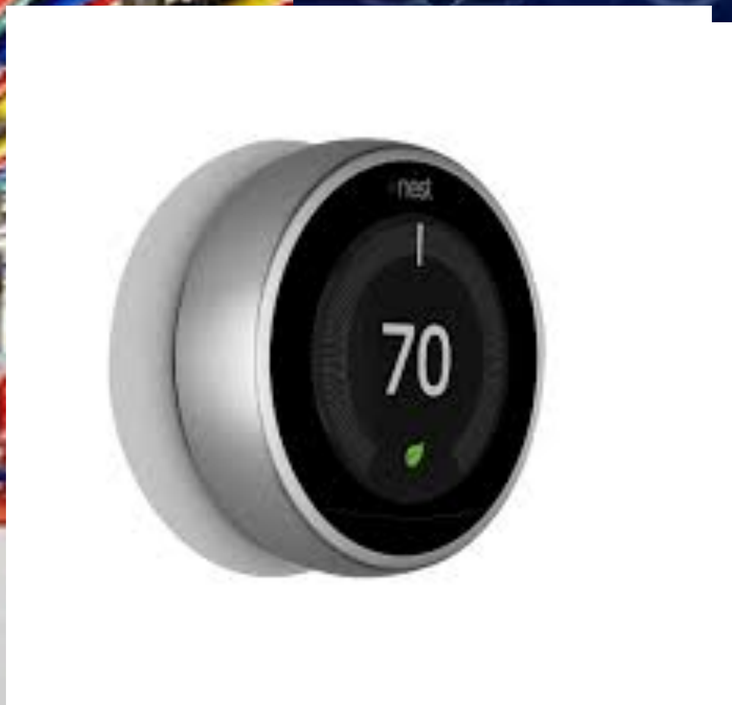
Reducing Sampling Error in the Monte Carlo Policy Gradient Estimator

Josiah Hanna and Peter Stone
Department of Computer Science
The University of Texas at Austin

50 millions actions taken

21 days, millions of games

1.5 years of compute



Can reinforcement learning be data efficient enough for real world applications?

Reinforcement Learning

Learn a policy that maps the world state to an action that maximizes long term utility.

Reinforcement Learning

$$\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

$$v(\pi_\theta) = \sum_s \Pr(s|\pi_\theta) \sum_a \pi_\theta(a|s) Q^{\pi_\theta}(s, a)$$

Unknown

Known

Reach Destination

+1

Probability = 0.85

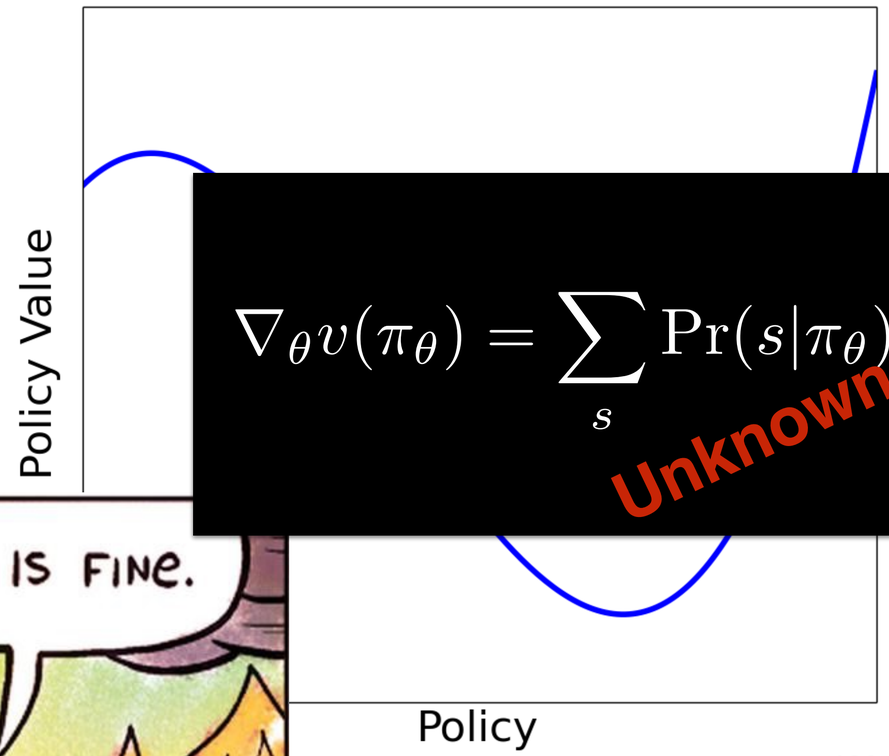
Crash

-100

$$v(\pi_\theta) = \mathbf{E}[Q^{\pi_\theta}(S, A)]$$

"How good is taking action A in state S"

Policy Gradient Reinforcement Learning



$$\nabla_\theta v(\pi_\theta) = \sum_s \Pr(s|\pi_\theta) \sum_a \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a|s)$$

Unknown

Known

$$\nabla_\theta v(\pi_\theta) \approx \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}[Q^{\pi_\theta}(S_i, A_i) \nabla_\theta \log \pi_\theta(A_i|S_i)]$$

Monte Carlo Policy Gradient

1. Execute current policy for m steps.
2. Update policy with Monte Carlo policy gradient estimate.
3. Throw away observed data and repeat (on-policy).

Sampling Error

Proportion = 0.15

Reach Destination

+1

Probability = 0.15

For a finite amount of data, it may appear that the wrong policy generated the data.

Probability = 0.85

Crash

-100

Proportion = 0.85

Correcting Sampling Error

Pretend data was generated by policy that most closely matches the observed data.

$$\pi_\phi = \texttt{argmax}_{\phi'} \sum_{i=1}^{m} \log \pi_{\phi'}(a_i|s_i)$$

Correct weight on each state-action pair towards the policy we actually took actions with.

Importance Sampling Correction

$$\nabla_\theta v(\pi_\theta) \approx \frac{1}{m} \sum_{i=1}^{m} \boxed{\frac{\pi_\theta(a_i|s_i)}{\pi_\phi(a_i|s_i)}} Q^{\pi_\theta}(S_i, A_i) \nabla_\theta \log \pi_\theta(A_i|S_i)$$

Is this method on-policy or off-policy?

On-policy: Can only use data from the current policy.

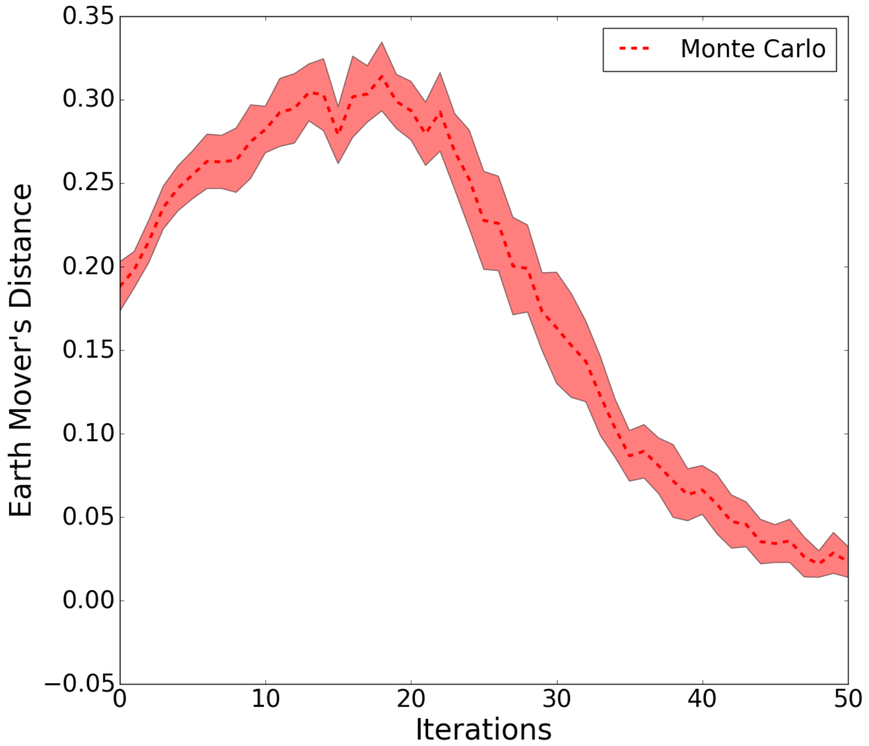
Off-policy: Can use data from any policy.

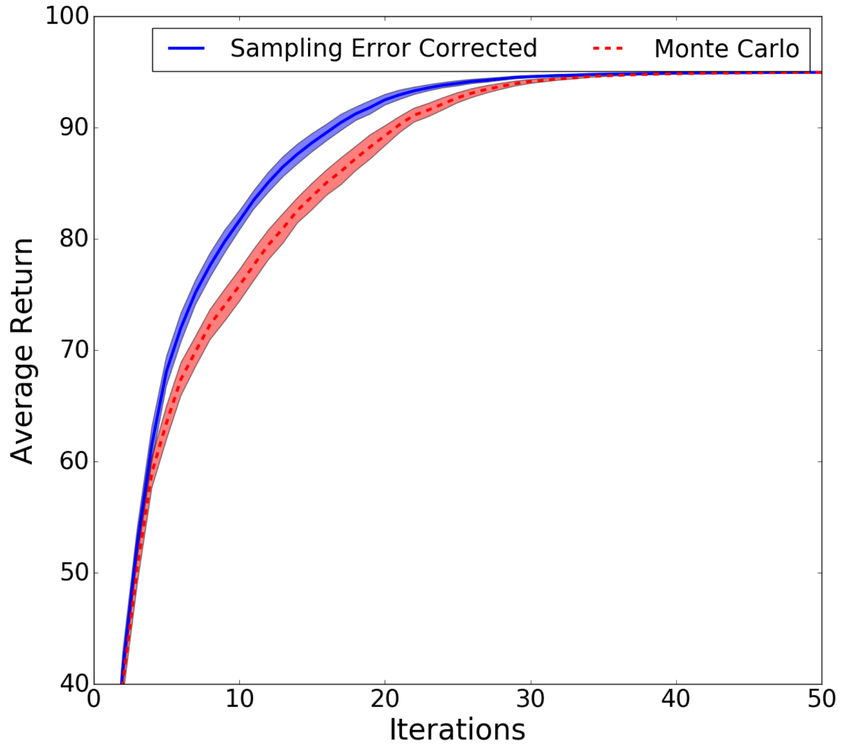
Our method pretends on-policy data is off-policy data
and uses importance sampling to correct!

Sampling Error Corrected Policy Gradient

1. Execute current policy for m steps.
2. Estimate empirical policy with maximum likelihood estimation.
3. Update policy with Sampling Error Corrected (SEC) policy gradient estimate.
4. Throw away data and repeat (on-policy).

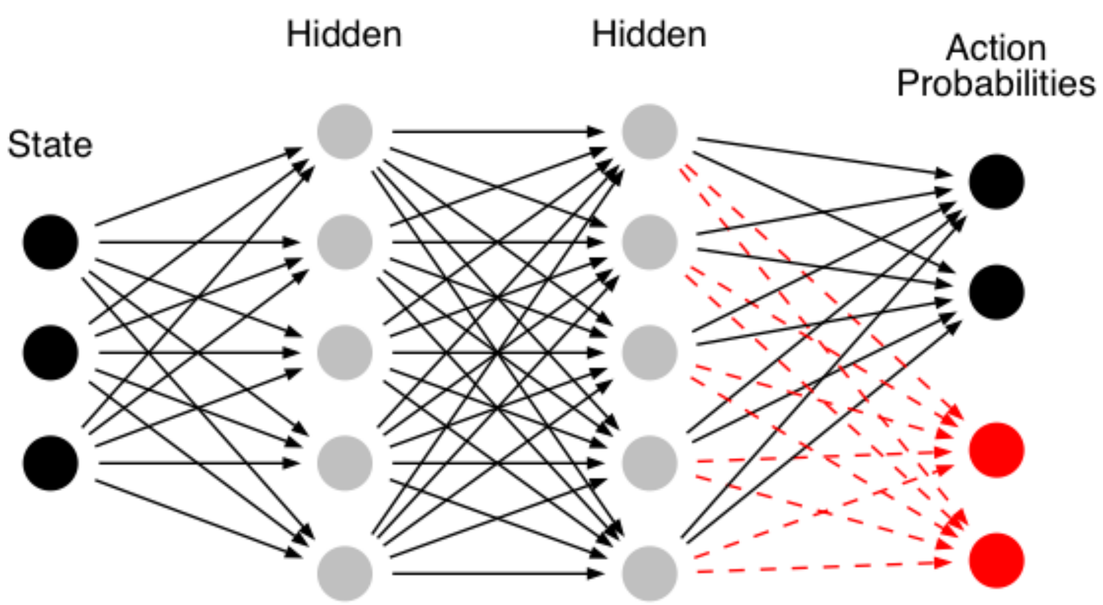
Empirical Results

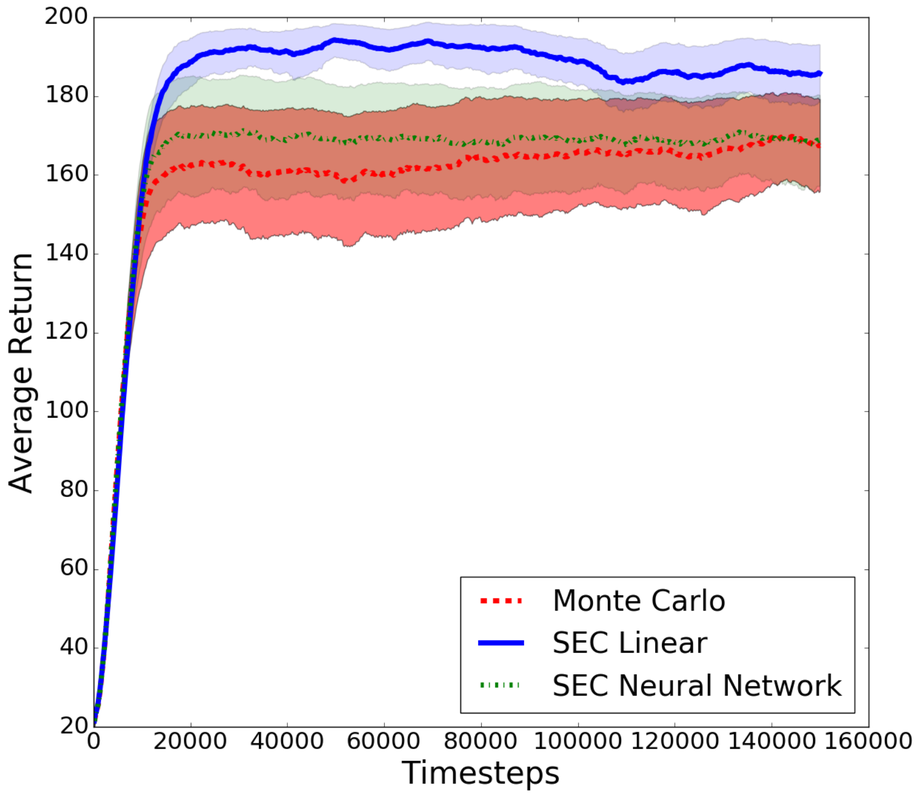




GridWorld
Discrete State and Actions

Empirical Results





Cartpole
Continuous state and discrete actions

Related Work

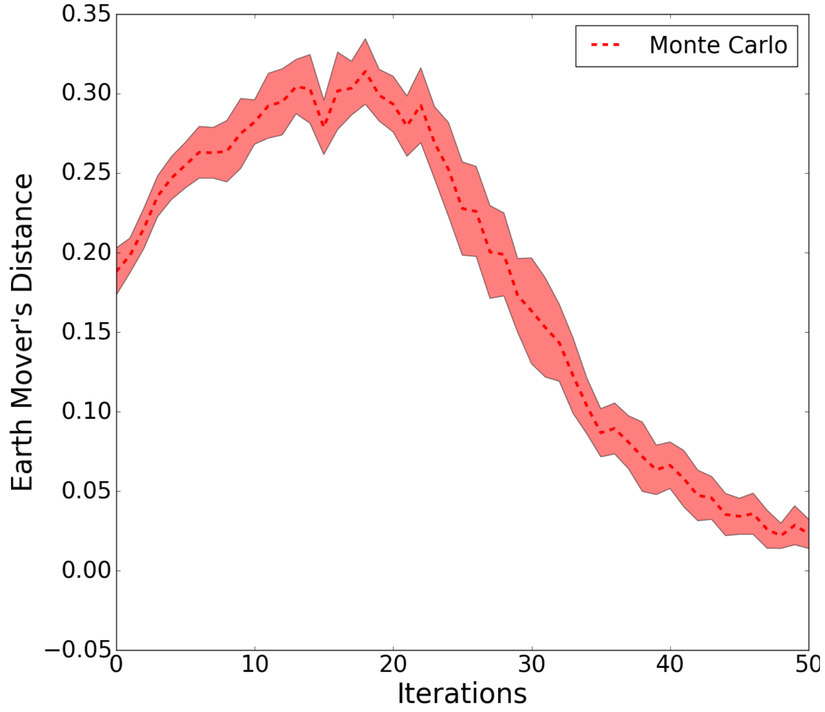
1. Expected SARSA (van Seijen et al. 2009).
2. Expected Policy Gradients (Ciosek and Whiteson 2018).
3. Estimated Propensity Scores (Hirano et al. 2003, Li et al. 2015).
4. Many people outside of RL + Bandits:
    - Blackbox importance sampling (Liu and Lee 2017), Bayesian Monte Carlo (Gharamani and Rasmussen 2003).

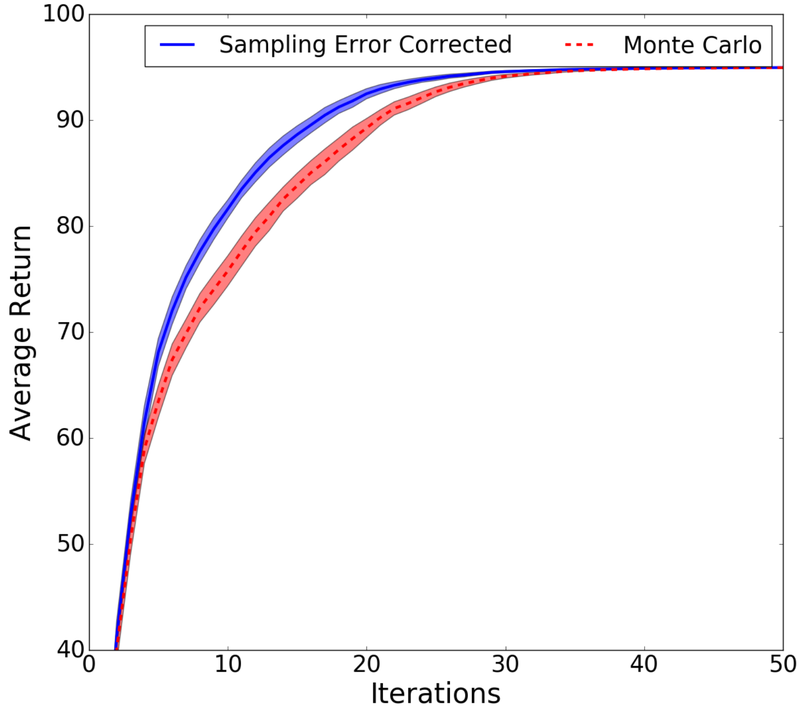
1. Any Monte Carlo method will have sampling error with finite data.
2. Sampling Error can slow down learning in policy gradient methods.
3. We introduced the sampling error corrected policy gradient estimator to address this problem.
4. Similar approach can be used for other Monte Carlo estimators.
   - For example: on- and off-policy policy evaluation.

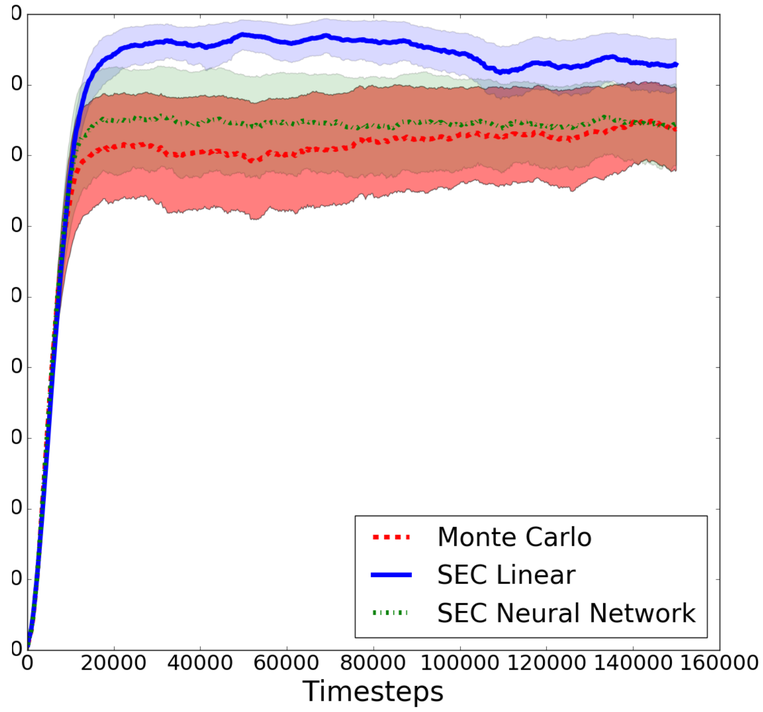
Josiah Hanna, Scott Niekum, Peter Stone (to appear ICML 2019)

Open Questions

1. Finite sample bias / variance analysis.
2. Correcting sampling error in online RL methods.



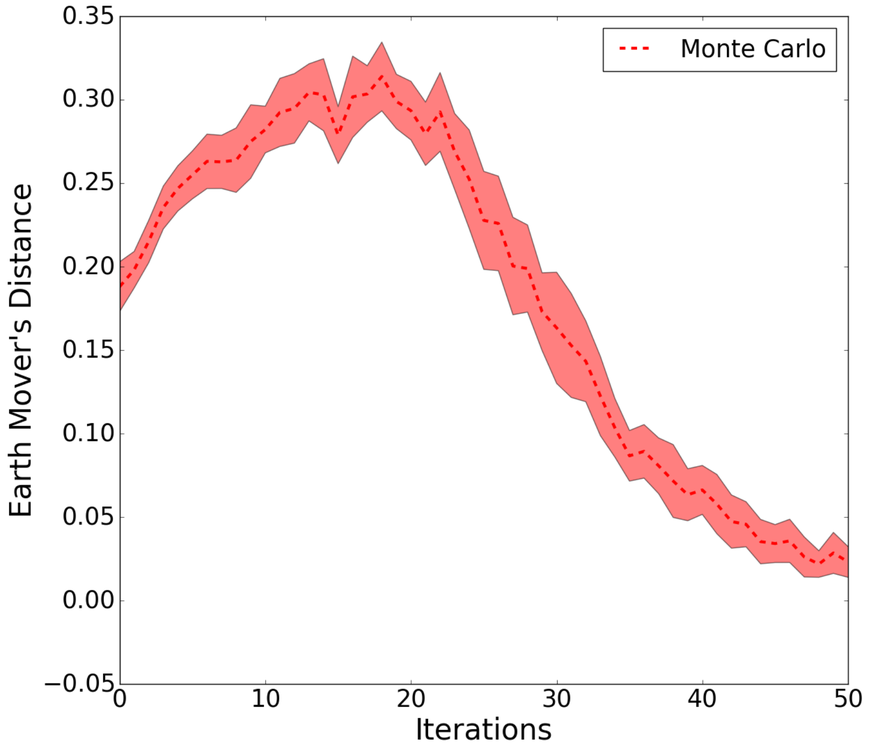


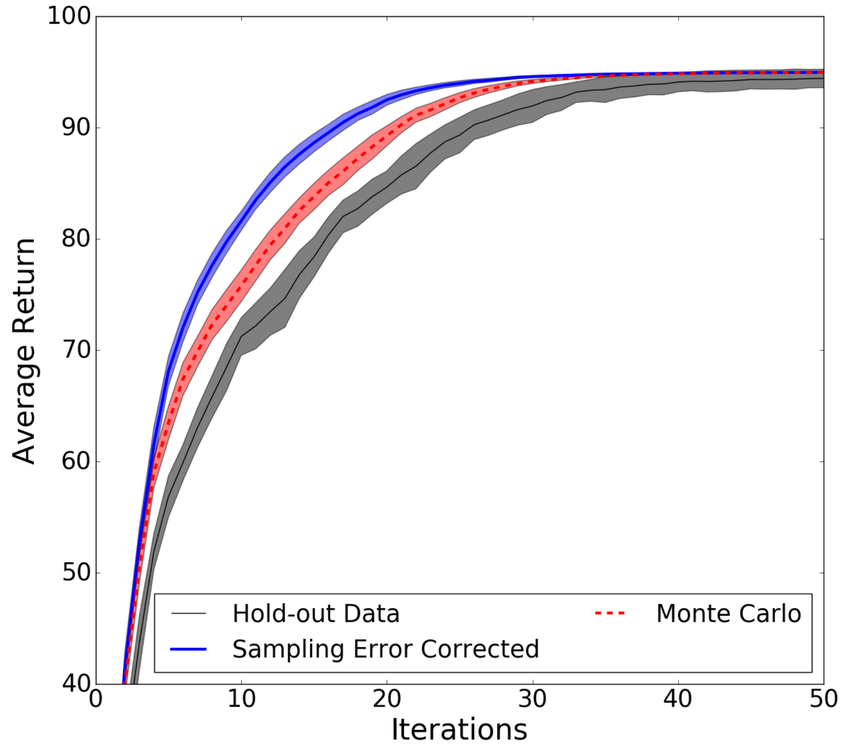


Thank you!
Questions?
jphanna@cs.utexas.edu

Ceci n’est pas un blank slide.

Empirical Results





GridWorld
Discrete State and Actions