Map

Model-Based

Model-Free

MLMBTRL

Q-value

π

On-Policy

SARSA

Policy Gradient

Actor Critic

Q-Learning

Off-Policy

1. Additional Actor-Critic
2. Advanced Exploration
3. Entropy Regularization
4. Wisdom

Alpha Zero: Actor Critic with MCTS

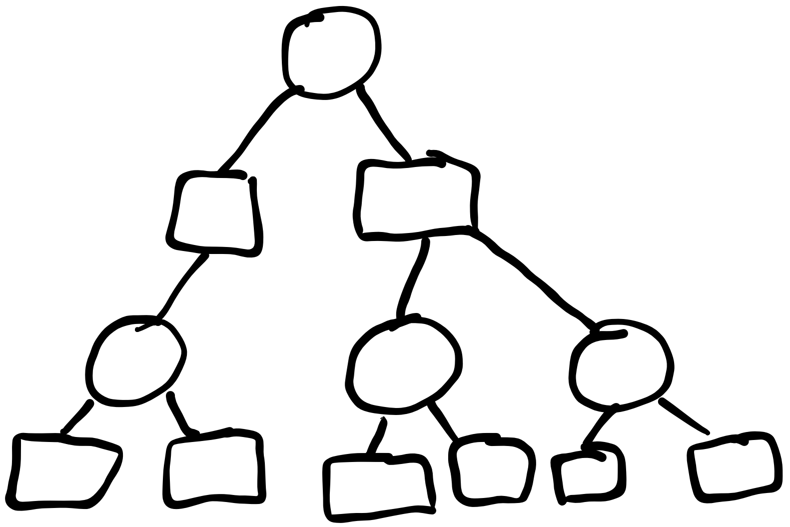
Alpha Zero: Actor Critic with MCTS

help choose actions
instead of rollout

1. Use π_θ and U_ϕ in MCTS

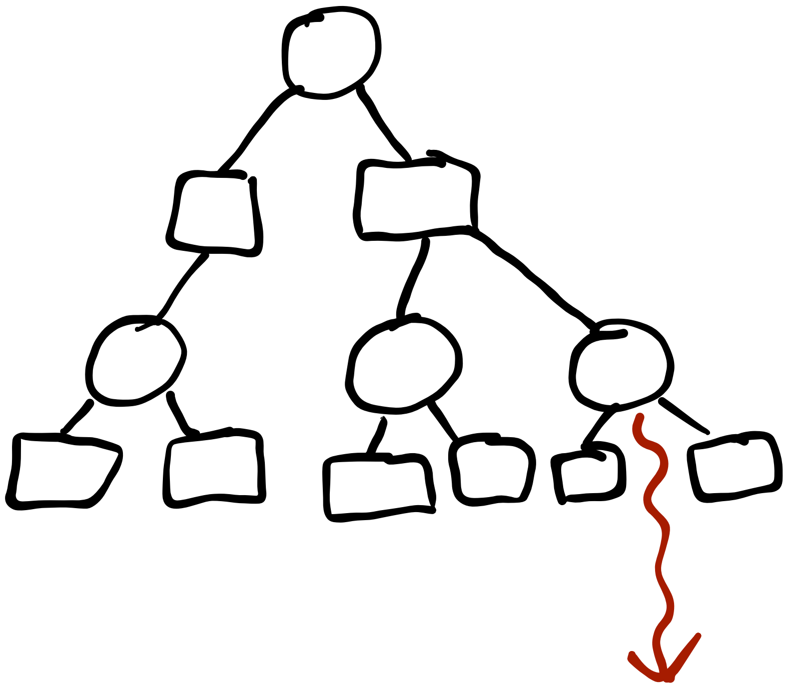
Alpha Zero: Actor Critic with MCTS

1. Use π_θ and U_ϕ in MCTS



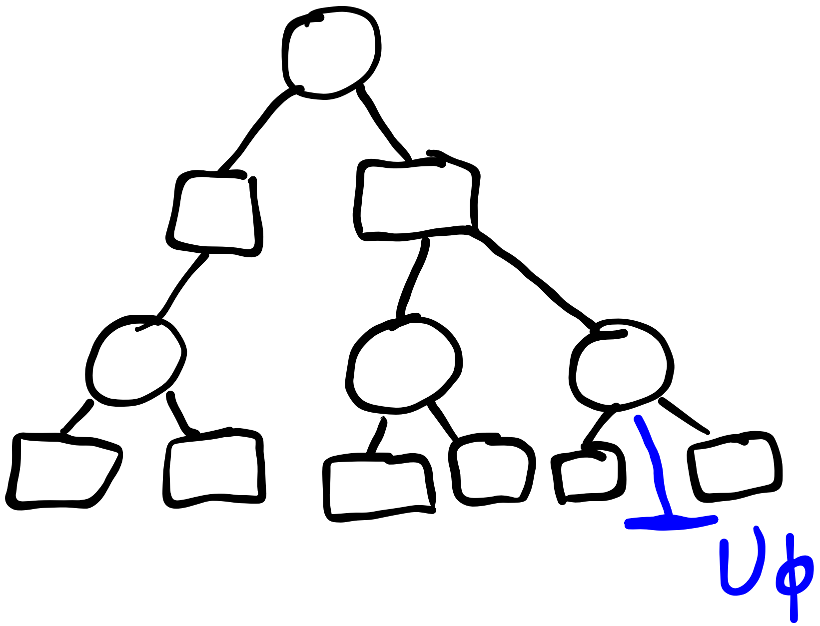
Alpha Zero: Actor Critic with MCTS

1. Use π_θ and U_ϕ in MCTS



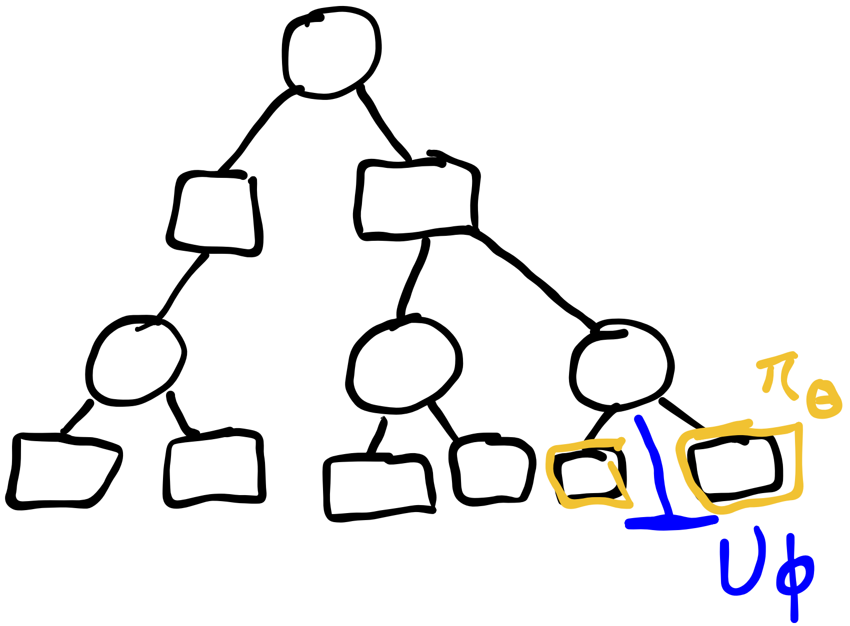
Alpha Zero: Actor Critic with MCTS

1. Use π_θ and U_ϕ in MCTS



Alpha Zero: Actor Critic with MCTS

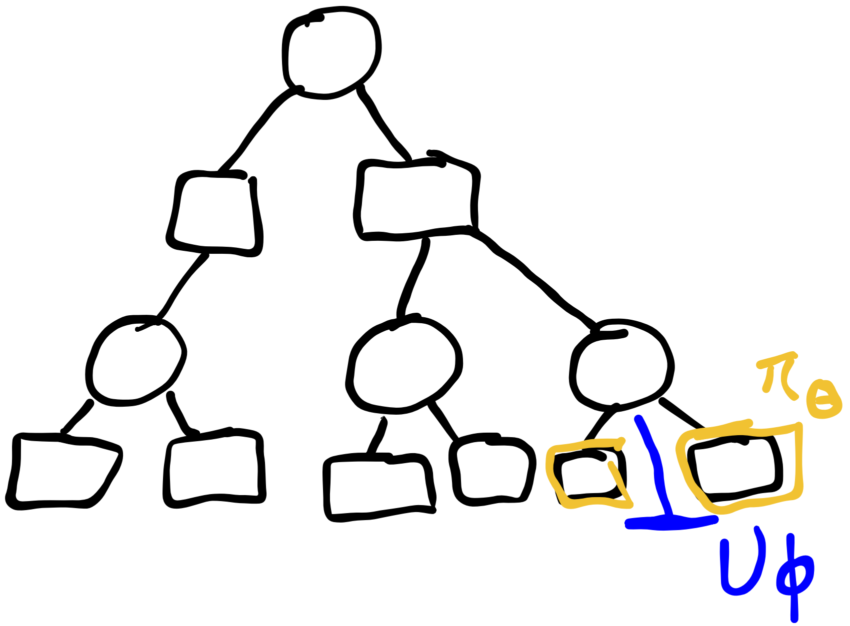
1. Use π_θ and U_ϕ in MCTS



$$a = \arg\max_a Q(s,a) + c\pi_\theta(a \mid s)\frac{\sqrt{N(s)}}{1 + N(s,a)}$$

Alpha Zero: Actor Critic with MCTS

1. Use π_θ and U_ϕ in MCTS
2. Learn π_θ and U_ϕ from tree



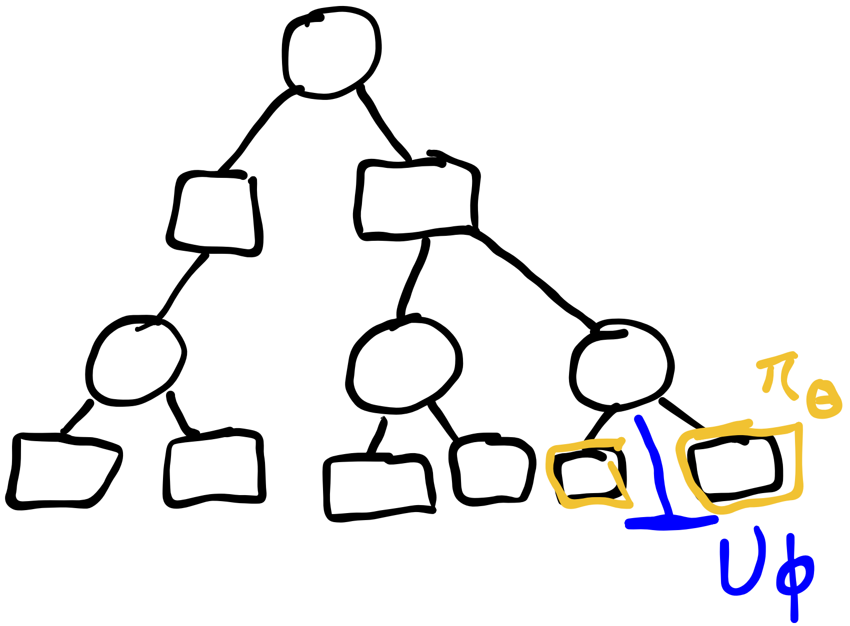
$$a = \arg\max_a Q(s,a) + c\pi_\theta(a \mid s)\frac{\sqrt{N(s)}}{1+N(s,a)}$$

Alpha Zero: Actor Critic with MCTS

1. Use π_θ and U_ϕ in MCTS
2. Learn π_θ and U_ϕ from tree

$$\ell(\theta) = -\mathbb{E}_s\left[\sum_a \pi_{\mathrm{MCTS}}(a \mid s) \log \pi_\theta(a \mid s)\right]$$

$$\pi_{\mathrm{MCTS}}(a \mid s) \propto \underline{N(s,a)}^\eta$$



$$a = \arg\max_a Q(s,a) + c\pi_\theta(a \mid s)\frac{\sqrt{N(s)}}{1+N(s,a)}$$

Alpha Zero: Actor Critic with MCTS

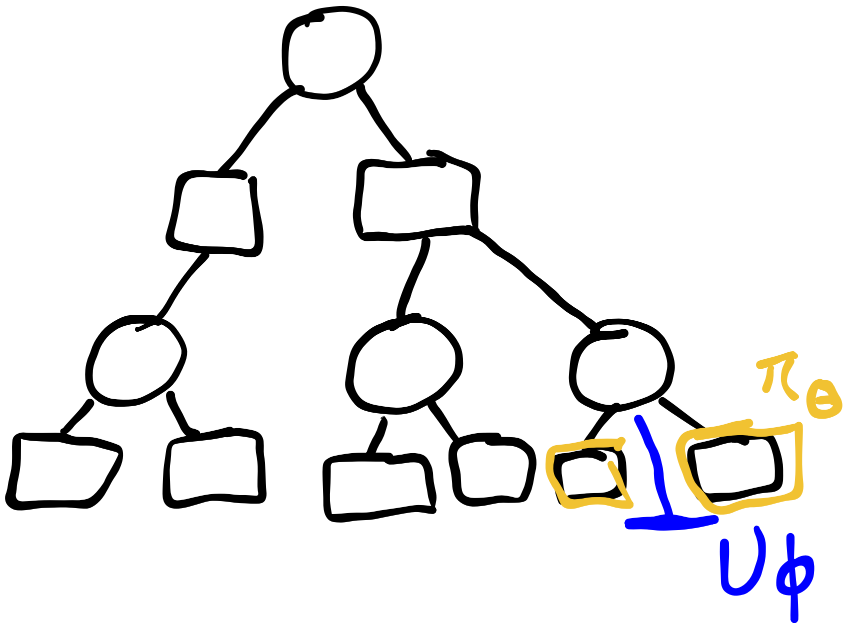
1. Use π_θ and U_ϕ in MCTS
2. Learn π_θ and U_ϕ from tree

$$\ell(\theta) = -\mathbb{E}_s\left[\sum_a \pi_{\text{MCTS}}(a \mid s) \log \pi_\theta(a \mid s)\right]$$

$$\pi_{\text{MCTS}}(a \mid s) \propto N(s,a)^\eta$$

$$\ell(\phi) = \frac{1}{2}\mathbb{E}_s\left[\left(U_\phi(s) - \underline{U_{\text{MCTS}}(s)}\right)^2\right]$$

$$U_{\text{MCTS}}(s) = \underline{\max_a Q(s,a)}$$



$$a = \arg\max_a Q(s,a) + c\pi_\theta(a \mid s)\frac{\sqrt{N(s)}}{1+N(s,a)}$$

Continuous Actions: Deep Deterministic Policy Gradient

Actor Critic

$$Q_\phi(s,a)$$

$$\pi_\theta(s) = \arg\max_a Q_\phi(s,a)$$

$$\ell(\phi) = \mathop{E}_{(s,a,r,s')\sim D}\left[\left(r + \gamma Q_{\bar\phi}(s', \pi_{\bar\theta}(s')) - Q_\phi(s,a)\right)^2\right]$$

$$U(\theta) = \mathop{E}_{s\sim D}\left[Q_\phi(s, \pi_\theta(s))\right]$$

$$\nabla U(\theta) = \mathop{E}_{s\sim D}\left[\nabla_\theta Q_\phi(s, \pi_\theta(s))\right]$$

Unstable

Is Exploration Important?
Montezuma's Revenge

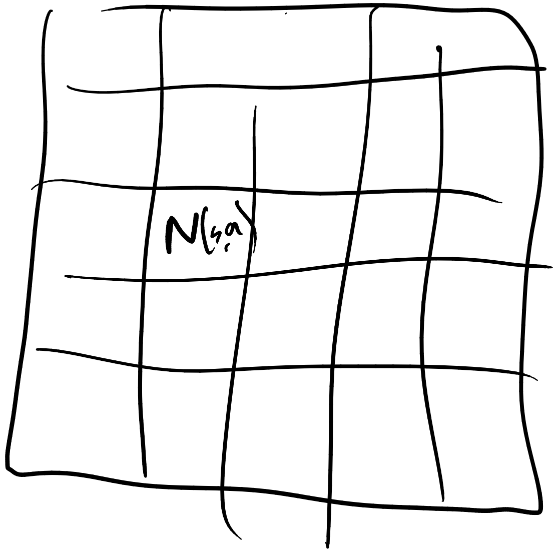
Is Exploration Important?
Theory

	Algorithm	Regret	Time	Space
Model-based	UCRL2 [10] [1]	at least $\tilde{\mathcal{O}}(\sqrt{H^4 S^2 AT})$	$\Omega(TS^2A)$	$\mathcal{O}(S^2AH)$
	Agrawal and Jia [1] [1]	at least $\tilde{\mathcal{O}}(\sqrt{H^3 S^2 AT})$		
	UCBVI [5] [2]	$\tilde{\mathcal{O}}(\sqrt{H^2 SAT})$	$\tilde{\mathcal{O}}(TS^2A)$	
	vUCQ [12] [2]	$\tilde{\mathcal{O}}(\sqrt{H^2 SAT})$ Horizon		
Model-free	Q-learning (ε-greedy) [14] (if 0 initialized)	$\Omega(\min\{T, A^{H/2}\})$	$\mathcal{O}(T)$	$\mathcal{O}(SAH)$
	Delayed Q-learning [25] [3]	$\tilde{\mathcal{O}}_{S,A,H}(T^{4/5})$		
	Q-learning (UCB-H)	$\tilde{\mathcal{O}}(\sqrt{H^4 SAT})$		
	Q-learning (UCB-B)	$\tilde{\mathcal{O}}(\sqrt{H^3 SAT})$		
	lower bound	$\Omega(\sqrt{H^2 SAT})$	-	-

Exploration Bonus

Exploration Bonus

- In General, $R^{+}(s,a) = R(s,a) + B(s,a)$
- UCB: $B(s,a) = c\sqrt{\frac{\log N(s)}{N(s,a)}}$



Exploration Bonus

Example 1: Learn Pseudocount

Bellemare, et al. 2016 "Unifying Count-Based Exploration..."

Exploration Bonus

Example 1: Learn Pseudocount

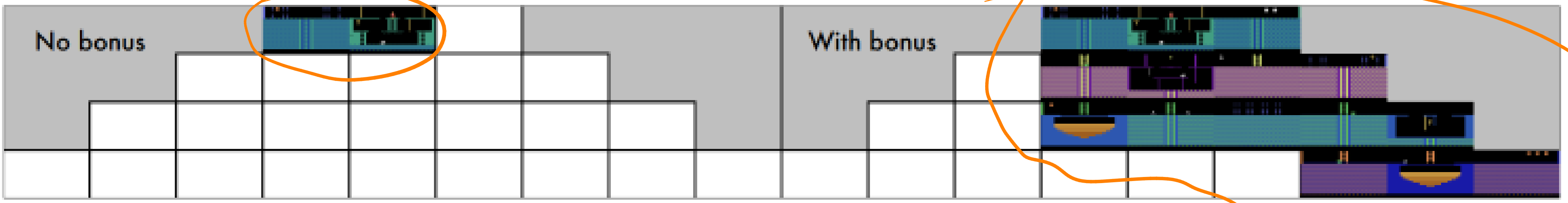
$B(s,a) \approx \frac{1}{\sqrt{\hat{N}(s)}}$ where $\hat{N}(s)$ is a learned function approximation

Bellemare, et al. 2016 "Unifying Count-Based Exploration..."

Exploration Bonus

Example 1: Learn Pseudocount

$B(s,a) \approx \frac{1}{\sqrt{\hat{N}(s)}}$ where $\hat{N}(s)$ is a learned function approximation

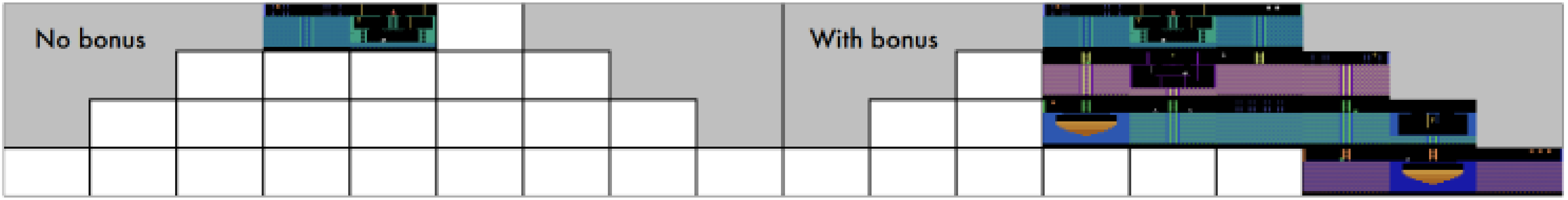


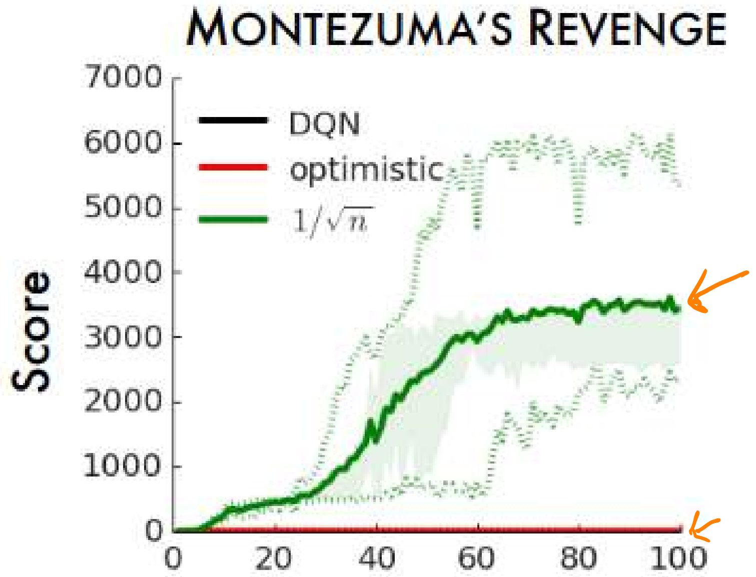
Bellemare, et al. 2016 "Unifying Count-Based Exploration..."

Exploration Bonus

Example 1: Learn Pseudocount

$B(s,a) \approx \frac{1}{\sqrt{\hat{N}(s)}}$ where $\hat{N}(s)$ is a learned function approximation



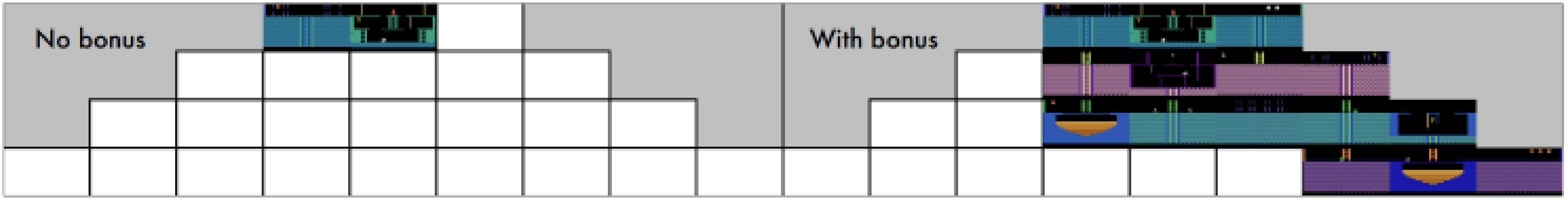


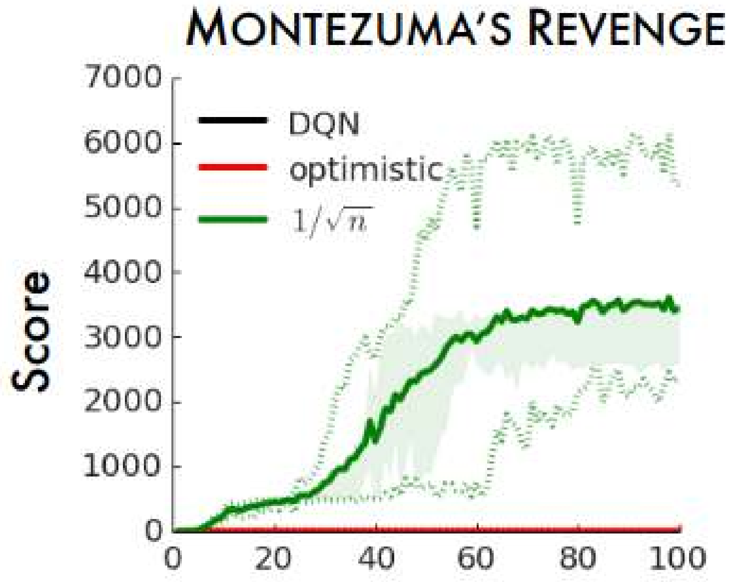
Bellemare, et al. 2016 "Unifying Count-Based Exploration..."

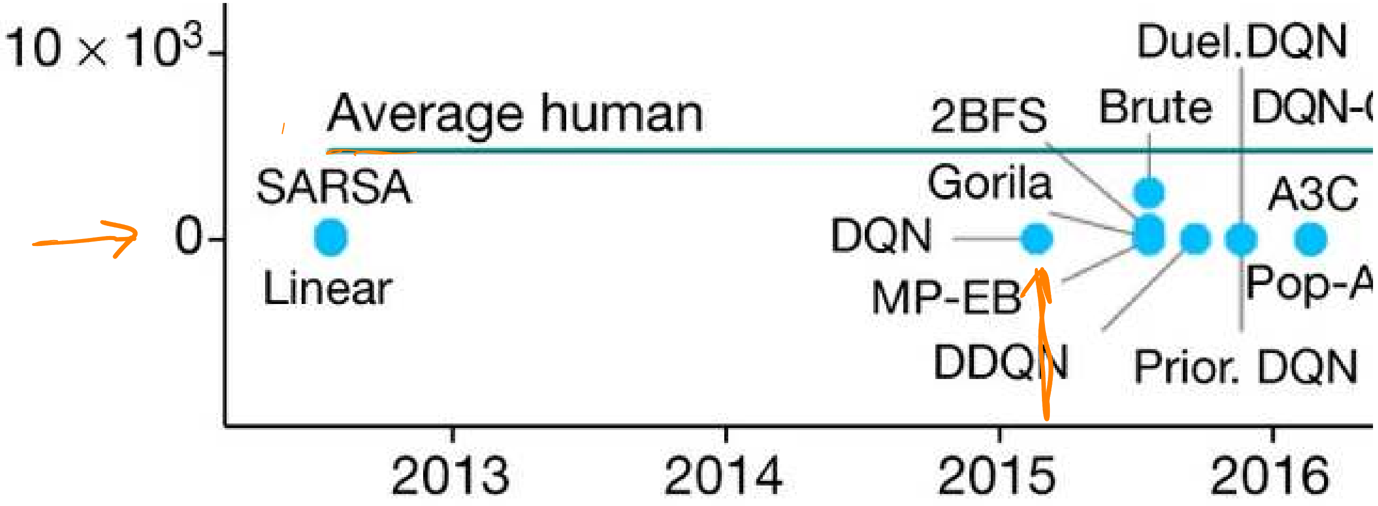
Exploration Bonus

Example 1: Learn Pseudocount

$B(s,a) \approx \frac{1}{\sqrt{\hat{N}(s)}}$ where $\hat{N}(s)$ is a learned function approximation





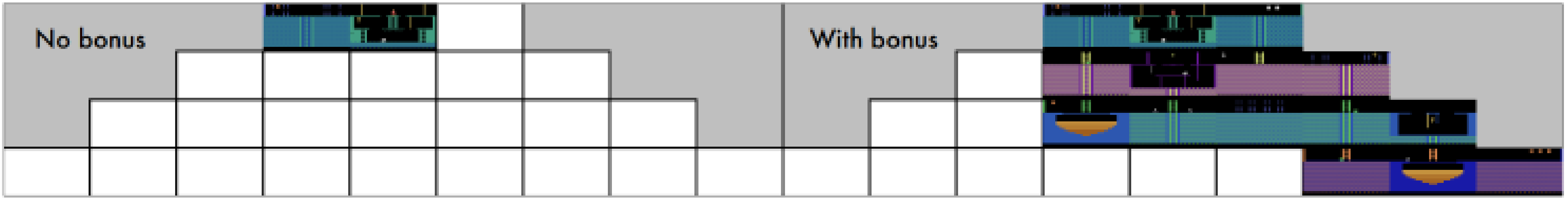


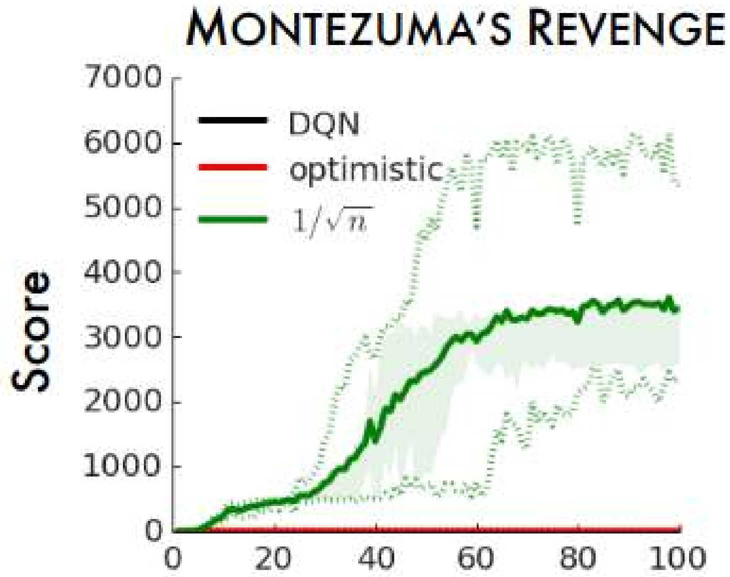
Bellemare, et al. 2016 "Unifying Count-Based Exploration..."

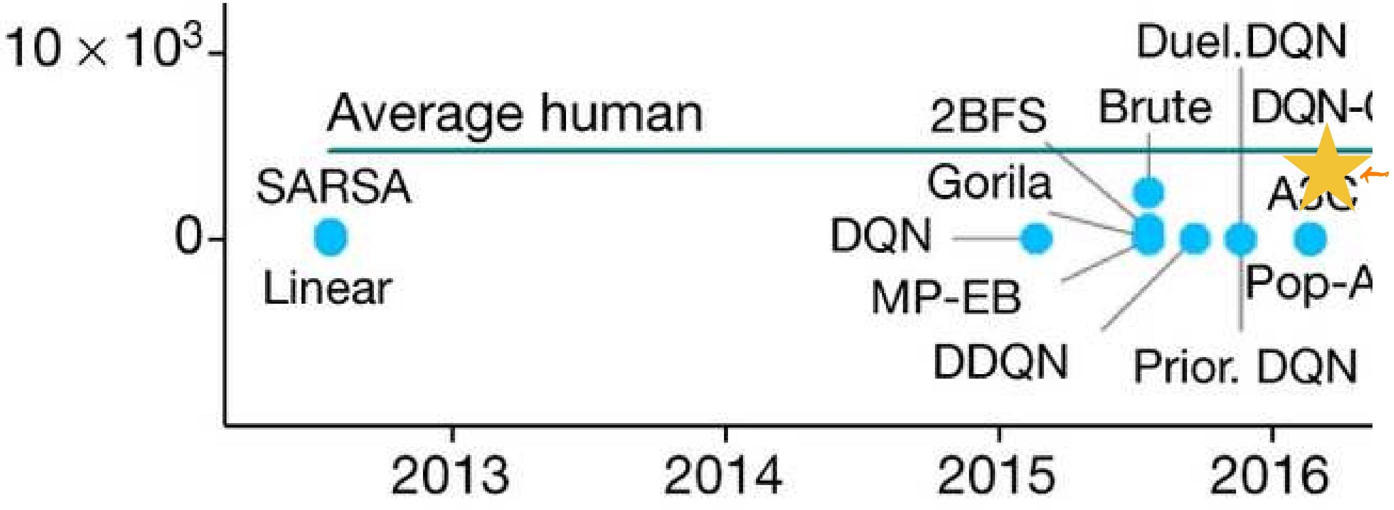
Exploration Bonus

Example 1: Learn Pseudocount

$B(s,a) \approx \frac{1}{\sqrt{\hat{N}(s)}}$ where $\hat{N}(s)$ is a learned function approximation







Bellemare, et al. 2016 "Unifying Count-Based Exploration..."

Exploration Bonus

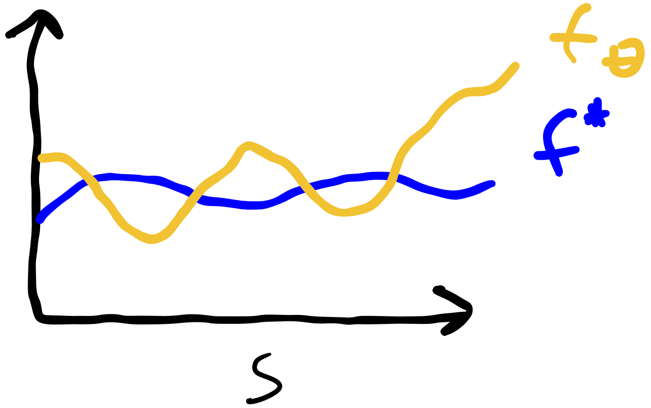
Exploration Bonus

Example 2: Learn a function of the state and action

Exploration Bonus

Example 2: Learn a function of the state and action

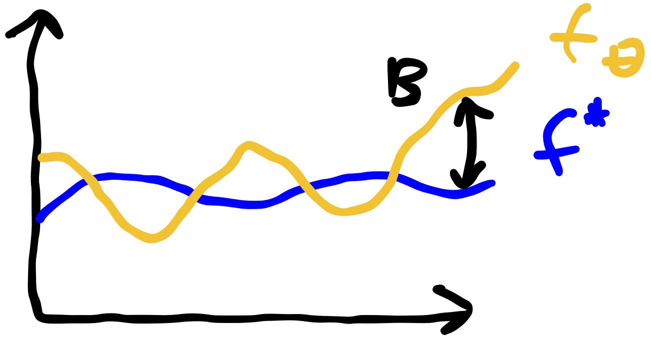
$$B(s,a) = \|\hat{f}_\theta(s,a) - f^*(s,a)\|^2$$



Exploration Bonus

Example 2: Learn a function of the state and action

$$B(s,a) = \|\hat{f}_\theta(s,a) - f^*(s,a)\|^2$$

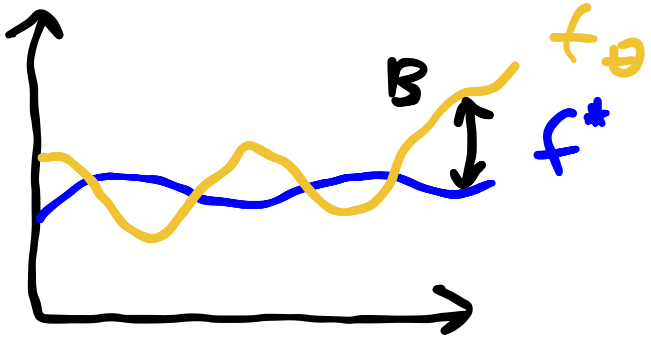


Exploration Bonus

Example 2: Learn a function of the state and action

$$B(s,a) = \|\hat{f}_\theta(s,a) - f^*(s,a)\|^2$$

What should f^* be?

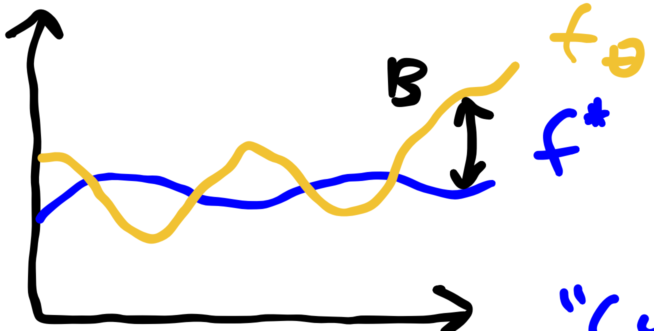


Exploration Bonus

Example 2: Learn a function of the state and action

$$B(s,a) = \|\hat{f}_\theta(s,a) - f^*(s,a)\|^2$$

What should f^* be?



"Curiosity"

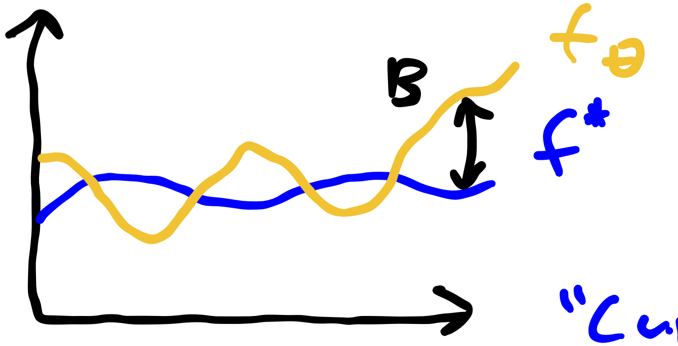
Exploration Bonus

Example 2: Learn a function of the state and action

$$B(s,a) = \|\hat{f}_\theta(s,a) - f^*(s,a)\|^2$$

What should f^* be?

- $f^*(s,a) = s'$ (Next state prediction)

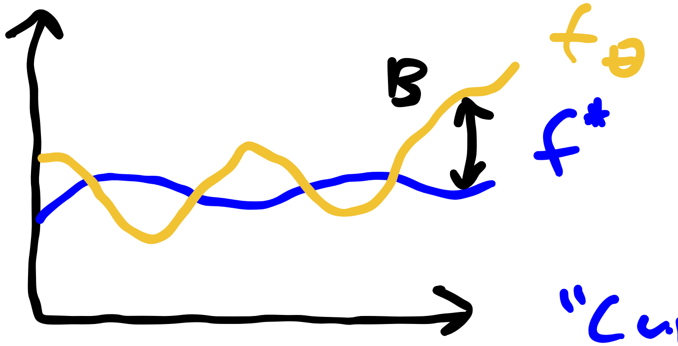


"Curiosity"

Exploration Bonus

Example 2: Learn a function of the state and action

$$B(s,a) = \|\hat{f}_\theta(s,a) - f^*(s,a)\|^2$$



What should f^* be?

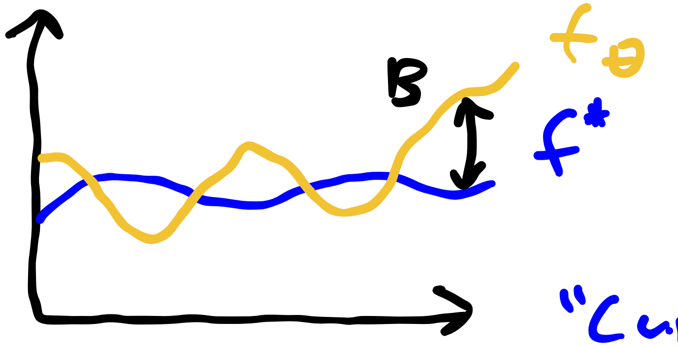
- $f^*(s,a) = s'$ (Next state prediction) "Curiosity"
- $f^*(s,a) = f_\phi(s,a)$ where f_ϕ is a random neural network.

"Random Network Distillation"

Exploration Bonus

Example 2: Learn a function of the state and action

$$B(s,a) = \|\hat{f}_\theta(s,a) - f^*(s,a)\|^2$$

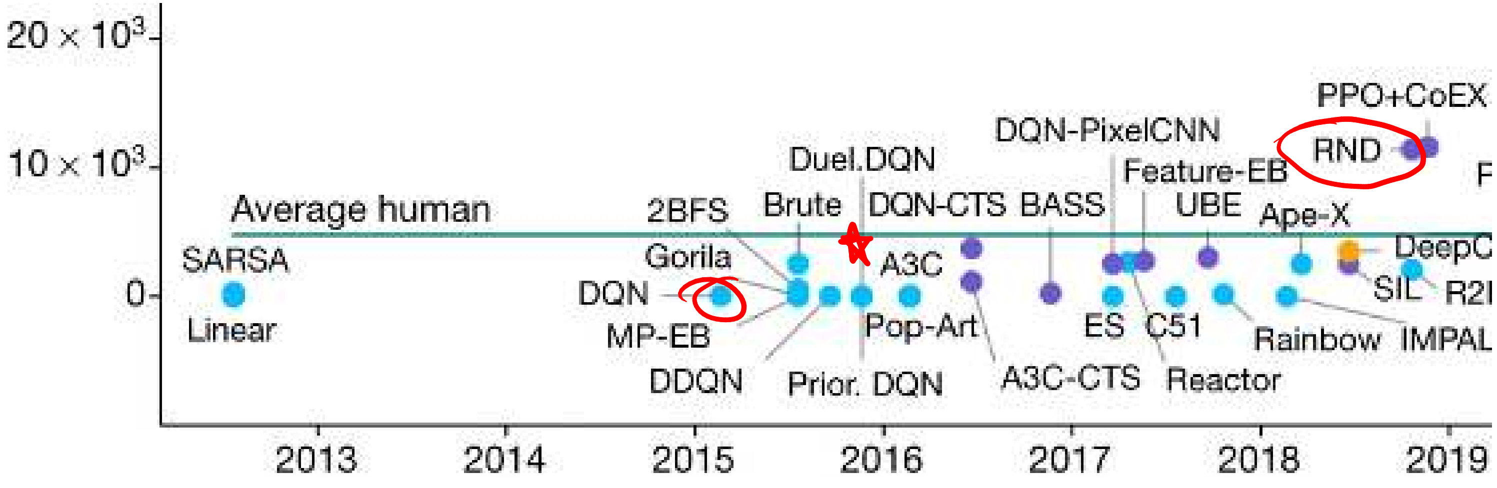


What should f^* be?

- $f^*(s,a) = s'$ (Next state prediction)
- $f^*(s,a) = f_\phi(s,a)$ where f_ϕ is a random neural network.

"Curiosity"

"Random Network Distillation"



Exploration Bonus

Exploration Bonus

Example 3: Thompson Sampling

Exploration Bonus

Example 3: Thompson Sampling

1. Maintain a distribution over Q

Exploration Bonus

Example 3: Thompson Sampling

1. Maintain a distribution over Q
2. Sample Q

Exploration Bonus

Example 3: Thompson Sampling

1. Maintain a distribution over Q
2. Sample $\hat{Q}$ from ~~Q~~ dist.
3. Act according to $\hat{Q}$

Exploration Bonus

Example 3: Thompson Sampling

1. Maintain a distribution over Q ← Hard
2. Sample Q
3. Act according to Q

Exploration Bonus

Example 3: Thompson Sampling

1. Maintain a distribution over Q ← Hard
2. Sample Q
3. Act according to Q

- Bootstrapping with multiple Q networks

Exploration Bonus

Example 3: Thompson Sampling

1. Maintain a distribution over Q ← Hard
2. Sample Q
3. Act according to Q

- Bootstrapping with multiple Q networks
- Dropout

Exploration Bonus

Exploration Bonus

Example 4: Go-Explore

Exploration Bonus

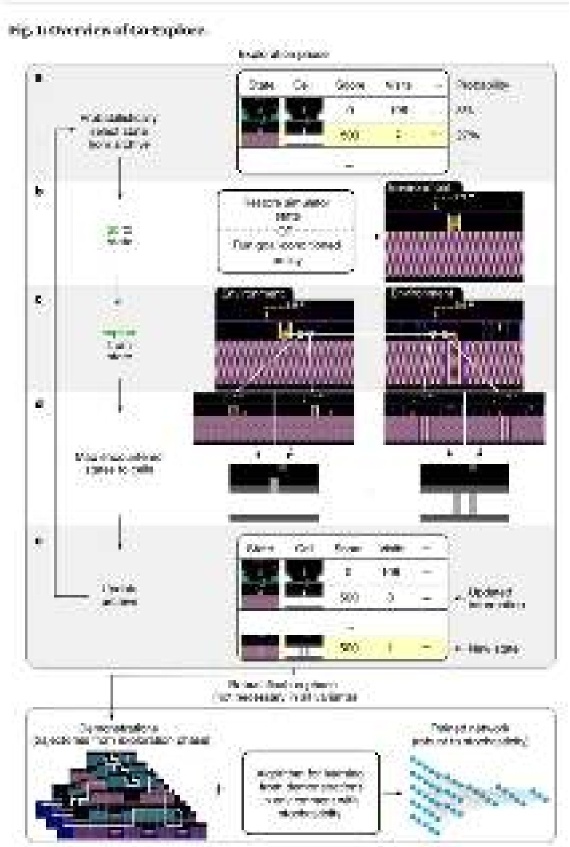
Example 4: Go-Explore

"First return, then explore"

Exploration Bonus

Example 4: Go-Explore

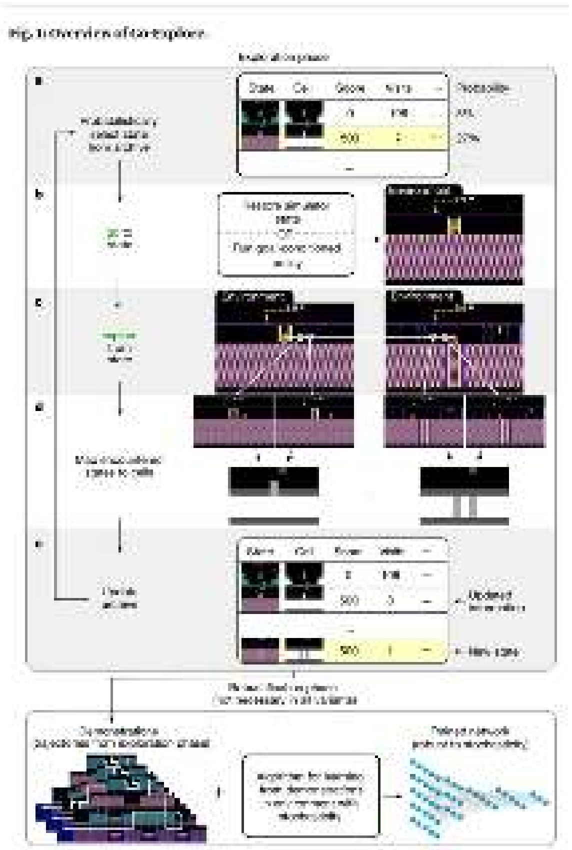
"First return, then explore"

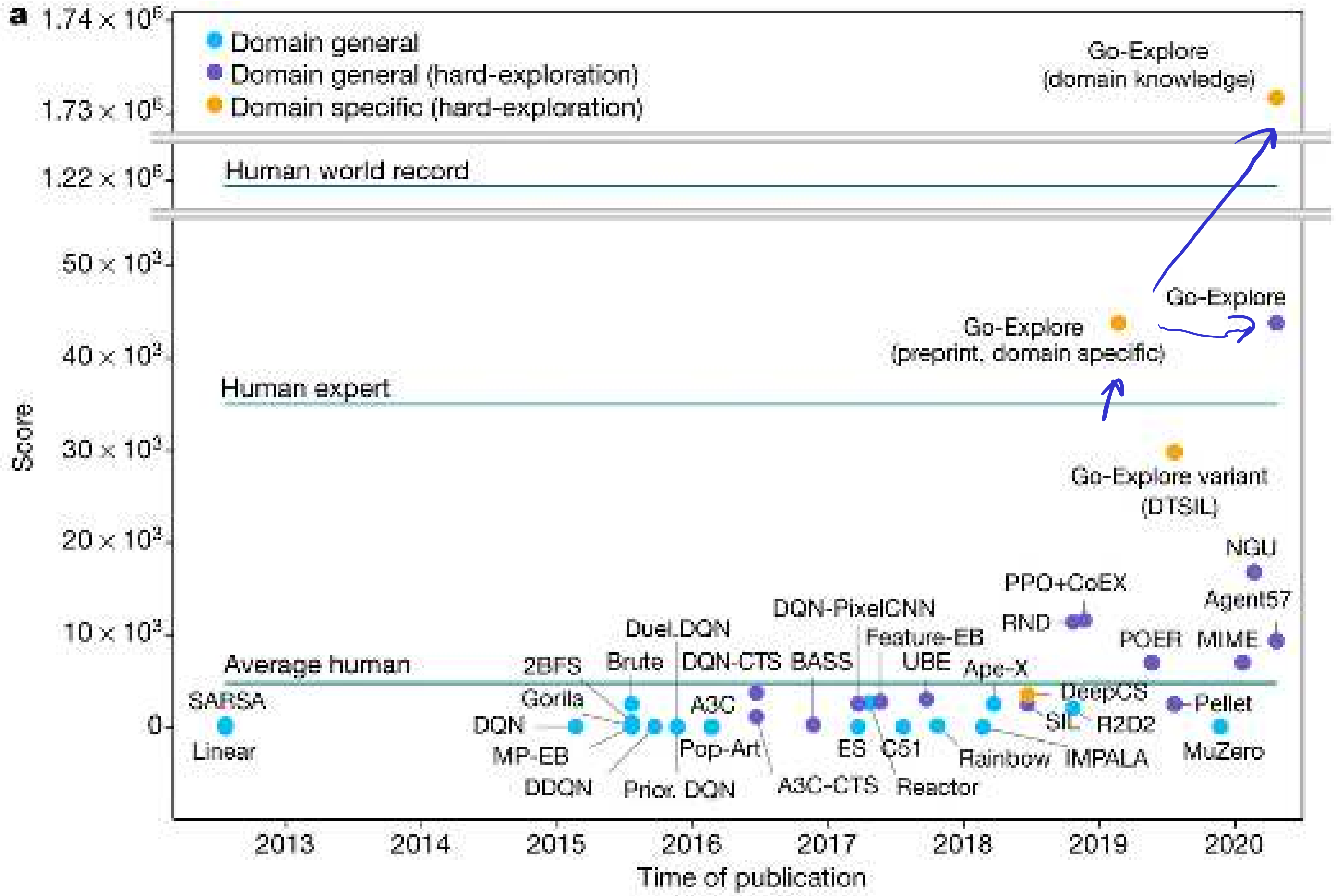


Exploration Bonus

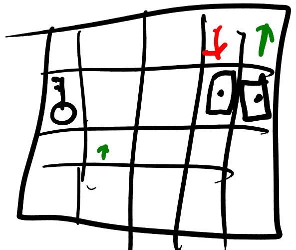
Example 4: Go-Explore

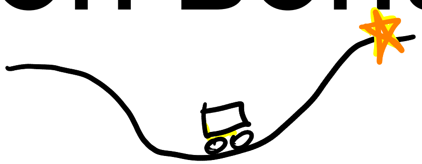
"First return, then explore"

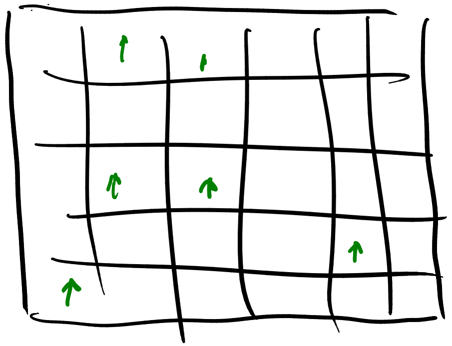




Exploration Bonus

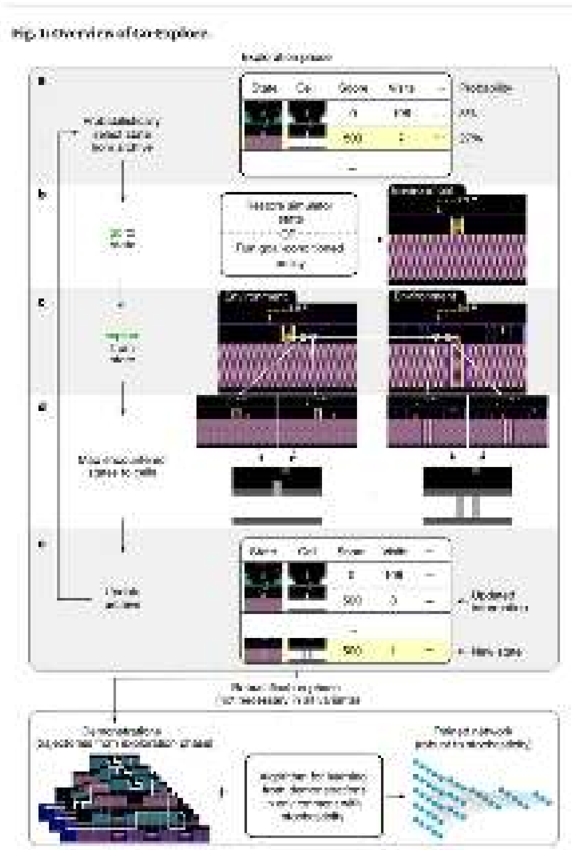


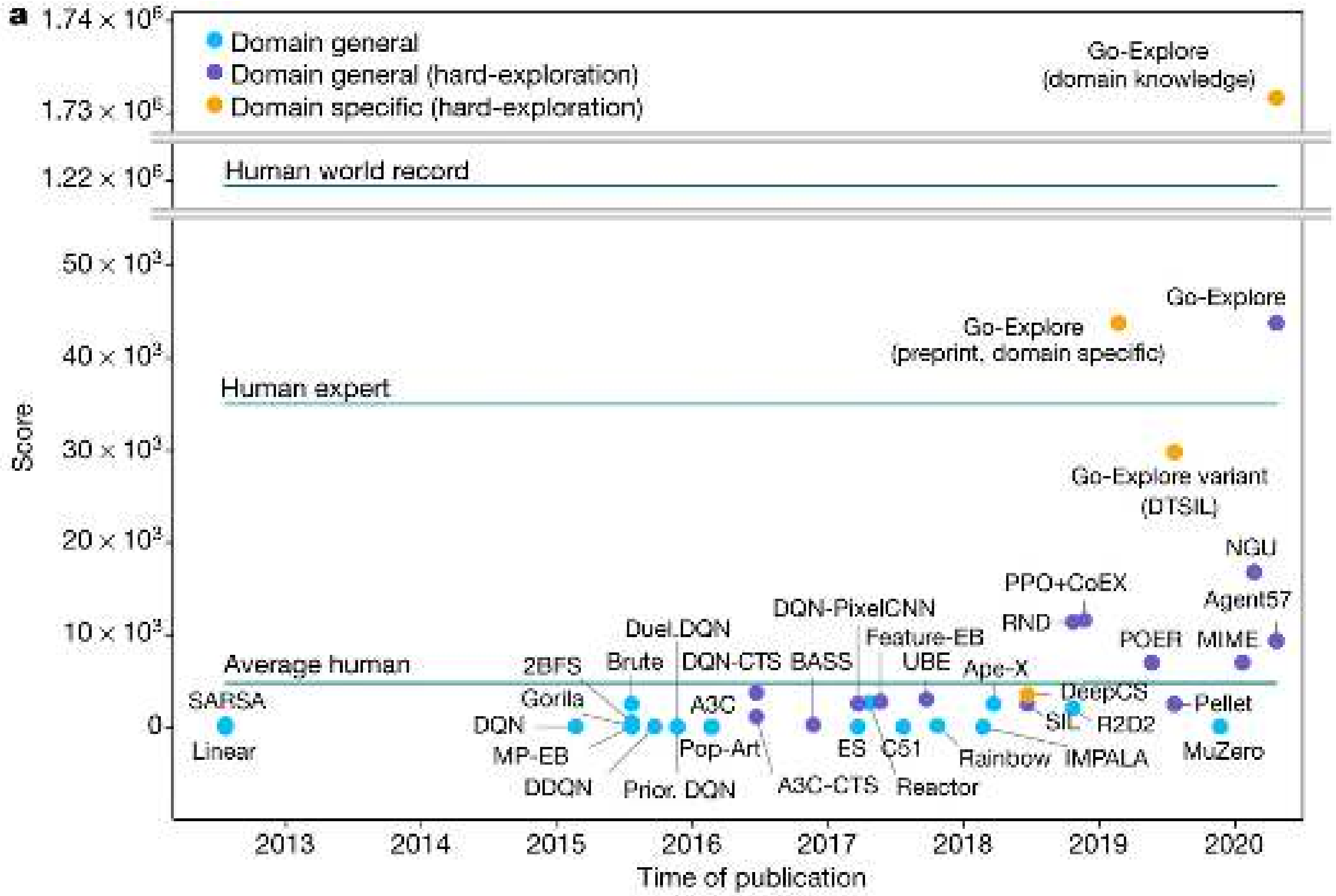




Example 4: Go-Explore

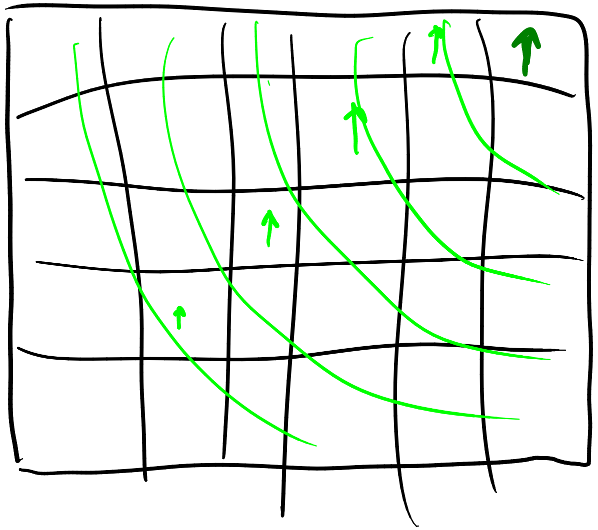
"First return, then explore"





(Uber AI Labs)

~~Soft Actor Critic: Entropy Regularization~~



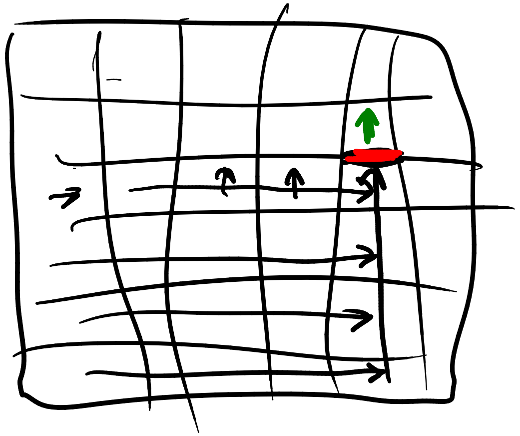
Intrinsic

Soft Actor Critic: Entropy Regularization

$U(\pi) = E\left[\sum \gamma^t r_t\right]$

entropy

$$U(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha \mathcal{H}(\pi(\cdot \mid s_t))\right)\right]$$



Soft Actor Critic: Entropy Regularization

$$U(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha \mathcal{H}(\pi(\cdot \mid s_t))\right)\right]$$

$$V(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi}\left[Q(\mathbf{s}_t, \mathbf{a}_t) - \log \pi(\mathbf{a}_t | \mathbf{s}_t)\right]$$

Soft Actor Critic: Entropy Regularization

$$U(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha \mathcal{H}(\pi(\cdot \mid s_t))\right)\right]$$

$$V(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi}\left[Q(\mathbf{s}_t, \mathbf{a}_t) - \log \pi(\mathbf{a}_t|\mathbf{s}_t)\right]$$

iterative policy evaluation

$$\mathcal{T}^\pi Q(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[V(\mathbf{s}_{t+1})\right]$$

Soft Actor Critic: Entropy Regularization

$$U(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha \mathcal{H}(\pi(\cdot \mid s_t))\right)\right]$$

soft policy iteration

soft policy eval

$$V(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi}\left[Q(\mathbf{s}_t, \mathbf{a}_t) - \log \pi(\mathbf{a}_t|\mathbf{s}_t)\right]$$

$$\mathcal{T}^\pi Q(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[V(\mathbf{s}_{t+1})\right]$$

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathrm{D}_{\mathrm{KL}}\left(\pi'(\,\cdot\,|\mathbf{s}_t) \,\middle\|\, \frac{\exp\left(Q^{\pi_{\text{old}}}(\mathbf{s}_t, \,\cdot\,)\right)}{Z^{\pi_{\text{old}}}(\mathbf{s}_t)}\right)$$

Soft Actor Critic

Soft Actor Critic

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$
  end for
end for

Soft Actor Critic

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  end for
end for

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\frac{1}{2}\left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\right]\right)^2\right]$$

Soft Actor Critic

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  end for
end for

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}}\left[\tfrac{1}{2}\left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi}\left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\right]\right)^2\right]$$

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t)\right)^2\right]$$

Soft Actor Critic

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  end for
end for

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\frac{1}{2}\left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t\sim\pi_\phi}\left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\right]\right)^2\right]$$

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t)\right)^2\right]$$

$$\hat{Q}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\,\mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[V_{\bar{\psi}}(\mathbf{s}_{t+1})\right]$$

Soft Actor Critic

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.

V_ψ Q_θ π_ϕ

for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$
  end for
end for

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\frac{1}{2}\left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t\sim\pi_\phi}\left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\right]\right)^2\right]$$

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t)\right)^2\right]$$

$$\hat{Q}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[V_{\bar{\psi}}(\mathbf{s}_{t+1})\right]$$

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\mathrm{D}_{\mathrm{KL}}\left(\pi_\phi(\,\cdot\,|\mathbf{s}_t)\,\middle\|\,\frac{\exp\left(Q_\theta(\mathbf{s}_t, \cdot)\right)}{Z_\theta(\mathbf{s}_t)}\right)\right]$$

Soft Actor Critic

Advantages:

Soft Actor Critic

Advantages:

- Stable

Soft Actor Critic

Advantages:

- Stable
- Learns many near-optimal policies

Soft Actor Critic

Advantages:

- Stable
- Learns many near-optimal policies
- Exploration

Soft Actor Critic

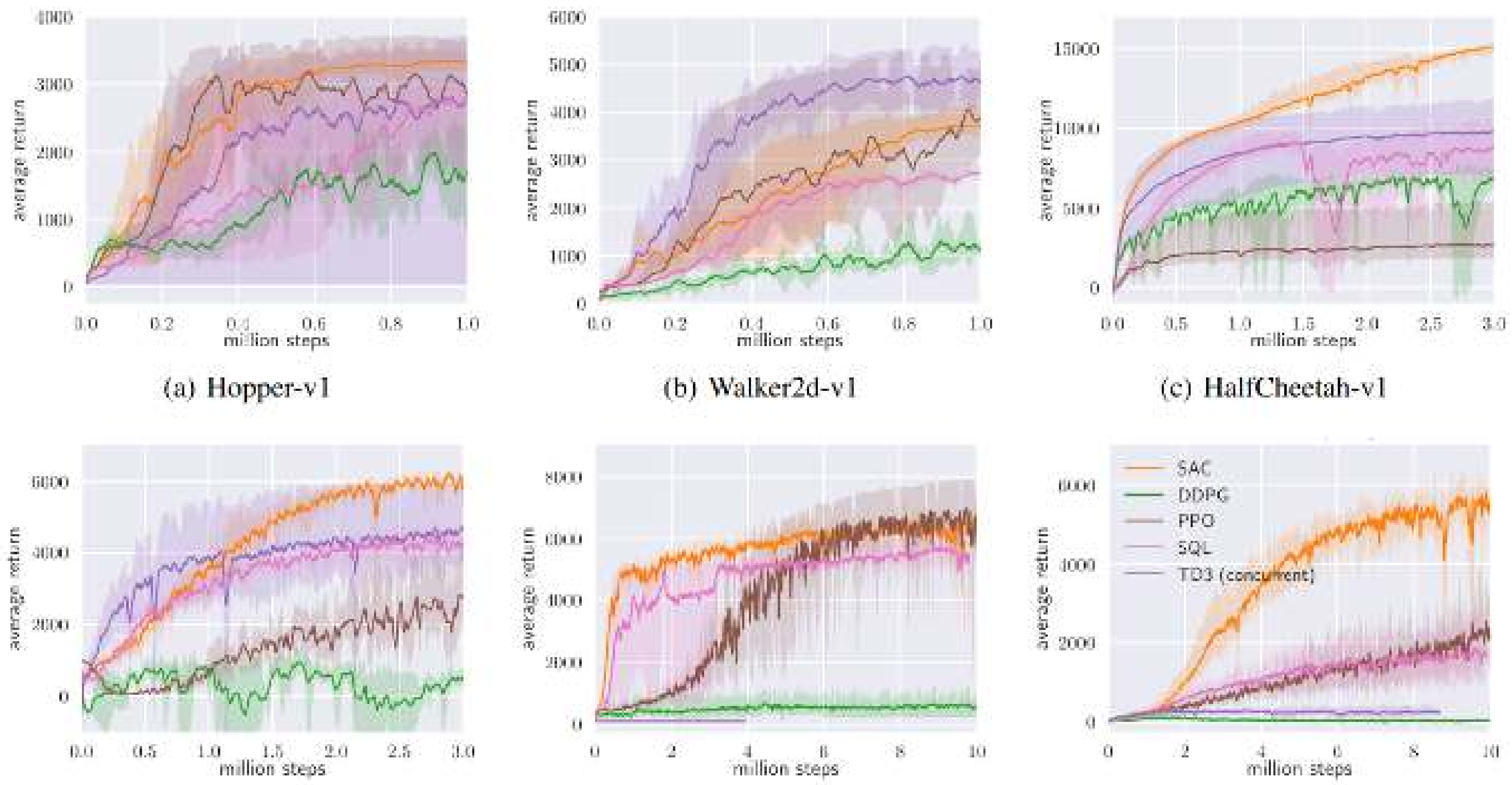
Advantages:

- Stable
- Learns many near-optimal policies
- Exploration
- Insensitivity to hyperparameters
- Off-Policy

Soft Actor Critic

Advantages:

- Stable
- Learns
- Explora
- Insensit
- Off-Poli



(a) Hopper-v1 (b) Walker2d-v1 (c) HalfCheetah-v1

(d) Ant-v1 (e) Humanoid-v1 (f) Humanoid (rllab)

Soft Actor Critic

Advantages:

- Stable
- Learns many near-optimal policies
- Exploration
- Insensitivity to hyperparameters
- Off-Policy

Disadvantages

- Sensitive to α Solution = Entropy *constraint* and adjust α

Soft Actor Critic

Advantages:

Disadvantages

- Sta
- Le
- Ex
- Ins
- Of

Algorithm 1 Soft Actor-Critic

Input: θ_1, θ_2, ϕ ▷ Initial parameters

  $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$ ▷ Initialize target network weights

  $\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay pool

  for each iteration **do**

    for each environment step **do**

      $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$ ▷ Sample action from the policy

      $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ ▷ Sample transition from the environment

      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$ ▷ Store the transition in the replay pool

    end for

    for each gradient step **do**

      $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$ ▷ Update the Q-function parameters

      $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ ▷ Update policy weights

      $\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$ ▷ Adjust temperature

      $\bar{\theta}_i \leftarrow \tau\theta_i + (1 - \tau)\bar{\theta}_i$ for $i \in \{1, 2\}$ ▷ Update target network weights

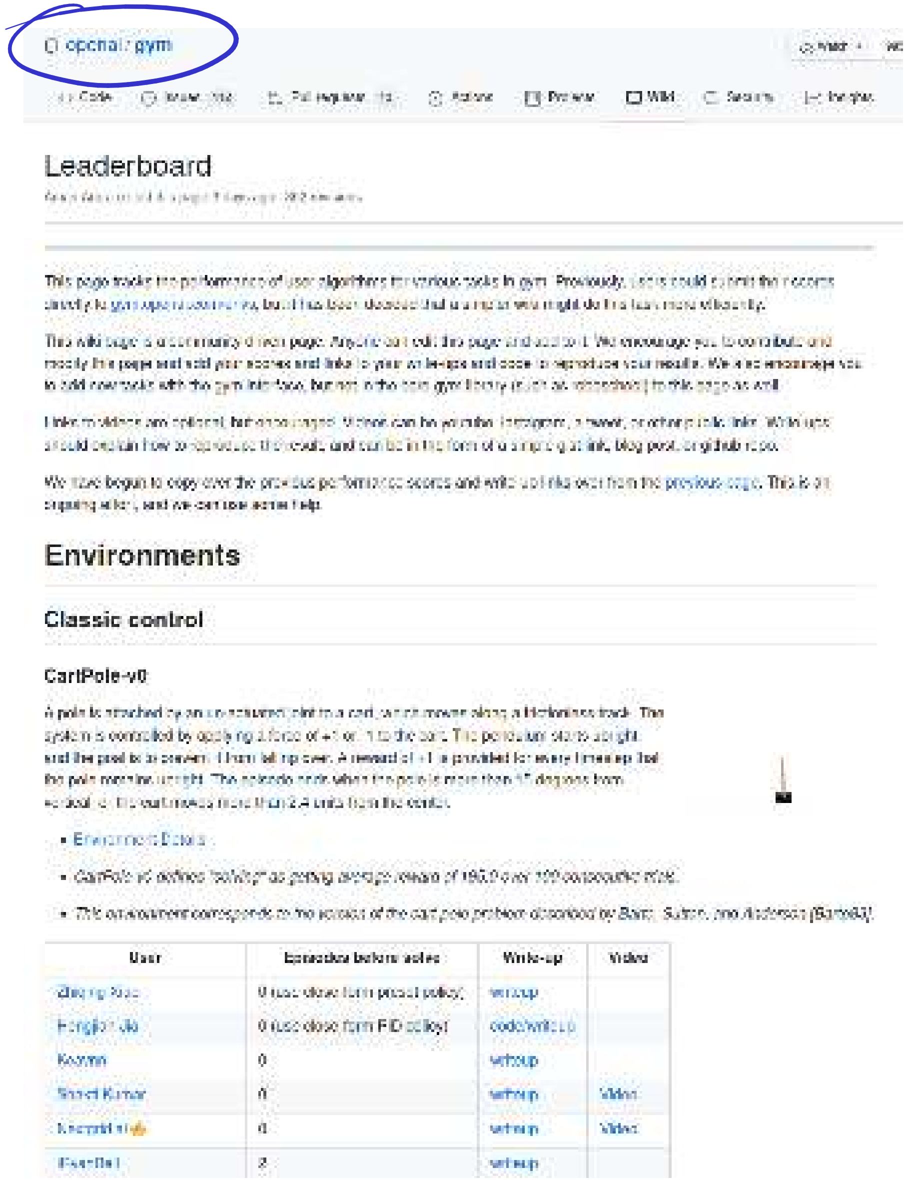
    end for

  end for

Output: θ_1, θ_2, ϕ ▷ Optimized parameters

Wisdom

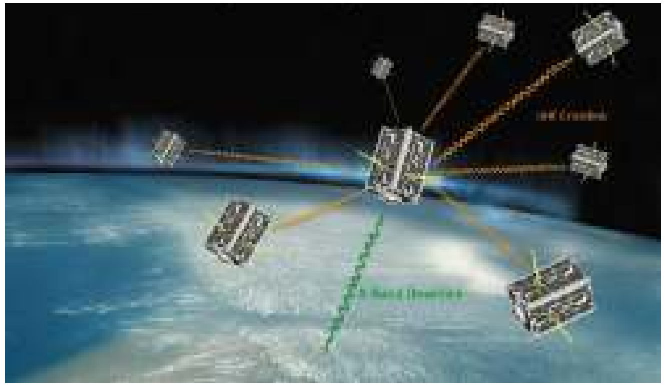
Deep RL: The Dream



Using Deep RL for your problem

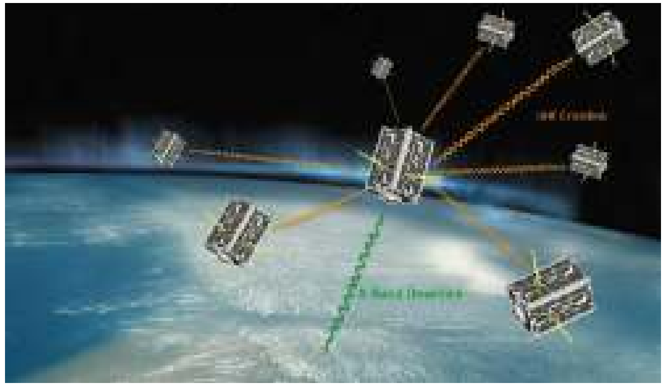
Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)



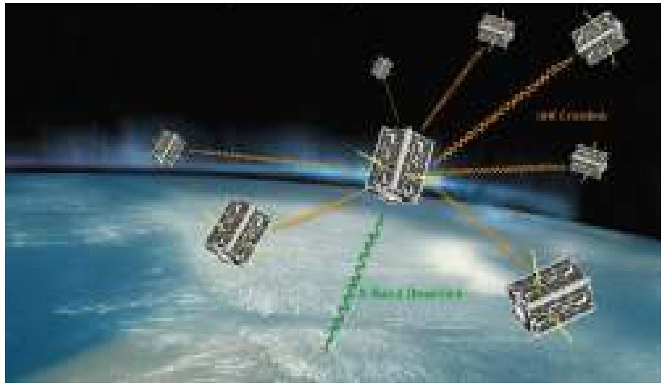
Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics



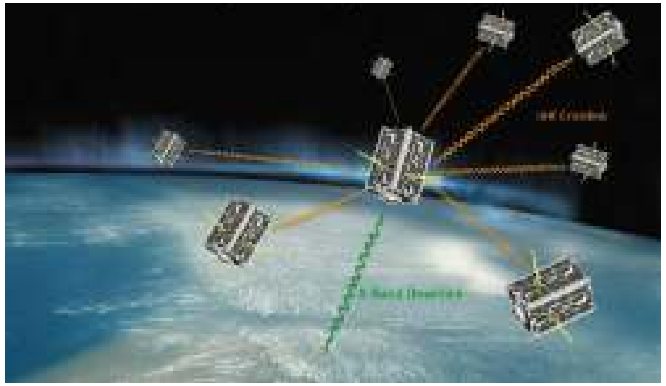
Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems



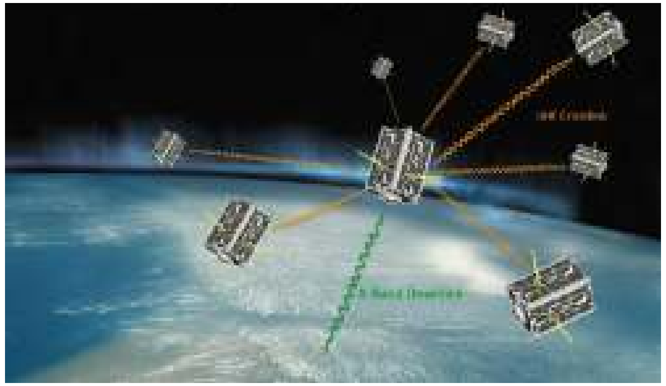
Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems
4. Fire up open-ai baselines



Using Deep RL for your problem

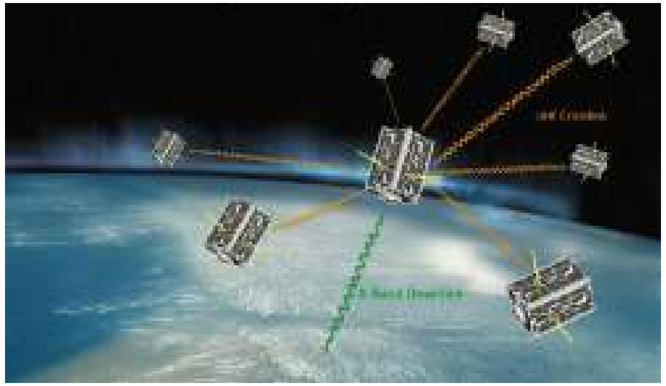
1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems
4. Fire up open-ai baselines
5. Does it work??

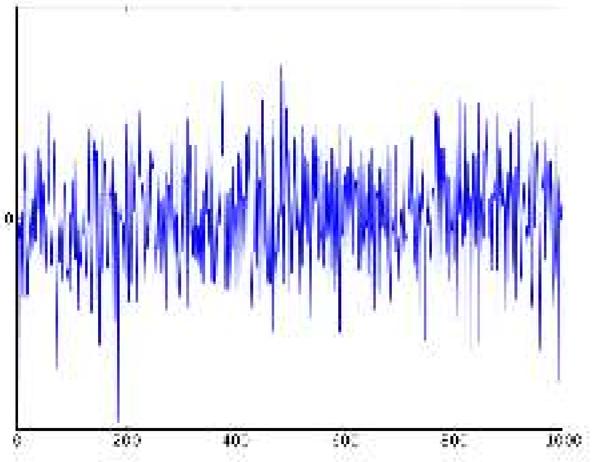


openai / **baselines**

Using Deep RL for your problem

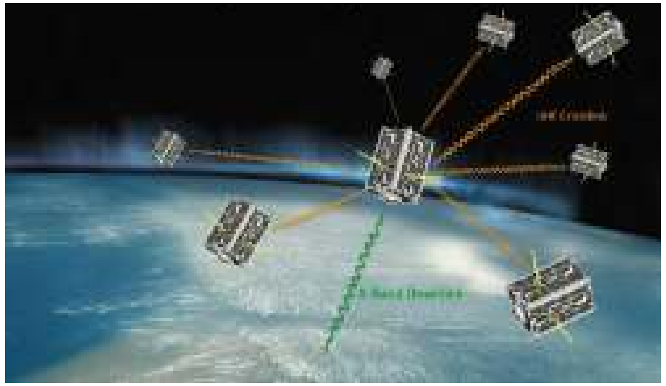
1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems
4. Fire up open-ai baselines
5. Does it work??

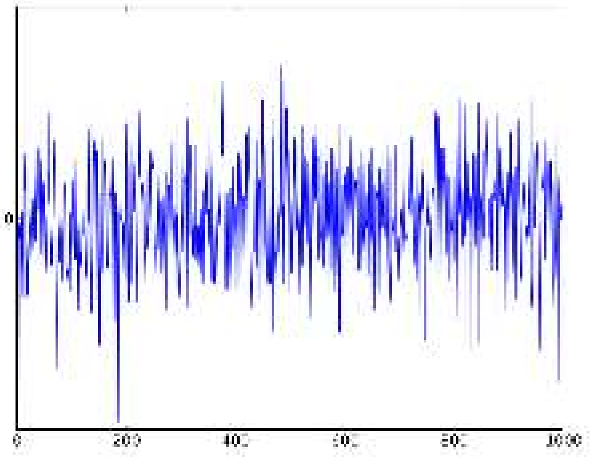




Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems
4. Fire up open-ai baselines
5. Does it work??

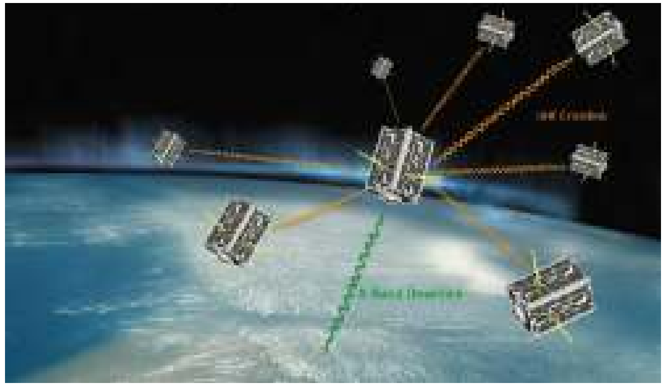


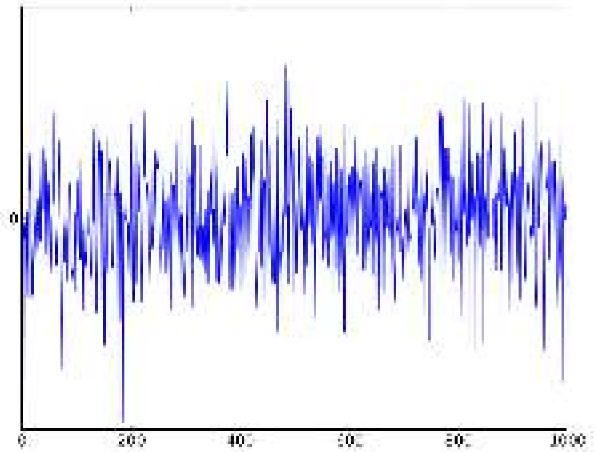


Why not?

Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems
4. Fire up open-ai baselines
5. Does it work??



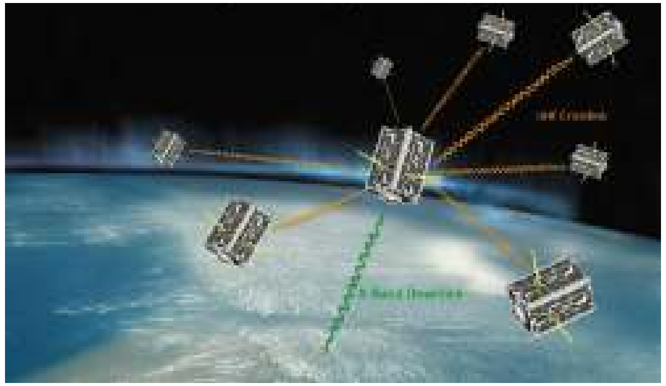


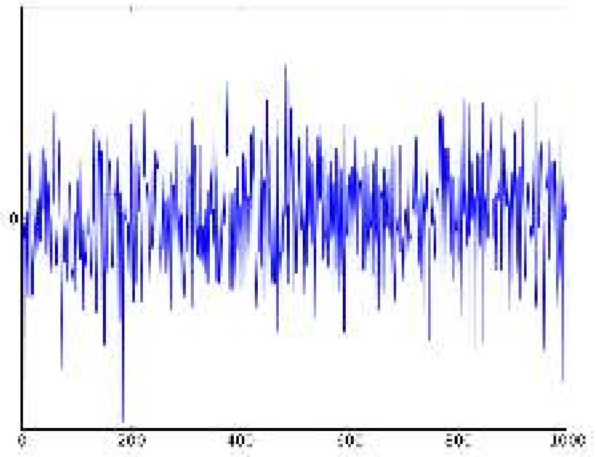
Why not?

- Hyperparameters?

Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems
4. Fire up open-ai baselines
5. Does it work??





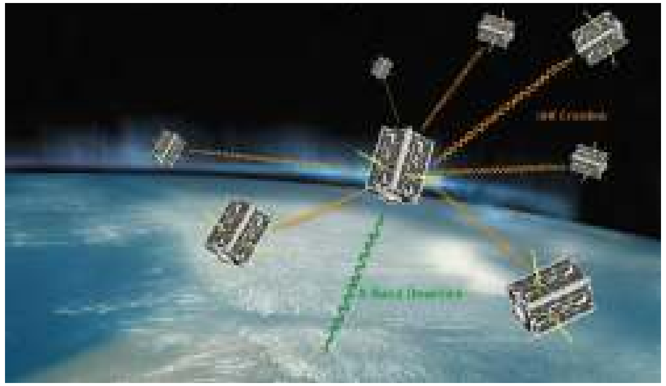
Why not?

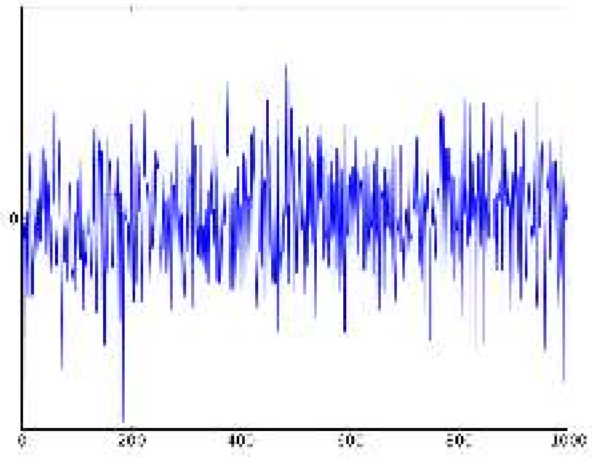
- Hyperparameters?
- Reward scaling?

Deep RL that matters

Using Deep RL for your problem

1. Some interesting problem (smallsat swarm)
2. Spend weeks theorizing about the exact-right cost function and dynamics
3. Decide RL can solve all of your problems
4. Fire up open-ai baselines
5. Does it work??

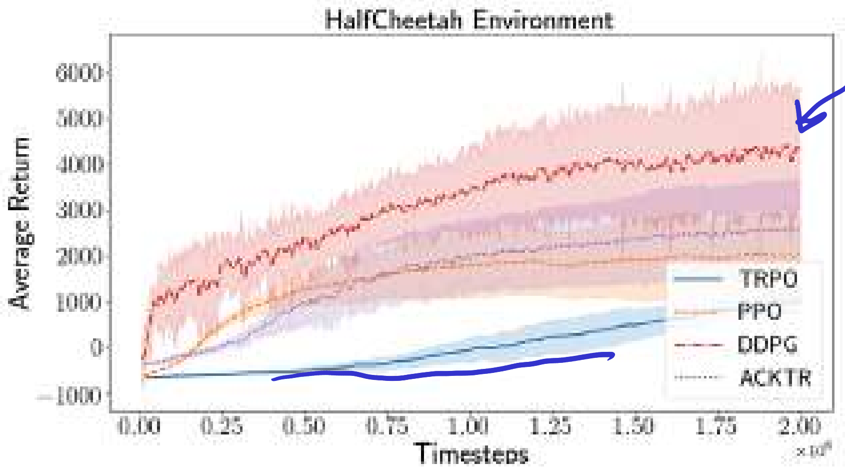


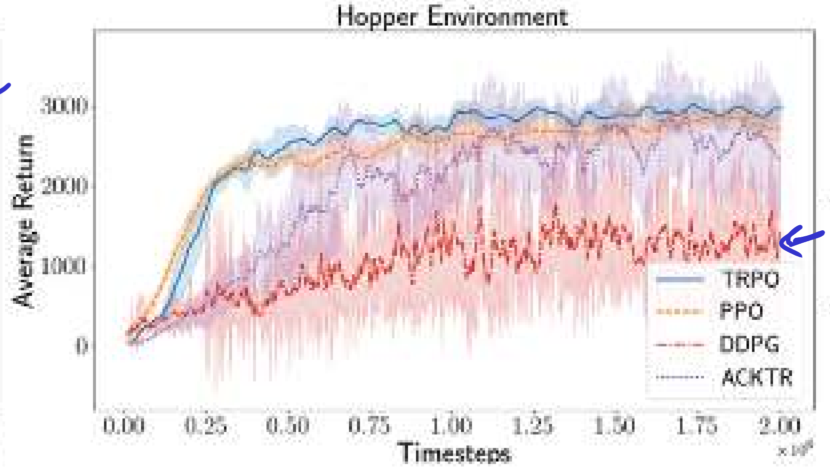


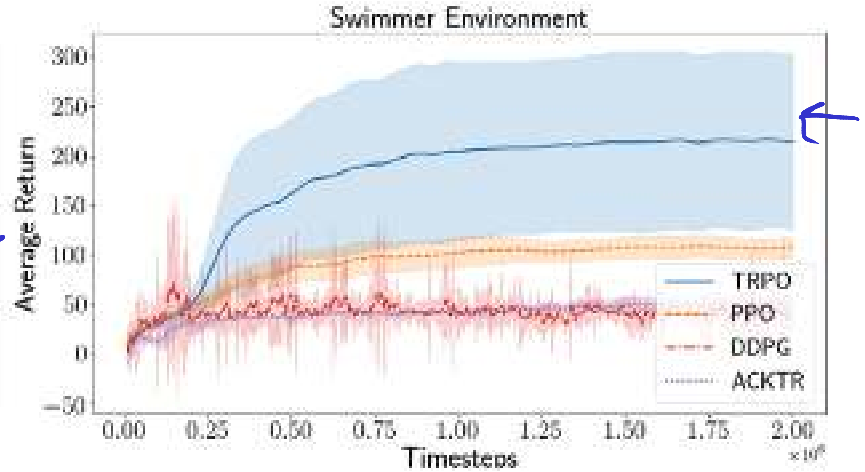
Why not?

- Hyperparameters?
- Reward scaling?
- Not enough training time????

Algorithms

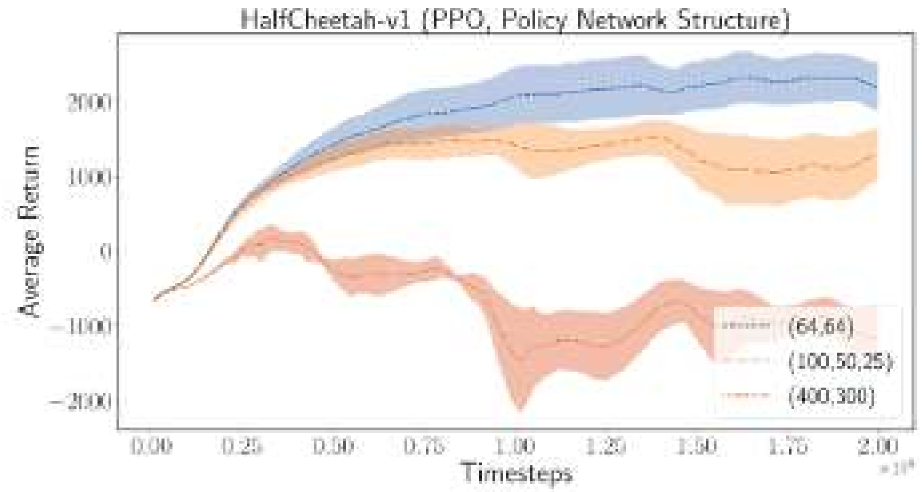

HalfCheetah Environment
Average Return
Timesteps
TRPO
PPO
DDPG
ACKTR


Hopper Environment
Average Return
Timesteps
TRPO
PPO
DDPG
ACKTR

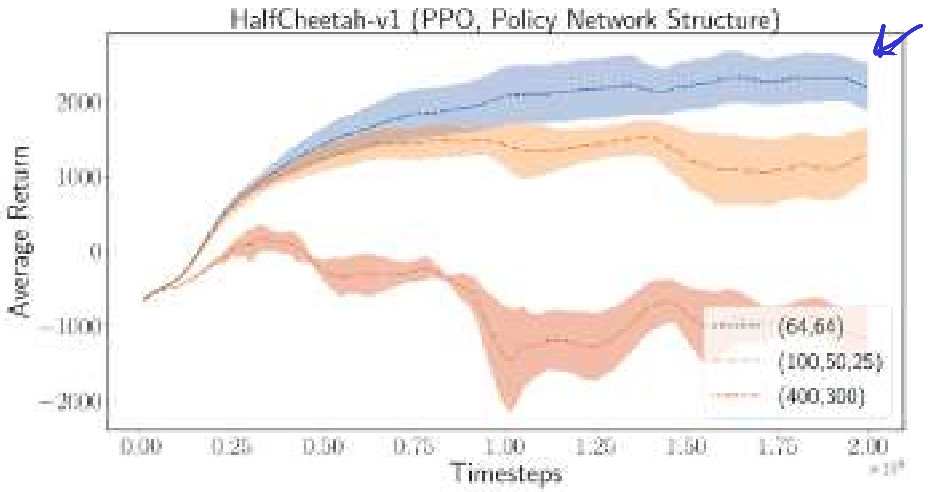

Swimmer Environment
Average Return
Timesteps
TRPO
PPO
DDPG
ACKTR

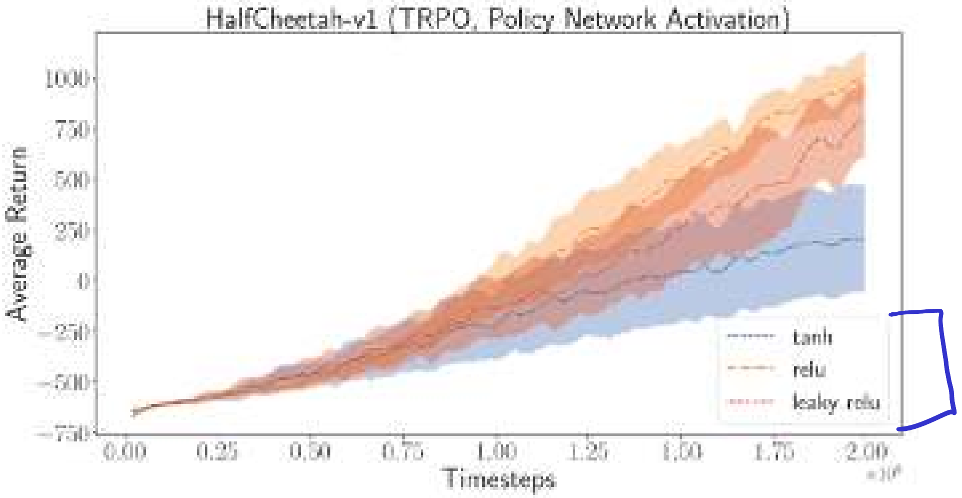
Policy Network Architecture

Policy Network Architecture

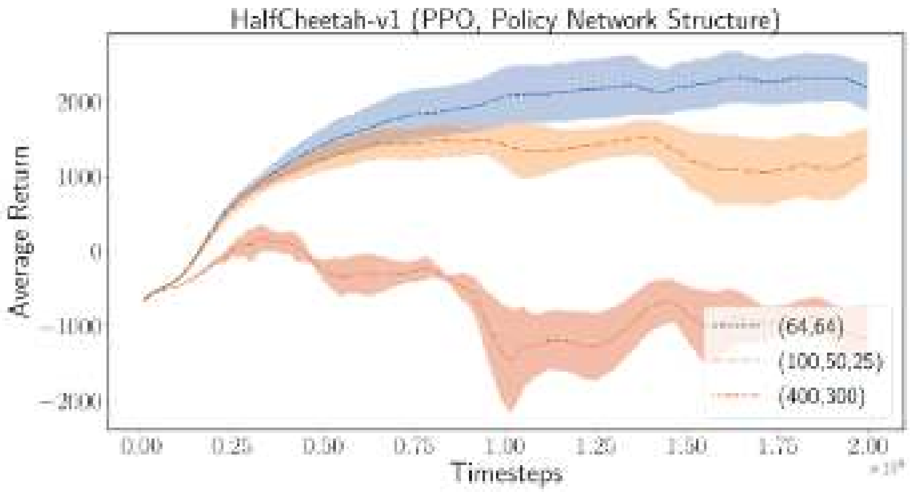


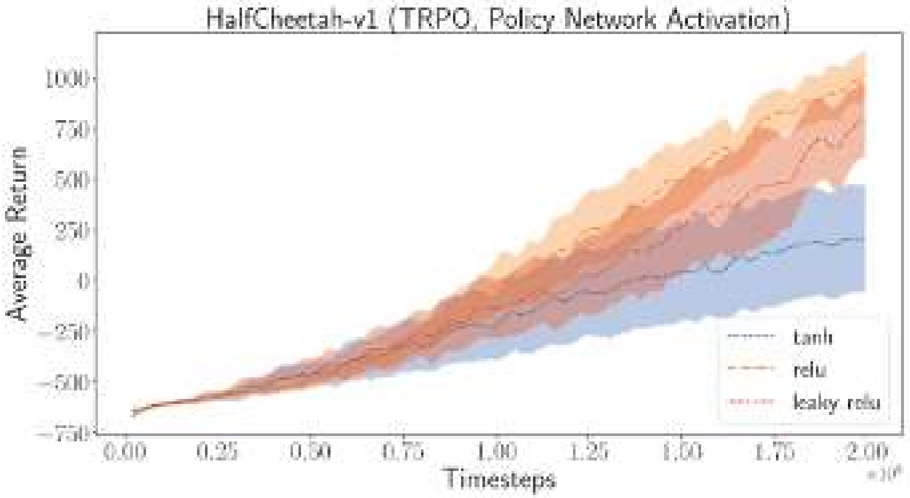
Policy Network Architecture

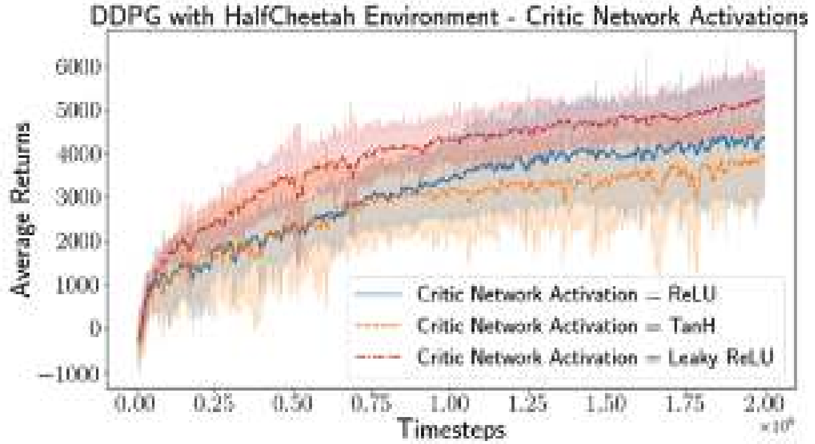




Policy Network Architecture







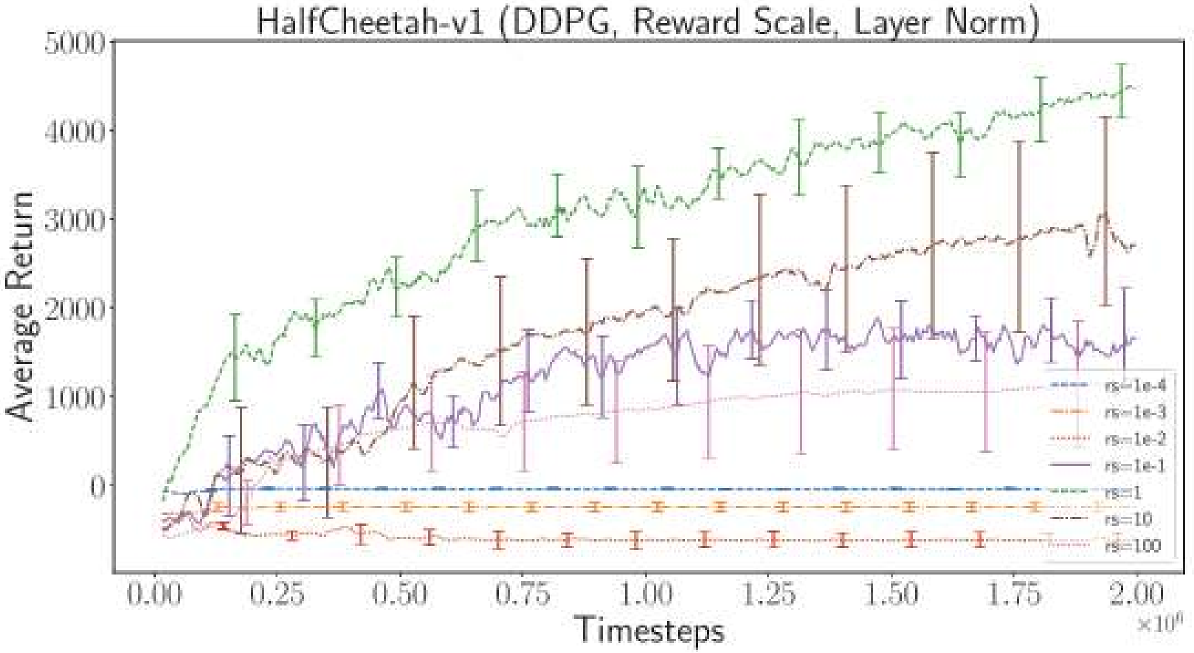
Reward Rescaling

Reward Rescaling

"simply multiplying the rewards generated from an environment by some scalar"

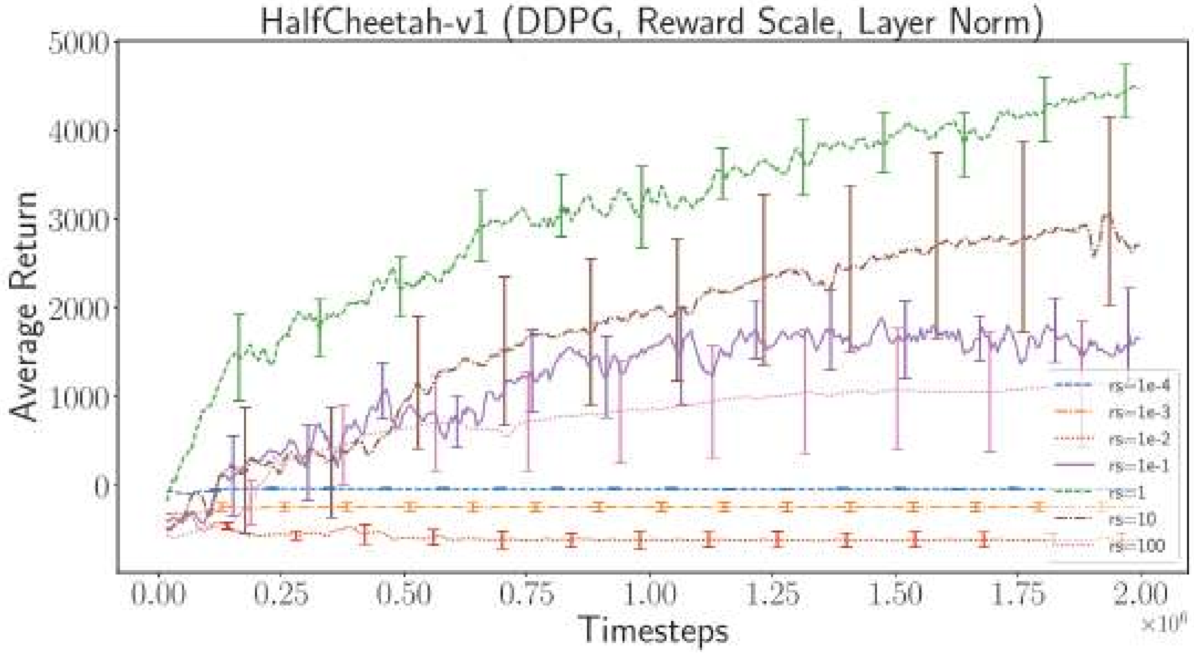
Reward Rescaling

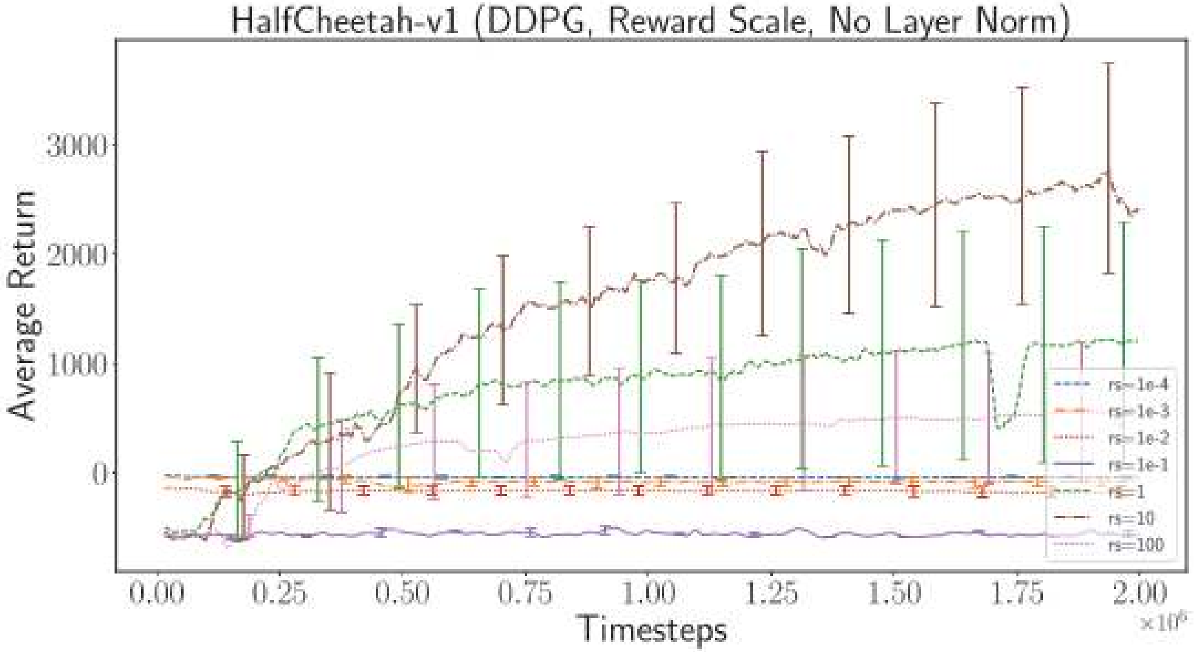
"simply multiplying the rewards generated from an environment by some scalar"



Reward Rescaling

"simply multiplying the rewards generated from an environment by some scalar"





Statistical Significance

Statistical Significance

"Unfortunately, in recent reported results, it is not uncommon for the top-N trials to be selected from among several trials (Wu et al. 2017; Mnih et al. 2016)"

Statistical Significance

"Unfortunately, in recent reported results, it is not uncommon for the top-N trials to be selected from among several trials (Wu et al. 2017; Mnih et al. 2016)"

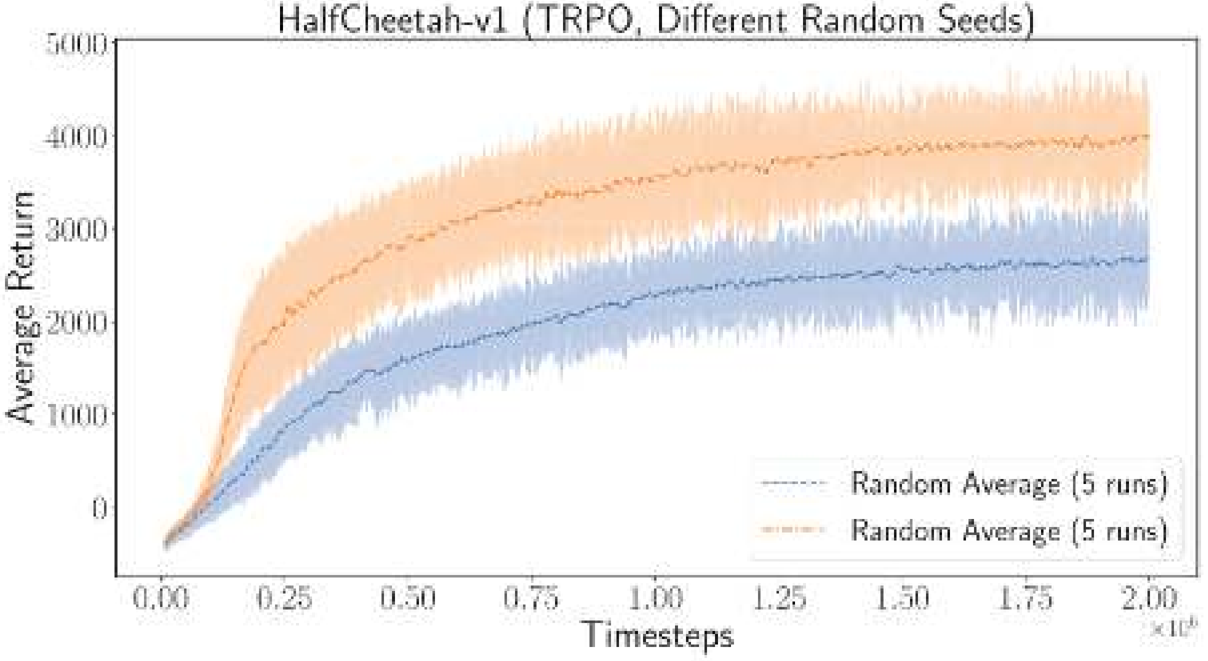
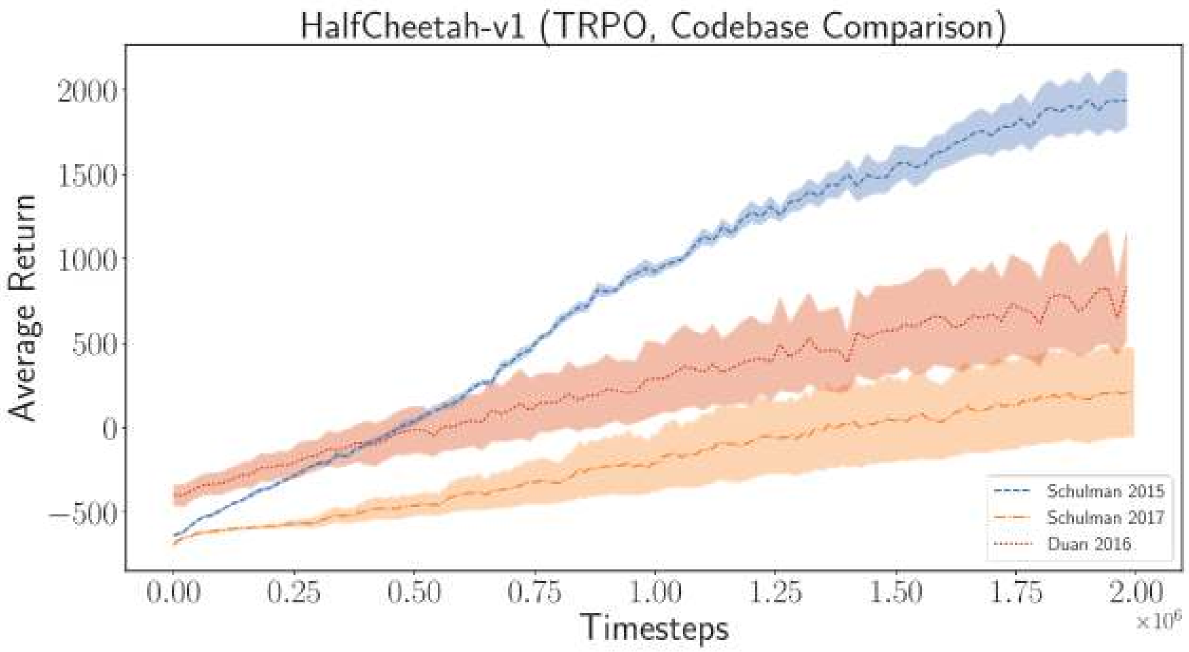
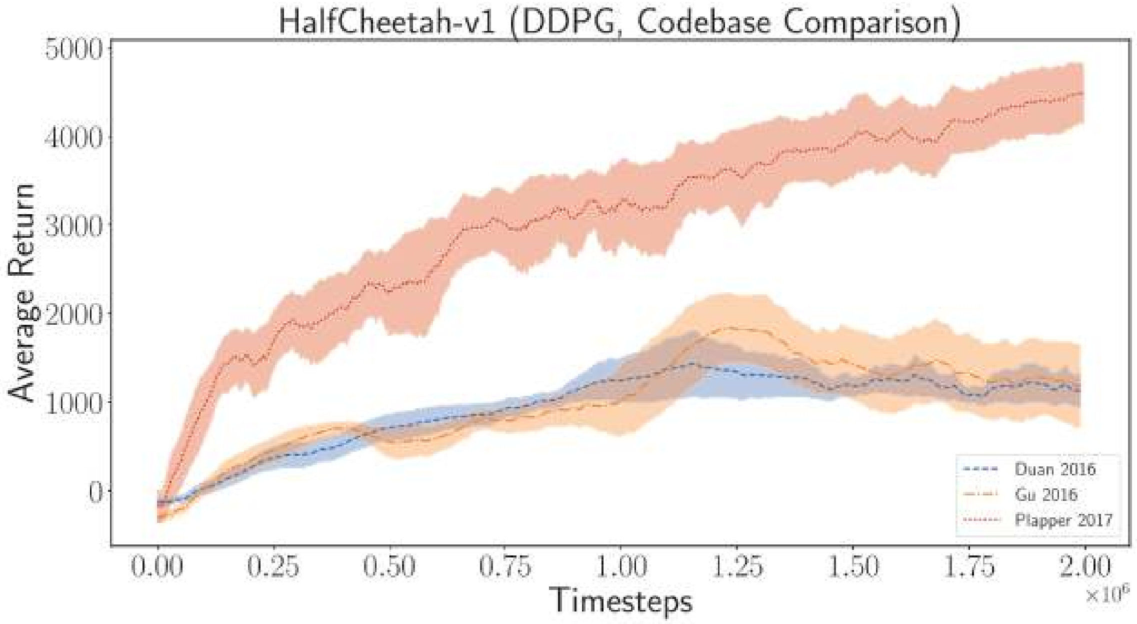


Figure 5: TRPO on HalfCheetah-v1 using the same hyperparameter configurations averaged over two sets of 5 different random seeds each. The average 2-sample t-test across entire training distribution resulted in $t = -9.0916$, $p = 0.0016$.

Codebases


HalfCheetah-v1 (TRPO, Codebase Comparison)
Average Return
Timesteps
$\times 10^6$
Schulman 2015
Schulman 2017
Duan 2016


HalfCheetah-v1 (DDPG, Codebase Comparison)
Average Return
Timesteps
$\times 10^6$
Duan 2016
Gu 2016
Plapper 2017

How to choose an RL Algorithm

How to choose an RL Algorithm

(According to Sergey Levine)

How to choose an RL Algorithm

(According to Sergey Levine)

How to choose an RL Algorithm

(According to Sergey Levine)

Sample
Efficiency

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

← Fewer Samples

How to choose an RL Algorithm

(According to Sergey Levine)

Sample
Efficiency

Ease of Use
/Stability

← Fewer Samples

More Stable (?) →

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

← Fewer Samples

More Stable (?) →

Model-Based RL

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use /Stability

← Fewer Samples

More Stable (?) →

Model-Based RL

Model-Based Deep RL

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

← Fewer Samples

More Stable (?) →

Off-Policy Q-Learning

Model-Based RL

Model-Based Deep RL

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

← Fewer Samples

More Stable (?) →

Model-Based RL

Off-Policy Q-Learning

Model-Based Deep RL

Actor Critic

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

← Fewer Samples

More Stable (?) →

Model-Based RL

Model-Based Deep RL

Off-Policy Q-Learning

Actor Critic

On Policy Policy Gradient

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

← Fewer Samples

More Stable (?) →

Model-Based RL

Model-Based Deep RL

Off-Policy Q-Learning

Actor Critic

On Policy Policy Gradient

Evolutionary/ Gradient Free

How to choose an RL Algorithm

(According to Sergey Levine)

Sample Efficiency

Ease of Use / Stability

← Fewer Samples

More Stable (?) →

Model-Based RL

Model-Based Deep RL

Off-Policy Q-Learning

SAC

Actor Critic

PPO

On Policy Policy Gradient

Evolutionary/ Gradient Free

(Most people use SAC or PPO)

Where Does RL Work?

Where Does RL Work?

- Cooling servers

Where Does RL Work?

- Cooling servers
- Winning at Go