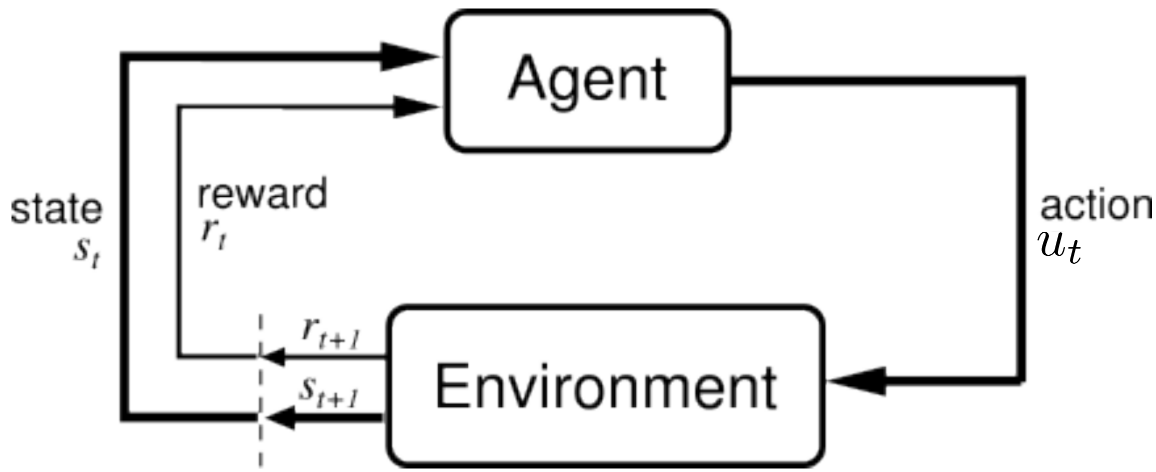
Representations in Deep Reinforcement Learning

Pieter Abbeel
Open AI / Berkeley AI Research Lab

Many slides made in collaboration with John Schulman and Sergey Levine

Reinforcement Learning



[Figure source: Sutton & Barto, 1998]

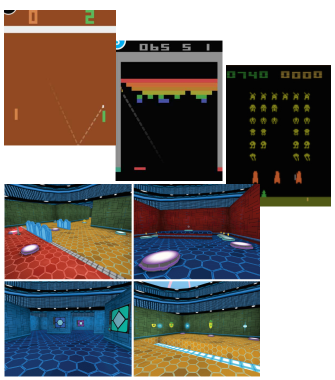
Some Reinforcement Learning Success Stories

Kohl and Stone, 2004

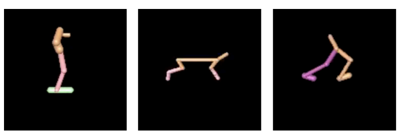
Ng et al, 2004

Tedrake et al, 2005

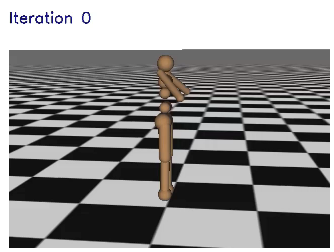
Kober and Peters, 2009



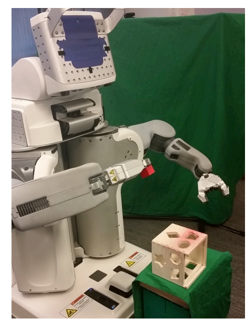
Mnih et al, 2015
(A3C)



Silver et al, 2014
(DPG)
Lillicrap et al, 2015
(DDPG)



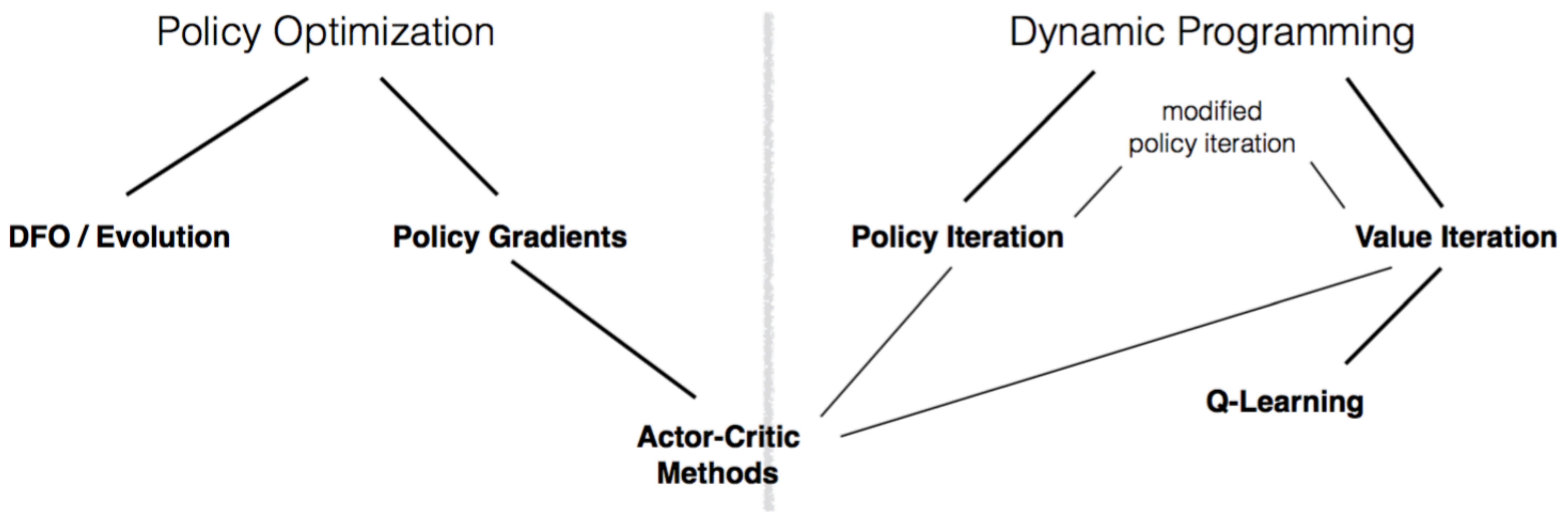
Schulman et al,
2016 (TRPO + GAE)



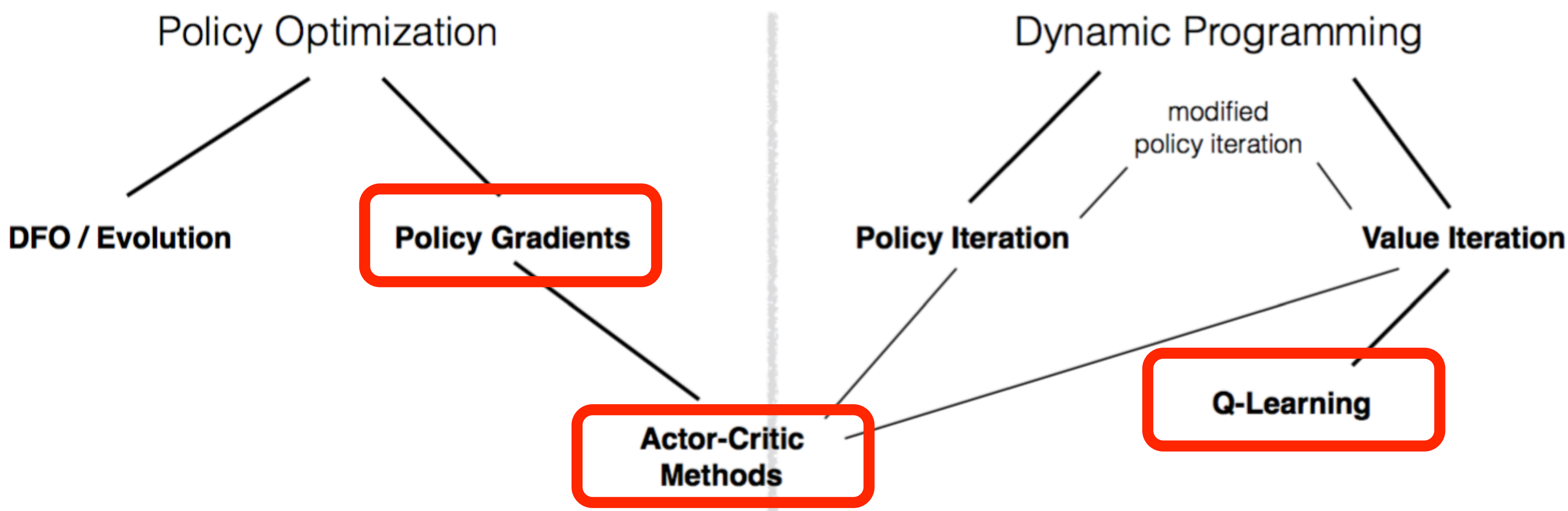
Levine*, Finn*, et
al, 2016
(GPS)

Silver*, Huang*, et
al, 2016
(AlphaGo)

RL Algorithms Landscape



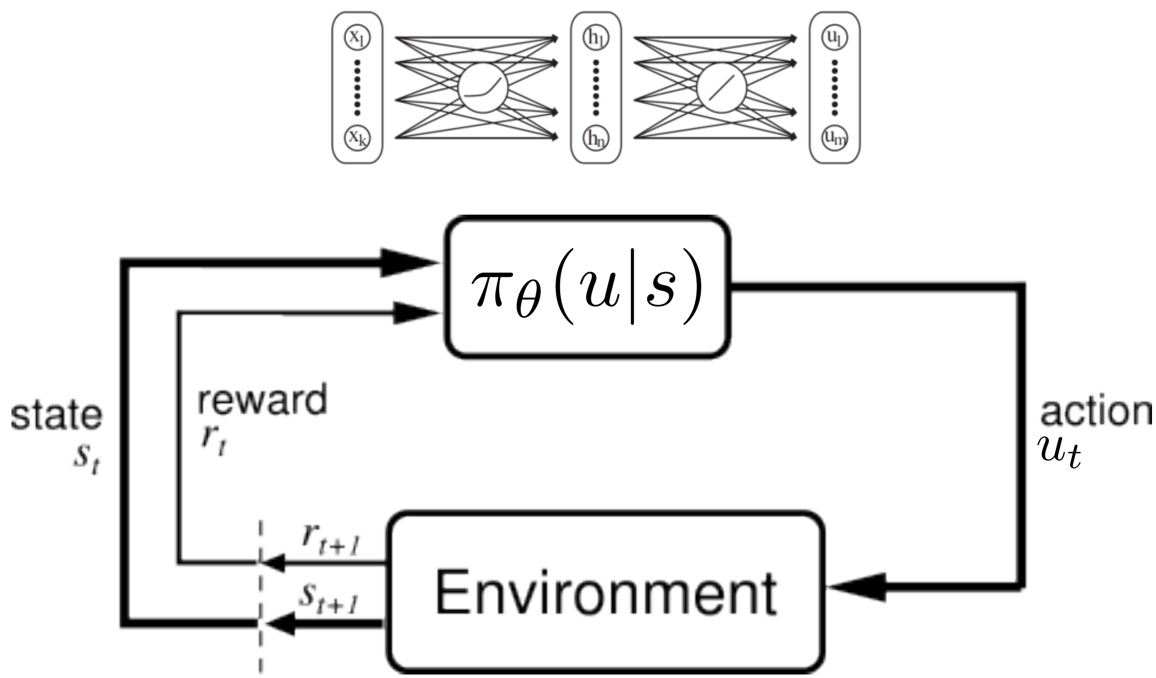
RL Algorithms Landscape



Talk Outline

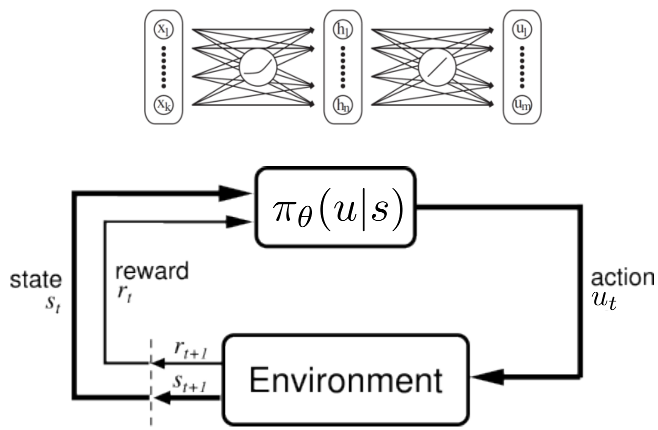
- Classical RL
  - Algorithms
    - Policy Gradients
    - Actor-Critic
    - Q-learning
  - Representation
- Representation in exploration
- Different Approaches / Architectures
  - Value Iteration Networks
  - Predictron
  - Modular Networks
  - Option-Critic
  - Feudal Networks
- Meta learning
  - MAML
  - RL2

Policy Optimization



[Figure source: Sutton & Barto, 1998]

Policy Optimization



- Consider control policy parameterized by parameter vector θ

$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

- Often stochastic policy class (smooths out the problem):

  $\pi_\theta(u|s)$: probability of action u in state s

[Figure source: Sutton & Barto, 1998]

Likelihood Ratio Policy Gradient

We let τ denote a state-action sequence $s_0, u_0, \ldots, s_H, u_H$. We overload notation: $R(\tau) = \sum_{t=0}^{H} R(s_t, u_t)$.

$$U(\theta) = \mathrm{E}[\sum_{t=0}^{H} R(s_t, u_t); \pi_\theta] = \sum_{\tau} P(\tau; \theta) R(\tau)$$

In our new notation, our goal is to find θ:

$$\max_{\theta} U(\theta) = \max_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau)$$

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_\tau P(\tau;\theta)R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\nabla_\theta U(\theta) = \nabla_\theta \sum_\tau P(\tau;\theta)R(\tau)$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]
[Rubinstein, 1969]
[Glynn, 1986]
[Reinforce, Williams 1992]
[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau;\theta)R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau} P(\tau;\theta)R(\tau) \\
&= \sum_{\tau} \nabla_\theta P(\tau;\theta)R(\tau)
\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]
[Rubinstein, 1969]
[Glynn, 1986]
[Reinforce, Williams 1992]
[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau;\theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau} P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} \nabla_\theta P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta) R(\tau)
\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]
[Rubinstein, 1969]
[Glynn, 1986]
[Reinforce, Williams 1992]
[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau;\theta)R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau} P(\tau;\theta)R(\tau) \\
&= \sum_{\tau} \nabla_\theta P(\tau;\theta)R(\tau) \\
&= \sum_{\tau} \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta)R(\tau) \\
&= \sum_{\tau} P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} R(\tau)
\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]
[Rubinstein, 1969]
[Glynn, 1986]
[Reinforce, Williams 1992]
[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau;\theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_{\tau} P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} \nabla_\theta P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta) R(\tau) \\
&= \sum_{\tau} P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} R(\tau) \\
&= \sum_{\tau} P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) R(\tau)
\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]
[Rubinstein, 1969]
[Glynn, 1986]
[Reinforce, Williams 1992]
[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_\tau P(\tau;\theta)R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_\tau P(\tau;\theta)R(\tau) \\
&= \sum_\tau \nabla_\theta P(\tau;\theta)R(\tau) \\
&= \sum_\tau \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta)R(\tau) \\
&= \sum_\tau P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} R(\tau) \\
&= \sum_\tau P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) R(\tau)
\end{aligned}$$

Approximate with the empirical estimate for m sample paths under policy π_θ:

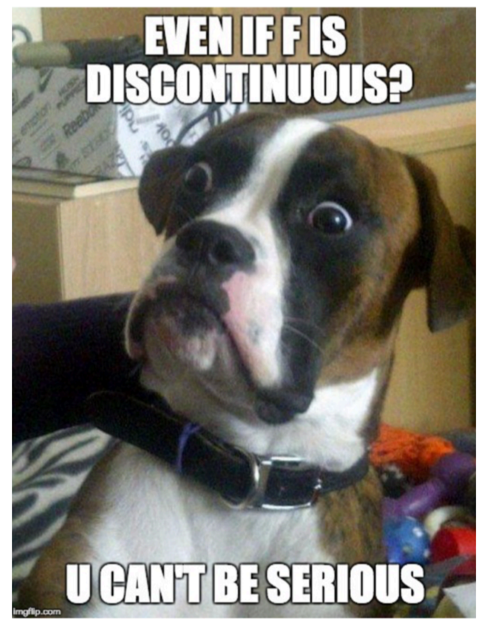
$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)};\theta) R(\tau^{(i)})$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]
[Rubinstein, 1969]
[Glynn, 1986]
[Reinforce, Williams 1992]
[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Gradient: Validity

$$\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

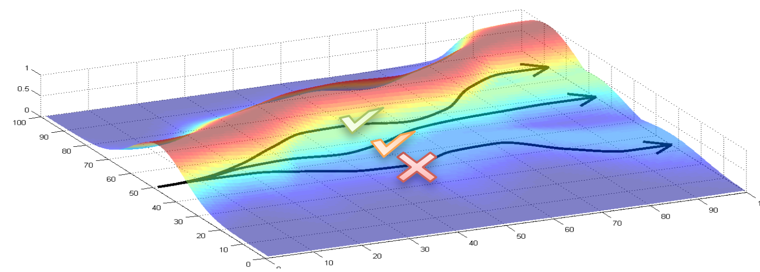
- Valid even if R is discontinuous, and unknown, or sample space (of paths) is a discrete set



Likelihood Ratio Gradient: Intuition

$$\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta} \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

- Gradient tries to:
  - Increase probability of paths with positive R
  - Decrease probability of paths with negative R



! Likelihood ratio changes probabilities of experienced paths, does not try to change the paths

Let's Decompose Path into States and Actions

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log \left[\prod_{t=0}^{H} \underbrace{P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta(u_t^{(i)}|s_t^{(i)})}_{\text{policy}} \right]$$

Let's Decompose Path into States and Actions

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log \left[\prod_{t=0}^{H} \underbrace{P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta(u_t^{(i)} | s_t^{(i)})}_{\text{policy}} \right]$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)} | s_t^{(i)}) \right]$$

Let’s Decompose Path into States and Actions

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log \left[\prod_{t=0}^{H} \underbrace{P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta(u_t^{(i)} | s_t^{(i)})}_{\text{policy}} \right]$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)} | s_t^{(i)}) \right]$$

$$= \nabla_\theta \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)} | s_t^{(i)})$$

Let's Decompose Path into States and Actions

$$\nabla_\theta \log P(\tau^{(i)};\theta) = \nabla_\theta \log \left[\prod_{t=0}^{H} \underbrace{P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_\theta(u_t^{(i)}|s_t^{(i)})}_{\text{policy}} \right]$$

$$= \nabla_\theta \left[\sum_{t=0}^{H} \log P(s_{t+1}^{(i)}|s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)}) \right]$$

$$= \nabla_\theta \sum_{t=0}^{H} \log \pi_\theta(u_t^{(i)}|s_t^{(i)})$$

$$= \sum_{t=0}^{H} \underbrace{\nabla_\theta \log \pi_\theta(u_t^{(i)}|s_t^{(i)})}_{\text{no dynamics model required!!}}$$

Likelihood Ratio Gradient Estimate

$$\hat{g} = \frac{1}{m} \sum_{k=1}^{m} \nabla_\theta \log P(\tau^{(k)}; \theta) R(\tau^{(k)})$$

Likelihood Ratio Gradient Estimate

$$\hat{g} = \frac{1}{m}\sum_{k=1}^{m} \nabla_\theta \log P(\tau^{(k)};\theta) R(\tau^{(k)})$$

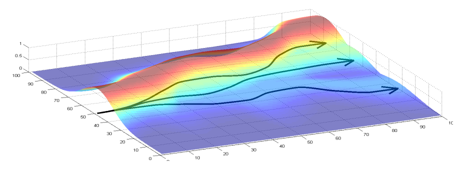
$$= \frac{1}{m}\sum_{k=1}^{m}\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(k)}|s_t^{(k)}) R(\tau^{(k)})$$

Likelihood Ratio Gradient Estimate

$$
\begin{aligned}
\hat{g} &= \frac{1}{m} \sum_{k=1}^{m} \nabla_\theta \log P(\tau^{(k)}; \theta) R(\tau^{(k)}) \\
&= \frac{1}{m} \sum_{k=1}^{m} \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(k)} | s_t^{(k)}) R(\tau^{(k)}) \\
&= \frac{1}{m} \sum_{k=1}^{m} \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(k)} | s_t^{(k)}) \sum_{t'=t}^{H-1} r_{t'}^{(k)}
\end{aligned}
$$

Likelihood Ratio Gradient Estimate

$$\begin{aligned}
\hat{g} &= \frac{1}{m}\sum_{k=1}^{m} \nabla_\theta \log P(\tau^{(k)};\theta) R(\tau^{(k)}) \\
&= \frac{1}{m}\sum_{k=1}^{m}\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(k)}|s_t^{(k)}) R(\tau^{(k)}) \\
&= \frac{1}{m}\sum_{k=1}^{m}\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(k)}|s_t^{(k)}) \sum_{t'=t}^{H-1} r_{t'}^{(k)} \\
&= \frac{1}{m}\sum_{k=1}^{m}\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(k)}|s_t^{(k)}) \left(\sum_{t'=t}^{H-1} r_{t'}^{(k)} - b(s_{t'}^{(k)})\right)
\end{aligned}$$



$$b(s_{t'}^{(k)}) = \frac{1}{m}\sum_{k=1}^{m}\sum_{t'=t}^{H-1} r_{t'}^{(k)}$$

Likelihood Ratio Policy Gradient

- Init π_{θ_0}
- Collect data {s, u, s', r}
- $\theta_{i+1} \leftarrow \theta_i + \alpha \;\; \frac{1}{m} \sum_{k=1}^{m} \sum_{t=0}^{H-1} \nabla_\theta \log \pi_{\theta_i}(u_t^{(k)} | s_t^{(k)}) \left(\sum_{t'=t}^{H-1} r_{t'}^{(k)} - b(s_{t'}^{(k)}) \right)$

$$b(s_{t'}^{(k)}) = \frac{1}{m} \sum_{k=1}^{m} \sum_{t'=t}^{H-1} r_{t'}^{(k)}$$

→ Increase logprob of action proportionally to how much its returns are better than the expected return under the current policy

- Can we get a better b ? Yes! V^π [-> "actor-critic"]

Estimation of V^π

- Bellman Equation for V^π

$$V^\pi(s) = \sum_u \pi(u|s) \sum_{s'} P(s'|s,u)[R(s,u,s') + \gamma V^\pi(s')]$$

- Fitted V iteration:
  - Init $V^\pi_{\phi_0}$
  - Collect data {s, u, s', r}
  - $\phi_{i+1} \leftarrow \min_\phi \sum_{(s,u,s',r)} \|r + V^\pi_{\phi_i}(s') - V_\phi(s)\|_2^2 + \lambda\|\phi - \phi_i\|_2^2$

Actor-Critic

- Policy Gradient + Fitted V iteration:
  - Init π_{θ_0} $V^{\pi}_{\phi_0}$
  - Collect data {s, u, s', r}
  - $$\phi_{i+1} \leftarrow \min_{\phi} \sum_{(s,u,s',r)} \|r + V^{\pi}_{\phi_i}(s') - V_{\phi}(s)\|_2^2 + \lambda\|\phi - \phi_i\|_2^2$$
    $$\theta_{i+1} \leftarrow \theta_i + \alpha \ \frac{1}{m}\sum_{k=1}^{m}\sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta_i}(u_t^{(k)}|s_t^{(k)}) \left(\sum_{t'=t}^{H-1} r_{t'}^{(k)} - V^{\pi}_{\phi_i}(s_{t'}^{(k)})\right)$$

- Generalized Advantage Estimation (Schulman et al, 2016) also uses learned value function to better estimate first term $\sum_{t'=t}^{H-1} r_{t'}^{(k)}$

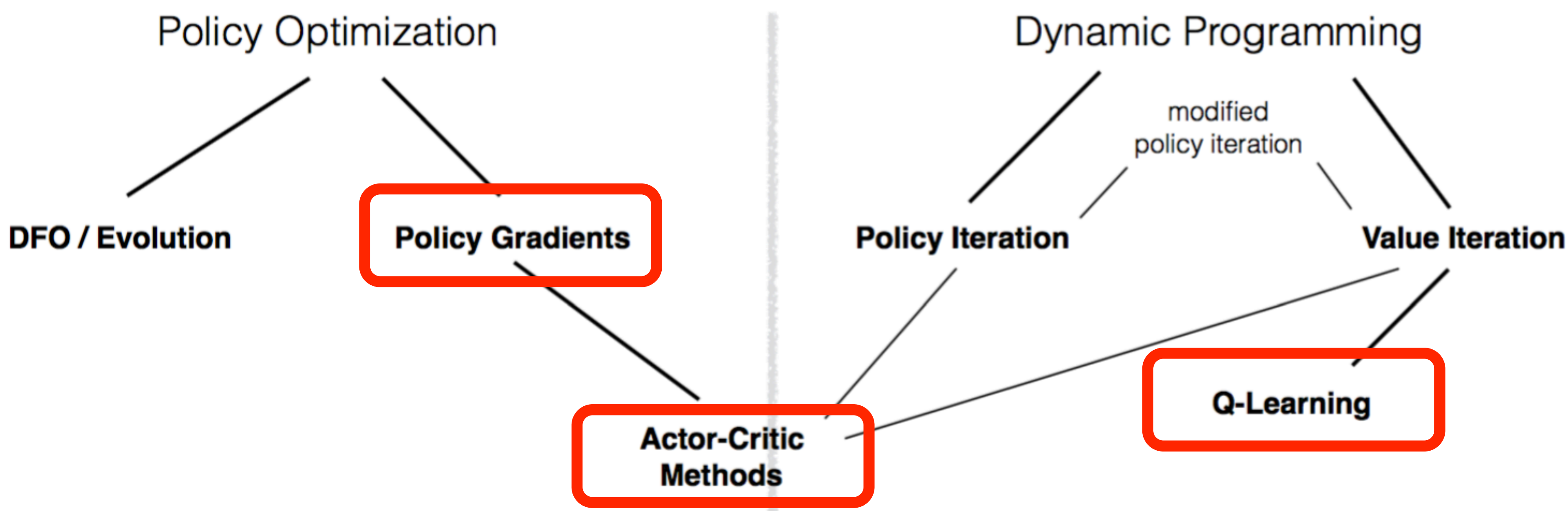
Q-learning

- Bellman equation for Q*:

$$Q^*(s,u) = \sum_{s'} P(s'|s,u)[R(s,u,s') + \max_{u'} Q^*(s',u')]$$

- Fitted Q iteration:
  - Init Q_{ϕ_0}
  - Collect data {s, u, s', r}
  - $\phi_{i+1} \leftarrow \min_{\phi} \sum_{(s,u,s',r)} \|r + \max_{u'} Q_{\phi_i}(s',u') - Q_{\phi}(s,u)\|_2^2 + \lambda\|\phi - \phi_i\|_2^2$

RL Algorithms Landscape

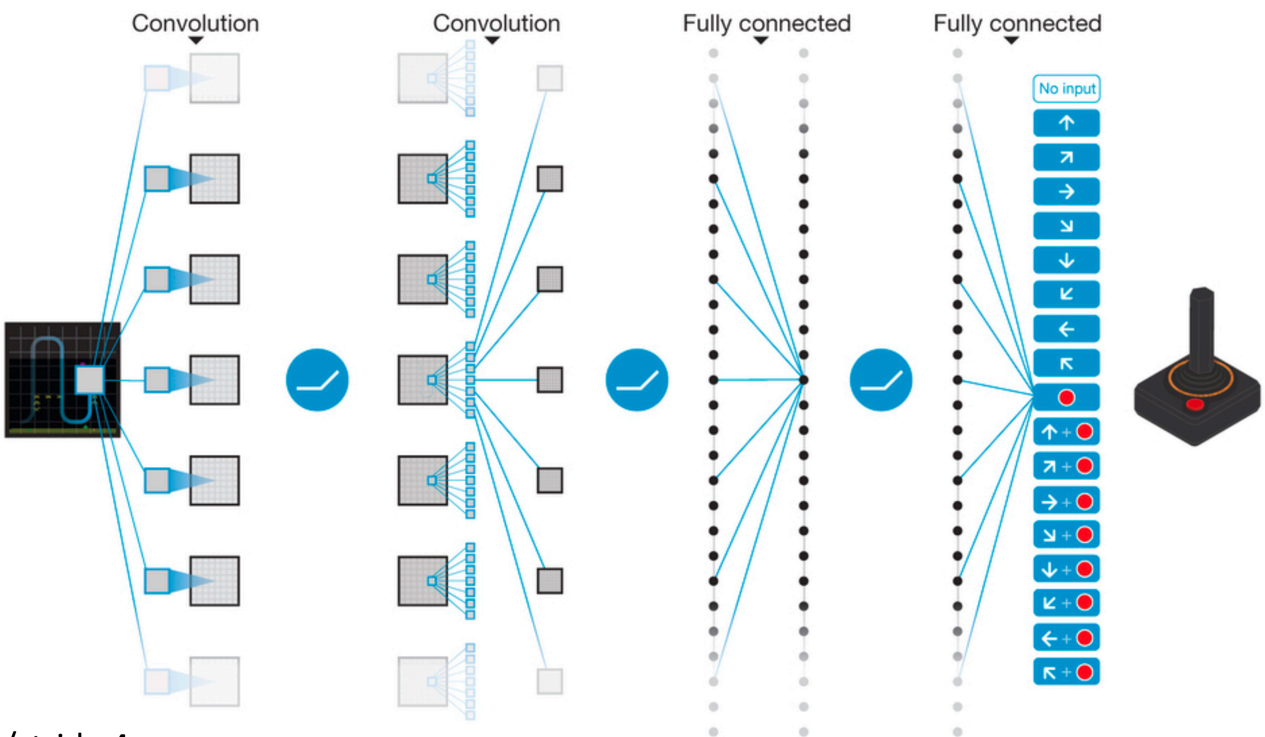


Talk Outline

- Classical RL
  - Algorithms
    - Policy Gradients
    - Actor-Critic
    - Q-learning
  - Representation
- Representation in exploration
- Different Approaches / Architectures
  - Value Iteration Networks
  - Predictron
  - Modular Networks
  - Option-Critic
  - Feudal Networks
- Meta learning
  - MAML
  - RL2

DQN

[Mnih et al, Nature 2015]

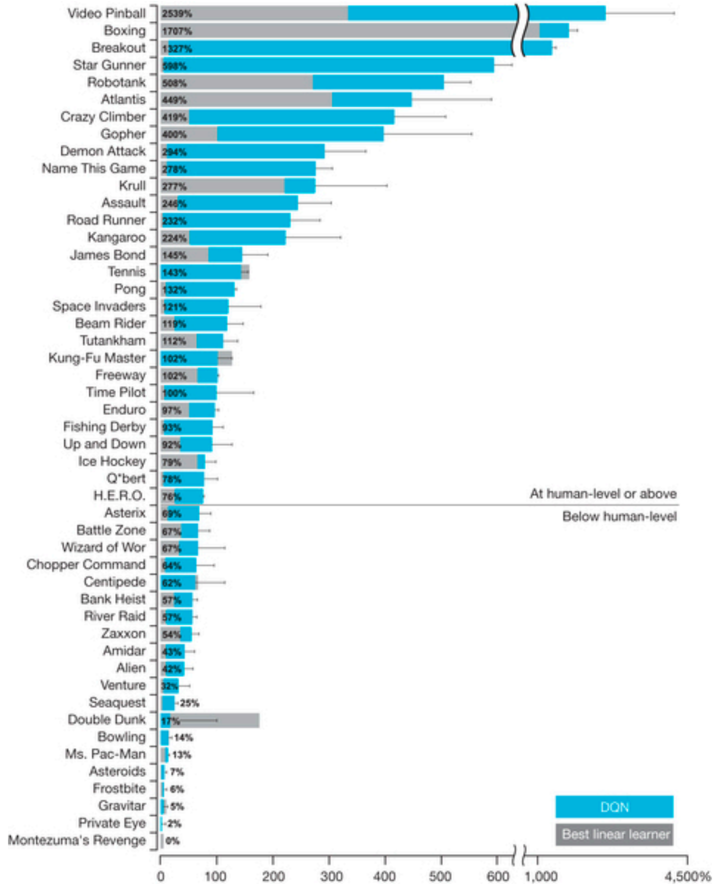


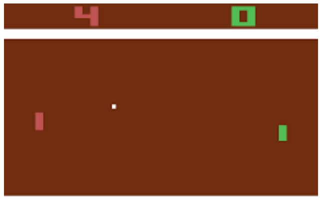
Q

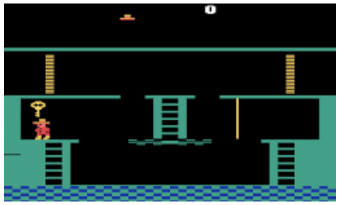
Conv1: 32 8x8 filters w/stride 4
Conv2: 64 4x4 filters with stride 2
Conv3: 64 3x3 filters with stride 1

DQN

[Mnih et al, Nature 2015]

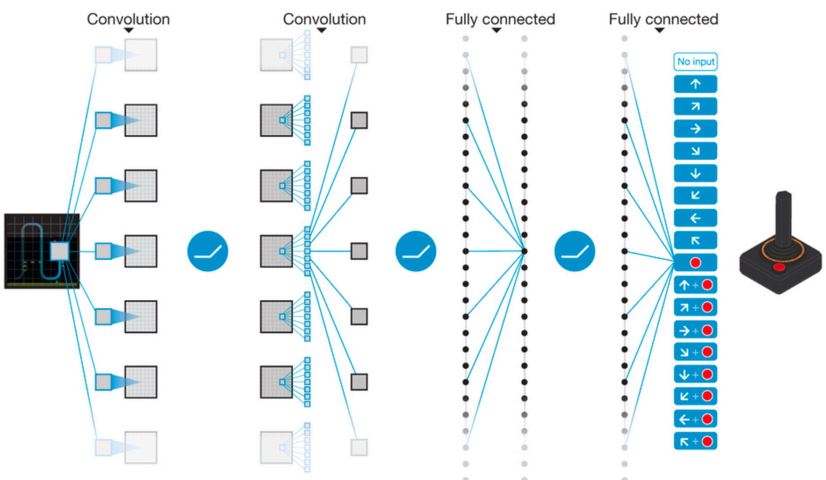




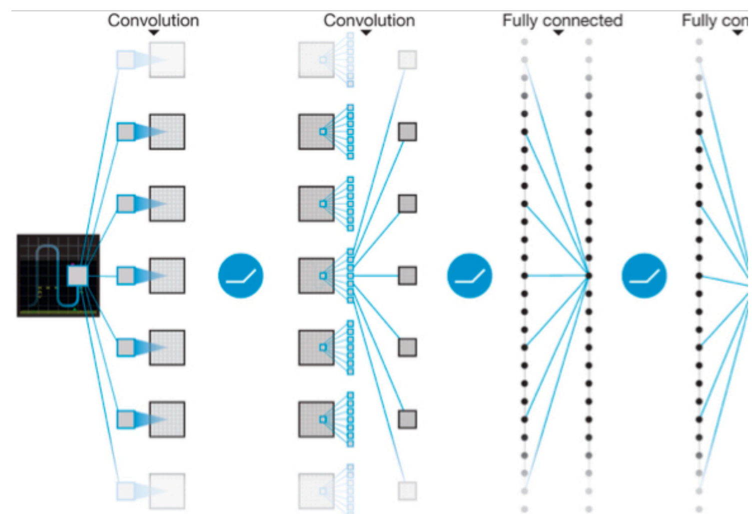


TRPO + GAE

[Schulman et al, 2015]



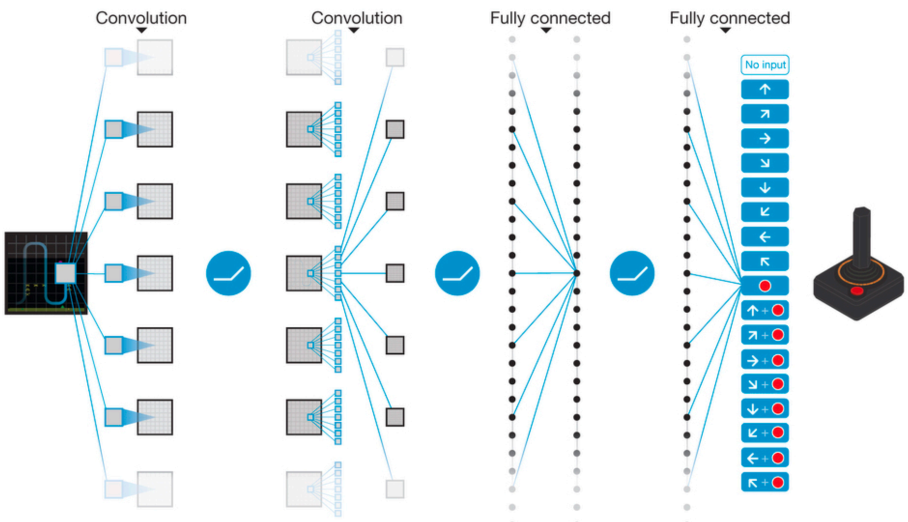
π_θ

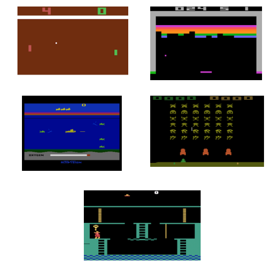


V_ϕ

A3C

[Mnih et al, 2016]

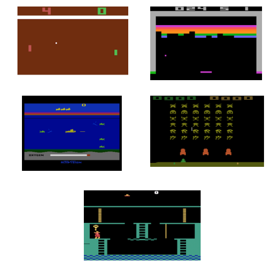


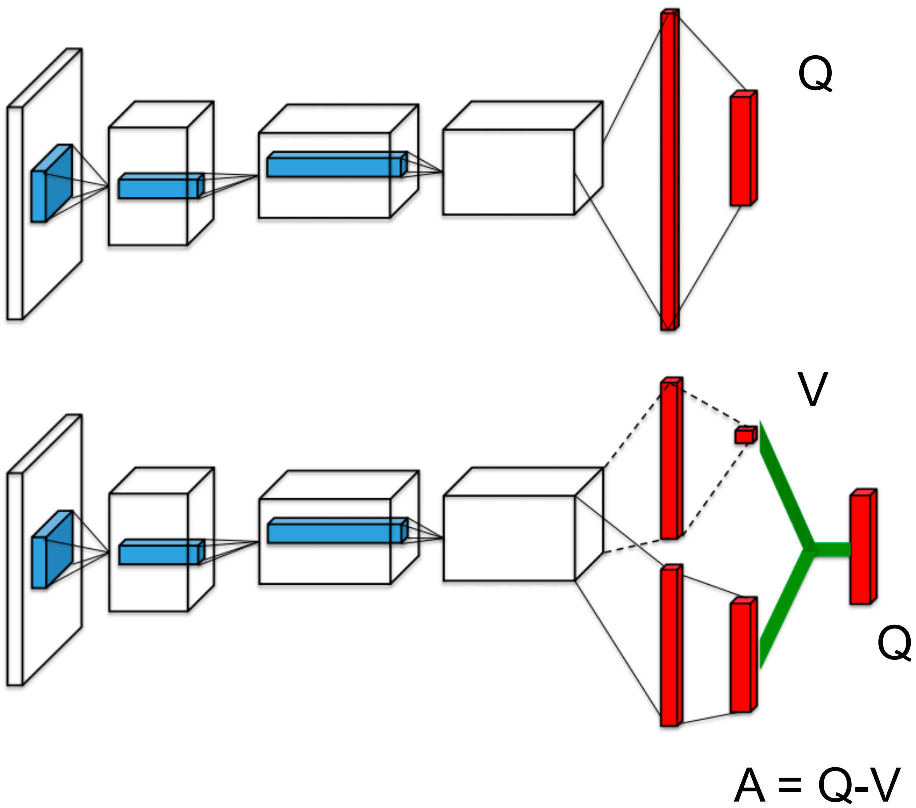


$$\pi_\theta \quad V_\phi$$

Dueling Network

[Wang et al, 2016]





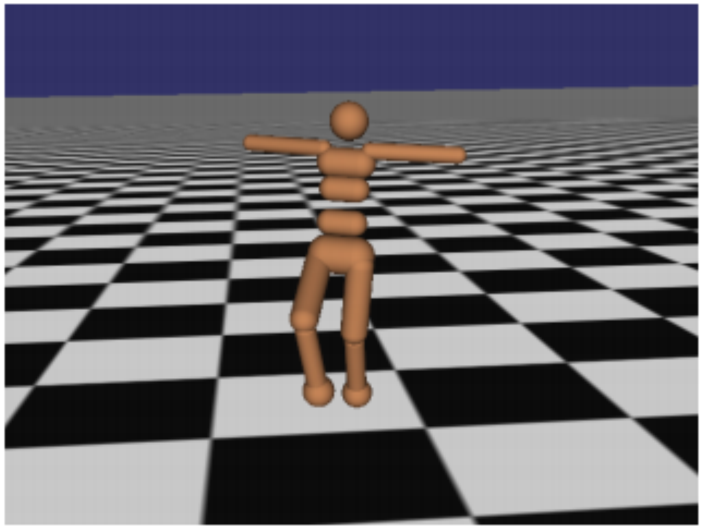
DQN

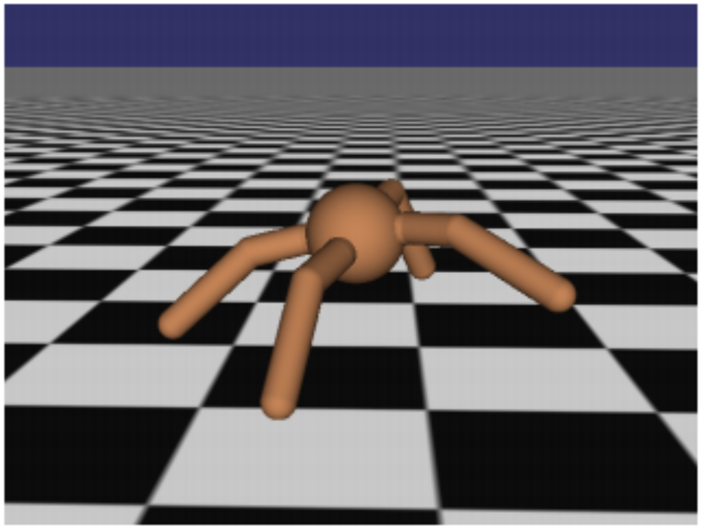
Dueling

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\alpha)\right)$$

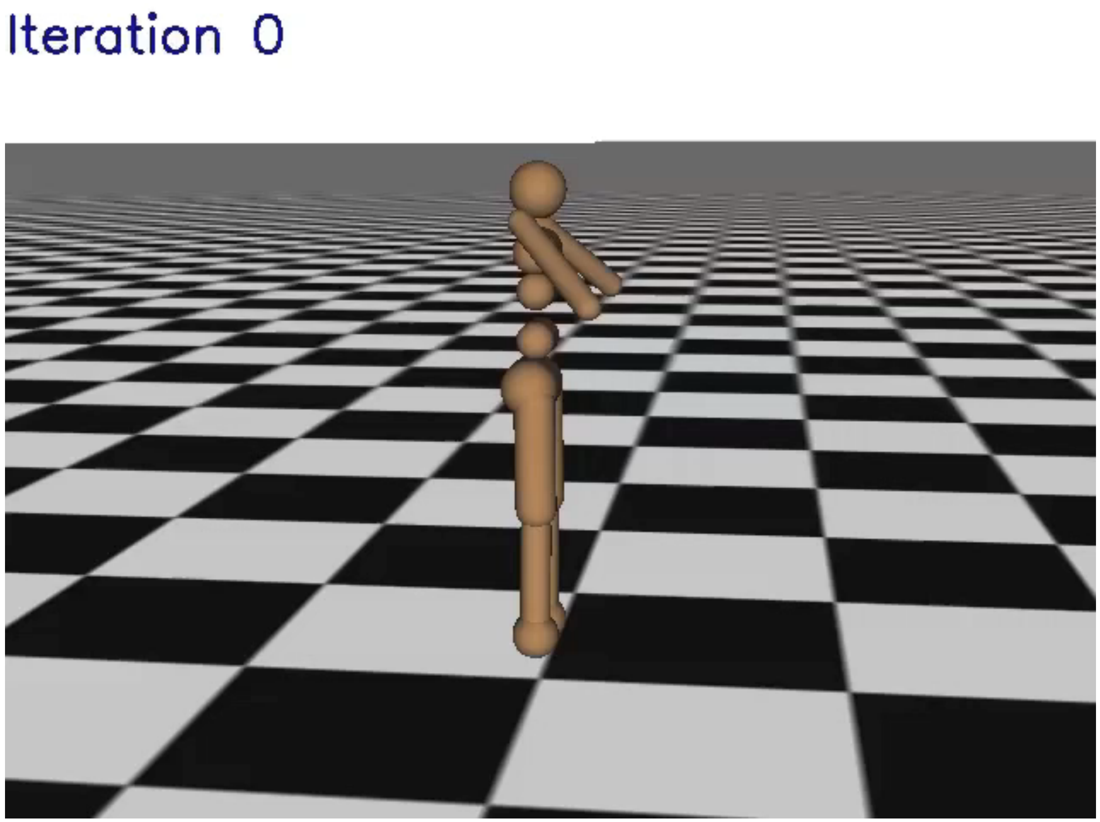
TRPO+GAE – Continuous Control

- Feedforward: h100 – h50 – h25 for both policy and value function



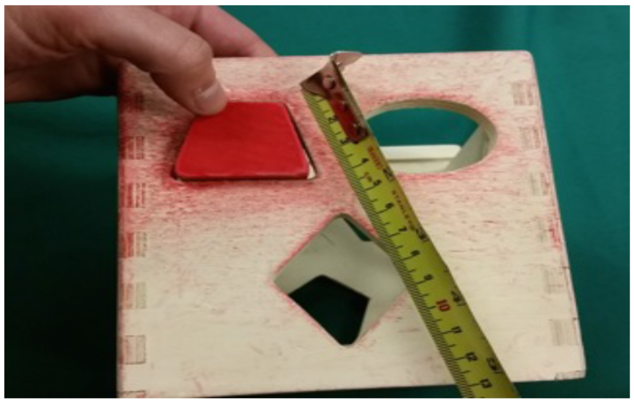


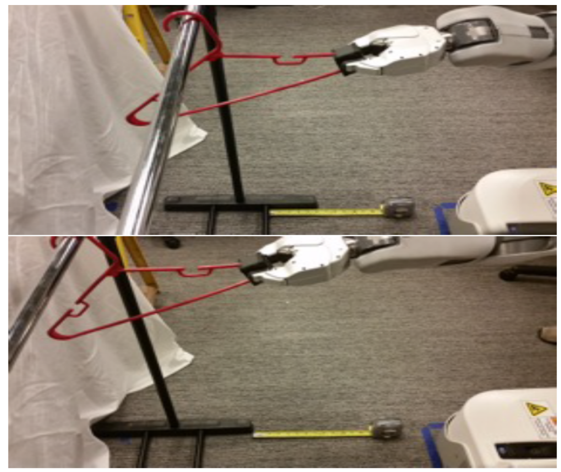
Learning Locomotion (TRPO + GAE)



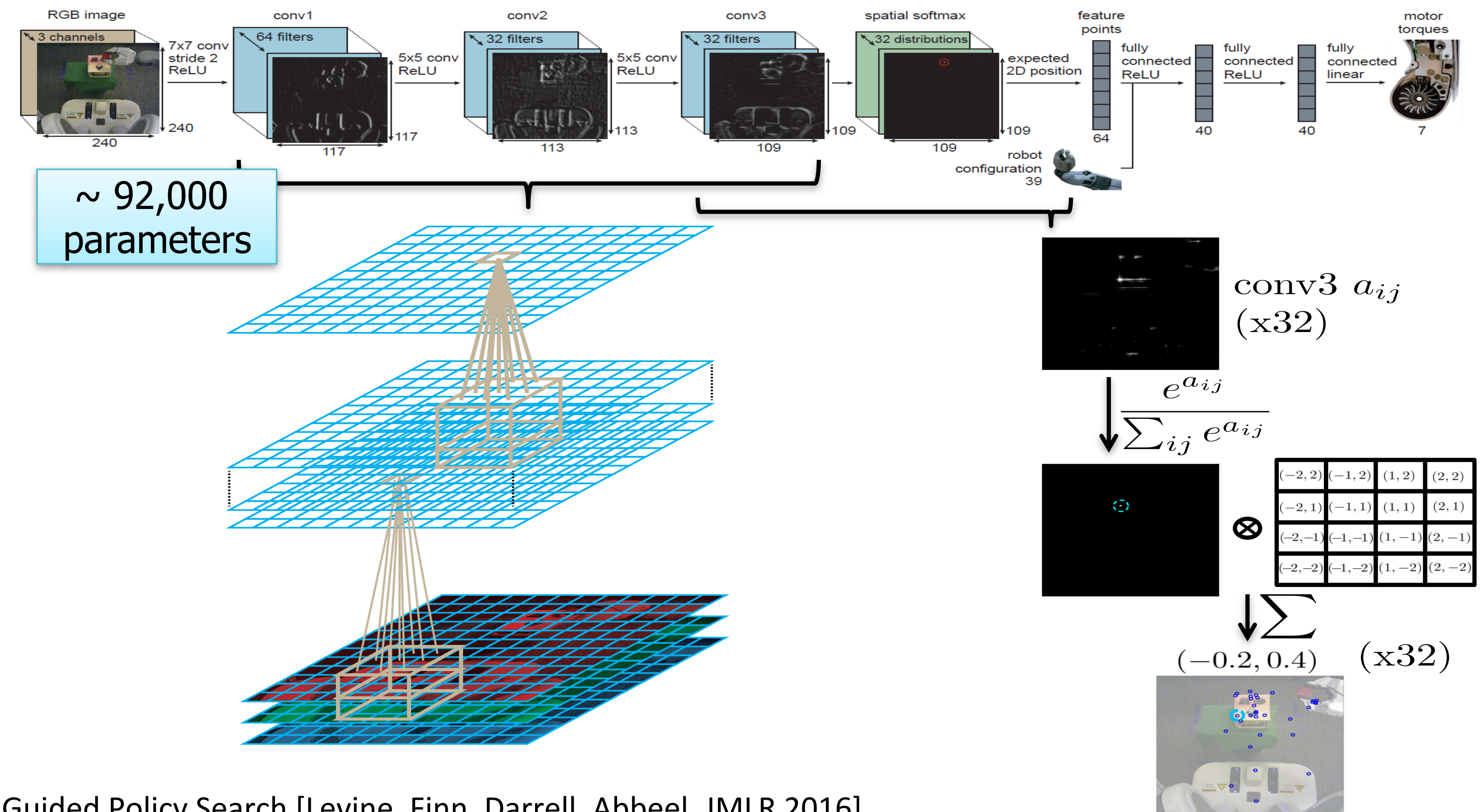
[Schulman, Moritz, Levine, Jordan, Abbeel, 2016]

Real Robots





[Levine*, Finn*, Darrell, Abbeel, JMLR 2016



Guided Policy Search [Levine, Finn, Darrell, Abbeel, JMLR 2016]

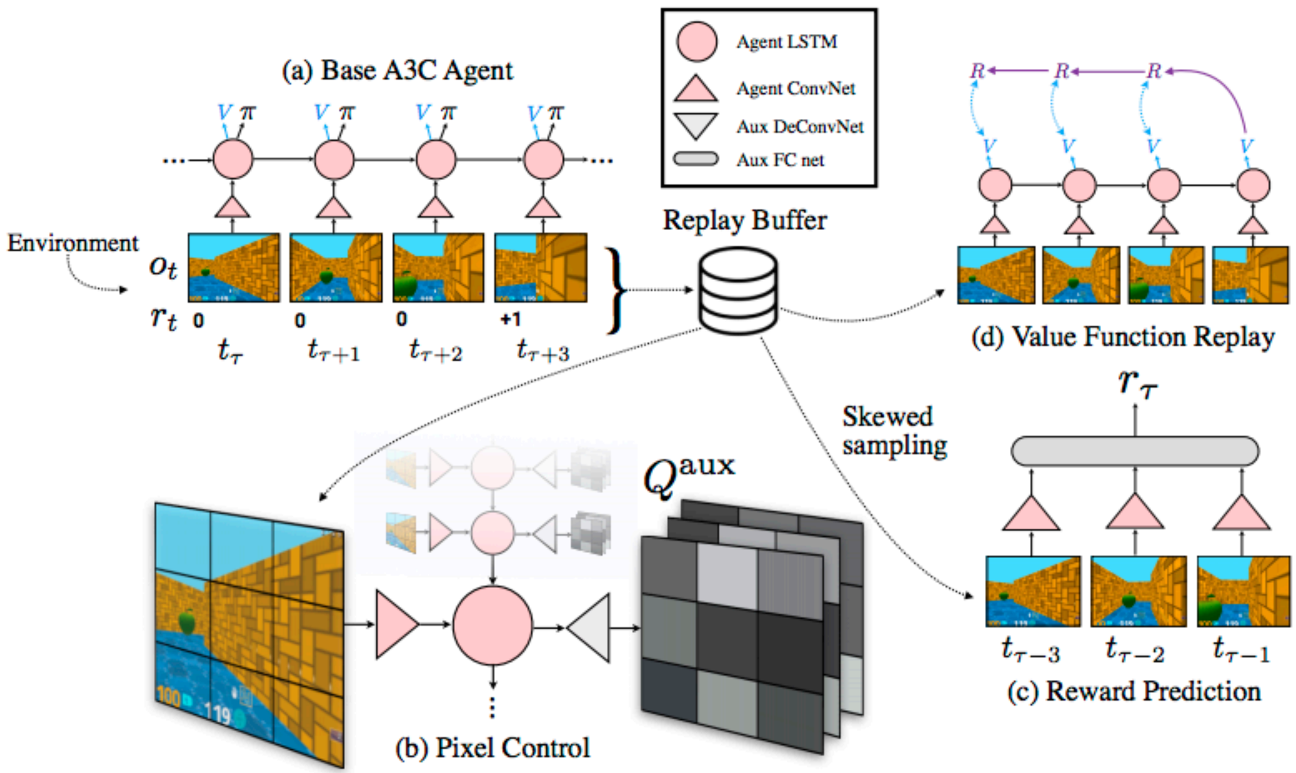
Learned Skills

[Levine*, Finn*, Darrell, Abbeel, JMLR 2016

Other Architectures

- A3C with LSTMs [Mnih et al, 2016]
- Memory [Oh et al, 2016]
- Auxiliary losses: Learning to Navigate [Mirowski, Pascanu et al, 2016]; [Jaderberg, Mnih, Czarnecki et al, 2016]

Auxiliary Losses



Unreal Agent [Jaderberg, Mnih, Czarnecki et al, 2016]

Talk Outline

- Classical RL
  - Algorithms
    - Policy Gradients
    - Actor-Critic
    - Q-learning
  - Representation
- ***Representation in exploration***
- Different Approaches / Architectures
  - Value Iteration Networks
  - Predictron
  - Modular Networks
  - Option-Critic
  - Feudal Networks
- Meta learning
  - RL2
  - MAML

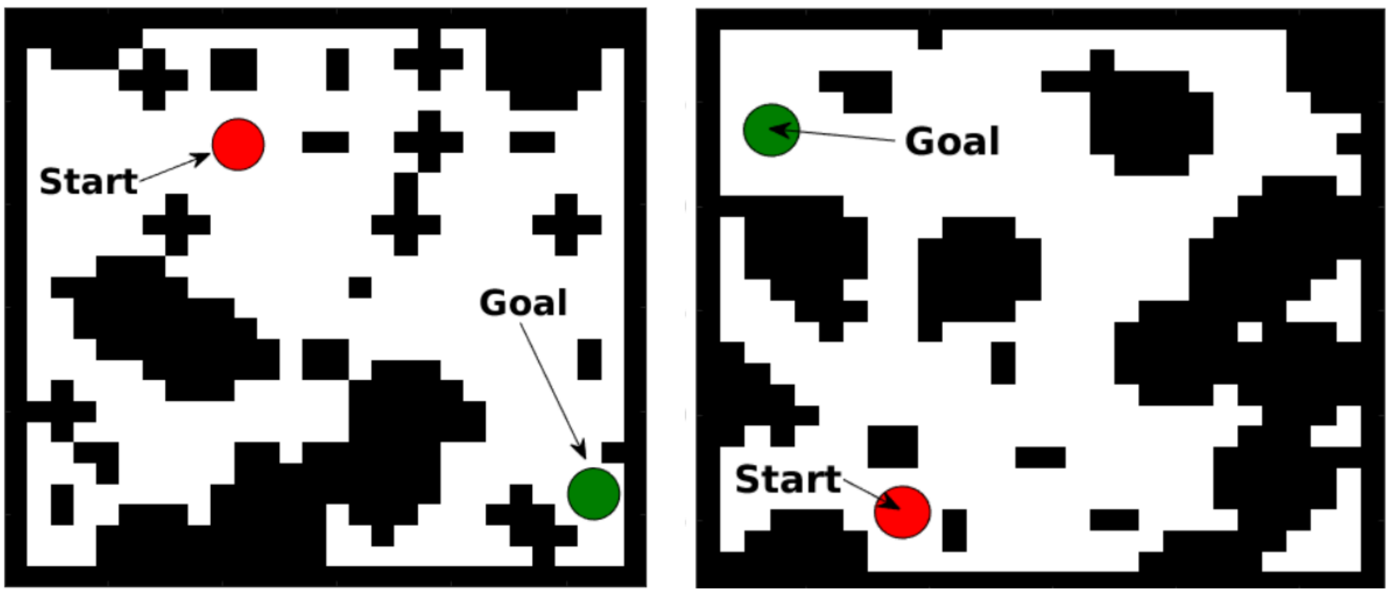
Exploration through Reward Bonuses

- Idea: infuse additional reward for encountering novelty / learning something about the environment
  - VIME: Bayesian Neural Net – Houthooft et al, 2016
  - Pixel-CNN density estimation – Ostrovski et al, 2017
  - …

Talk Outline

- Classical RL
  - Algorithms
    - Policy Gradients
    - Actor-Critic
    - Q-learning
  - Representation
- Representation in exploration
- ***Different Approaches / Architectures***
  - Value Iteration Networks
  - Predictron
  - Modular Networks
  - Option-Critic
  - Feudal Networks
- Meta learning
  - RL2
  - MAML

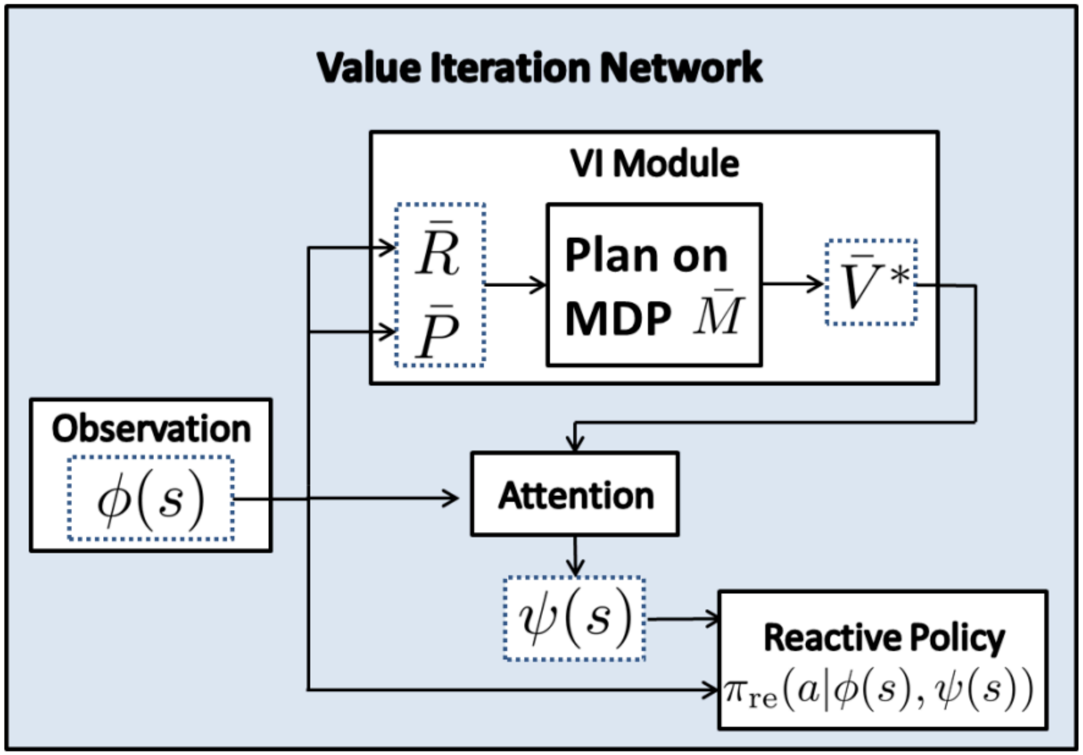
Value Iteration Networks

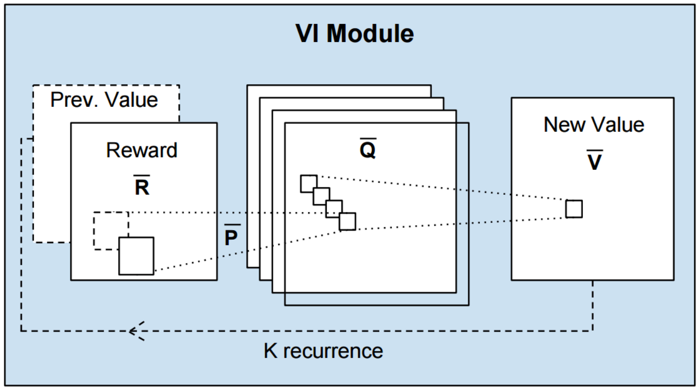


[Tamar, Wu, Thomas, Levine, Abbeel, NIPS 2016]

Value Iteration Network (VIN)

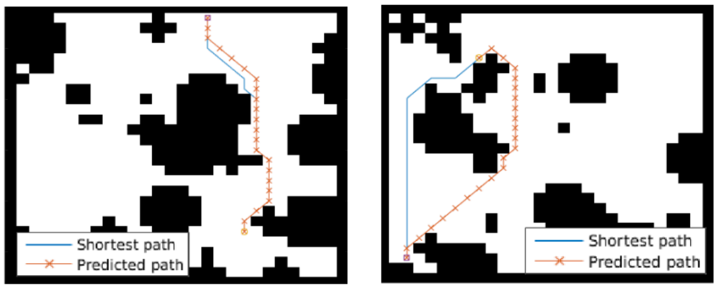
- VIN Architecture:



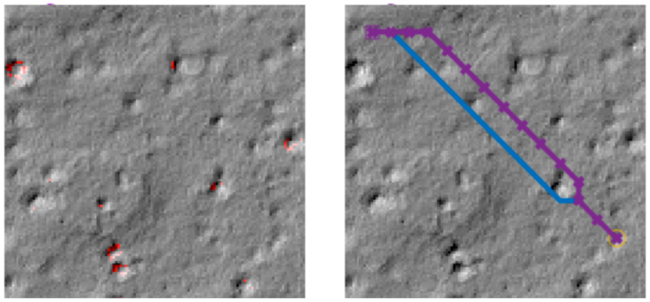


End-to-end differentiable planner!

VIN -- Evaluation

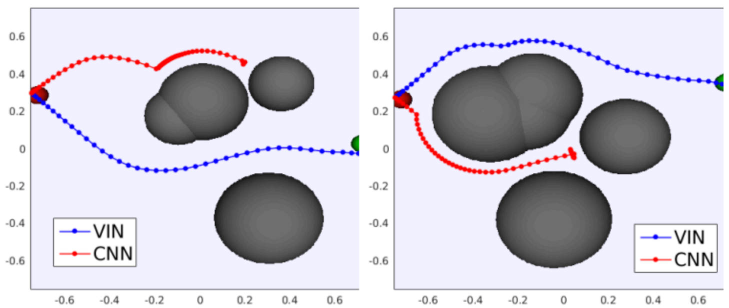


Gridworld navigation



Mars navigation

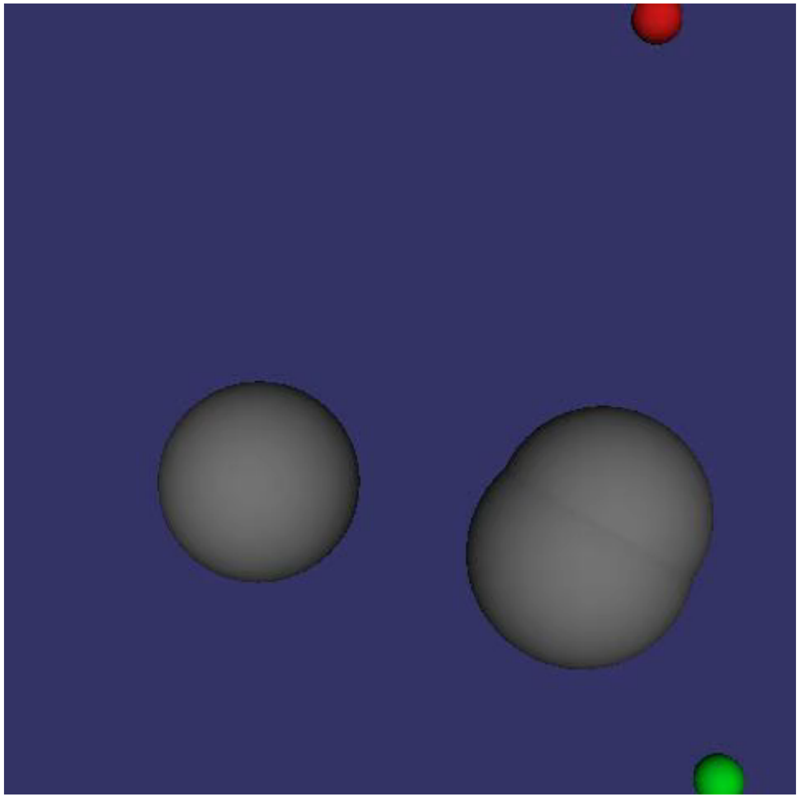
Network	Train Error	Test Error
VIN	0.30	0.35
CNN	0.39	0.58



Continuous domain

[Tamar, Wu, Thomas, Levine, Abbeel, NIPS 2016]

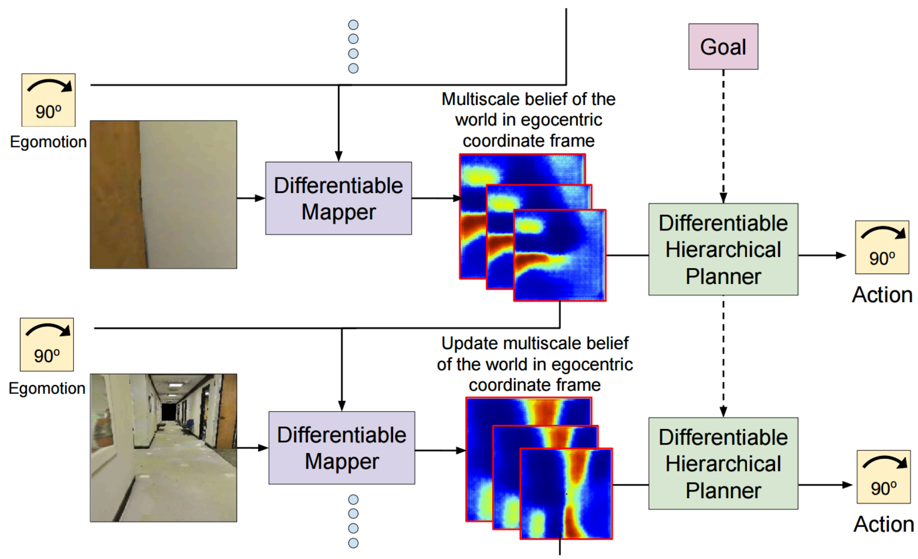
Continuous Domain -- Video



[Tamar, Wu, Thomas, Levine, Abbeel, NIPS 2016]

1st Person Mapping + Navigation with VIN

[Gupta, Davidson, Levine, Sukthankar, Malik, 2017]



After learning

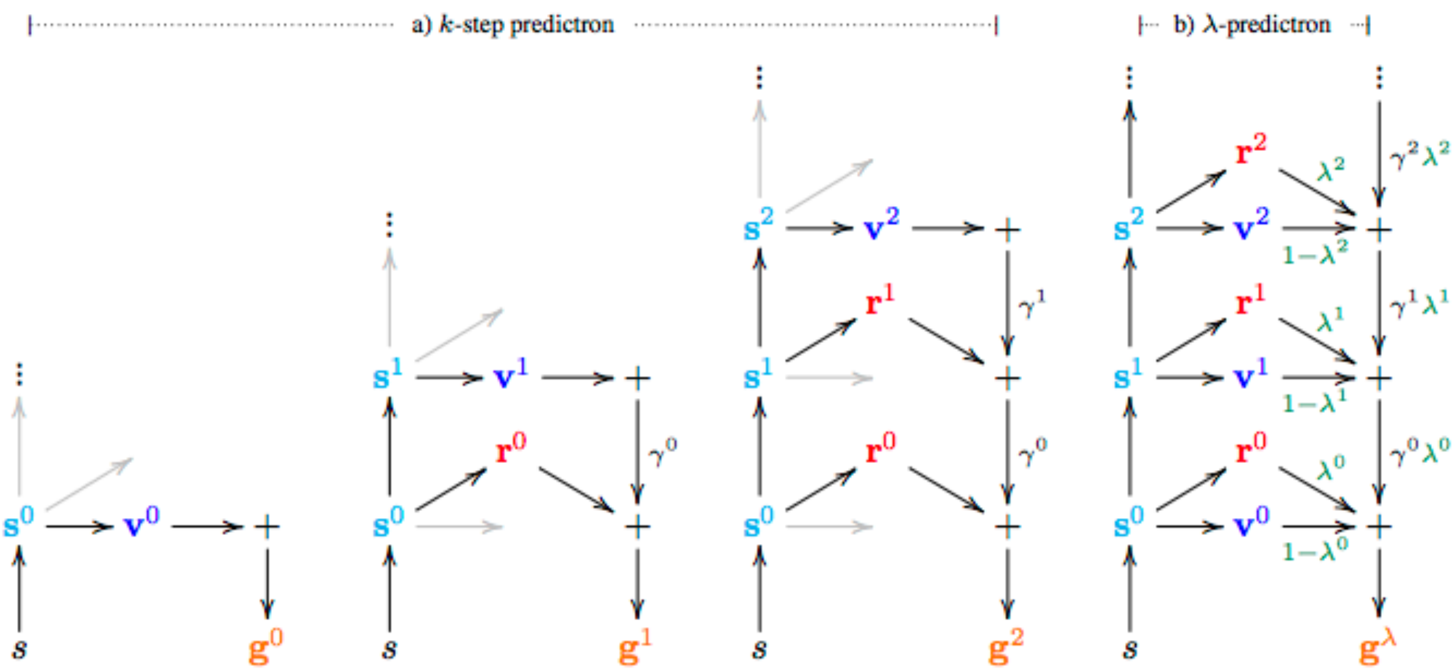
Most Closely Related Work

- Embed to Control – Watter, Springenberg, Boedecker, Riedmiller, 2015
- Hindsight MPC – Aviv Tamar et al., ICRA 2017

Talk Outline

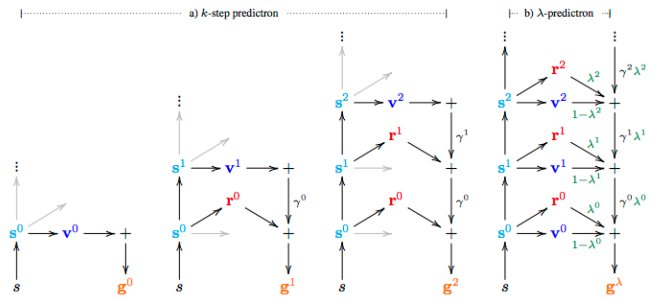
- Classical RL
  - Algorithms
    - Policy Gradients
    - Actor-Critic
    - Q-learning
  - Representation
- Representation in exploration
- Different Approaches / Architectures
  - Value Iteration Networks
  - ***Predictron***
  - ***Modular Networks***
  - ***Option-Critic***
  - ***Feudal Networks***
- Meta learning
  - RL2
  - MAML

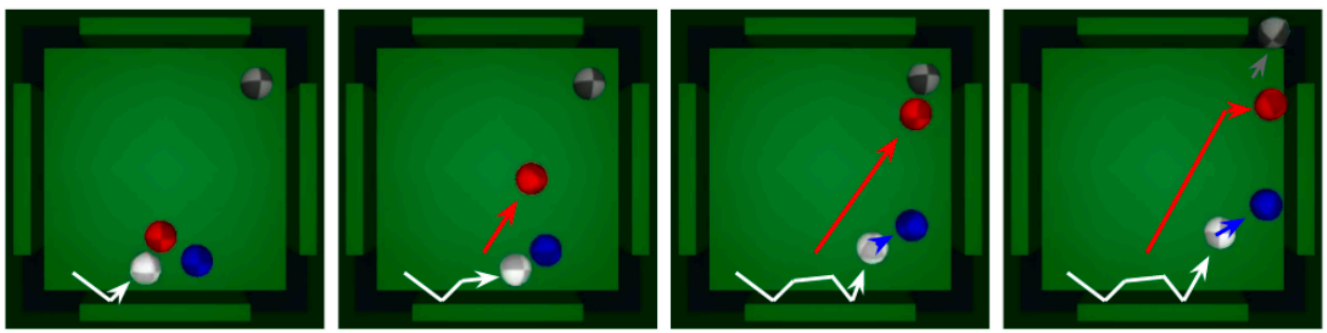
Predictron



[Silver et al, 2016]

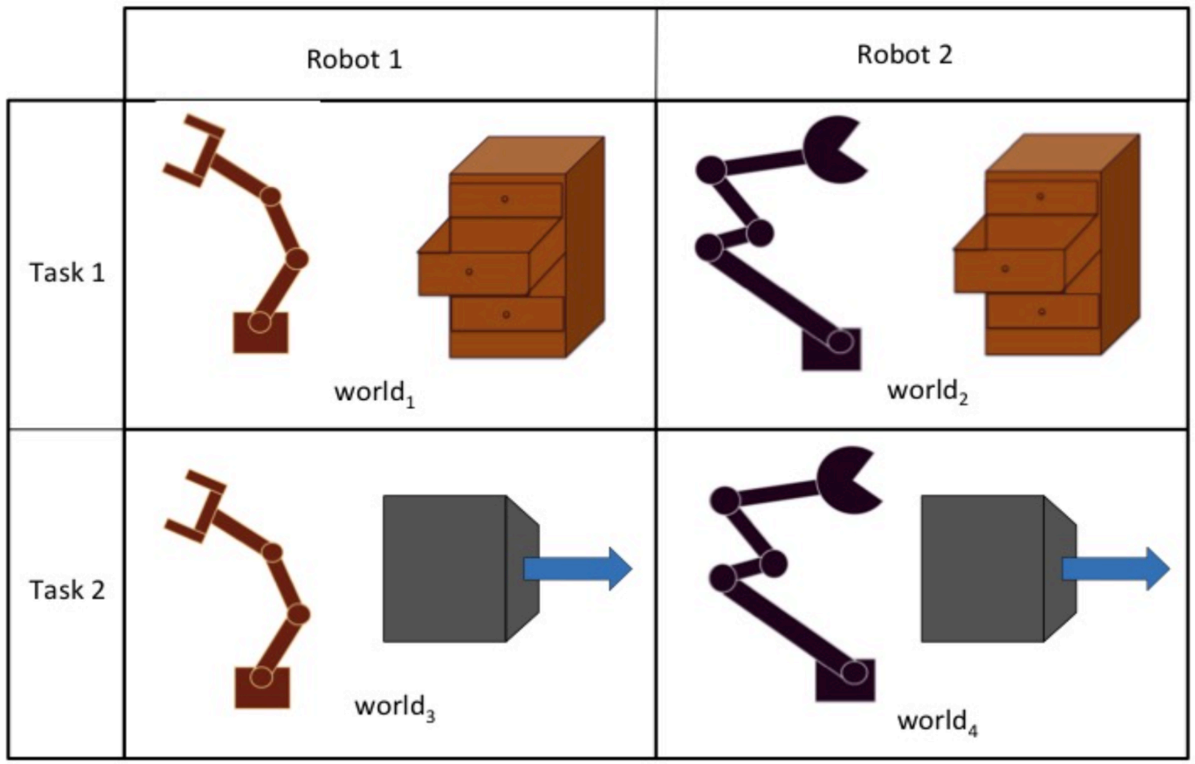
Predictron



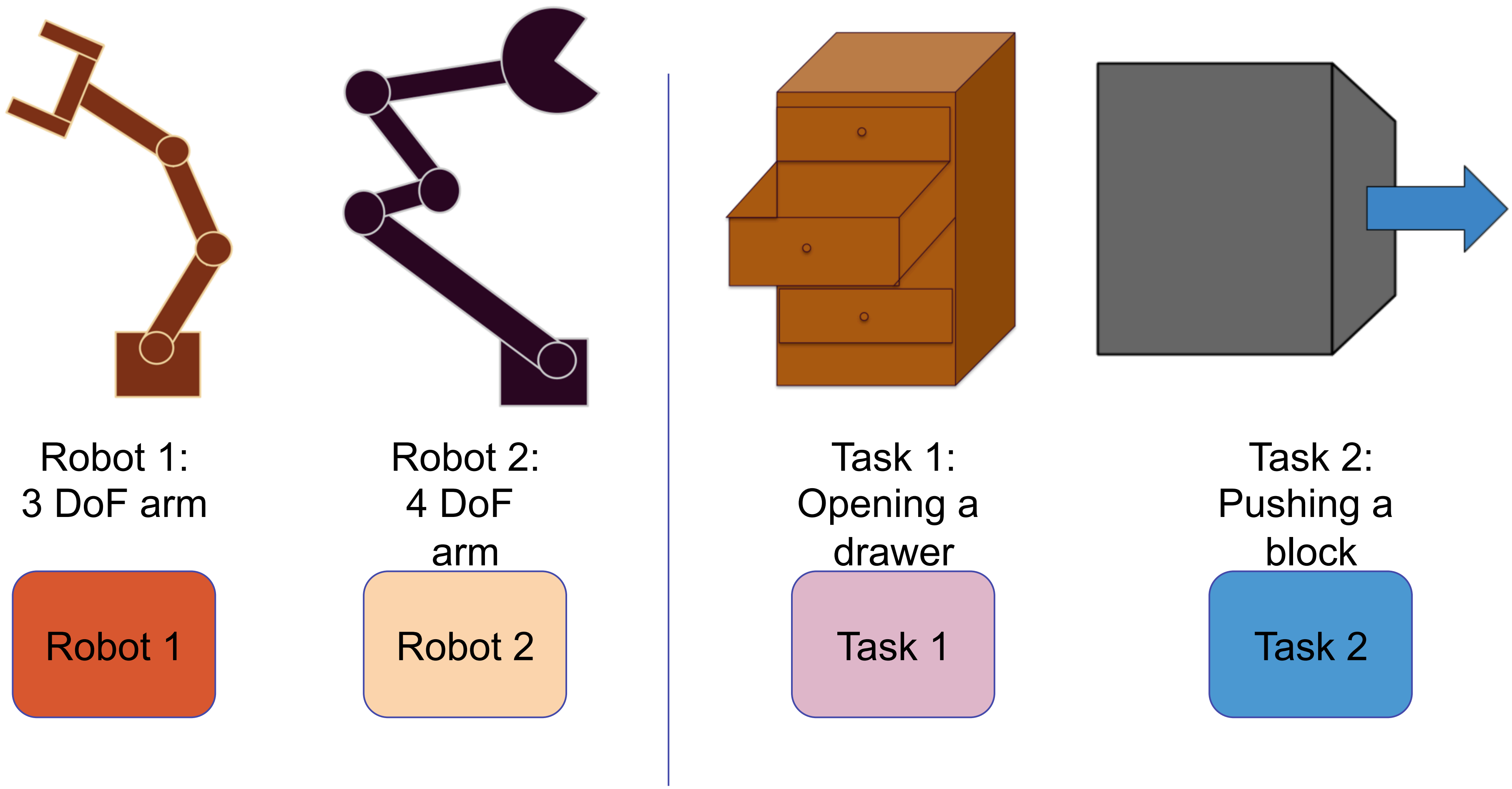


[Silver et al, 2016]

Modular Networks



[Devin, Gupta et al, 2016]



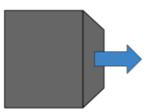
[Devin, Gupta et al, 2016]

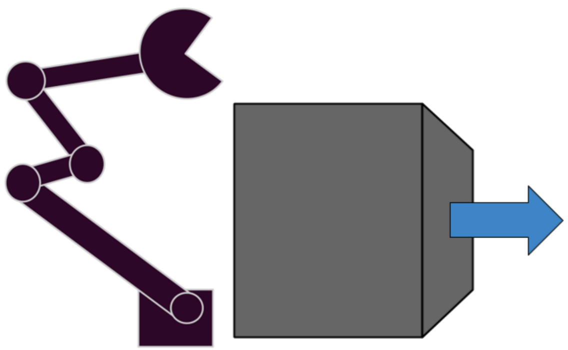
Robot 1

Robot 2

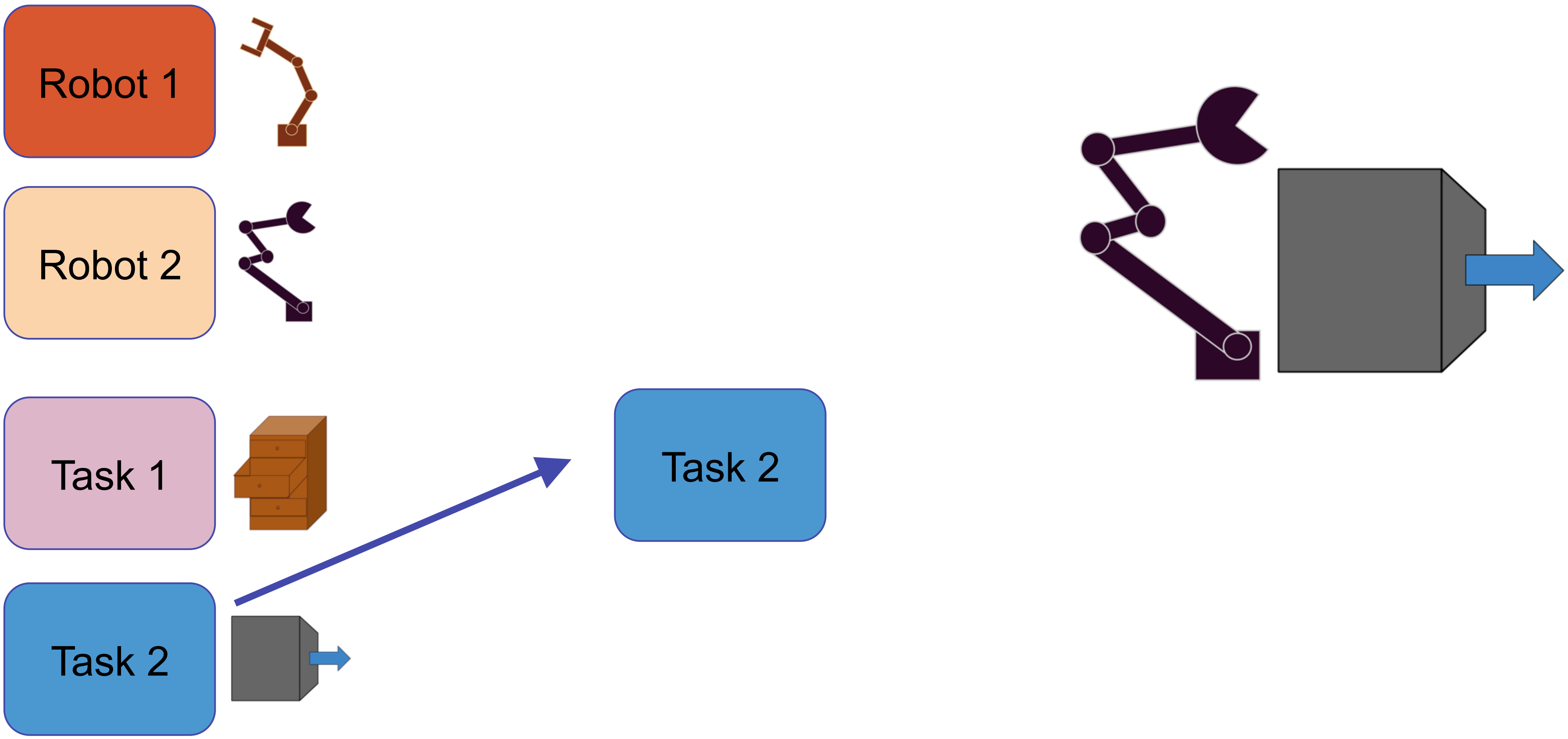
Task 1

Task 2

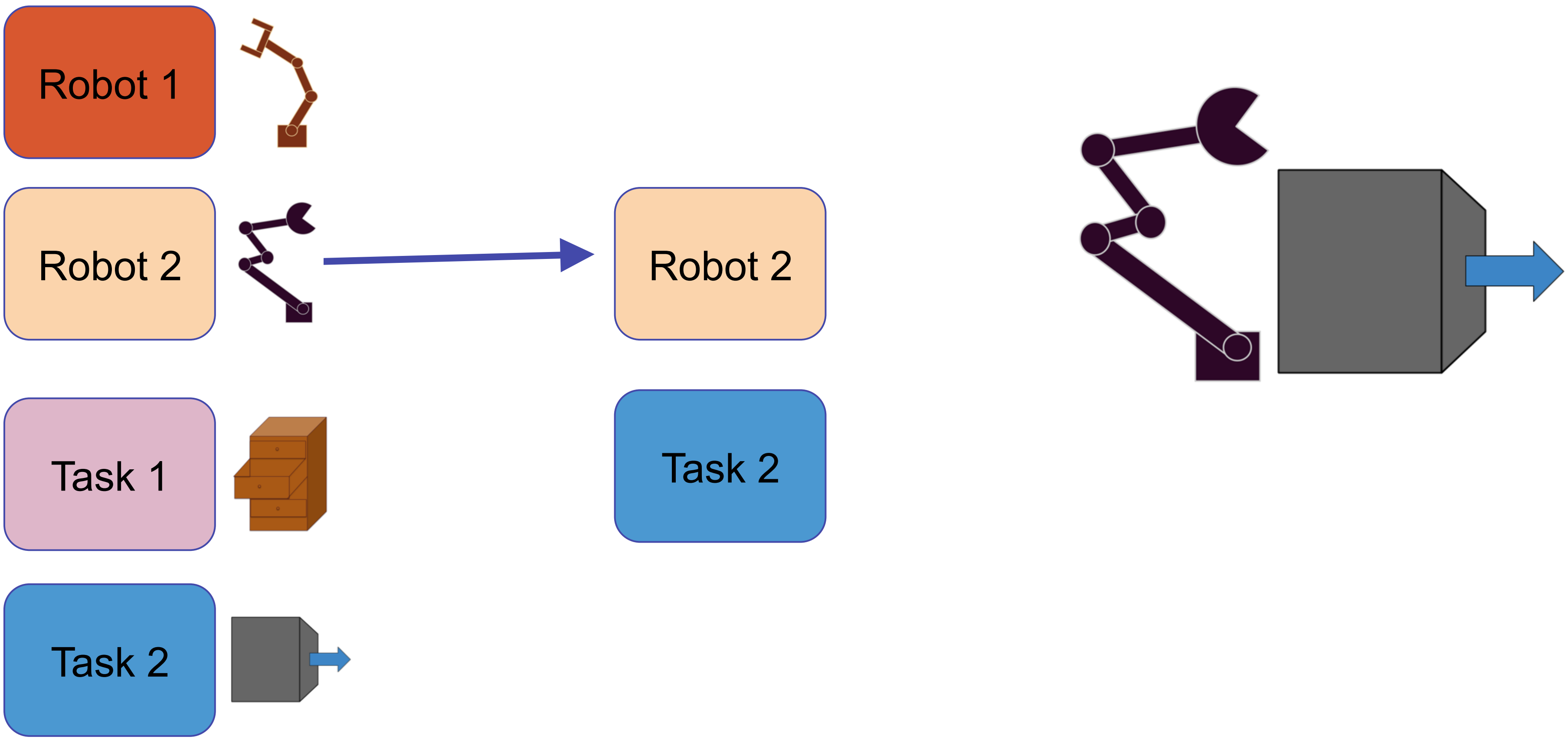




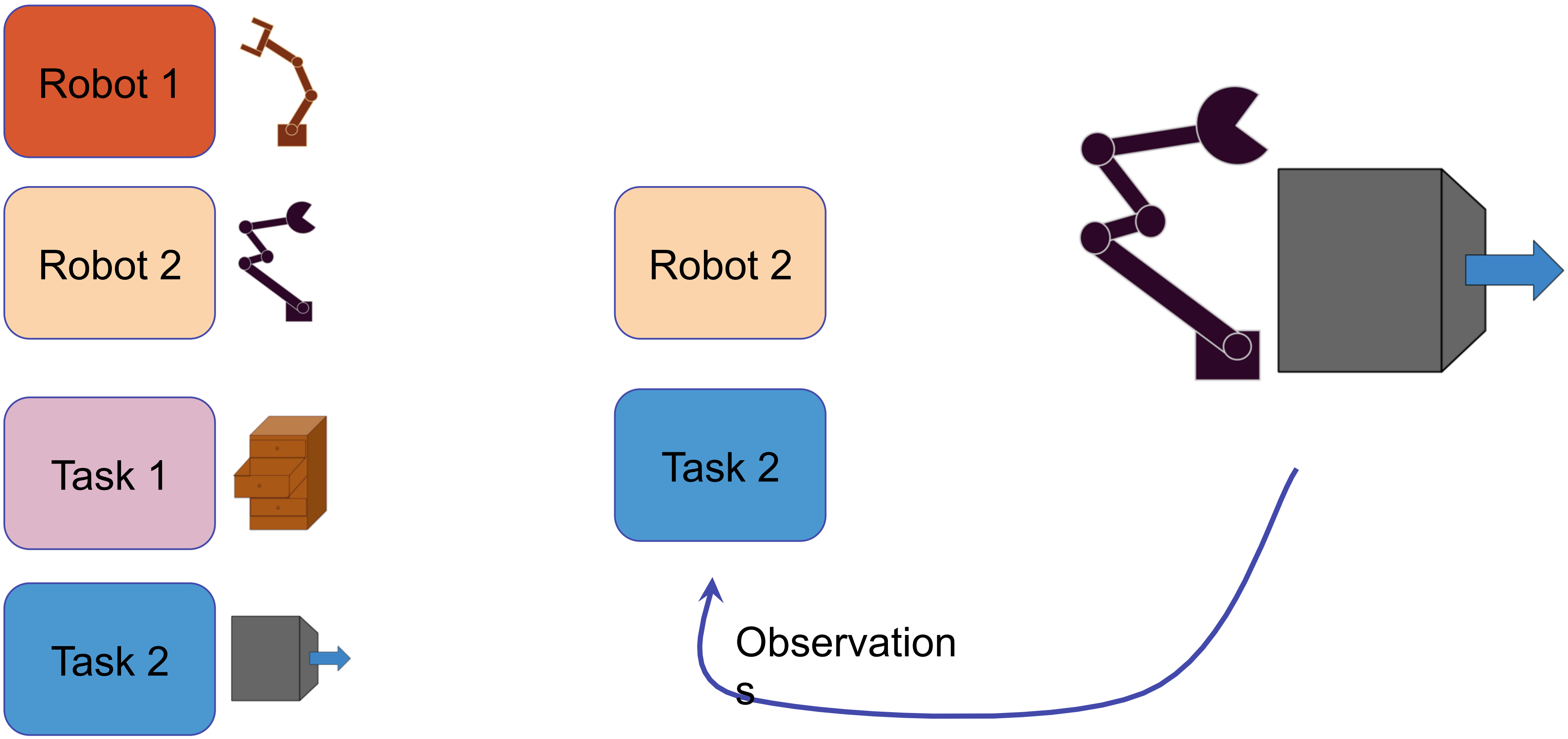
[Devin, Gupta et al, 2016]



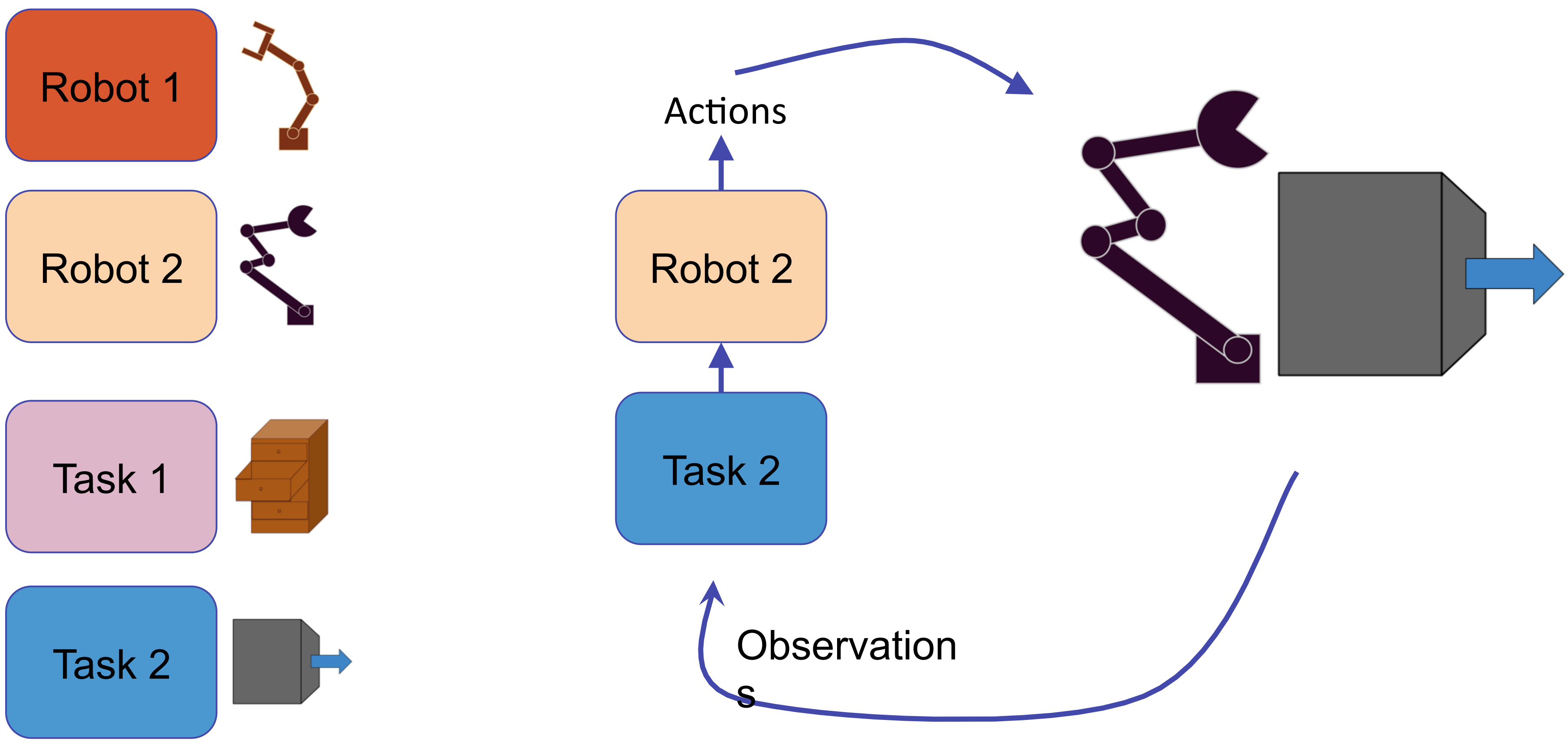
[Devin, Gupta et al, 2016]



[Devin, Gupta et al, 2016]

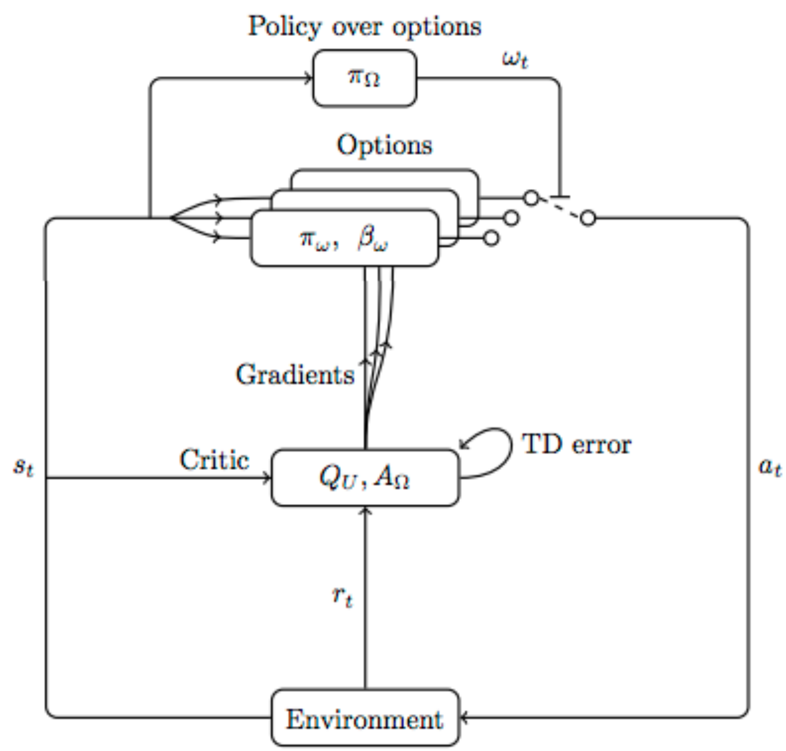


[Devin, Gupta et al, 2016]



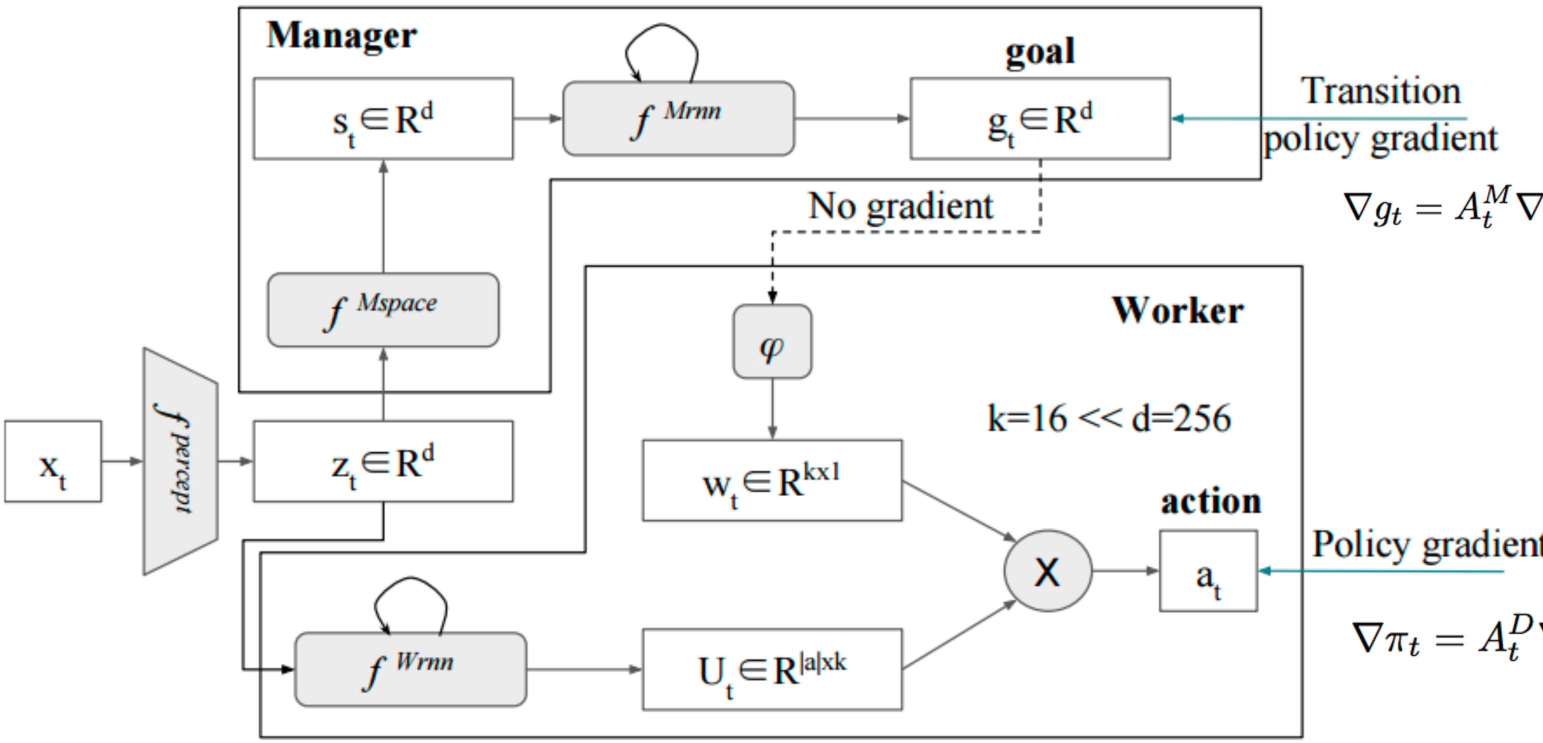
[Devin, Gupta et al, 2016]

Option-Critic Architecture



[Bacon, Harb, Precup, 2017]

Feudal Networks



$$\nabla g_t = A_t^M \nabla_\theta d_{\cos}(s_{t+c} - s_t, g_t(\theta))$$

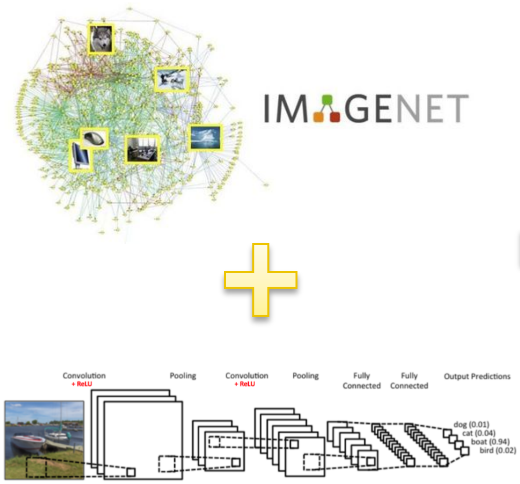
$$\nabla \pi_t = A_t^D \nabla_\theta \log \pi(a_t | x_t; \theta)$$

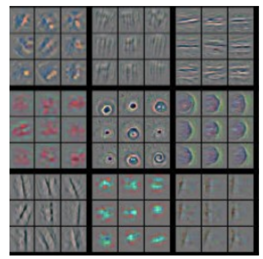
[Dayan and Hinton, 1993; Vezhnevets et al, 2017]

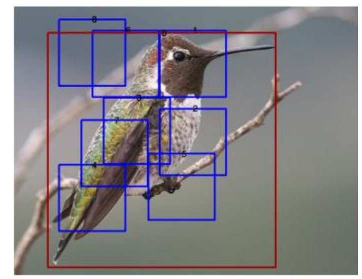
Talk Outline

- Classical RL
  - Algorithms
    - Policy Gradients
    - Actor-Critic
    - Q-learning
  - Representation
- Representation in exploration
- Different Approaches / Architectures
  - Value Iteration Networks
  - Predictron
  - Modular Networks
  - Option-Critic
  - Feudal Networks
- ***Meta learning***
  - ***MAML***
  - RL2

Learning useful representations with deep learning



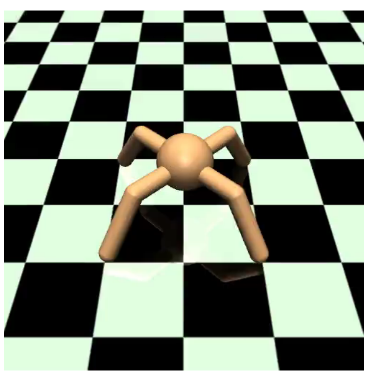




Where are the “ImageNet” features of motor control?

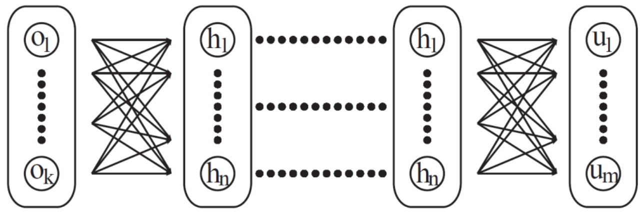
The trouble with RL

- Large-scale
- Emphasizes diversity
- Evaluated on generalization



- Small-scale
- Emphasizes mastery
- Evaluated on performance
- Can we force RL to generalize?

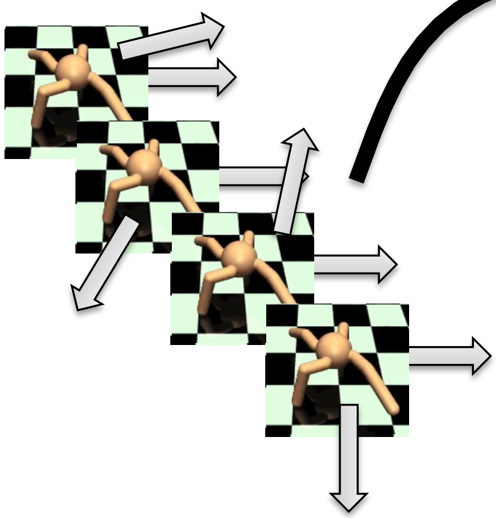
Multi-task training for adaptability

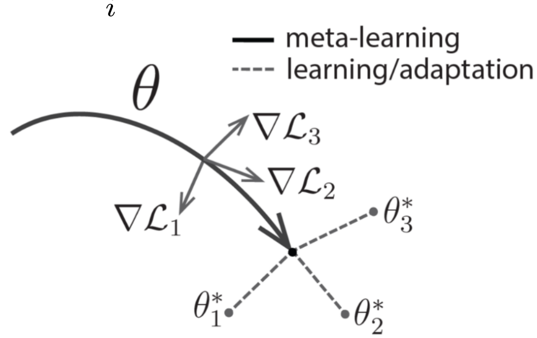


θ

$$\cancel{\theta \leftarrow \theta + \alpha \nabla_\theta R(\theta)}$$

$$\sum_i L_i(\theta + \alpha \nabla_\theta R_i(\theta))$$



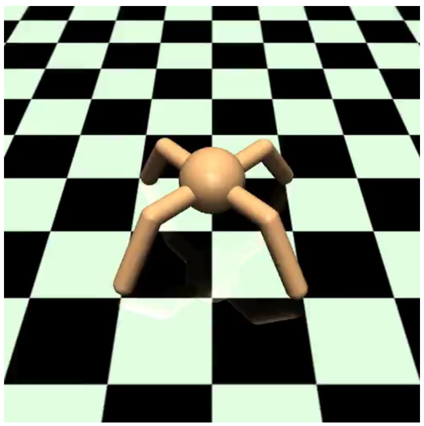


θ'

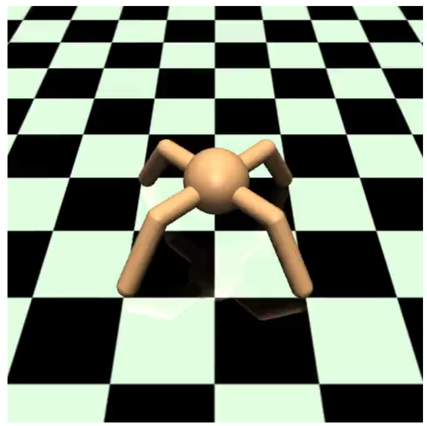
[Finn, Abbeel, Levine, 2017]

Model-agnostic meta-learning: forward/backward locomotion

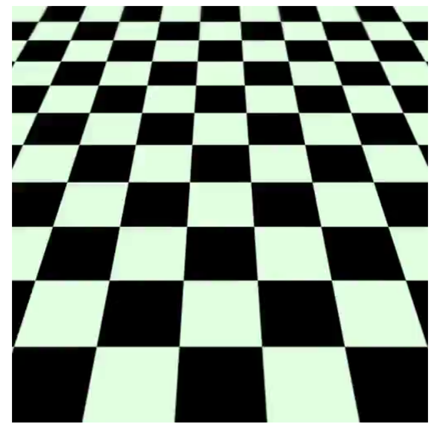
after MAML training

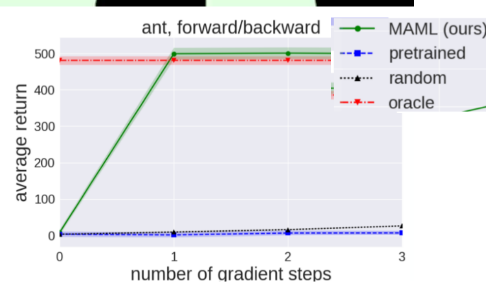


after 1 gradient step (forward reward)



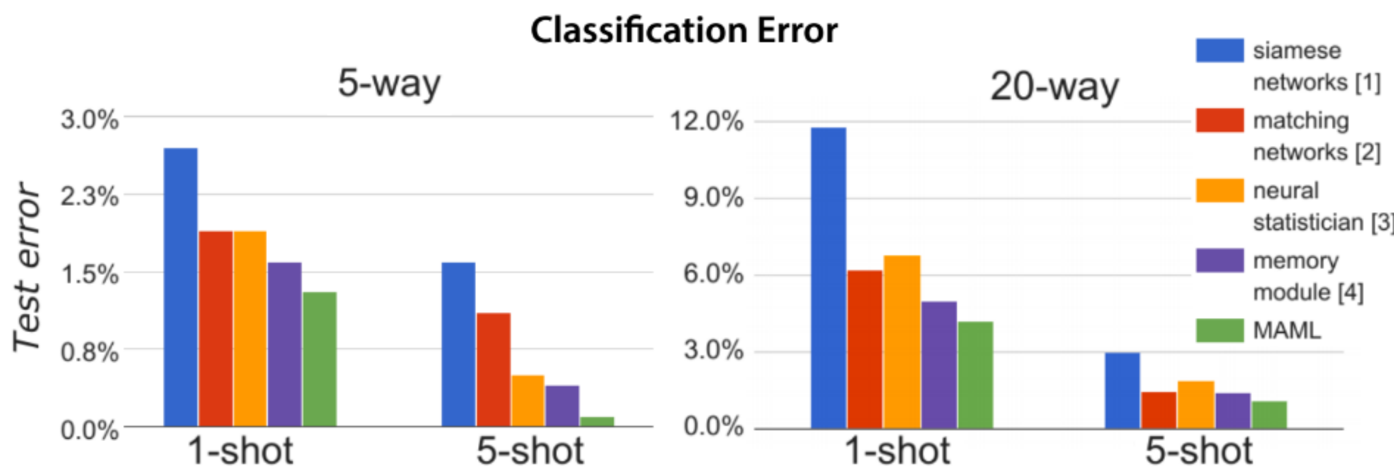
after 1 gradient step (backward reward)





Model-agnostic meta-learning benchmark results

Omniglot Few-Shot Classification

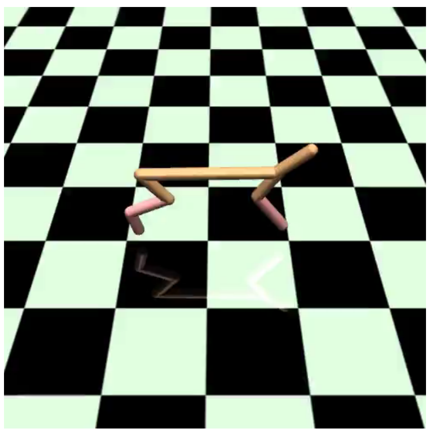


Omniglot Dataset: 1200 training classes, 423 test classes

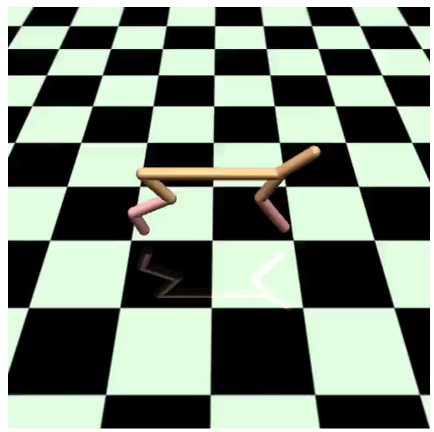
[1] Koch '15 [2] Vinyals et al. '16
[3] Edwards & Storkey '17 [4] Kaiser et al. '17

Model-agnostic meta-learning: forward/backward locomotion

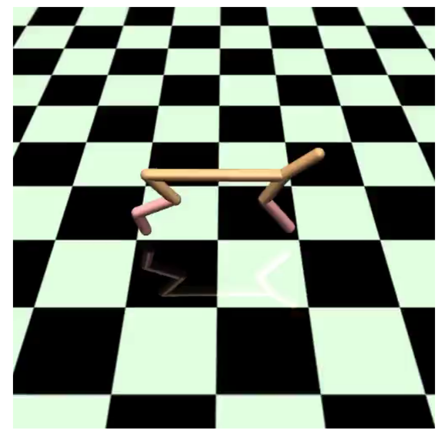
after MAML training



after 1 gradient step (backward reward)



after 1 gradient step (forward reward)



Talk Outline

- Classical RL
  - Algorithms
    - Policy Gradients
    - Actor-Critic
    - Q-learning
  - Representation
- Representation in exploration
- Different Approaches / Architectures
  - Value Iteration Networks
  - Predictron
  - Modular Networks
  - Option-Critic
  - Feudal Networks
- ***Meta learning***
  - MAML
  - ***RL2***

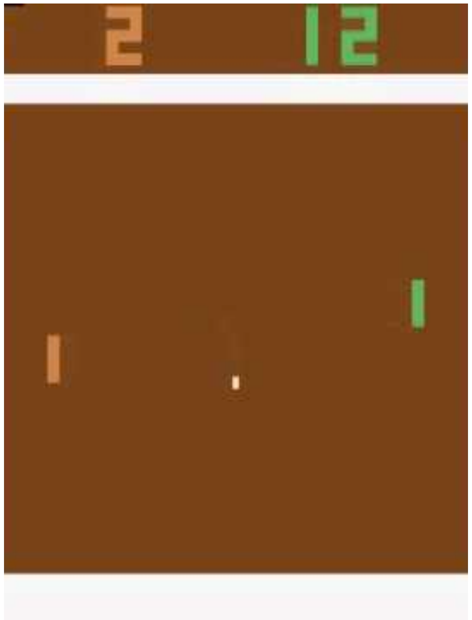
Speed of Learning

Deep RL (DQN) vs. **Human**

Score: **18.9**

#Experience measured in real time: 40 days

"Slow"



Score: 9.3

#Experience measured in real time: **2 hours**

"Fast"

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Starting Observations

- TRPO, DQN, A3C are fully general RL algorithms
  - i.e., for any MDP that can be mathematically defined, these algorithms are equally applicable

- MDPs encountered in real world

  = tiny, tiny subset of all MDPs that could be defined

- Can we design "fast" RL algorithms that take advantage of such knowledge?

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Research Questions

- How to acquire a good prior for real-world MDPs?
  - Or for starters, e.g., for real-games MDPs?
- How to design algorithms that make use of such prior information?

Key idea: Learn a fast RL algorithm that encodes this prior

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Formulation

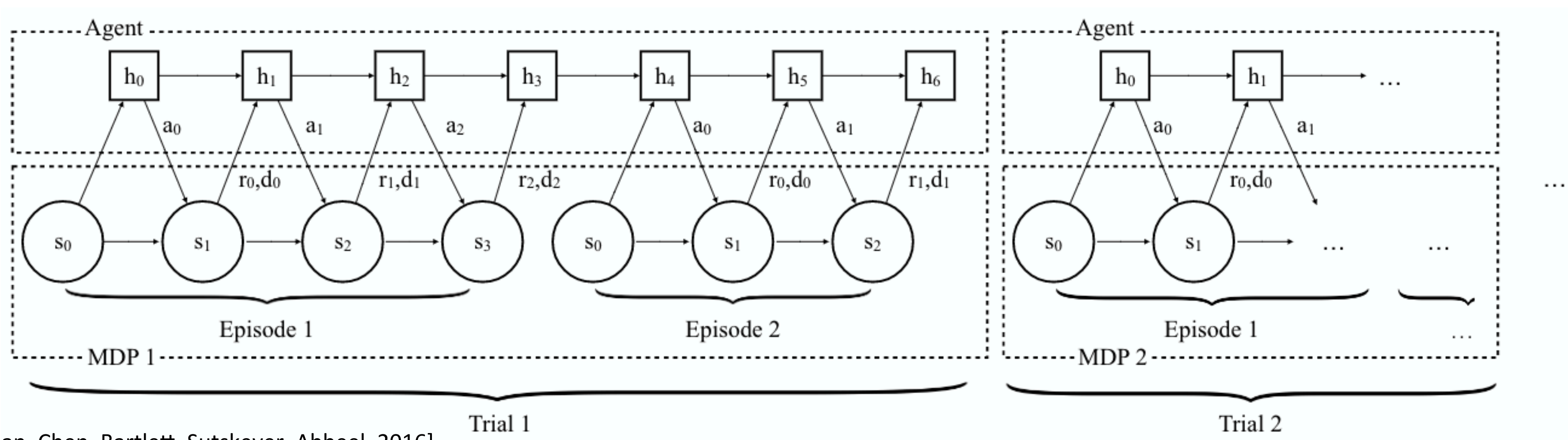
- Given: Distribution over relevant MDPs
- Train the fast RL algorithm to be fast on a training set of MDPs

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Learning the Fast RL Algorithm

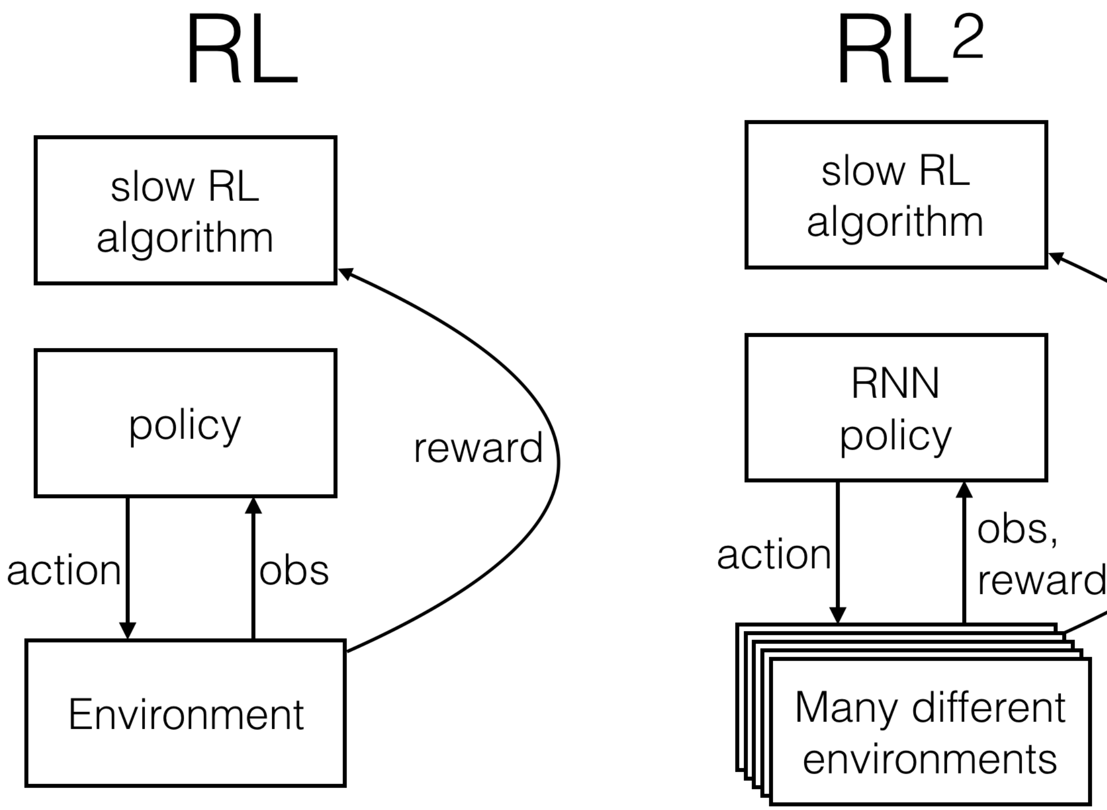
- Representation of the fast RL algorithm:
  - RNN = generic computation architecture
  - different weights in the RNN means different RL algorithm
  - different activations in the RNN means different current policy

- Training setup:



[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Formulation



[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Alternative View on RL2

- RNN = policy for acting in a POMDP
  - Part of what's not observed in the POMDP is which MDP the agent is in

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Related Work

- Wang et al., (2016) Learning to Reinforcement Learn, in submission to ICLR 2017,
- Chen et al. (2016) Learning to Learn for Global Optimization of Black Box Functions
- Andrychowicz et al., (2016) Learning to learn by gradient descent by gradient descent
- Santoro et al., (2016) One-shot Learning with Memory-Augmented Neural Networks
- Larochelle et al., (2008), Zero-data Learning of New Tasks.
- Younger et al. (2001), Meta learning with backpropagation
- Schmidhuber et al. (1996), Simple principles of metalearning

Evaluation

- Multi-Armed Bandits
- Provably (asymptotically) optimal RL algorithms have been invented by humans: Gittins index, UCB1, Thompson sampling, …

5-armed bandit
(source: ebay)

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation

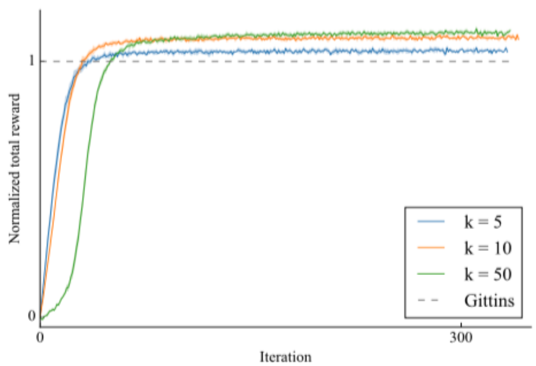
- Multi-Armed Bandits

Setup	Random	Gittins	TS	OTS	UCB1	ϵ-Greedy	Greedy	RL^2
$n = 10, k = 5$	5.0	**6.6**	5.7	6.5	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 10$	5.0	**6.6**	5.5	6.2	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 50$	5.1	6.5	5.2	5.5	**6.6**	6.5	6.5	**6.8**
$n = 100, k = 5$	49.9	**78.3**	74.7	**77.9**	**78.0**	75.4	74.8	**78.7**
$n = 100, k = 10$	49.9	**82.8**	76.7	81.4	82.4	77.4	77.1	**83.5**
$n = 100, k = 50$	49.8	**85.2**	64.5	67.7	84.3	78.3	78.0	**84.9**
$n = 500, k = 5$	249.8	**405.8**	**402.0**	**406.7**	**405.8**	388.2	380.6	**401.6**
$n = 500, k = 10$	249.0	**437.8**	429.5	**438.9**	**437.1**	408.0	395.0	432.5
$n = 500, k = 50$	249.6	**463.7**	427.2	437.6	457.6	413.6	402.8	438.9

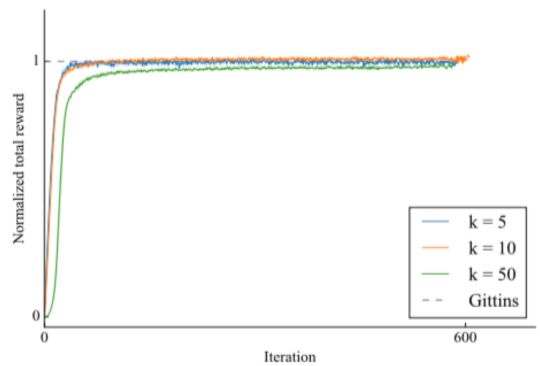
[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation

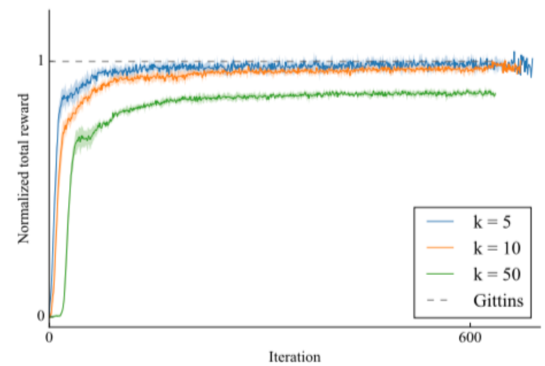
- Multi-Armed Bandits



(a) $n = 10$



(b) $n = 100$

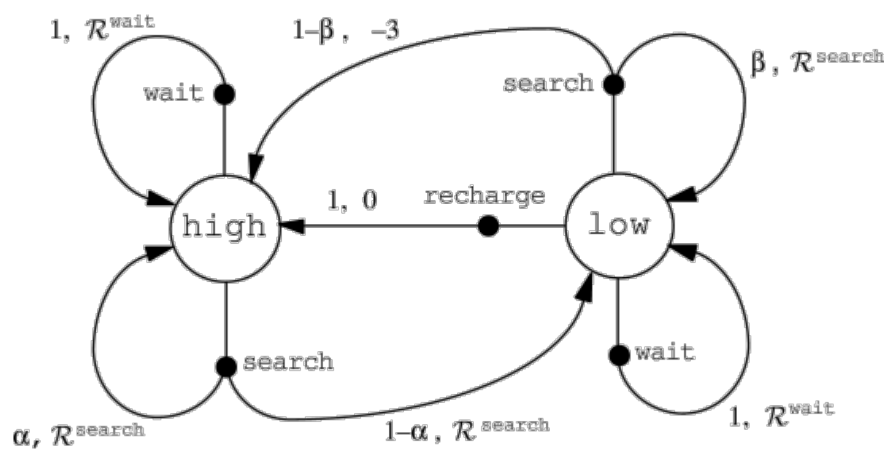


(c) $n = 500$

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation: Tabular MDPs

- Provably (asymptotically) optimal algorithms:
  - BEB, PSRL, UCRL2, …



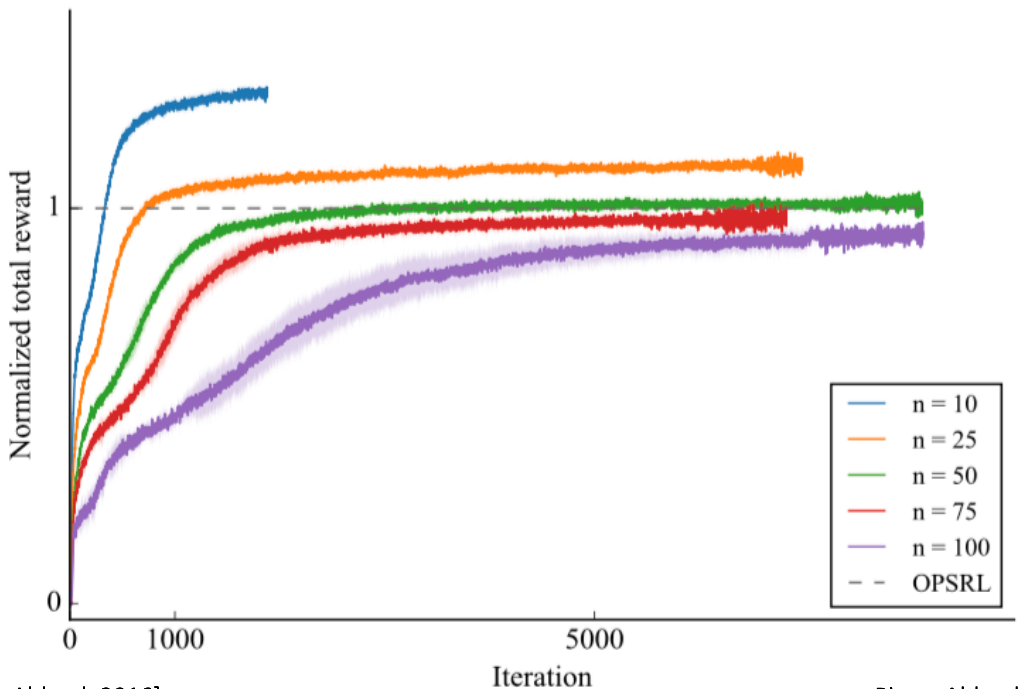
[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation: Tabular MDPs

Setup	Random	PSRL	OPSRL	UCRL2	BEB	ϵ-Greedy	Greedy	RL^2
$n = 10$	100.1	138.1	144.1	146.6	150.2	132.8	134.8	**156.2**
$n = 25$	250.2	408.8	425.2	424.1	427.8	377.3	368.8	**445.7**
$n = 50$	499.7	904.4	**930.7**	918.9	917.8	823.3	769.3	**936.1**
$n = 75$	749.9	1417.1	**1449.2**	1427.6	1422.6	1293.9	1172.9	1428.8
$n = 100$	999.4	1939.5	**1973.9**	1942.1	1935.1	1778.2	1578.5	1913.7

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation: Tabular MDPs

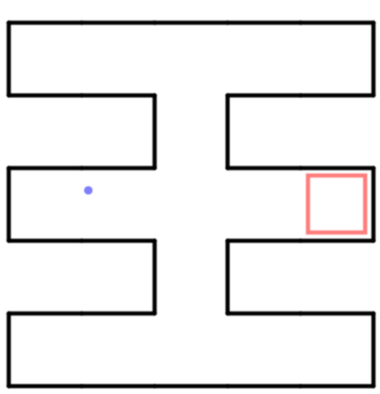


[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

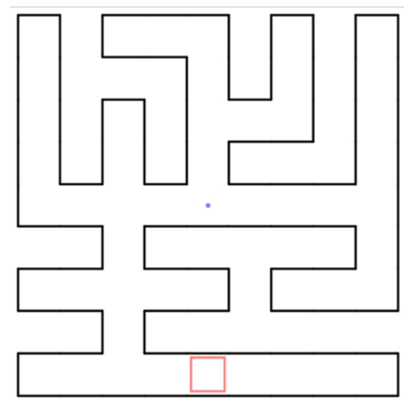
Evaluation: Visual Navigation

(built on top of ViZDoom)

Agent's view



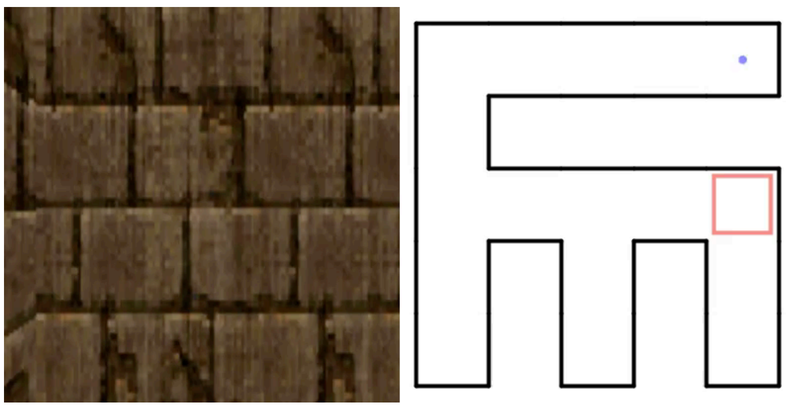
Small maze



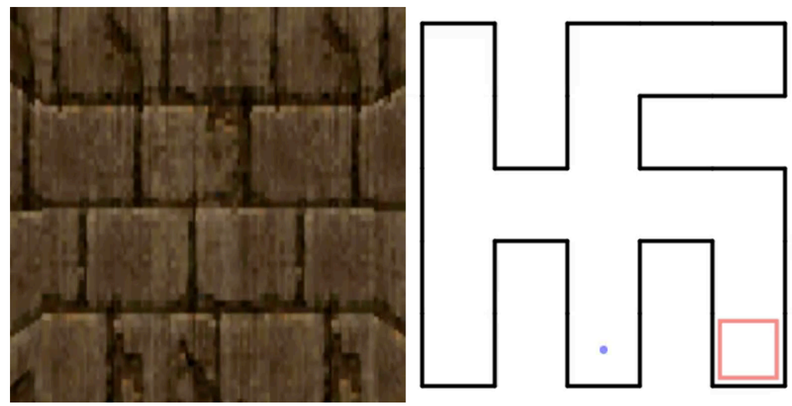
Large maze

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation: Visual Navigation



Before learning



After learning

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation

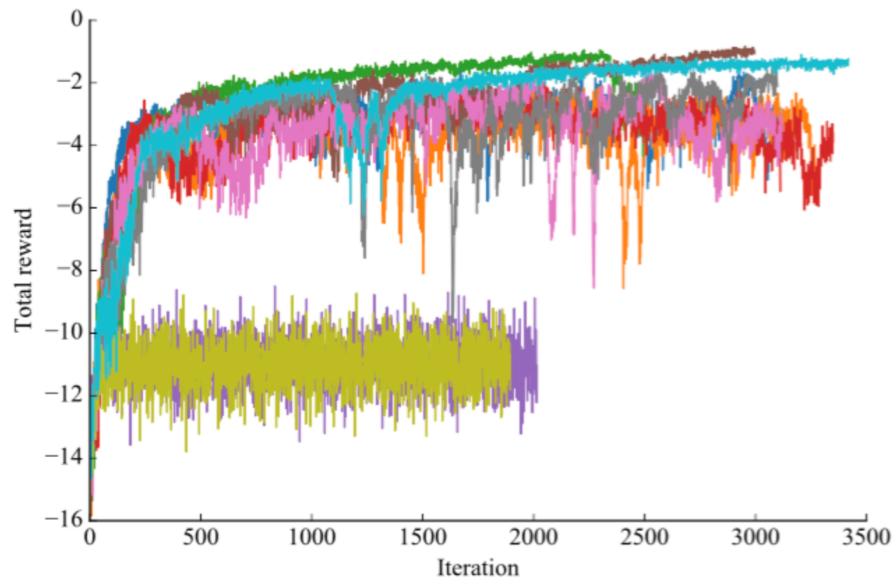


Figure 5: RL^2 learning curves for visual navigation. Each curve shows a different random initialization of the RNN weights. Performance varies greatly across different initializations.

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

OpenAI Universe

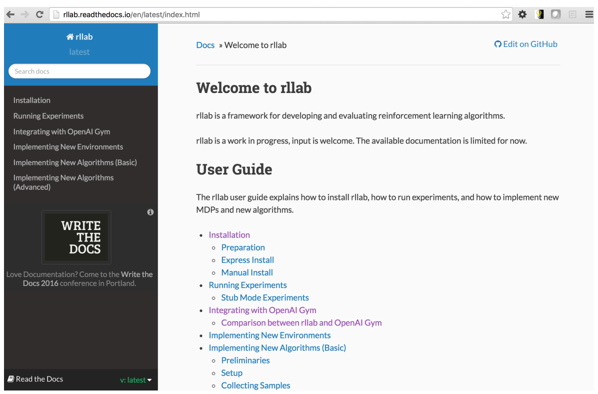
Pieter Abbeel – OpenAI / UC Berkeley / Gradescope

How to Learn More and Get Started?

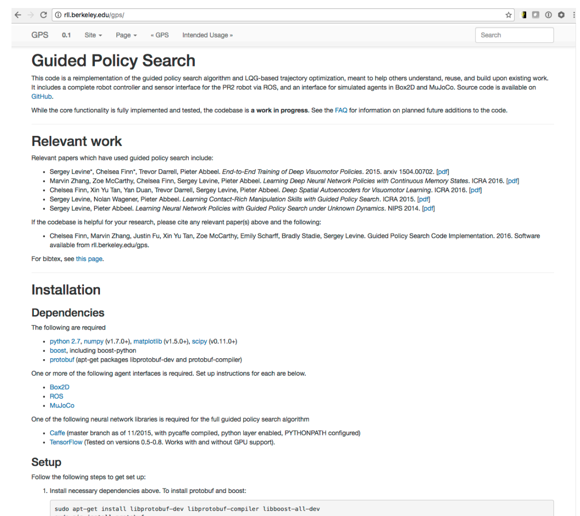
- **(1) Deep RL Courses**
  - CS294-112 Deep Reinforcement Learning (UC Berkeley): http://rll.berkeley.edu/deeprlcourse/ by Sergey Levine, John Schulman, Chelsea Finn
  - COMPM050/COMPGI13 Reinforcement Learning (UCL): http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching.html by David Silver

How to Learn More and Get Started?

- **(2) Deep RL Code Bases**
  - **rllab:** https://github.com/openai/rllab
    Duan, Chen, Houthooft, Schulman et al



  - Rlpy: https://rlpy.readthedocs.io/en/latest/
    Geramifard, Klein, Dann, Dabney, How
  - **GPS:** http://rll.berkeley.edu/gps/
    Finn, Zhang, Fu, Tan, McCarthy, Scharff, Stadie, Levine

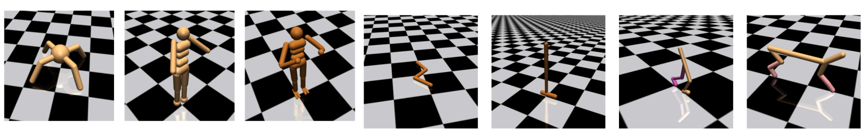


How to Learn More and Get Started?

- **(3) Environments**
  - **Arcade Learning Environment (ALE)** (Bellemare et al, JAIR 2013)

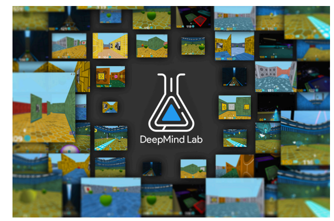
…

  - **MuJoCo:** http://mujoco.org (Todorov)

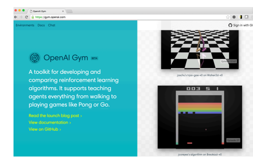


  - **Minecraft** (Microsoft)

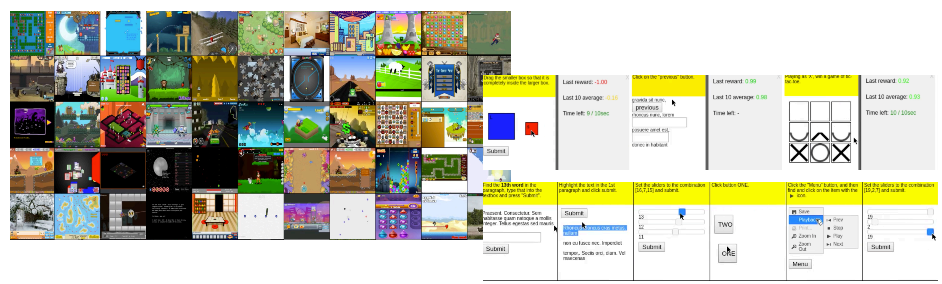
- **Deepmind Lab / Labyrinth** (Deepmind)



- **OpenAI Gym:** https://gym.openai.com/



- **Universe:** https://universe.openai.com/



Current / Future Directions

- Faster learning / Hierarchy
  - Exploration (Stadie, Levine, Abbeel 2015; Houthooft, Duan, Chen, Schulman Abbeel, 2016)
  - Meta-learning: RL2 (Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016); MAML (Finn, Abbeel, Levine, 2017)
- Transfer learning
  - Modular networks (Devin, Gupta, Darrell, Abbeel, Levine, 2017) ; Invariant feature spaces (Gupta Devin, Liu, Abbeel, Levine, 2017)
  - Domain randomization (Tobin, Fong, Schneider, Zaremba, Abbeel, 2017)
- Safe learning
  - Kahn, Villaflor, Pong, Abbeel, Levine, 2017; Held, McCarthy, Zhang, Shentu, Abbeel, 2016
- Unsupervised / Semisupervised learning
  - InfoGAN (Chen, Duan, Houthooft, Schulman, Sutskever, Abbeel 2016), VLAE (Chen, Kigma, Salimans, Duan, Dhariwal, Schulman, Sutskever, Abbeel, 2017)
  - Semisupervised RL (Finn, Yu, Fu, Abbeel, Levine, 2017)
- Grounded language / Multi-agent
  - "Inventing" language (Mordatch & Abbeel, 2017)
- Imitation
  - First-person from VR Tele-op (McCarthy, Zhang, Jow, Lee, Goldberg, Abbeel, 2017)
  - Third-person (Stadie, Abbeel, Sutskever, 2017)
- Value alignment / AI Safety
  - CIRL (Hadfield-Menell, Dragan, Abbeel, Russell, 2016), Off-switch (Hadfield Menell, Dragan, Abbeel, Russell, 2017)
  - Communication (Huang, Held, Abbeel, Dragan, 2017)