Information-Theoretic Exploration, Challenges and Open Problems

CS 285: Deep Reinforcement Learning, Decision Making, and Control

Sergey Levine

Class Notes

1. Today: concluding lecture
2. Wednesday: guest lecture, Ofir Nachum
3. Next week: guest lecture, Chelsea Finn
4. Next next week: guest lectures, Karol Hausman, Karen Liu
5. Please attend the guest lectures!!

Today's Lecture

1. Part 1: information theoretic exploration – how can we learn **without** any reward function at all?
2. Part 2: challenges and open problems in deep RL, takeaways and last-minute gift ideas

- Goals:
  - Provide high-level overview of information theoretic exploration and unsupervised reinforcement learning
  - Briefly summarize tradeoffs of current deep RL algorithms
  - Provide some perspective on current open problems and challenges

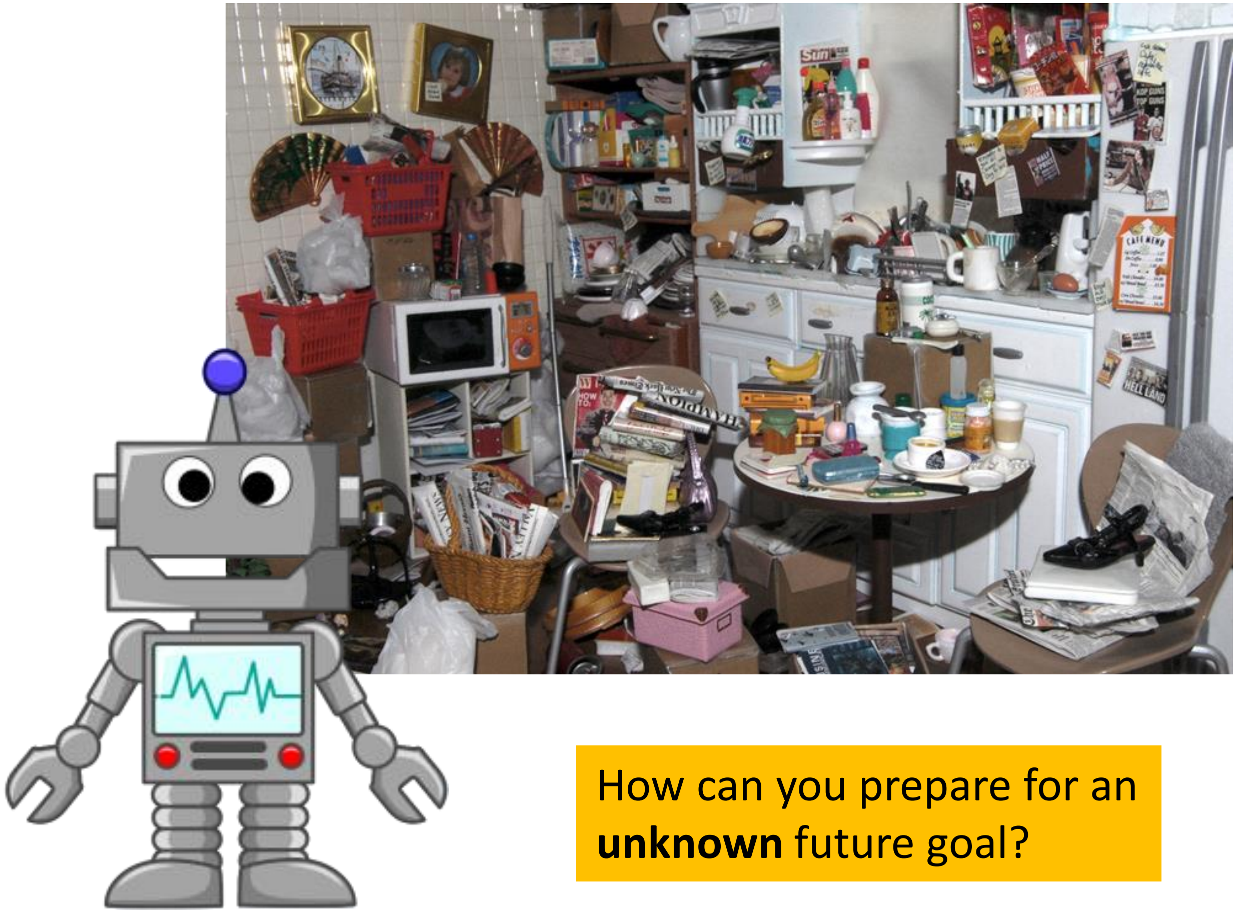
Unsupervised learning of diverse behaviors

What if we want to recover diverse behavior **without any reward function at all**?

Why?

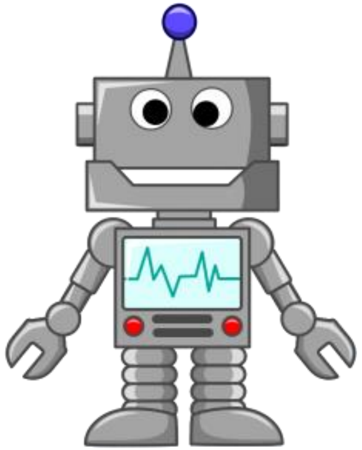
- *Learn skills without supervision, then use them to accomplish goals*
- *Learn sub-skills to use with hierarchical reinforcement learning*
- *Explore the space of possible behaviors*

An Example Scenario



How can you prepare for an **unknown** future goal?

training time: unsupervised



In this lecture…

- ➢ Definitions & concepts from information theory
- ➢ Learning without a reward function by reaching goals
- ➢ Beyond state covering: covering the *space of skills*
- ➢ Using unsupervised reinforcement learning for meta-learning

In this lecture...

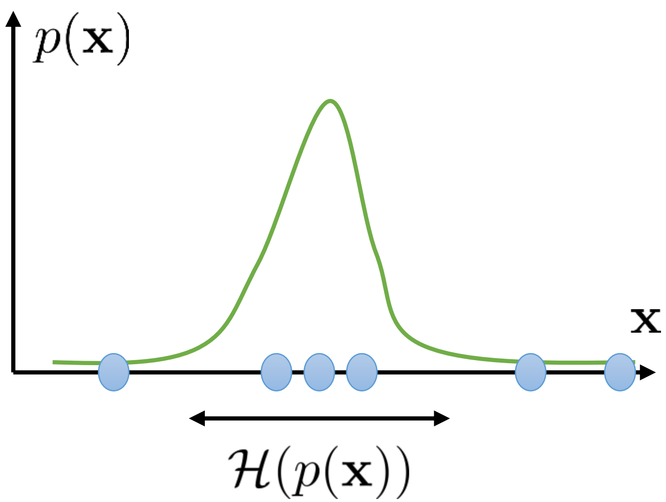
- Definitions & concepts from information theory
- Learning without a reward function by reaching goals
- Beyond state covering: covering the *space of skills*
- Using unsupervised reinforcement learning for meta-learning

Some useful identities

$p(\mathbf{x})$ distribution (e.g., over observations $\mathbf{x}$)

$$\mathcal{H}(p(\mathbf{x})) = -E_{\mathbf{x}\sim p(\mathbf{x})}[\log p(\mathbf{x})]$$

entropy – how "broad" $p(\mathbf{x})$ is



Some useful identities

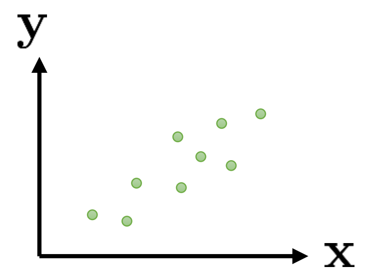
$$\mathcal{H}(p(\mathbf{x})) = -E_{\mathbf{x}\sim p(\mathbf{x})}[\log p(\mathbf{x})]$$

entropy – how "broad" $p(\mathbf{x})$ is

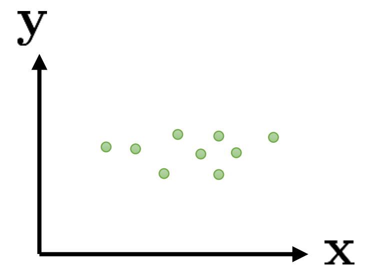
$$\mathcal{I}(\mathbf{x};\mathbf{y}) = D_{\mathrm{KL}}(p(\mathbf{x},\mathbf{y}) \| p(\mathbf{x})p(\mathbf{y}))$$

$$= E_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\left[\log \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}\right]$$

$$= \mathcal{H}(p(\mathbf{y})) - \mathcal{H}(p(\mathbf{y}|\mathbf{x}))$$



high MI: $\mathbf{x}$ and $\mathbf{y}$ are *dependent*



low MI: $\mathbf{x}$ and $\mathbf{y}$ are *independent*

Information theoretic quantities in RL

$\pi(\mathbf{s})$ state *marginal* distribution of policy π

quantifies *coverage*

$\mathcal{H}(\pi(\mathbf{s}))$ state *marginal* entropy of policy π

example of mutual information: "empowerment" (Polani et al.)

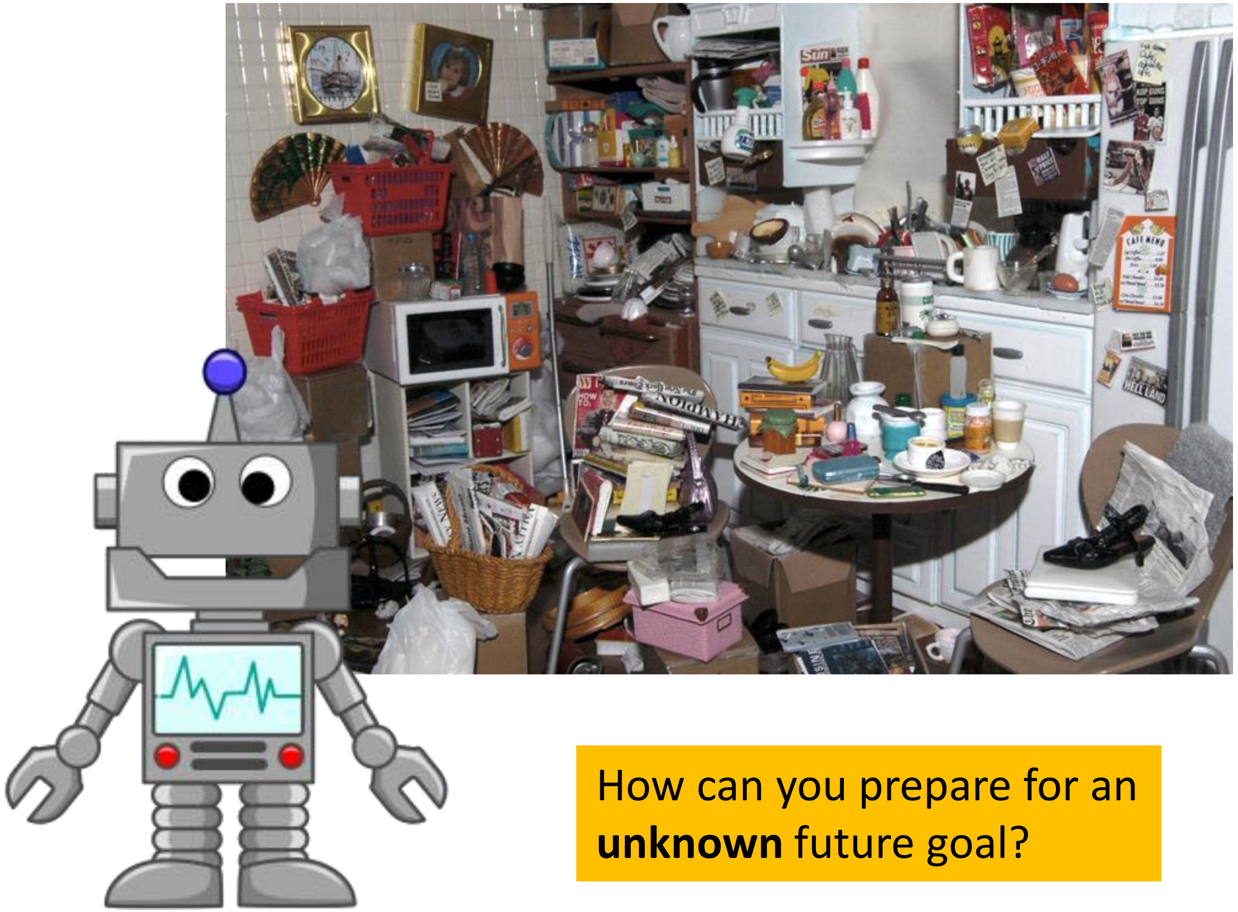
$$\mathcal{I}(\mathbf{s}_{t+1};\mathbf{a}_t) = \mathcal{H}(\mathbf{s}_{t+1}) - \mathcal{H}(\mathbf{s}_{t+1}|\mathbf{a}_t)$$

can be viewed as quantifying "control authority" in an information-theoretic way

In this lecture...

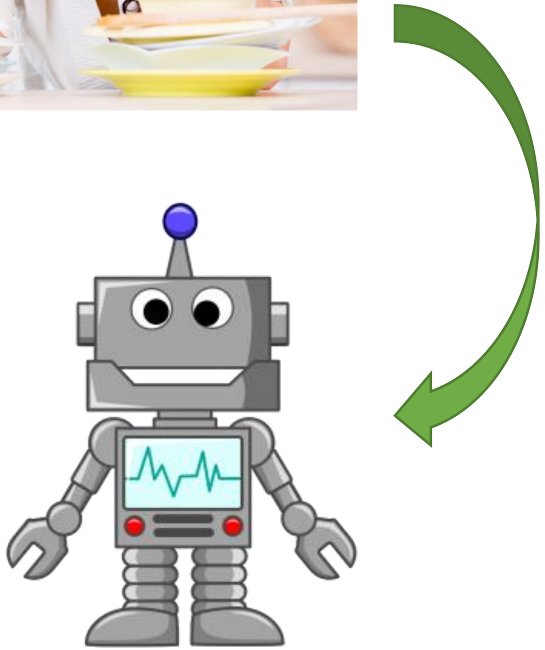
- Definitions & concepts from information theory
- Learning without a reward function by reaching goals
- Beyond state covering: covering the *space of skills*
- Using unsupervised reinforcement learning for meta-learning

An Example Scenario



How can you prepare for an **unknown** future goal?

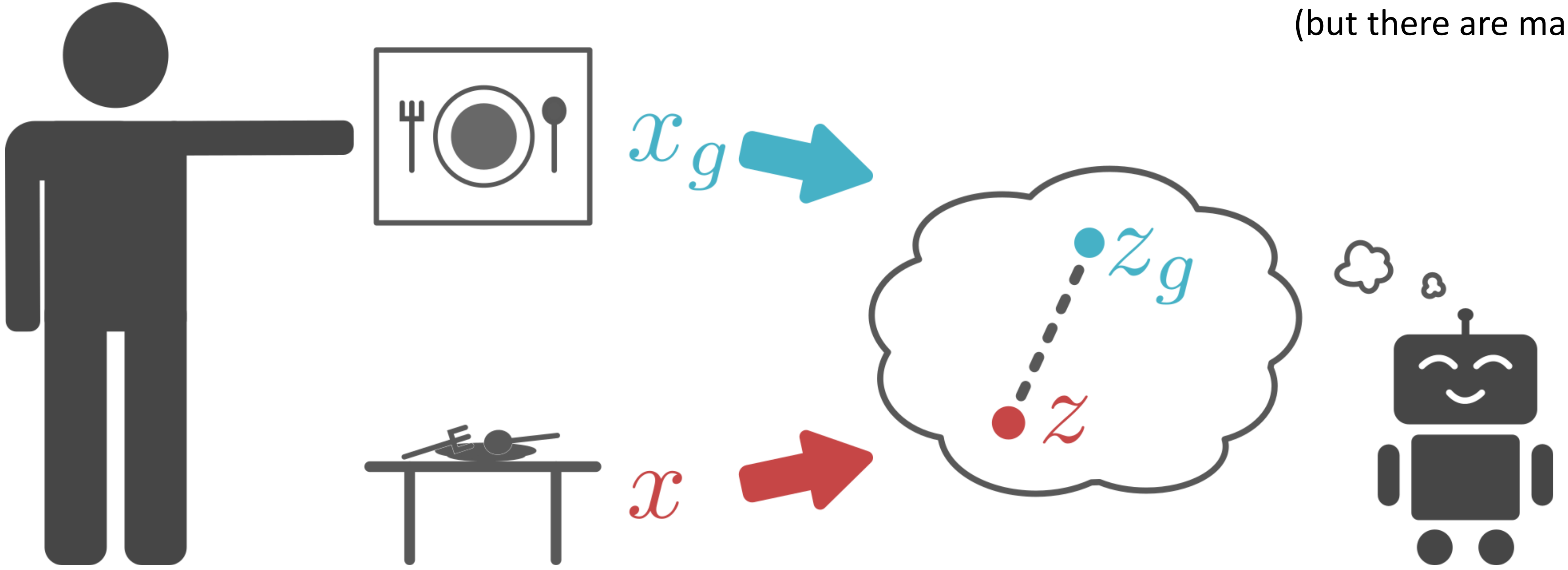
training time: unsupervised

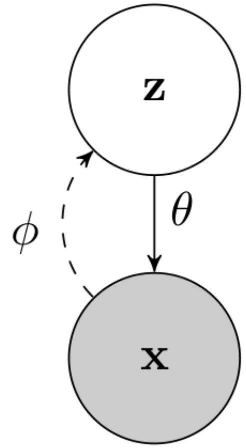


Learn without any rewards at all

VAE (Kingma & Welling '13)

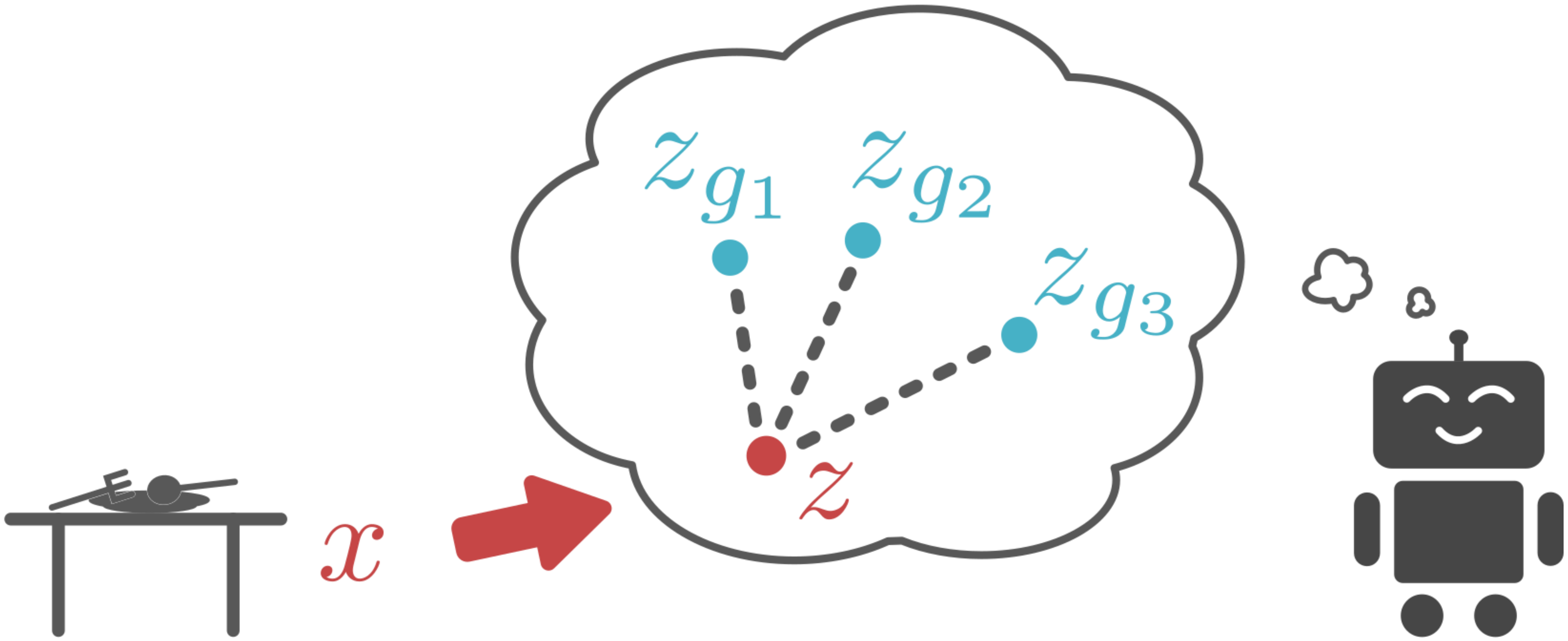
(but there are many other choices)





Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. '18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** '19

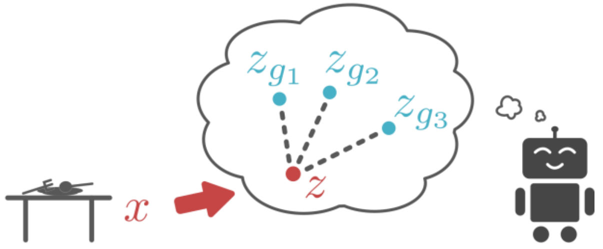
Learn without any rewards at all

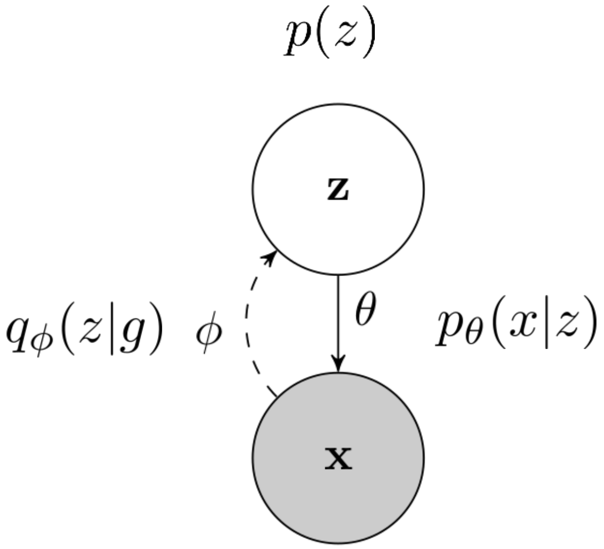


Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. ’18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** ‘19

Learn without any rewards at all

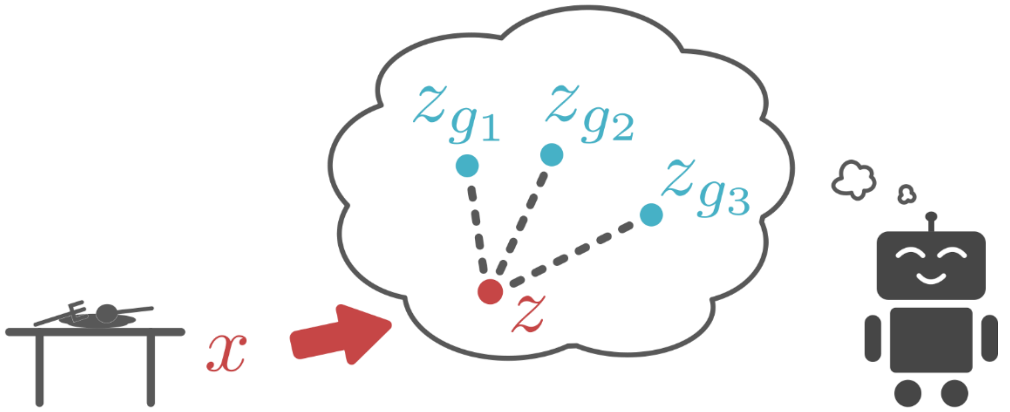
1. Propose goal: $z_g \sim p(z)$, $x_g \sim p_\theta(x_g|z_g)$
2. Attempt to reach goal using $\pi(a|x, x_g)$, reach $\bar{x}$
3. Use data to update π
4. Use data to update $p_\theta(x_g|z_g)$, $q_\phi(z_g|x_g)$

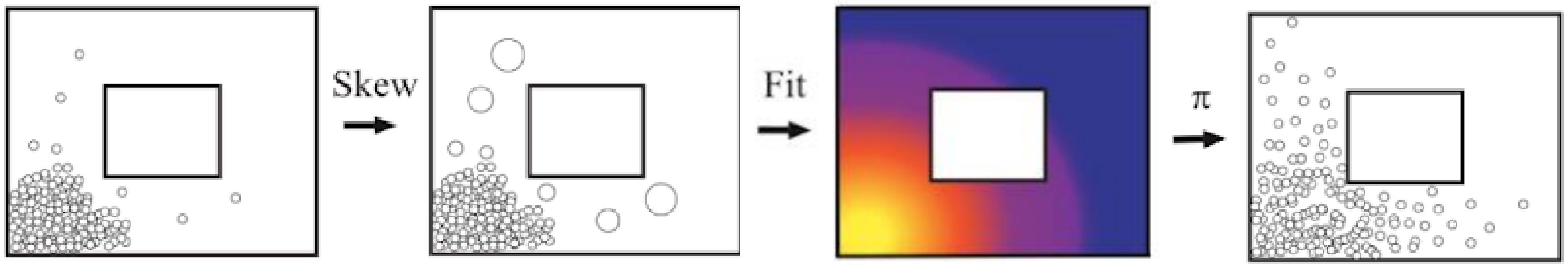




Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. '18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** '19

How do we get diverse goals?

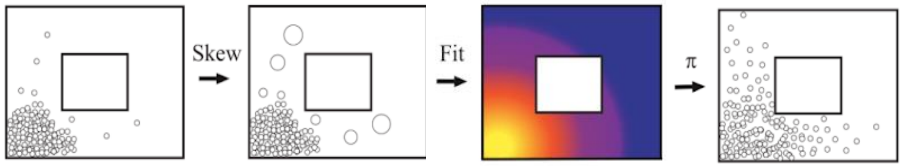




Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. '18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** '19

How do we get diverse goals?

1. Propose goal: $z_g \sim p(z)$, $x_g \sim p_\theta(x_g|z_g)$
2. Attempt to reach goal using $\pi(a|x, x_g)$, reach $\bar{x}$
3. Use data to update π
4. Use data to update $p_\theta(x_g|z_g)$, $q_\phi(z_g|x_g)$

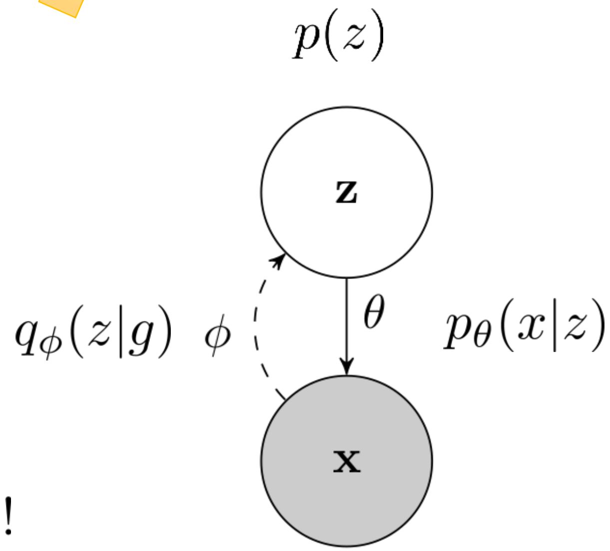


standard MLE: $\theta, \phi \leftarrow \arg\max_{\theta,\phi} E[\log p(\bar{x})]$

weighted MLE: $\theta, \phi \leftarrow \arg\max_{\theta,\phi} E[w(\bar{x}) \log p(\bar{x})]$

$w(\bar{x}) = p_\theta(\bar{x})^\alpha$

key result: for any $\alpha \in [-1, 0)$, entropy $\mathcal{H}(p_\theta(x))$ increases!



Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. '18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** '19

How do we get diverse goals?

what is the objective?

$$\max \mathcal{H}(p(G)) - \mathcal{H}(p(G|S))$$

goals get higher
entropy due to Skew-Fit

what does RL do?

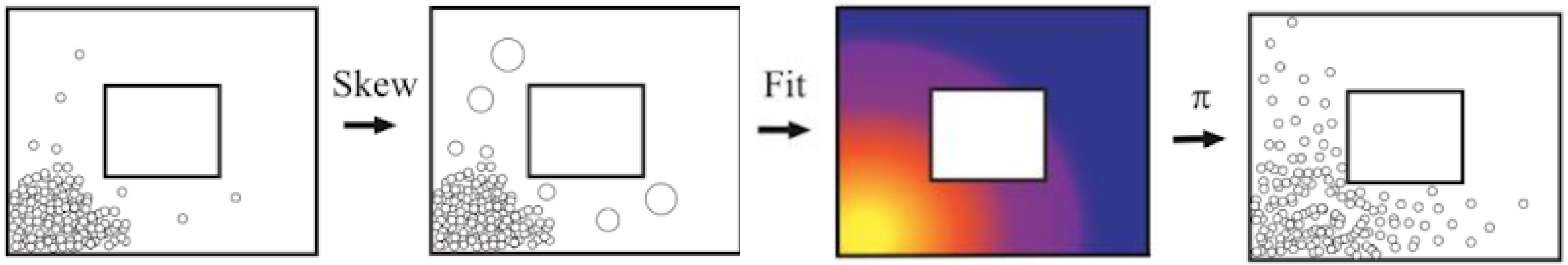
$\pi(a|S,G)$ trained to reach goal G

as π gets better, final state S gets close to G

that means $p(G|S)$ becomes more deterministic!

goal final state

$$w(\bar{x}) = p_\theta(\bar{x})^\alpha$$
$$\alpha \in [-1, 0)$$



Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. ’18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** ‘19

How do we get diverse goals?

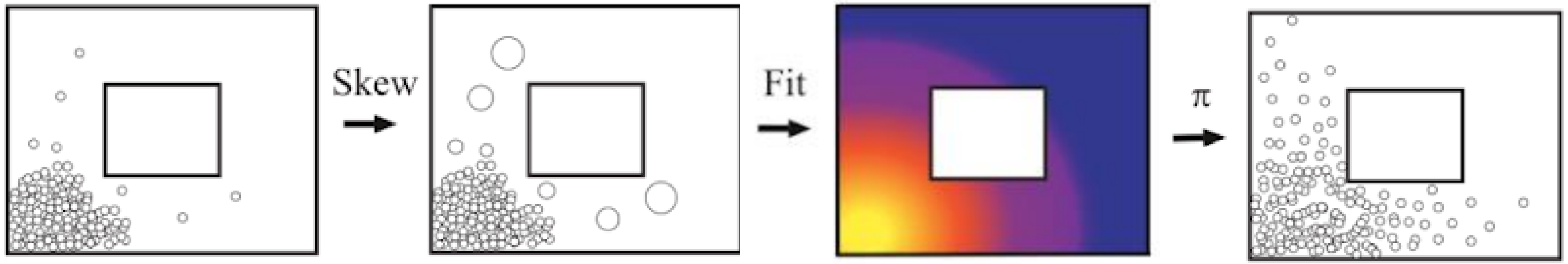
what is the objective?

$$\max \mathcal{H}(p(G)) - \mathcal{H}(p(G|S)) = \max \mathcal{I}(S;G)$$

maximizing mutual information between S and G leads to

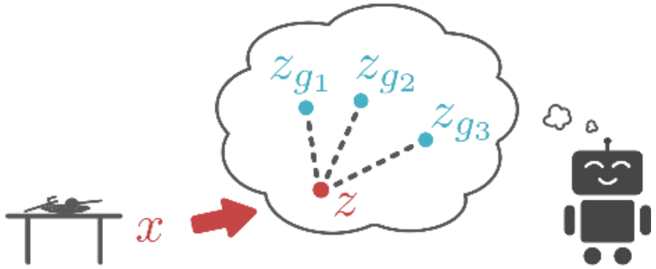
good exploration (state coverage) – $\mathcal{H}(p(G))$

effective goal reaching – $\mathcal{H}(p(G|S))$



Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. ’18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** ‘19

Reinforcement learning with *imagined* goals



Nair*, Pong*, Bahl, Dalal, Lin, L. **Visual Reinforcement Learning with Imagined Goals**. '18
Dalal*, Pong*, Lin*, Nair, Bahl, Levine. **Skew-Fit: State-Covering Self-Supervised Reinforcement Learning.** '19

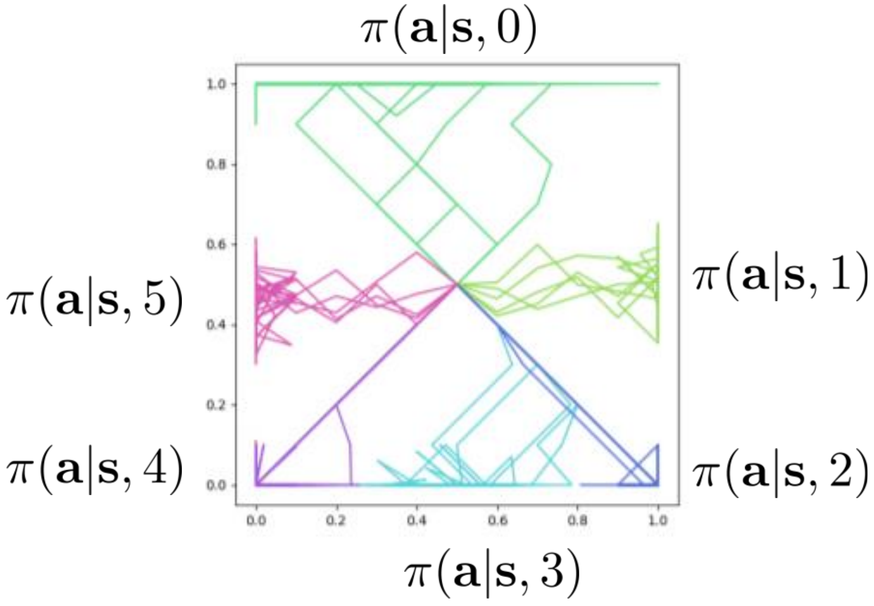
In this lecture…

- ➢ Definitions & concepts from information theory
- ➢ Learning without a reward function by reaching goals
- ➢ Beyond state covering: covering the *space of skills*
- ➢ Using unsupervised reinforcement learning for meta-learning

Learning diverse skills

$$\pi(\mathbf{a}|\mathbf{s}, z)$$

task index



Why can't we just use MaxEnt RL or goal-reaching?

1. **action** entropy is not the same as **state** entropy

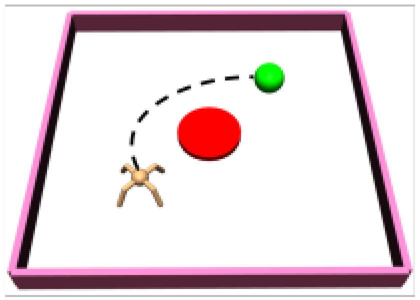
   agent can take very different actions, but land in similar states

2. Reaching diverse **goals** is not the same as performing diverse **tasks**

   not all behaviors can be captured by **goal-reaching**

3. MaxEnt policies are stochastic, but not always **controllable**

   intuitively, we want **low** diversity for a fixed z, high diversity *across* z*'s*



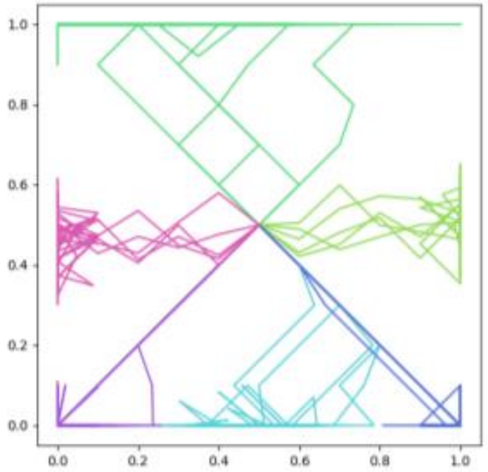
Intuition: different **skills** should visit different **state-space regions**

Eysenbach, Gupta, Ibarz, Levine. **Diversity is All You Need.**

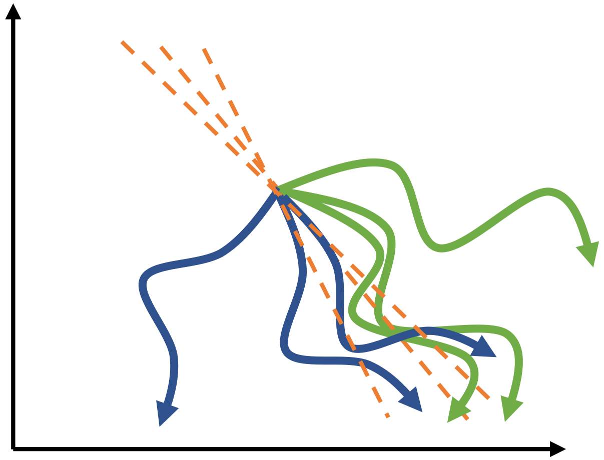
Diversity-promoting reward function

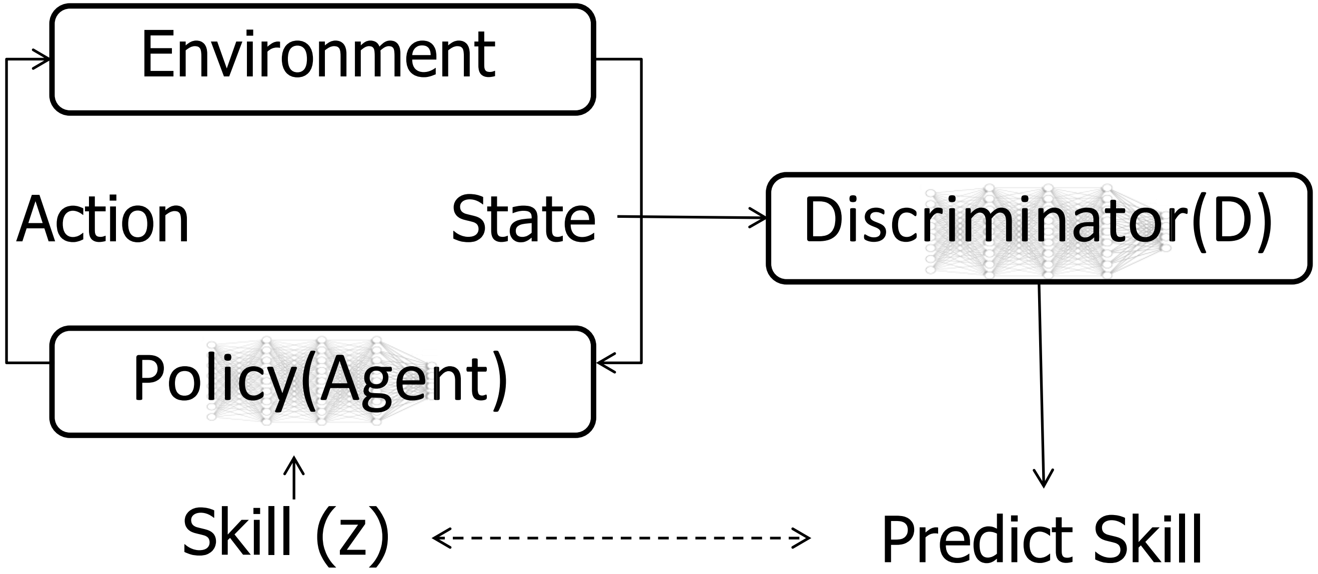
$$\pi(\mathbf{a}|\mathbf{s}, z) = \arg\max_{\pi} \sum_{z} E_{\mathbf{s}\sim\pi(\mathbf{s}|z)}[r(\mathbf{s}, z)]$$

reward states that are unlikely for other $z' \neq z$



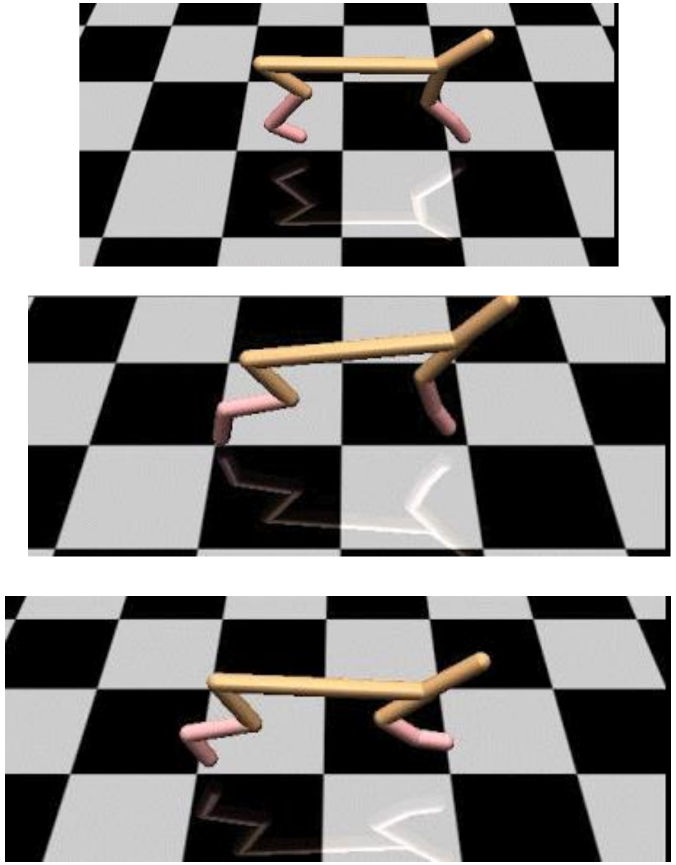
$$r(\mathbf{s}, z) = \log p(z|\mathbf{s})$$



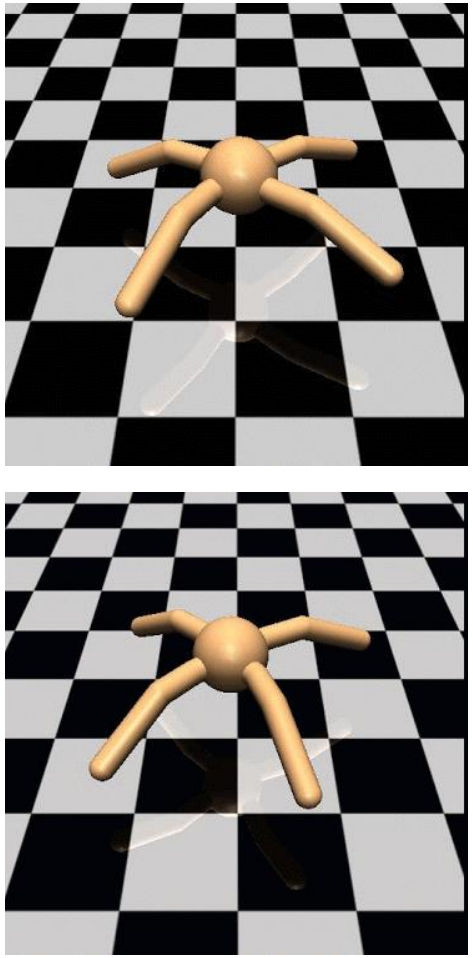


Eysenbach, Gupta, Ibarz, Levine. **Diversity is All You Need.**

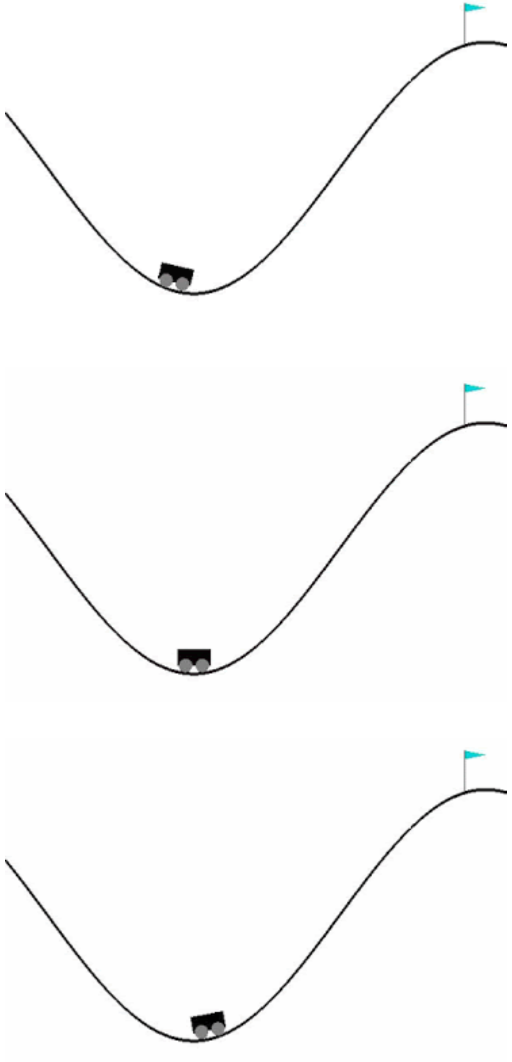
Examples of learned tasks



Cheetah



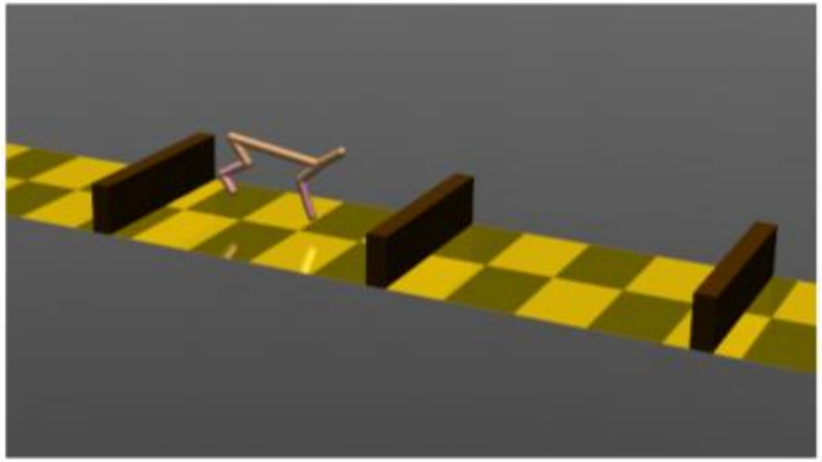
Ant



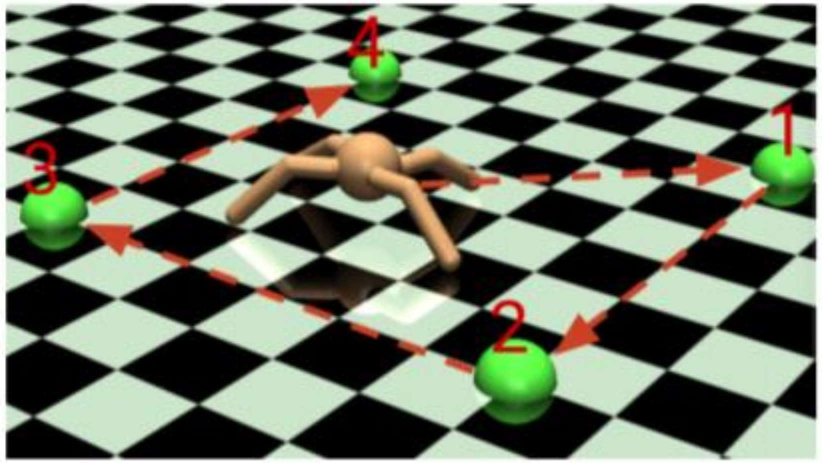
Mountain car

Eysenbach, Gupta, Ibarz, Levine. **Diversity is All You Need.**

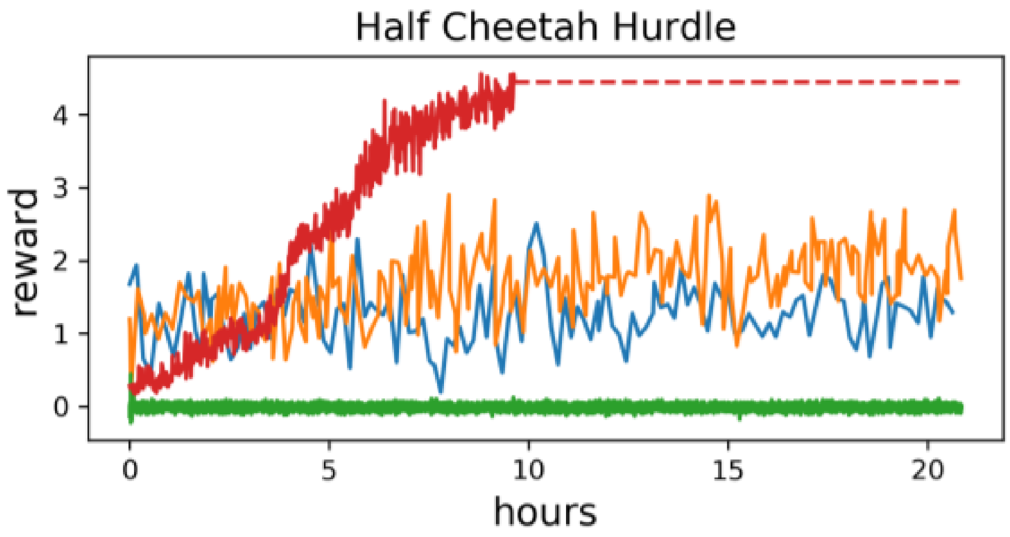
Results: hierarchical RL

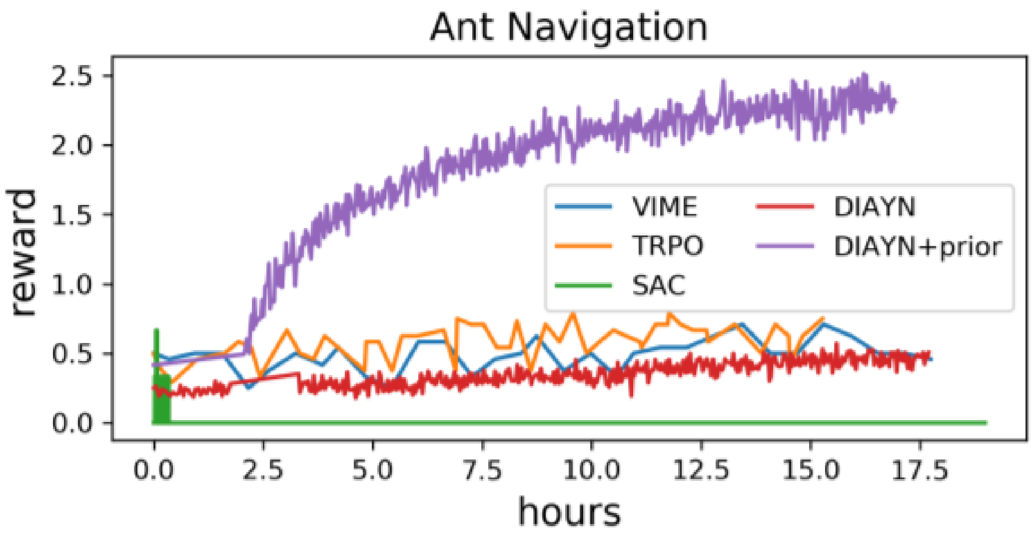


Cheetah Hurdle



Ant Navigation





Eysenbach, Gupta, Ibarz, Levine. **Diversity is All You Need.**

A connection to mutual information

$$\pi(\mathbf{a}|\mathbf{s}, z) = \arg\max_{\pi} \sum_{z} E_{\mathbf{s}\sim\pi(\mathbf{s}|z)}[r(\mathbf{s}, z)]$$

$$r(\mathbf{s}, z) = \log p(z|\mathbf{s})$$

$$I(z, \mathbf{s}) = H(z) - H(z|s)$$

maximized by using uniform prior $p(z)$

minimized by maximizing $\log p(z|\mathbf{s})$

Eysenbach, Gupta, Ibarz, Levine. **Diversity is All You Need.**

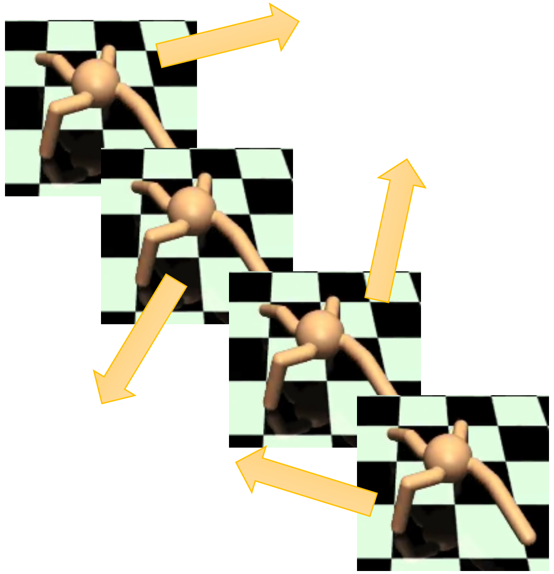
See also: Gregor et al. **Variational Intrinsic Control.** 2016

In this lecture...

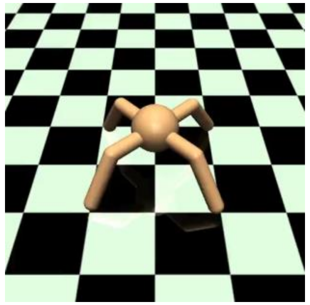
- Definitions & concepts from information theory
- Learning without a reward function by reaching goals
- Beyond state covering: covering the *space of skills*
- Using unsupervised reinforcement learning for meta-learning

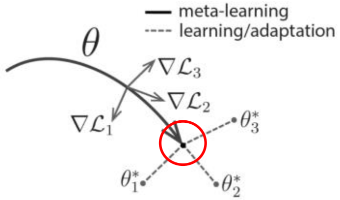
Aside: Meta-Overfitting

- Meta learning requires task distributions
- When there are too few meta-training tasks, we can *meta-overfit*
- Specifying task distributions is hard, especially for meta-RL!
- Can we propose tasks *automatically*?

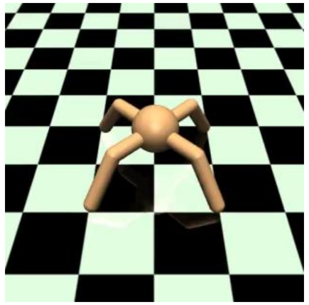


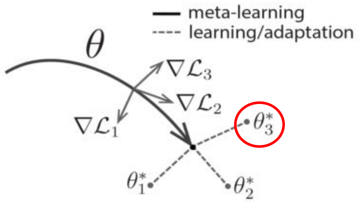
after MAML training





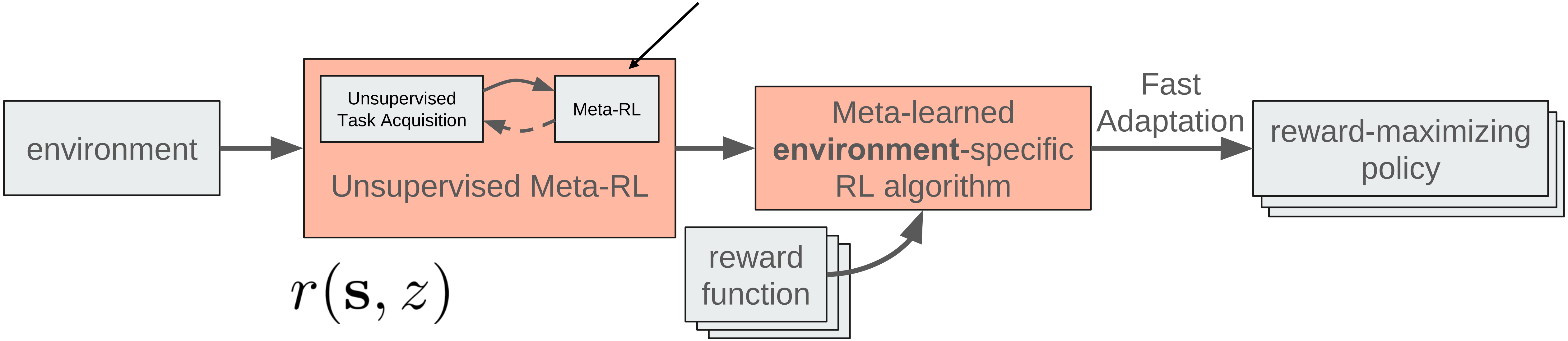
after 1 gradient step





A General Recipe for Unsupervised Meta-RL

$$\max_{\pi} E_z[E_{\pi'_z}[r(\mathbf{s}, z)]] \text{ s.t. } \pi'_z = \text{Adapt}(\pi, r(\mathbf{s}, z))$$



$$r(\mathbf{s}, z)$$

reward for task z

$$I(z, \mathbf{s}) = H(z) - H(z|s)$$

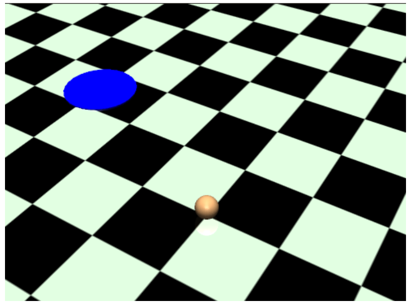
difference from before: result is *not* $\pi(\mathbf{a}|\mathbf{s}, z)$!

result is a model that can learn (quickly) from rewards!

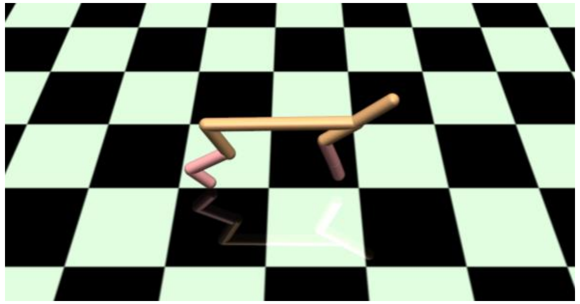
Gupta, Eysenbach, Finn, Levine. **Unsupervised Meta-Learning for Reinforcement Learning.**

Does it work?

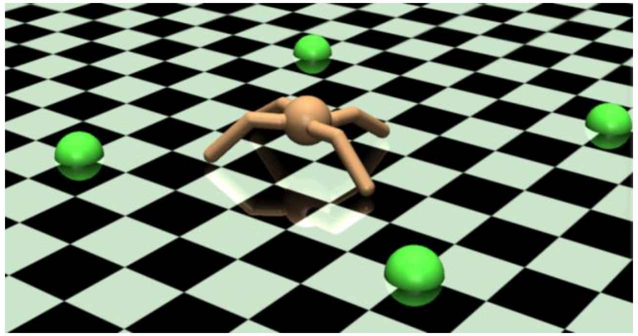
2D Navigation

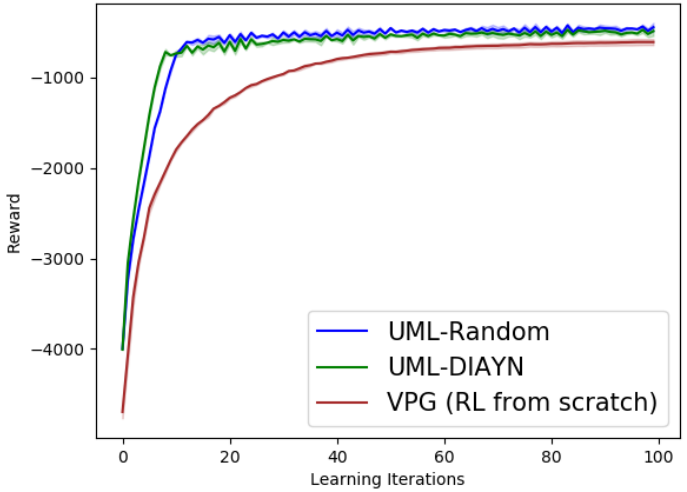


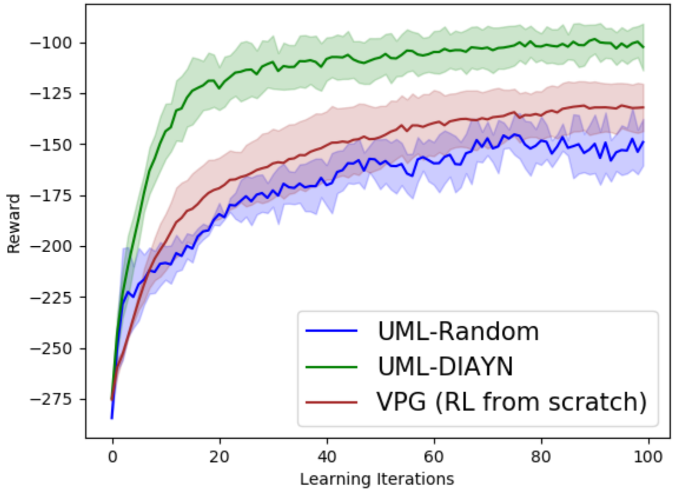
Cheetah

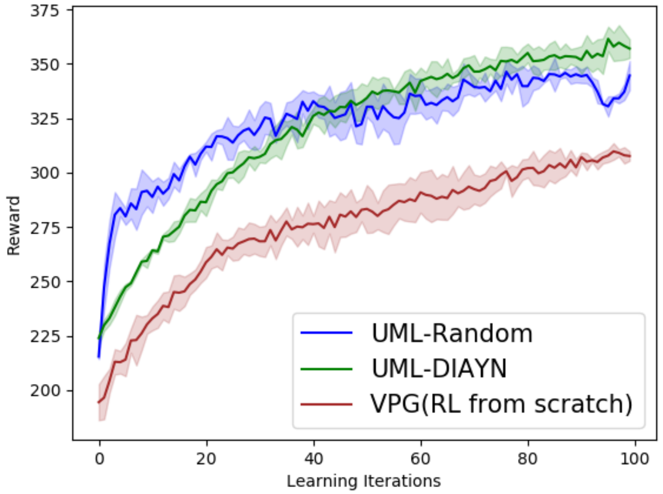


Ant









Meta-test performance with rewards

Gupta, Eysenbach, Finn, Levine. **Unsupervised Meta-Learning for Reinforcement Learning.**

In this lecture…

- ➢ Definitions & concepts from information theory
- ➢ Learning without a reward function by reaching goals
- ➢ Beyond state covering: covering the *space of skills*
- ➢ Using unsupervised reinforcement learning for meta-learning

Break

Challenges in Deep Reinforcement Learning

What’s the problem?

Challenges with **core algorithms**:

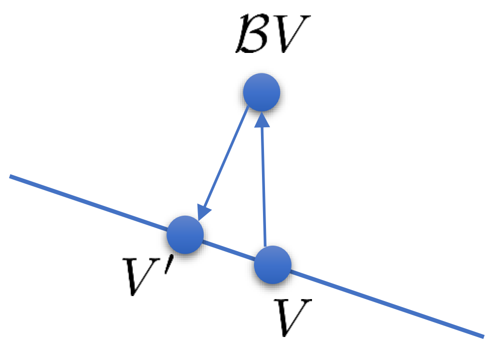
- Stability: does your algorithm converge?
- Efficiency: how long does it take to converge? (how many samples)
- Generalization: after it converges, does it generalize?

Challenges with **assumptions**:

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

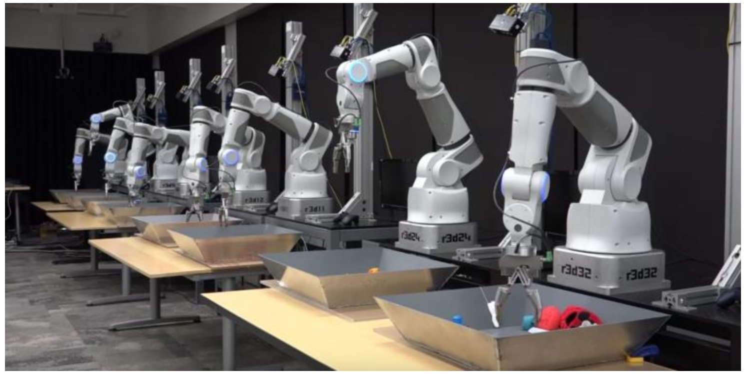
Stability and hyperparameter tuning

- 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



The challenge with hyperparameters

- Can't run hyperparameter sweeps in the real world
  - How representative is your simulator? Usually the answer is "not very"
- Actual sample complexity = time to run algorithm x number of runs to sweep
  - In effect stochastic search + gradient-based optimization
- Can we develop more stable algorithms that are less sensitive to hyperparameters?



What can we do?

- Algorithms with favorable improvement and convergence properties
  - Trust region policy optimization [Schulman et al. ‘16]
  - Safe reinforcement learning, High-confidence policy improvement [Thomas ‘15]
- Algorithms that adaptively adjust parameters
  - Q-Prop [Gu et al. ‘17]: adaptively adjust strength of control variate/baseline
- More research needed here!
- Not great for beating benchmarks, but absolutely essential to make RL a viable tool for real-world problems

Sample Complexity

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)

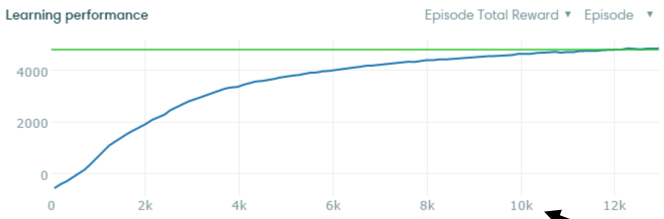
10x

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

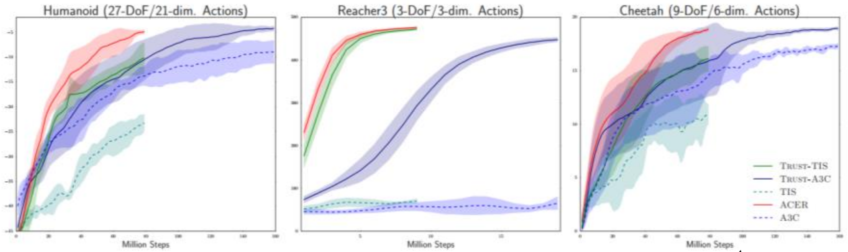
Evolution Strategies as a Scalable Alternative to Reinforcement Learning

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

half-cheetah (slightly different version)



TRPO+GAE (Schulman et al. '16)

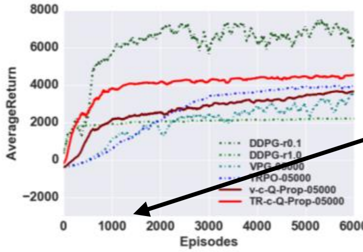


Wang et al. '17

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

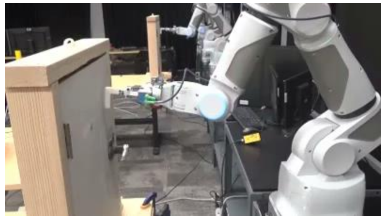
10,000,000 steps
(10,000 episodes)
(~ 1.5 days real time)

half-cheetah

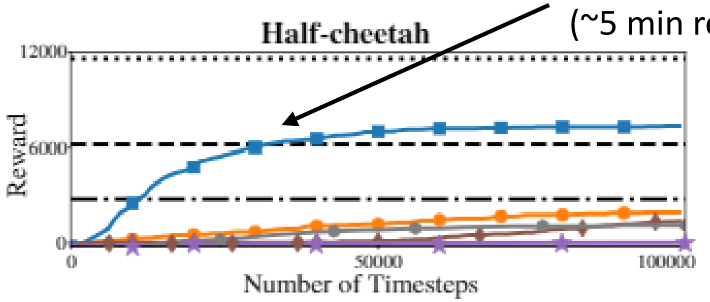


Gu et al. '16

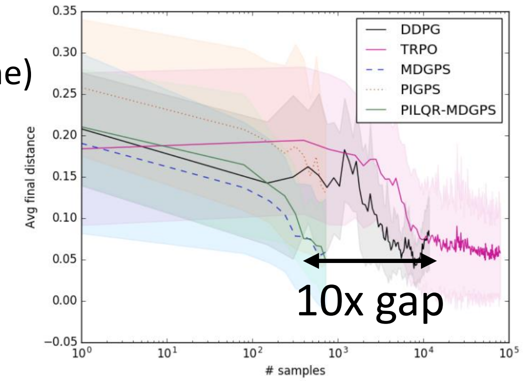
1,000,000 steps
(1,000 episodes)
(~3 hours real time)



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



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

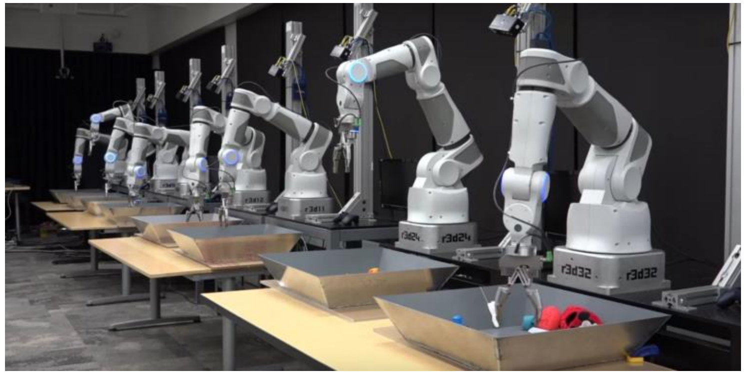


Chebotar et al. '17 (note log scale)

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



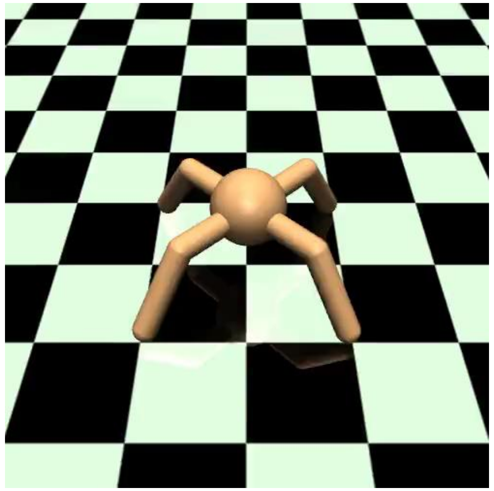
What can we do?

- Better model-based RL algorithms
- Design faster algorithms
  - Addressing Function Approximation Error in Actor-Critic Algorithms (Fujimoto et al. ‘18): simple and effective tricks to accelerate DDPG-style algorithms
  - Soft Actor-Critic (Haarnoja et al. ‘18): very efficient maximum entropy RL algorithm
- Reuse prior knowledge to accelerate reinforcement learning
  - RL2: Fast reinforcement learning via slow reinforcement learning (Duan et al. ‘17)
  - Learning to reinforcement learning (Wang et al. ‘17)
  - Model-agnostic meta-learning (Finn et al. ‘17)

Scaling & Generalization

Scaling up deep RL & generalization

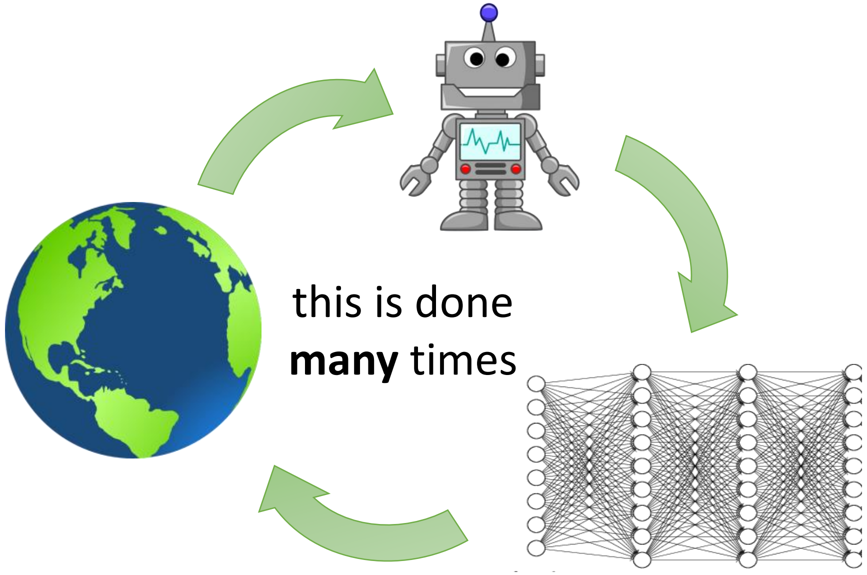
- Large-scale
- Emphasizes diversity
- Evaluated on generalization



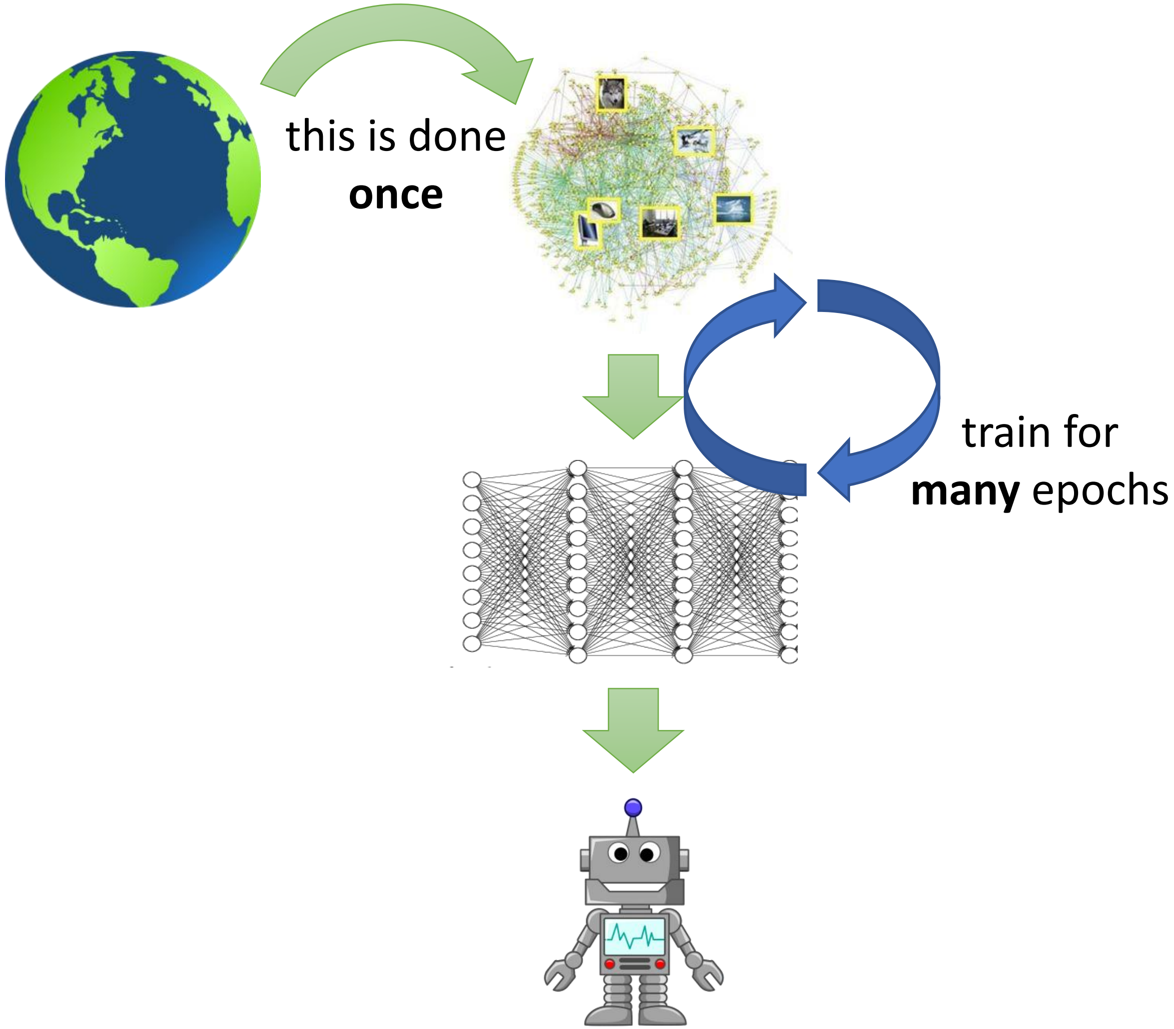
- Small-scale
- Emphasizes mastery
- Evaluated on performance
- Where is the generalization?

RL has a **big** problem

reinforcement learning

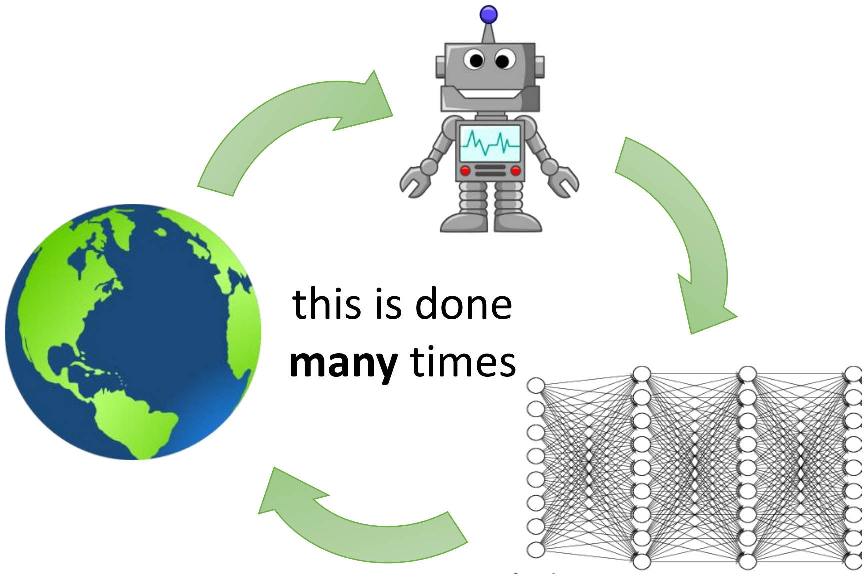


supervised machine learning

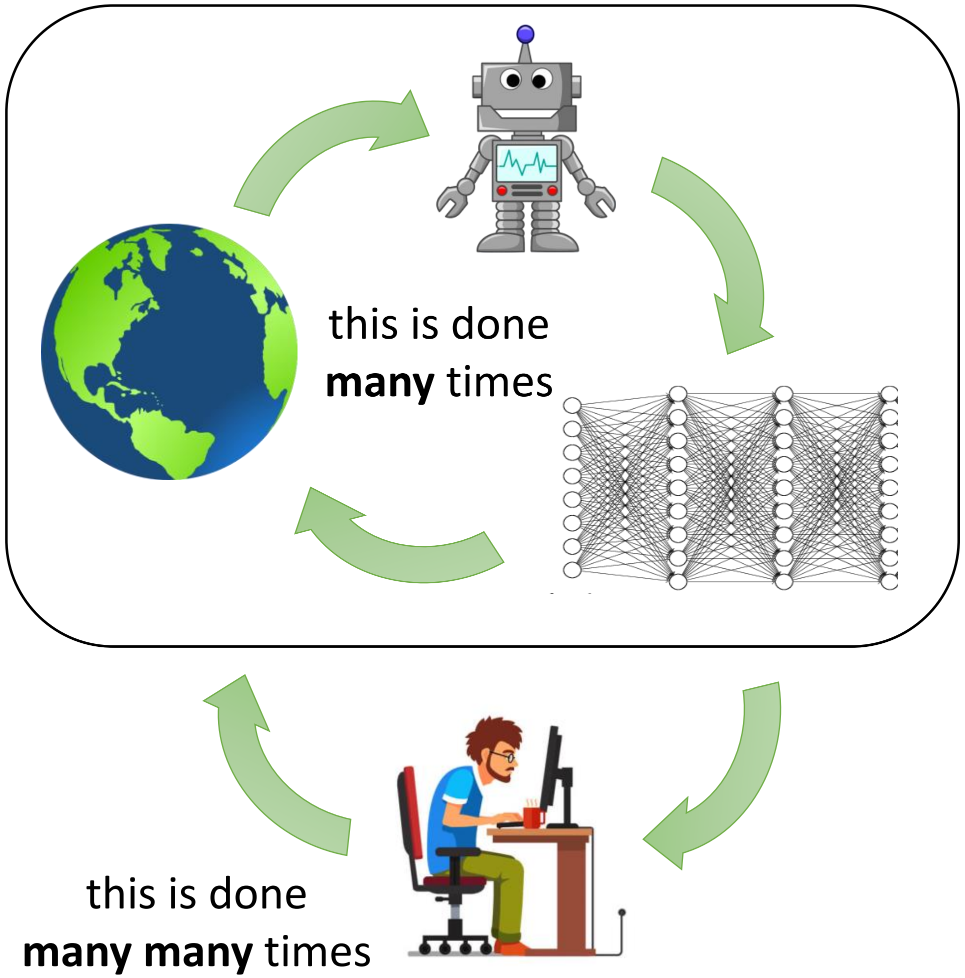


RL has a **big** problem

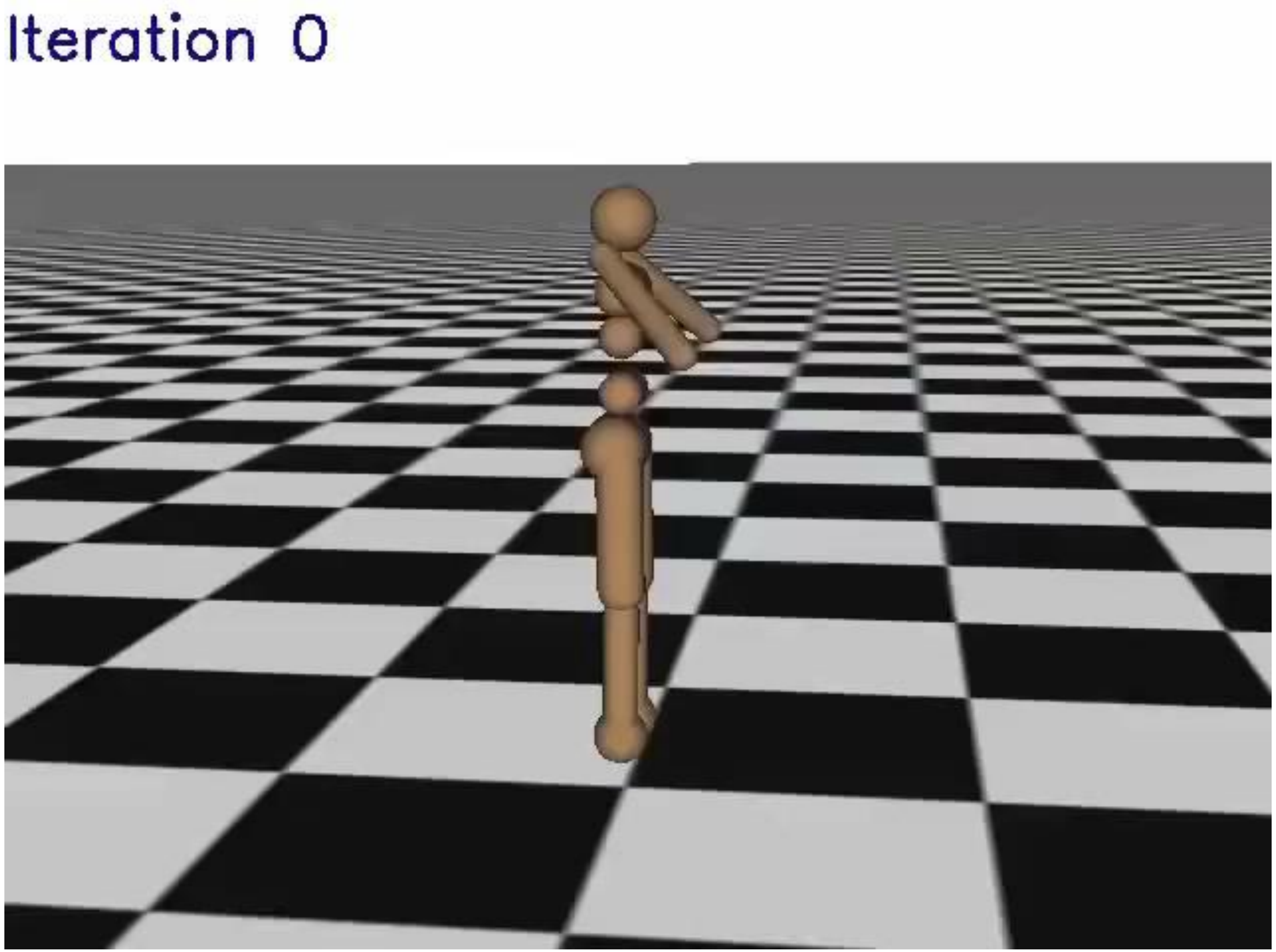
reinforcement learning



actual reinforcement learning



How bad is it?



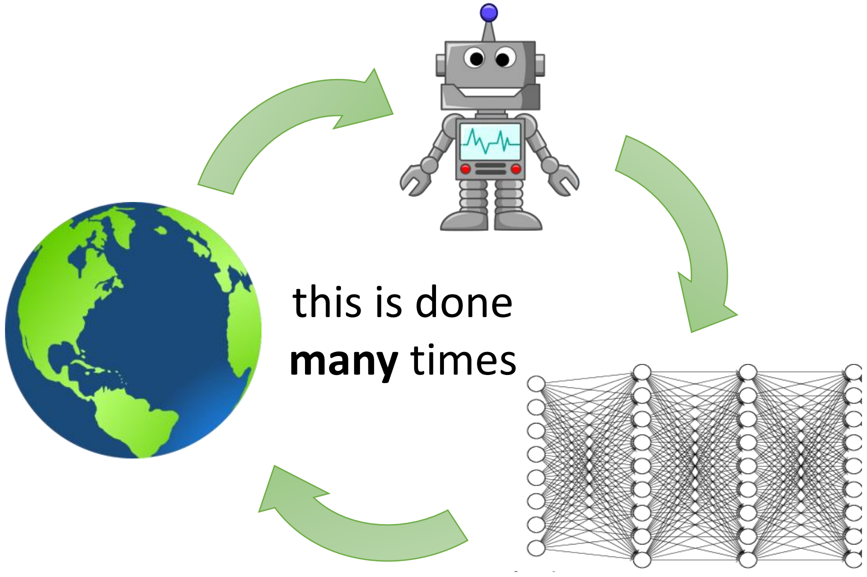
- This is quite cool
- It takes 6 days of real time (if it was real time)
- …to run on an infinite flat plane

The real world is not so simple!

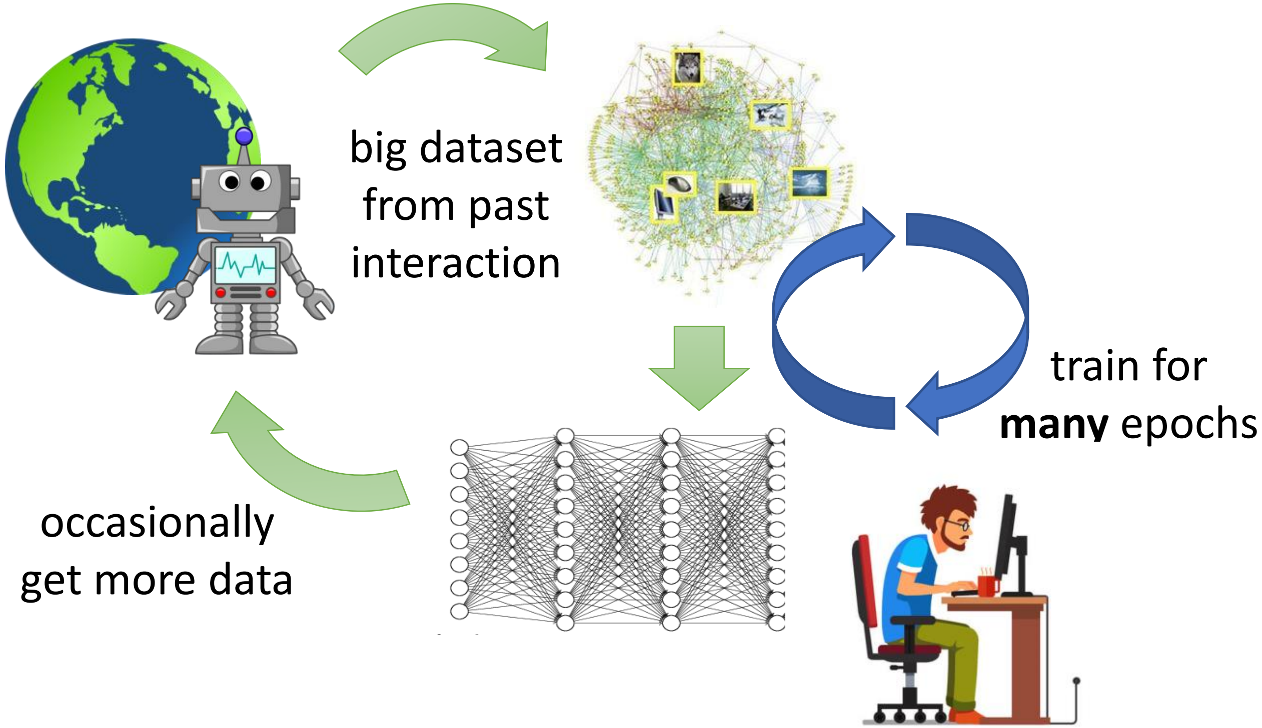
Schulman, Moritz, L., Jordan, Abbeel '16

Off-policy RL?

reinforcement learning



off-policy reinforcement learning



Not just robots!

autonomous driving

language & dialogue
(structured prediction)

finance

What’s the problem?

Challenges with **core algorithms**:

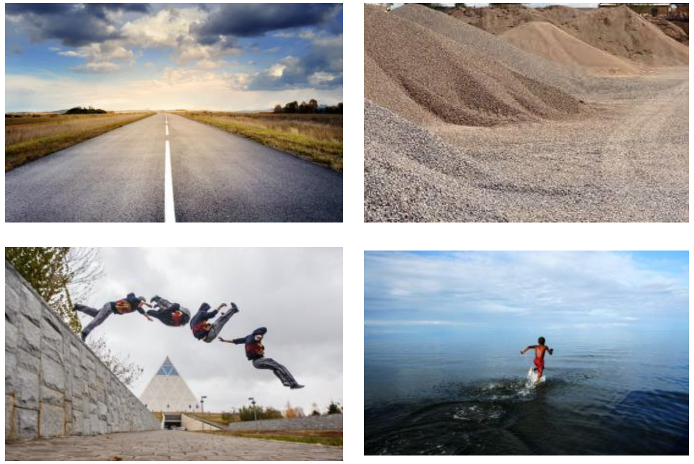
- Stability: does your algorithm converge?
- Efficiency: how long does it take to converge? (how many samples)
- Generalization: after it converges, does it generalize?

Challenges with **assumptions**:

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

Problem Formulation

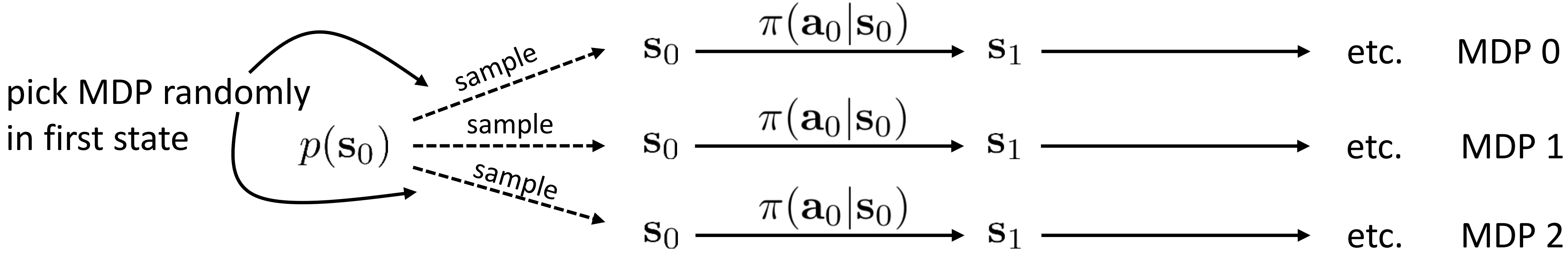
Single task or multi-task?



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



Generalizing from multi-task learning

- Train on multiple tasks, then try to generalize or finetune
  - Policy distillation (Rusu et al. '15)
  - Actor-mimic (Parisotto et al. '15)
  - Model-agnostic meta-learning (Finn et al. '17)
  - many others…
- Unsupervised or weakly supervised learning of diverse behaviors
  - Stochastic neural networks (Florensa et al. '17)
  - Reinforcement learning with deep energy-based policies (Haarnoja et al. '17)
  - many others…

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?

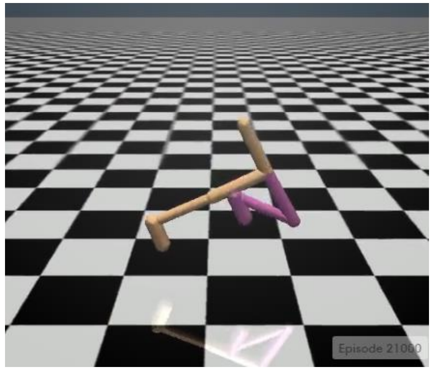
reward

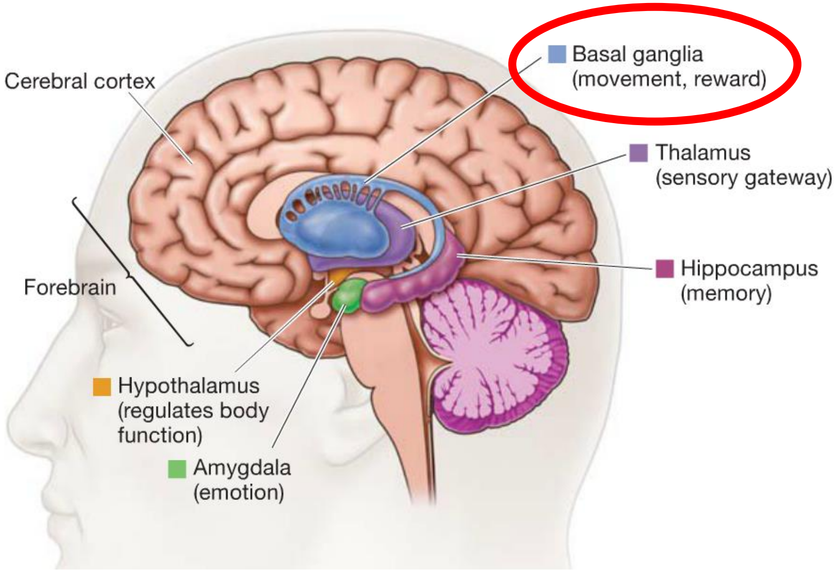
Mnih et al. '15

reinforcement learning agent

what is the reward?

What is the role of the reward function?



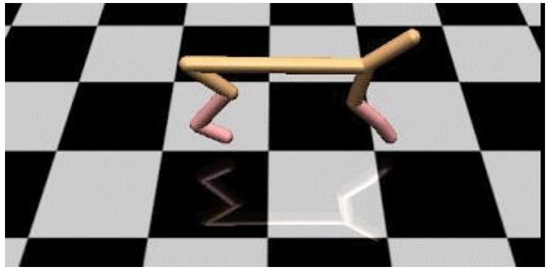


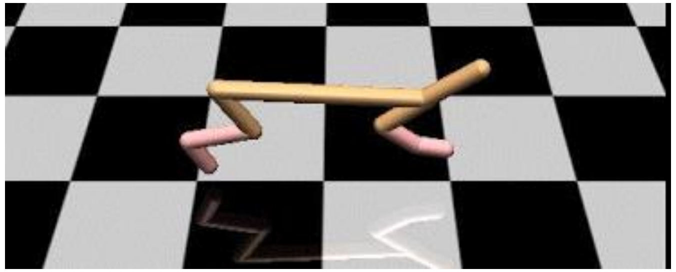
$$r(\mathbf{s},\mathbf{a}) = \begin{cases} 1 \text{ if walker is running} \\ 0 \text{ otherwise} \end{cases}$$

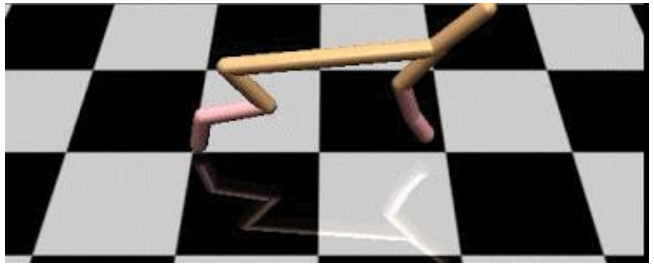
$$\begin{aligned} r(\mathbf{s},\mathbf{a}) =& w_1 v(\mathbf{s}) + \\ & w_2 \delta(|\theta_{\text{torso}}(\mathbf{s})| < \epsilon) + \\ & w_3 \delta(h_{\text{torso}}(\mathbf{s}) \geq h) \end{aligned}$$

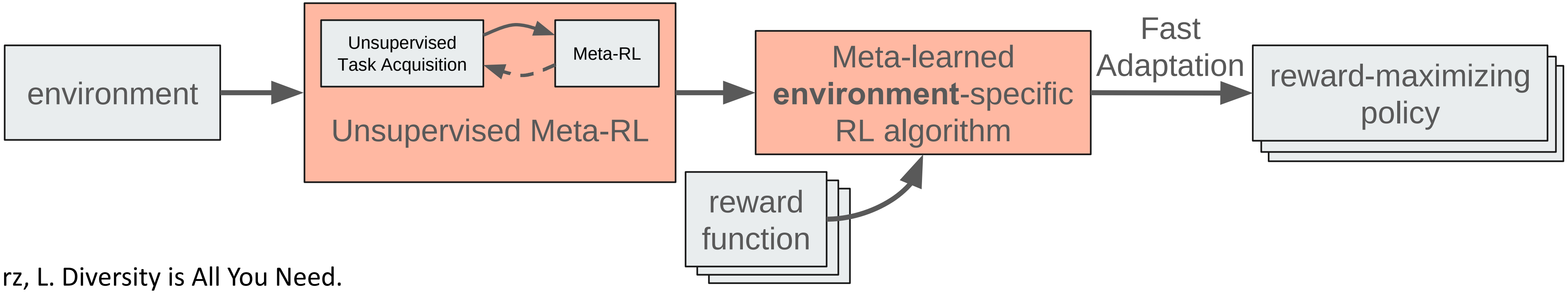
Unsupervised reinforcement learning?

1. Interact with the world, without a reward function
2. Learn *something* about the world (what?)
3. Use what you learned to quickly solve new tasks









Eysenbach, Gupta, Ibarz, L. Diversity is All You Need.

Gupta, Eysenbach, Finn, L. Unsupervised Meta-Learning for Reinforcement Learning.

Other sources of supervision

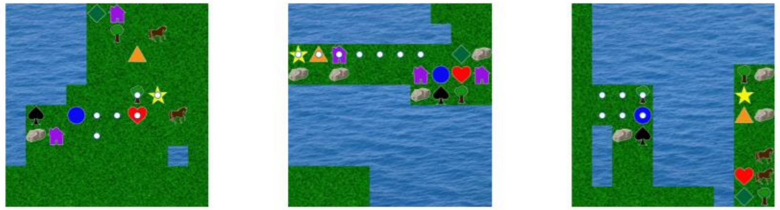
Should supervision tell us **what** to do or **how** to do it?

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

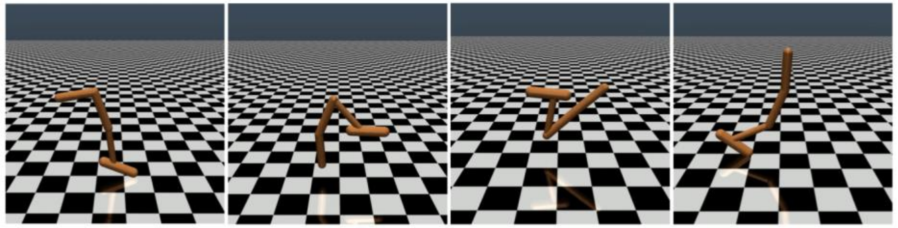
- Language
  - Andreas et al. (2018). Learning with latent language

Human description:
`move to the star`

Inferred description:
`reach the star cell`



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

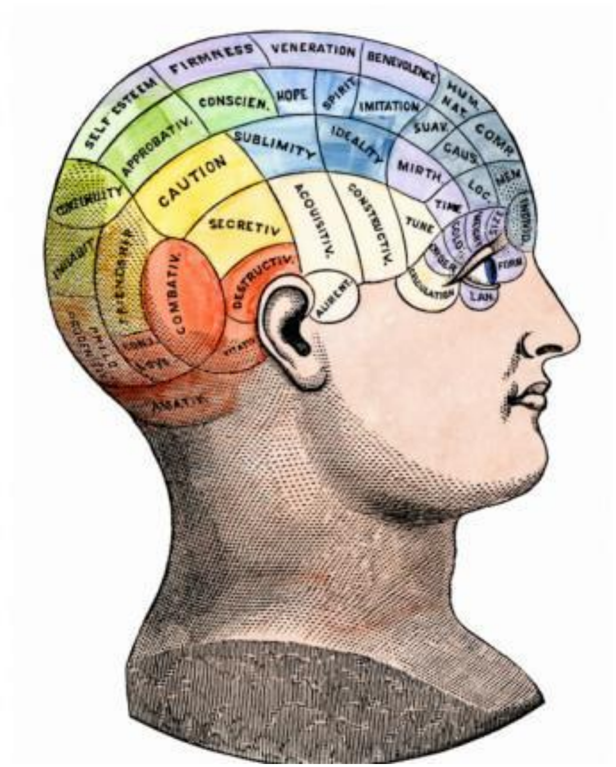


Rethinking the Problem Formulation

- 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

Back to the Bigger Picture

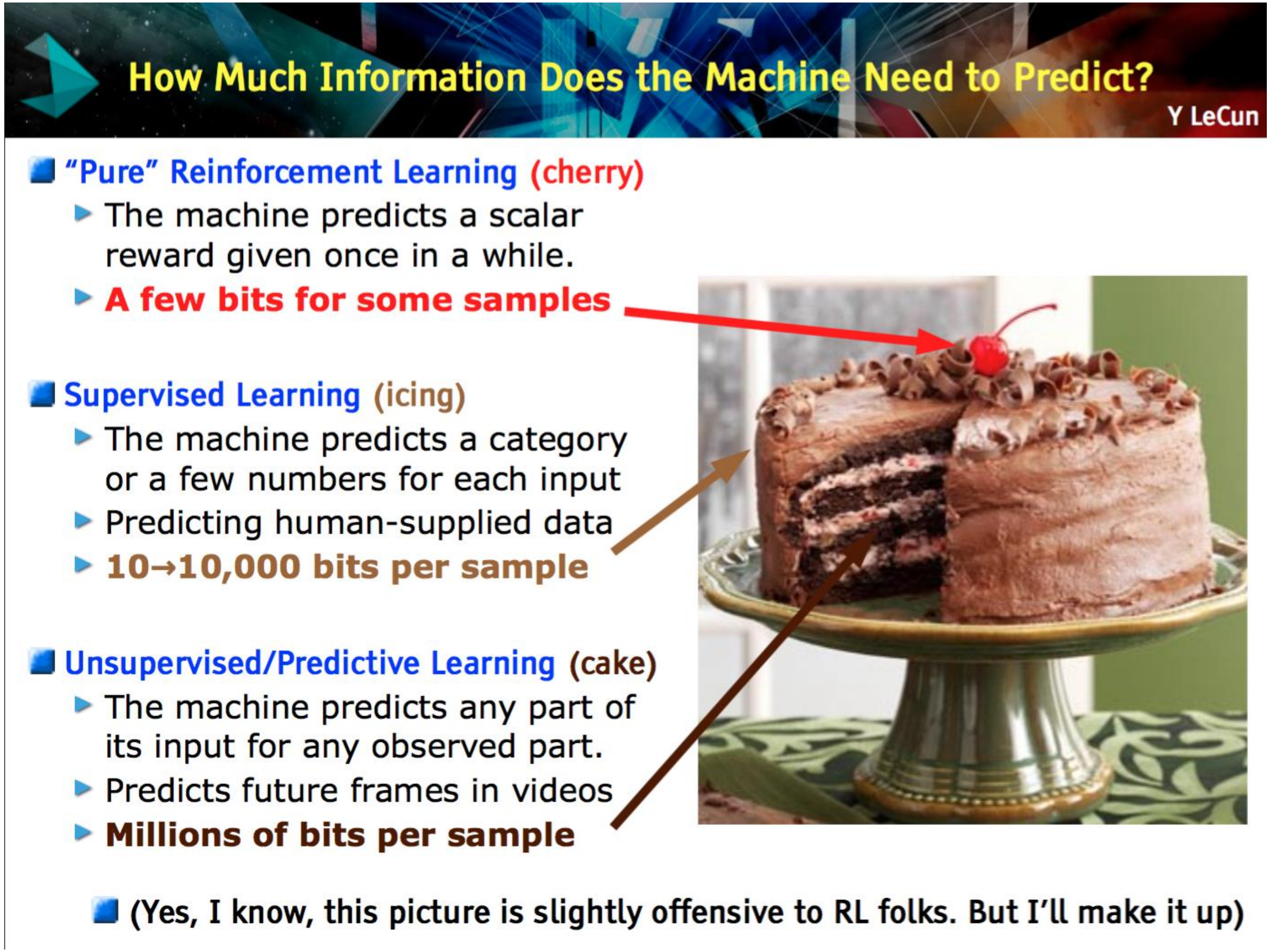
Learning as the basis of intelligence



- 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!
- Pay attention to generative models, prediction, etc., not just RL algorithms
- Carefully understand the relationship between RL and other ML fields