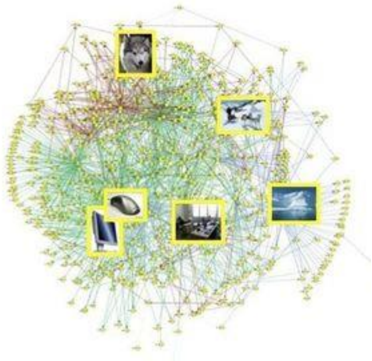
Reinforcement Learning as "Universal" Learning

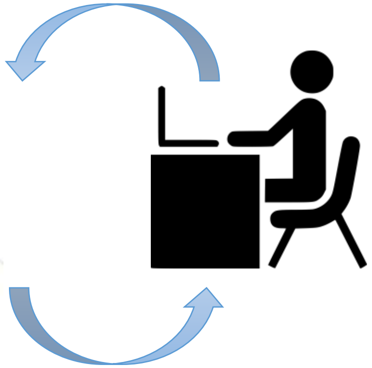
Large-scale machine learning

Why does deep learning work?



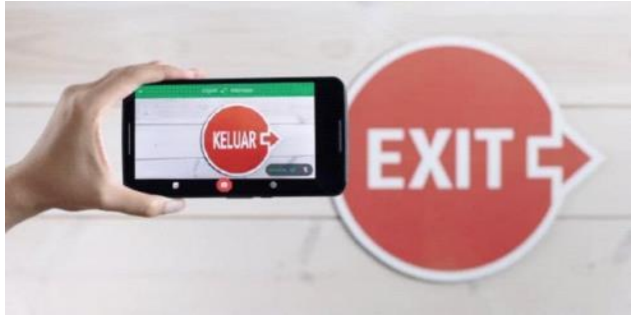
big models

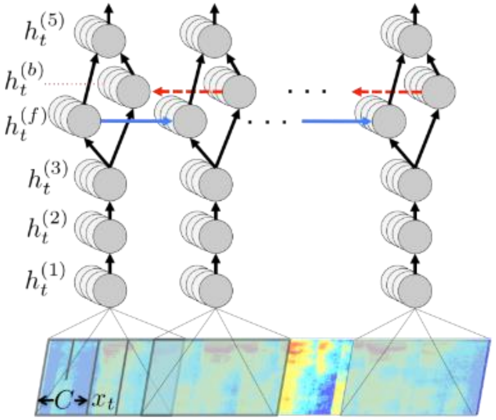




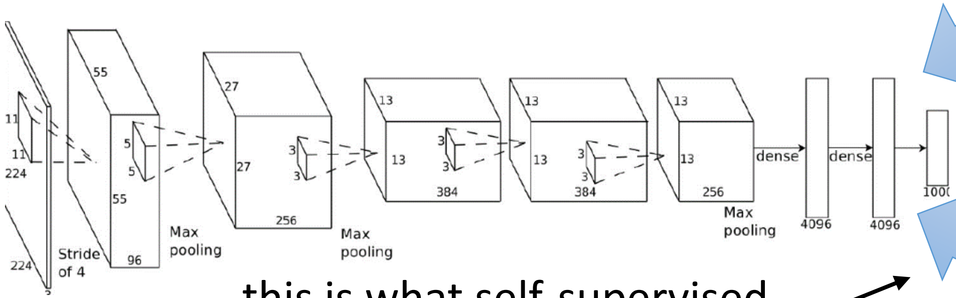
big datasets

<u>labeled</u>

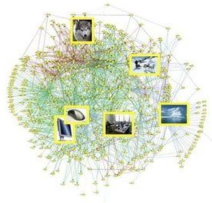


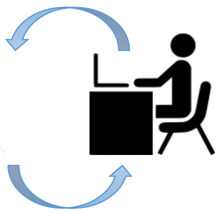


Reducing the supervision burden



this is what self-supervised learning is supposed to do





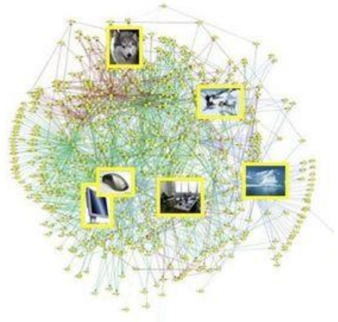
small labeled dataset

giant unlabeled garbage dataset (aka the Internet)

But then where does the knowledge come from?

"Classic" unsupervised learning: $p_\theta(\mathbf{x})$

(this is what, for example, large language models do)



Aside: perhaps this is why "prompting" large language models is such an art!

Stepping back a bit…

Why do we need **machine learning** anyway?

A postulate:

We need machine learning for one reason and one reason only – that's **to produce <u>adaptable and complex decisions</u>.**

Decision: how do I move my joints?

Decision: how do I steer the car?

What is the decision? The image label?

Usually not!

What happens with that label **afterwards**?

Is it used to tag a user's photo?

Detect an endangered animal in a camera trap?

these are **decisions**

they lead to **consequences**

Aside: why do we need **brains** anyway?

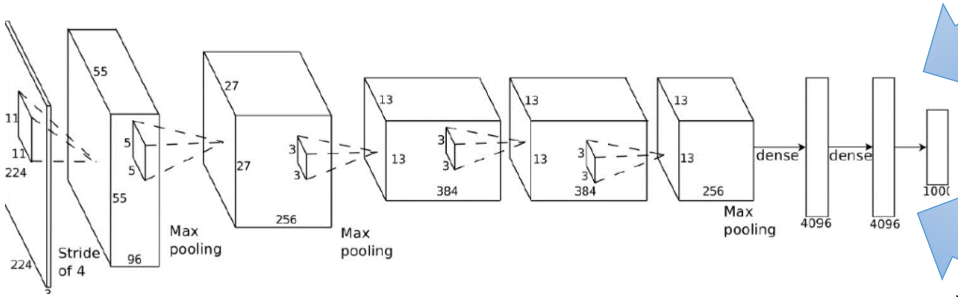
Daniel Wolpert

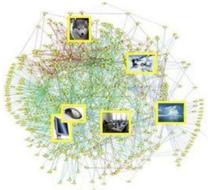
(knows quite a lot about brains)

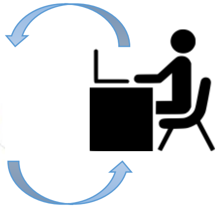
"We have a brain for one reason and one reason only -- that's **to produce adaptable and complex movements**. Movement is the only way we have affecting the world around us… I believe that to understand movement is to understand the whole brain."

Reinforcement learning as a way to use “cheap” (previously collected) data

this tells us what the **task** is
(e.g., reward function)



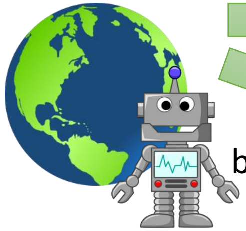




limited amount
of supervision

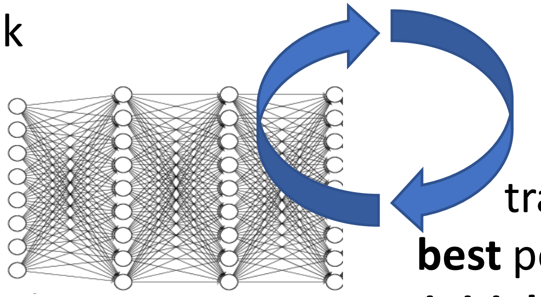
large, cheap dataset
of previously
collected data

this tells us **how the world works** (i.e., “dynamics”)



big datasets
from past
interaction

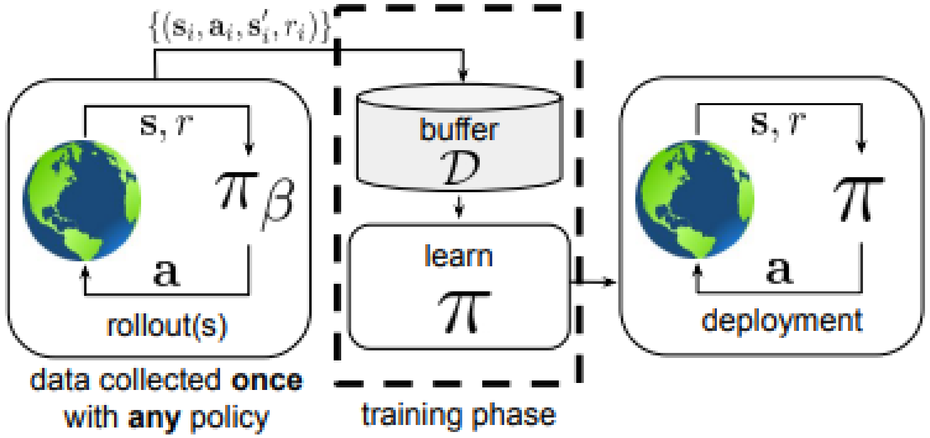
new task

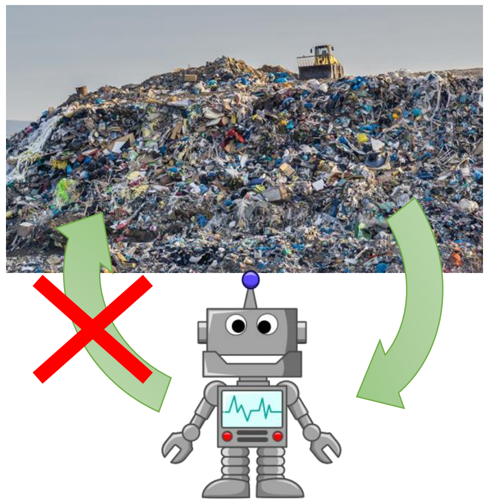


train the
best possible
initial model

The RL + data problem

offline reinforcement learning



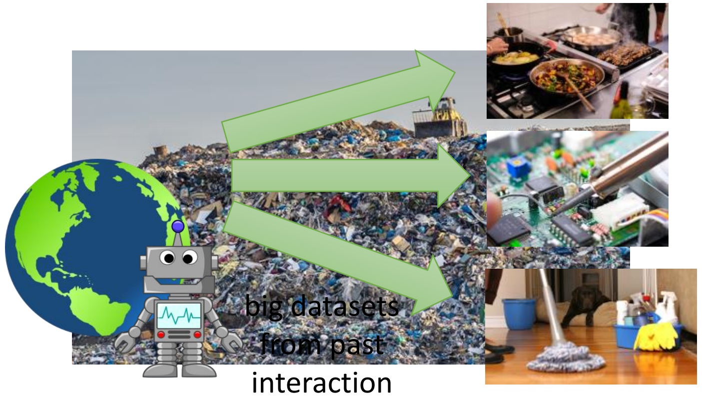


with naïve RL, this is a **costly** interactive process if done in the real world!

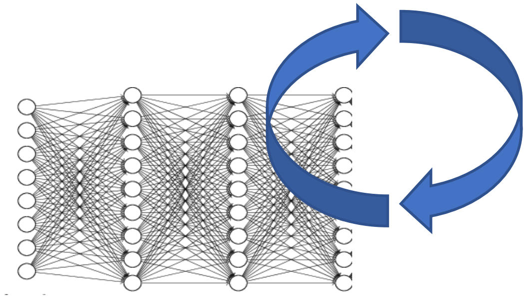
but self-supervised learning is about using **cheap** data that we already have lying around (in the garbage heap)!

The recipe

We need machine learning for one reason and one reason only – that's **to produce <u>adaptable and complex decisions</u>.**



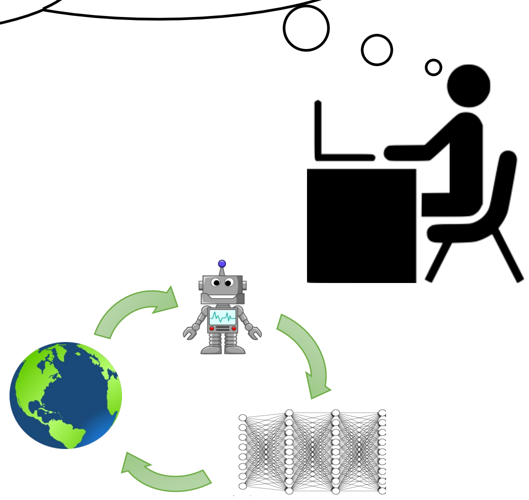
large dataset of diverse (but possibly low-quality) behavior



offline reinforcement learning

there are a few different choices here:

- human-defined skills
- goal-conditioned RL
- self-supervised skill discovery

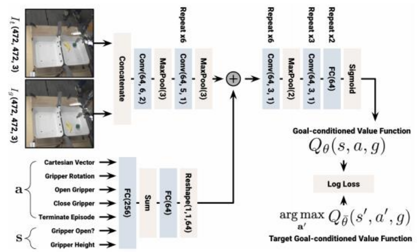


learning the downstream task

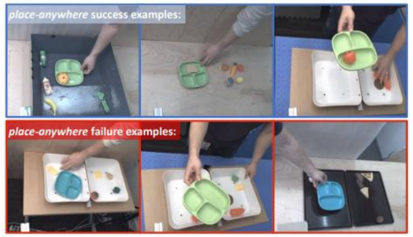
Can we learn from offline data without **well-defined tasks?**

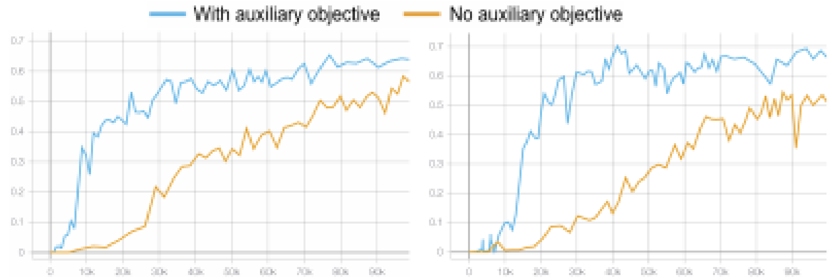
- No reward function at all, task is defined entirely using a **goal image**
- Uses a conservative offline RL method designed for goal-reaching, based on CQL
- Works very well as an **unsupervised pretraining** objective

1. Train **goal-conditioned** Q-function with offline RL



2. Finetune with a **task reward** and limited data

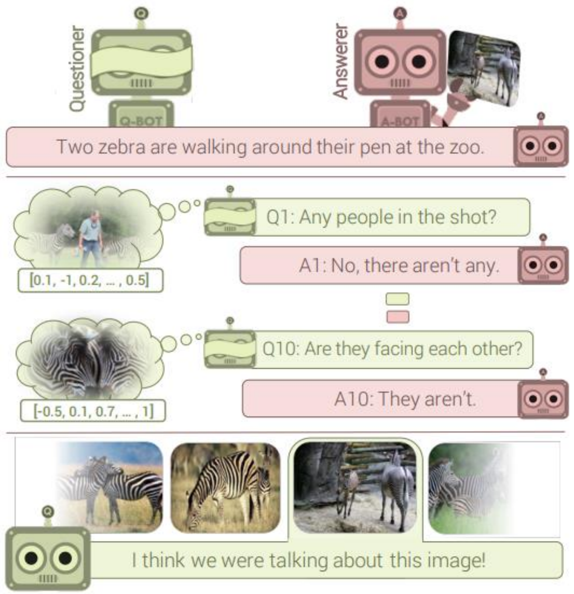




(c) Bottle grasping (d) Banana grasping

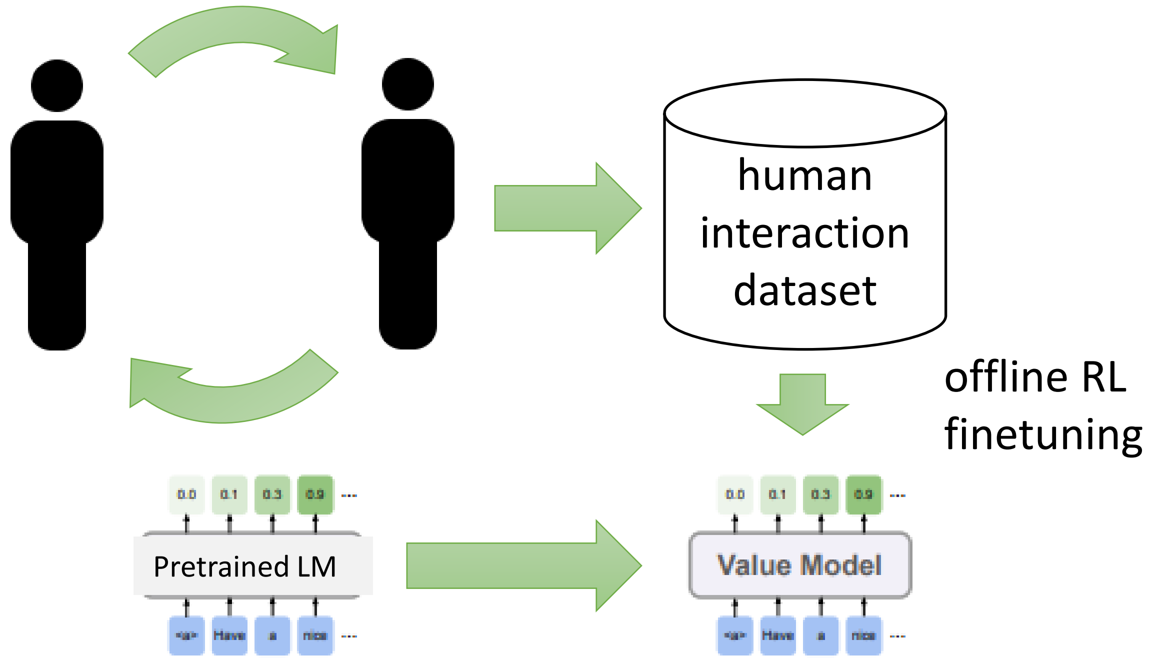
Chebotar, Hausman, Lu, Xiao, Kalashnikov, Varley, Irpan, Eysenbach, Julian, Finn, Levine. **Actionable Models: Unsupervised Offline Reinforcement Learning of Robotic Skills.** 2021.

Can offline RL train large language models?



Das et al. **Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning**. 2017.

Image Caption: *Tour buses are lined up on the street waiting for people.*
Questioner: how many buses?
Answerer: 2
Questioner: what color are buses?
Answerer: white and red
Questioner: how many people?
Answerer: 2
Questioner: what gender are people?
Answerer: 1 is male and 1 is female
Questioner: what are they wearing?
Answerer: 1 is wearing shorts and other is wearing shorts and shirt
Questioner: what color is their hair?
Answerer: dark brown



ILQL: Influencing speaker behavior with offline RL trained dialogue systems

Standard Reward Function:

Image Caption: *A man, with other people in the background, is sitting at a picnic table with a cell phone in front of him while he uses a fork and knife to cut a dessert on a plate.*
Questioner: is man young?
Answerer: yes
Questioner: is he wearing glasses?
Answerer: no
Questioner: is there food on table?
Answerer: yes
Questioner: is there food on table?
Answerer: yes

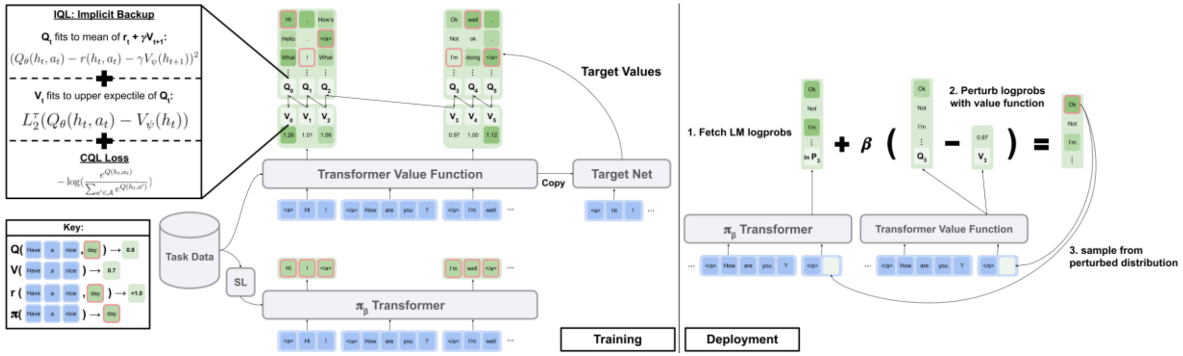
Yes/No Penalty:

Image Caption: *A person flying a kite over a sand dune.*
Questioner: what color is kite?
Answerer: it's black and white
Questioner: is person male or female?
Answerer: can't tell
Questioner: what is person wearing?
Answerer: i can't tell
Questioner: what color is sand?
Answerer: dark brown
Questioner: is person flying kite flying in air?
Answerer: yes
Questioner: what color is kite?
Answerer: black

Conservative Yes/No Penalty:

Image Caption: *Tour buses are lined up on the street waiting for people.*
Questioner: how many buses?
Answerer: 2
Questioner: what color are buses?
Answerer: white and red
Questioner: how many people?
Answerer: 2
Questioner: what gender are people?
Answerer: 1 is male and 1 is female
Questioner: what are they wearing?
Answerer: 1 is wearing shorts and other is wearing shorts and shirt
Questioner: what color is their hair?
Answerer: dark brown

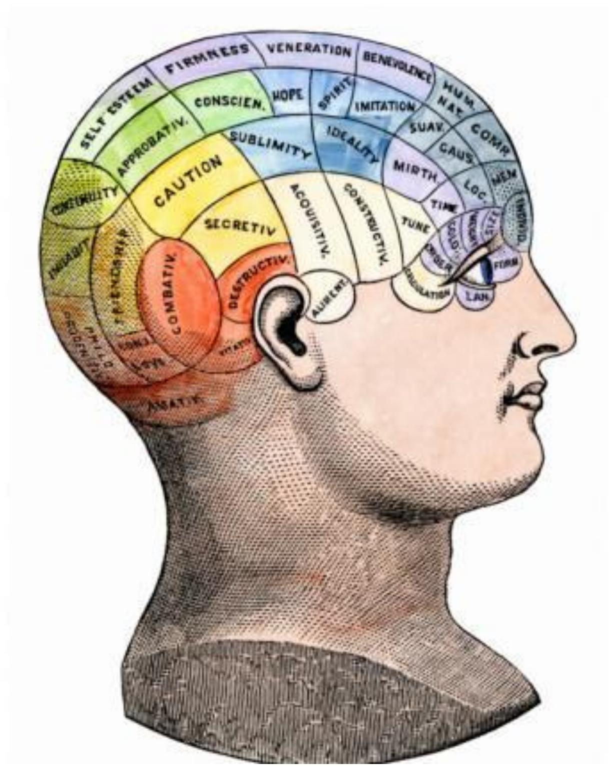
We present a set of example dialogues produced by our Visual Dialogue agents (in green) optimized for different rewards. Each of our reward functions, from left to right, penalizes the asking of yes/no questions to increasing degrees. As the yes/no penalty increases, our agents learn to ask fewer such questions.



Snell, Kostrikov, Su, Yang, Levine. **Offline RL for Natural Language Generation with Implicit Language Q Learning**. 2022.

Back to the Bigger Picture

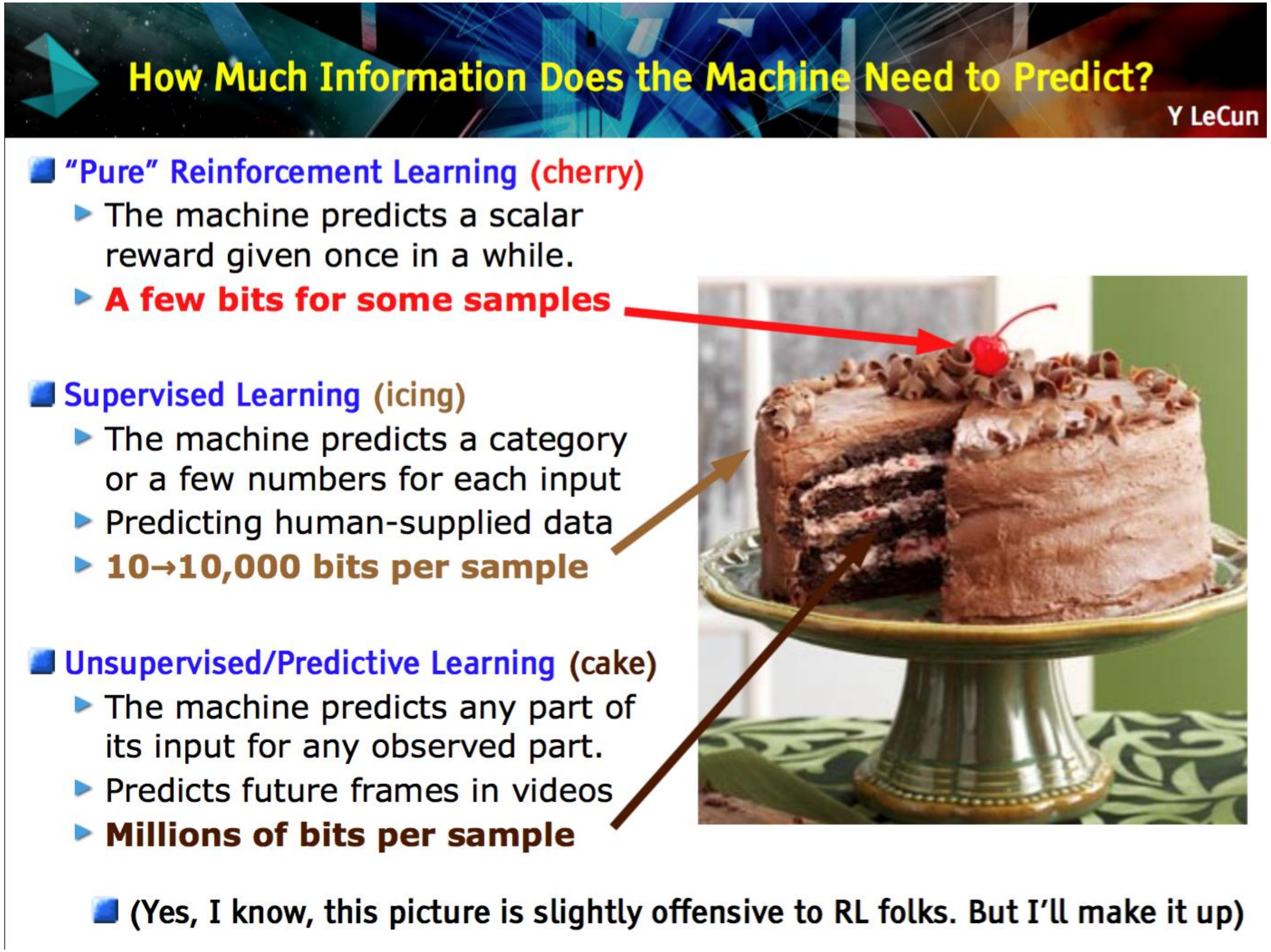
Learning as the basis of intelligence



- Reinforcement learning = can reason about decision making
- Deep models = allows RL algorithms to learn and represent complex input-output mappings

Deep models are what allow reinforcement learning algorithms to solve complex problems end to end!

What is missing?



Where does the *signal* come from?

- Yann LeCun’s cake
  - Unsupervised or self-supervised learning
  - Model learning (predict the future)
  - Generative modeling of the world
  - Lots to do even before you accomplish your goal!
- Imitation & understanding other agents
  - We are social animals, and we have culture – for a reason!
- The giant value backup
  - All it takes is one +1
- All of the above

How should we answer these questions?

- Pick the right problems!
  - Ask: does this have a **chance** of solving an important problem?
  - Optimism in the face of uncertainty is a good exploration strategy!
- Don’t be afraid to change the problem statement
  - Many of these challenges won’t be met by iterating on existing benchmarks!
- Applications matter
  - Sometimes applying methods to realistic and challenging real-world domains can teach us a lot about the important things that are missing
  - RL has a long history of disregarding this fact
- Think big and start small