Reinforcement Learning

Today, Direct Policy Optimization

- Policy may be a simpler function to learn
- More naturally deal with stochastic policies

Policies may be simpler

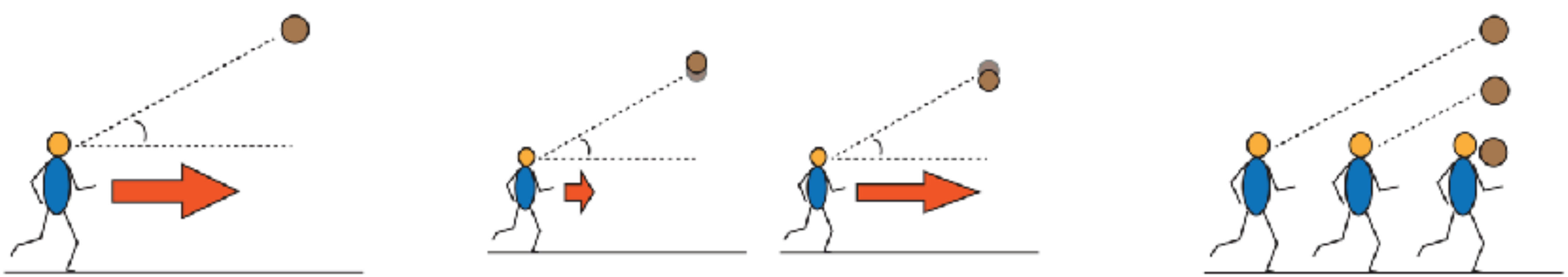
Journal of Experimental Psychology:
Human Perception and Performance
1996, Vol. 22, No. 3, 531–543

Do Fielders Know Where to Go to Catch the Ball or Only How to Get There?

Peter McLeod
Oxford University

Zoltan Dienes
Sussex University

Skilled fielders were filmed as they ran backward or forward to catch balls projected toward them from a bowling machine 45 m away. They ran at a speed that kept the acceleration of the tangent of the angle of elevation of gaze to the ball at 0. This algorithm does not tell fielders where or when the ball will land, but it ensures that they run through the place where the ball drops to catch height at the precise moment that the ball arrives there. The algorithm leads to interception of the ball irrespective of the effect of wind resistance on the trajectory of the ball.



Modulate running speed to maintain angle between ball and ground.

If ball rises in the field of view, slow down.
If ball drops, speed up.
Thus, position of the ball in the field of view is maintained.

Using this heuristic, human catcher arrives at landing point exactly when the ball lands.

Do fielders know where to go to catch the ball or only how to get there? P. McLeod, Z. Dienes   Image Credit: Sergey Levine

Stochastic policies

Rock, paper, scissors

Image Credit: David Silver

Directly Optimize Policies?

$\pi_\theta(s, a)$ policy (parameterized by θ)

$J(\theta) =$ Average return when acting as per $\pi_\theta(s, a)$

$$\theta_* = argmax_\theta \, J(\theta)$$

Image Credit: David Silver

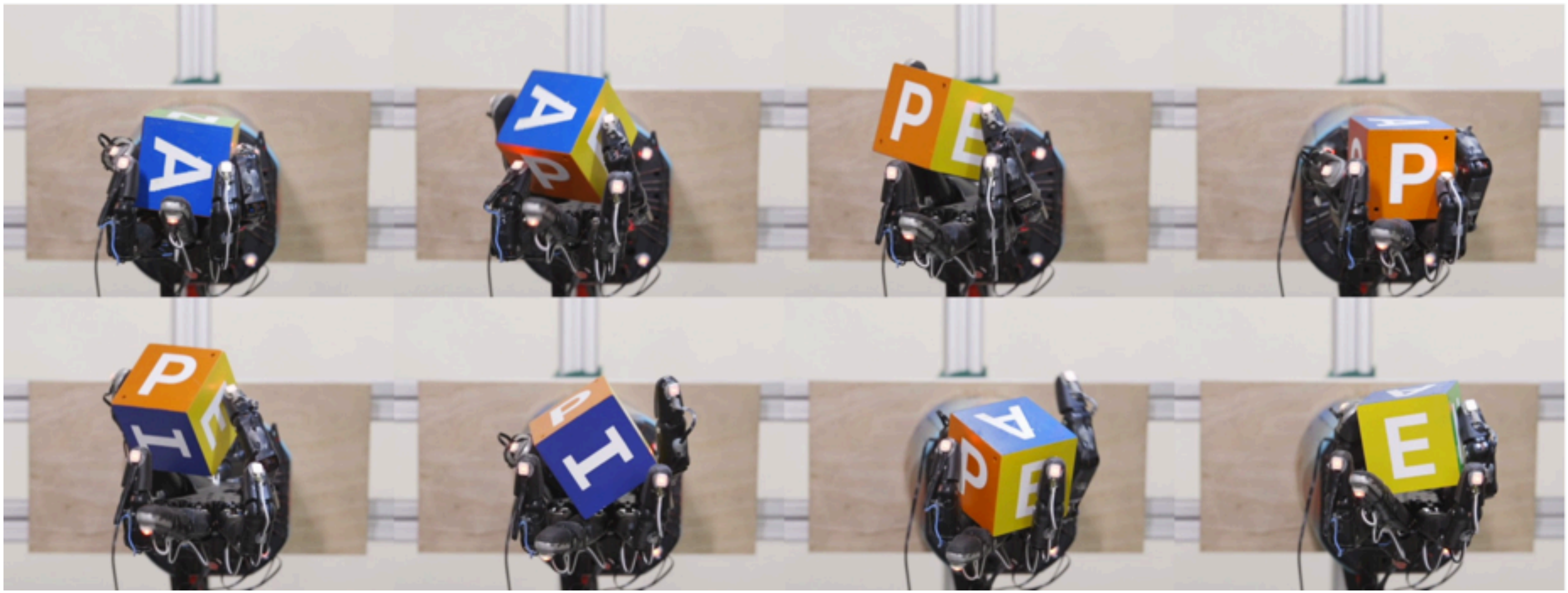
Summary

The policy gradient has many equivalent forms

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ v_t\right] && \text{REINFORCE} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ Q^w(s,a)\right] && \text{Q Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ A^w(s,a)\right] && \text{Advantage Actor-Critic} \\
&= \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a)\ \delta\right] && \text{TD Actor-Critic}
\end{aligned}$$

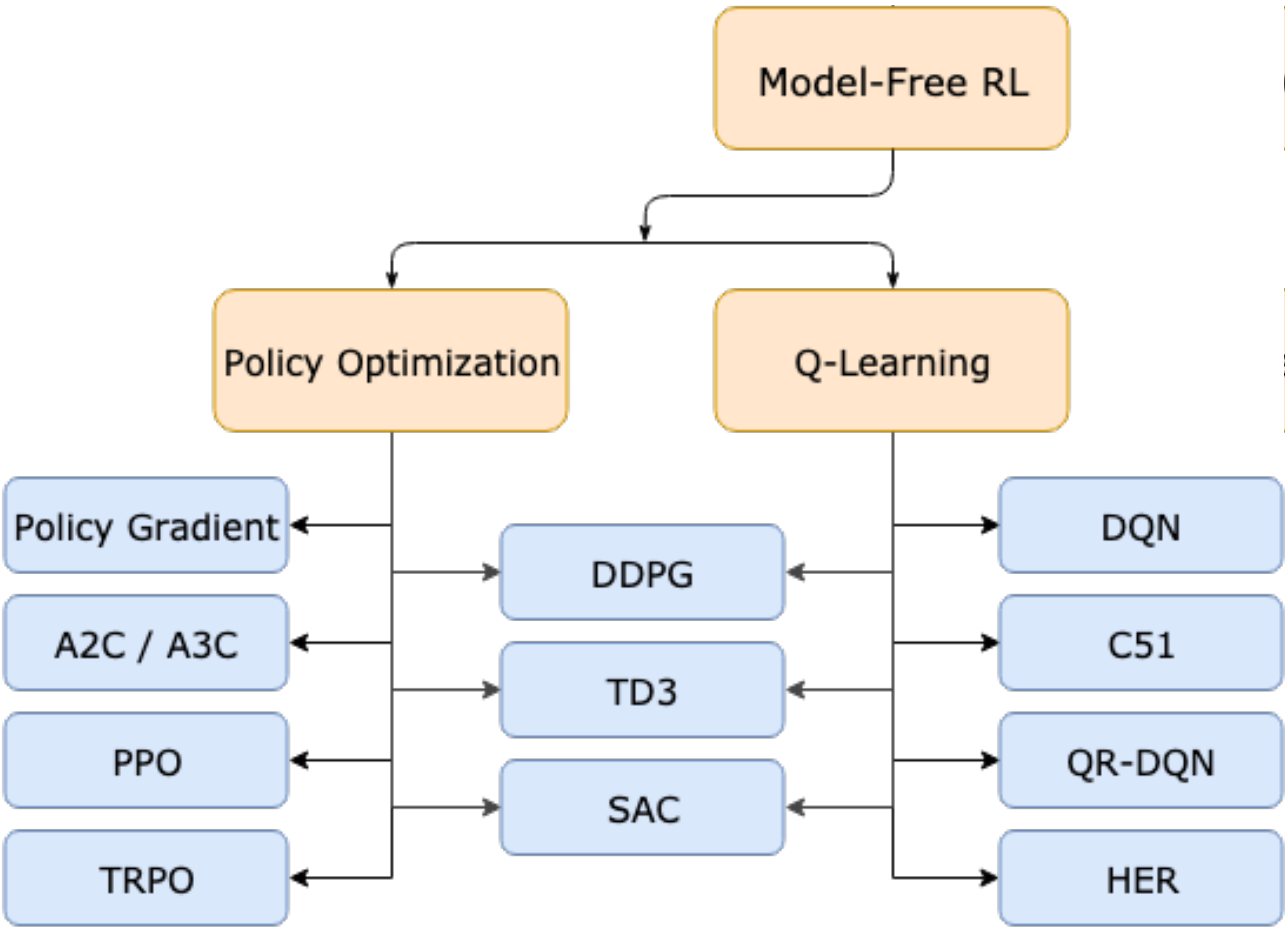
Image Credit: David Silver

Many successes in simulation



Learning Dexterous In-Hand Manipulation, https://arxiv.org/pdf/1808.00177.pdf

Summary

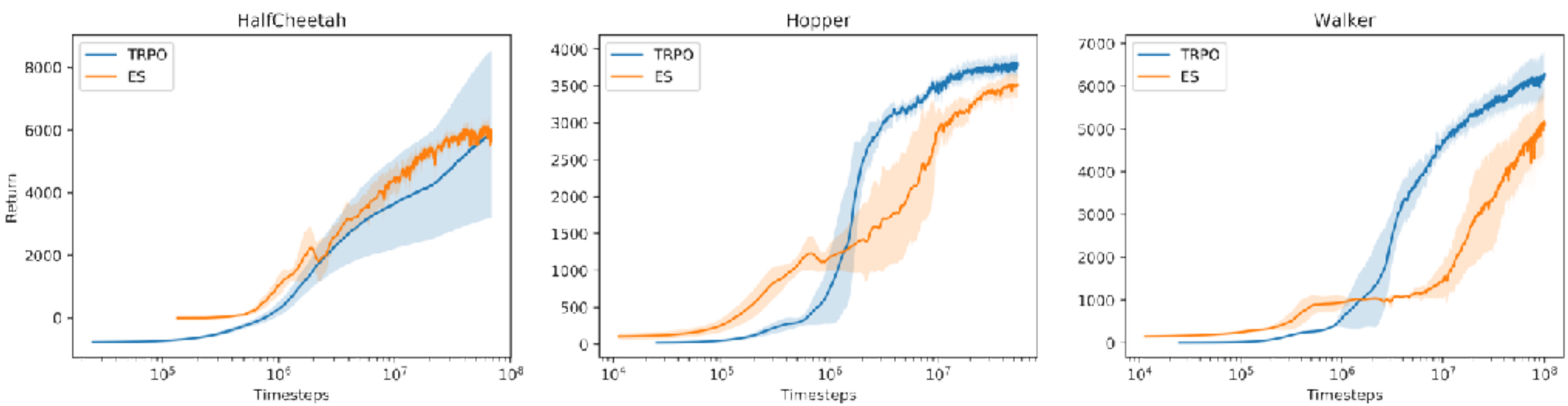


Source: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#id20

Policy gradient really a gradient?

Evolution Strategies as a Scalable Alternative to Reinforcement Learning

Tim Salimans **Jonathan Ho** **Xi Chen** **Szymon Sidor** **Ilya Sutskever**
OpenAI

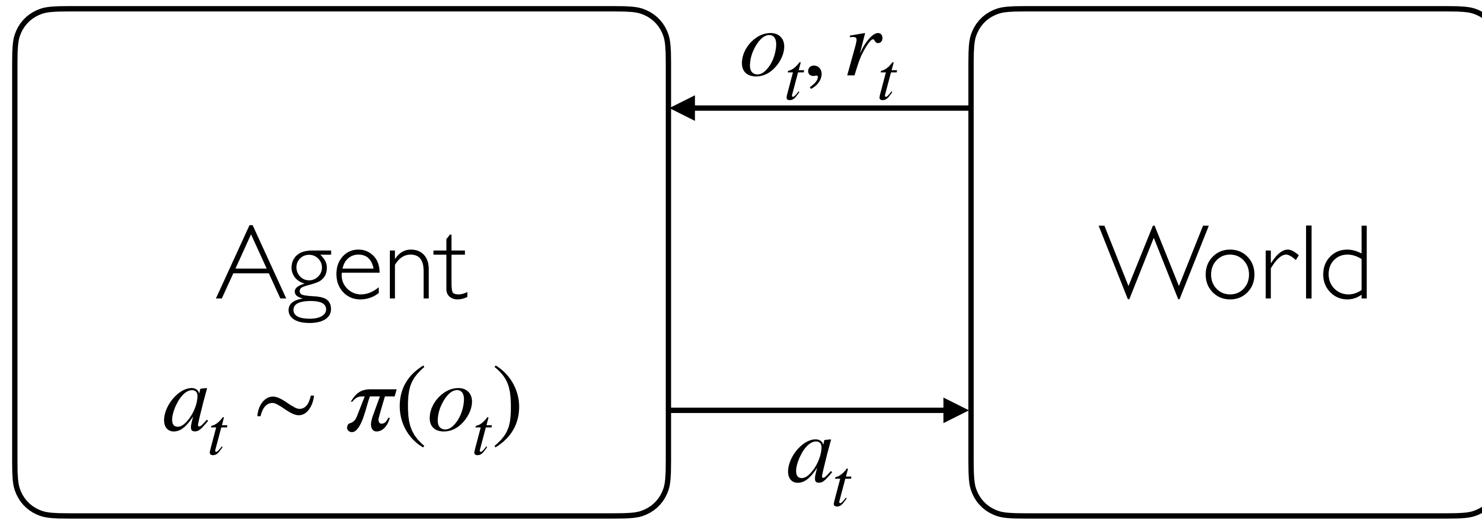


See also: http://www.argmin.net/2018/02/20/reinforce/

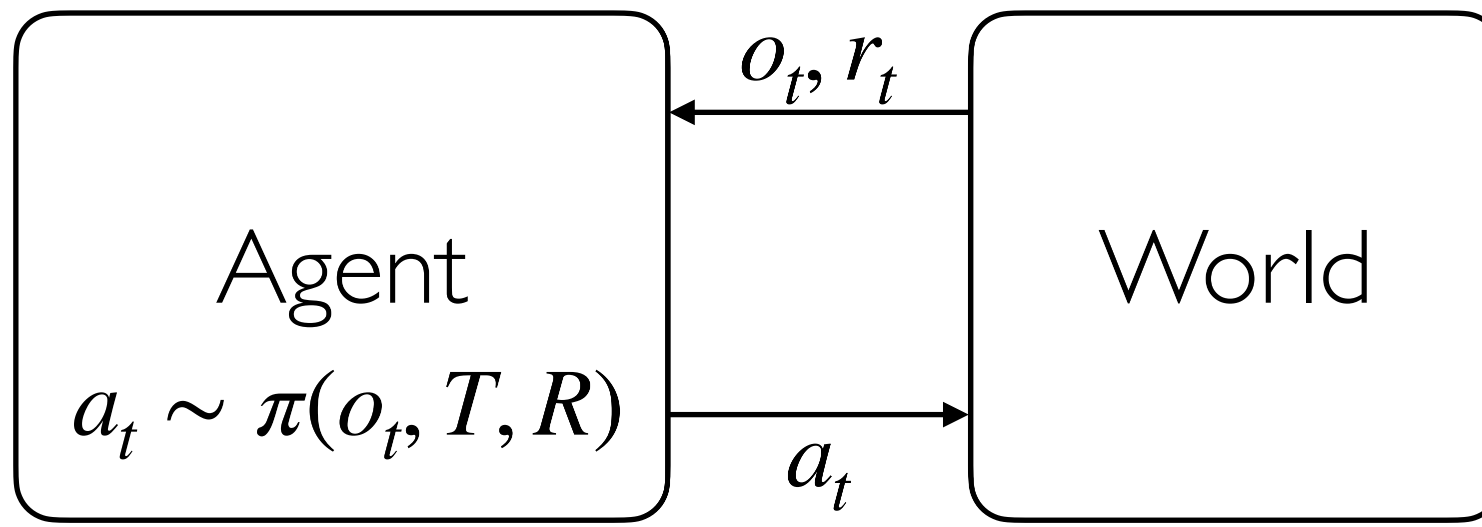
Solving MDPs

Policy: $a_t \sim \pi(o_t)$

Most General Case



More Specific Case



Fully Observed System $o_t = s_t$

Known Transition Function $s_{t+1} \sim T(s_t, a_t)$

Known Reward Function $R(s_{t+1}, s_t, a_t)$

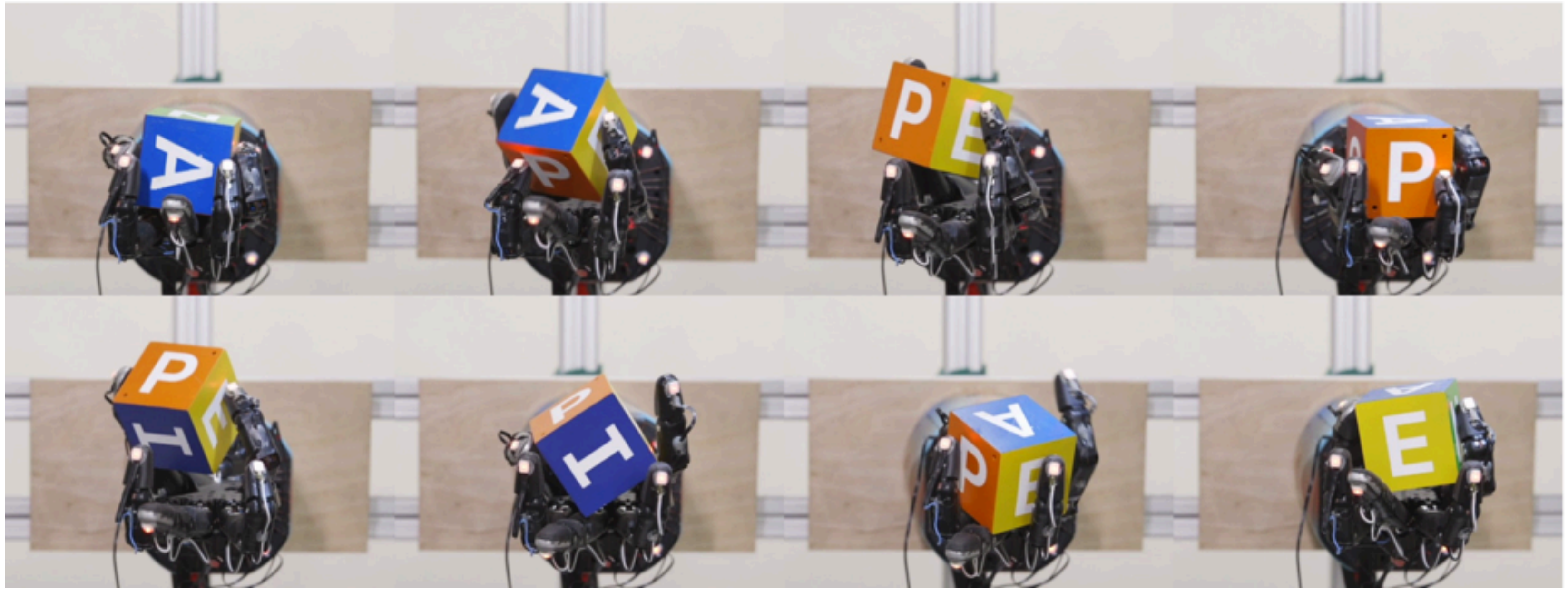
So, are we done?

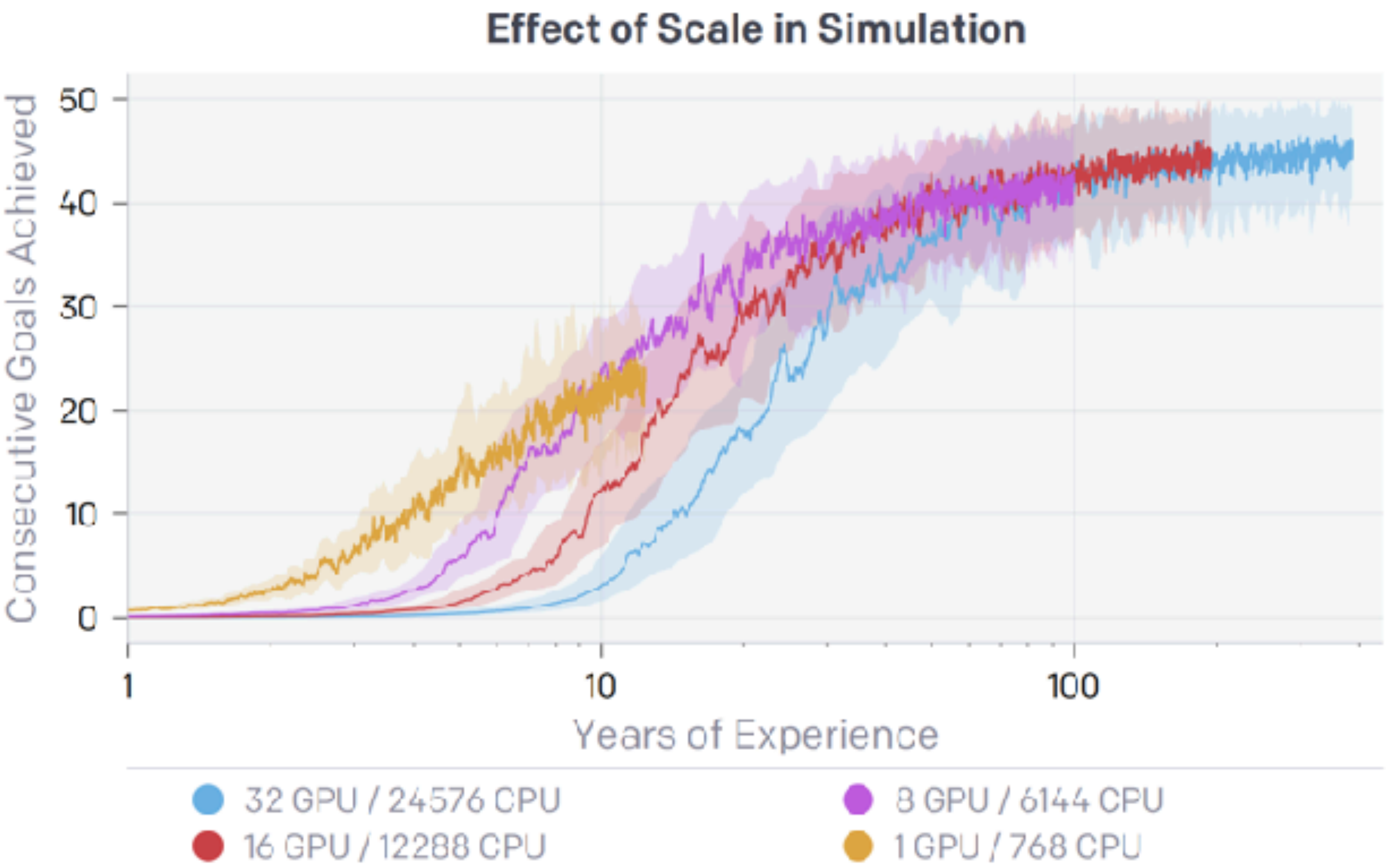
- Exploration is challenging
- Credit assignment problem

Poor sample complexity

Yann LeCun's Cake

Sample Complexity





Learning Dexterous In-Hand Manipulation, https://arxiv.org/pdf/1808.00177.pdf

Solving a RL Problem

Better reward signals

Sim2Real

Better optimization

Convert into a supervised training problem

Solve a related but supervision rich problem

Build models and plan with them

Model-free RL with sparse rewards

Known reward, known model. Model-based RL