Data Efficient Reinforcement Learning with Off-Policy and Simulated Data

Josiah Hanna

PhD Oral Defense

TEXAS
The University of Texas at Austin


900
OXYGEN
ACTIVISION


900
OXYGEN
ACTIVISION


ALPHAGO
00:10:29
AlphaGo
Google DeepMind
LEE SEDOL
00:01:00


900


ALPHAGO
00:10:29
phaGo
Google DeepMind
LEE SEDOL
00:01:00


900


ALPHAGO
00:10:29
phaGo
Google DeepMind
LEE SEDOL
00:01:00

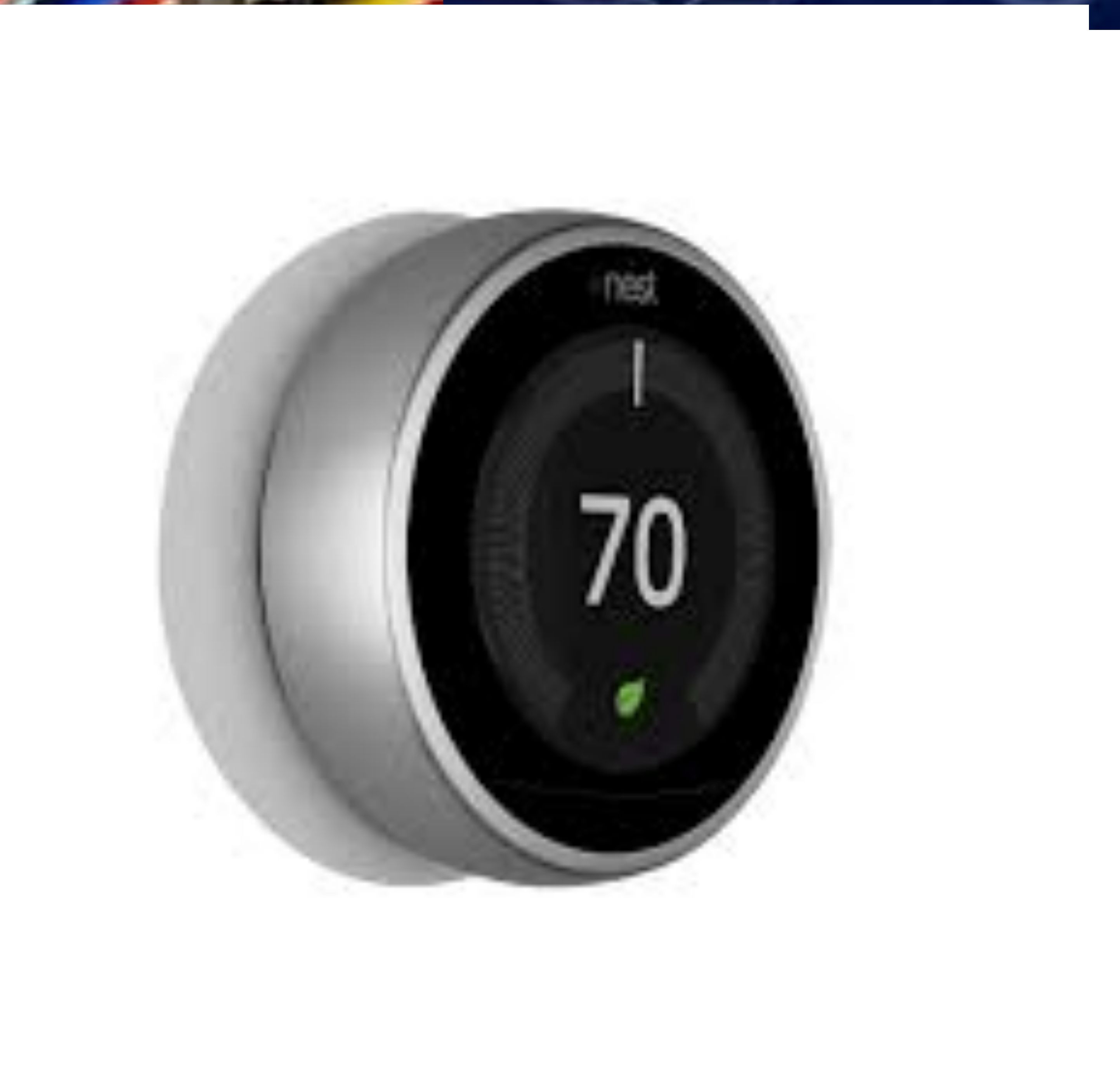

INTERNET
ADVERTISING
ENGINE
WEB
PER
PAY
CONSUMER
MARKETERS
COMPANIES
ONLINE
INFORMATION
PRODUCT
BUSINESS
PRODUCTS
SERVICES
AFFILIATE
CONSUMERS
WEBSITE
SPECIFIC
MARKETING
SALES
SEARCH

Josiah Hanna

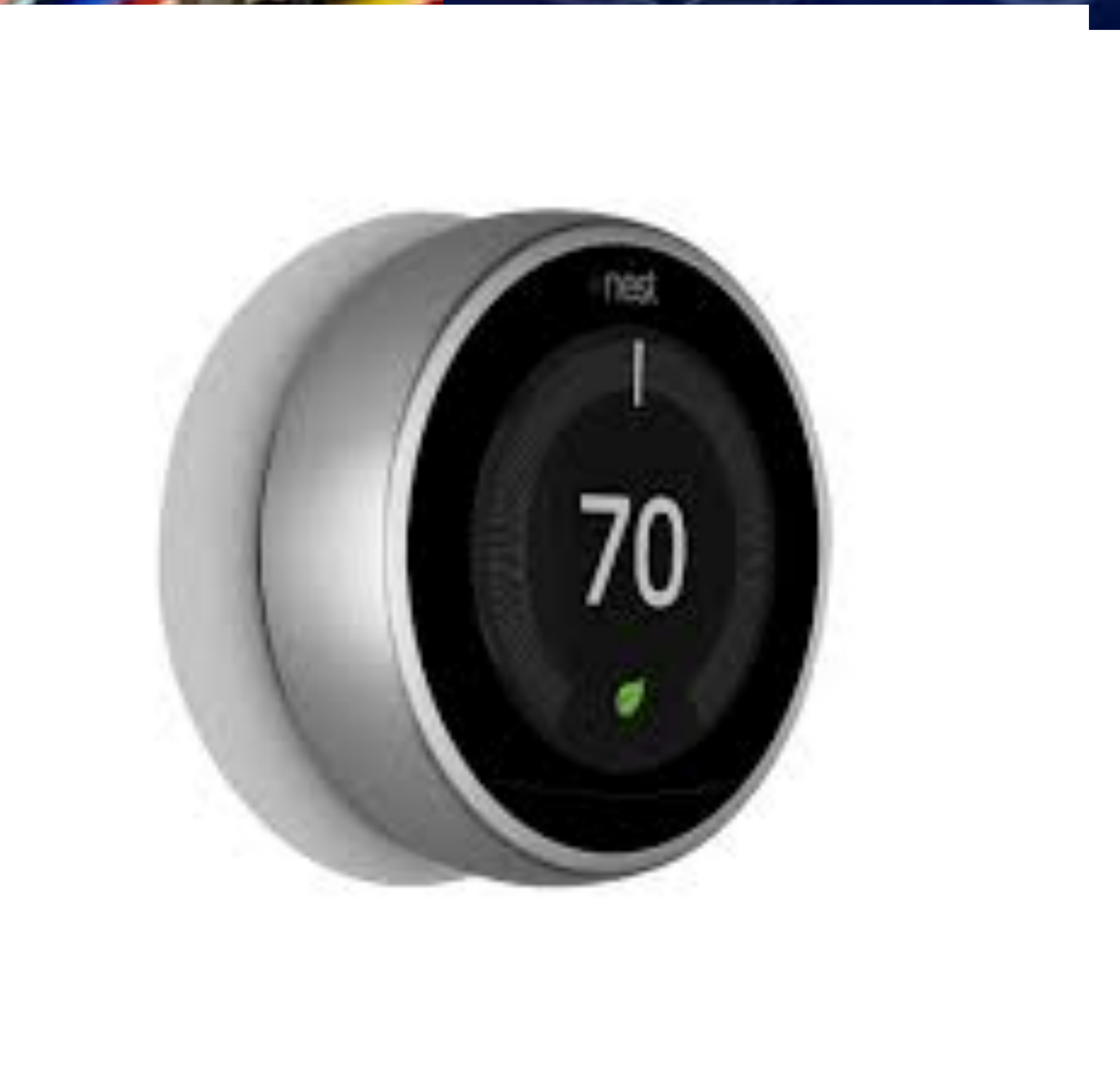

900

ALPHAGO
00:10:29
phaGo
LEE SEDOL
00:01:00


INTERNET
ADVERTISING
ENGINE
WEB
MARKETING
ONLINE
MARKETERS
PRODUCT
BUSINESS
PRODUCTS
CONSUMERS
SPECIFIC
WEBSITE
SERVICES
AFFILIATE
INFORMATION
COMPANIES
PER
PAY
CONSUMER
USE
SALES
SEARCH


nest
70

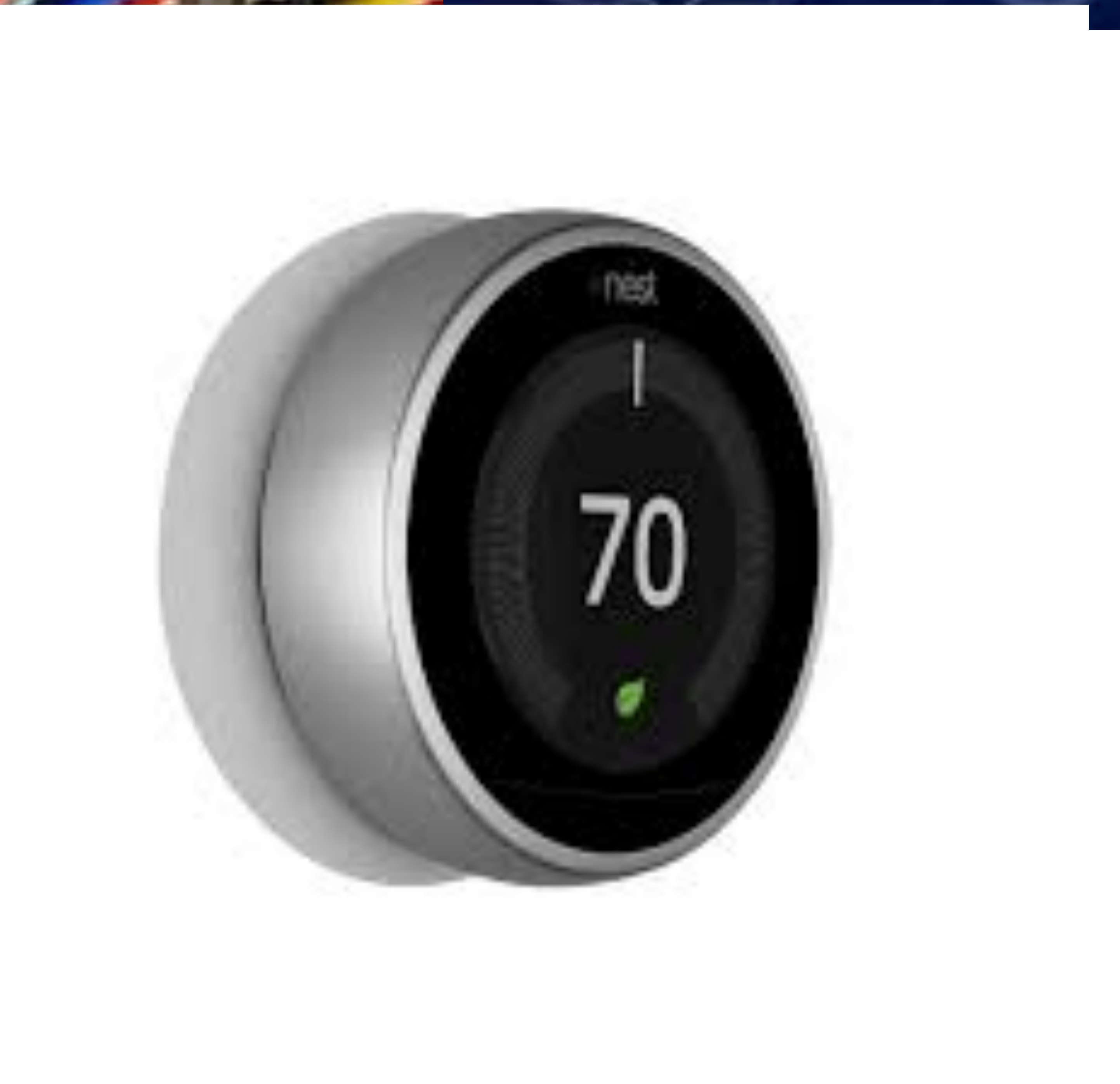
50 millions
actions taken




900
50 millions actions taken


ALPHAGO
00:10:29
21 days, millions of games
LEE SEDOL
00:01:00
phaGo


INTERNET
ADVERTISING
ENGINE
USE
WEB
PER
PAY
ONLINE
COMPANIES
MARKETERS
PRODUCTS
SERVICES
AFFILIATE
INFORMATION
CONSUMER
PRODUCT
BUSINESS
CONSUMERS
WEBSITE
SPECIFIC
MARKETING
SALES
SEARCH


nest
70


900
50 millions actions taken
ALPHAGO 00:10:29
21 days, millions of games
LEE SEDOL 00:01:00
phaGo
1.5 years of compute
INTERNET
ADVERTISING
ENGINE
WEB
MARKETERS
ONLINE
PRODUCT
BUSINESS
PRODUCTS
COMPANIES
CONSUMERS
WEBSITE
SERVICES
INFORMATION
PER
CONSUMER
SPECIFIC
MARKETING
SEARCH
nest
70

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

Limitations of Reinforcement Learning Algorithms

Limitations of Reinforcement Learning Algorithms

On-Policy

Limitations of Reinforcement Learning Algorithms

On-Policy

- Only use data generated by the current policy.

Limitations of Reinforcement Learning Algorithms

On-Policy

- Only use data generated by the current policy.

On-Environment

Limitations of Reinforcement Learning Algorithms

On-Policy

- Only use data generated by the current policy.

On-Environment

- Simulated data is useless

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

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

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

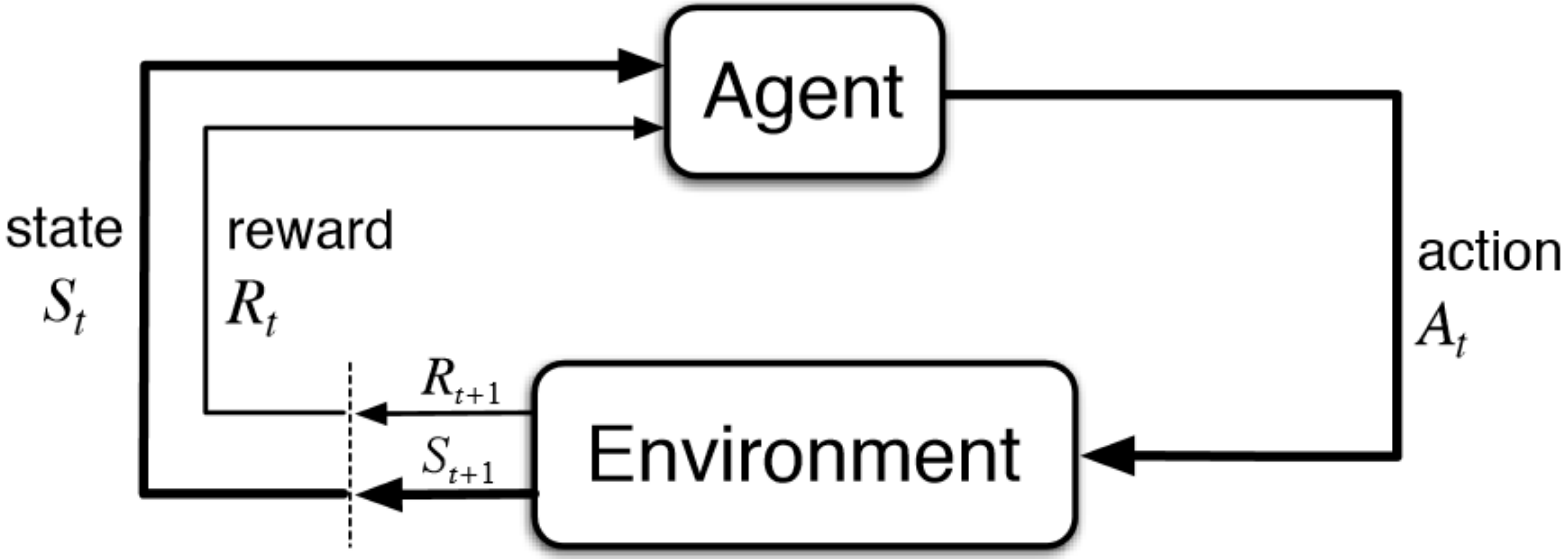
How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

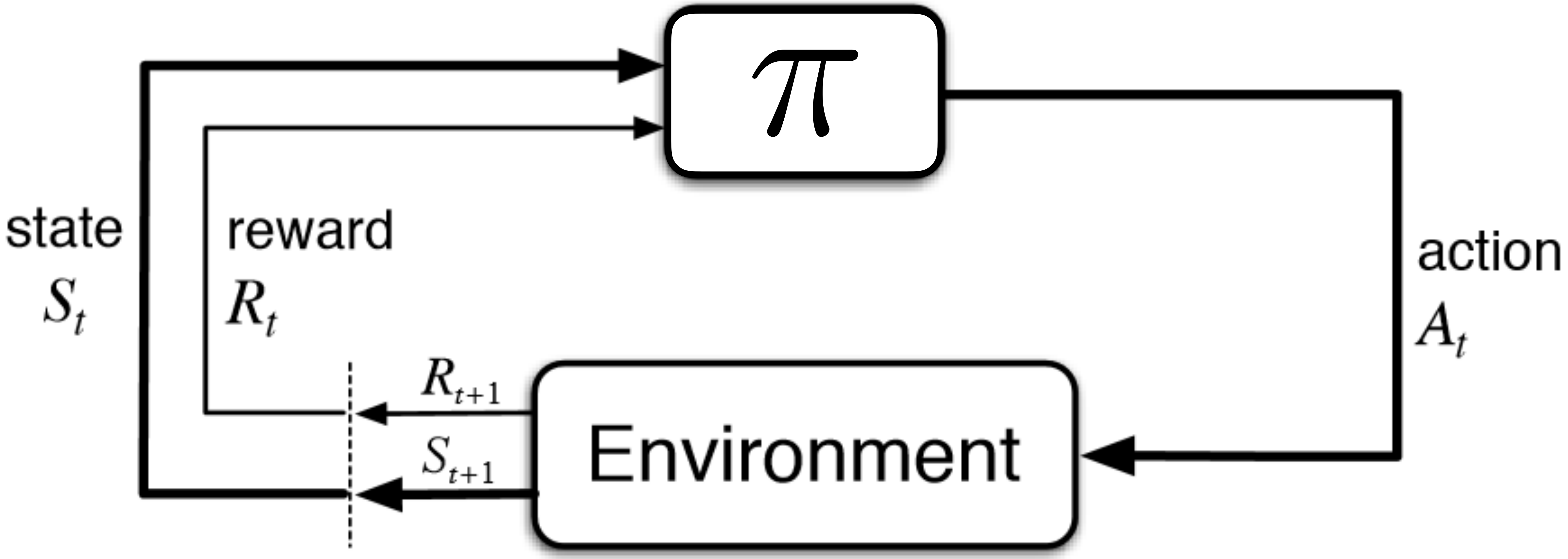
How can an RL agent combine simulated and off-policy data?

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

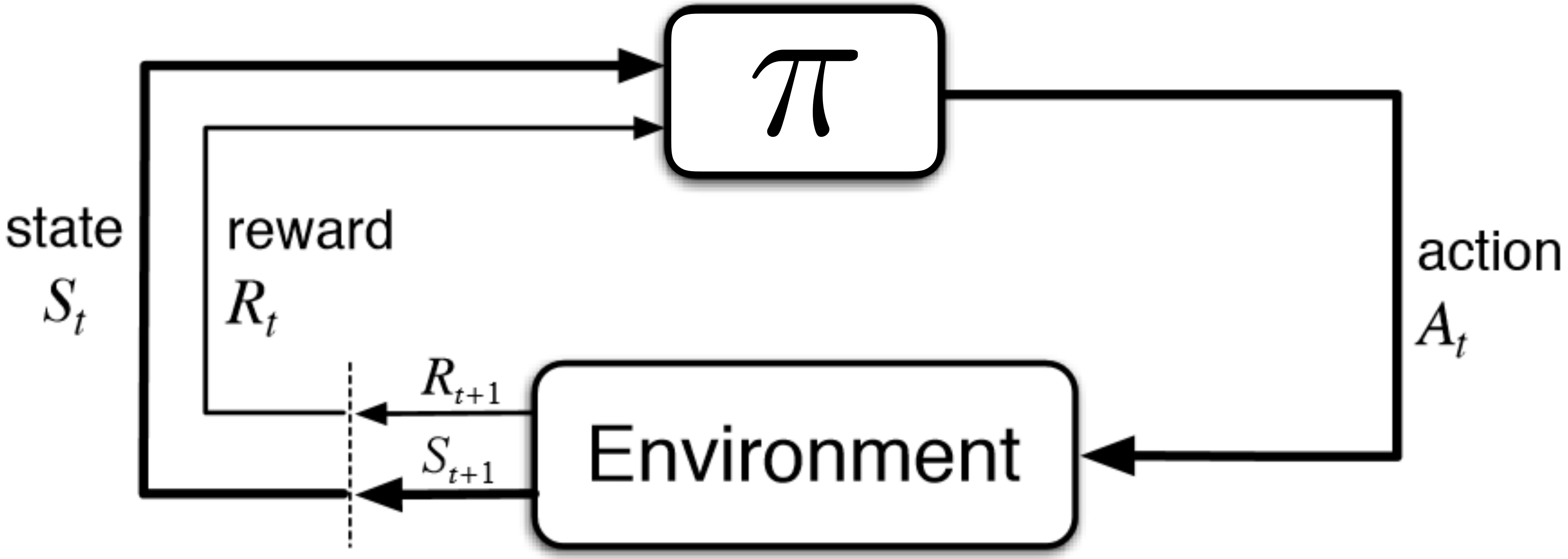
The Reinforcement Learning World



The Reinforcement Learning World



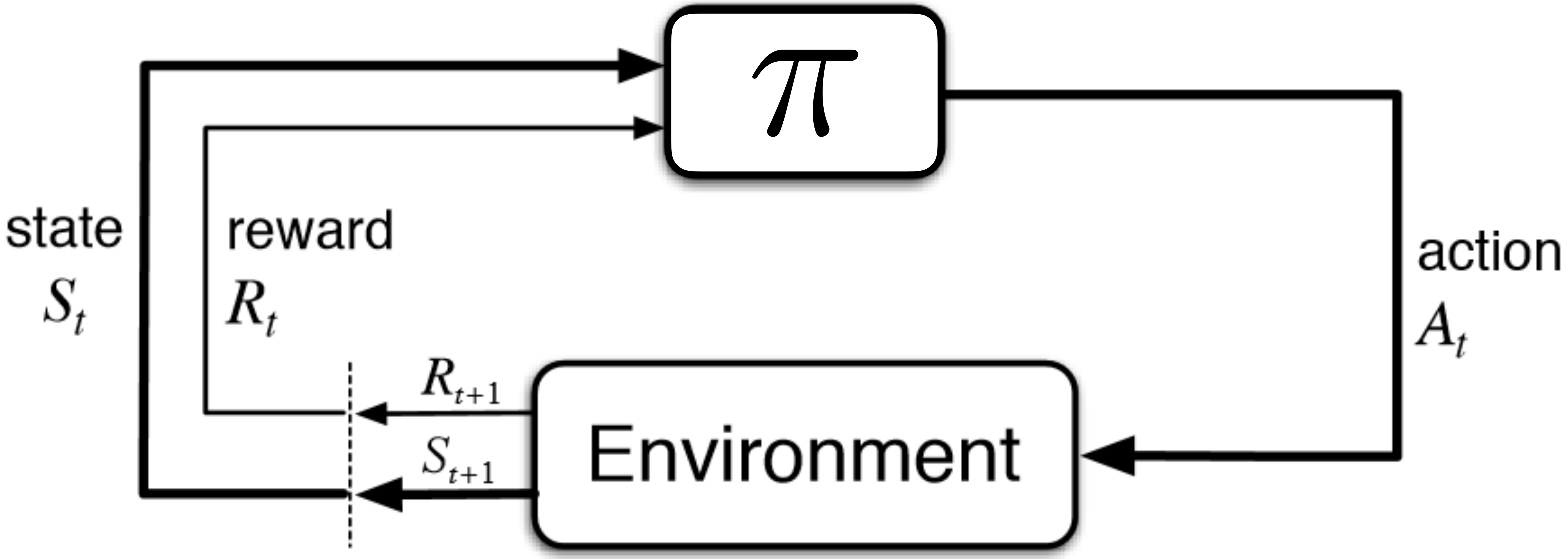
The Reinforcement Learning World



$$S_0$$

Josiah Hanna

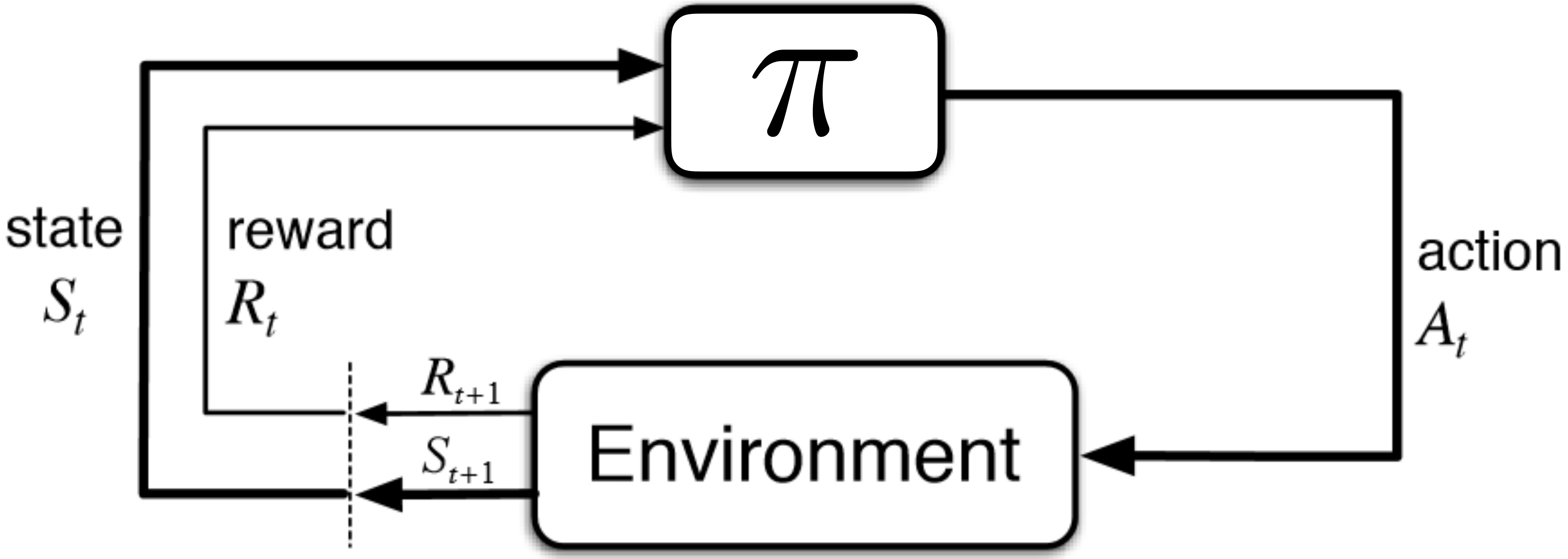
The Reinforcement Learning World



$$S_0, A_0$$

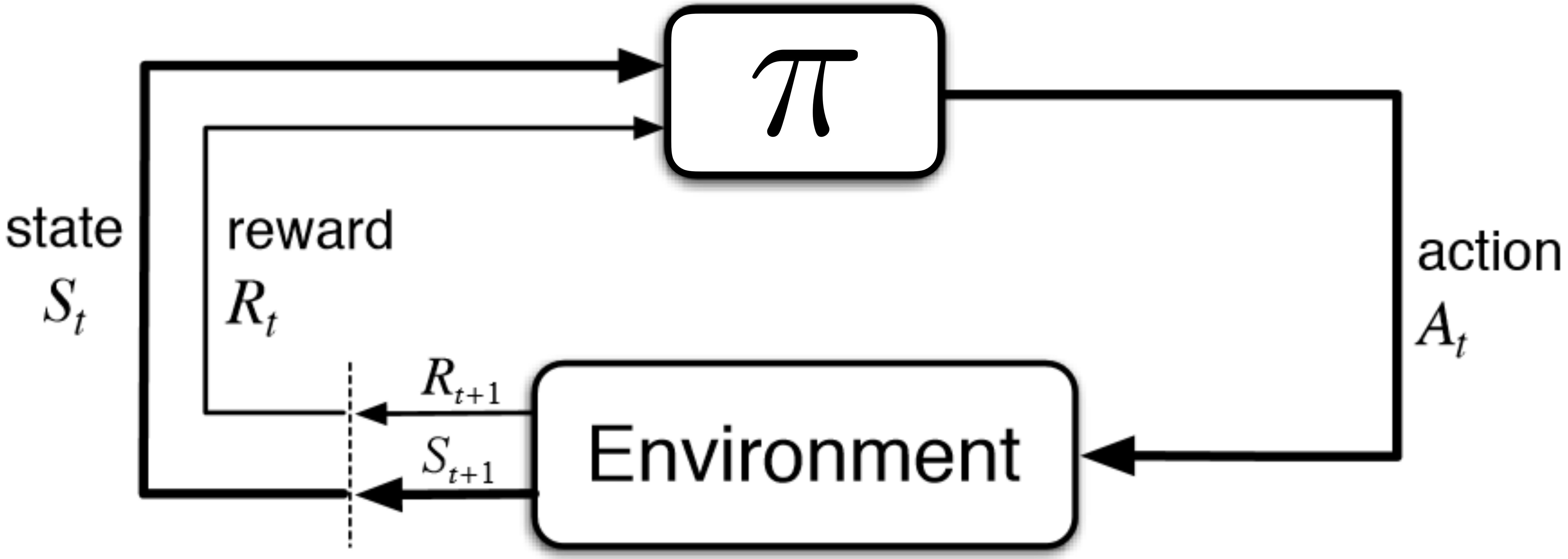
Josiah Hanna

The Reinforcement Learning World



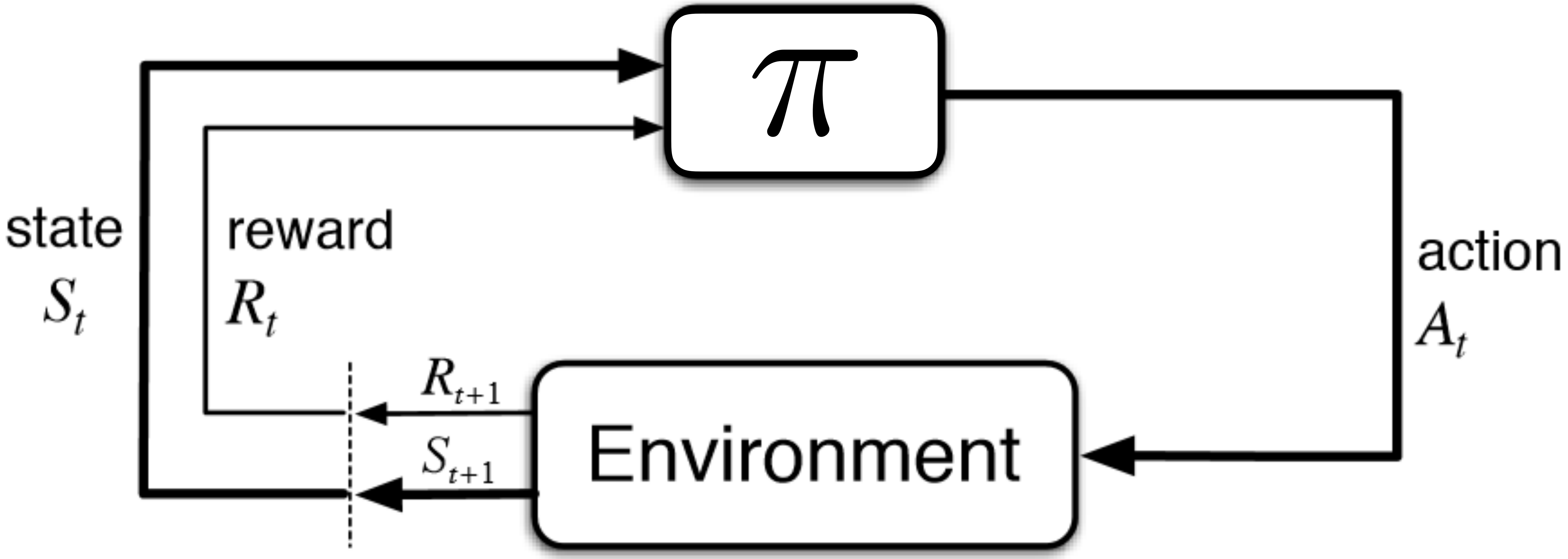
$$S_0, A_0, R_0$$

The Reinforcement Learning World



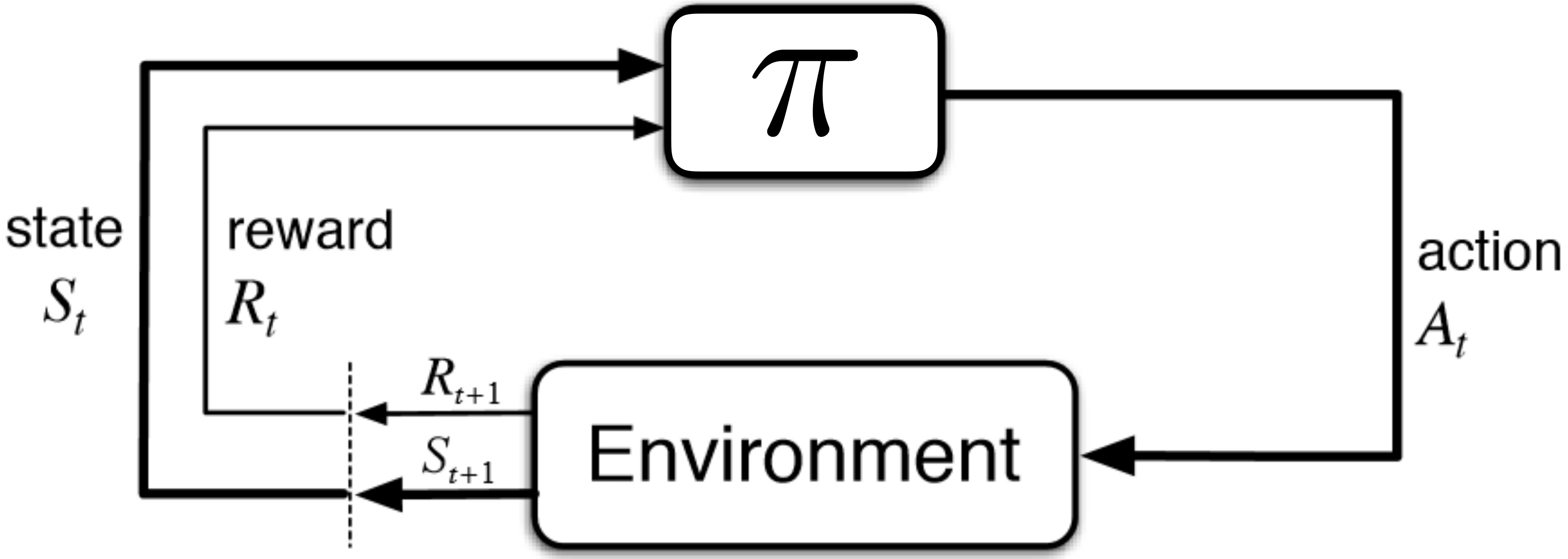
$$S_0, A_0, R_0, S_1$$

The Reinforcement Learning World



$$S_0, A_0, R_0, S_1, \ldots, S_L, A_L, R_L$$

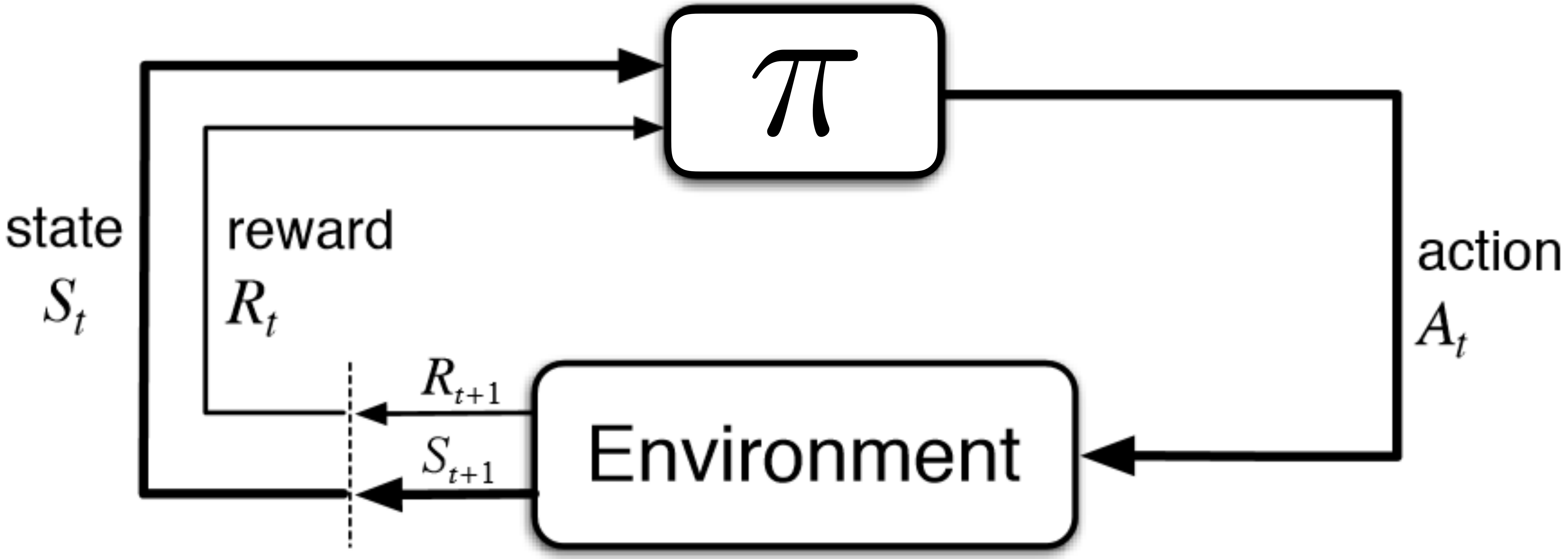
The Reinforcement Learning World



$$\underbrace{S_0, A_0, R_0, S_1, \ldots, S_L, A_L, R_L}_{\texttt{Trajectory}}$$

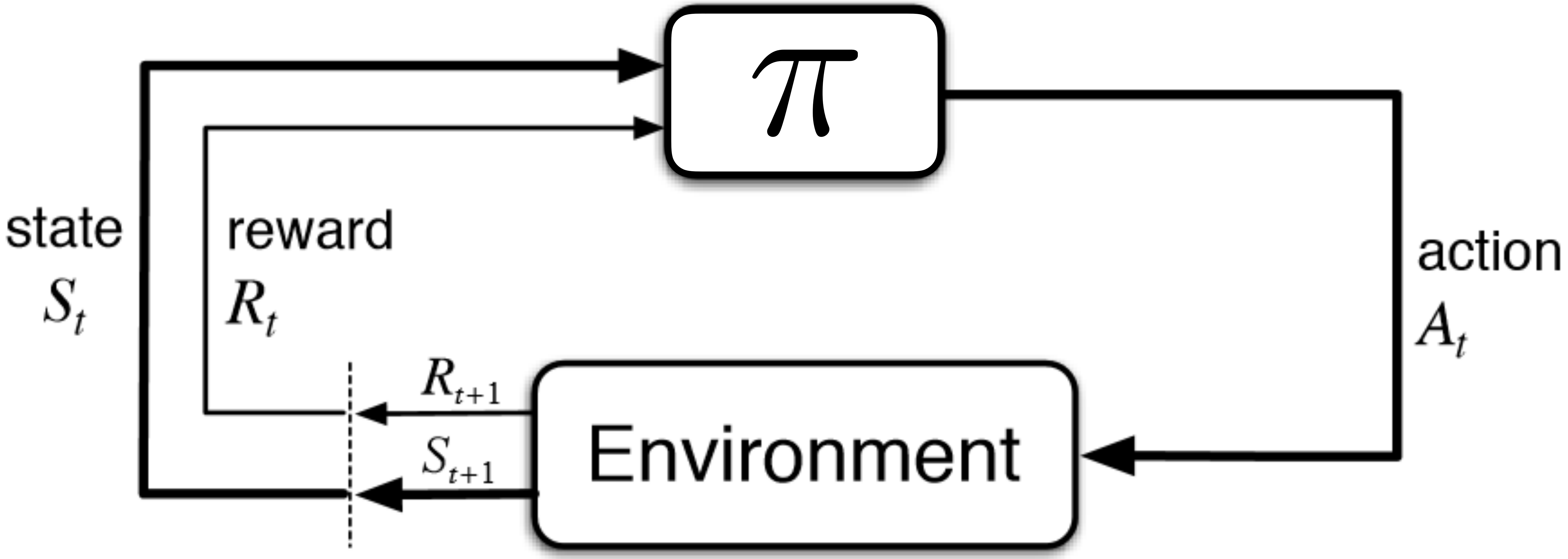
Josiah Hanna

The Reinforcement Learning World



Policy Improvement: Find policy that maximizes expected cumulative reward.

The Reinforcement Learning World



Policy Value Estimation: Given a **fixed** policy, determine the expected cumulative reward of that policy.

Policy Value Estimation

$$v(\pi) = \mathbf{E}_\pi \left[\sum_{t=0}^{L} R_t \right]$$

Josiah Hanna

Policy Value Estimation

$$v\boxed{(\pi)} = \mathbf{E}_\pi \left[\sum_{t=0}^{L} R_t \right]$$

Evaluation Policy

Policy Value Estimation

$$v\boxed{(\pi)} = \mathbf{E}_\pi \left[\sum_{t=0}^{L} R_t \right]$$

Evaluation Policy

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

Policy Value Estimation

$$v\boxed{(\pi)} = \boxed{\mathbf{E}_\pi \left[\sum_{t=0}^{L} R_t \right]}$$

Evaluation Policy

Expected Total Reward

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

On-Policy Monte Carlo Value Estimation

On-Policy Monte Carlo Value Estimation

1. Repeatedly run the evaluation policy.

$$S_0, A_0, R_0, \ldots, S_L, A_L, R_L$$

On-Policy Monte Carlo Value Estimation

1. Repeatedly run the evaluation policy.

$$S_0, A_0, R_0, \ldots, S_L, A_L, R_L$$

2. Average the total reward seen each trajectory.

$$\hat{v} = \frac{1}{m} \sum_{j=1}^{m} \sum_{t=0}^{L} R_t^{(j)}$$

On-Policy Monte Carlo Value Estimation

Off-Policy Value Estimation via Importance Sampling

Precup, Sutton, and Singh (ICML 2000)

Off-Policy Value Estimation via Importance Sampling

1. Repeatedly run a different behavior policy.

Precup, Sutton, and Singh (ICML 2000)

Off-Policy Value Estimation via Importance Sampling

1. Repeatedly run a different behavior policy.

2. Add up all of the reward received along each trajectory.

Precup, Sutton, and Singh (ICML 2000)

Off-Policy Value Estimation via Importance Sampling

1. Repeatedly run a different behavior policy.

2. Add up all of the reward received along each trajectory.

3. Re-weight the reward total.

Precup, Sutton, and Singh (ICML 2000)

Off-Policy Value Estimation via Importance Sampling

1. Repeatedly run a different behavior policy.

2. Add up all of the reward received along each trajectory.

3. Re-weight the reward total.

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Precup, Sutton, and Singh (ICML 2000)

Off-Policy Value Estimation via Importance Sampling

1. Repeatedly run a different behavior policy.

2. Add up all of the reward received along each trajectory.

3. Re-weight the reward total.

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \boxed{\left(\sum_{t=0}^{L} R_t\right)}$$

Total Reward

Precup, Sutton, and Singh (ICML 2000)

Off-Policy Value Estimation via Importance Sampling

1. Repeatedly run a different behavior policy.

2. Add up all of the reward received along each trajectory.

3. Re-weight the reward total.

$$\boxed{\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right)} \times \boxed{\left(\sum_{t=0}^{L} R_t\right)}$$

Relative Likelihood **Total Reward**

Precup, Sutton, and Singh (ICML 2000)

Off-Policy Value Estimation via Importance Sampling

1. Repeatedly run a different behavior policy.

2. Add up all of the reward received along each trajectory.

3. Re-weight the reward total.

$$\boxed{\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right)} \times \boxed{\left(\sum_{t=0}^{L} R_t\right)}$$

Relative Likelihood **Total Reward**

4. Average the re-weighted rewards.

Precup, Sutton, and Singh (ICML 2000)

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

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

How to collect off-policy data?

How to collect off-policy data?

How to choose the behavior policy for importance sampling?

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Josiah Hanna

How to collect off-policy data?

How to choose the behavior policy for importance sampling?

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Contribution 1: Formulation of behavior policy search problem and behavior policy gradient algorithm for policy value estimation.

How to collect off-policy data?

How to choose the behavior policy for importance sampling?

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Contribution 1: Formulation of behavior policy search problem and behavior policy gradient algorithm for policy value estimation.

Contribution 2: Initial study of the behavior policy gradient algorithm combined with policy gradient policy improvement.

Josiah Hanna

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

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

How to weight off-policy data?

How to weight off-policy data?

How to correct for off-policy distribution shift?

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Josiah Hanna

How to weight off-policy data?

How to correct for off-policy distribution shift?

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Contribution 3: Family of regression importance sampling estimators that improve over ordinary importance sampling.

How to weight off-policy data?

How to correct for off-policy distribution shift?

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Contribution 3: Family of regression importance sampling estimators that improve over ordinary importance sampling.

Contribution 4: Sampling error corrected policy gradient estimator that improves over Monte Carlo policy gradient estimators.

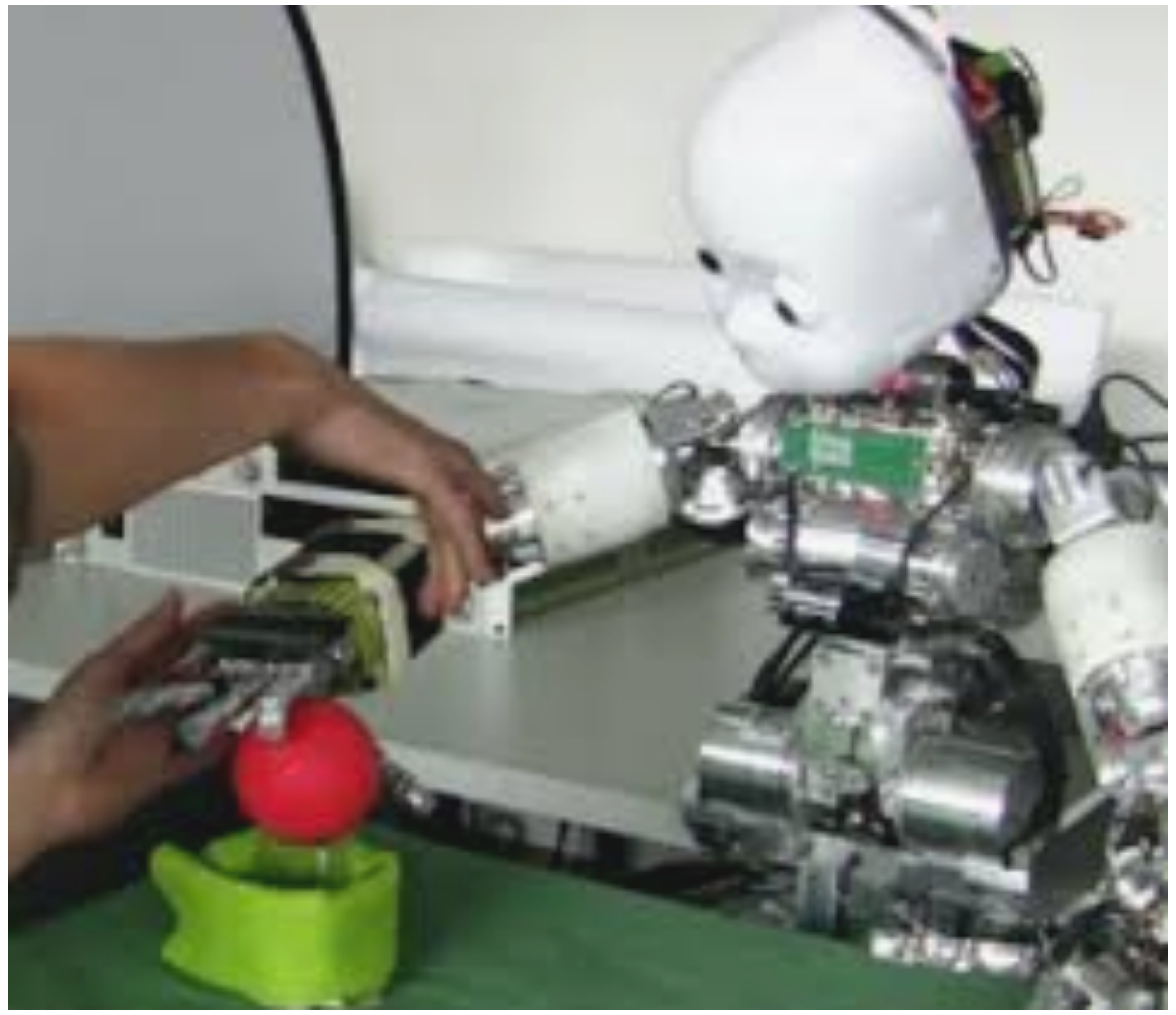
Proposal Time: Importance sampling with an unknown behavior policy

Proposal Time: Importance sampling with an unknown behavior policy

Importance sampling requires the behavior policy probabilities to be known.

$$\frac{\pi(a|s)}{\pi_b(a|s)}$$

Proposal Time: Importance sampling with an unknown behavior policy

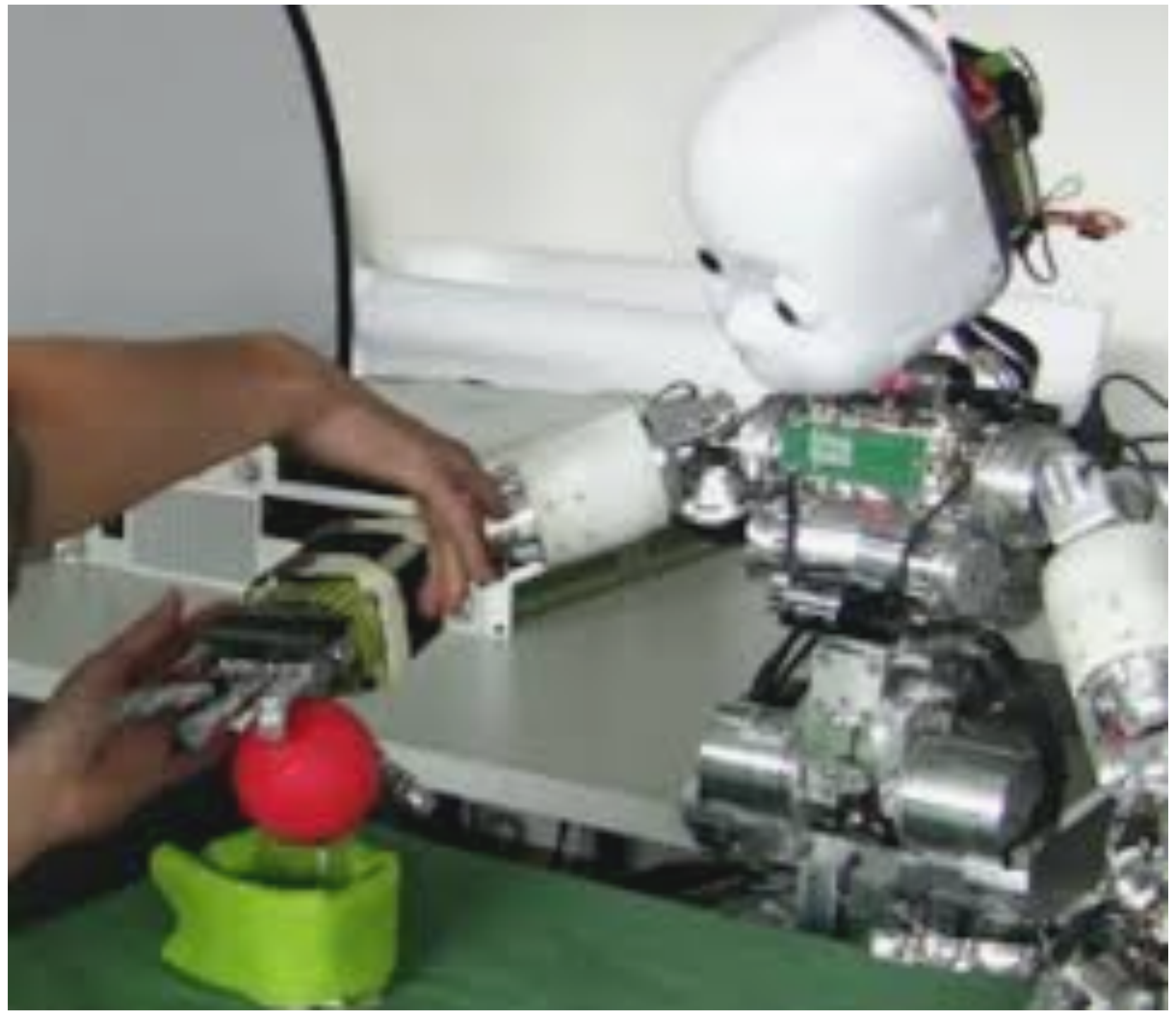


Credit: Brenna Argall

Importance sampling requires the behavior policy probabilities to be known.

$$\frac{\pi(a|s)}{\pi_b(a|s)}$$

Proposal Time: Importance sampling with an unknown behavior policy

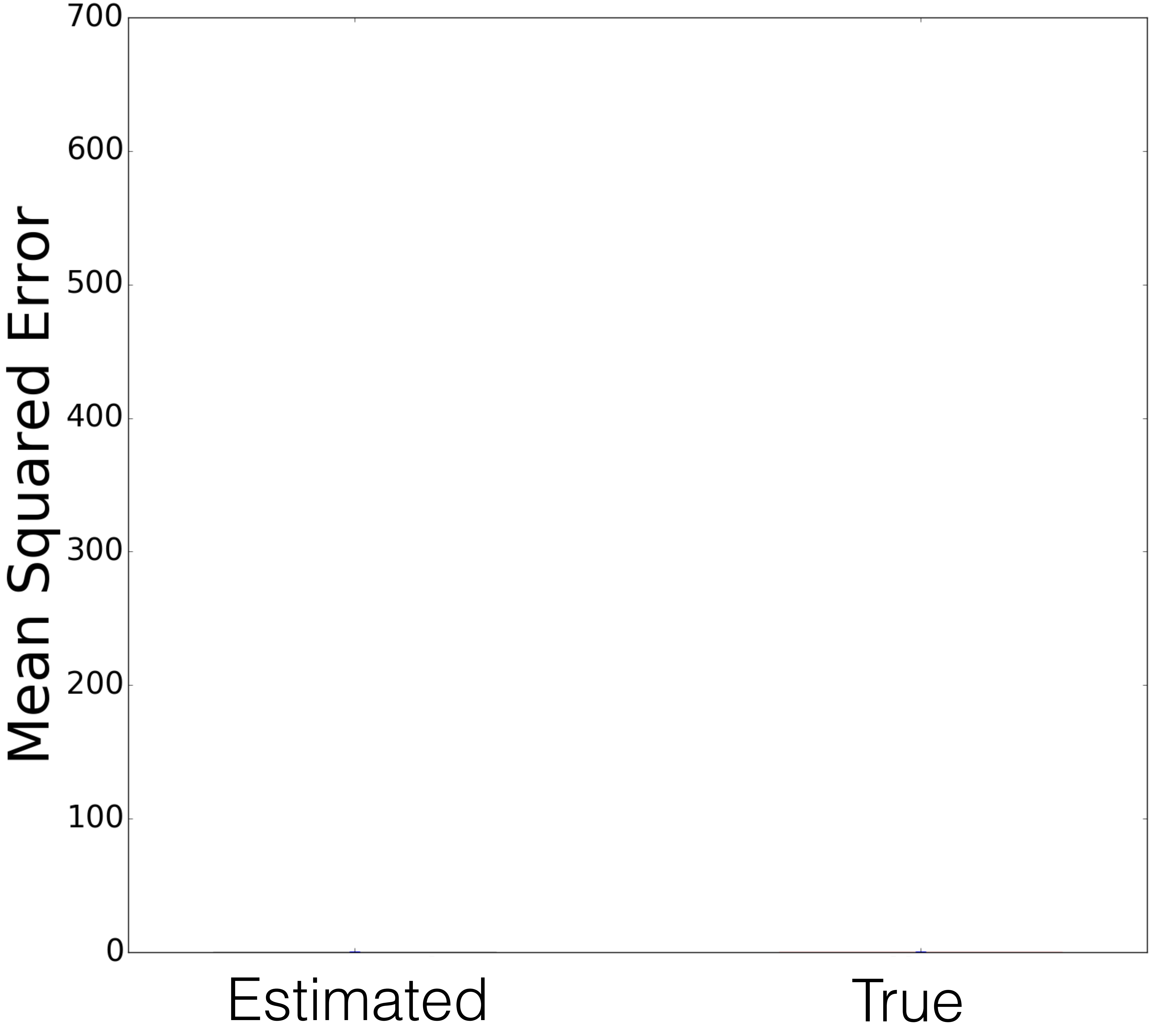


Credit: Brenna Argall

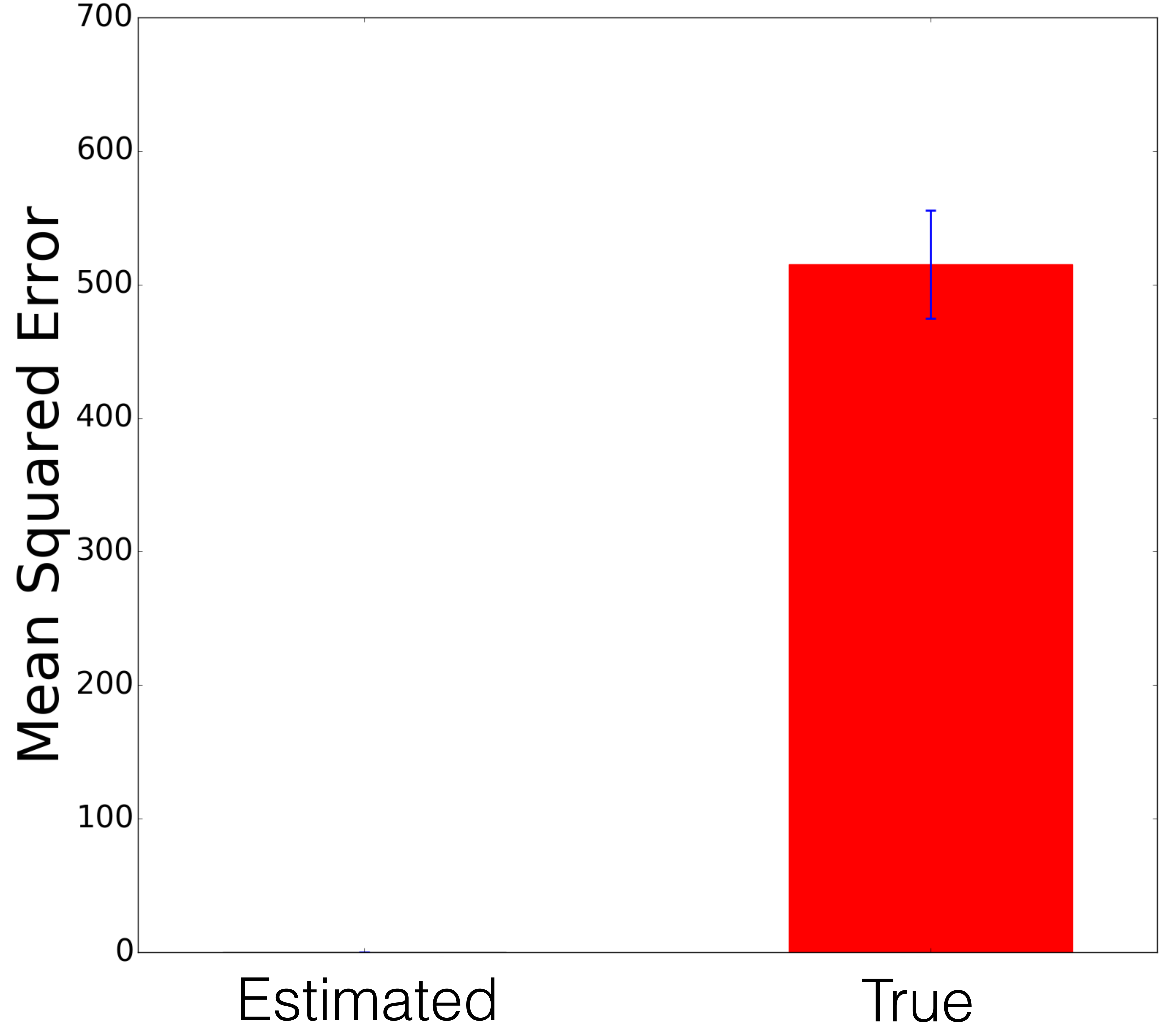
Importance sampling requires the behavior policy probabilities to be known.

$$\frac{\pi(a|s)}{\pi_b(a|s)} \rightarrow \frac{\pi(a|s)}{\pi_{\mathcal{D}}(a|s)}$$

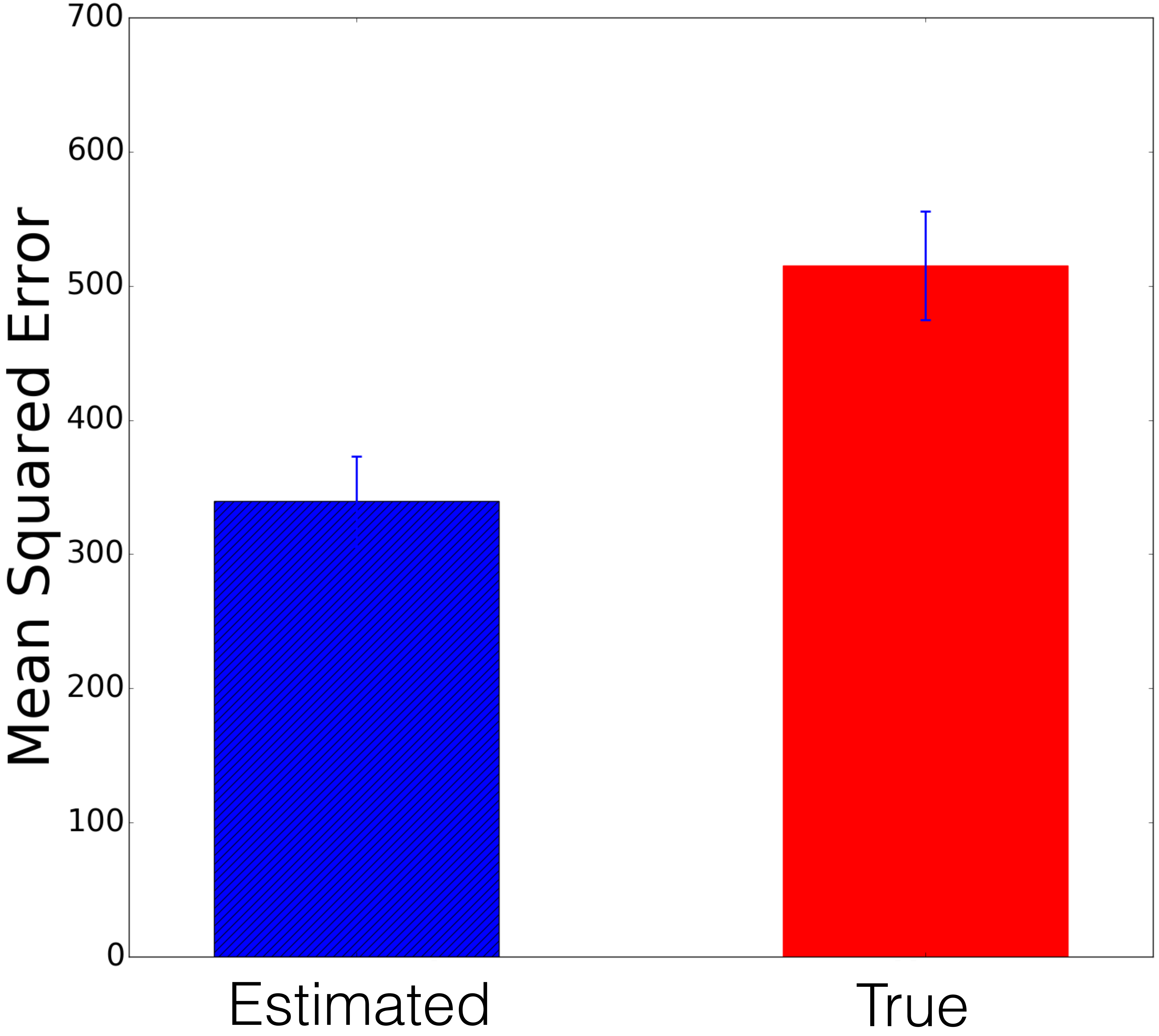
Baseline approach: maximum likelihood behavior policy estimation.



OpenAI's RoboschoolHopper-v1



OpenAI's RoboschoolHopper-v1



OpenAI's RoboschoolHopper-v1

Policy Value Estimation

Policy Value Estimation

Given batch of trajectory data:

$$\mathcal{D} = \{(S_0^i, A_0^i, R_0^i, ..., S_L^i, A_L^i, R_L^i)\}_{i=1}^m$$

Policy Value Estimation

Given batch of trajectory data:

$$\mathcal{D} = \{(S_0^i, A_0^i, R_0^i, ..., S_L^i, A_L^i, R_L^i)\}_{i=1}^m$$

Given an evaluation policy:

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

Josiah Hanna

Policy Value Estimation

Given batch of trajectory data:

$$\mathcal{D} = \{(S_0^i, A_0^i, R_0^i, ..., S_L^i, A_L^i, R_L^i)\}_{i=1}^m$$

Given an evaluation policy:

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

Estimate:

$$v(\pi) := \mathbf{E}\left[\sum_{t=0}^{L} \gamma^t R_t\right]$$

Josiah Hanna

Ordinary Importance Sampling

Precup, Sutton, and Singh (ICML 2000)

Ordinary Importance Sampling

$$\texttt{OIS}(\pi, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)} \sum_{t=0}^{L} \gamma^t R_t$$

Precup, Sutton, and Singh (ICML 2000)

Ordinary Importance Sampling

$$\texttt{OIS}(\pi, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)} \boxed{\sum_{t=0}^{L} \gamma^t R_t}$$

Discounted sum of rewards

Precup, Sutton, and Singh (ICML 2000)

Ordinary Importance Sampling

$$\texttt{OIS}(\pi, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \boxed{\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)}} \boxed{\sum_{t=0}^{L} \gamma^t R_t}$$

Correction from behavior policy to evaluation policy

Discounted sum of rewards

Precup, Sutton, and Singh (ICML 2000)

Regression Importance Sampling

$$\texttt{RIS}(n)(\pi, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, ..., s_t)} \sum_{t=0}^{L} \gamma^t R_t$$

Regression Importance Sampling

$$\texttt{RIS}(n)(\pi, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\boxed{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, ..., s_t)}} \sum_{t=0}^{L} \gamma^t R_t$$

Maximum likelihood
behavior policy estimate
(empirical policy).

Regression Importance Sampling

$$\texttt{RIS}(n)(\pi, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \boxed{\prod_{t=0}^{L} \frac{\pi(a_t | s_t)}{\pi_{\mathcal{D}}(a_t | s_{t-n}, a_{t-n}, ..., s_t)}} \sum_{t=0}^{L} \gamma^t R_t$$

Correction from empirical policy to evaluation policy.

Related Work

Related Work

1. Estimated Propensity Scores (Hirano et al. 2003, Li et al. 2015).

Related Work

1. Estimated Propensity Scores (Hirano et al. 2003, Li et al. 2015).
2. Learning in contextual bandits (Xie et al. 2019, Narita et al. 2019)

Related Work

1. Estimated Propensity Scores (Hirano et al. 2003, Li et al. 2015).

2. Learning in contextual bandits (Xie et al. 2019, Narita et al. 2019)

We are the first to show using an estimated behavior policy improves importance sampling in multi-step environments.

Regression Importance Sampling

$$\texttt{RIS}(n)(\pi, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \prod_{t=0}^{L} \frac{\pi(a_t | s_t)}{\pi_{\mathcal{D}}(a_t | s_{t-n}, a_{t-n}, ..., s_t)} \sum_{t=0}^{L} \gamma^t R_t$$

Regression Importance Sampling

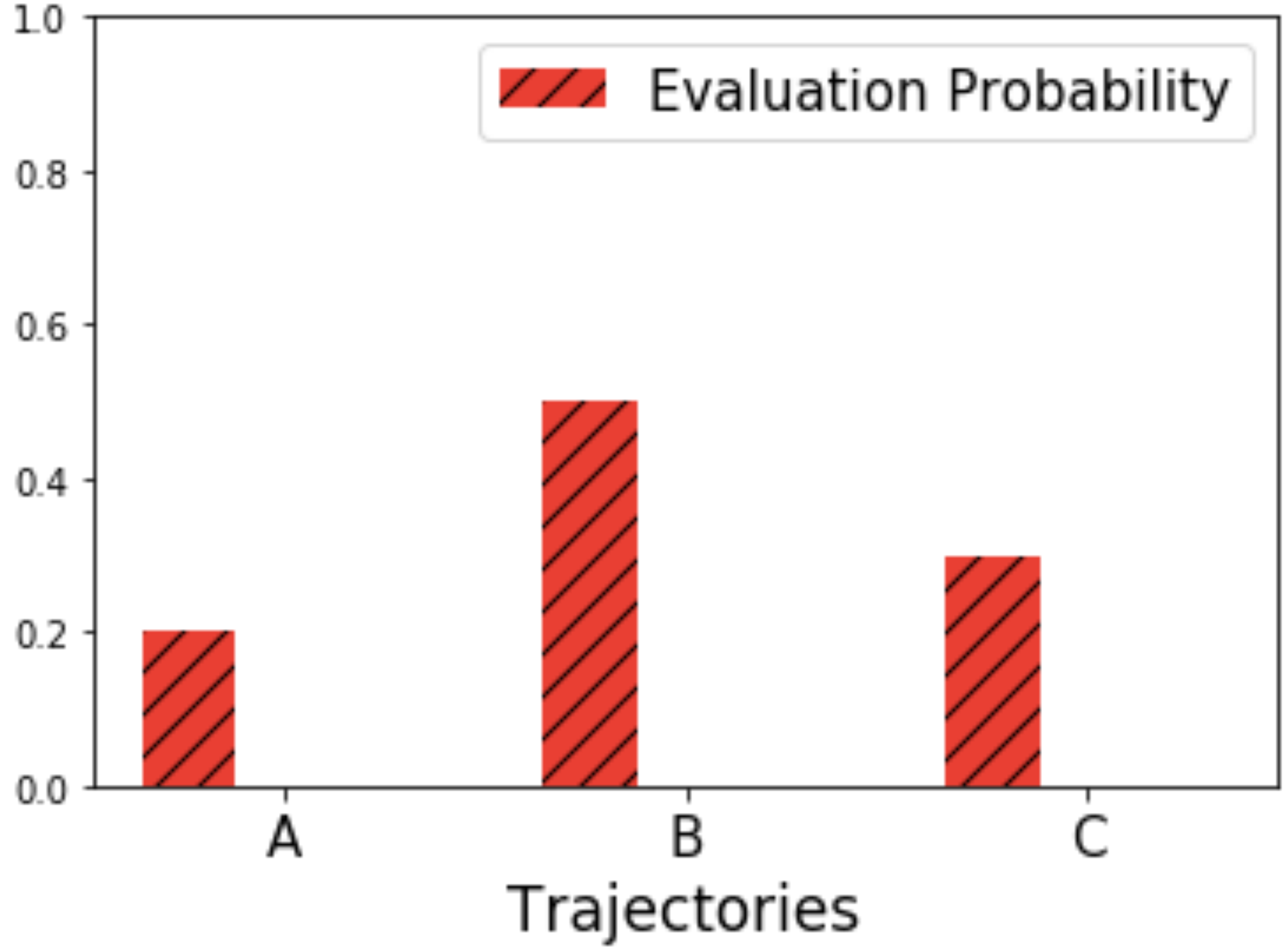
$$\texttt{RIS}(n)(\pi, \mathcal{D}) = \frac{1}{m}\sum_{i=1}^{m} \boxed{\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, \ldots, s_t)}} \sum_{t=0}^{L} \gamma^t R_t$$

Correction from empirical policy to evaluation policy.

Josiah Hanna

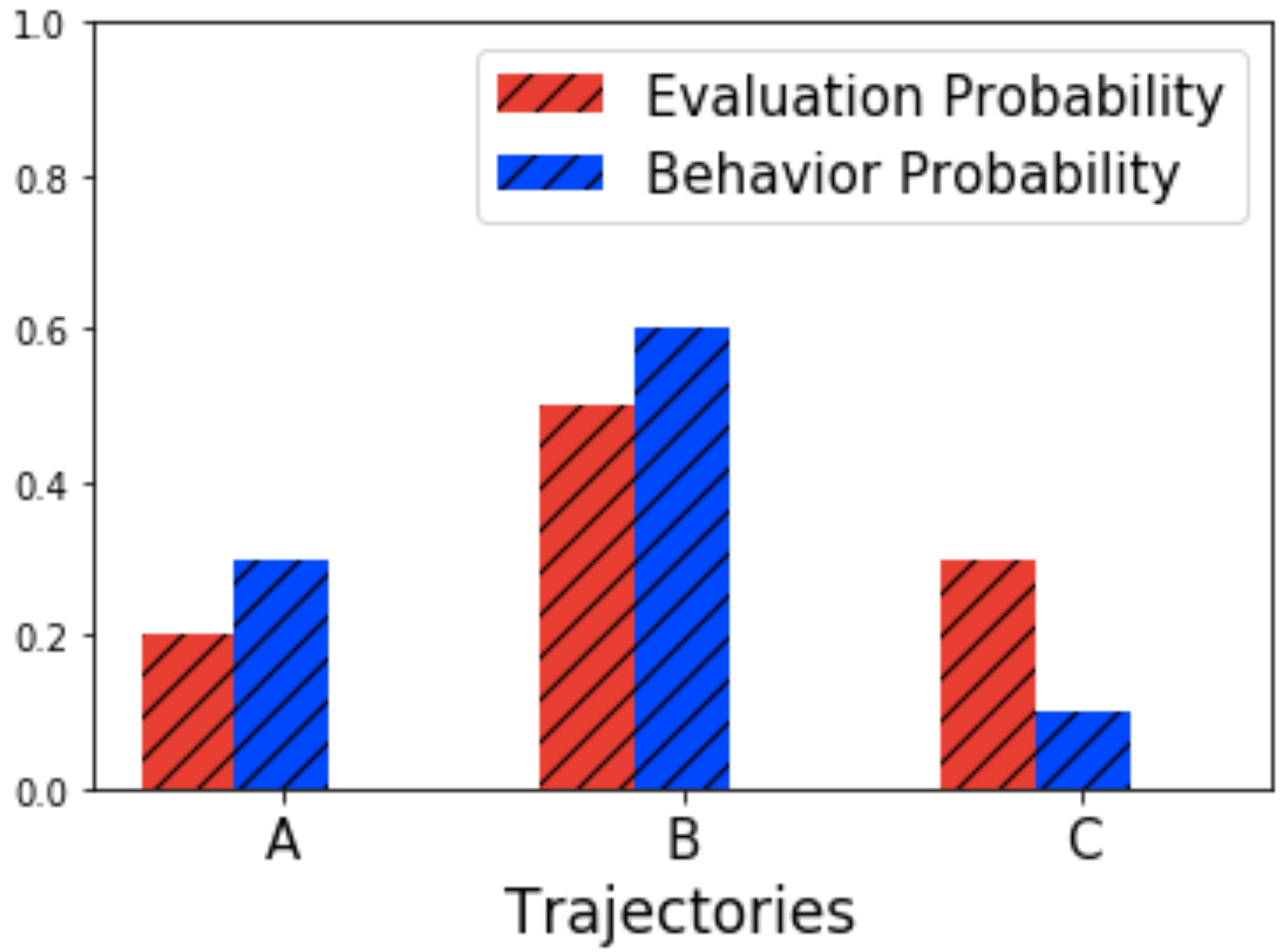
Regression Importance Sampling

Regression Importance Sampling

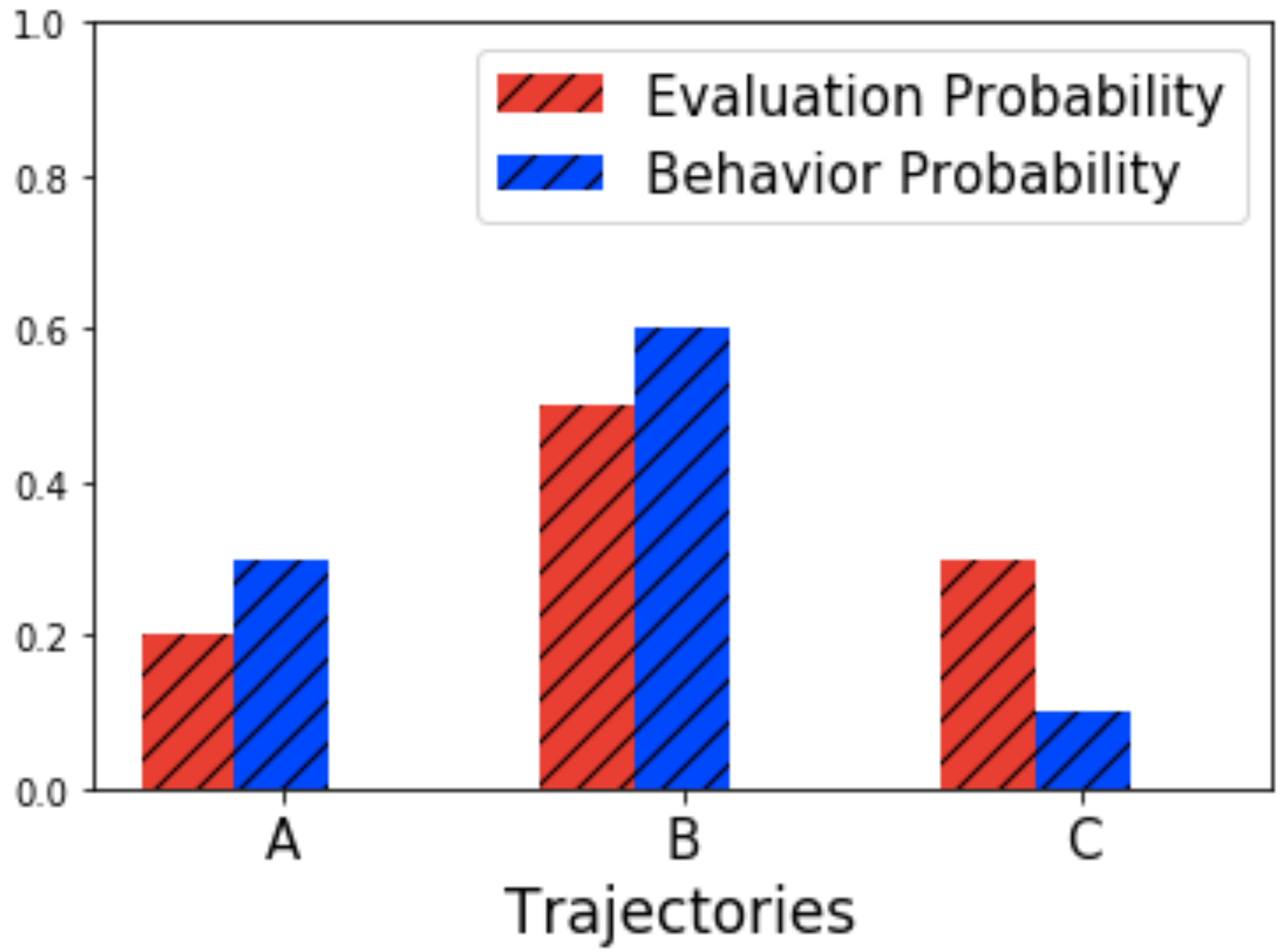


Josiah Hanna

Regression Importance Sampling

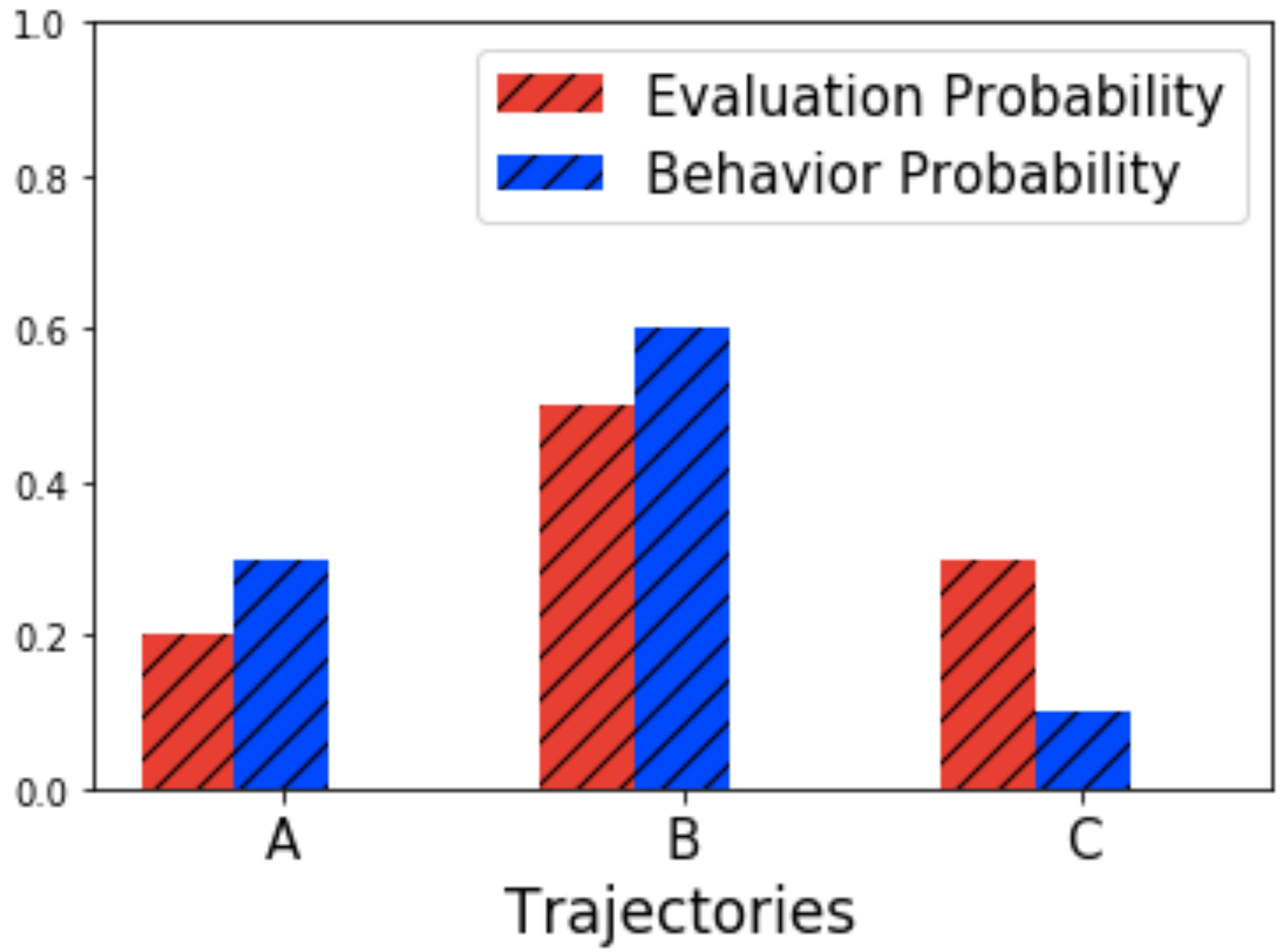


Regression Importance Sampling



$$\frac{\pi(a|s)}{\pi_b(a|s)}$$

Regression Importance Sampling

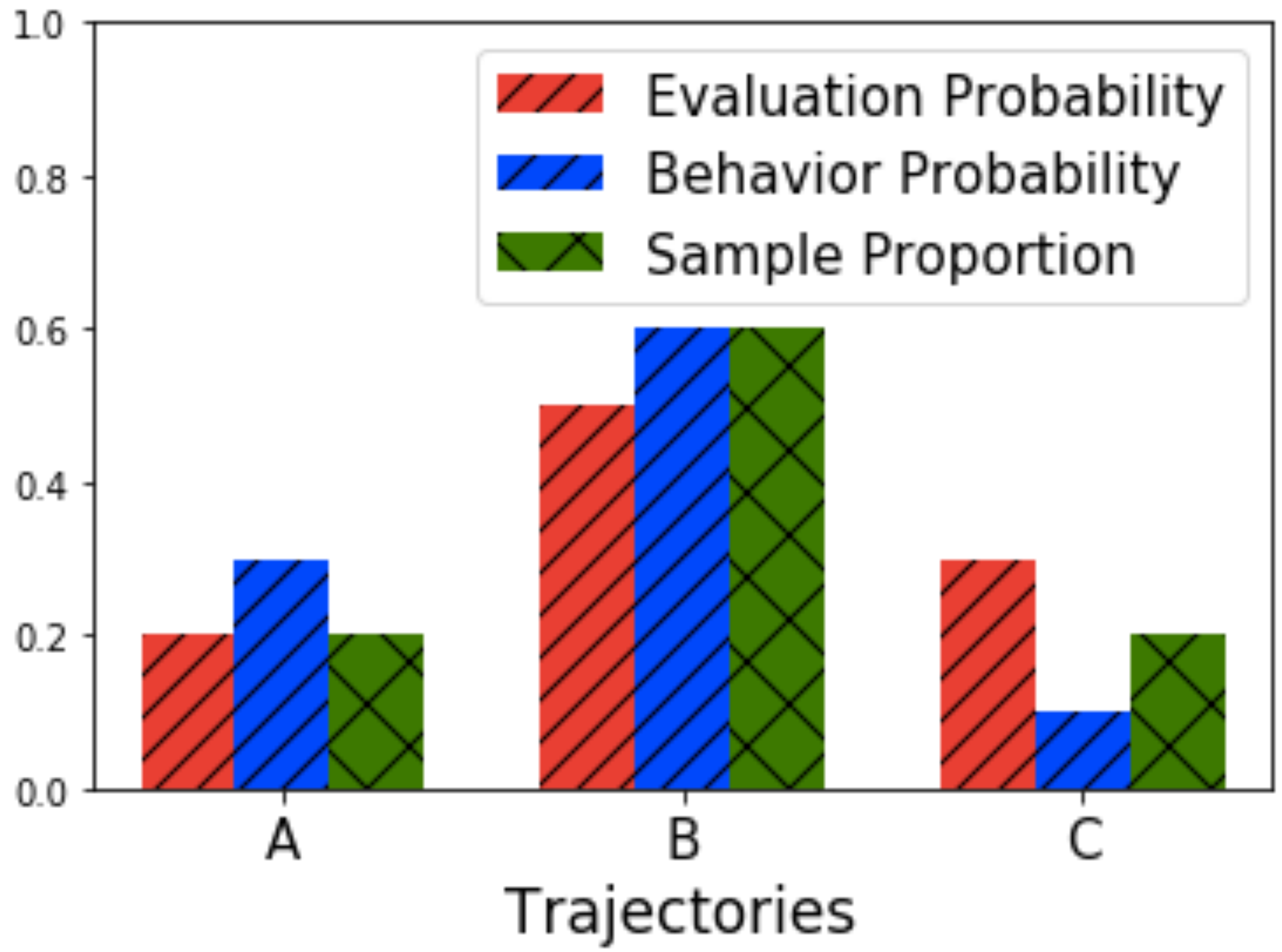


Observed data contains 1 of A, 3 of B, and 1 of C

$$\frac{\pi(a|s)}{\pi_b(a|s)}$$

Josiah Hanna

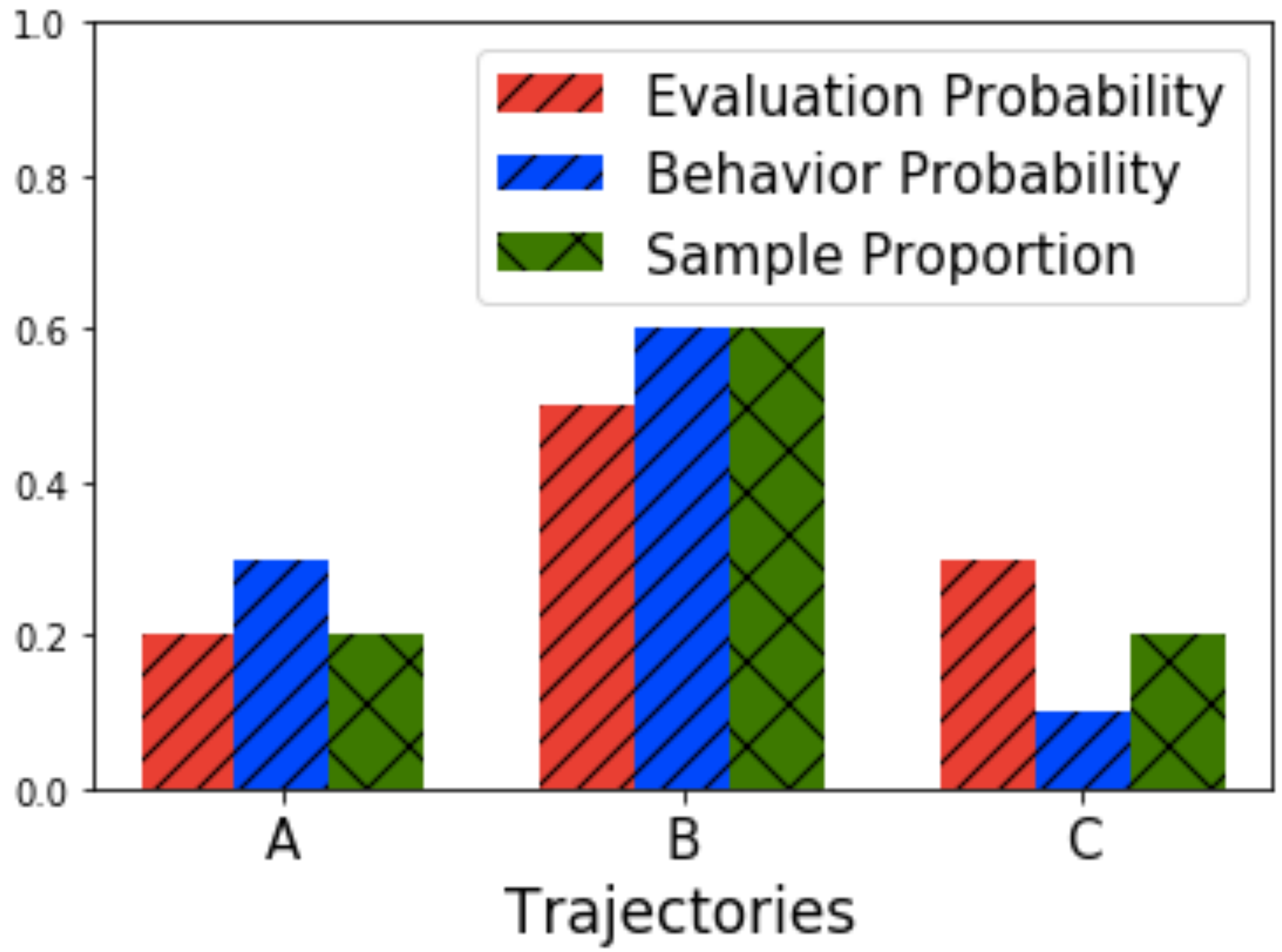
Regression Importance Sampling



$$\frac{\pi(a|s)}{\pi_b(a|s)}$$

Observed data contains 1 of A, 3 of B, and 1 of C

Regression Importance Sampling

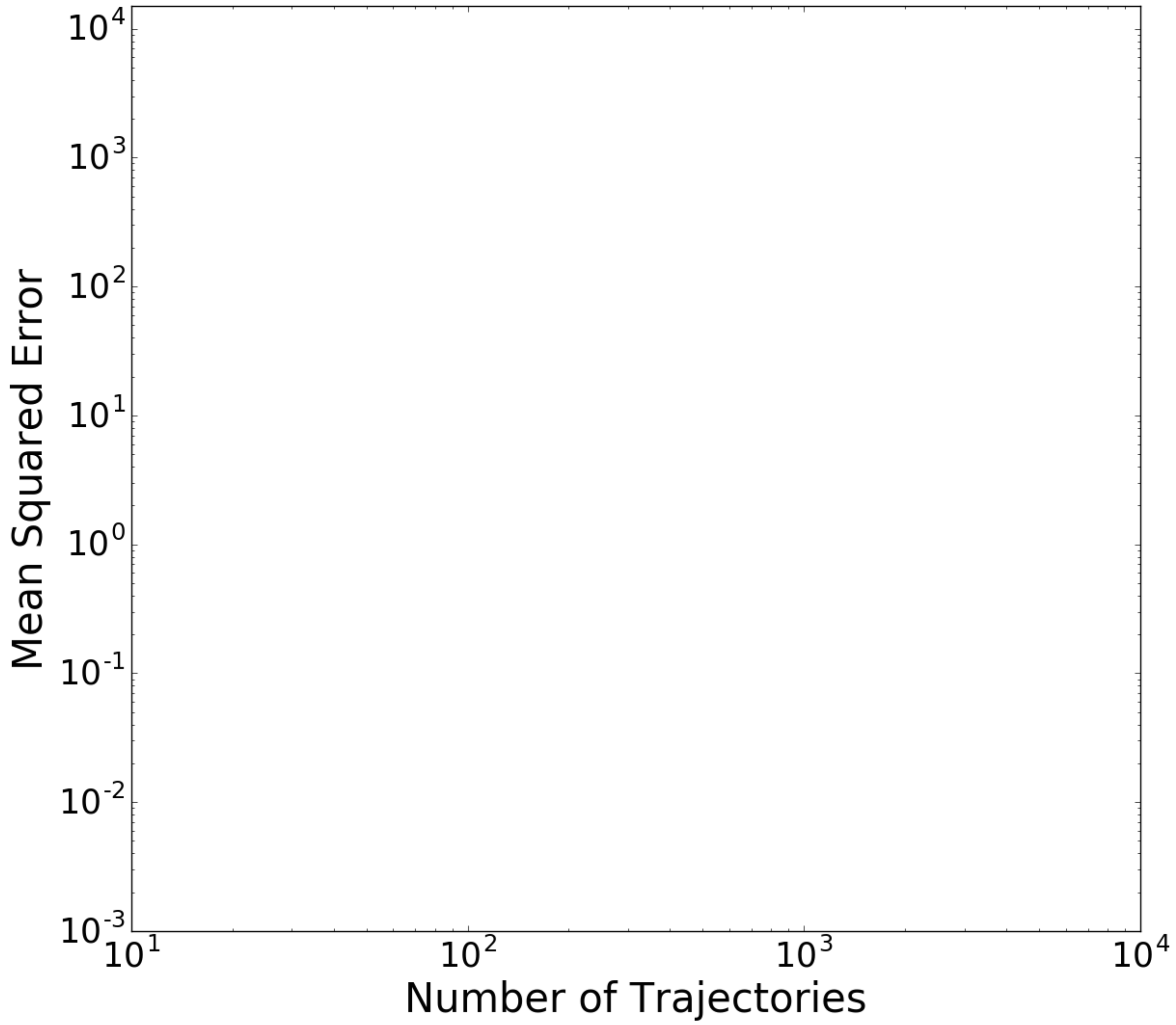


Observed data contains 1 of A, 3 of B, and 1 of C

$$\frac{\pi(a|s)}{\pi_b(a|s)} \to \frac{\pi(a|s)}{\pi_{\mathcal{D}}(a|s)}$$

Josiah Hanna

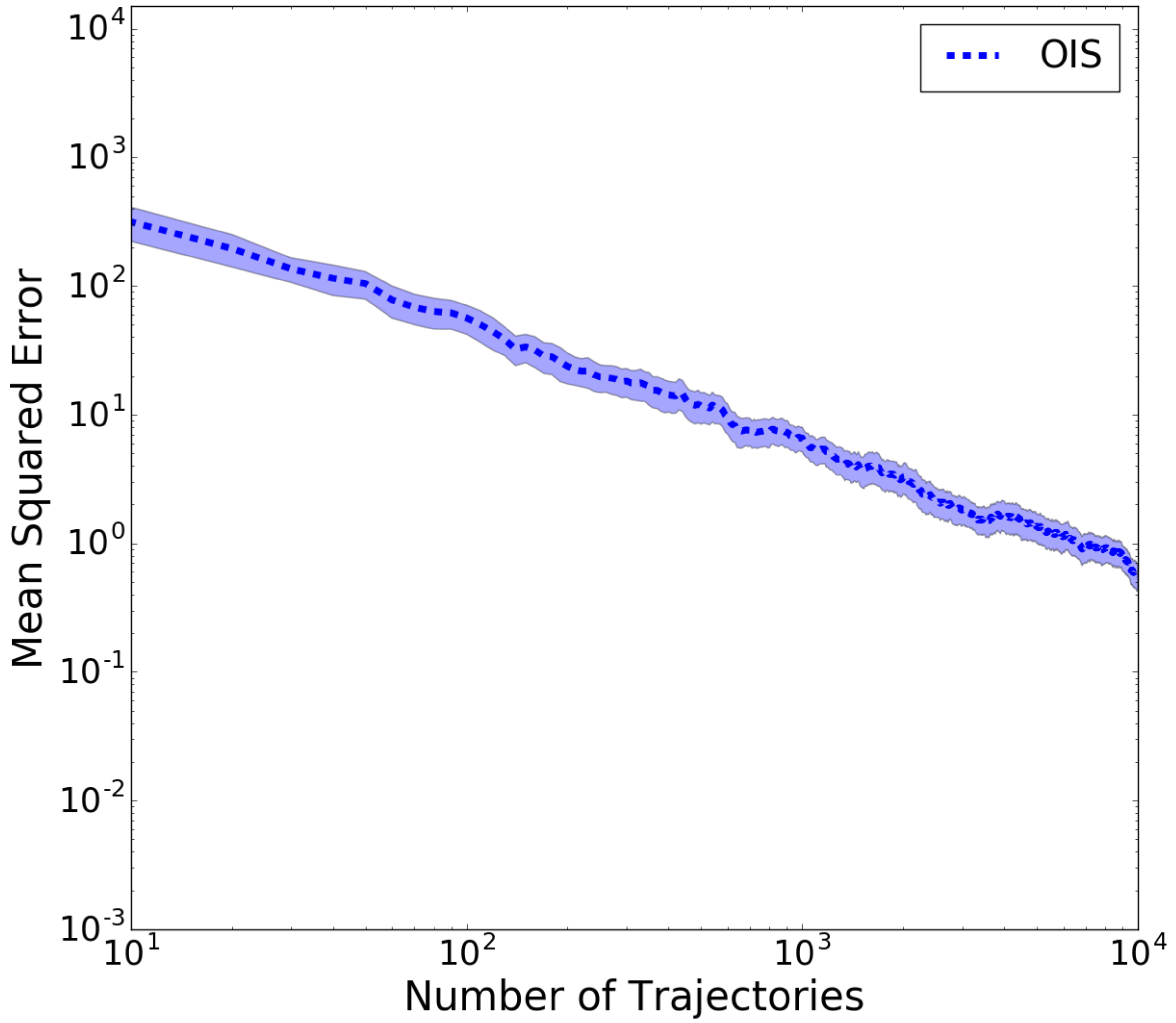
Empirical Results



Gridworld

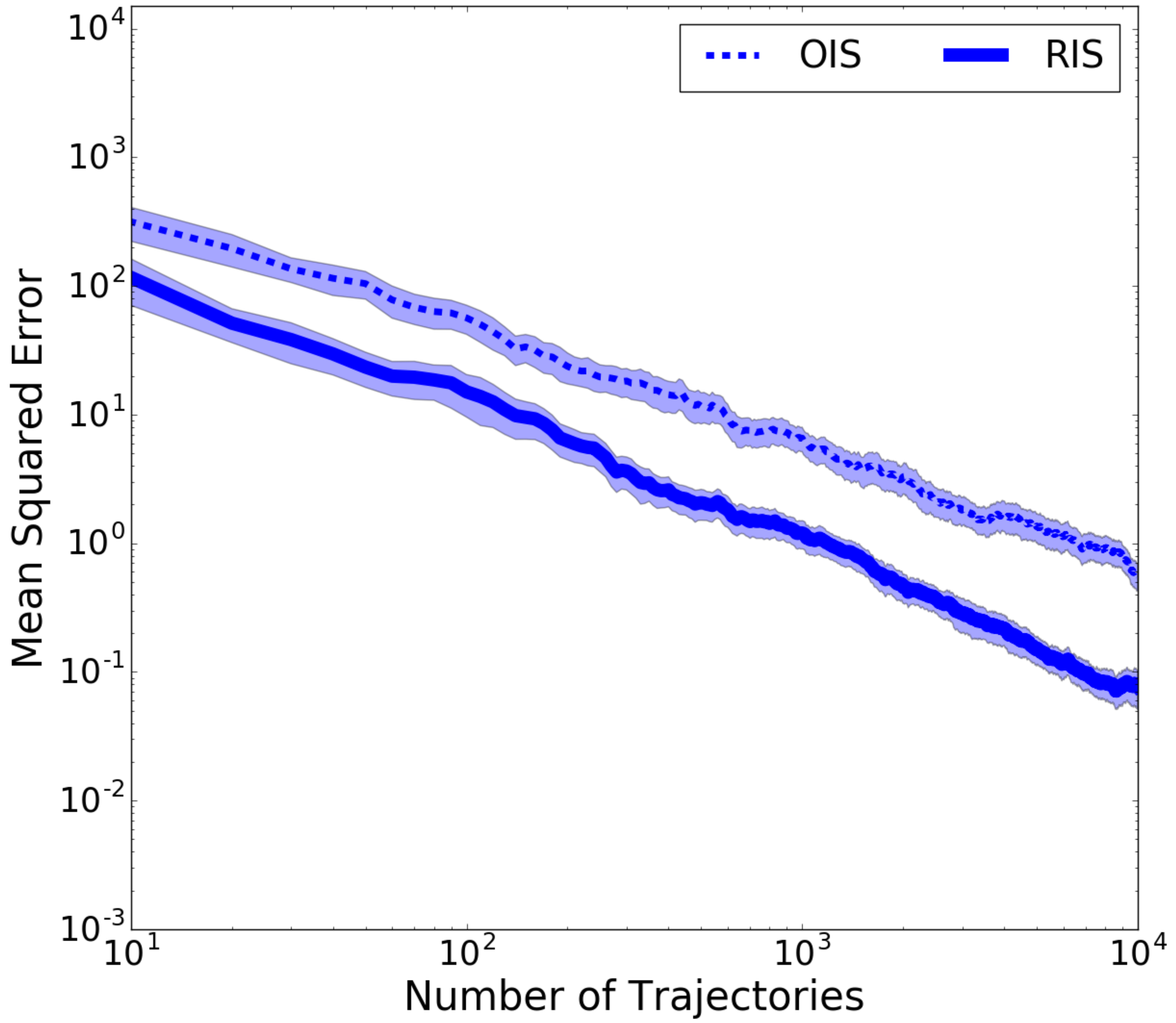
Josiah Hanna

Empirical Results



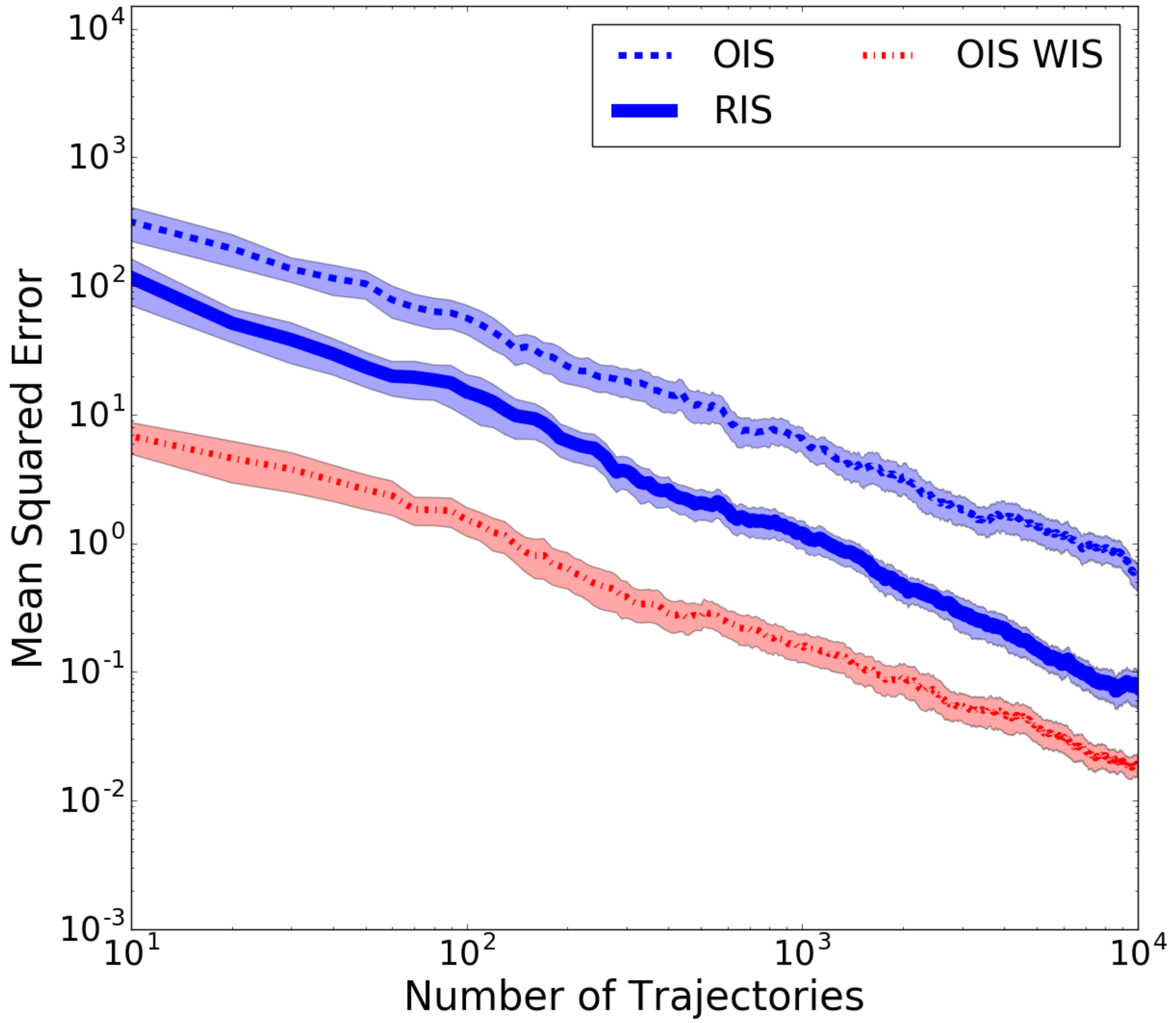
Gridworld

Empirical Results



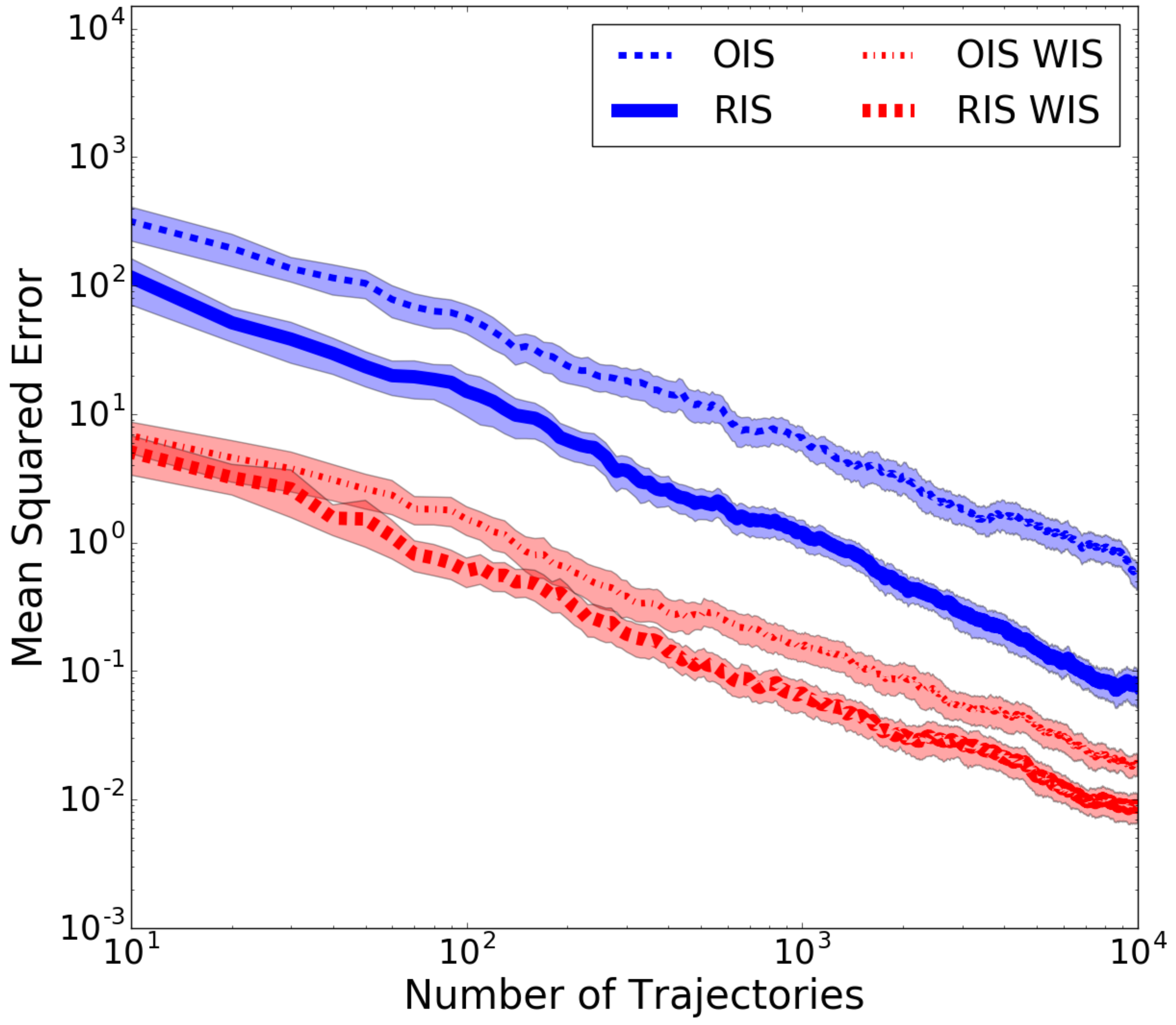
Gridworld

Empirical Results



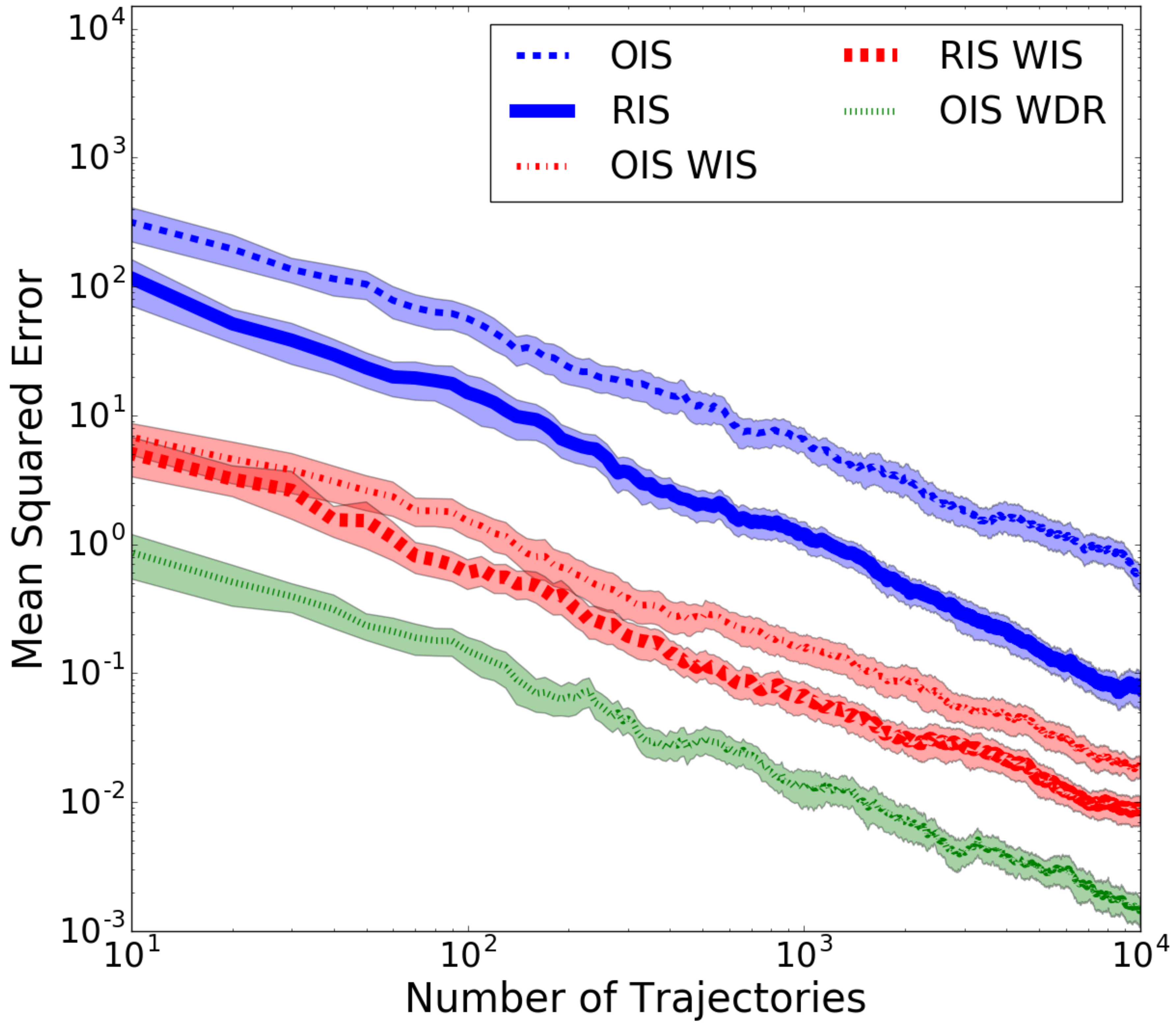
Gridworld

Empirical Results



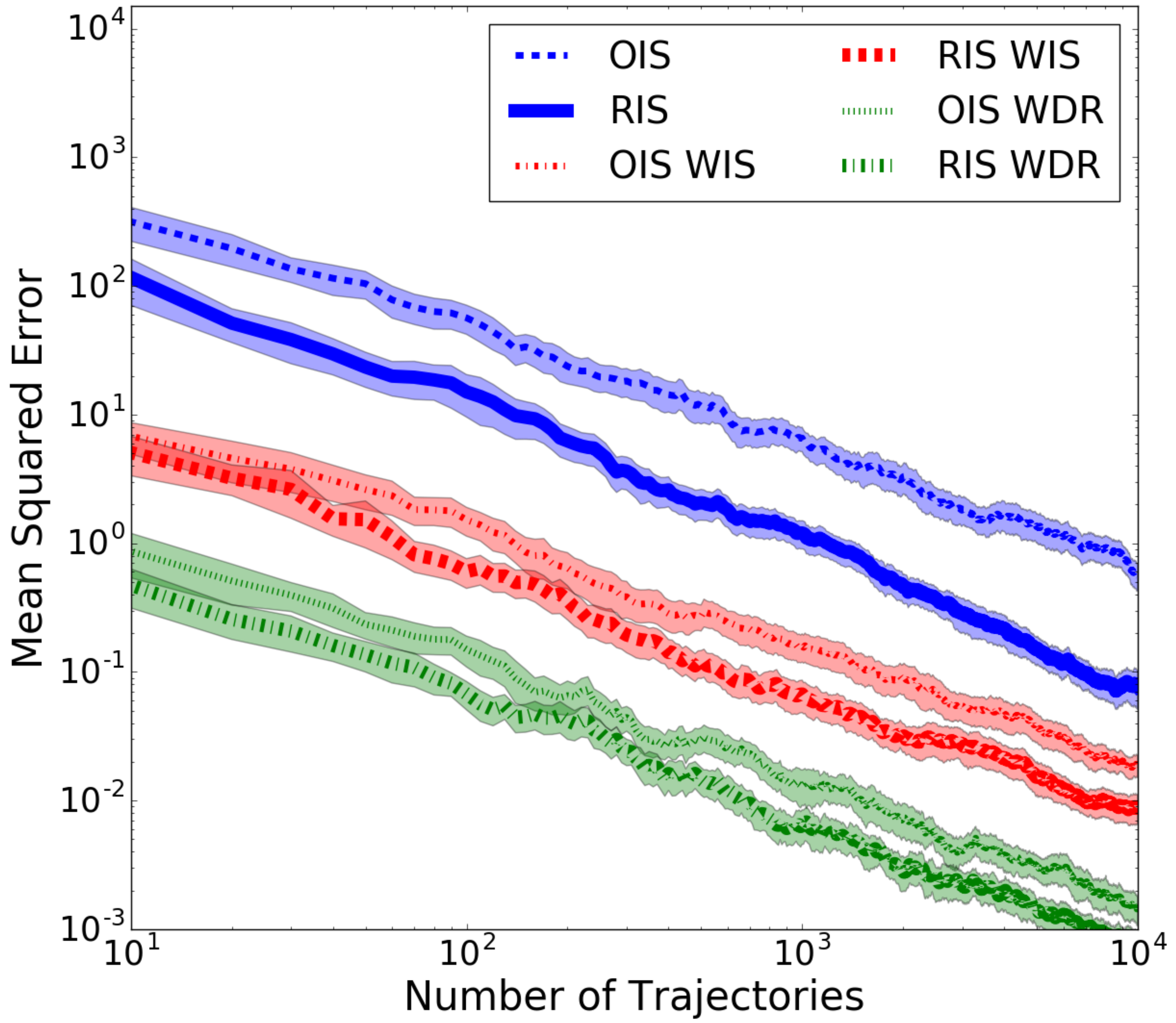
Gridworld

Empirical Results



Gridworld

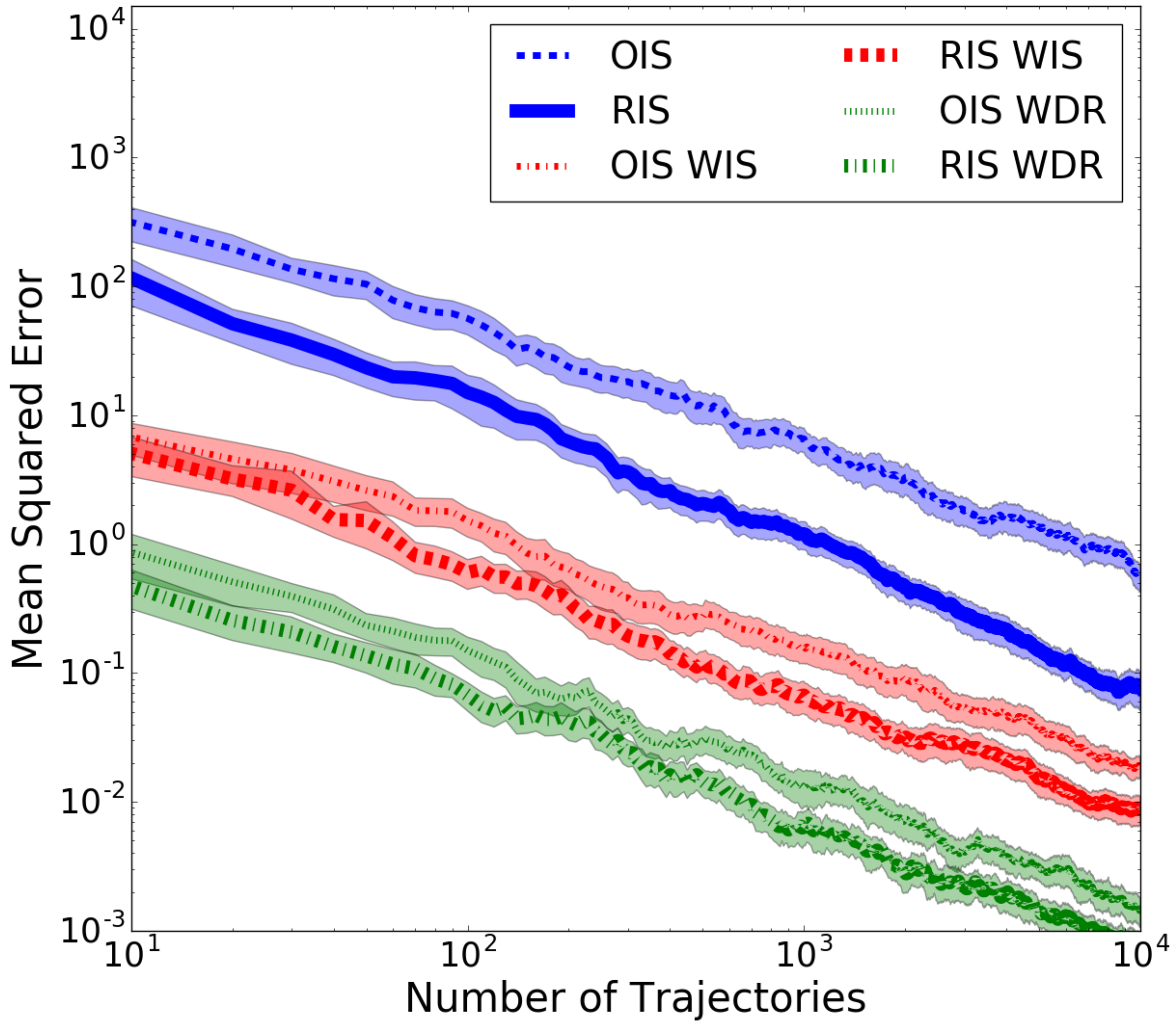
Empirical Results



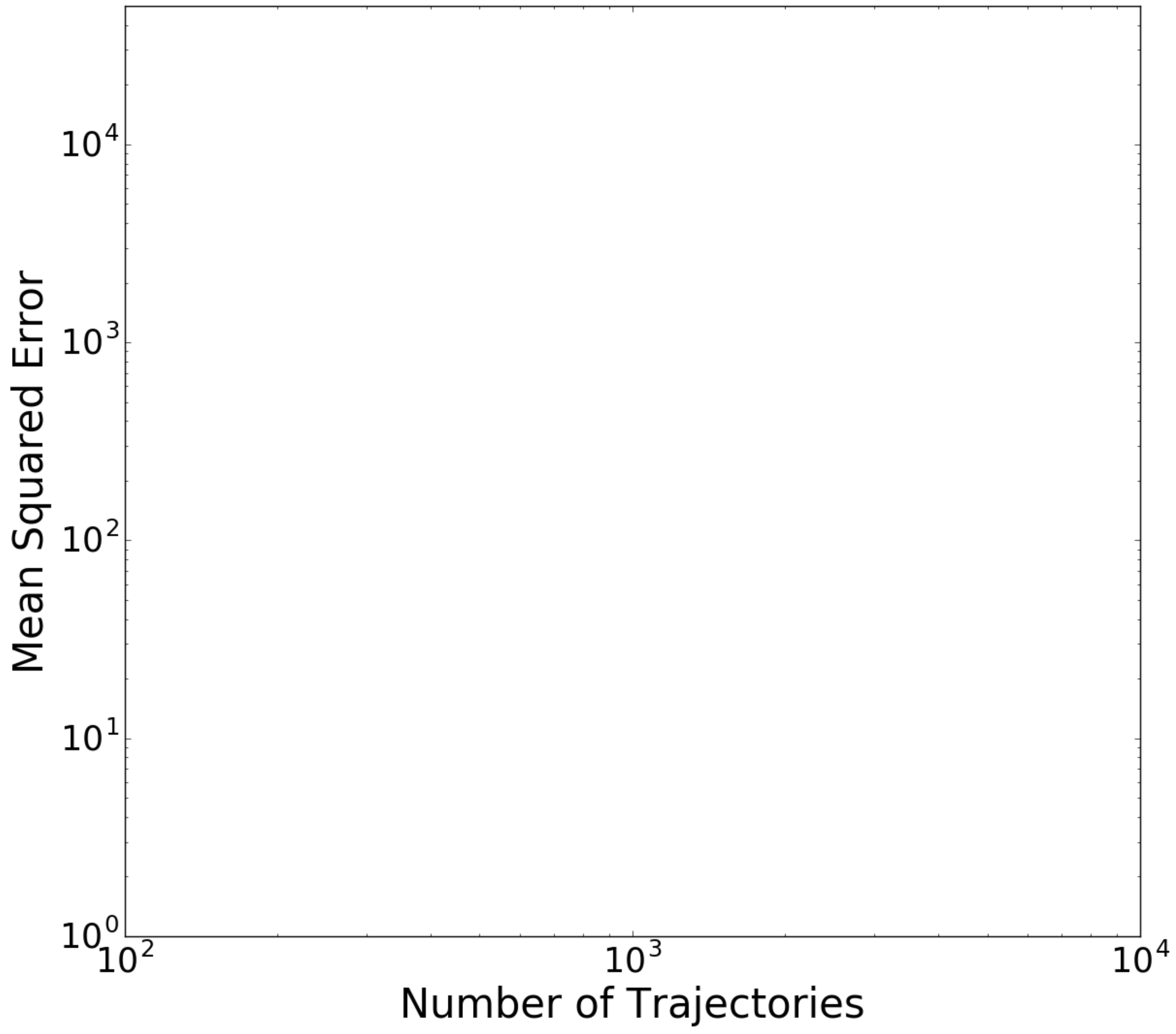
Gridworld

Josiah Hanna

Empirical Results



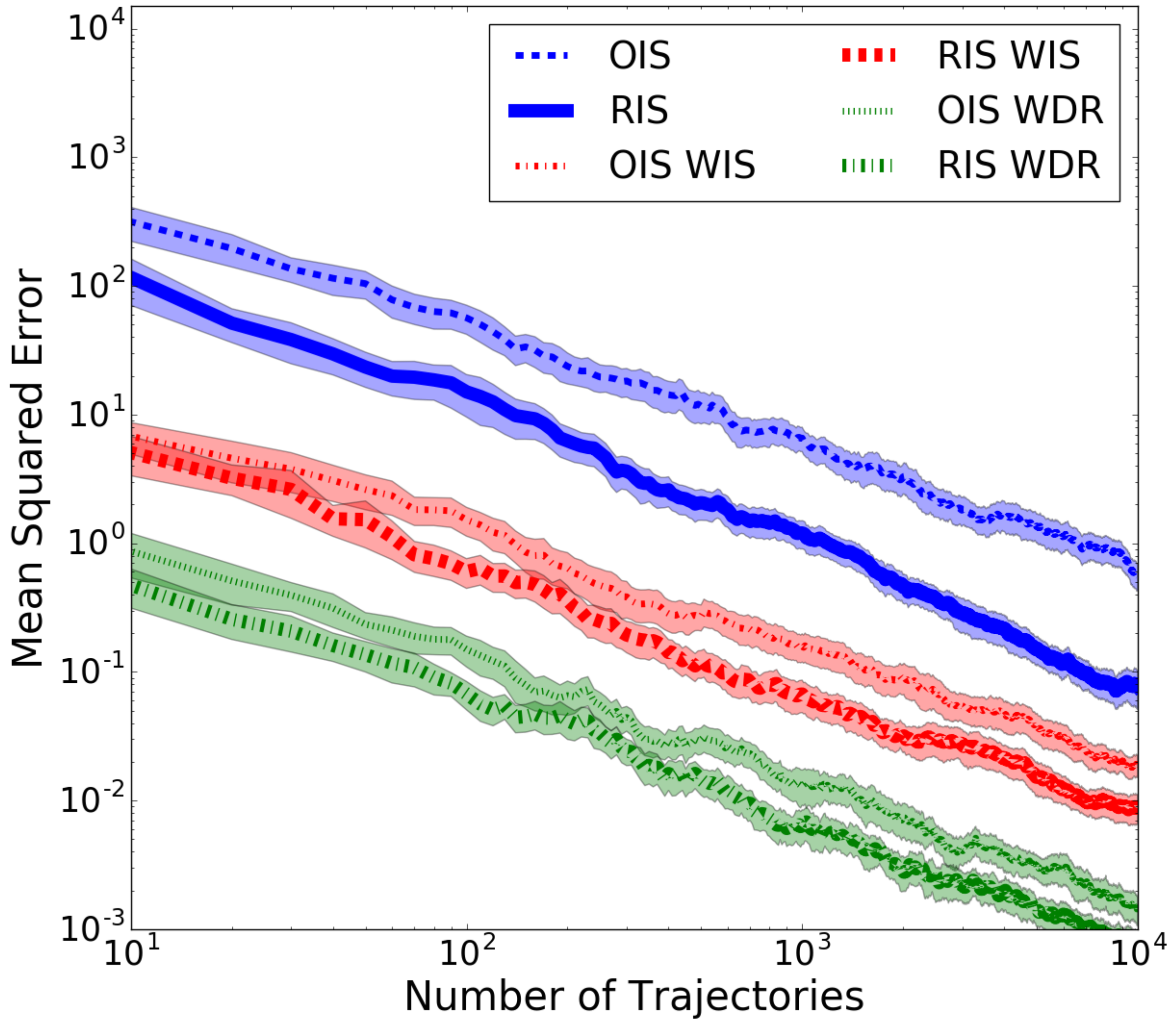
Gridworld



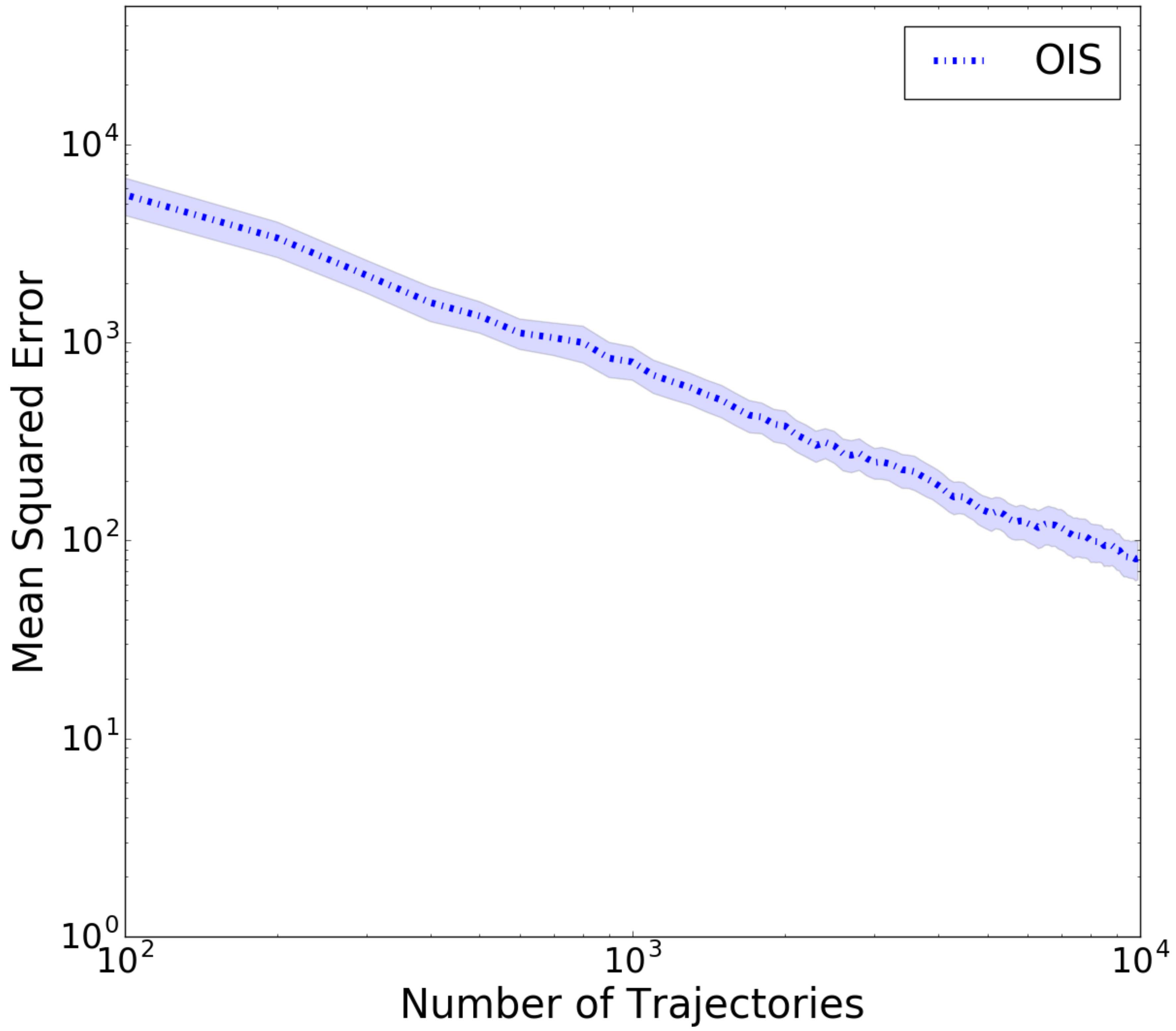
Linear Dynamical System

Josiah Hanna

Empirical Results

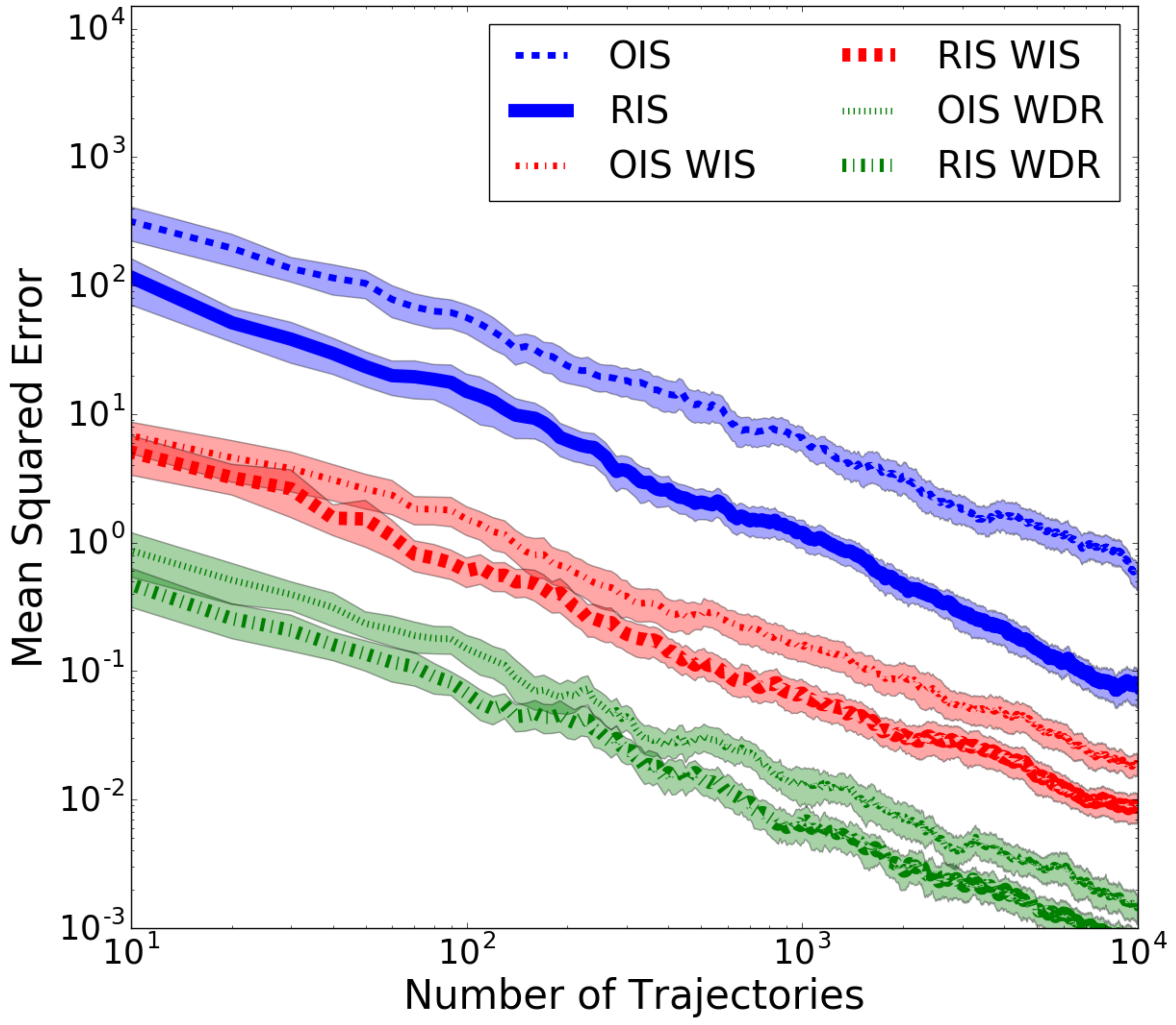


Gridworld

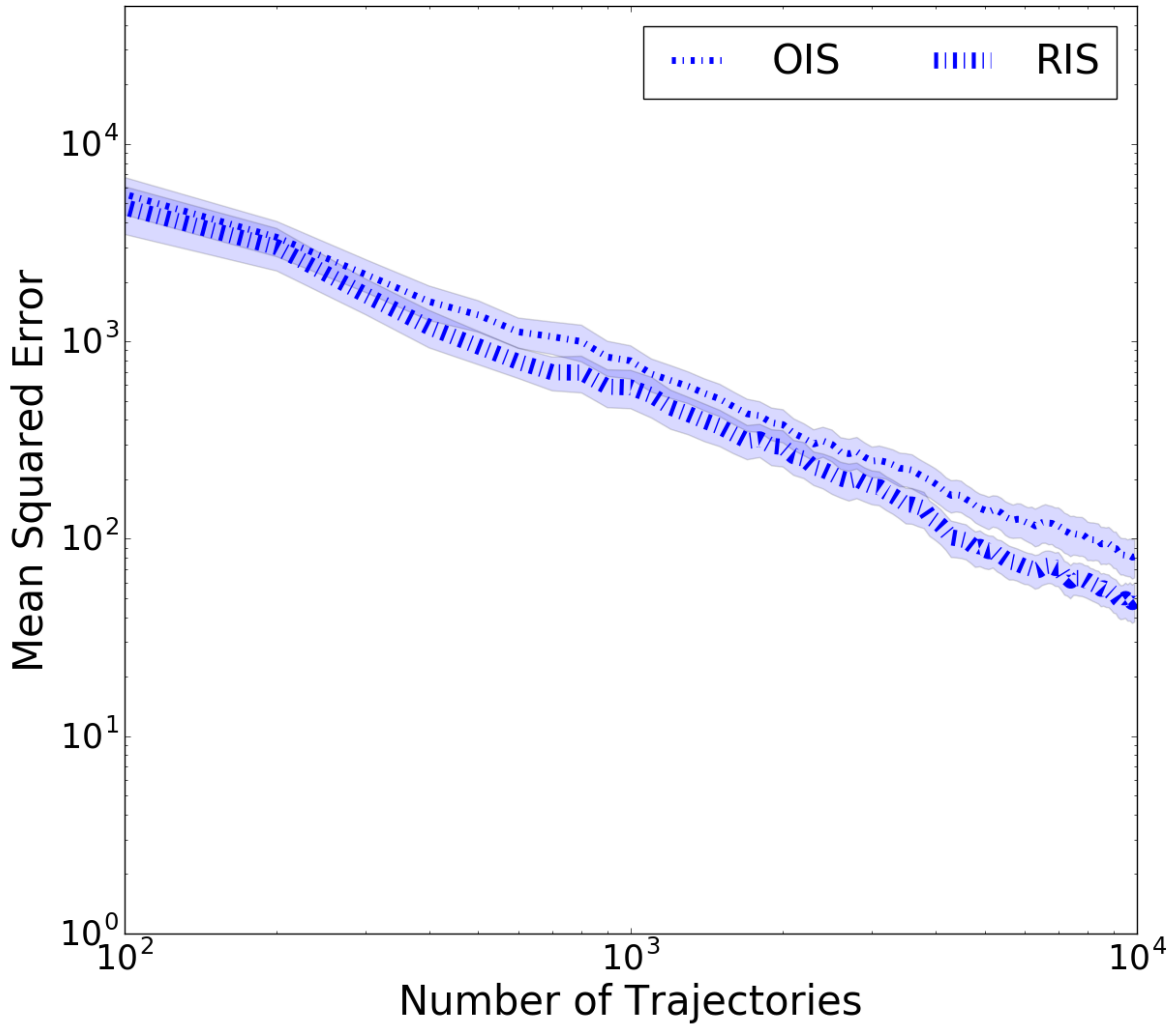


Linear Dynamical System

Empirical Results



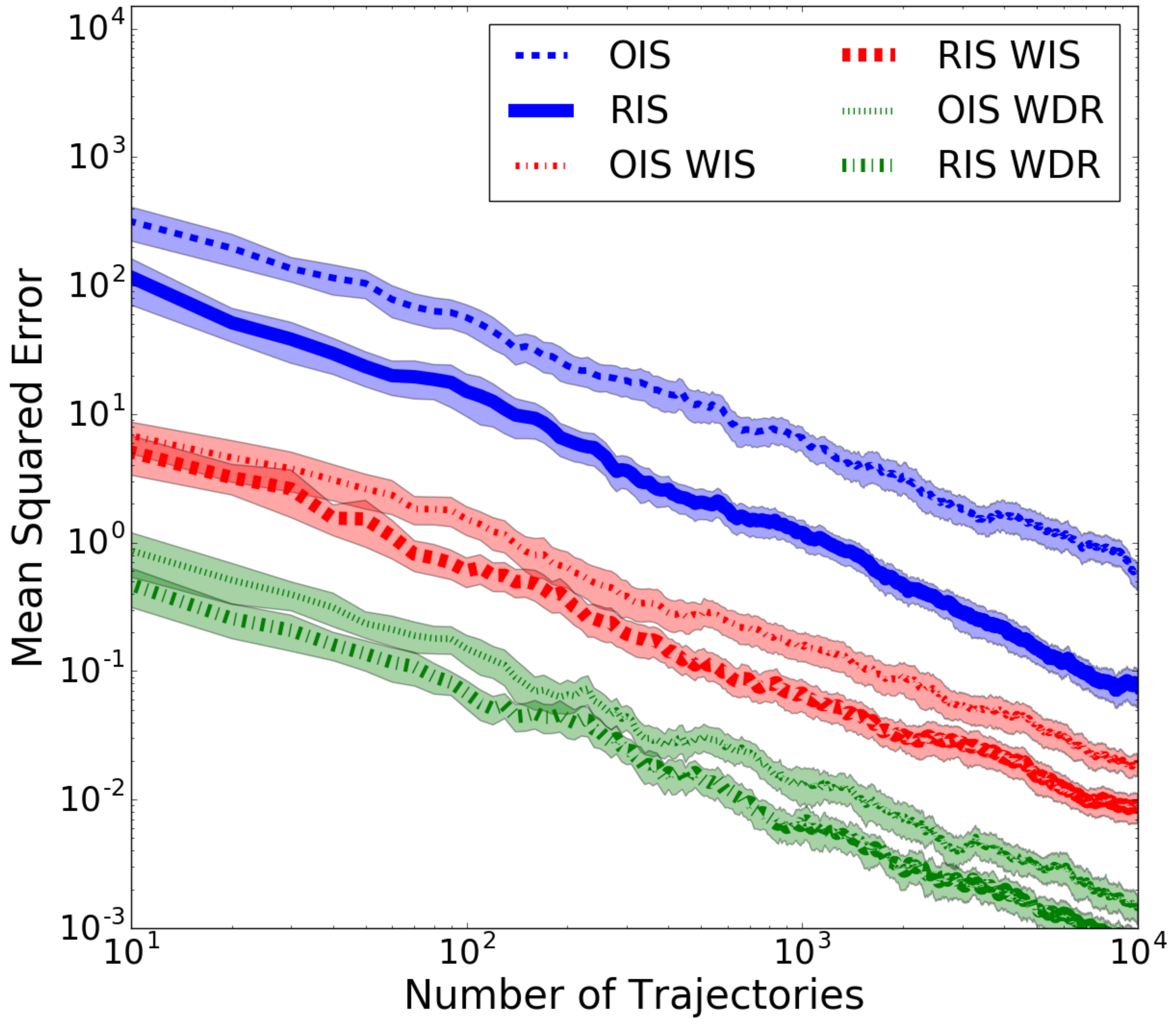
Gridworld



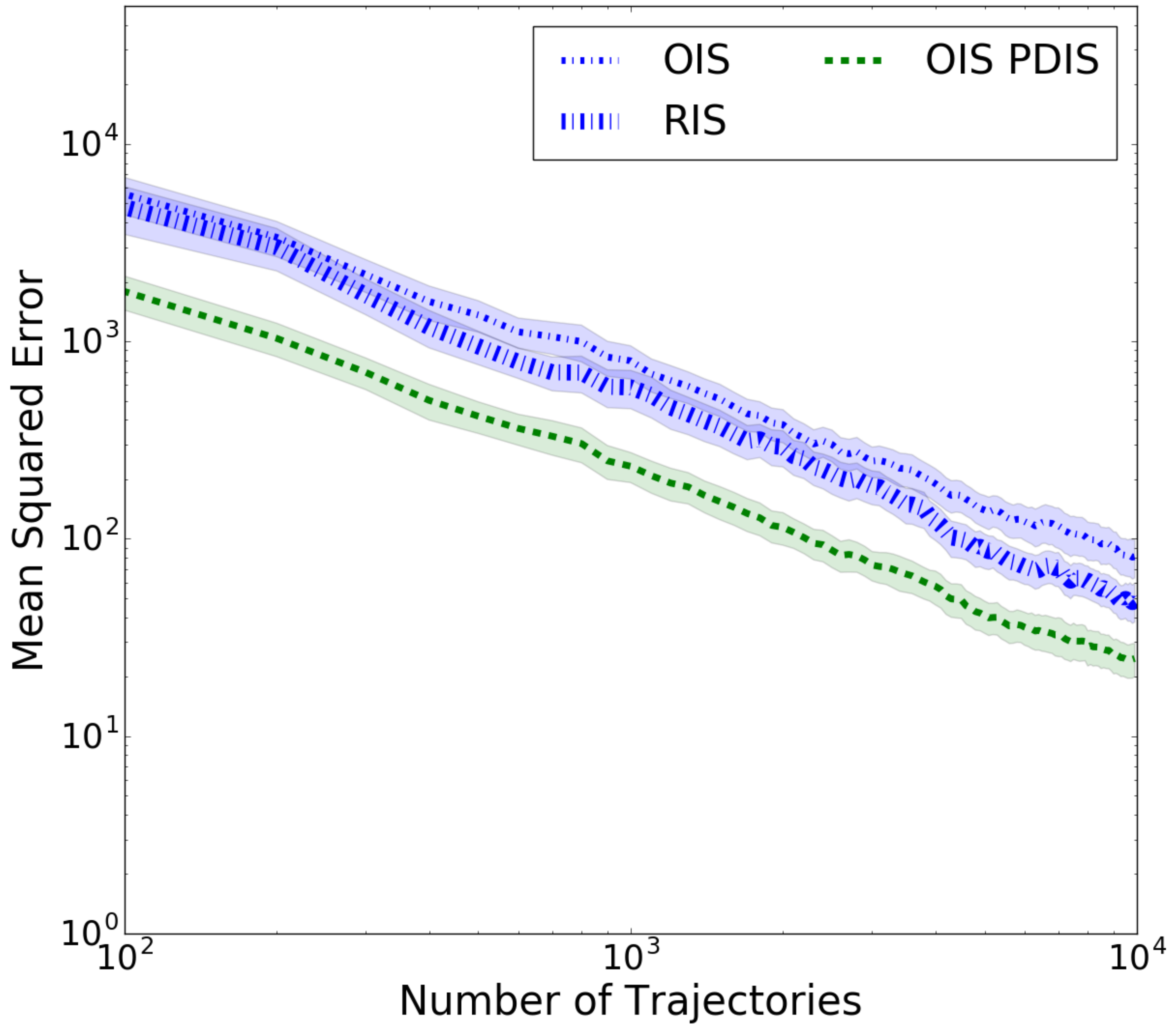
Linear Dynamical System

Josiah Hanna

Empirical Results

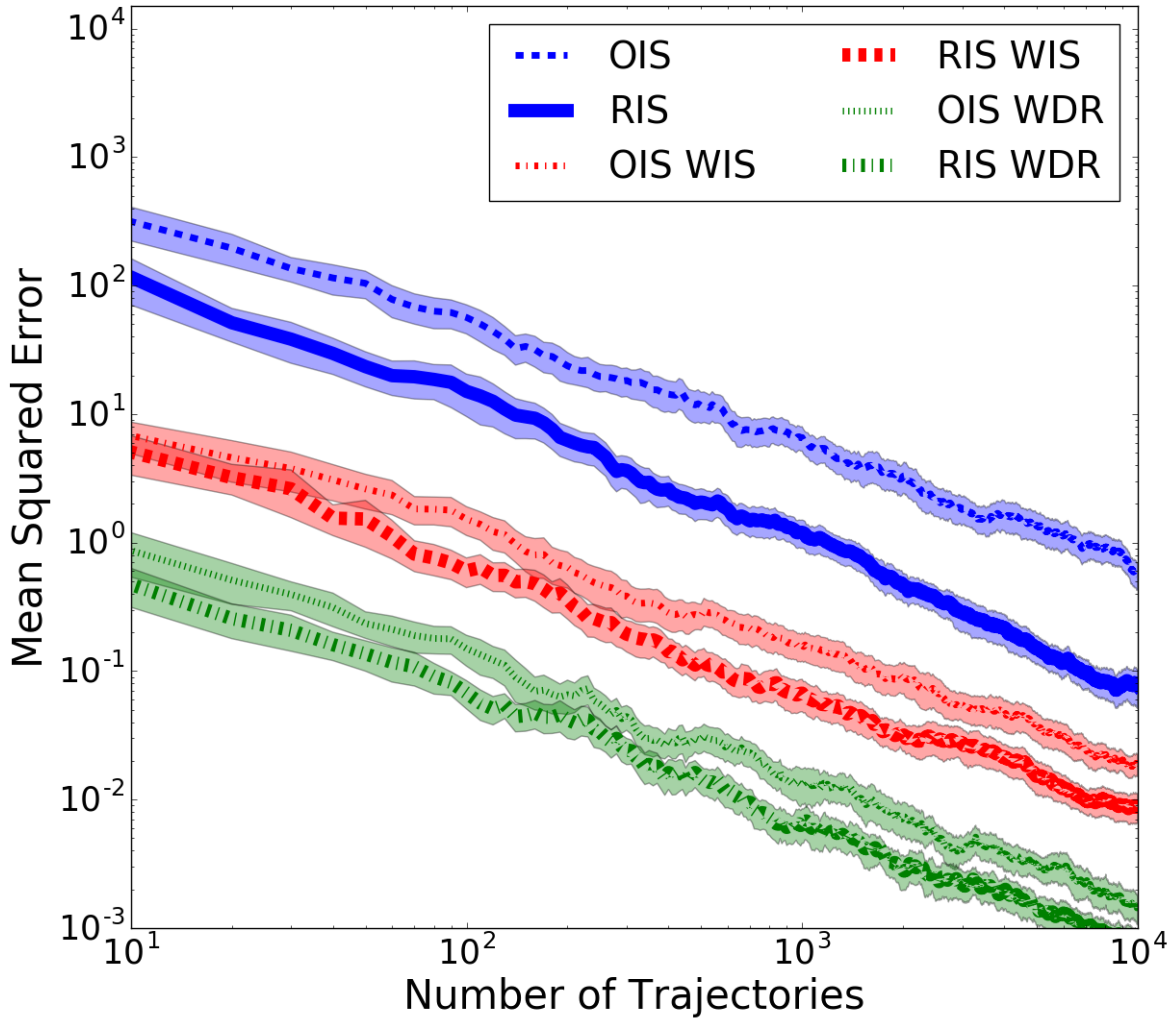


Gridworld

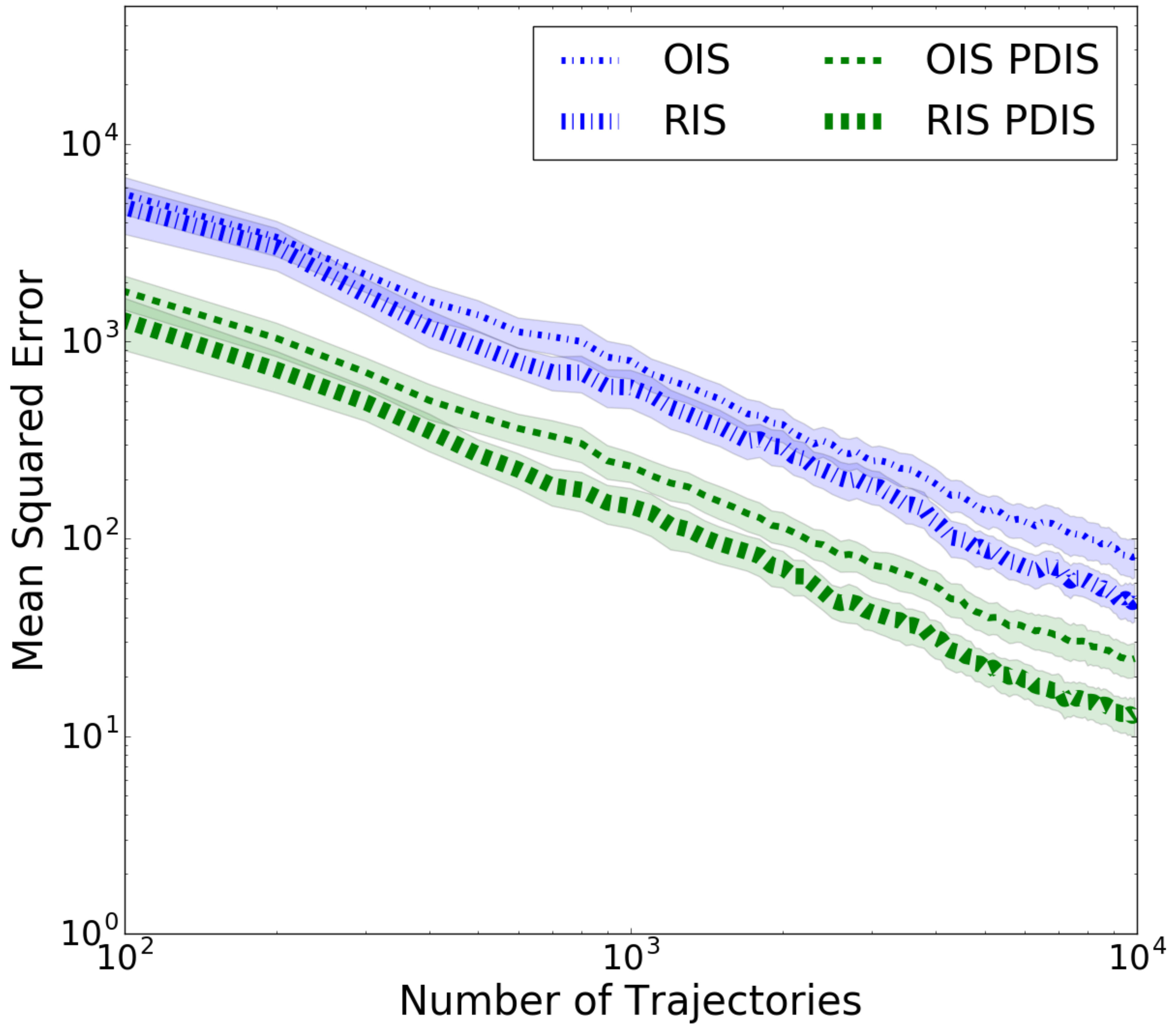


Linear Dynamical System

Empirical Results



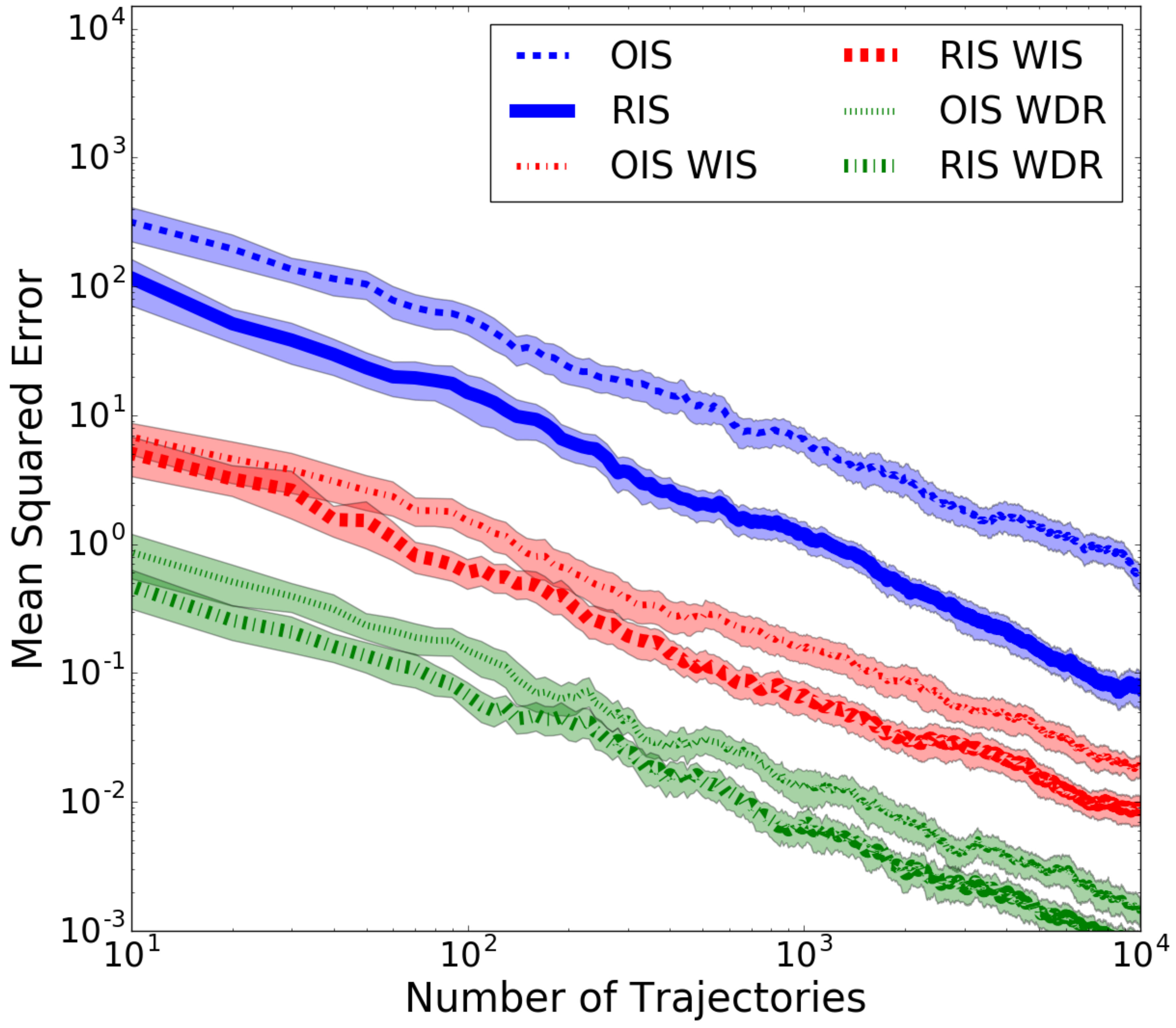
Gridworld



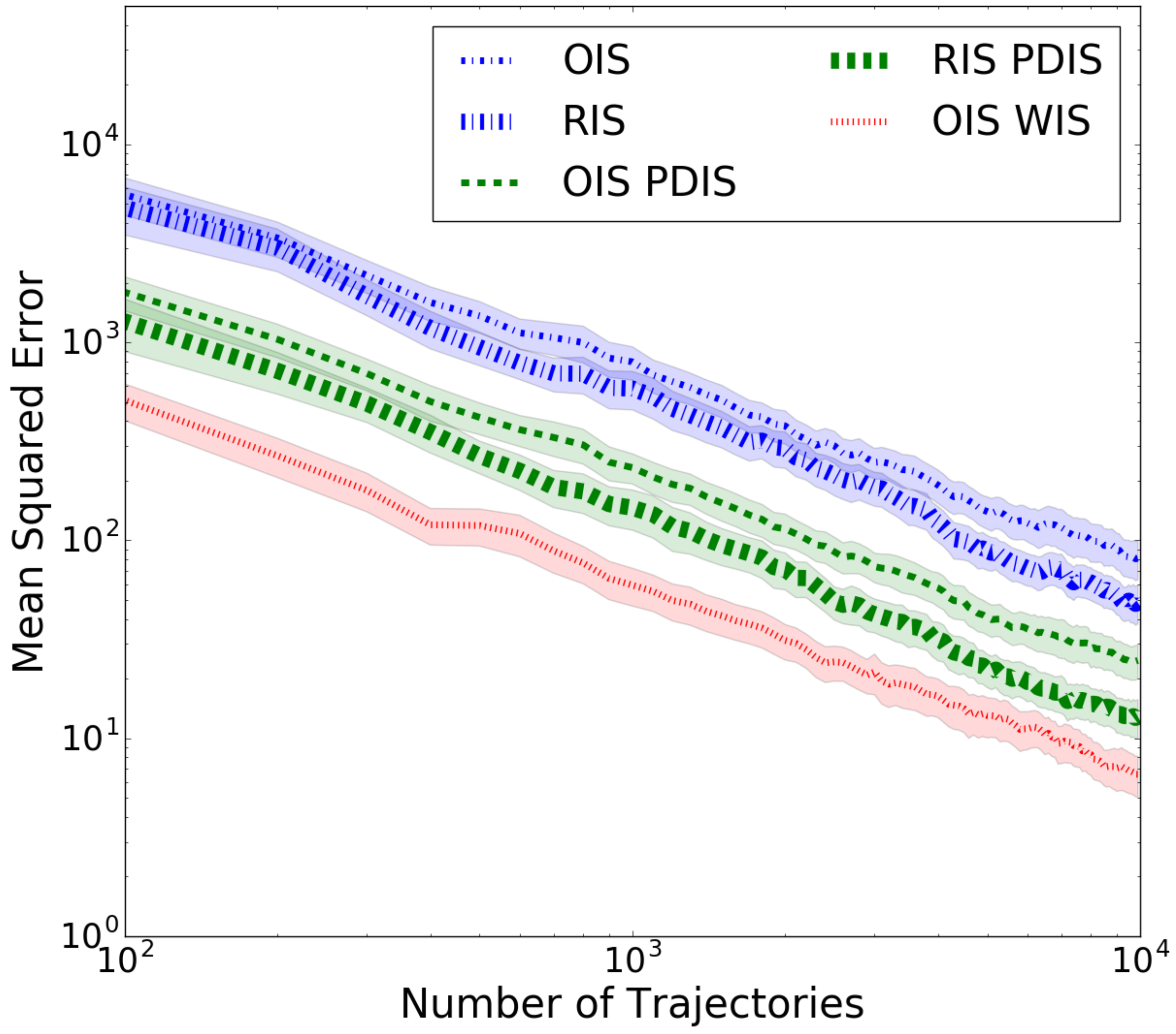
Linear Dynamical System

Josiah Hanna

Empirical Results

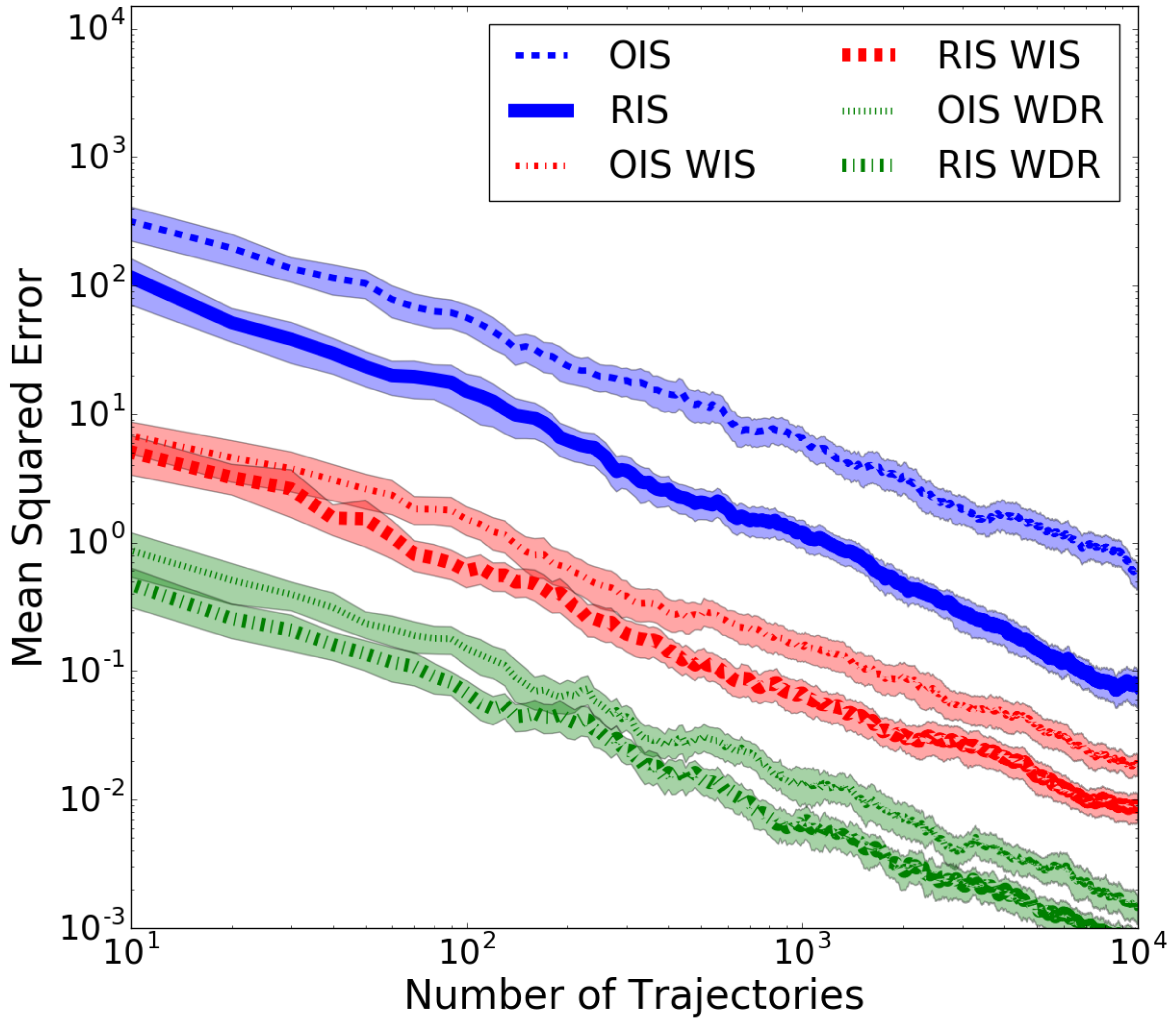


Gridworld

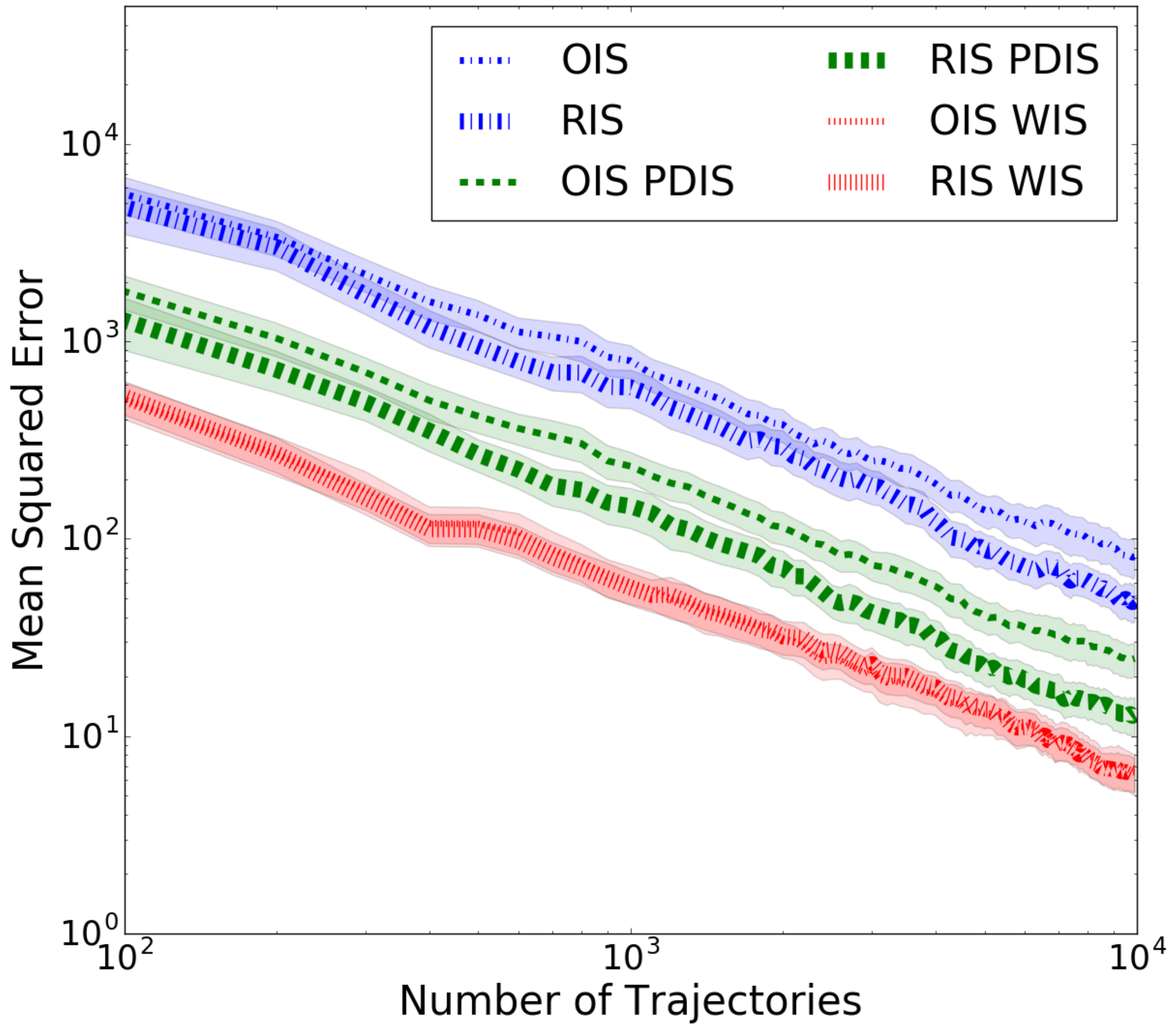


Linear Dynamical System

Empirical Results



Gridworld



Linear Dynamical System

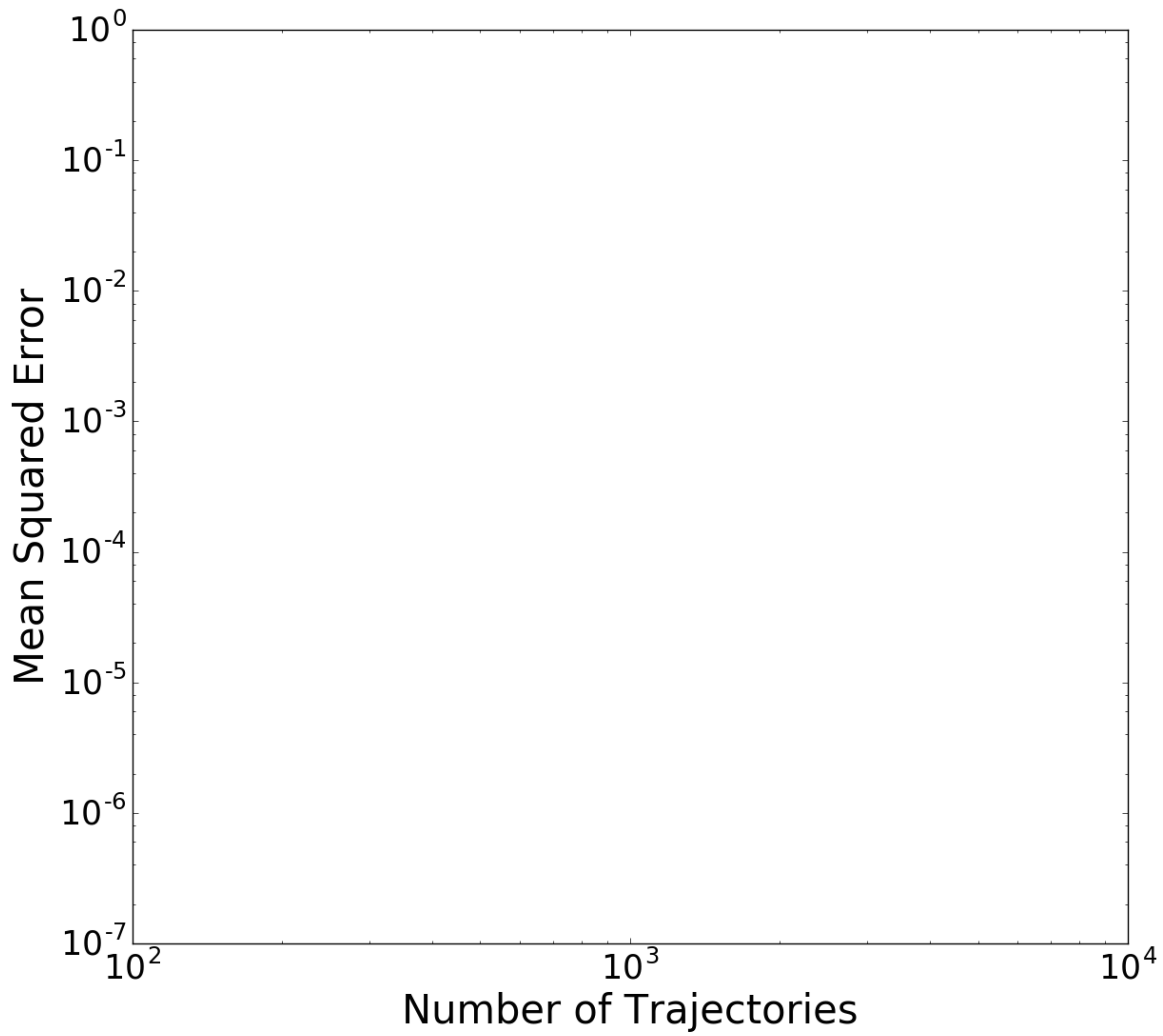
Non-Markovian Empirical Policies

Non-Markovian Empirical Policies

$$\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, ..., s_t)}$$

Non-Markovian Empirical Policies

$$\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, ..., s_t)}$$

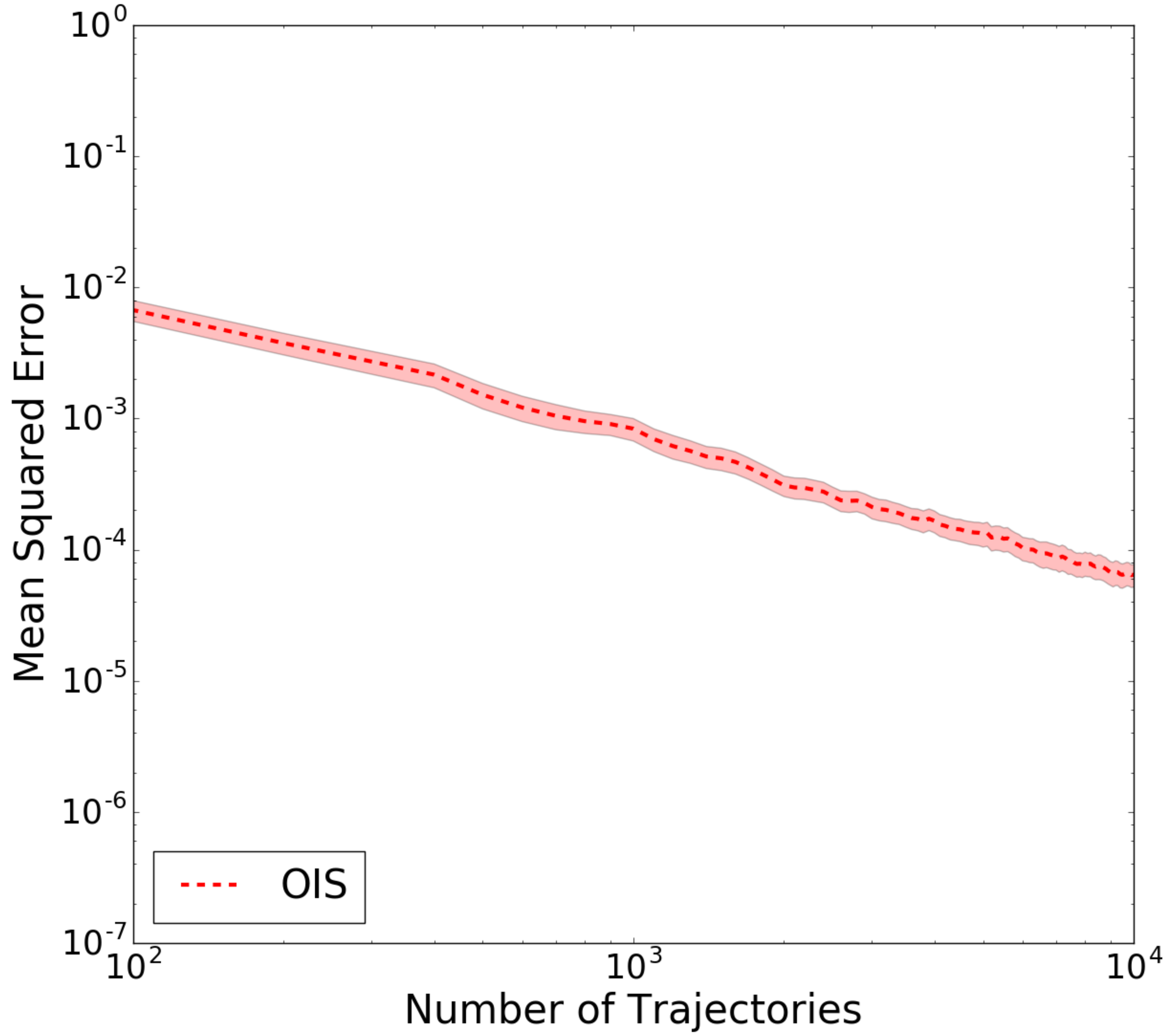


SinglePath MDP (horizon of 5)

Josiah Hanna

Non-Markovian Empirical Policies

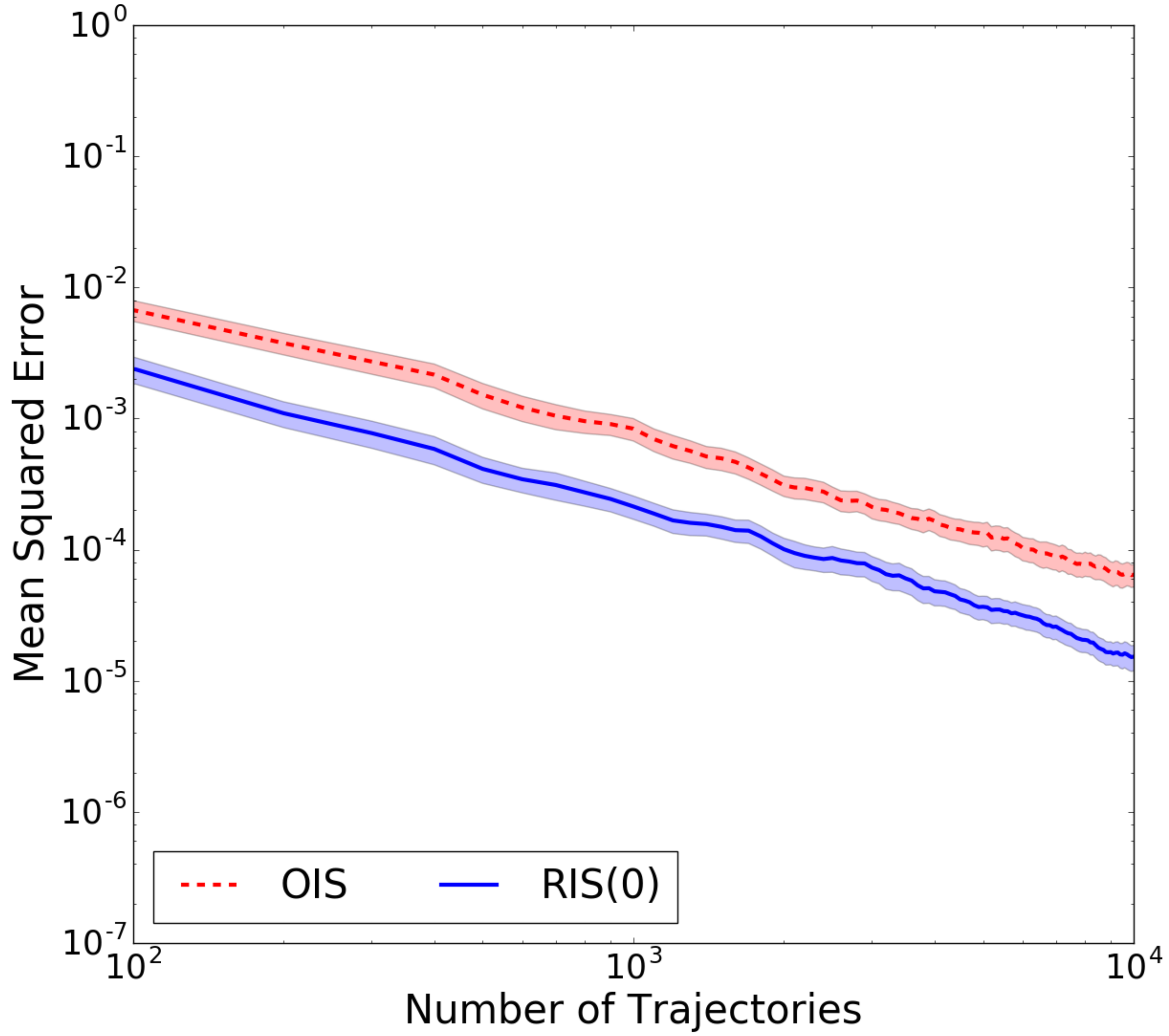
$$\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, ..., s_t)}$$



SinglePath MDP (horizon of 5)

Non-Markovian Empirical Policies

$$\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, ..., s_t)}$$

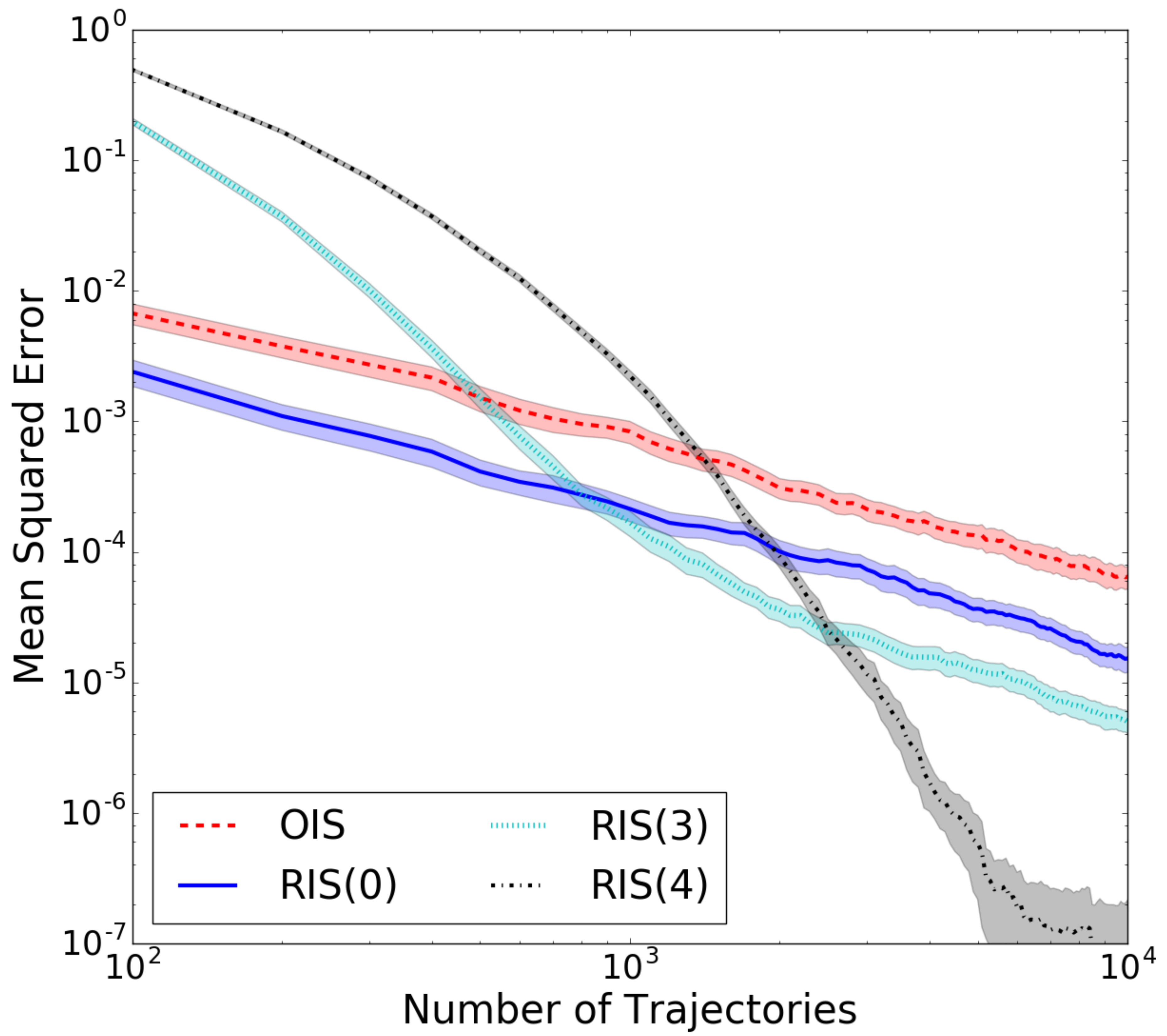


SinglePath MDP (horizon of 5)

Josiah Hanna

Non-Markovian Empirical Policies

$$\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, ..., s_t)}$$

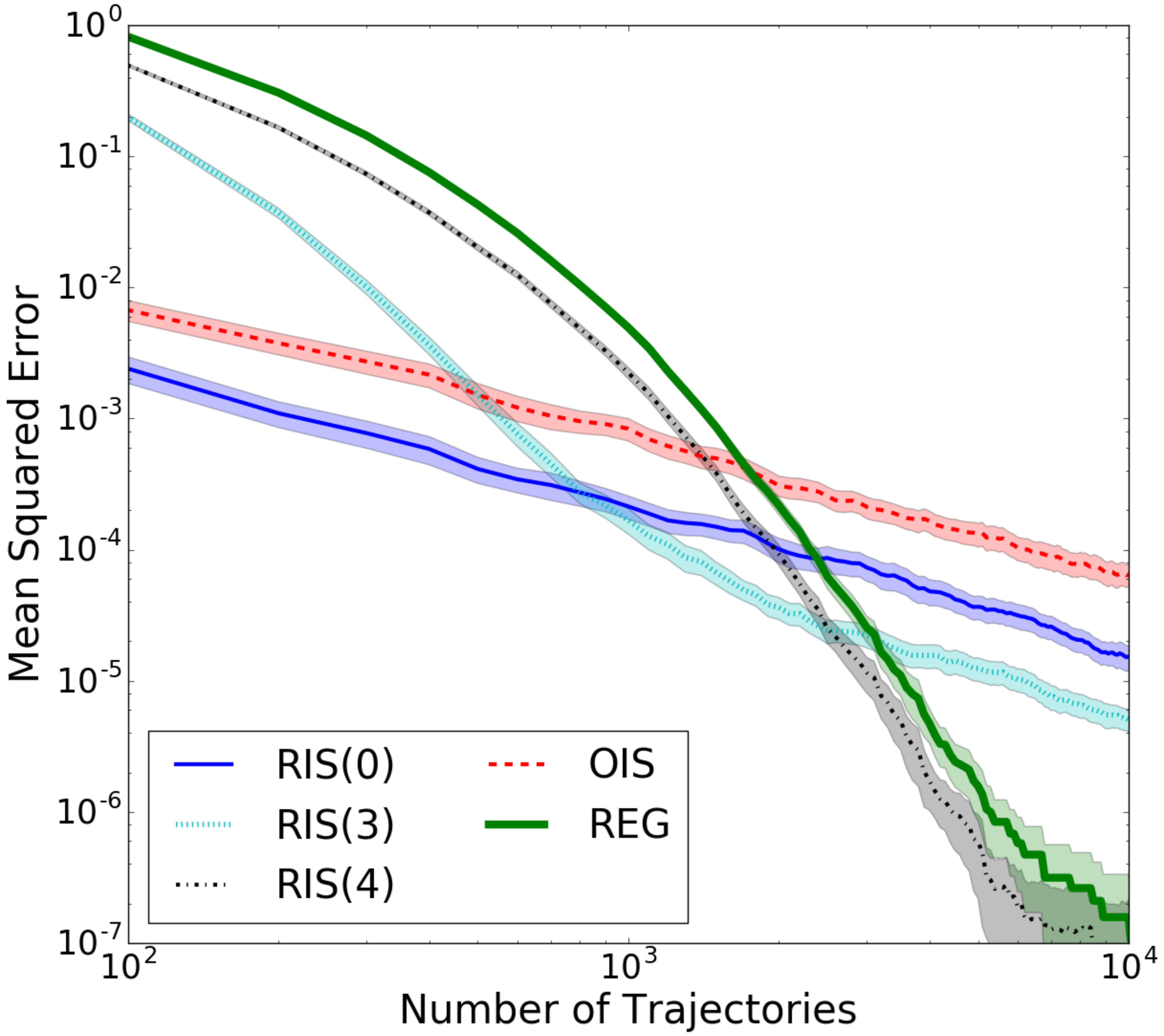


SinglePath MDP (horizon of 5)

Josiah Hanna

Non-Markovian Empirical Policies

$$\prod_{t=0}^{L} \frac{\pi(a_t|s_t)}{\pi_{\mathcal{D}}(a_t|s_{t-n}, a_{t-n}, \ldots, s_t)}$$



SinglePath MDP (horizon of 5)

Not Only for Off-Policy Data

Not Only for Off-Policy Data

Same results when behavior policy and evaluation policy are identical.

Not Only for Off-Policy Data

Same results when behavior policy and evaluation policy are identical.

1. Any Monte Carlo sampling method may suffer from sampling error.

Not Only for Off-Policy Data

Same results when behavior policy and evaluation policy are identical.

1. Any Monte Carlo sampling method may suffer from sampling error.

2. If we know the desired action probability we can potentially correct this error.

Not Only for Off-Policy Data

Same results when behavior policy and evaluation policy are identical.

1. Any Monte Carlo sampling method may suffer from sampling error.
2. If we know the desired action probability we can potentially correct this error.
3. Can correcting sampling error improve other types of reinforcement learning algorithms?

Not Only for Off-Policy Data

Same results when behavior policy and evaluation policy are identical.

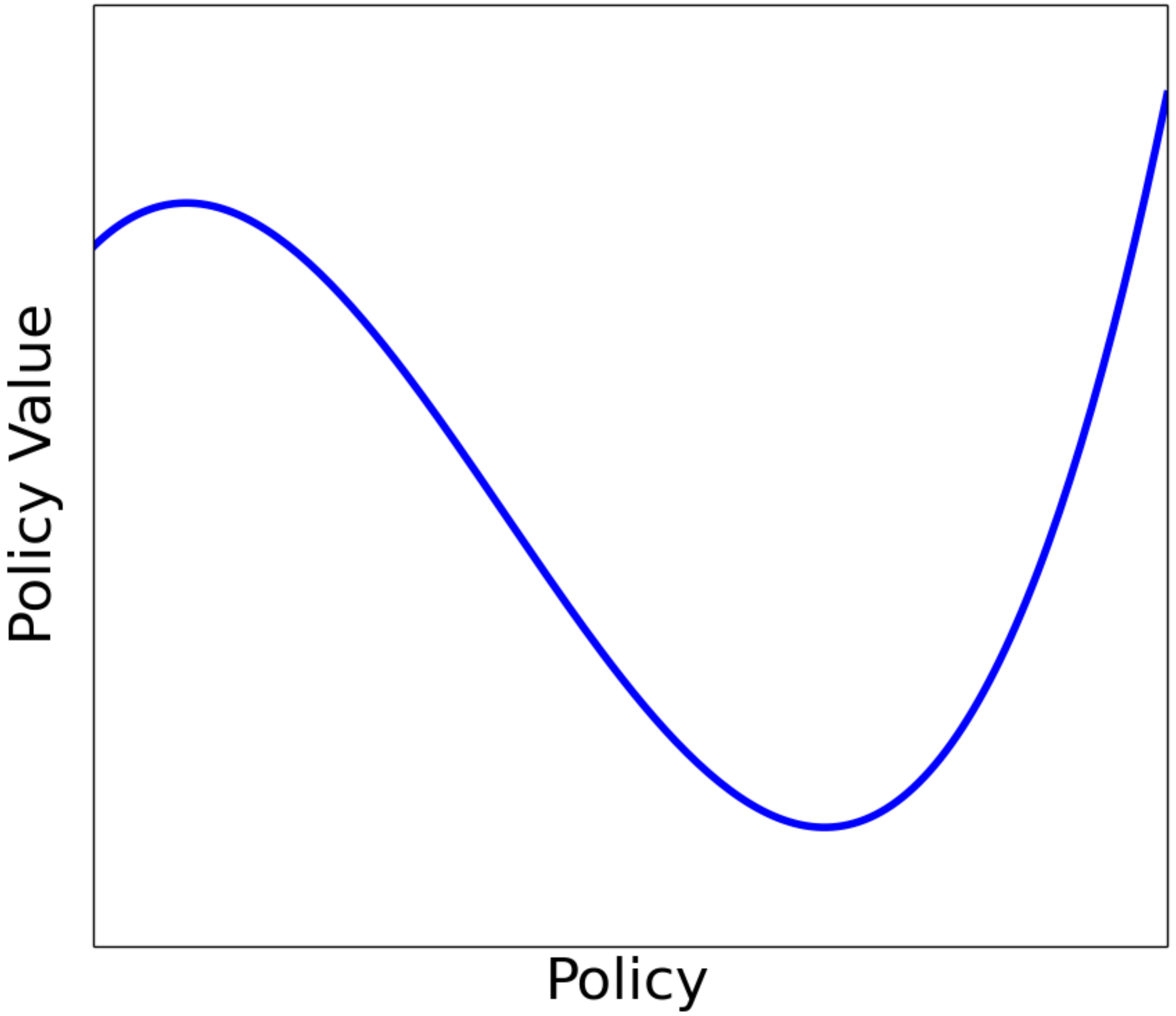
1. Any Monte Carlo sampling method may suffer from sampling error.
2. If we know the desired action probability we can potentially correct this error.
3. Can correcting sampling error improve other types of reinforcement learning algorithms?

Contribution 4:

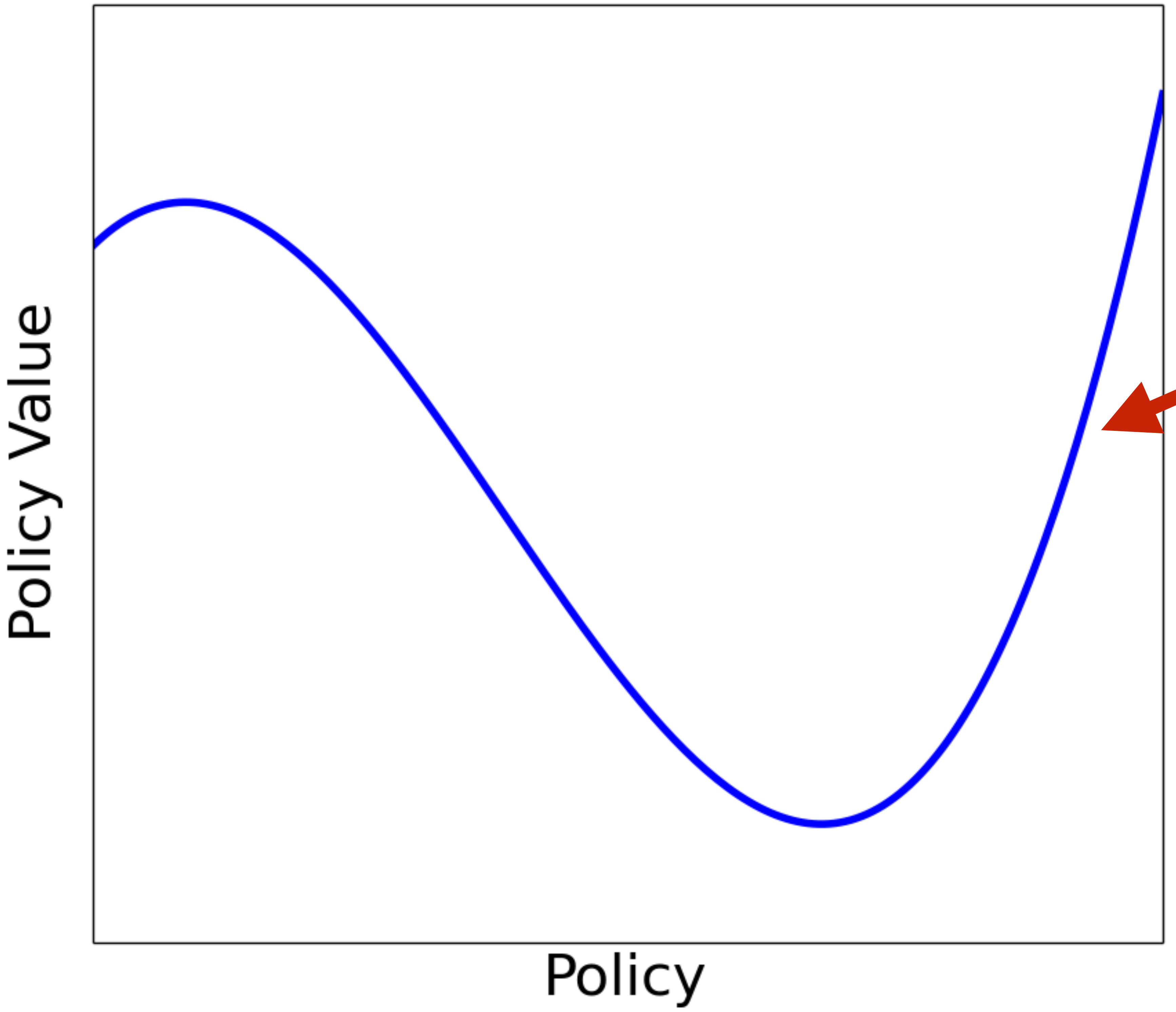
Sampling error corrected policy gradient estimator that improves over Monte Carlo policy gradient estimators.

Sampling Error in Policy Gradient RL

Sampling Error in Policy Gradient RL

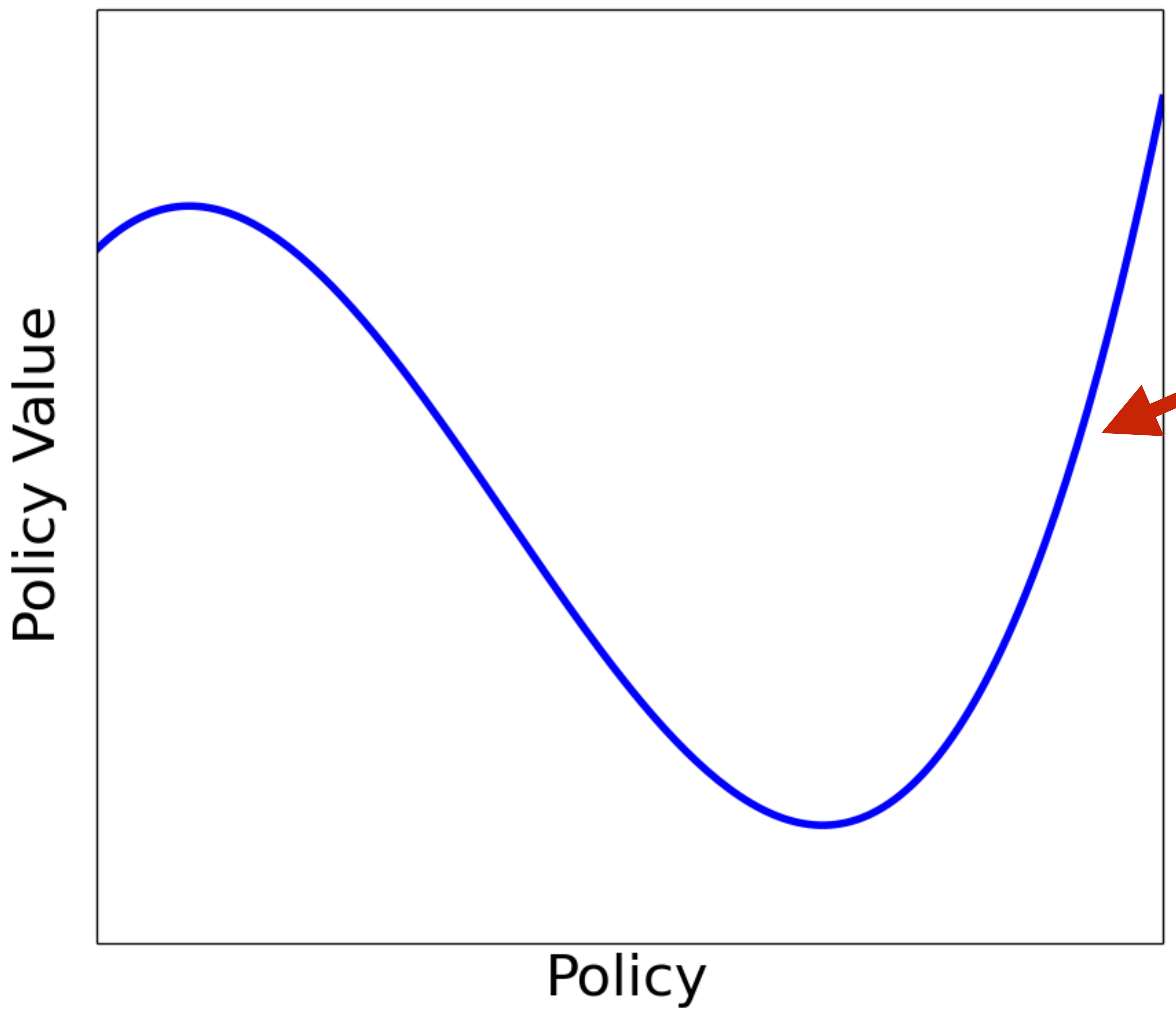


Sampling Error in Policy Gradient RL



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

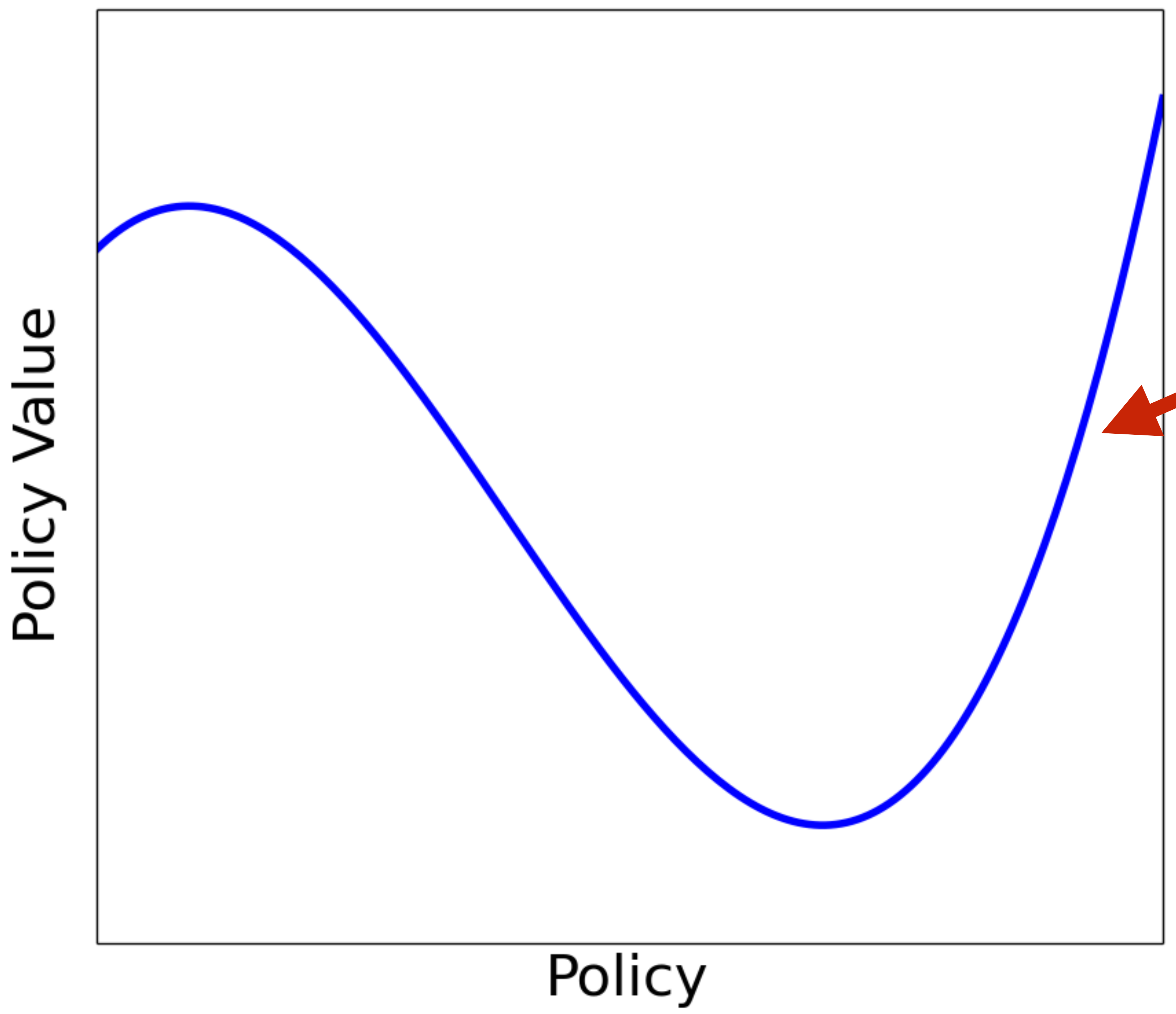
Sampling Error in Policy Gradient RL



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

State from policy's state distribution.

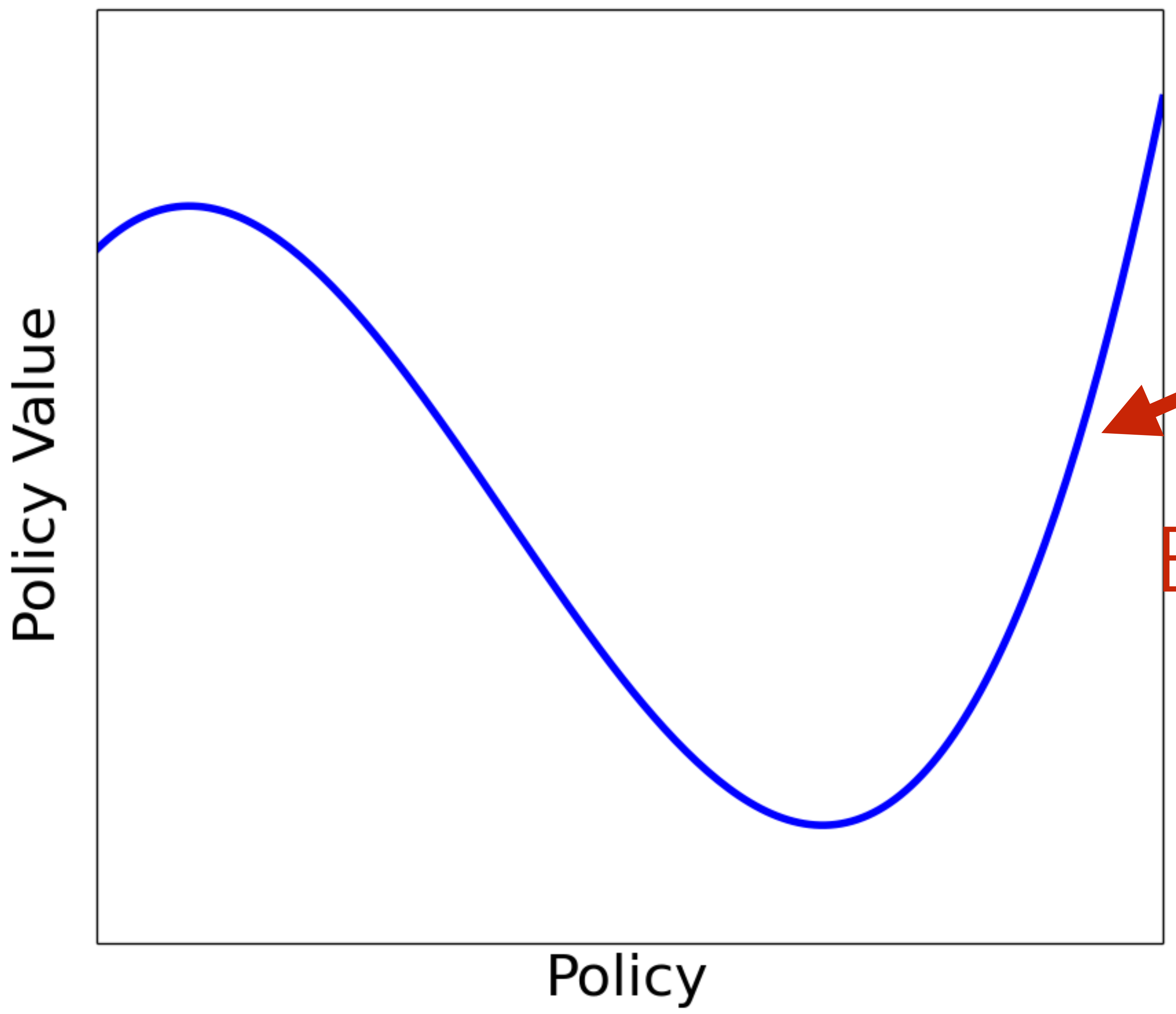
Sampling Error in Policy Gradient RL



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

Action from policy

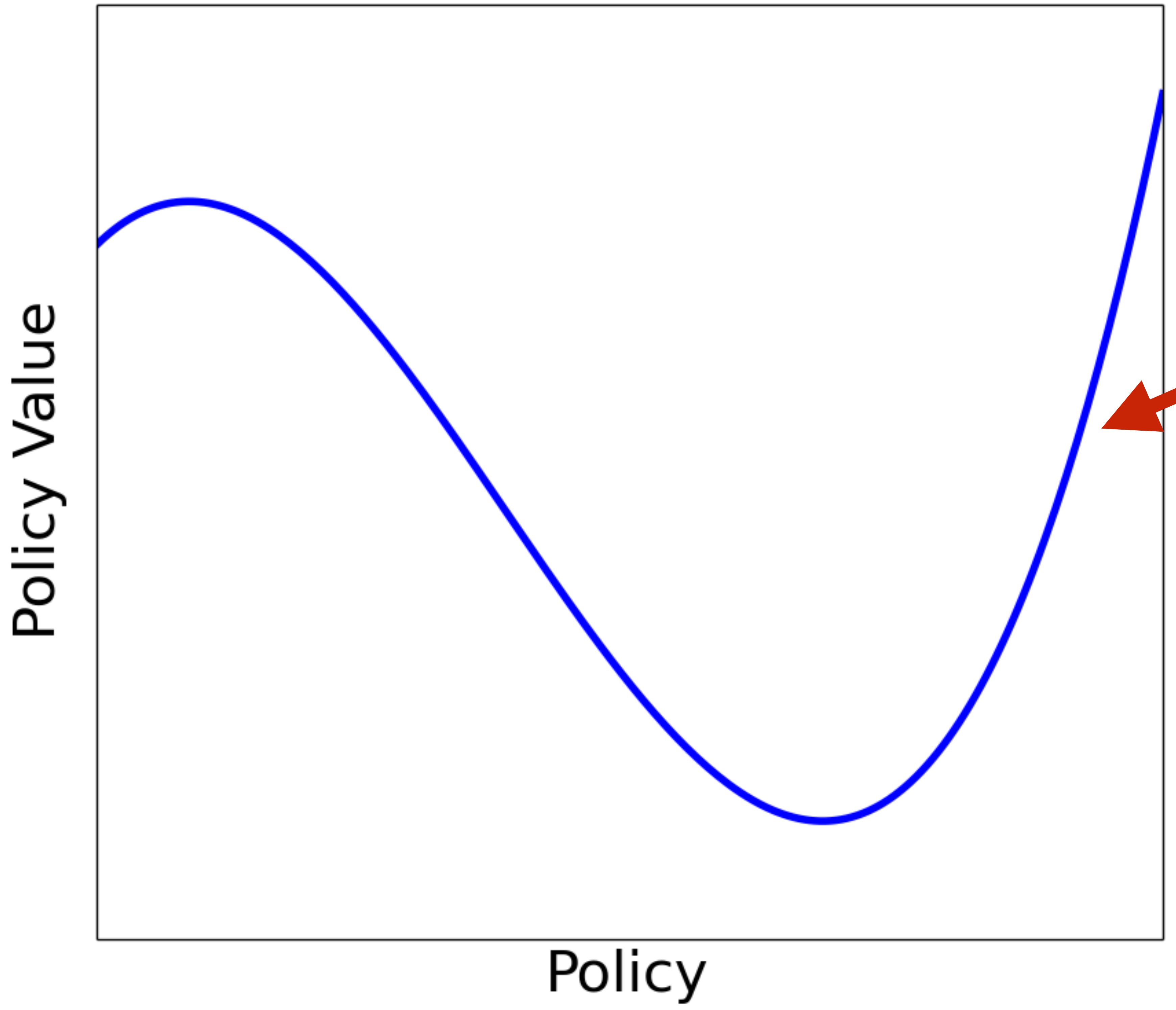
Sampling Error in Policy Gradient RL



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

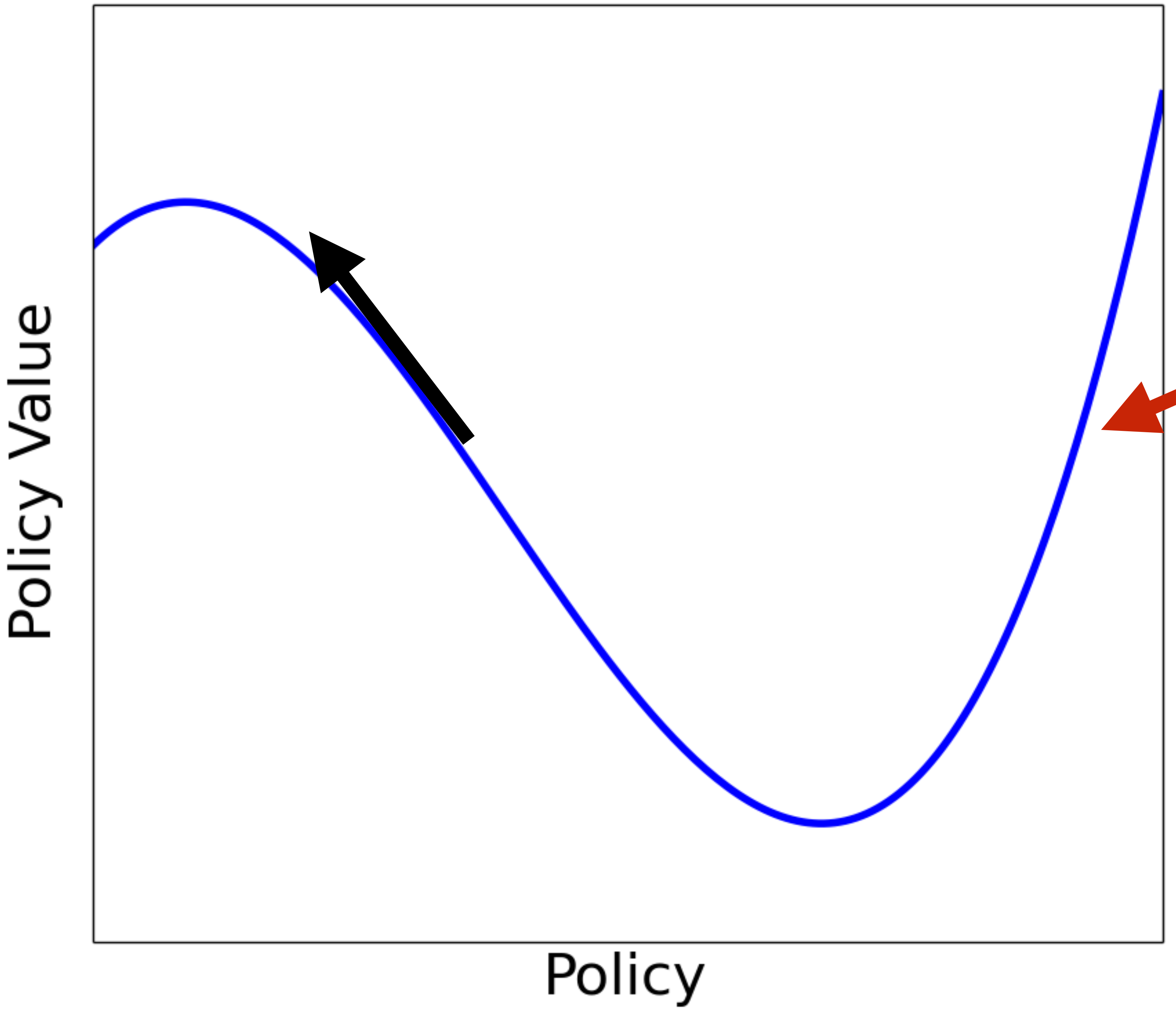
Expected reward following S,A

Sampling Error in Policy Gradient RL



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

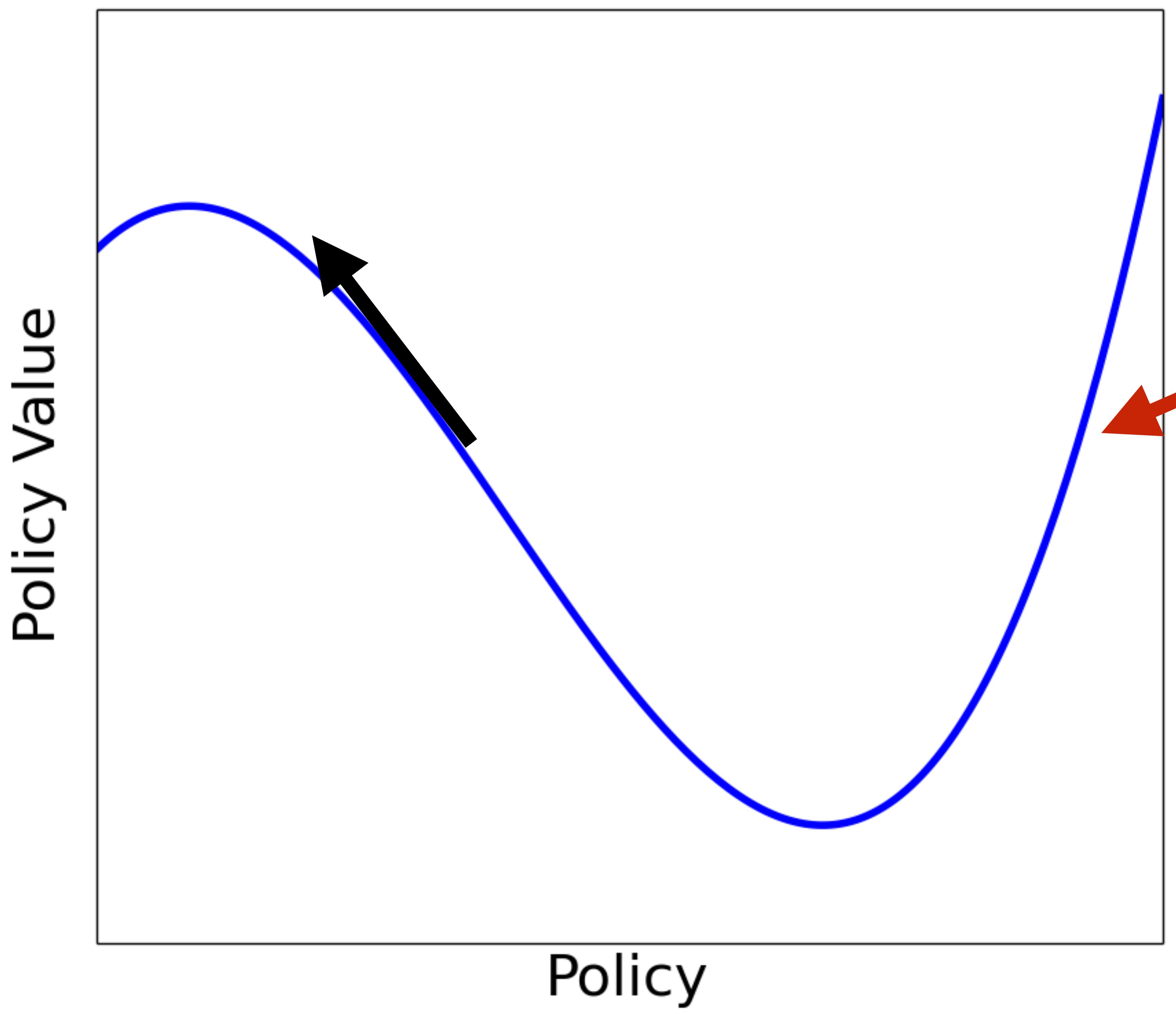
Sampling Error in Policy Gradient RL



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

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

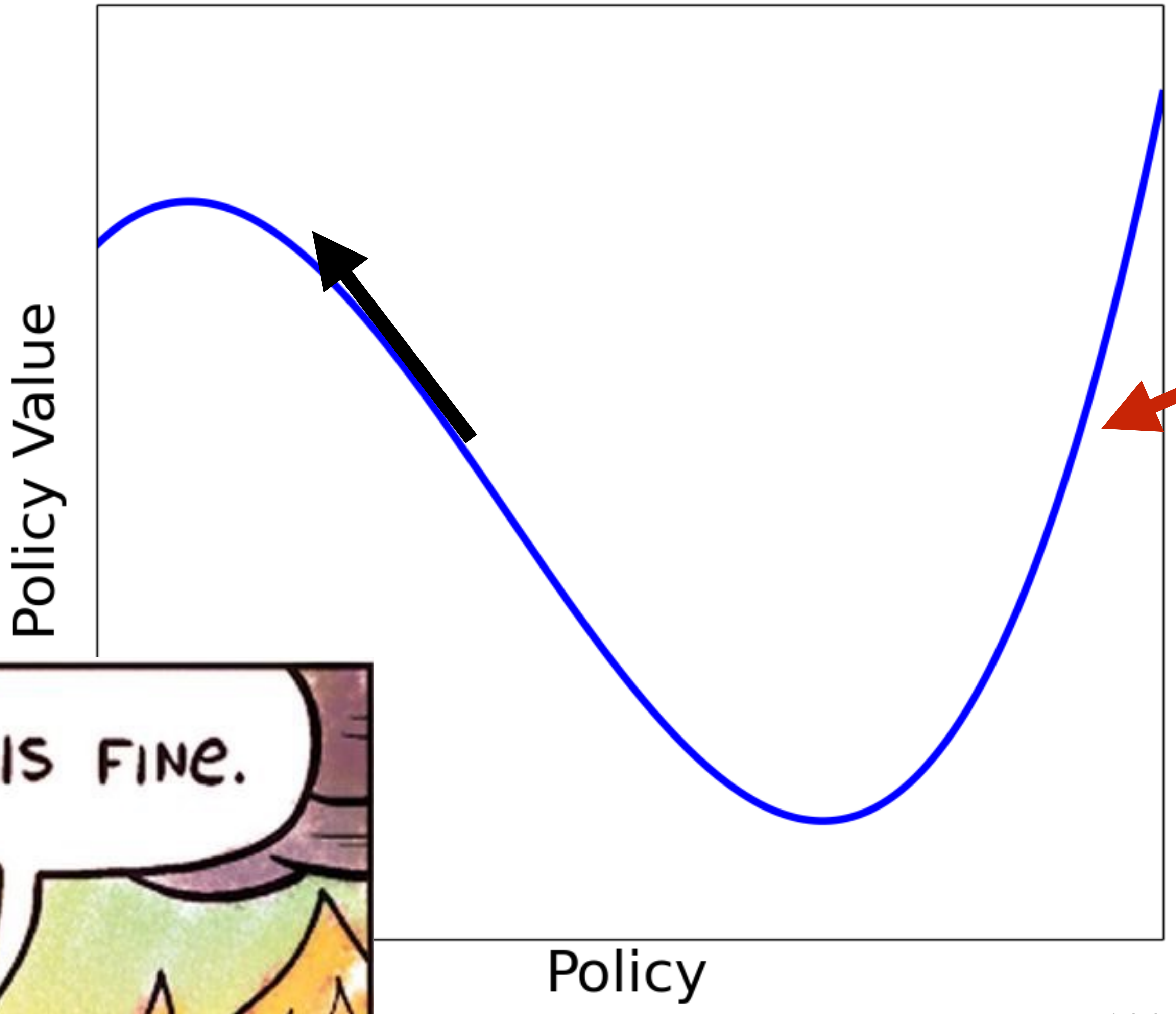
Sampling Error in Policy Gradient RL

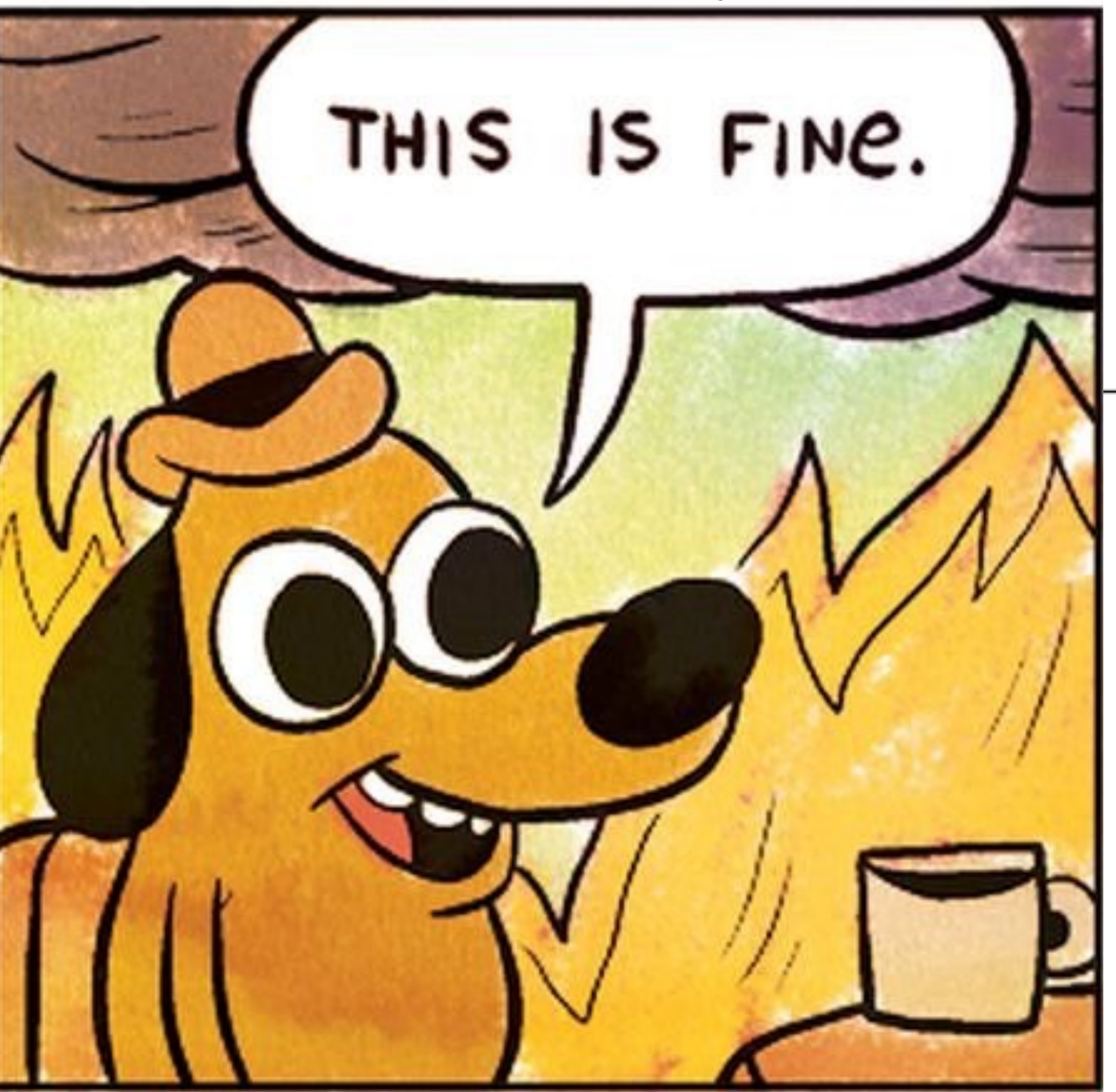


$$v(\pi_\theta) = \mathbf{E}[Q^{\pi_\theta}(S, A)]$$

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

Sampling Error in Policy Gradient RL

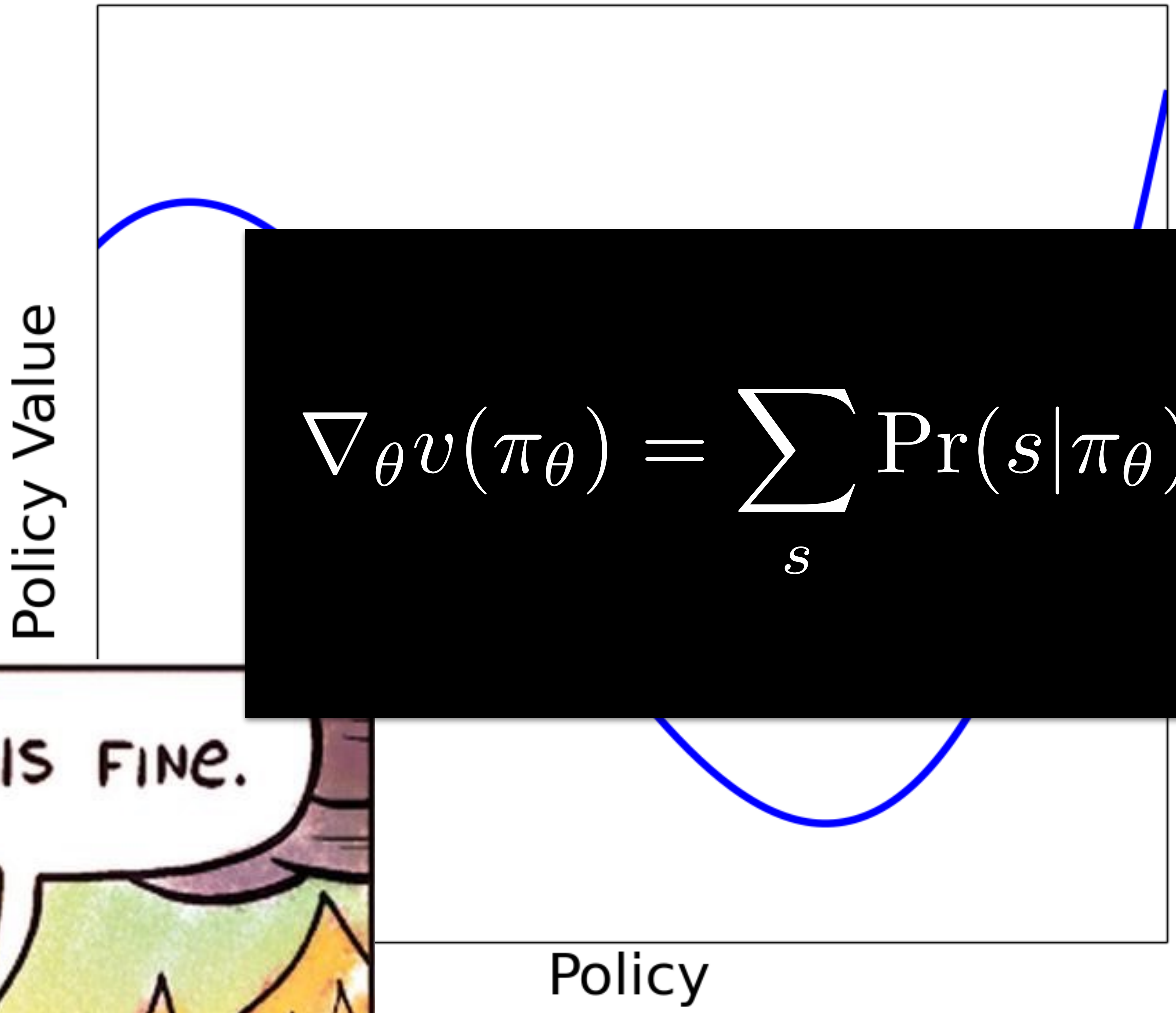




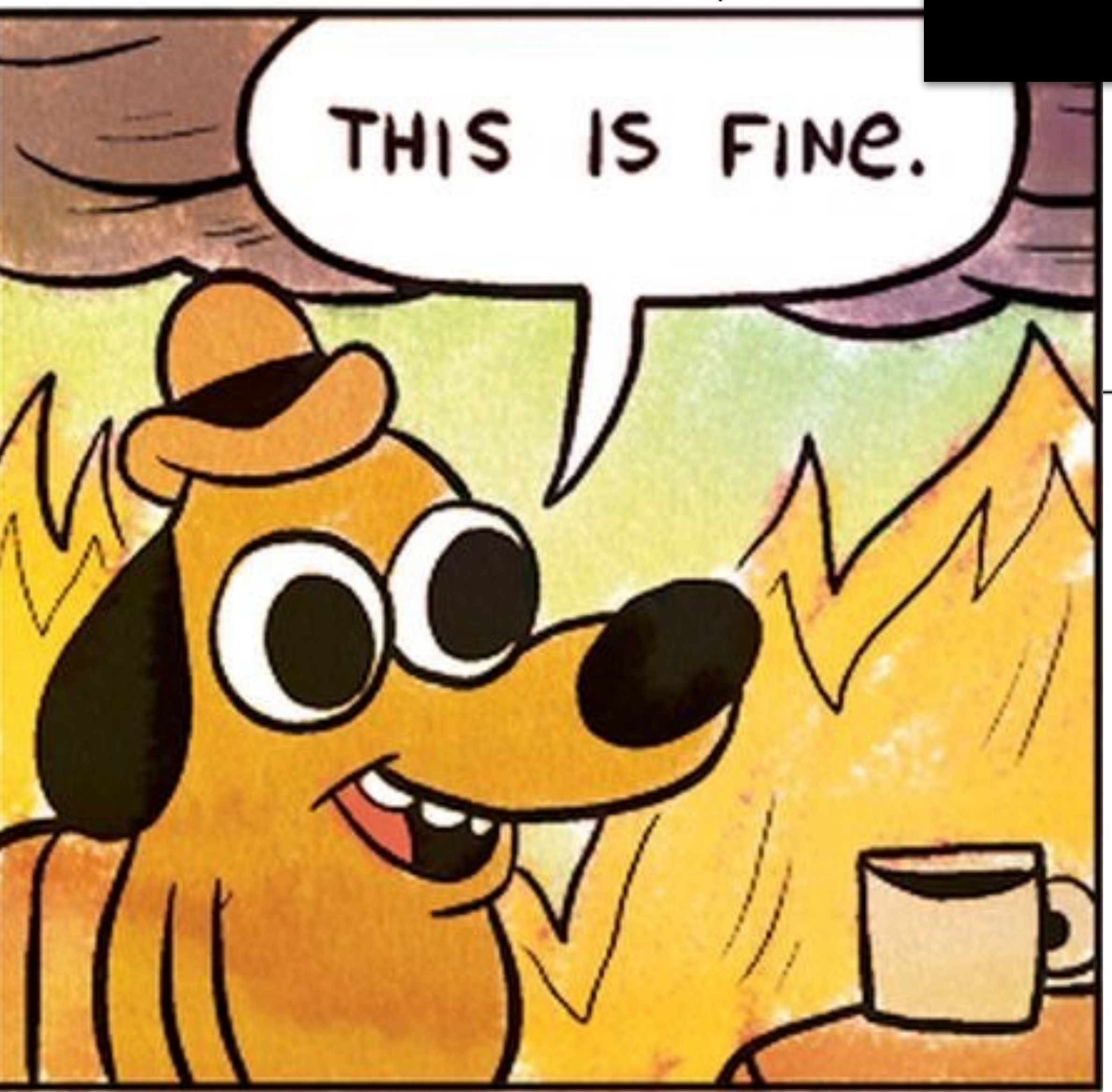
$$v(\pi_\theta) = \mathbf{E}[Q^{\pi_\theta}(S, A)]$$

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

Sampling Error in Policy Gradient RL

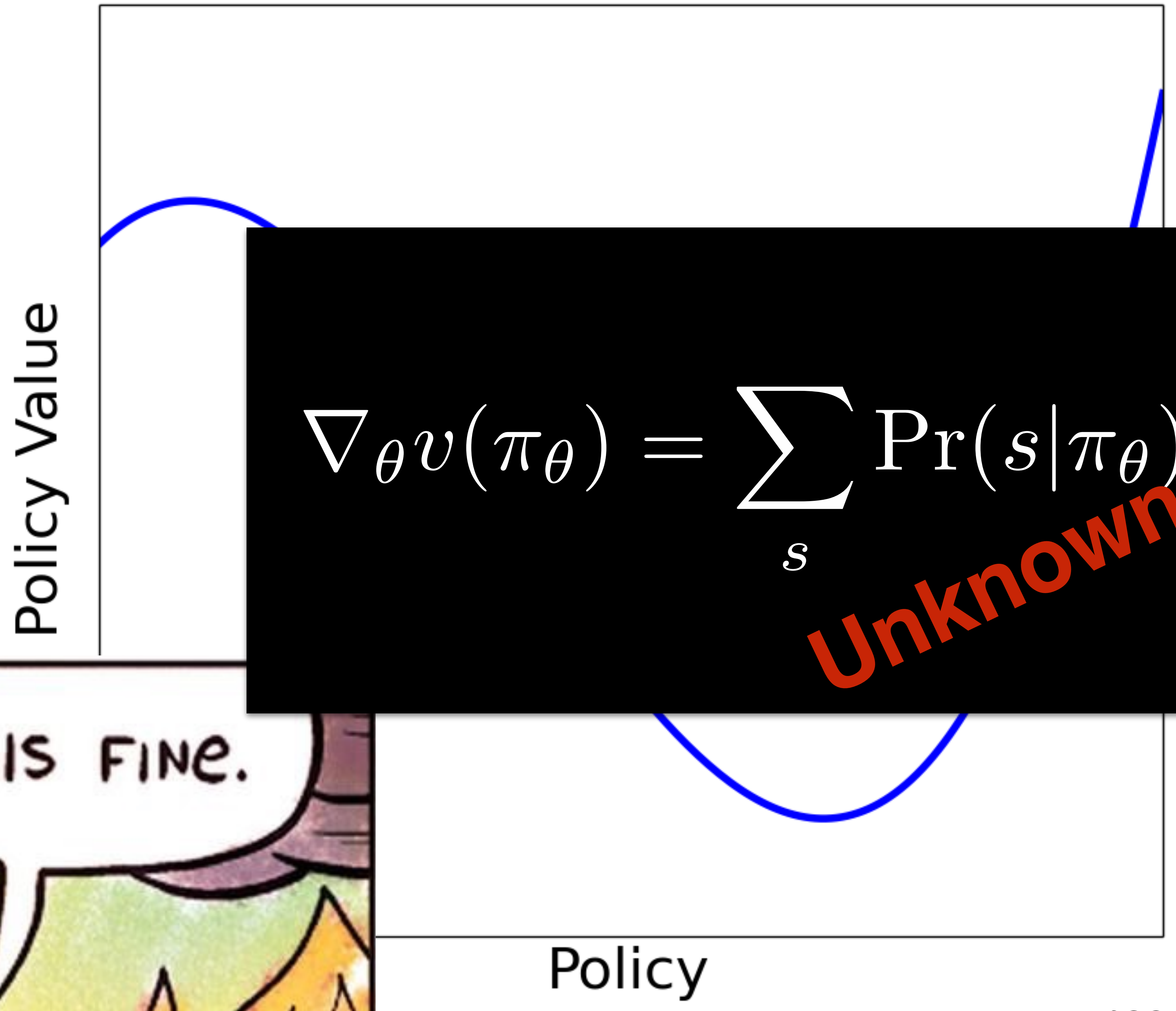


$$\nabla_\theta v(\pi_\theta) = \sum_s \Pr(s|\pi_\theta) \sum_a \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a|s)$$



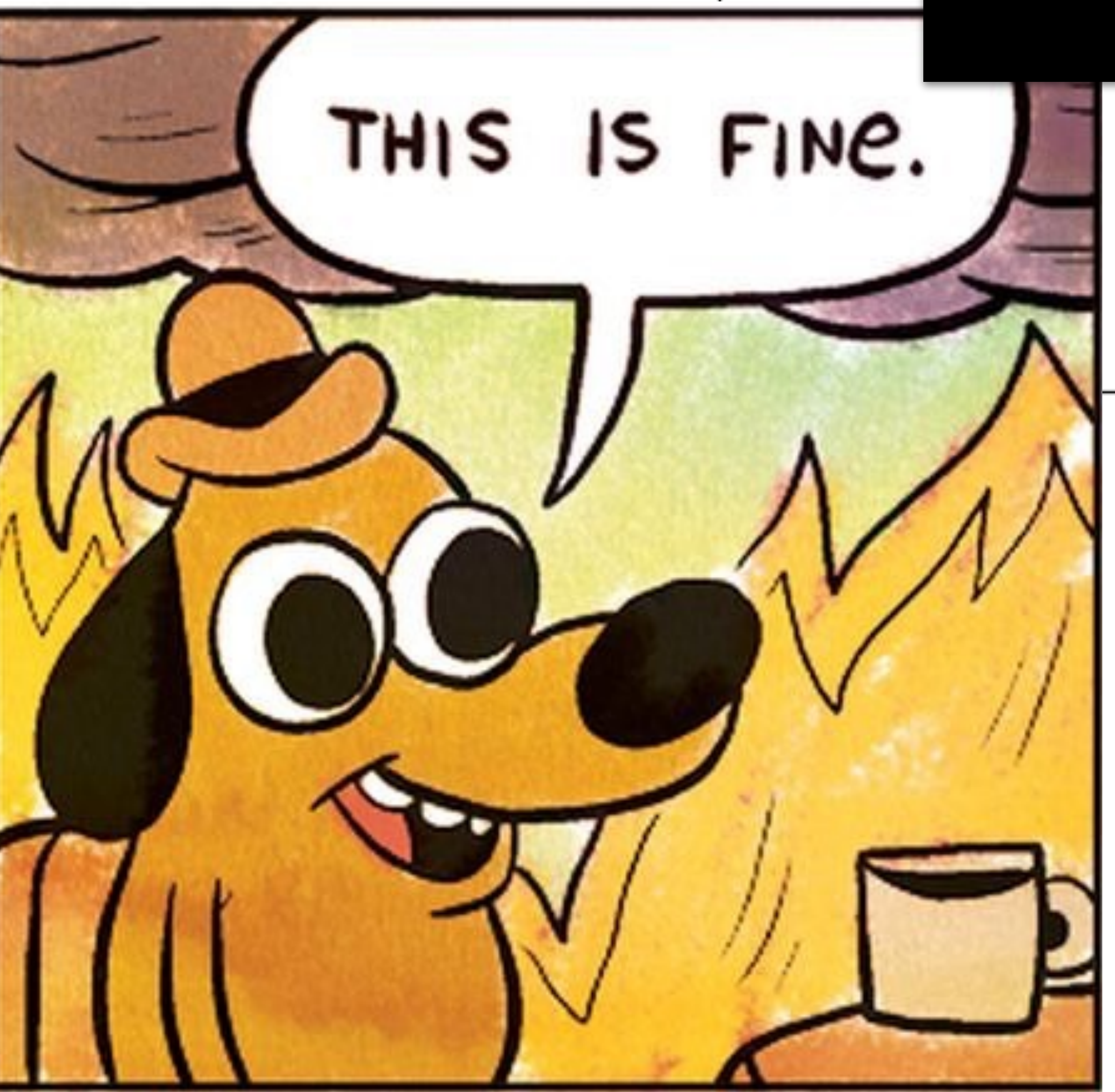
$$\nabla_\theta v(\pi_\theta) \approx \frac{1}{m} \sum_{i=1}^{m} Q^{\pi_\theta}(S_i, A_i) \nabla_\theta \log \pi_\theta(A_i|S_i)]$$

Sampling Error in Policy Gradient RL



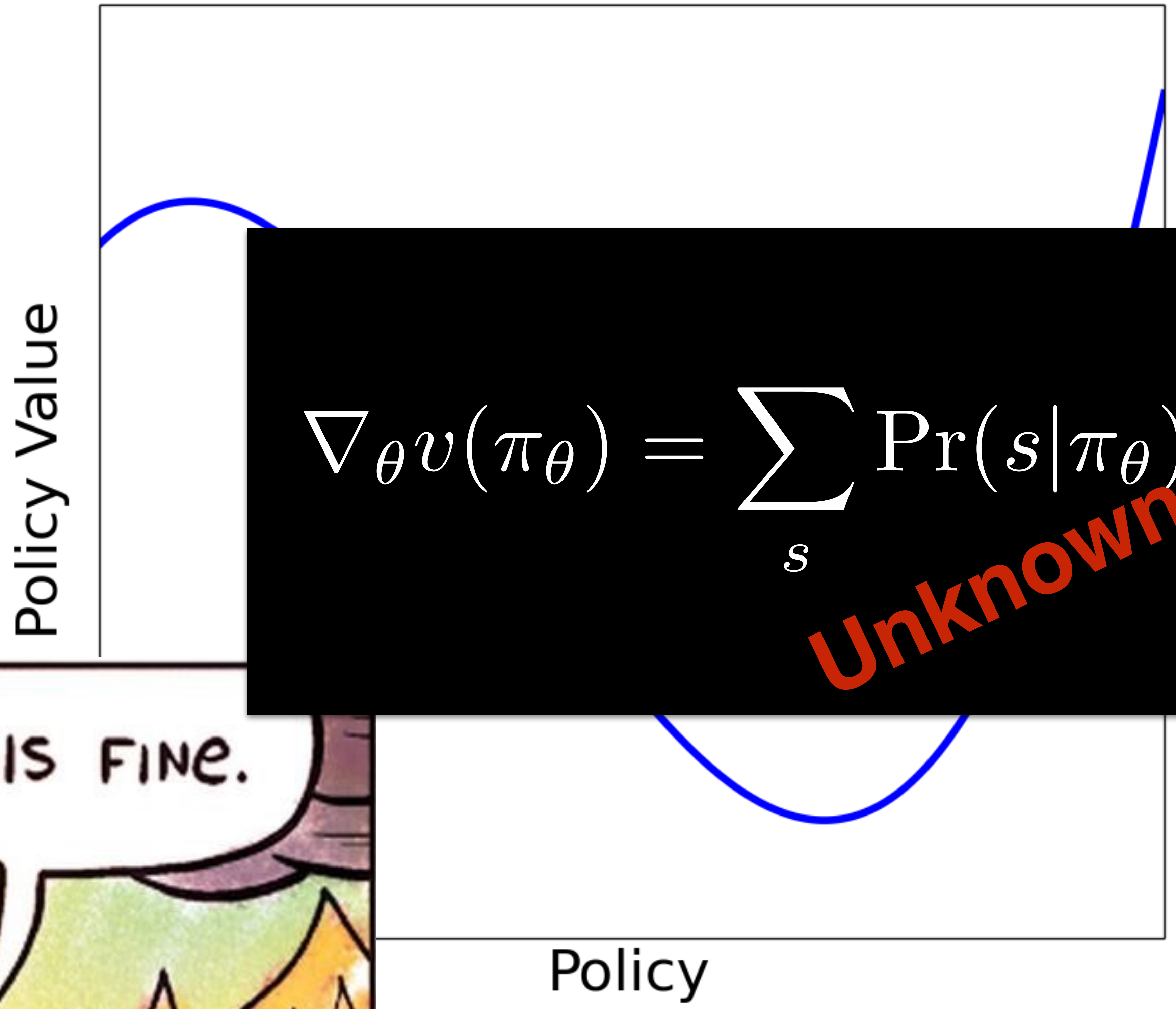
$$\nabla_\theta v(\pi_\theta) = \sum_s \Pr(s|\pi_\theta) \sum_a \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a|s)$$

Unknown



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

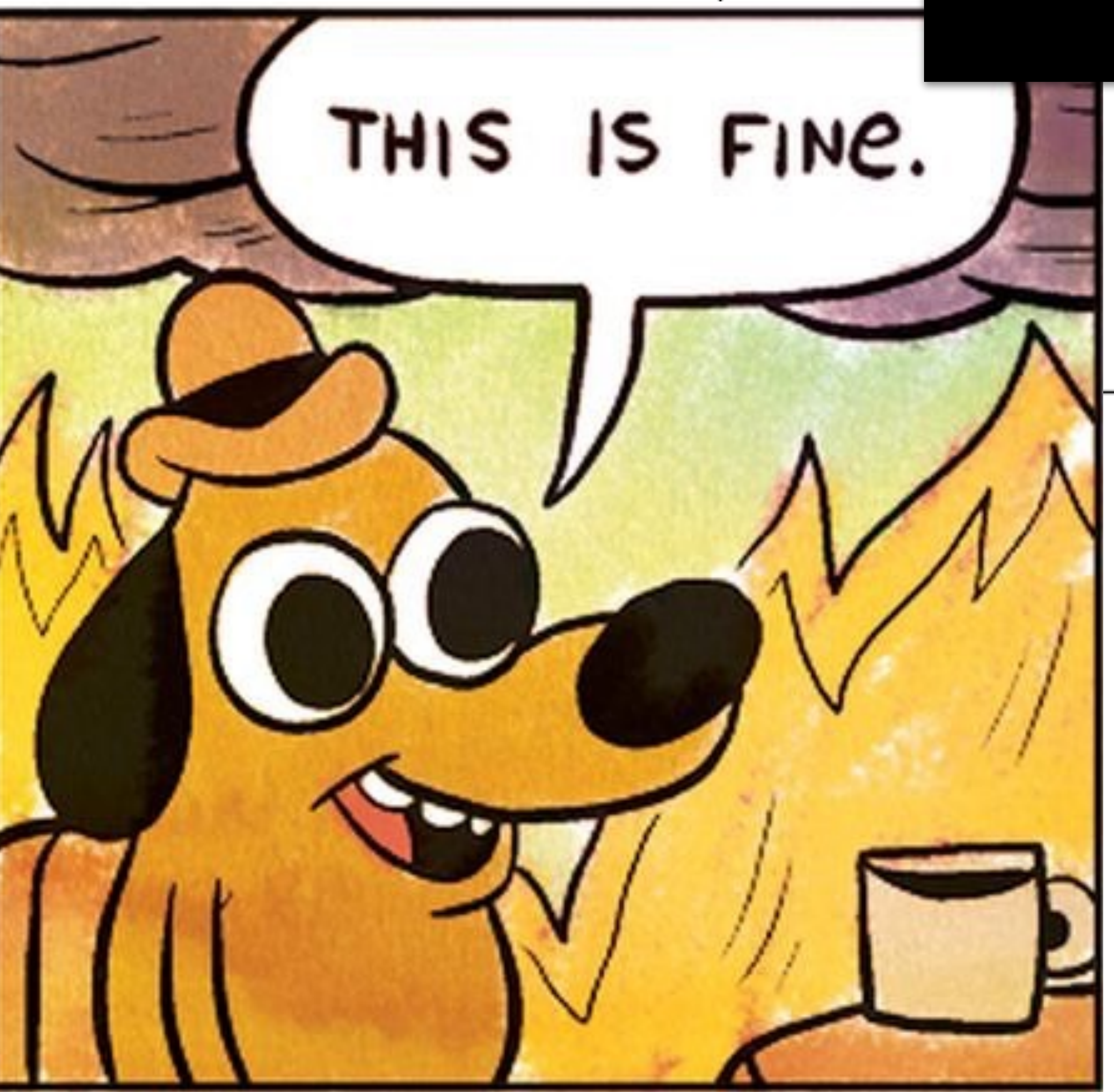
Sampling Error in Policy Gradient RL



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

Unknown

Known



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

Monte Carlo Policy Gradient

Monte Carlo Policy Gradient

1. Execute current policy for m steps.

Monte Carlo Policy Gradient

1. Execute current policy for m steps.

2. Update policy with Monte Carlo policy gradient estimate.

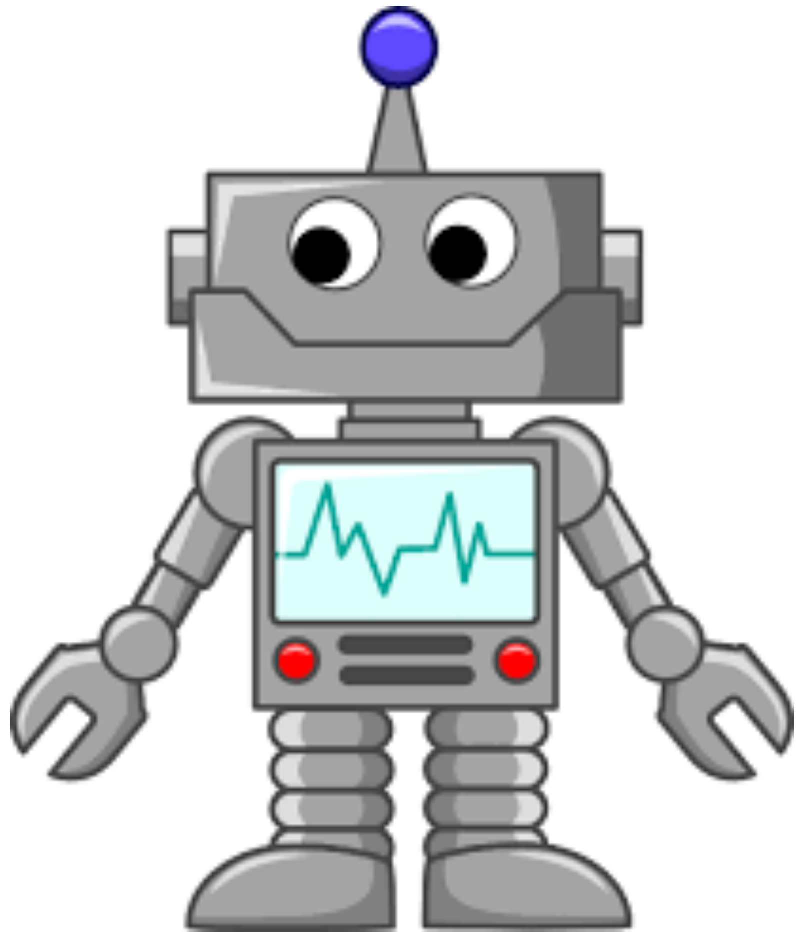
Monte Carlo Policy Gradient

1. Execute current policy for m steps.

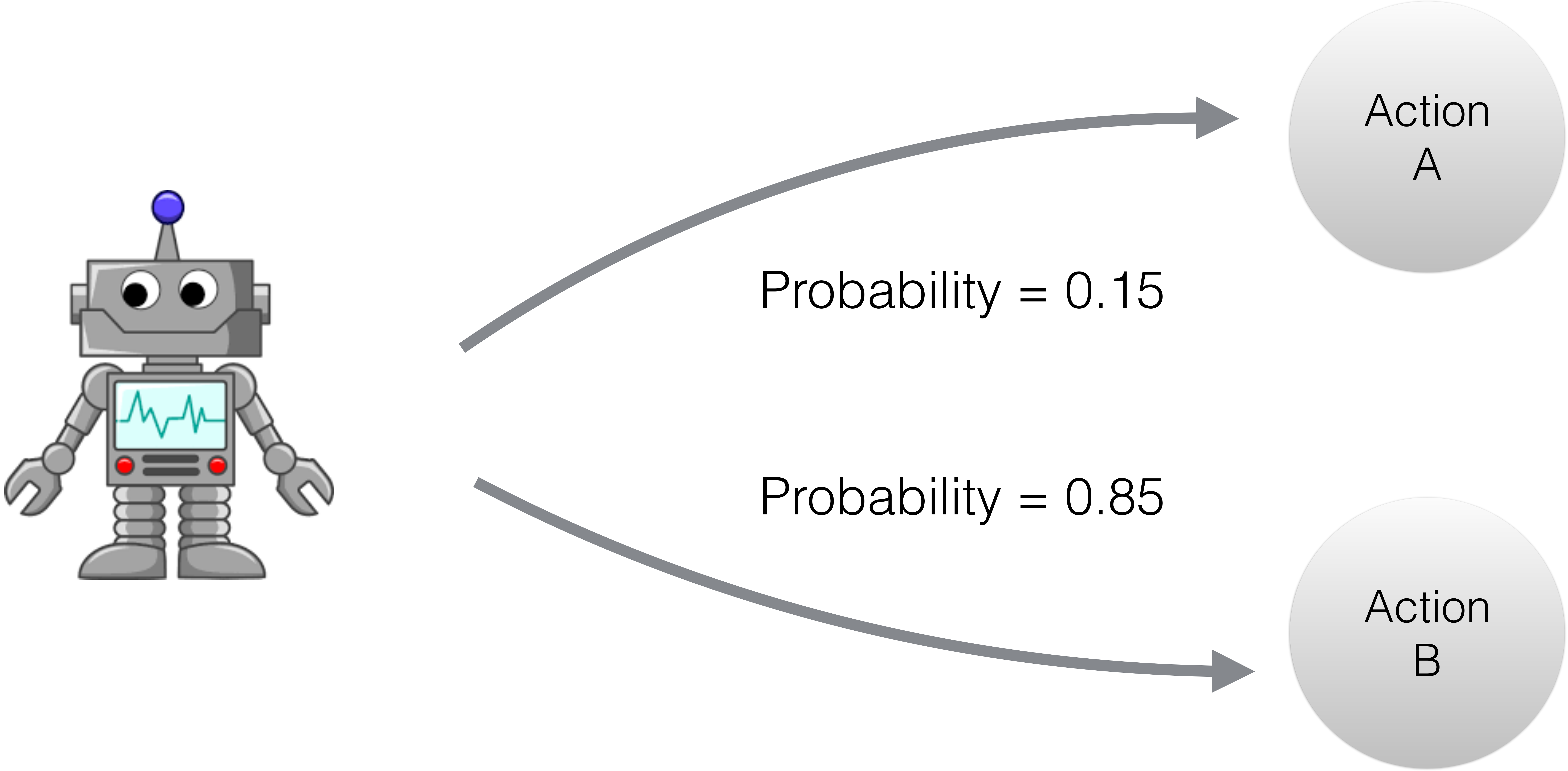
2. Update policy with Monte Carlo policy gradient estimate.

3. Throw away observed data and repeat (on-policy).

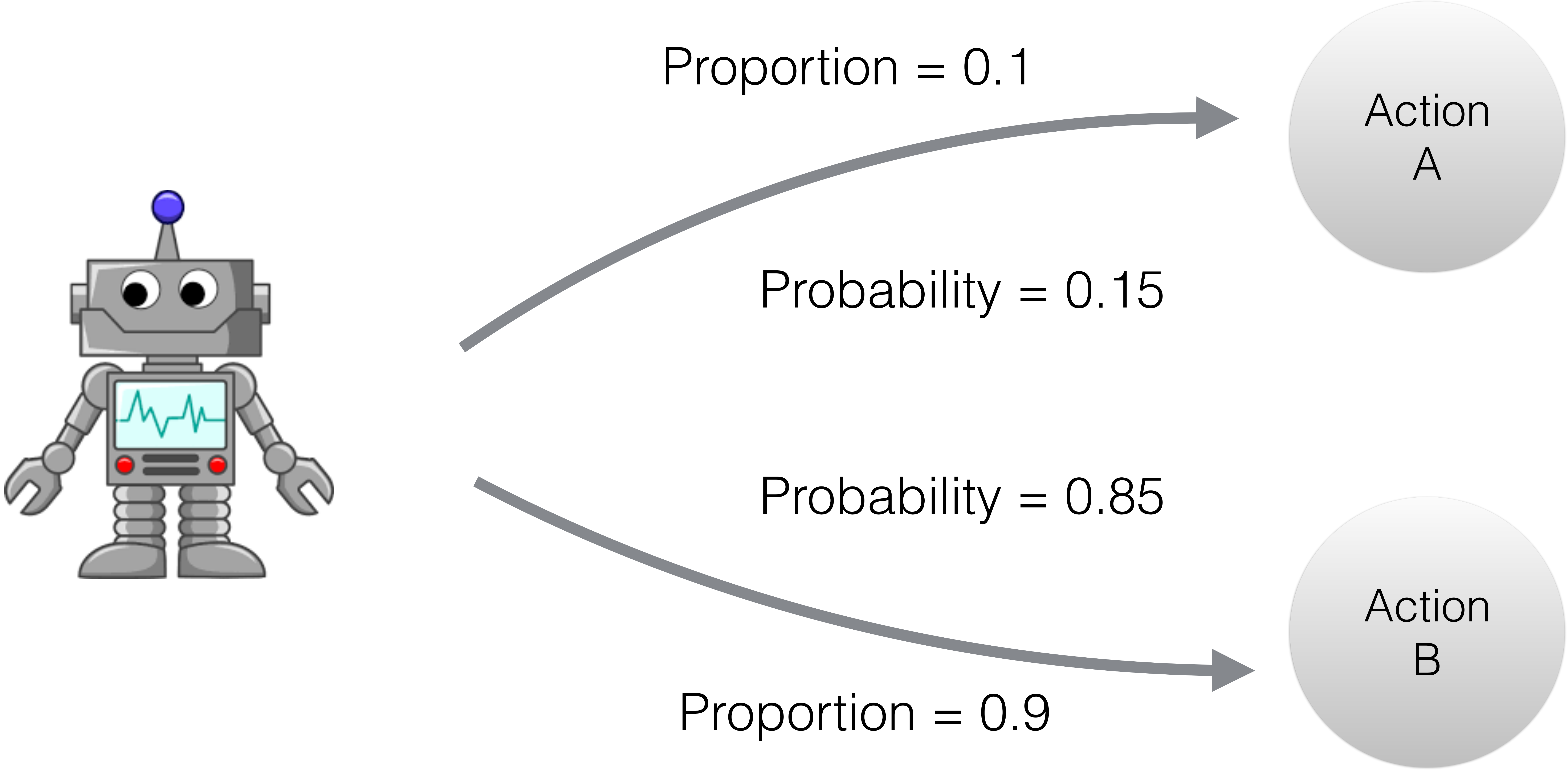
Sampling Error



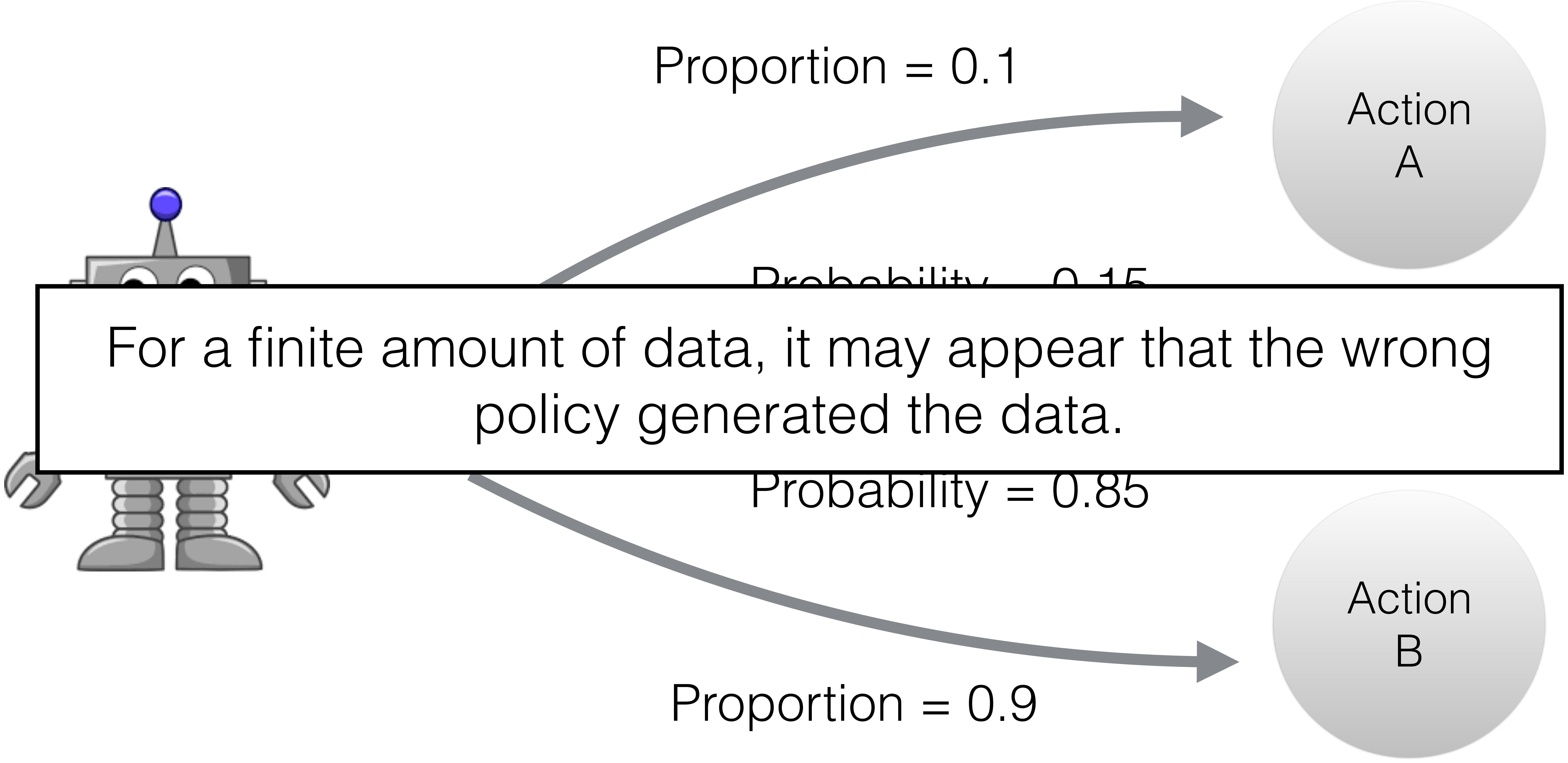
Sampling Error



Sampling Error

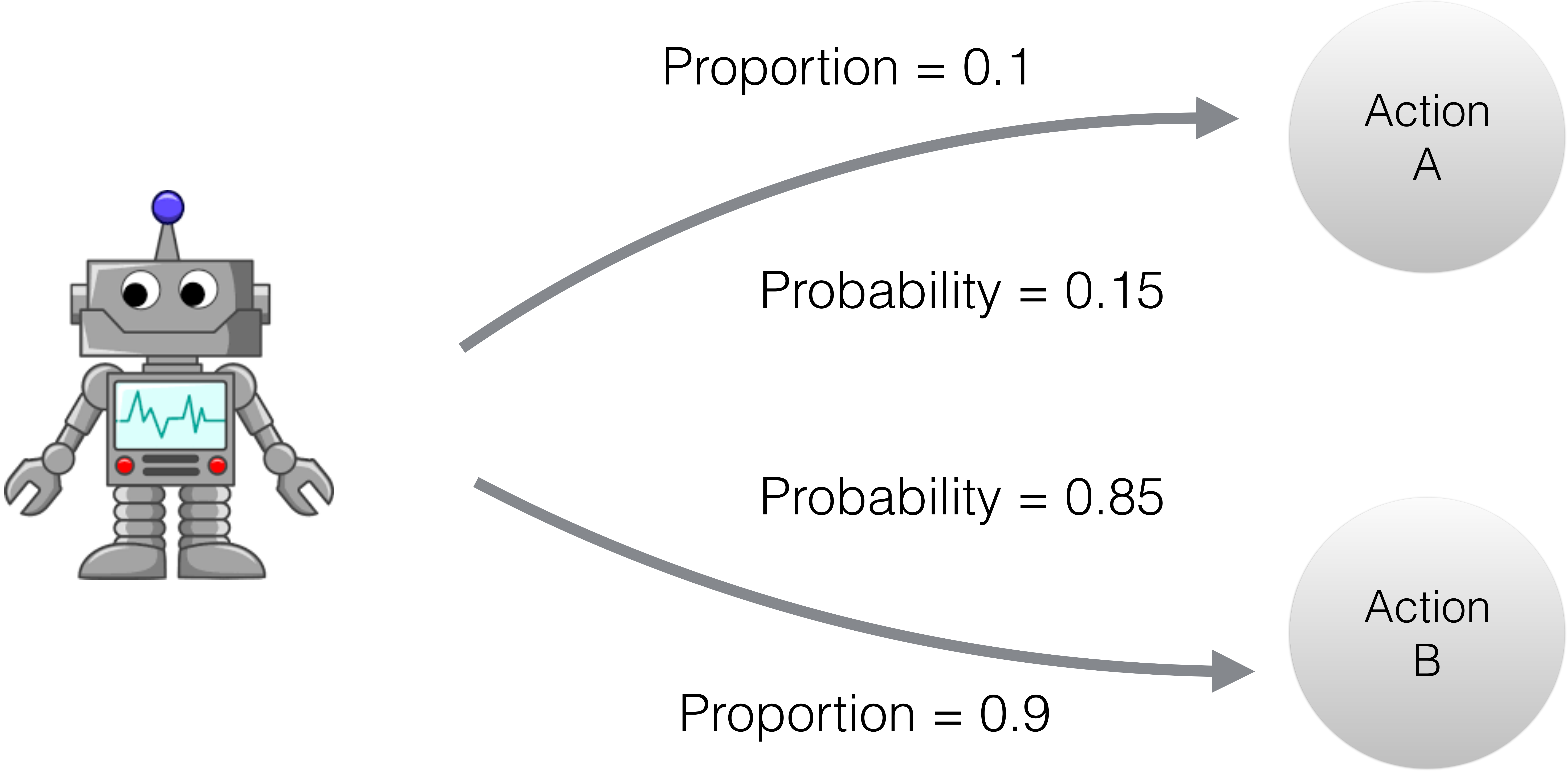


Sampling Error

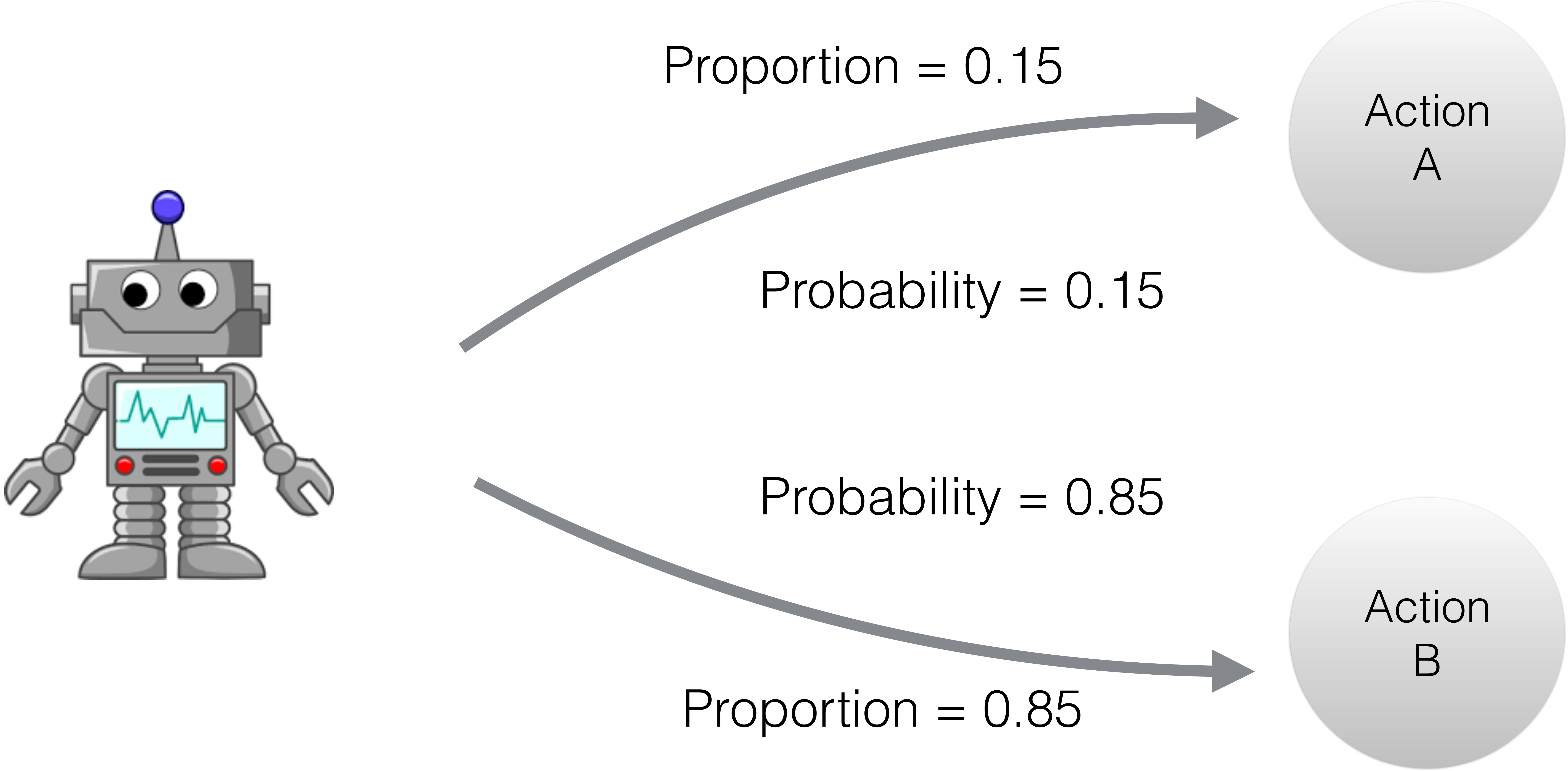


For a finite amount of data, it may appear that the wrong policy generated the data.

Sampling Error



Sampling Error



Correcting Sampling Error

Correcting Sampling Error

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

Correcting Sampling Error

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

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

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 know actually took actions.

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 know actually took actions.

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

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 know actually took actions.

Importance Sampling Correction

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

Sampling Error Corrected Policy Gradient

Sampling Error Corrected Policy Gradient

1. Execute current policy for m steps.

Sampling Error Corrected Policy Gradient

1. Execute current policy for m steps.
2. Estimate empirical policy with maximum likelihood estimation.

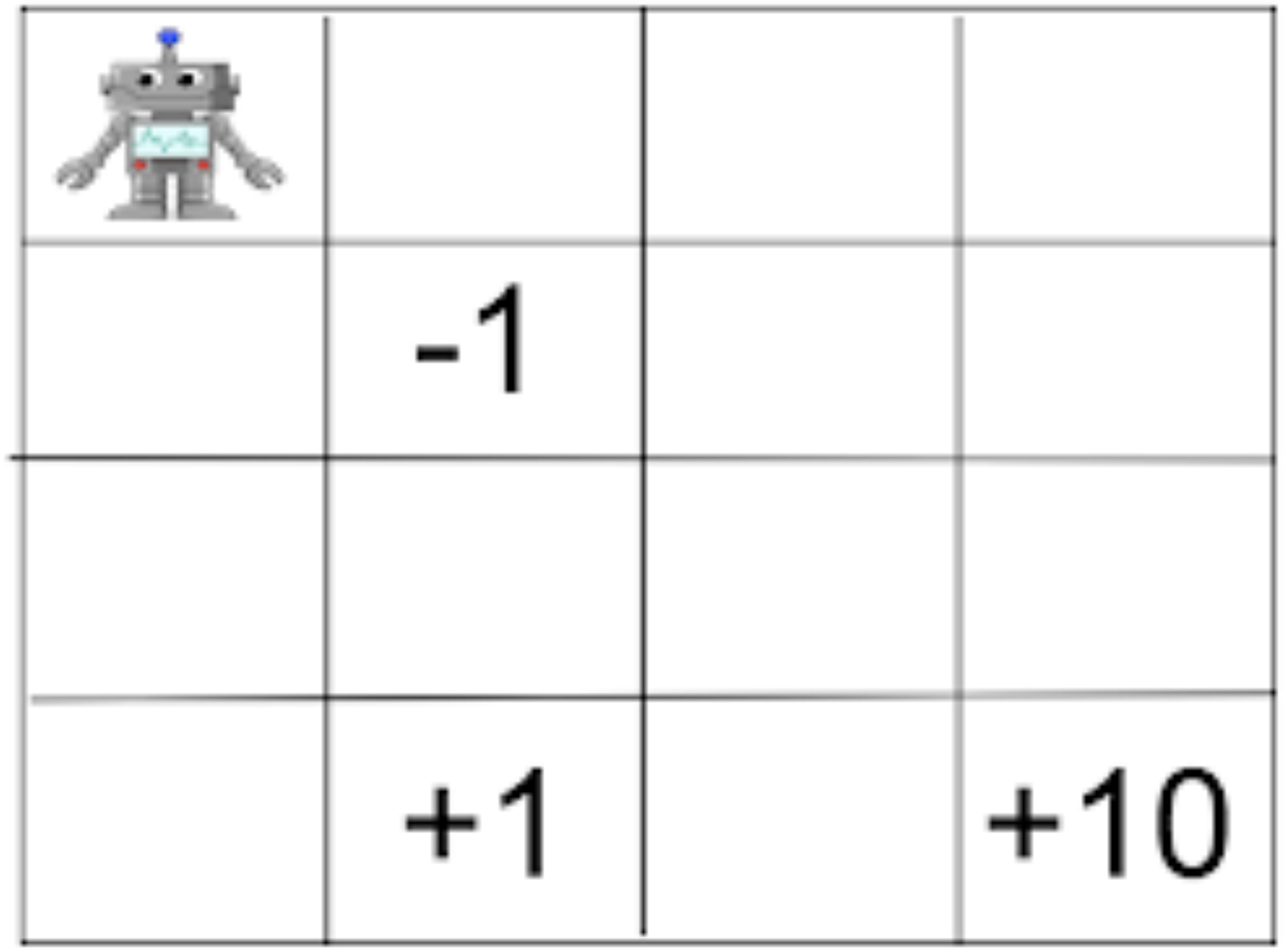
Sampling Error Corrected Policy Gradient

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.

Sampling Error Corrected Policy Gradient

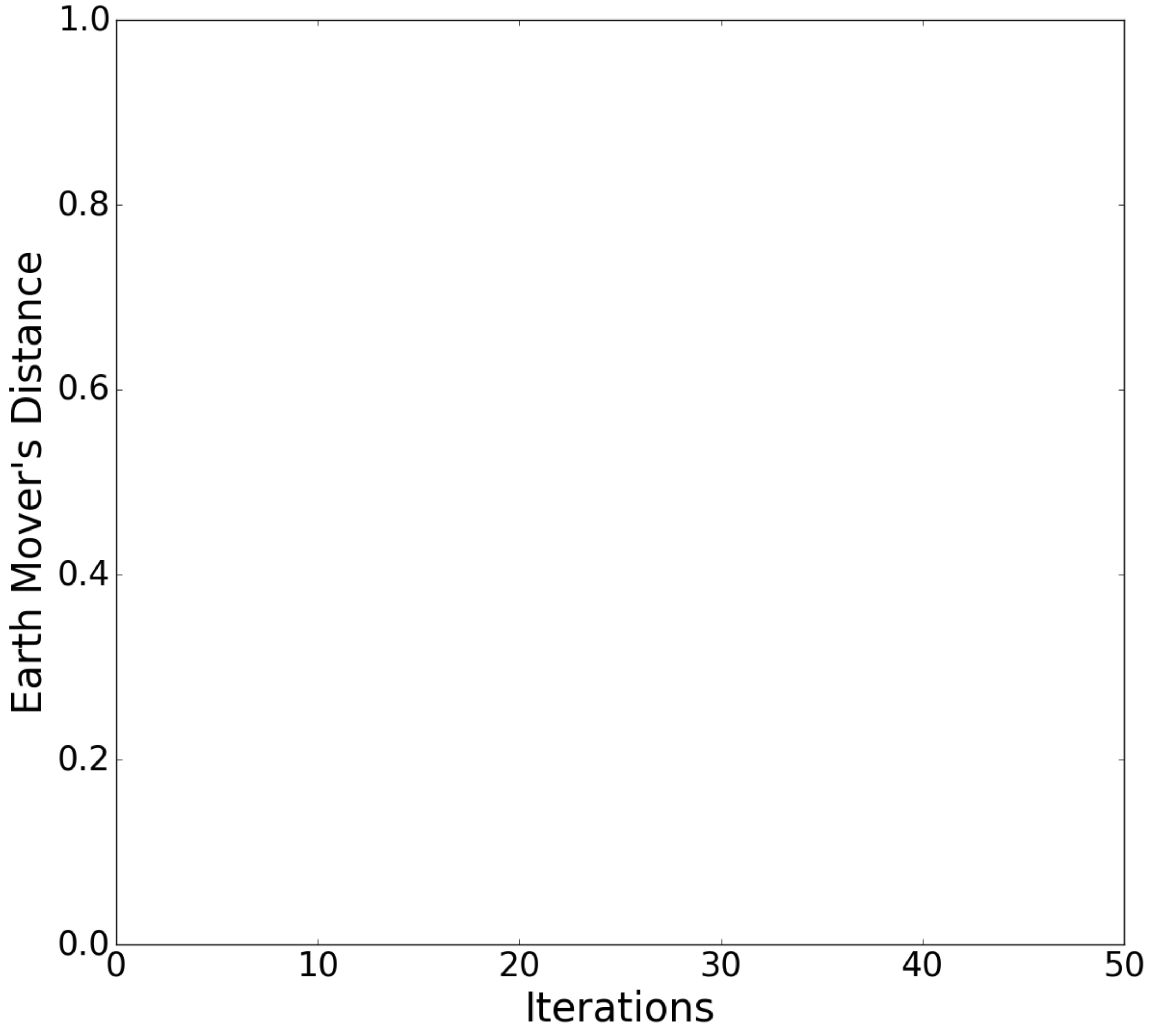
1. Execute current policy for m steps.
2. Estimate empirical policy with maximum likelihood estimation.
3. Update policy with Sampling Error Corrected (SEC) policy gradient estimate.
4. Throw away data and repeat (on-policy).

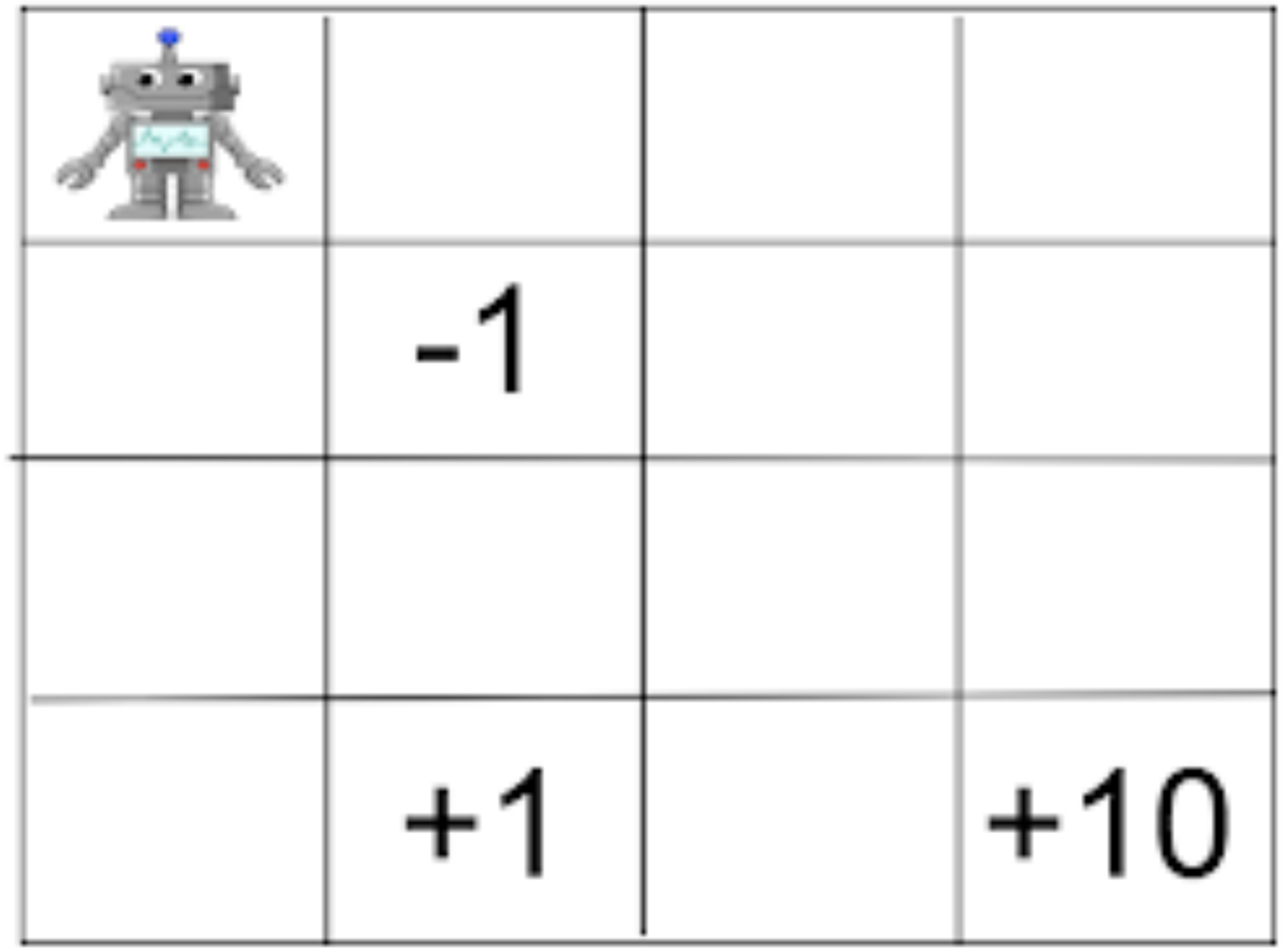
Empirical Results



GridWorld
Discrete State and Actions

Empirical Results



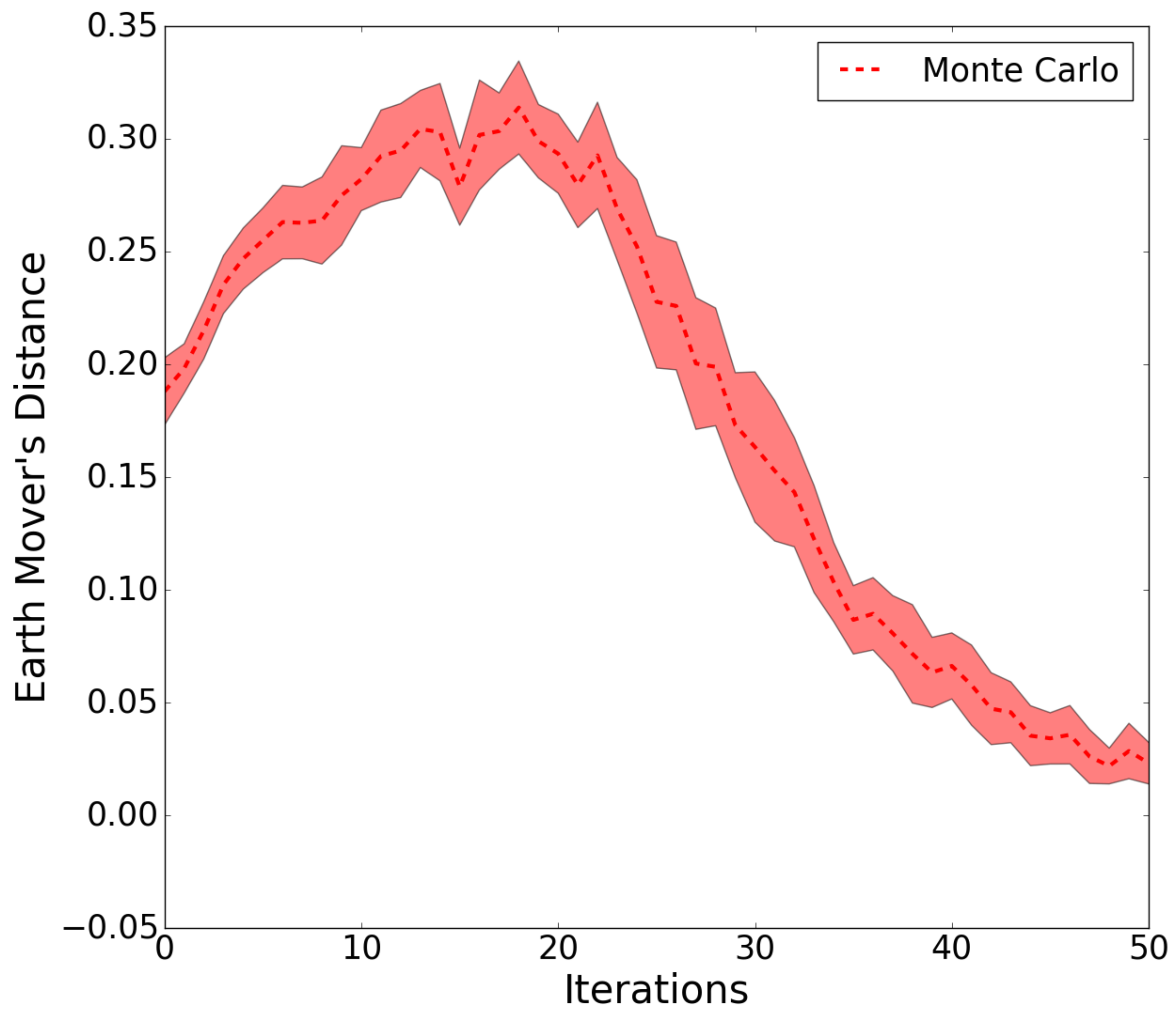


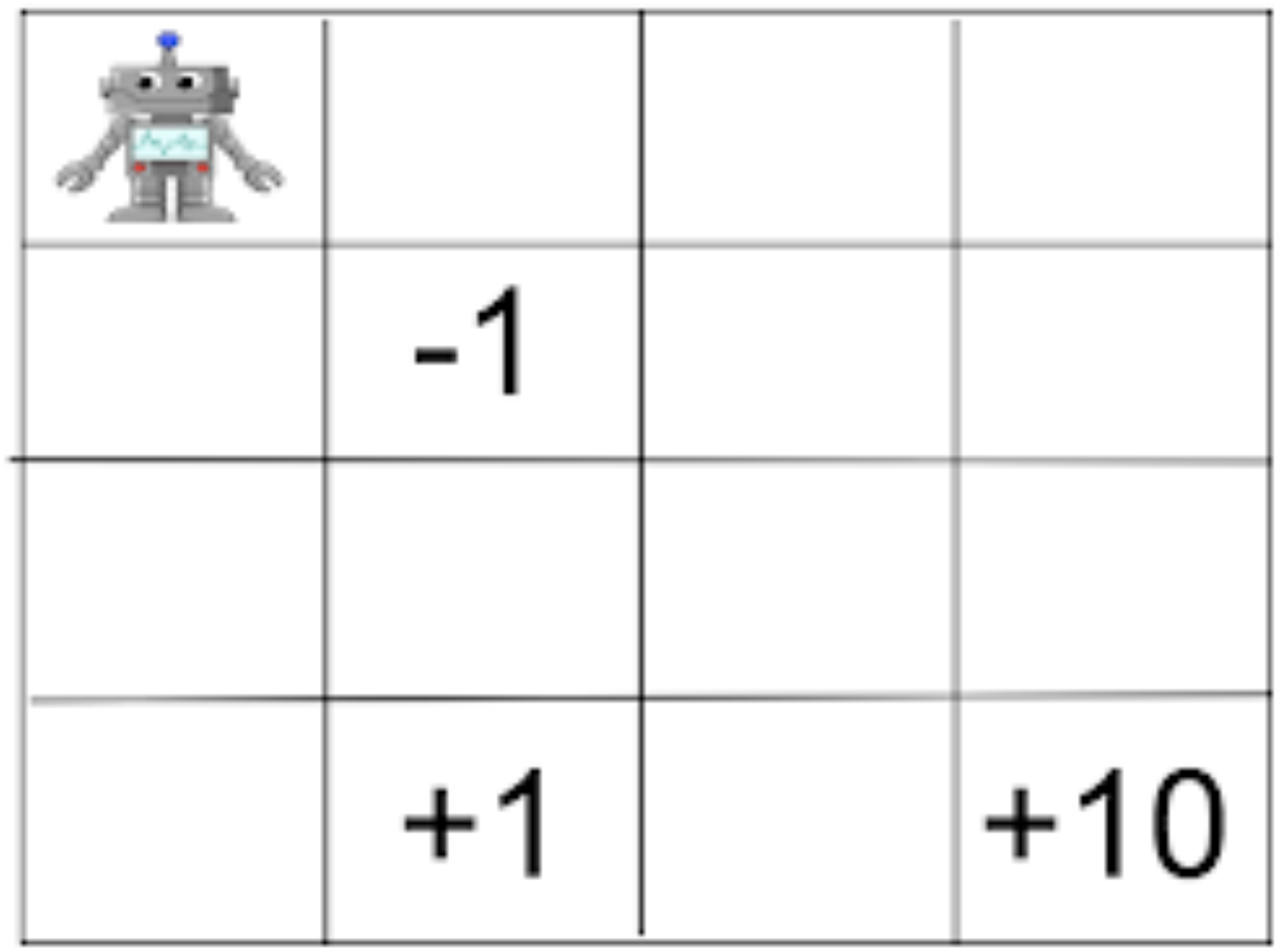
GridWorld

Discrete State and Actions

Josiah Hanna

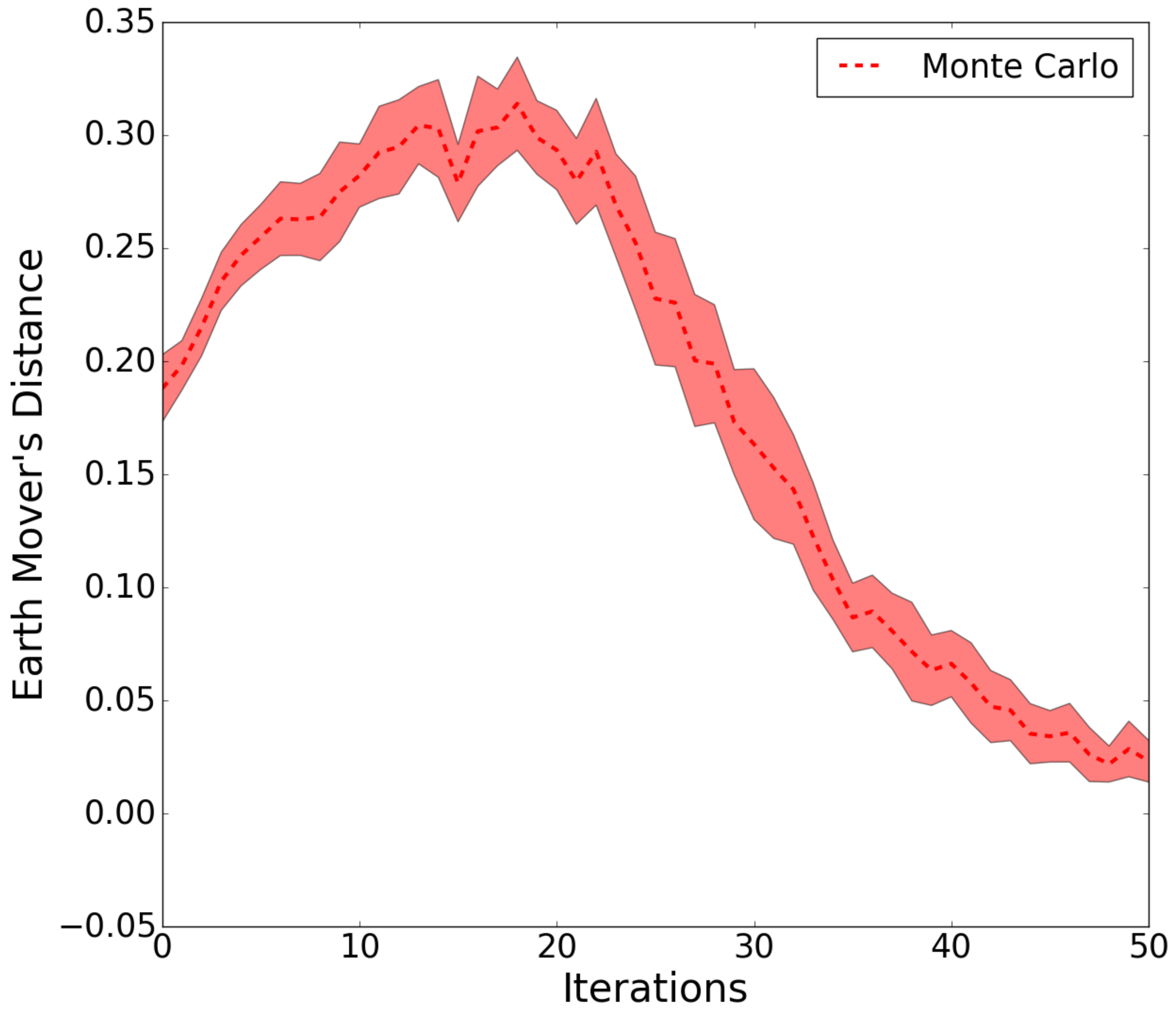
Empirical Results

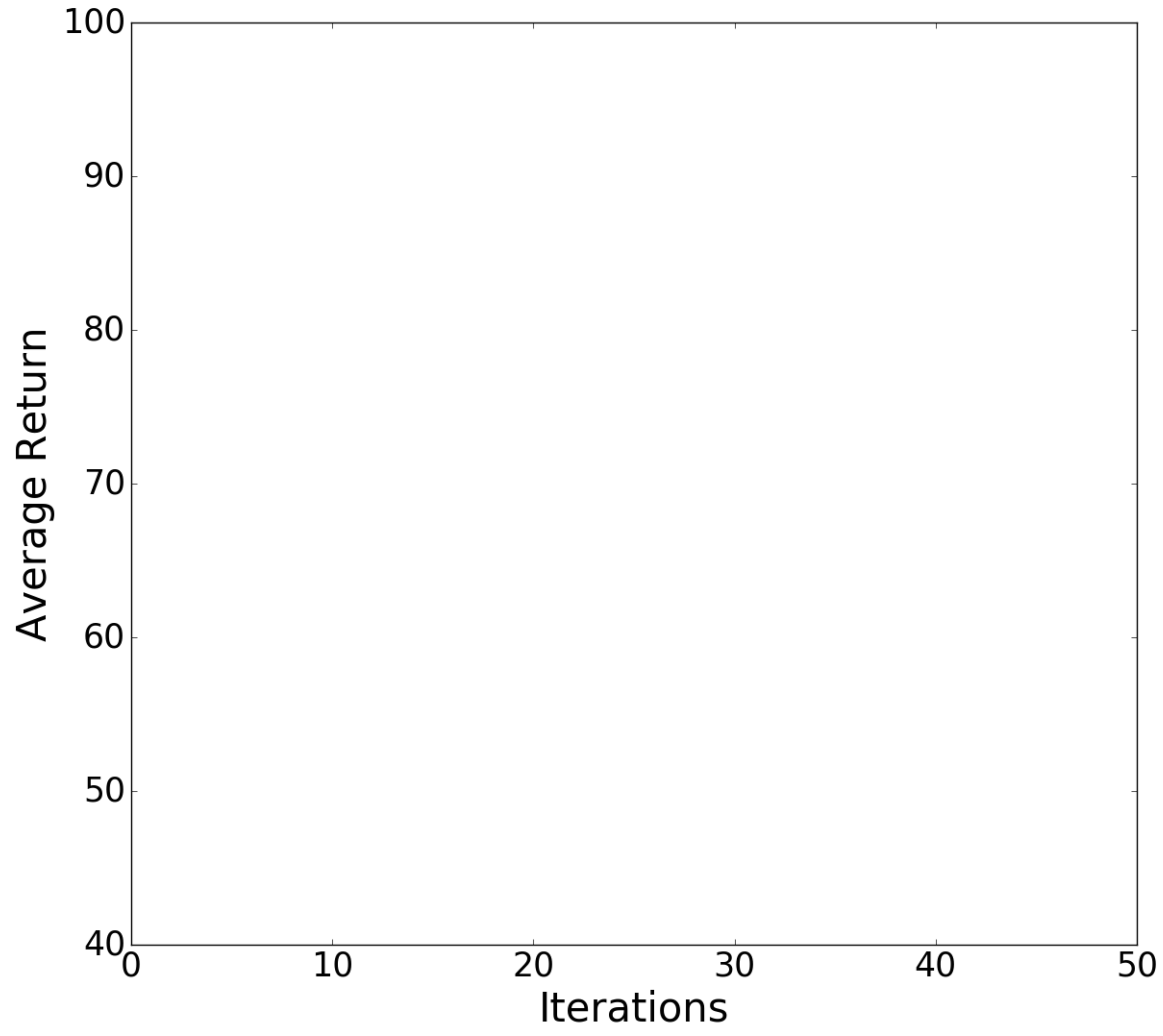




GridWorld
Discrete State and Actions

Empirical Results



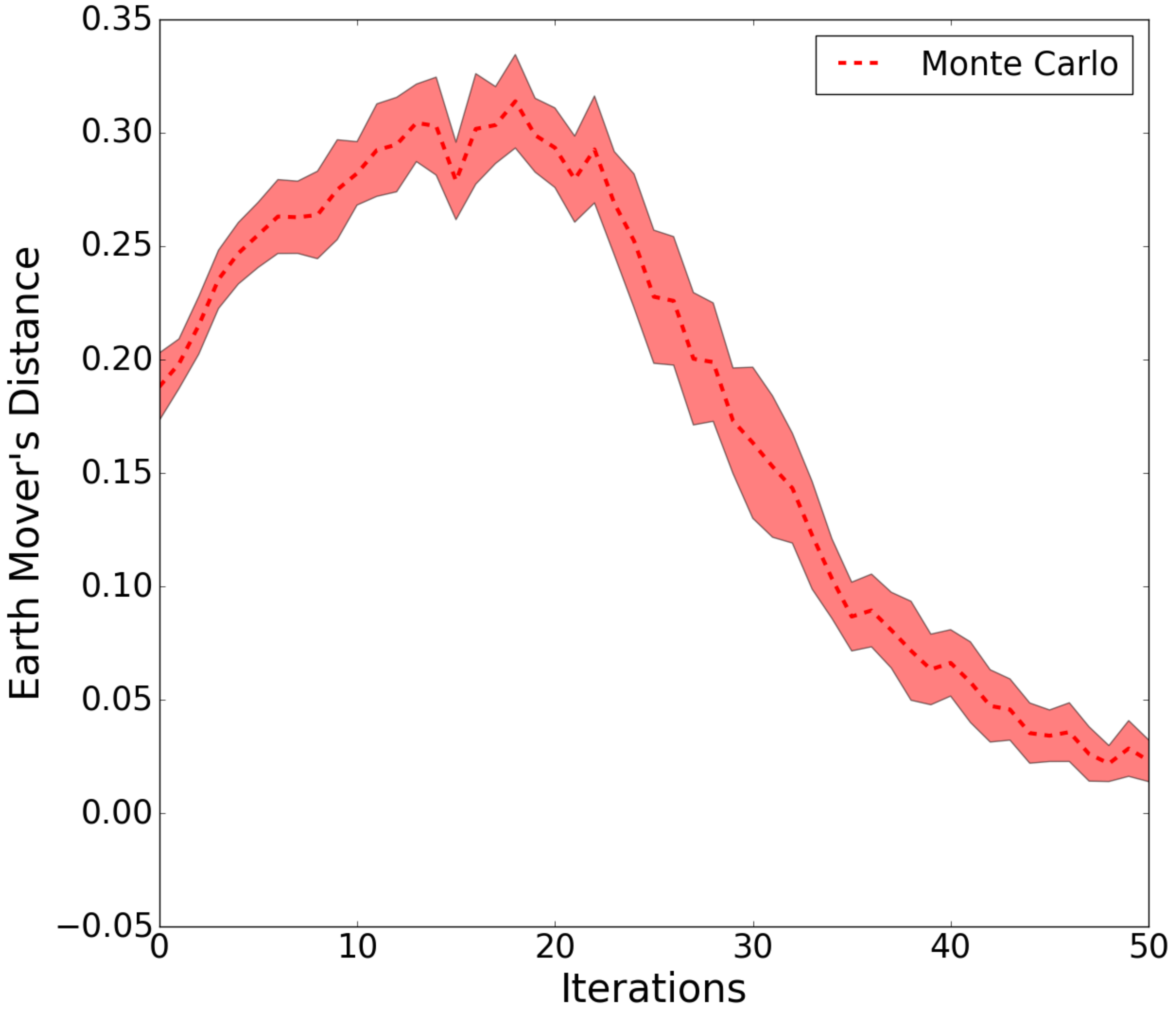


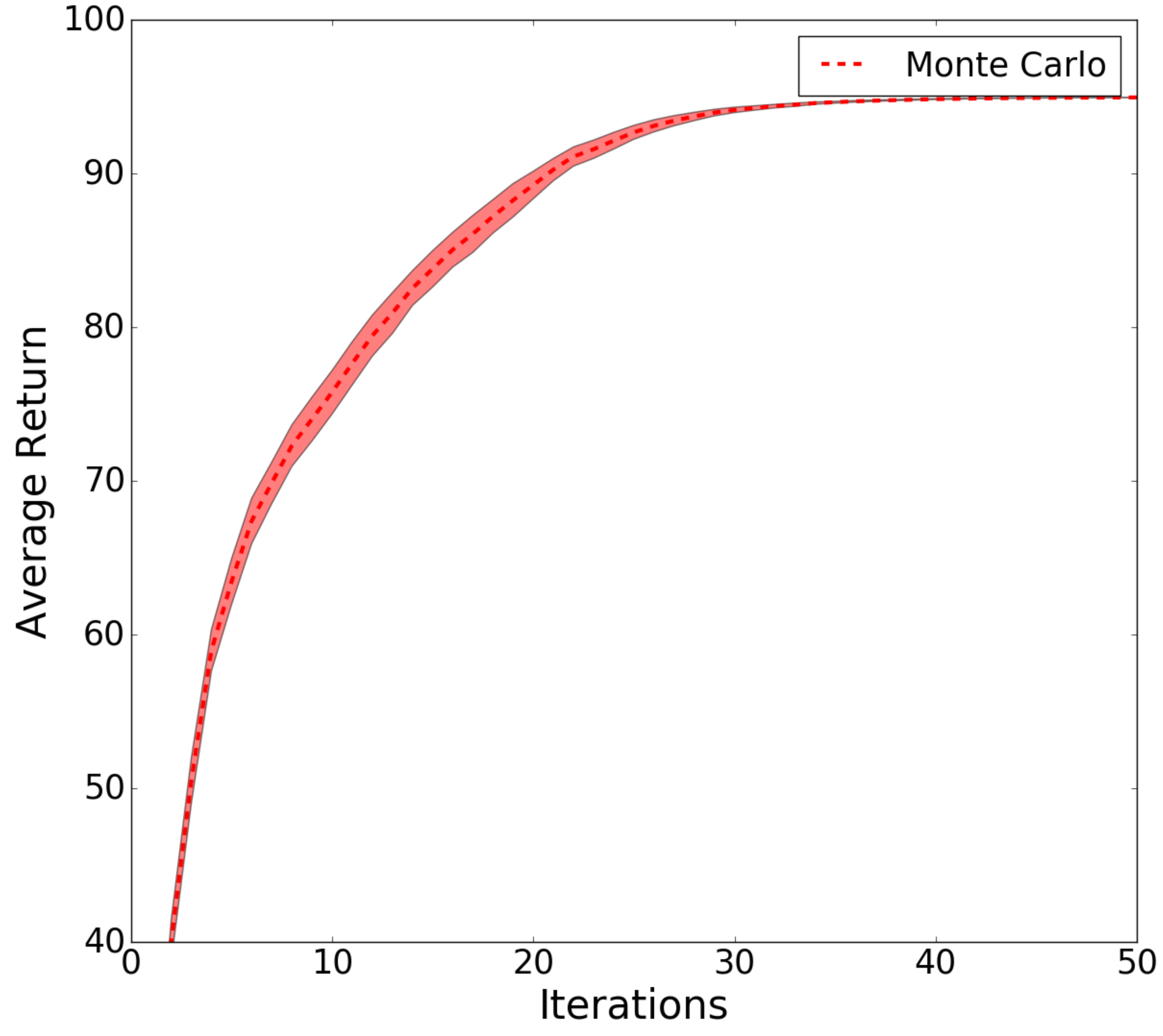
GridWorld

Discrete State and Actions

Josiah Hanna

Empirical Results

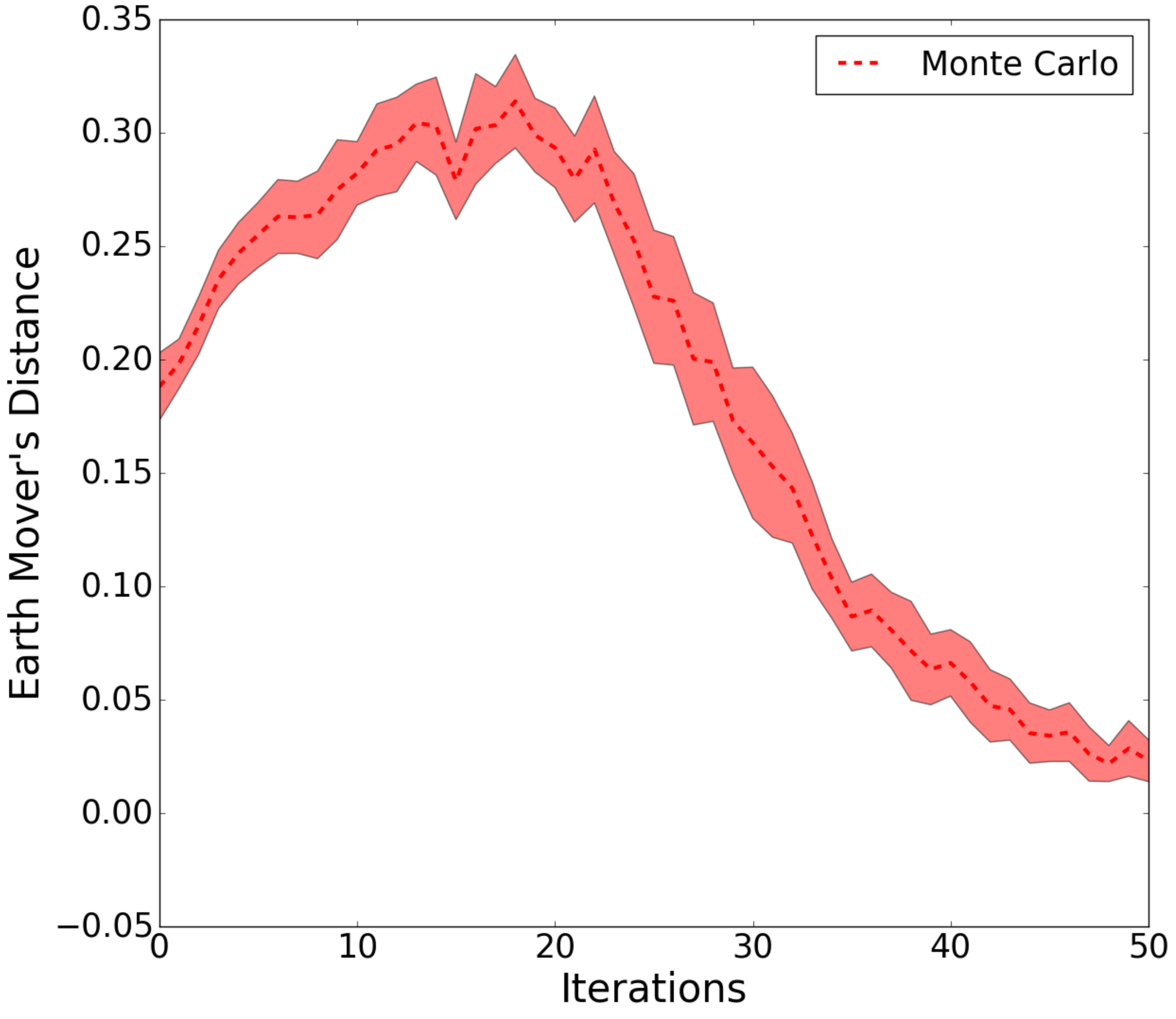


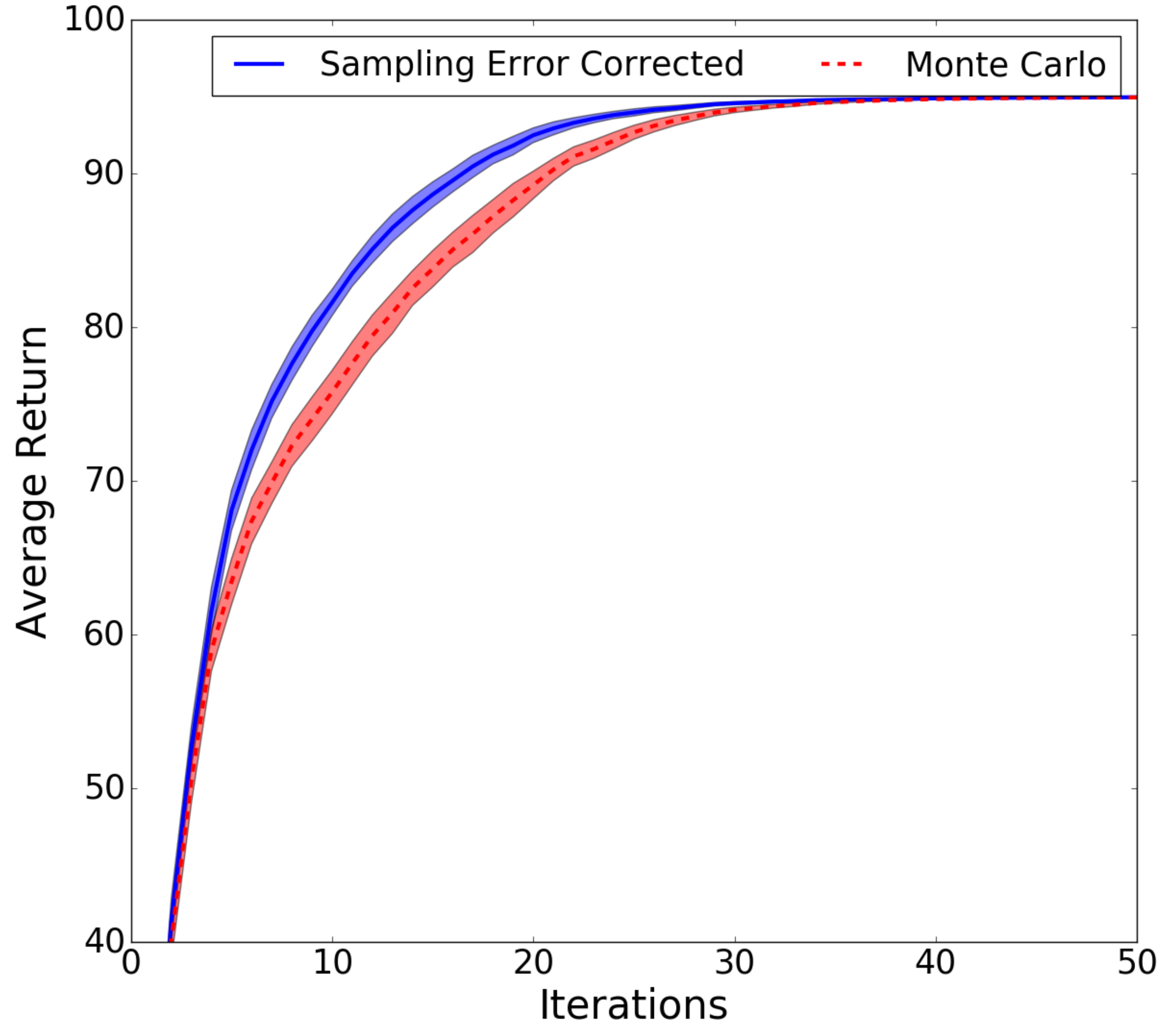


GridWorld

Discrete State and Actions

Empirical Results





GridWorld
Discrete State and Actions

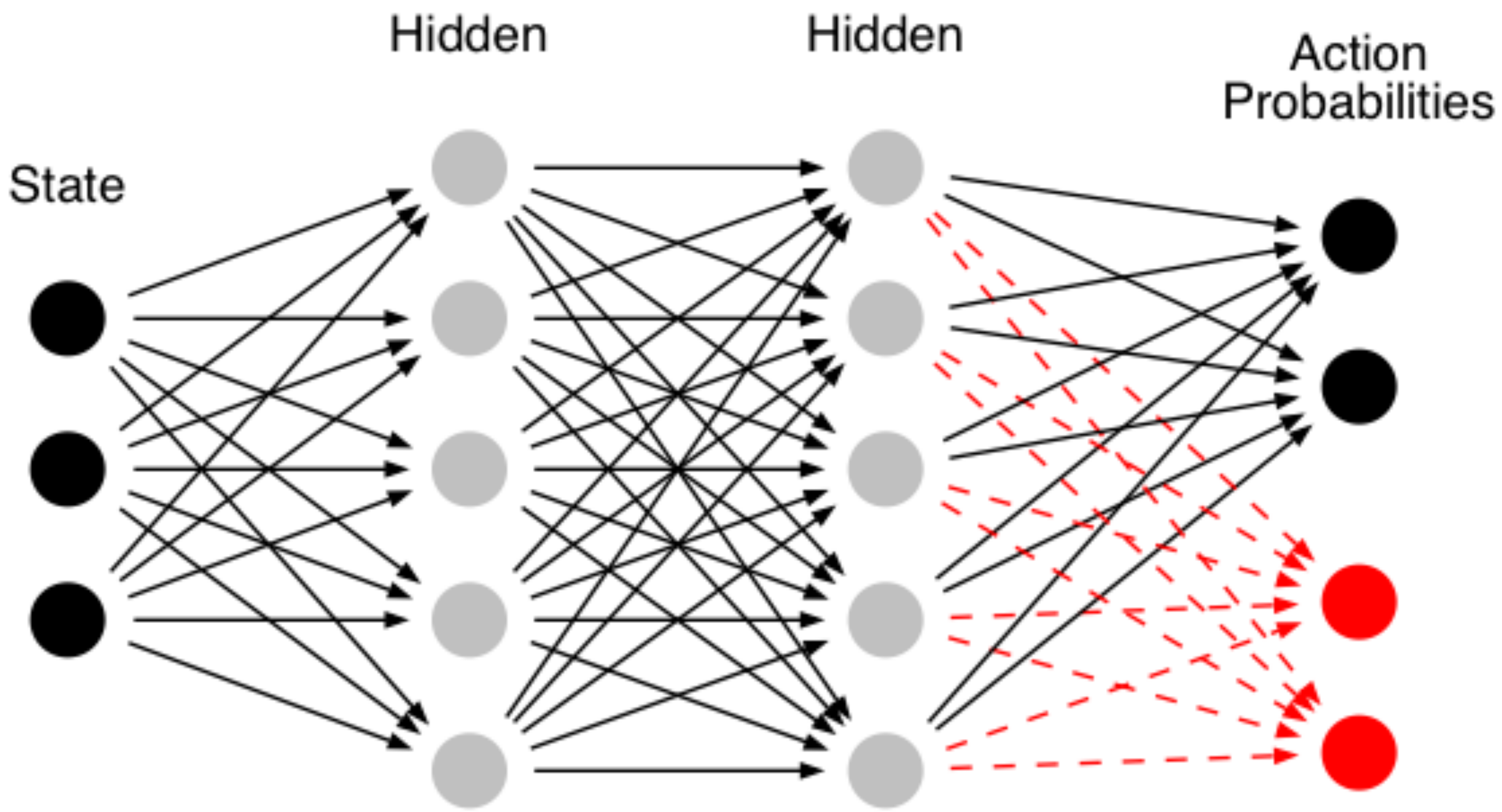
Josiah Hanna

Empirical Results

Cartpole

Continuous state and discrete actions

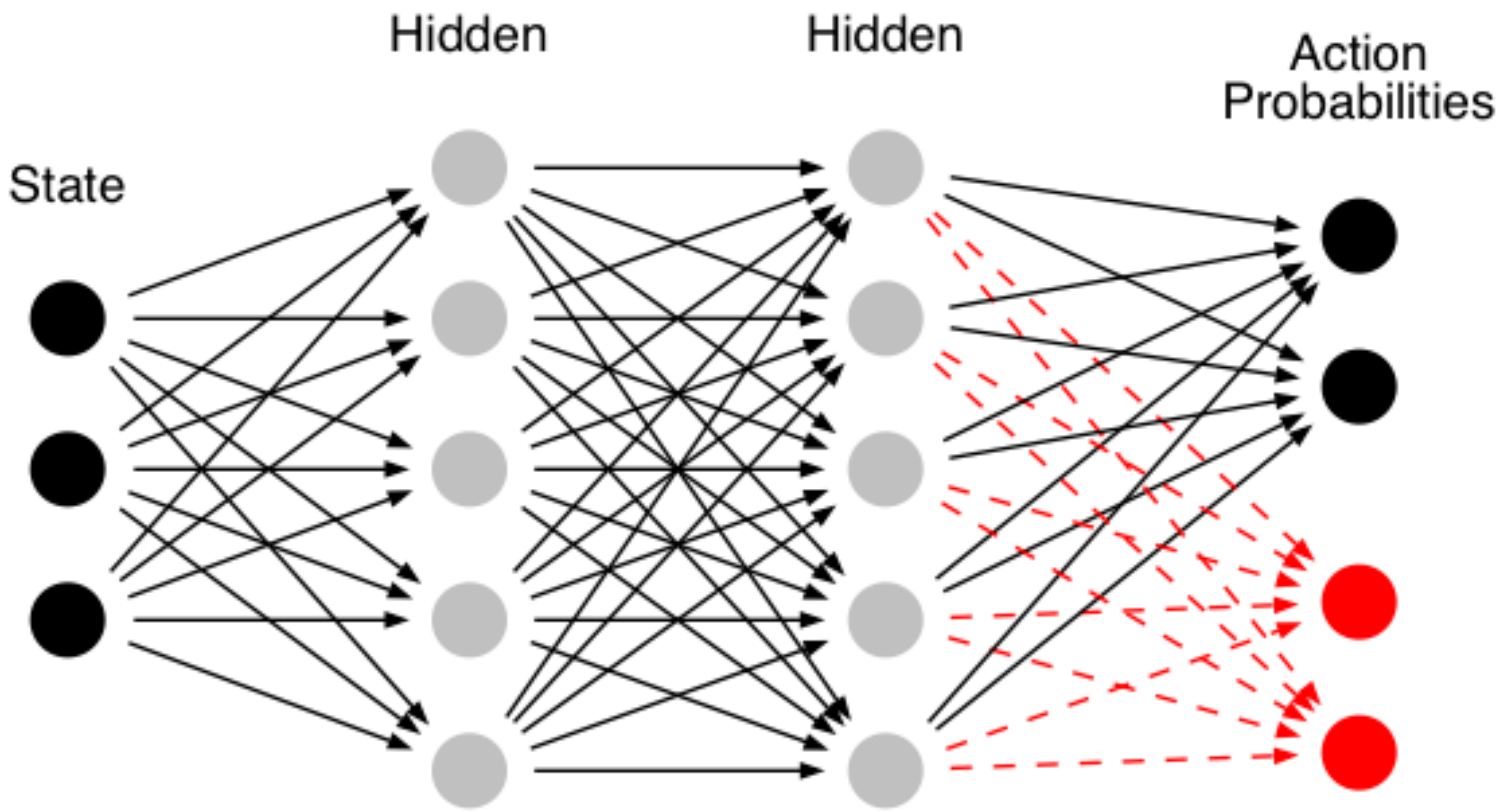
Empirical Results

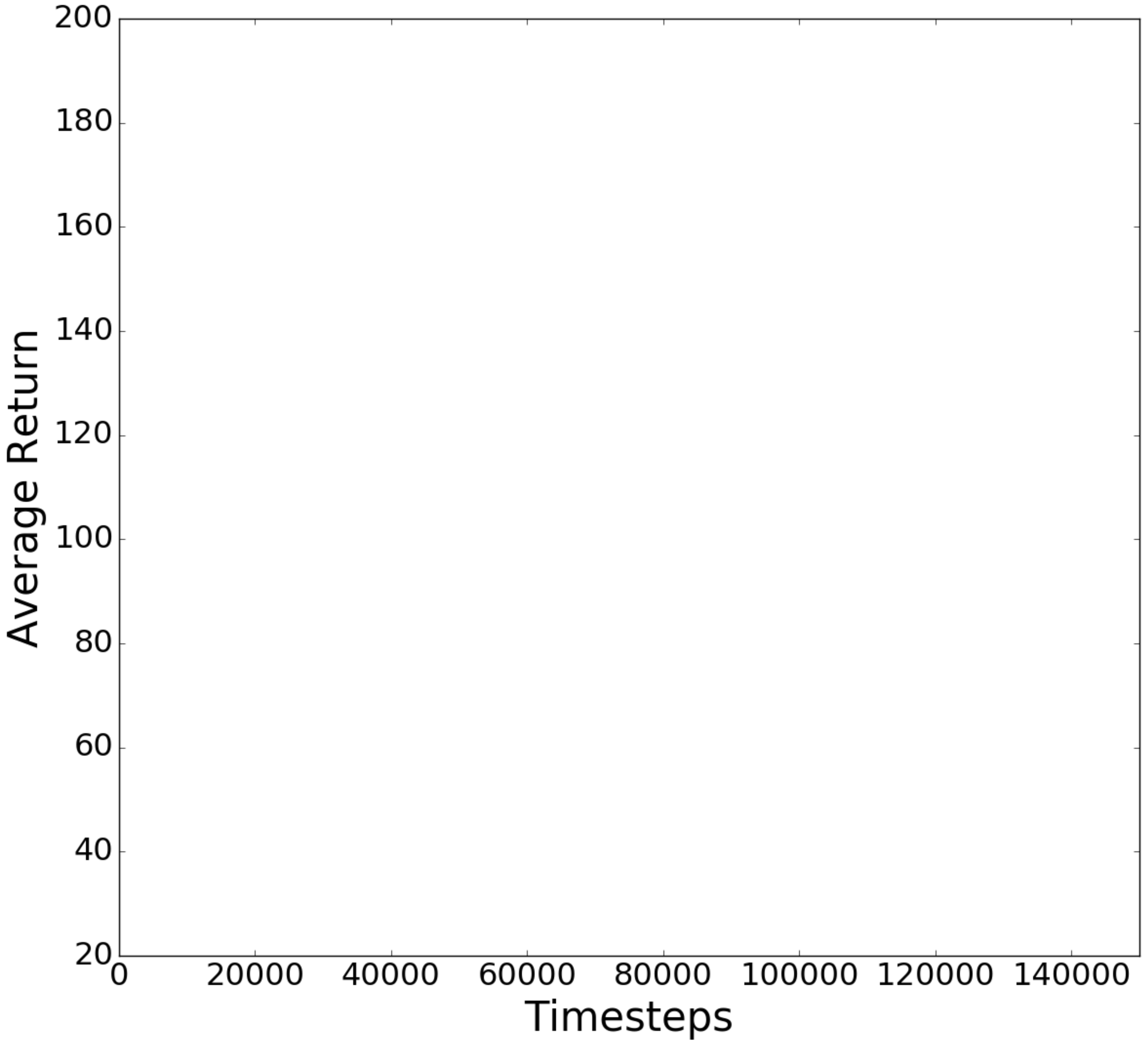


Cartpole

Continuous state and discrete actions

Empirical Results



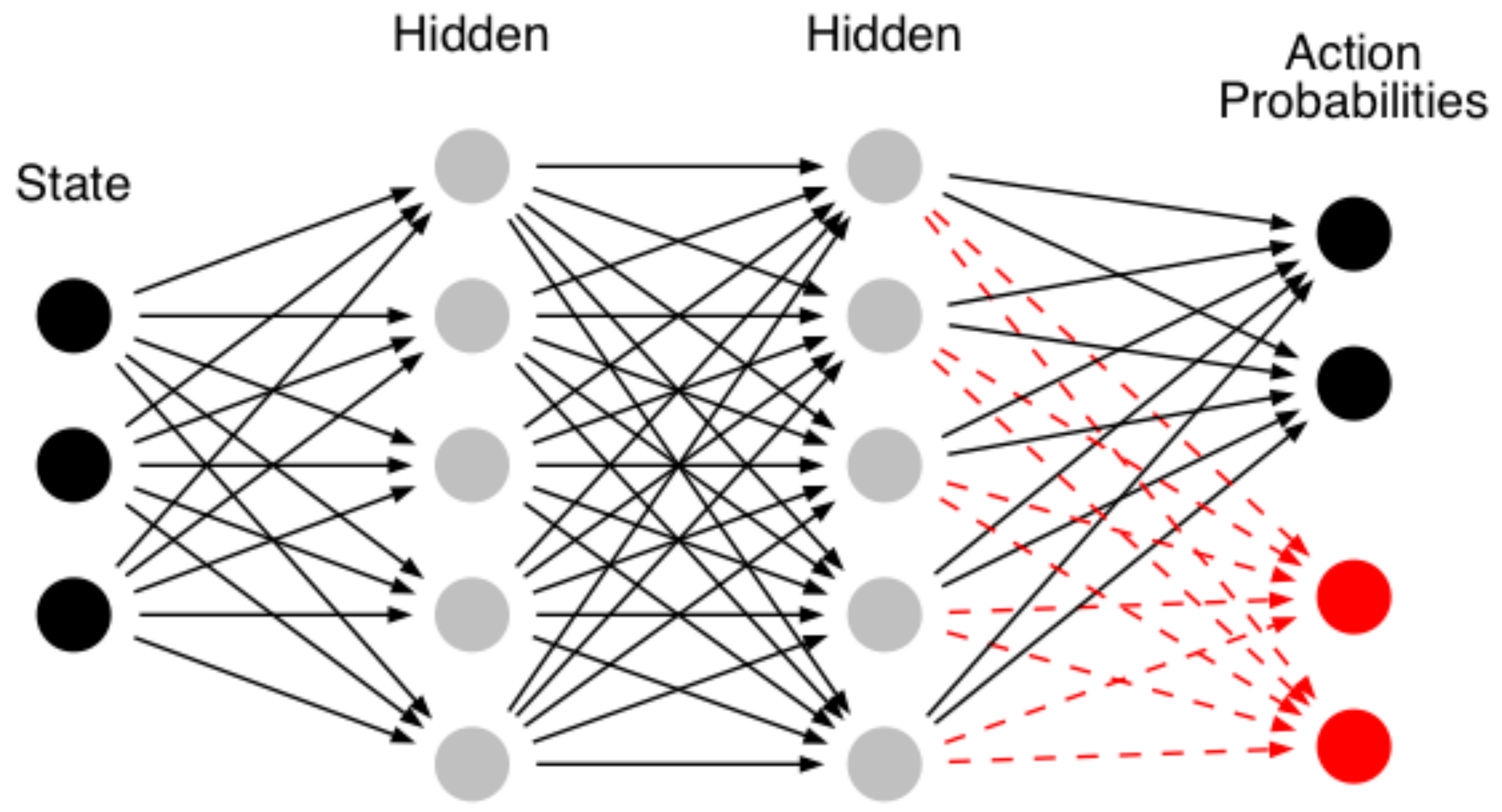


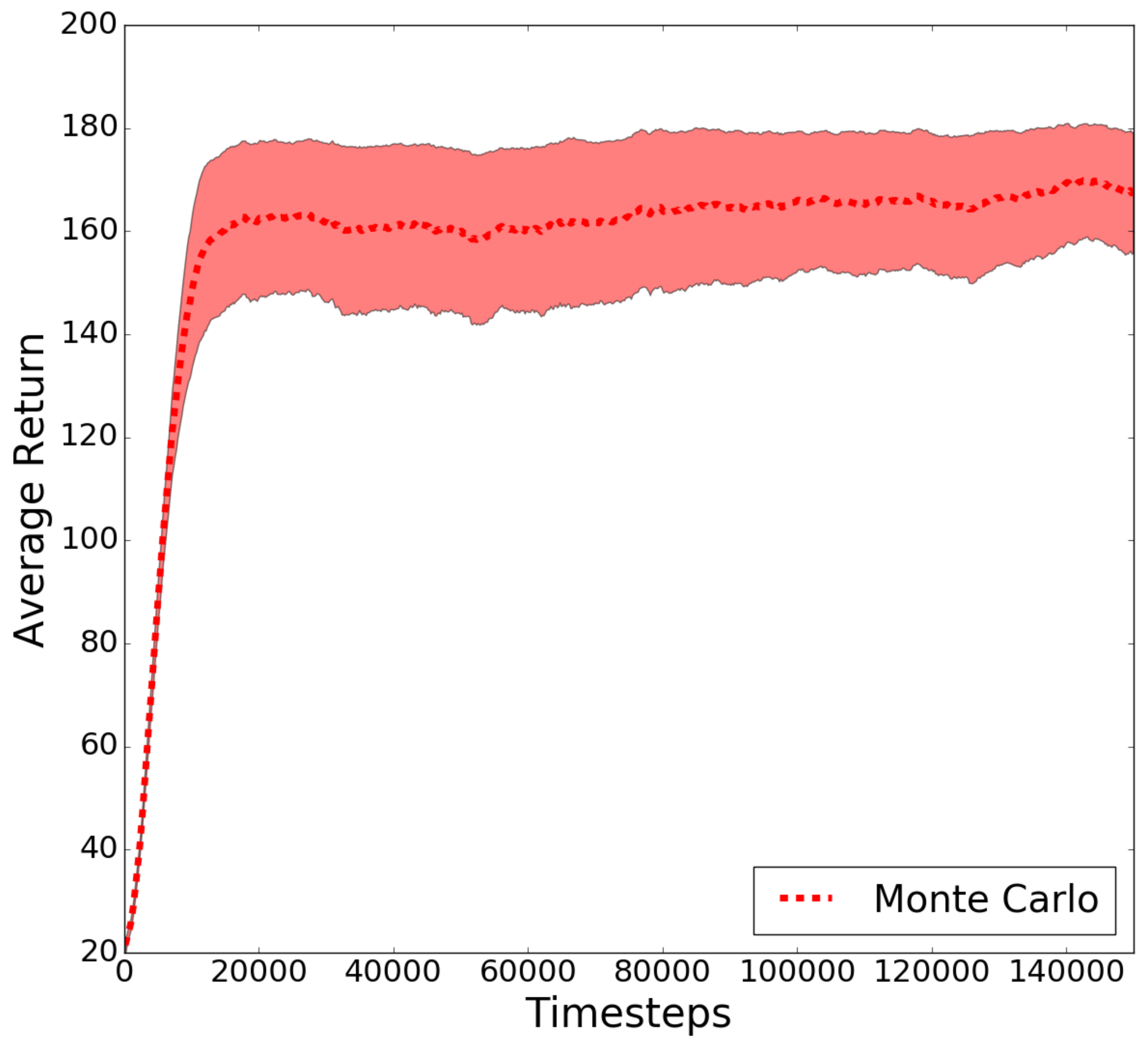
Cartpole

Continuous state and discrete actions

Josiah Hanna

Empirical Results



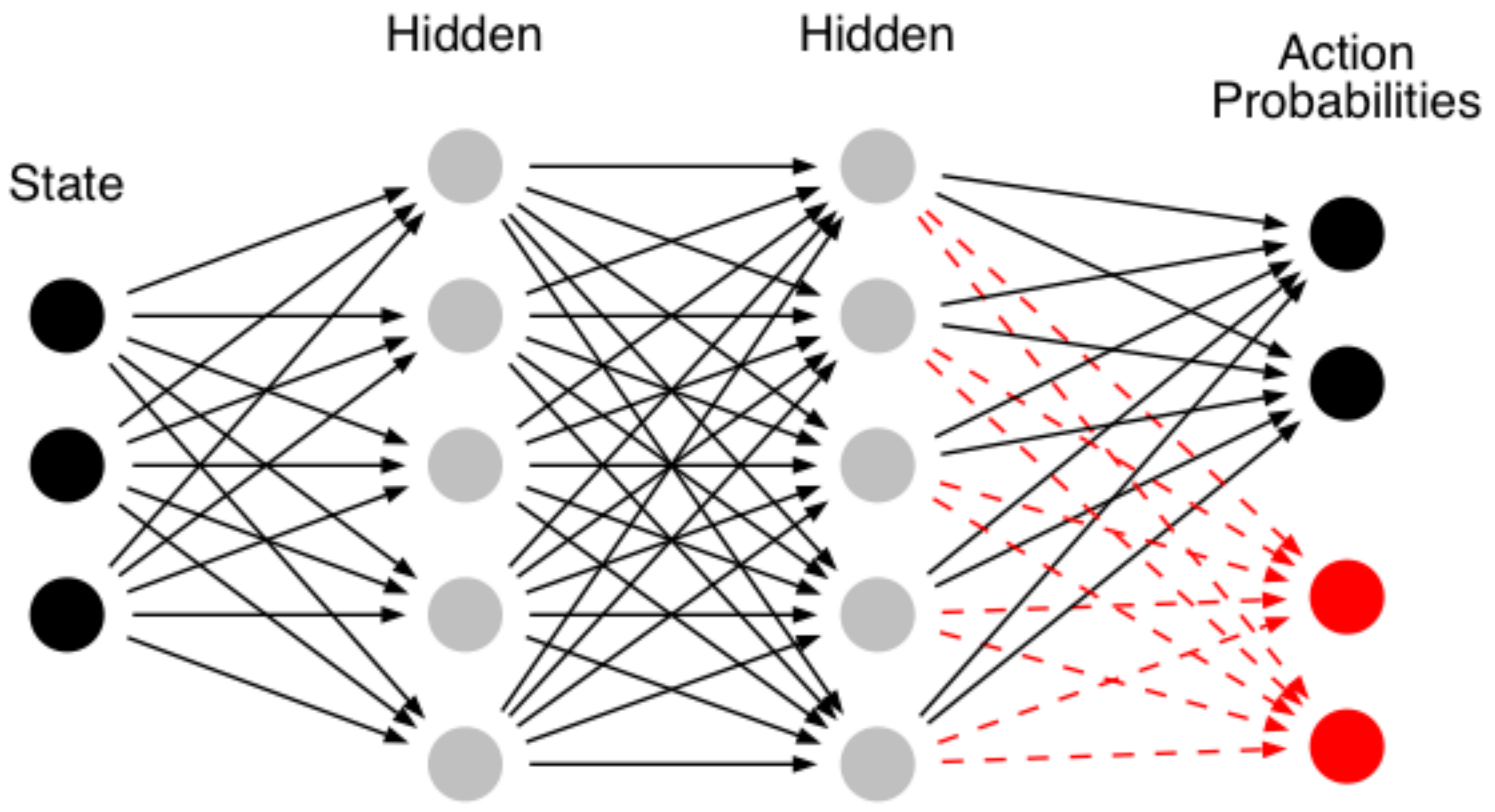


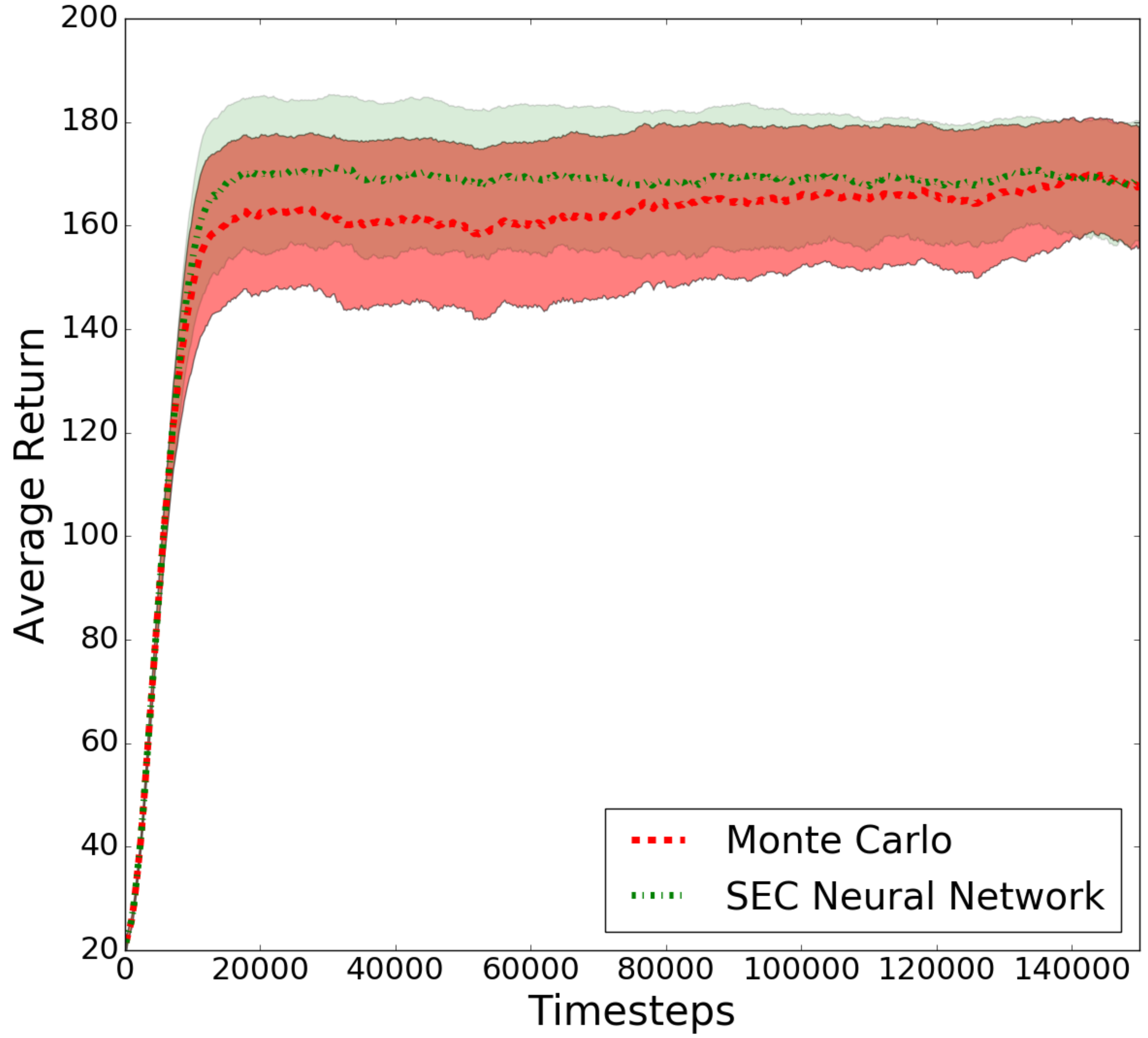
Cartpole

Continuous state and discrete actions

Josiah Hanna

Empirical Results

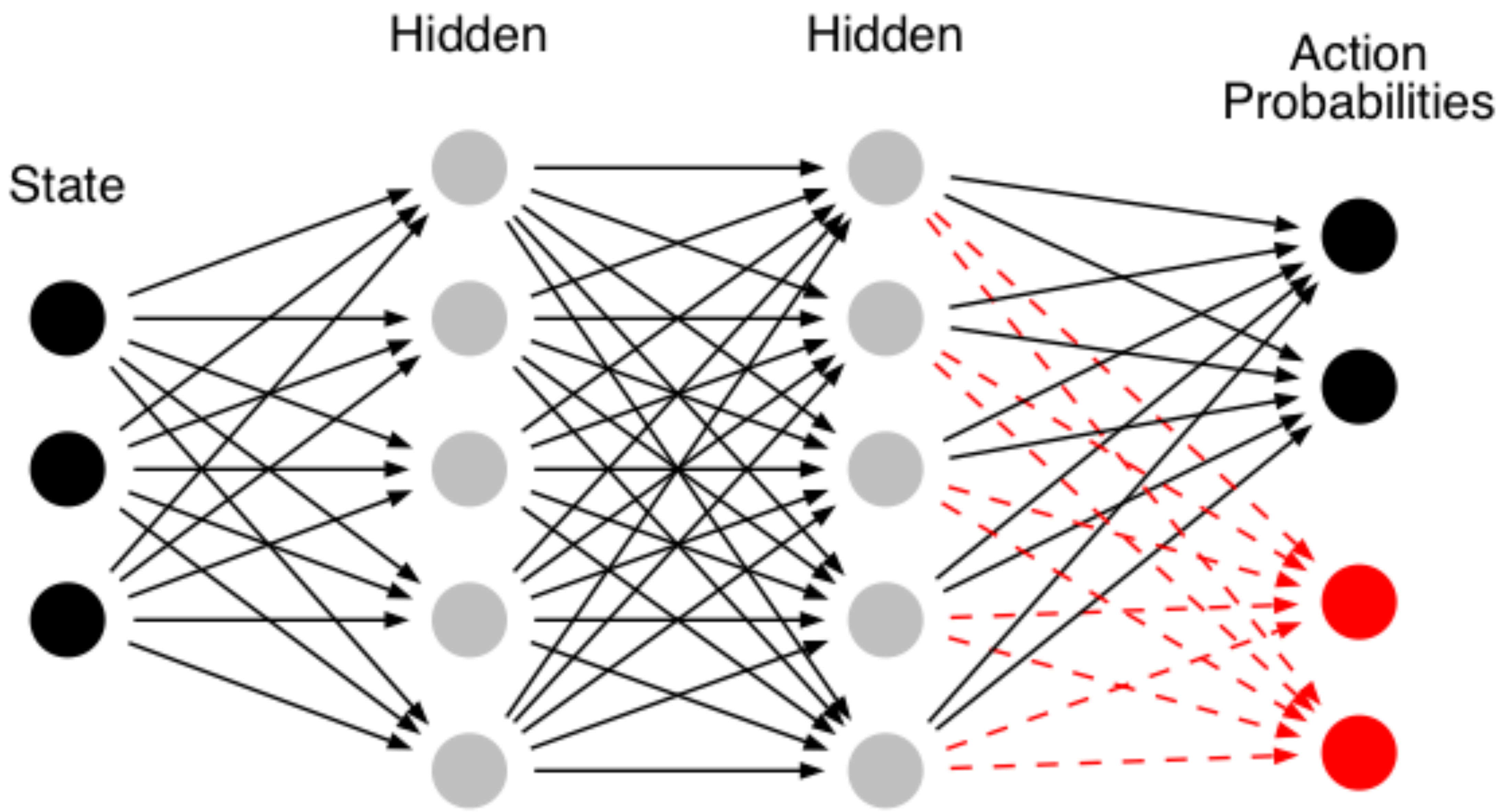


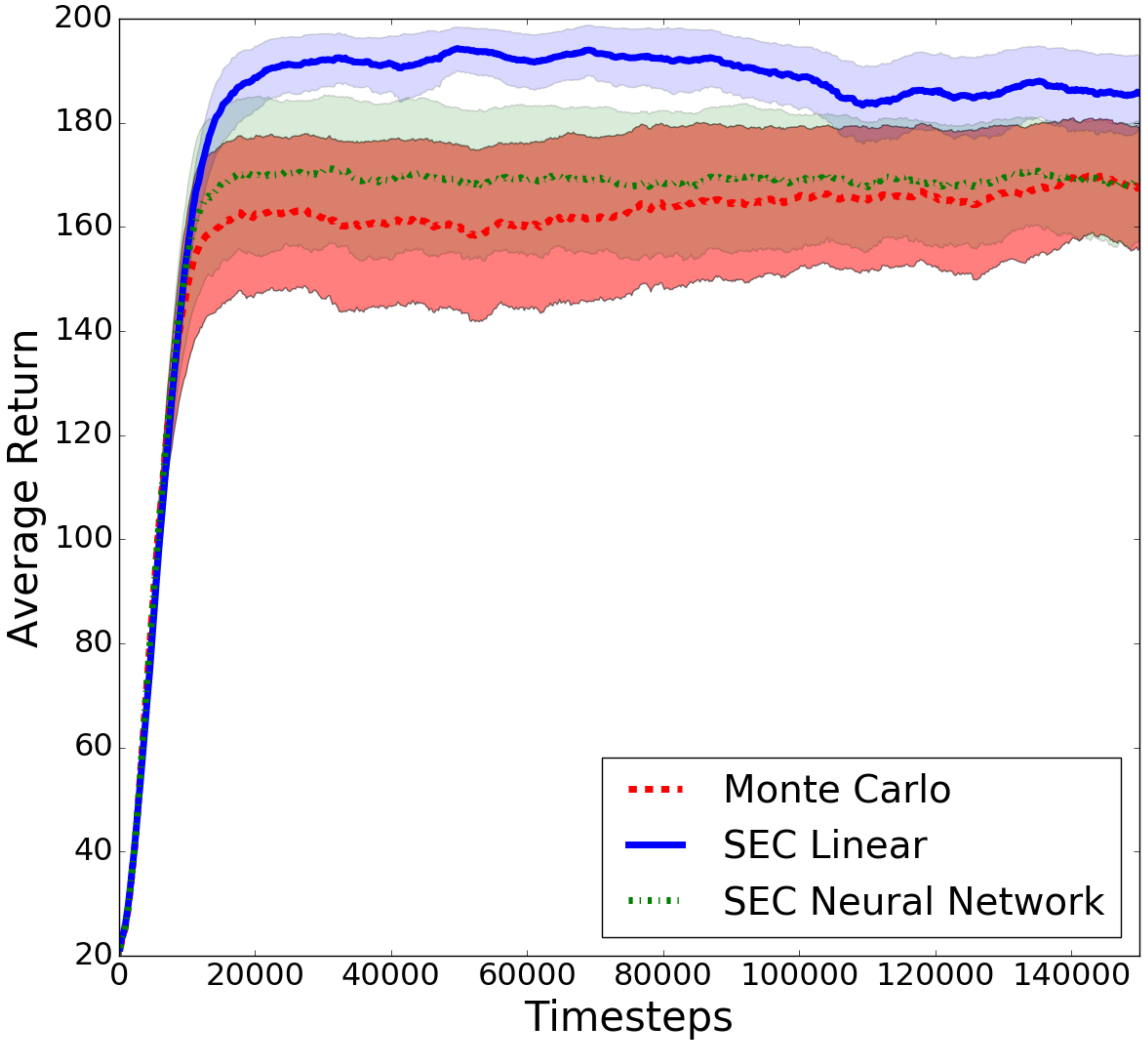


Cartpole

Continuous state and discrete actions

Empirical Results





Cartpole

Continuous state and discrete actions

Josiah Hanna

Related Work

Related Work

1. Expected SARSA (van Seijen et al. 2009).

Related Work

1. Expected SARSA (van Seijen et al. 2009).
2. Tree back-up methods (Precup et al. 2000, Asis et al. 2018).

Related Work

1. Expected SARSA (van Seijen et al. 2009).
2. Tree back-up methods (Precup et al. 2000, Asis et al. 2018).
3. Expected Policy Gradients (Ciosek and Whiteson 2018).

Related Work

1. Expected SARSA (van Seijen et al. 2009).
2. Tree back-up methods (Precup et al. 2000, Asis et al. 2018).
3. Expected Policy Gradients (Ciosek and Whiteson 2018).
4. Estimated Propensity Scores (Hirano et al. 2003, Li et al. 2015).

Related Work

1. Expected SARSA (van Seijen et al. 2009).
2. Tree back-up methods (Precup et al. 2000, Asis et al. 2018).
3. Expected Policy Gradients (Ciosek and Whiteson 2018).
4. Estimated Propensity Scores (Hirano et al. 2003, Li et al. 2015).
5. Many people outside of RL + Bandits:

Related Work

1. Expected SARSA (van Seijen et al. 2009).
2. Tree back-up methods (Precup et al. 2000, Asis et al. 2018).
3. Expected Policy Gradients (Ciosek and Whiteson 2018).
4. Estimated Propensity Scores (Hirano et al. 2003, Li et al. 2015).
5. Many people outside of RL + Bandits:
    - Blackbox importance sampling (Liu and Lee 2017), Bayesian Monte Carlo (Gharamani and Rasmussen 2003).

Weighting Off-policy Data

Weighting Off-policy Data

Contribution 3: Family of regression importance sampling estimators that improve over ordinary importance sampling.

Weighting Off-policy Data

Contribution 3: Family of regression importance sampling estimators that improve over ordinary importance sampling.

Contribution 4: Sampling error corrected policy gradient estimator that improves over Monte Carlo policy gradient estimators.

Weighting Off-policy Data

Contribution 3: Family of regression importance sampling estimators that improve over ordinary importance sampling.

Contribution 4: Sampling error corrected policy gradient estimator that improves over Monte Carlo policy gradient estimators.

Additional results in dissertation: Asymptotic variance analysis, consistency of RIS, additional experiments.

Weighting Off-policy Data

Contri

estima

Contri

that im

Take-away Message

It is better to estimate the behavior policy than use the true behavior policy.

Doing so corrects sampling error in policy value and policy gradient estimates.

Addition ... f RIS,

additional experiments.

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

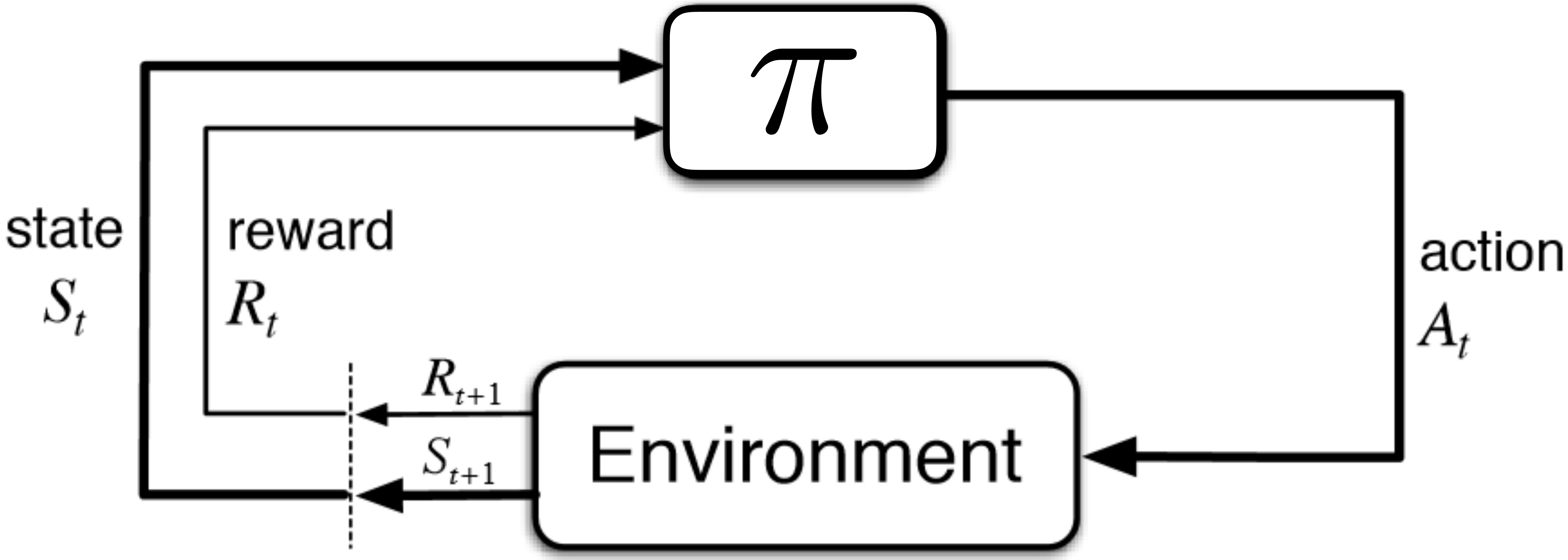
How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

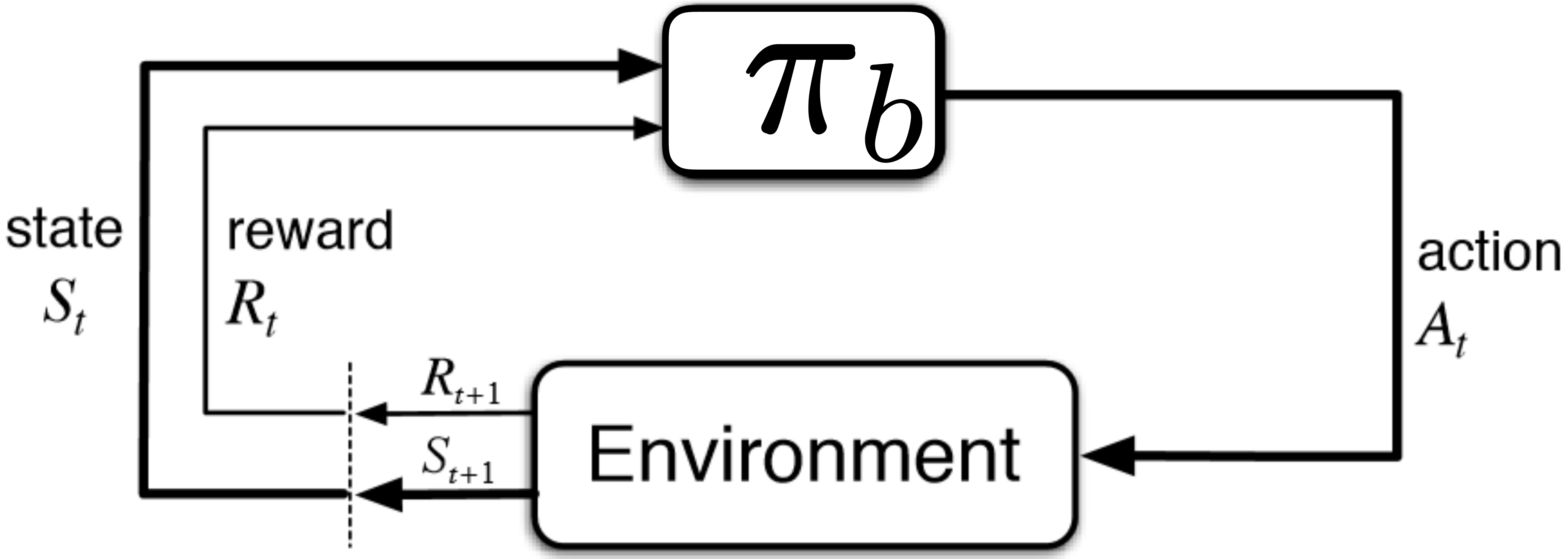
Can reinforcement learning be data efficient enough for real world applications?

Off-Environment RL



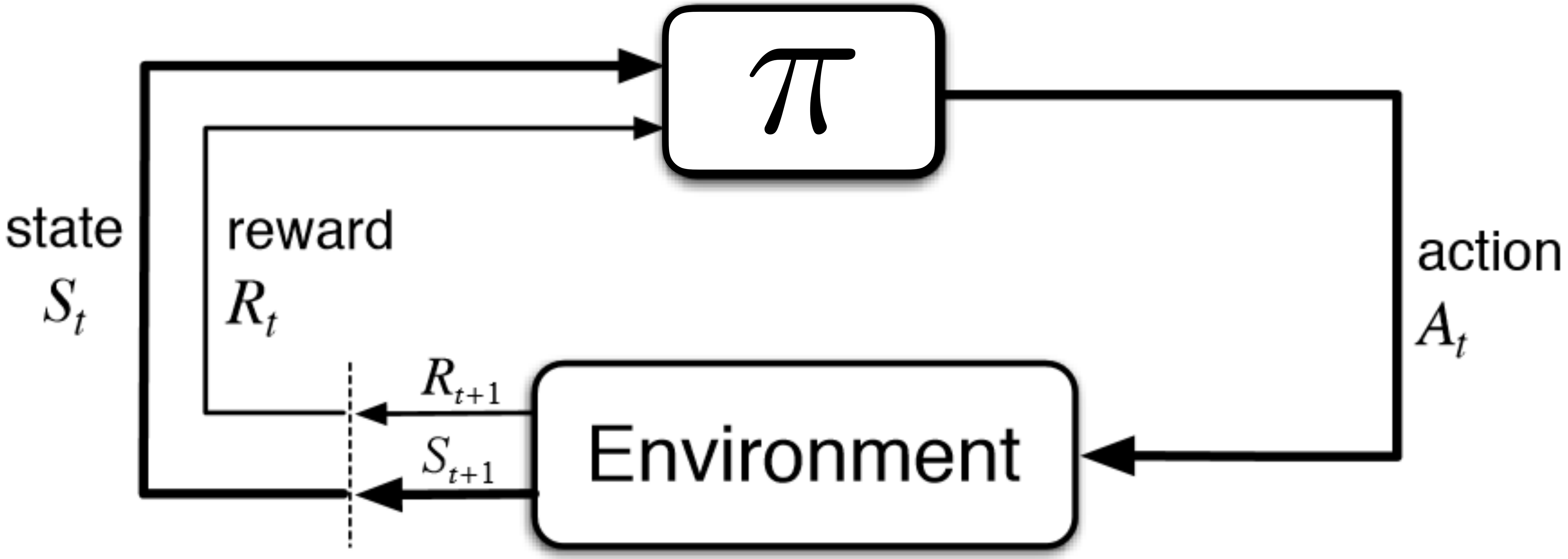
$$S_0, A_0, R_0, S_1, \ldots, S_L, A_L, R_L$$

Off-Environment RL



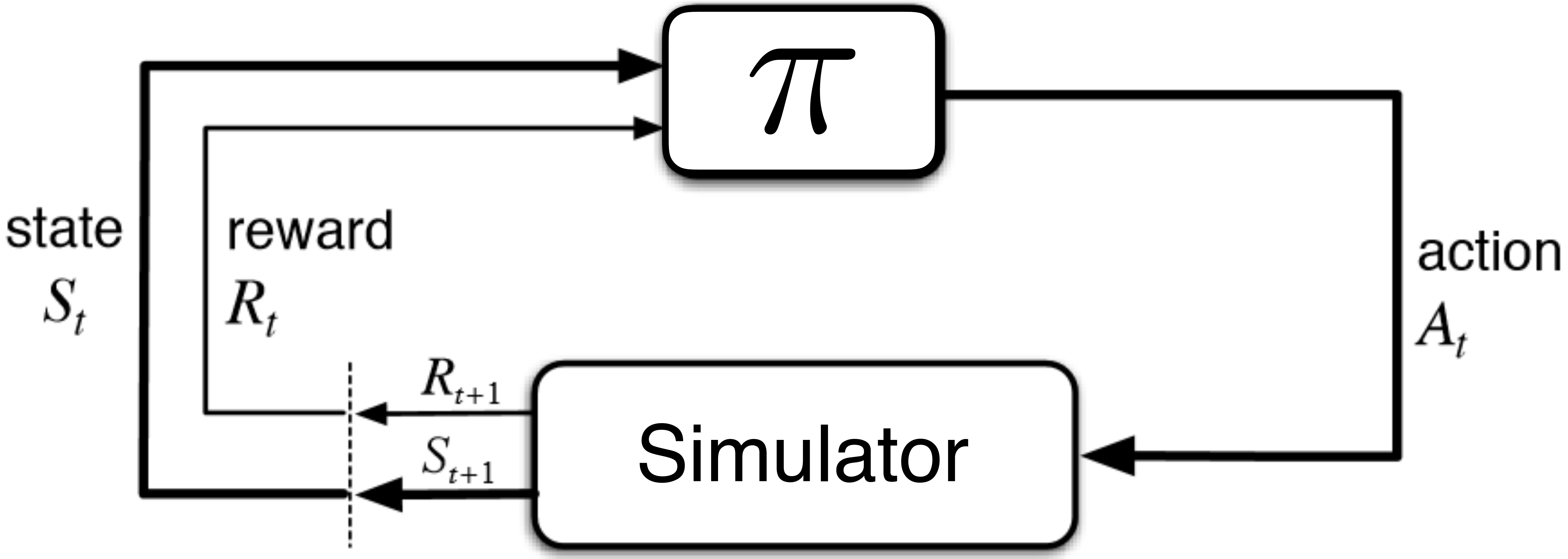
$$S_0, A_0, R_0, S_1, \ldots, S_L, A_L, R_L$$

Off-Environment RL



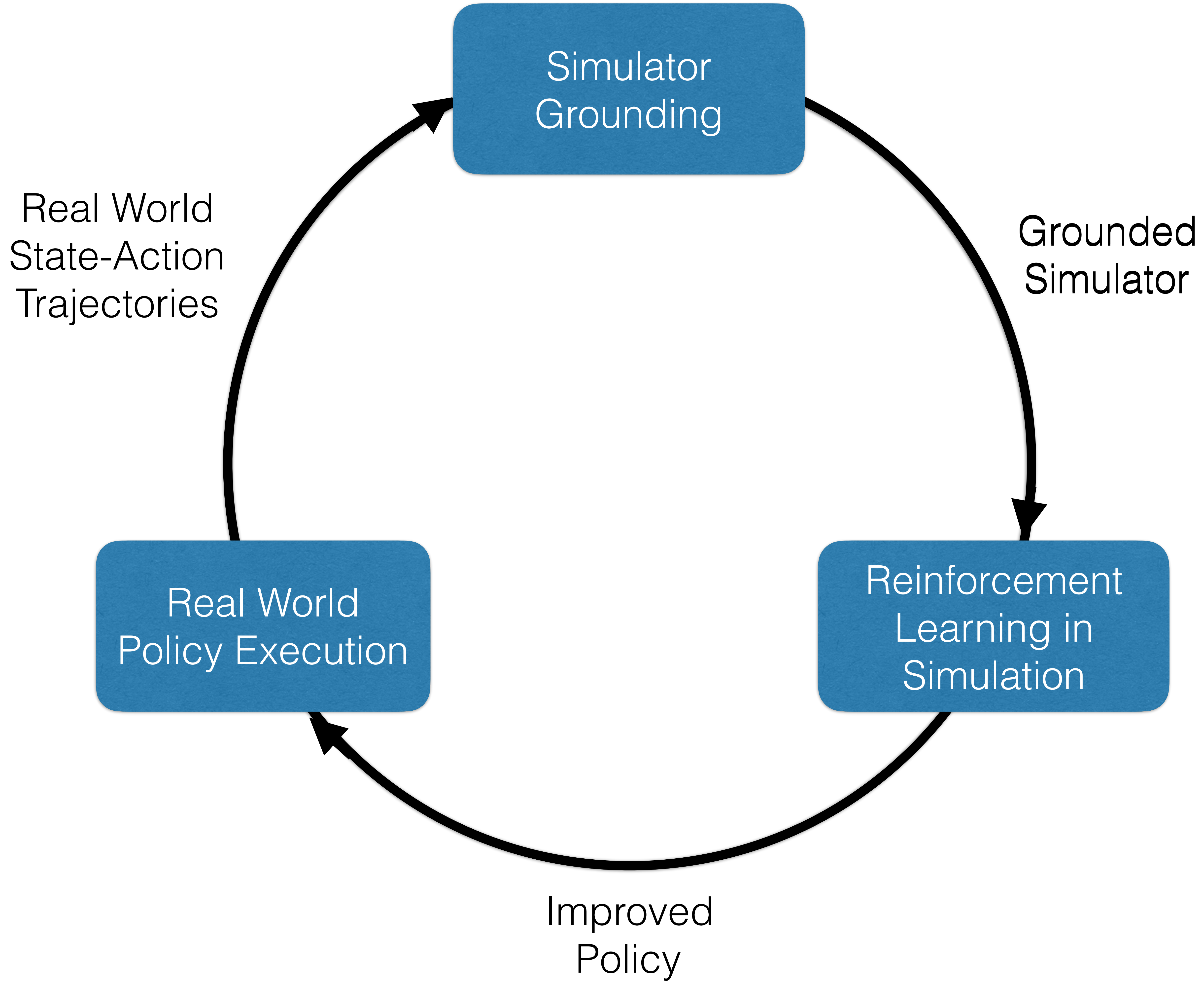
$$S_0, A_0, R_0, S_1, \ldots, S_L, A_L, R_L$$

Off-Environment RL



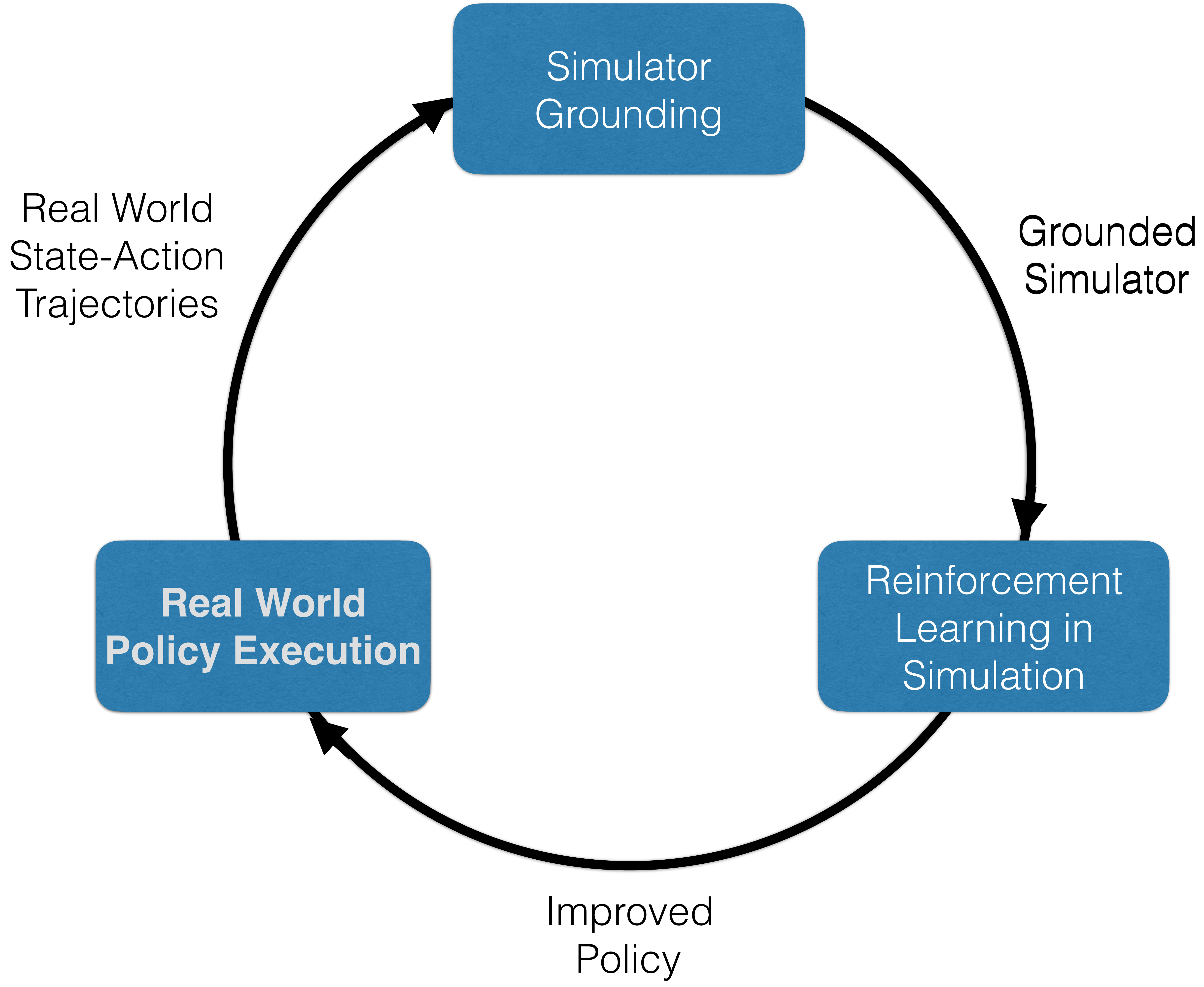
$$S_0, A_0, R_0, S_1, \ldots, S_L, A_L, R_L$$

Grounded Simulation Learning



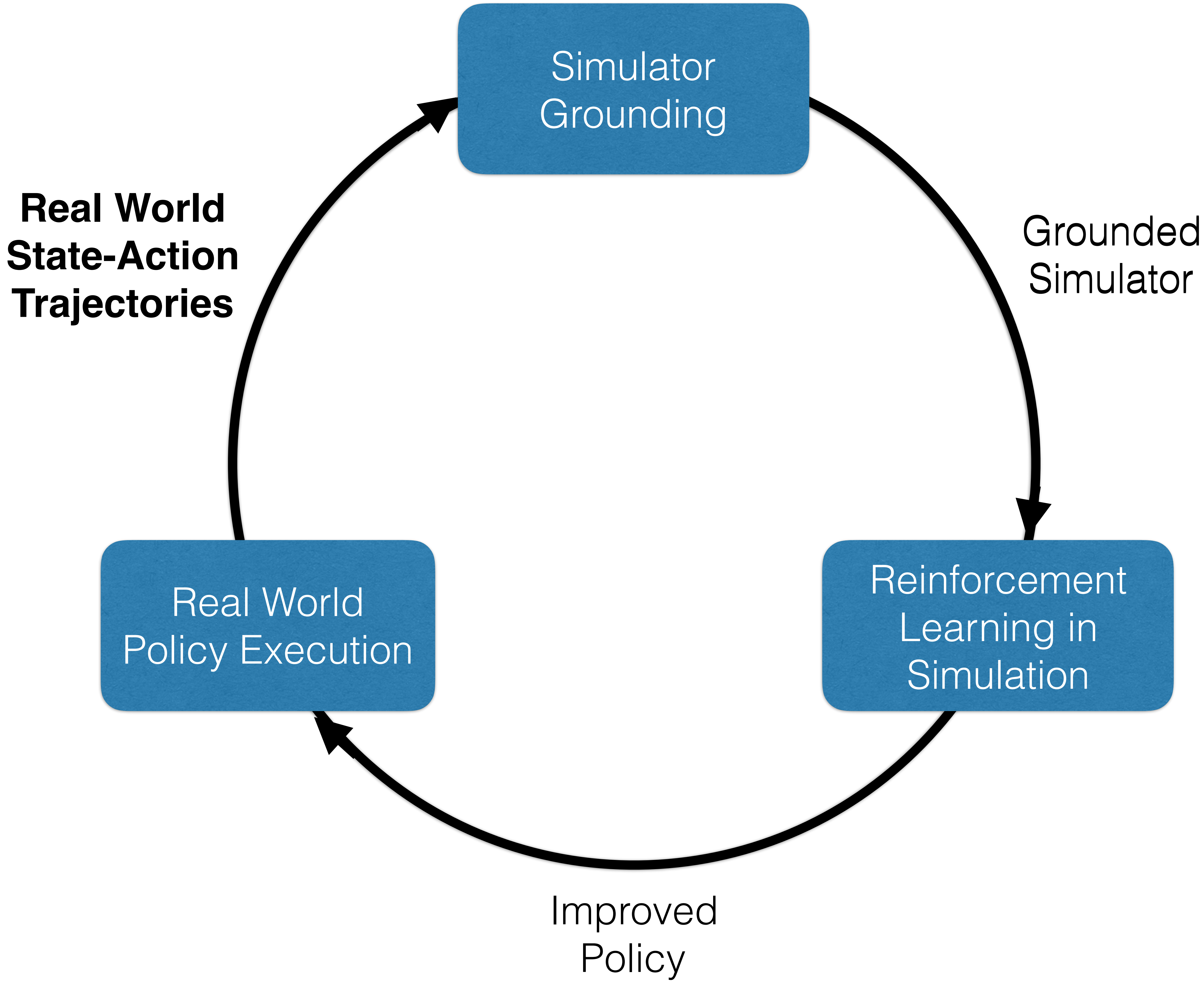
Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone (AAMAS 2013)

Grounded Simulation Learning



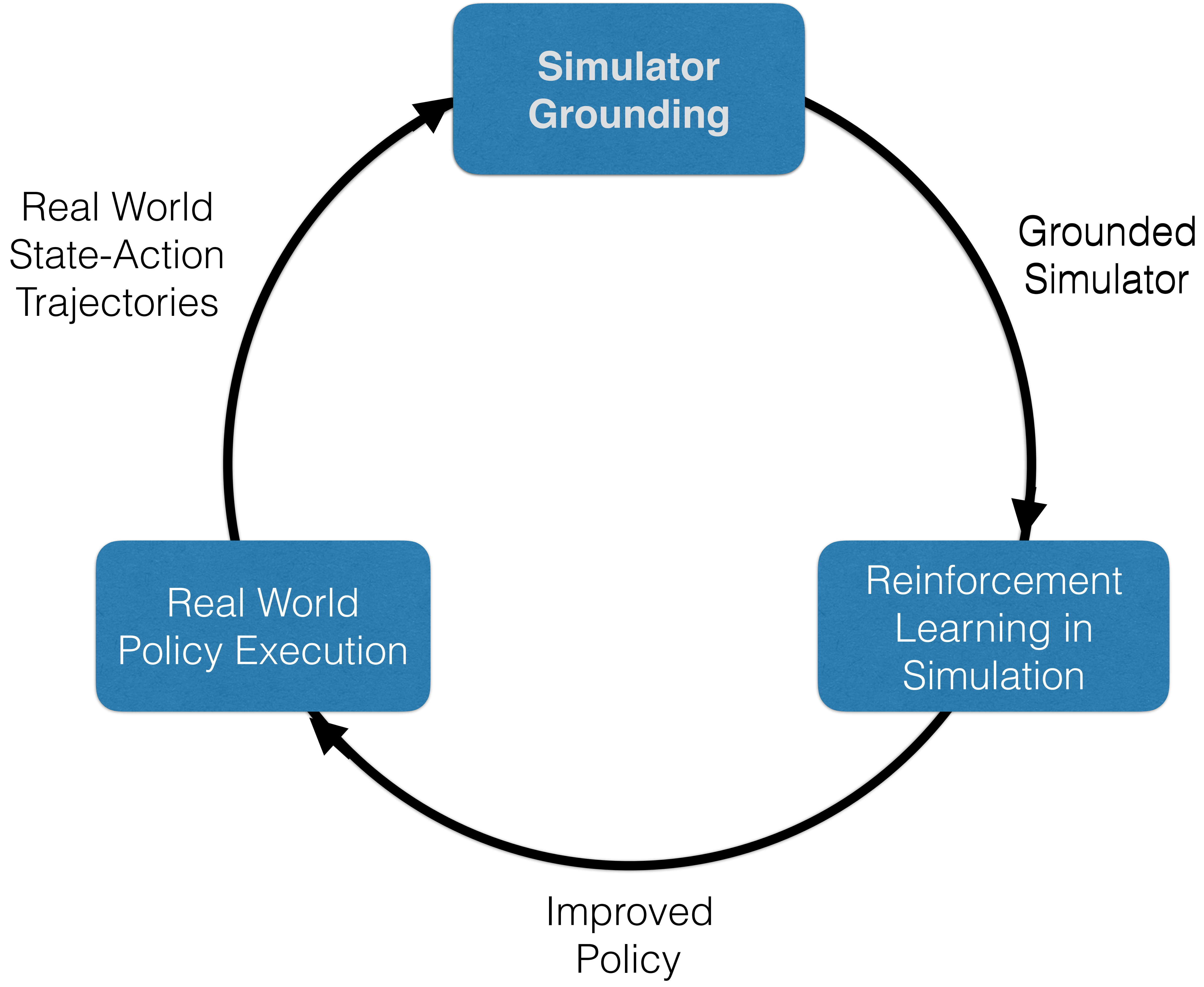
Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone (AAMAS 2013)

Grounded Simulation Learning



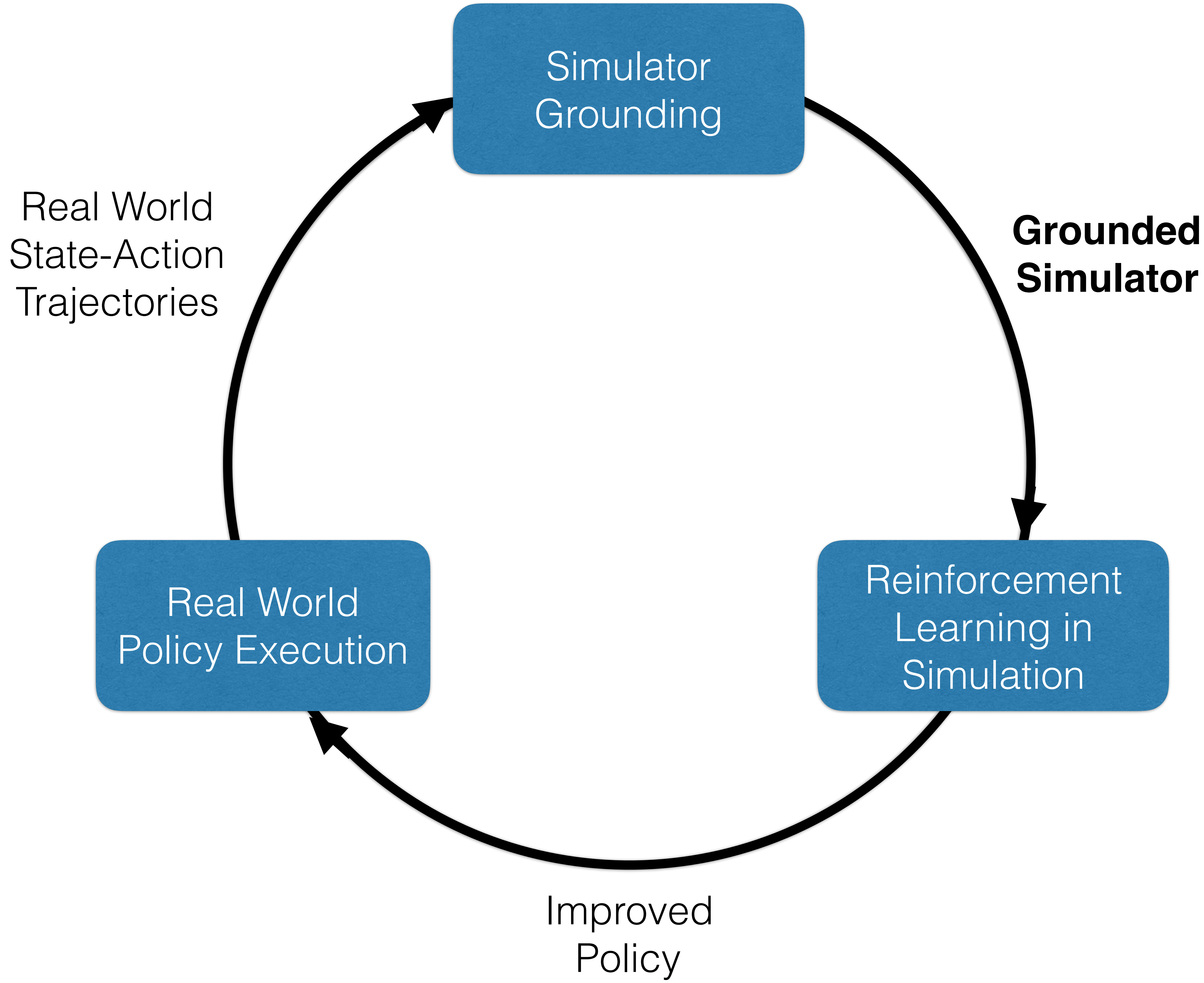
Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone (AAMAS 2013)

Grounded Simulation Learning



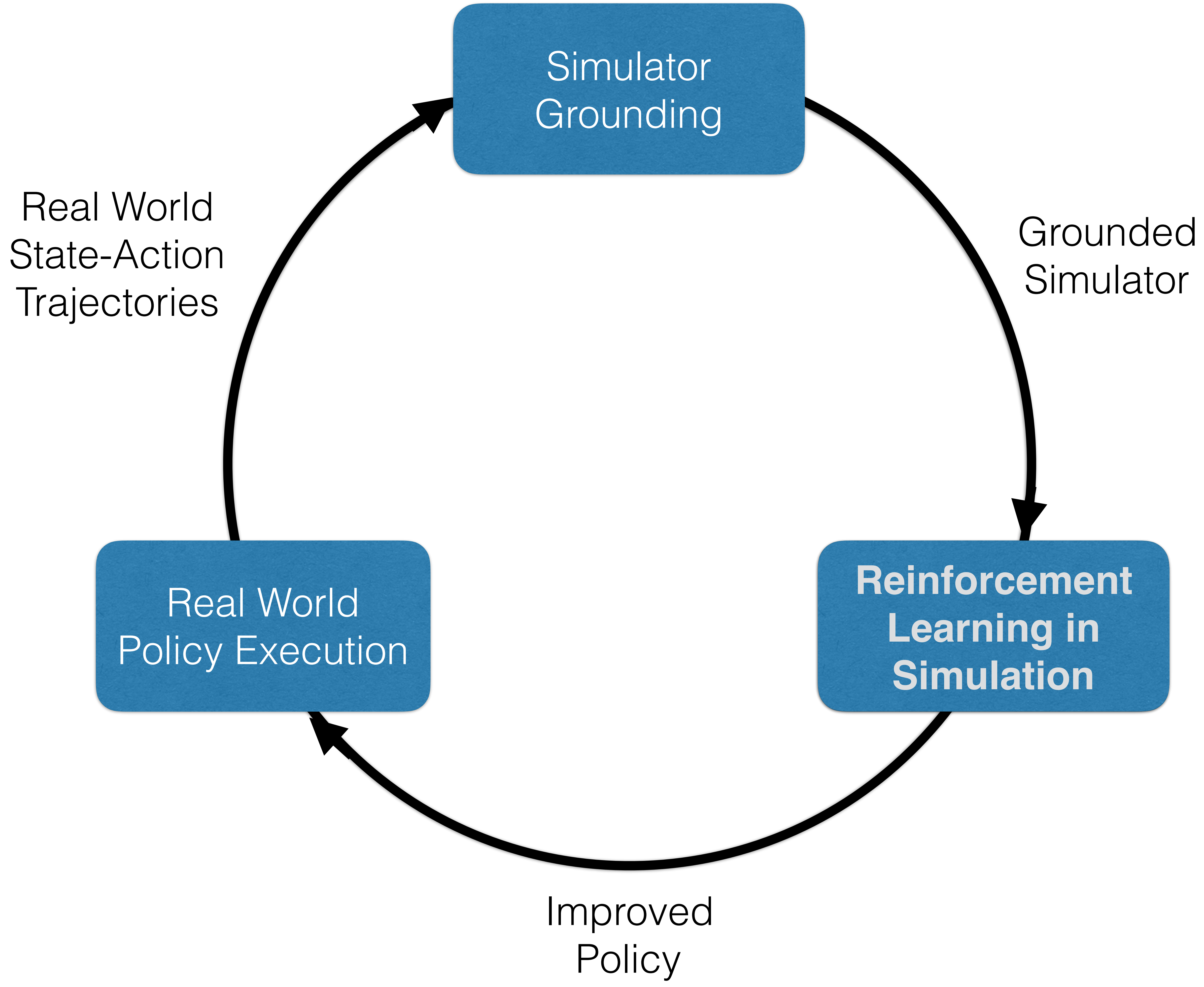
Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone (AAMAS 2013)

Grounded Simulation Learning



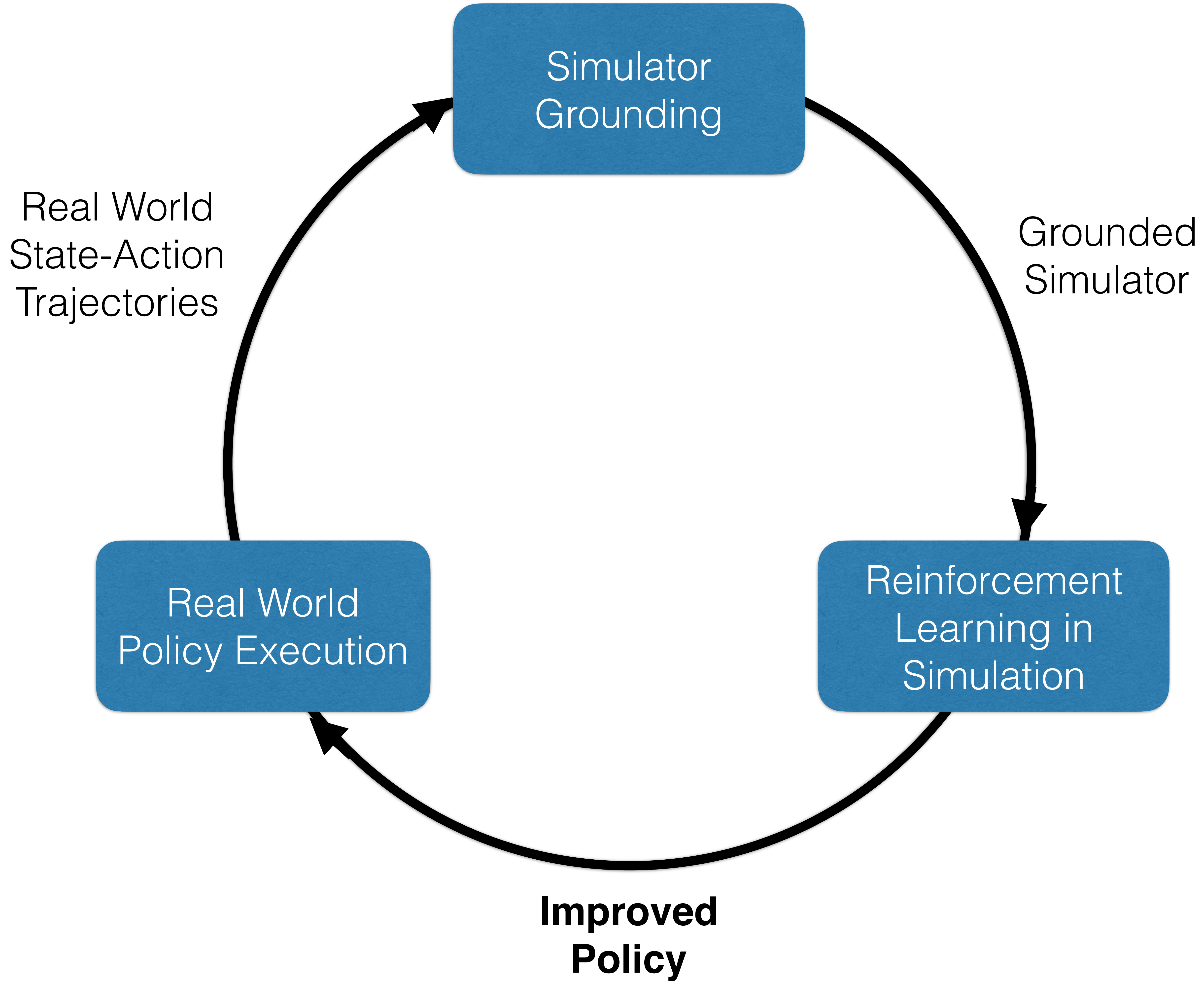
Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone (AAMAS 2013)

Grounded Simulation Learning



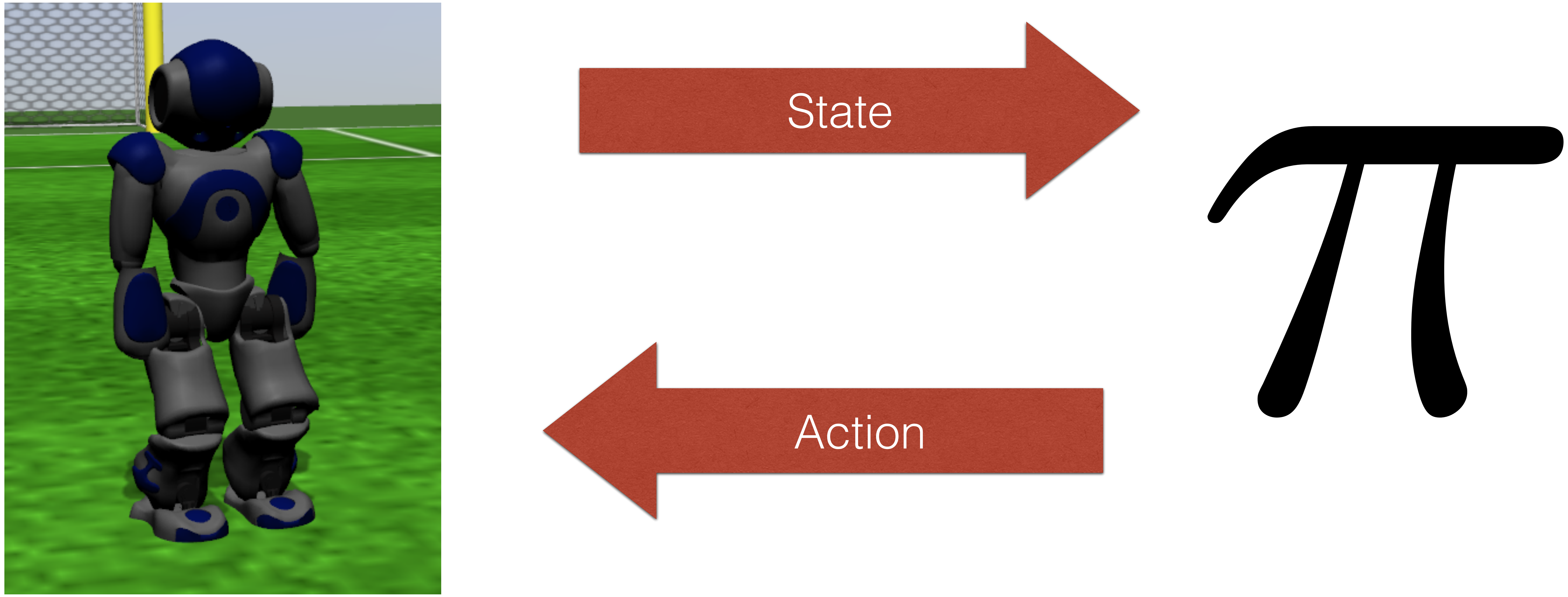
Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone (AAMAS 2013)

Grounded Simulation Learning



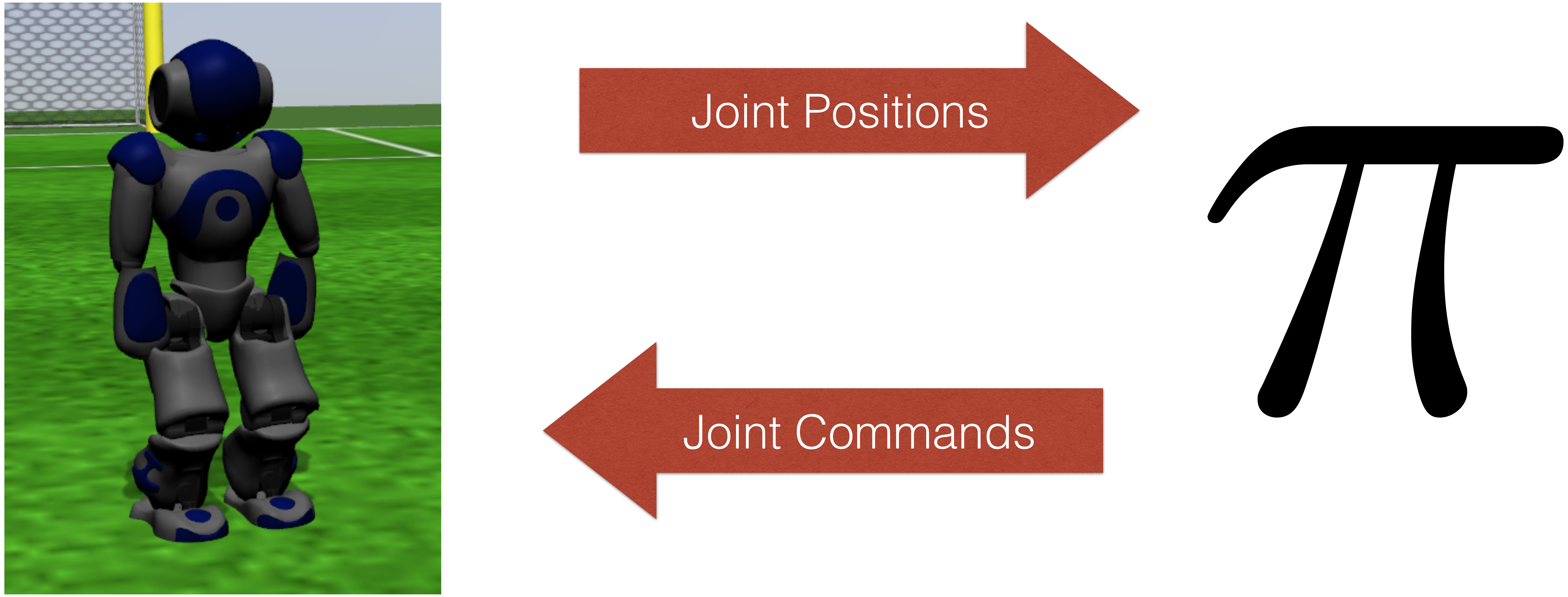
Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone (AAMAS 2013)

How do we make simulation more realistic?



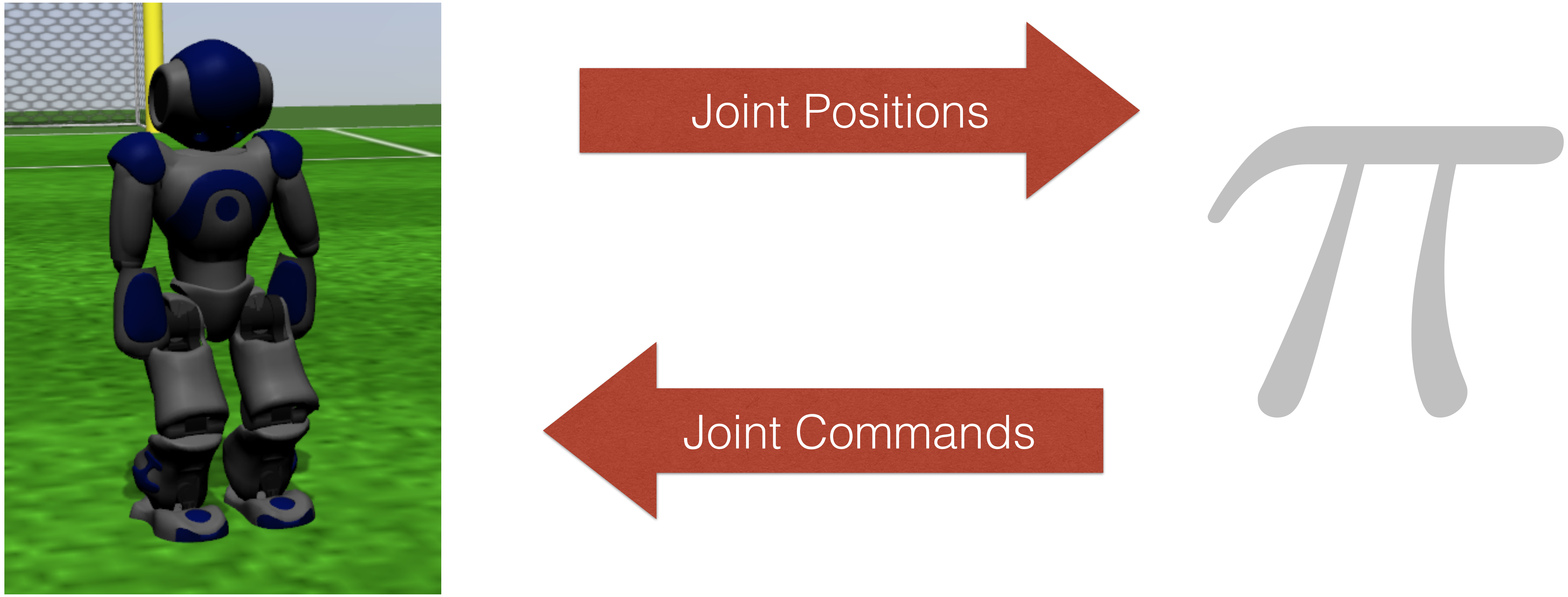
Josiah Hanna and Peter Stone (AAAI 2017)

How do we make simulation more realistic?



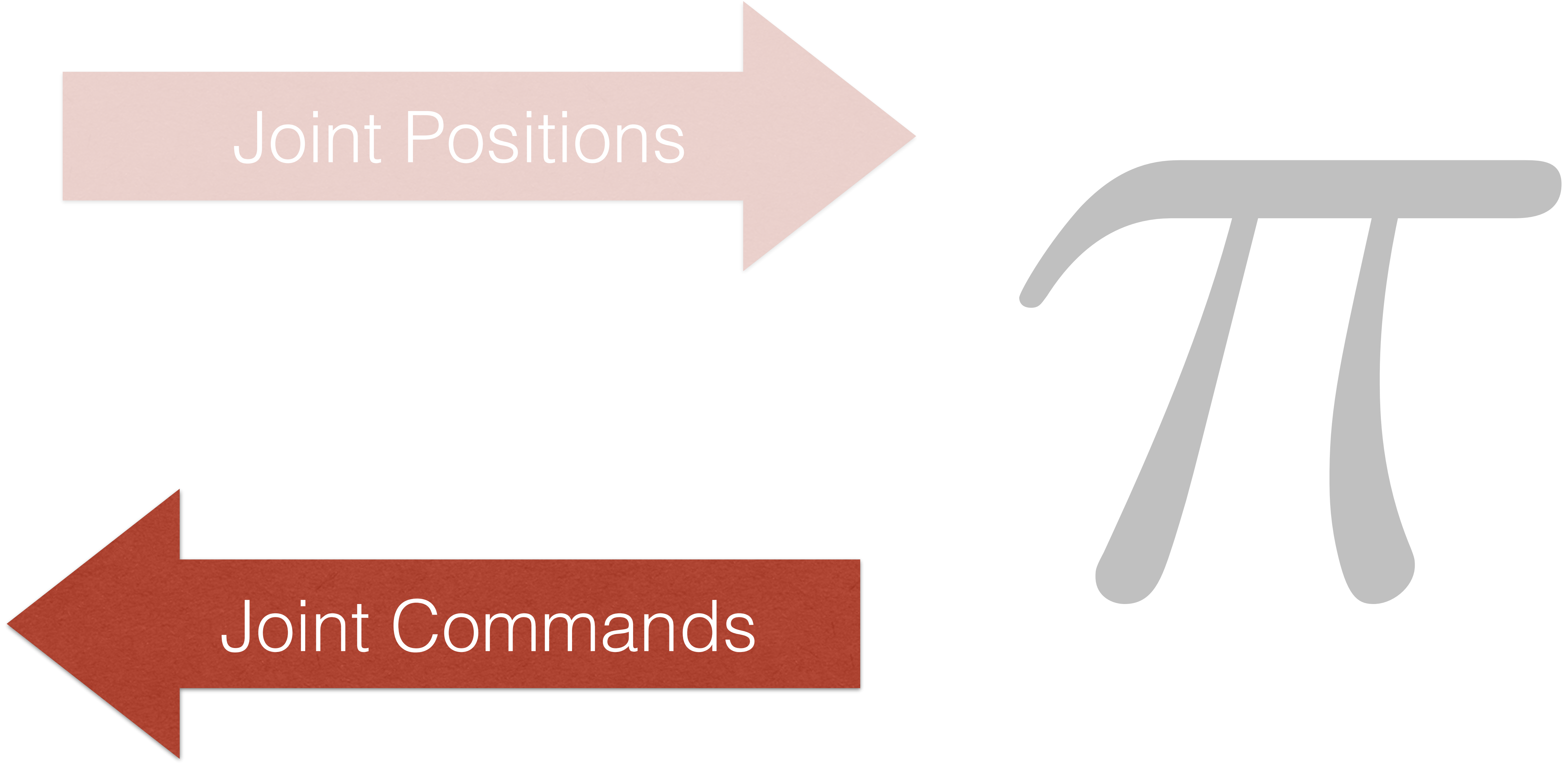
Josiah Hanna and Peter Stone (AAAI 2017)

How do we make simulation more realistic?



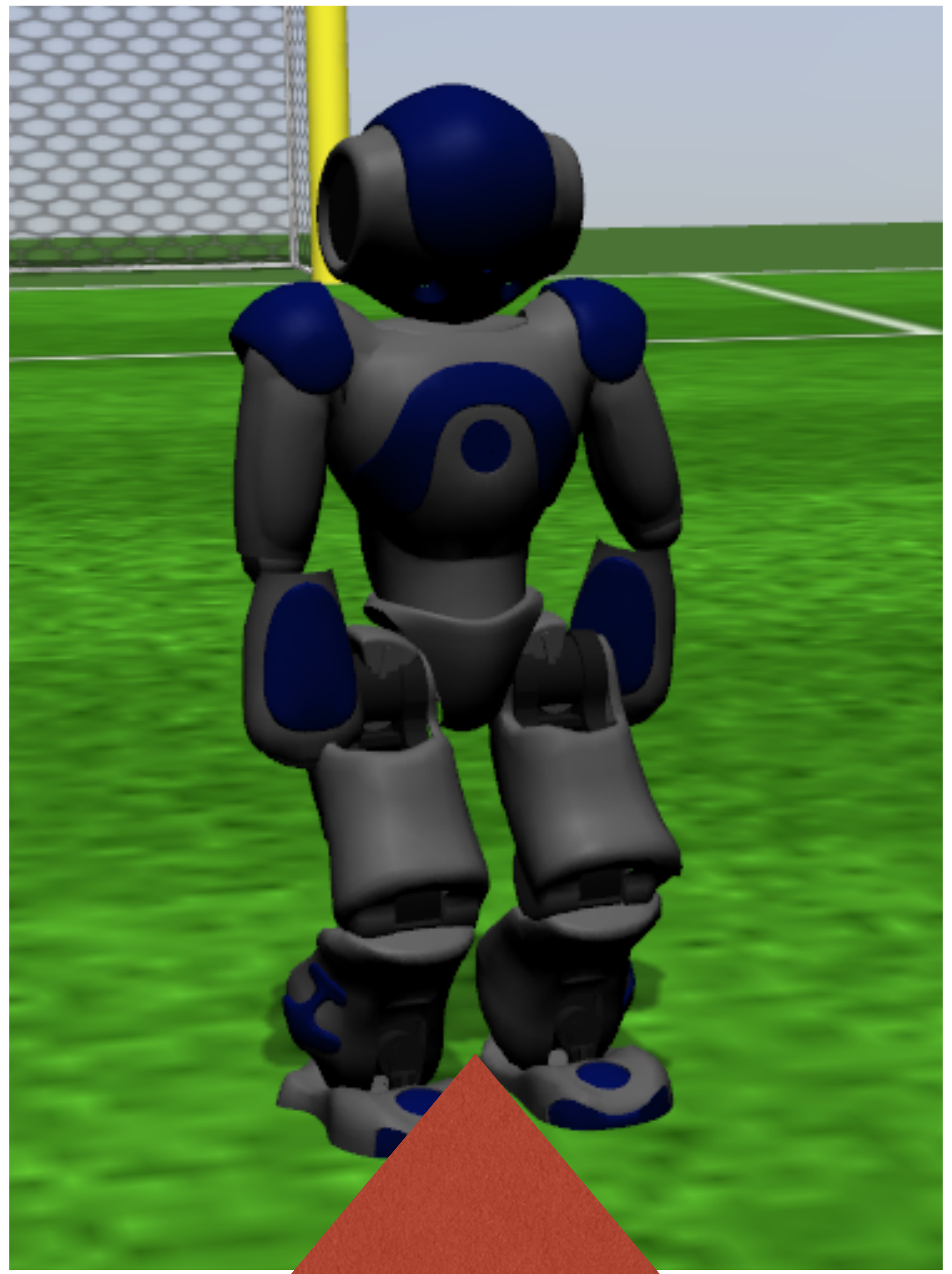
Josiah Hanna and Peter Stone (AAAI 2017)

How do we make simulation more realistic?



Josiah Hanna and Peter Stone (AAAI 2017)

How do we make simulation more realistic?



Joint Positions

π

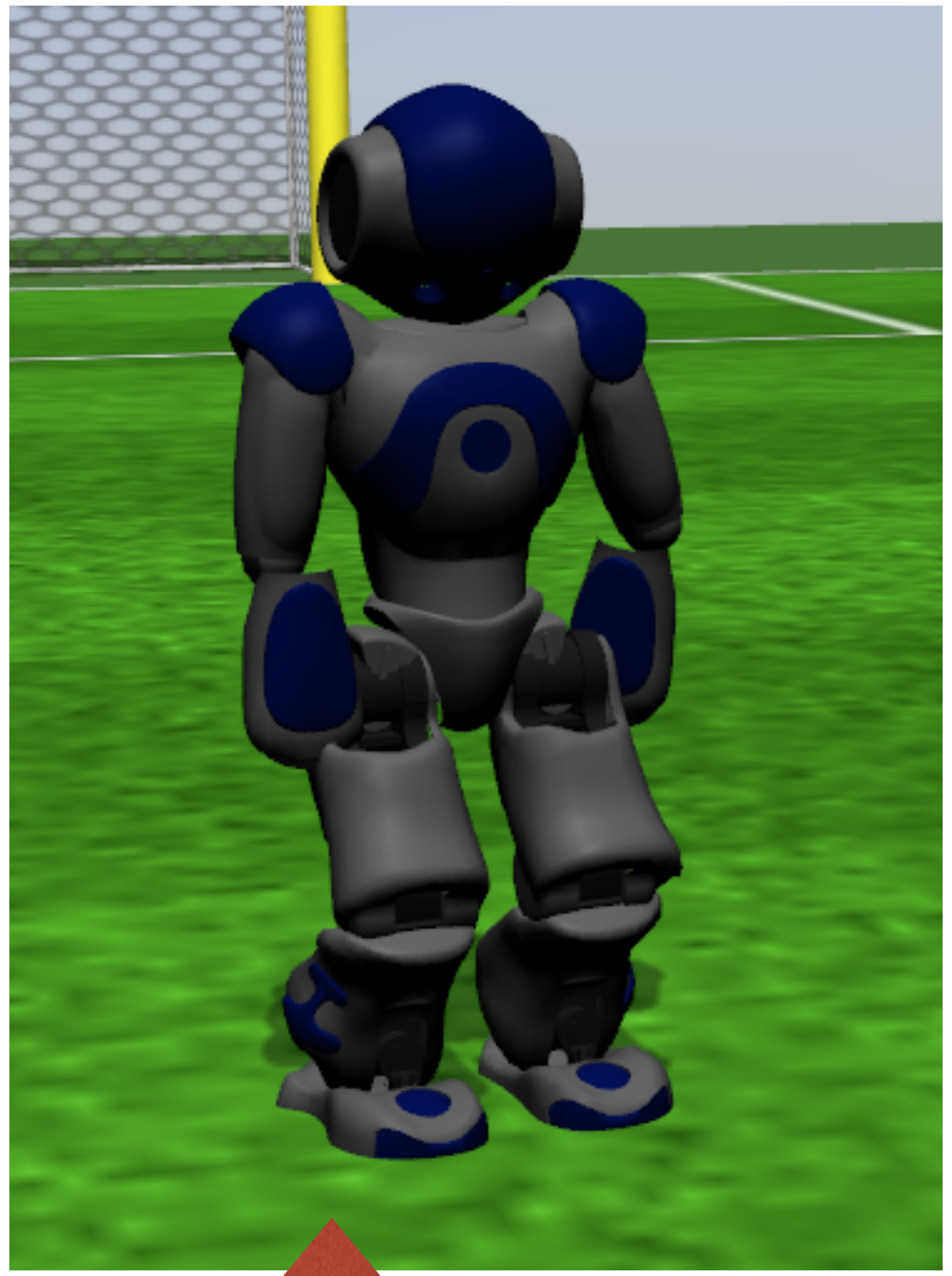
Joint Commands

Grounding Module

Modified Joint Commands

Josiah Hanna and Peter Stone (AAAI 2017)

How do we make simulation more realistic?



Joint Positions

π

Joint Commands

Predict real world effect.

Choose action that causes same effect in simulation.

Modified Joint Commands

Josiah Hanna and Peter Stone (AAAI 2017)

NAO Walking

Josiah Hanna and Peter Stone (AAAI 2017)

NAO Walking

Josiah Hanna and Peter Stone (AAAI 2017)

NAO Walking

Josiah Hanna and Peter Stone (AAAI 2017)

NAO Walking

Josiah Hanna and Peter Stone (AAAI 2017)

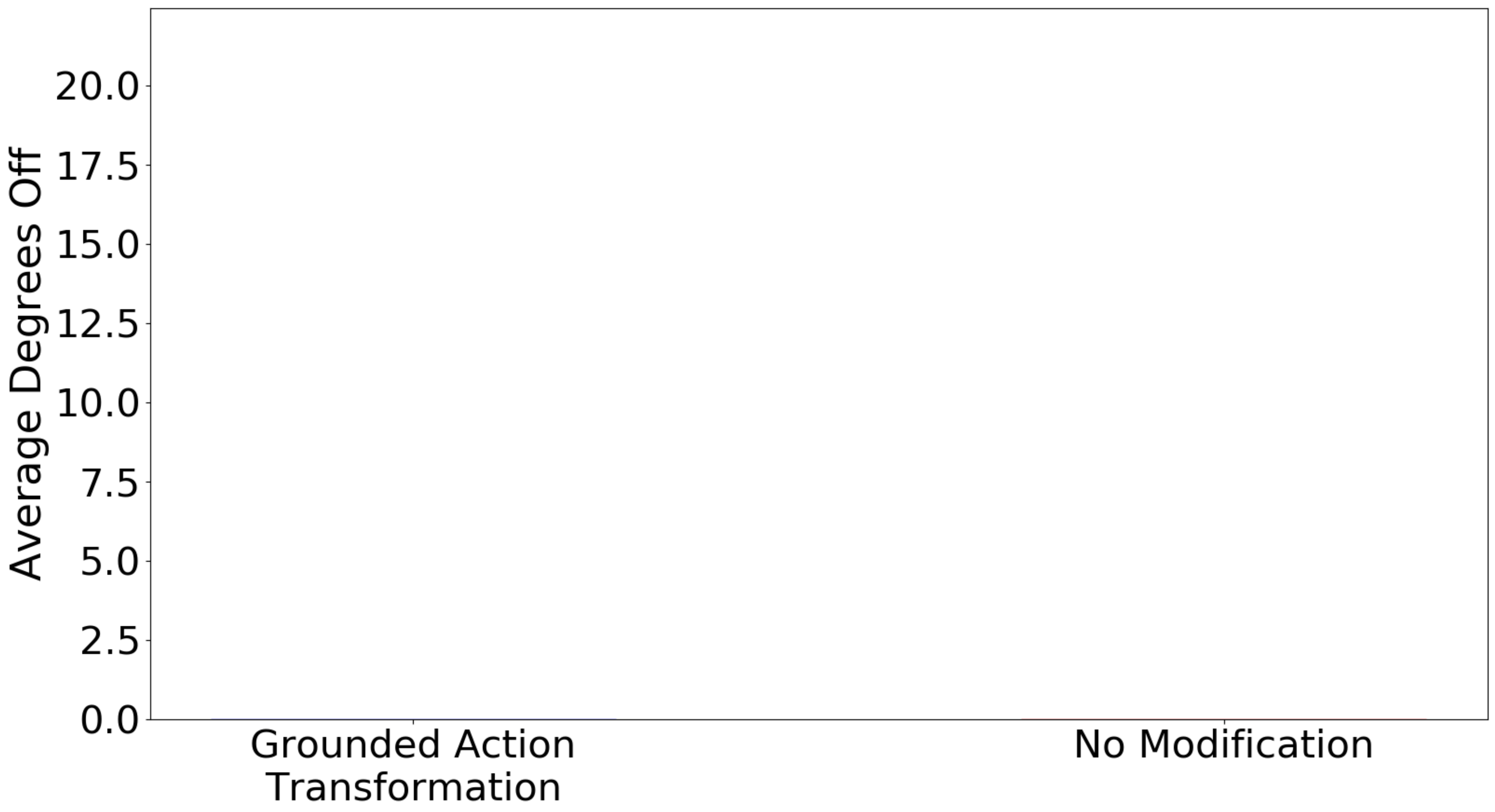
NAO Walking

Josiah Hanna and Peter Stone (AAAI 2017)

Sim-to-sim transfer: Learning Arm Control

NAO robot learning to move arm joints to target position.

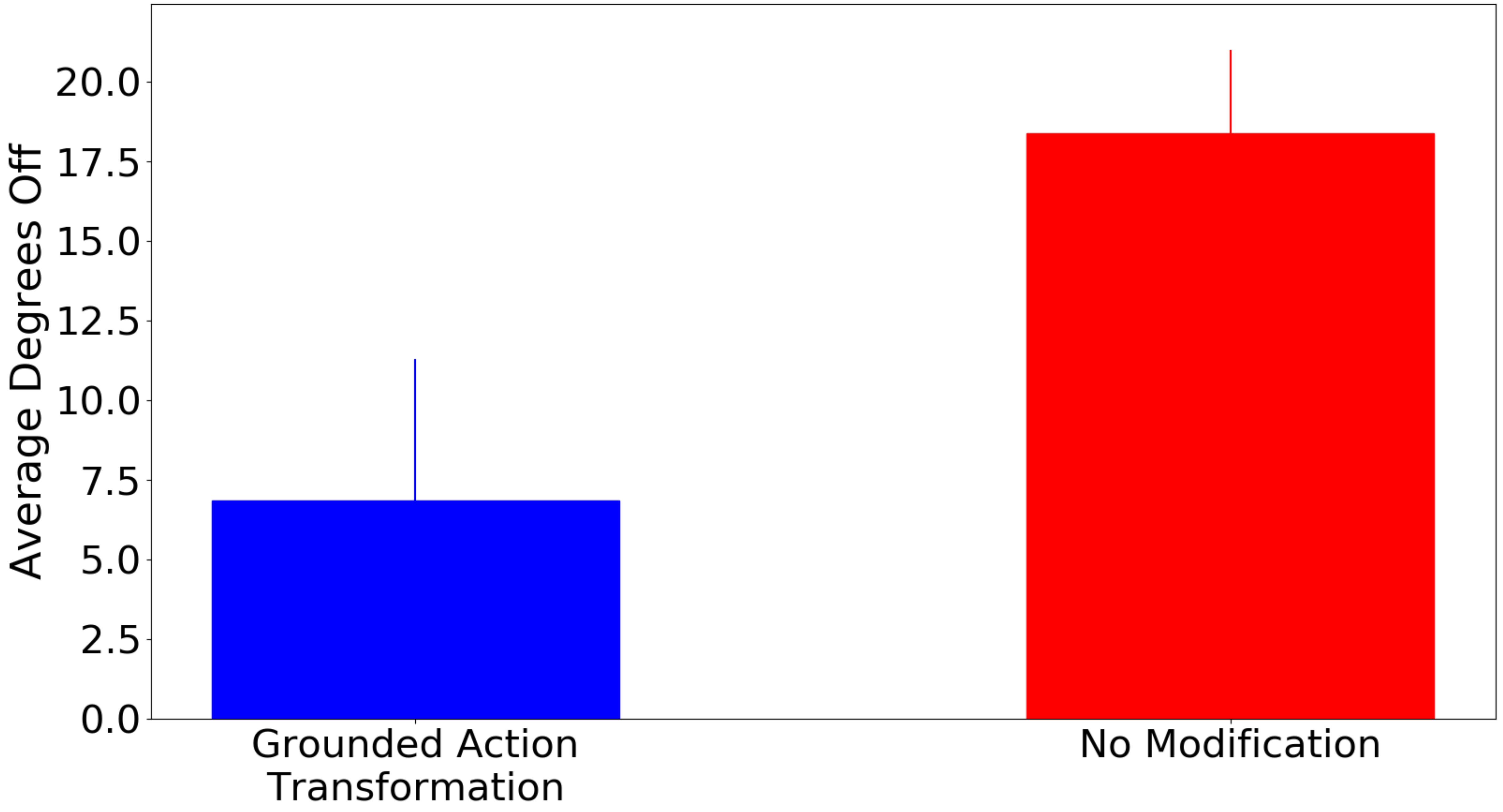
Transfer from Simspark simulator to Gazebo simulator.



Sim-to-sim transfer: Learning Arm Control

NAO robot learning to move arm joints to target position.

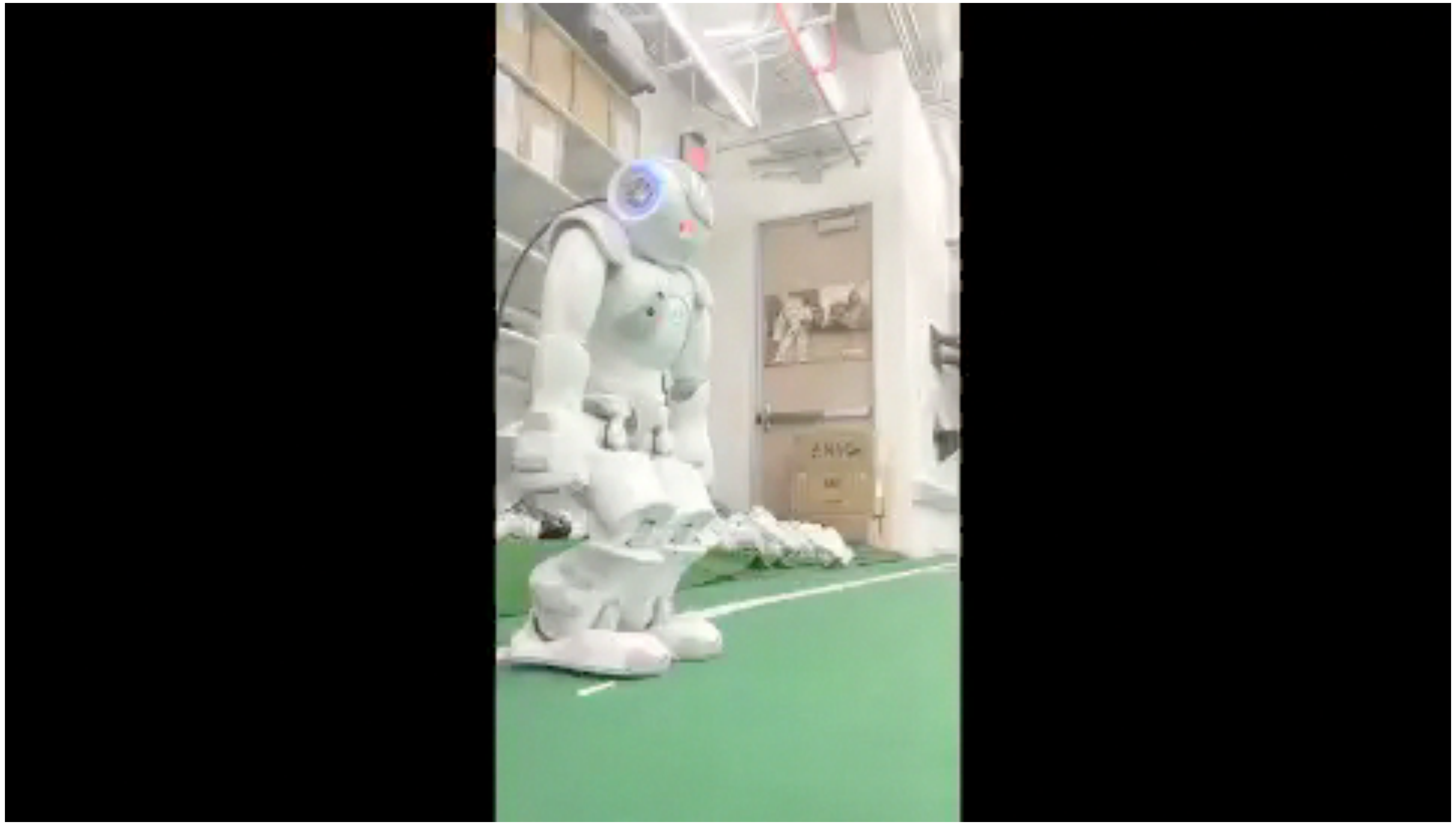
Transfer from Simspark simulator to Gazebo simulator.



Learning to walk with less prior knowledge

Learning to walk with less prior knowledge

Learning to walk with less prior knowledge



Josiah Hanna

Learning with Simulated Data

Learning with Simulated Data

Contribution 5: Grounded action transformation algorithm allowing an RL agent to learn from simulated data.

Learning with Simulated Data

Contr
allowi

Take-away Message

Modifying the policy's actions can correct discrepancy between simulation and reality.

Learning with Simulated Data

Contr
allowi

Take-away Message

Modifying the policy's actions can correct discrepancy between simulation and reality.

Additional results in dissertation: Bound on error in model-based policy value estimation, additional empirical results.

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

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

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How should an RL agent weight off-policy data?

How can an RL agent use simulated data?

How can an RL agent combine simulated and off-policy data?

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

Combining Simulated and Off-Policy Data

Combining Simulated and Off-Policy Data

Contribution 6: Model-based bootstrap algorithm for approximate high confidence off-policy value estimation.

Combining Simulated and Off-Policy Data

Contribution 6: Model-based bootstrap algorithm for approximate high confidence off-policy value estimation.

Contribution 7: Weighted doubly robust bootstrap algorithm for approximate high confidence off-policy value estimation.

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

How should an RL agent collect off-policy data?

How can an RL agent use simulated data?

How should an RL agent weight off-policy data?

How can an RL agent combine simulated and off-policy data?

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

Future Directions

Future Directions

1. Hierarchical sim-to-real.

Future Directions

1. Hierarchical sim-to-real.

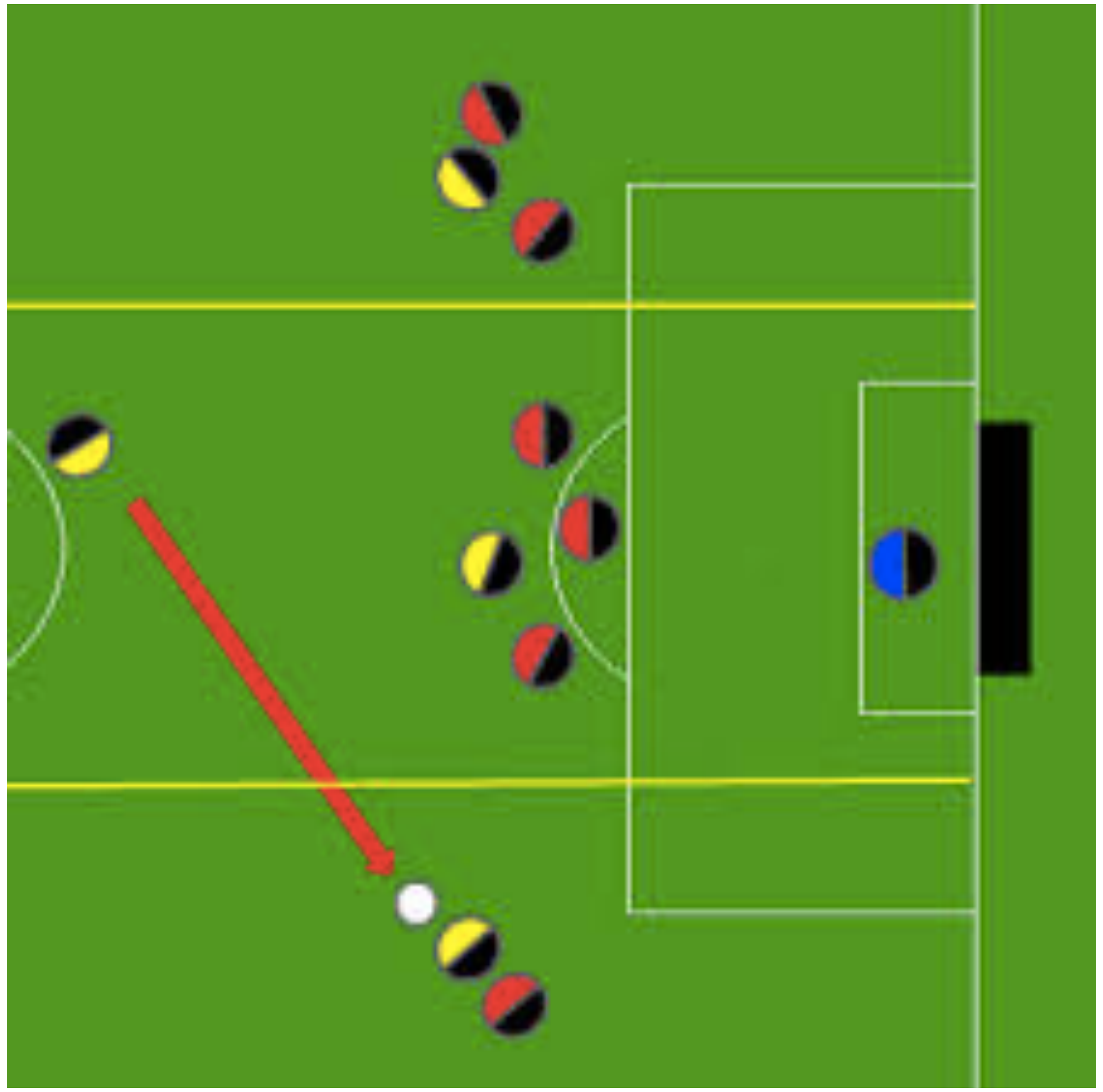
2. Optimal sampling for regression importance sampling.

Future Directions

1. Hierarchical sim-to-real.
2. Optimal sampling for regression importance sampling.
3. From policy value estimation to policy evaluation.

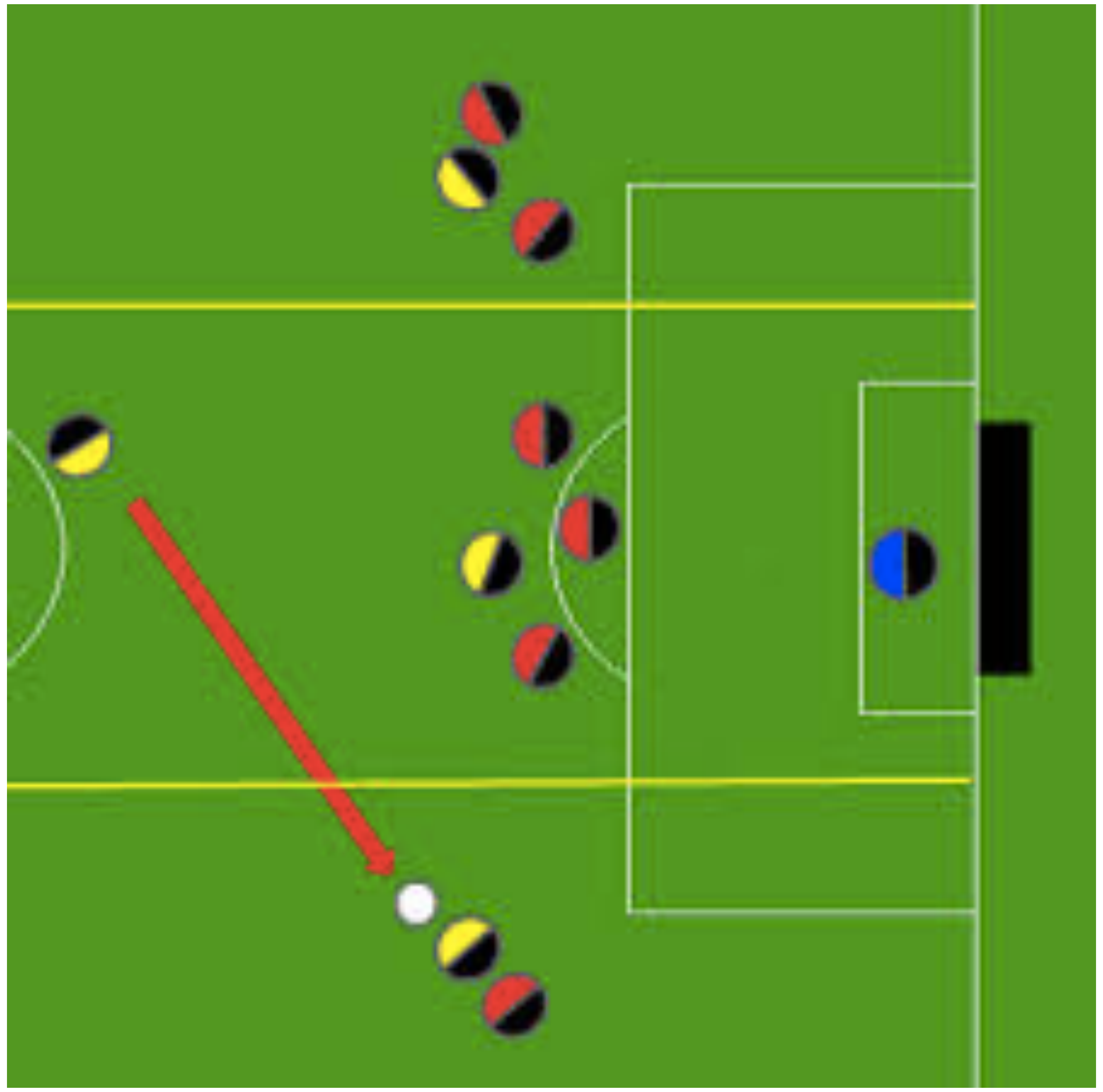
Learning in an abstract simulation

Learning in an abstract simulation

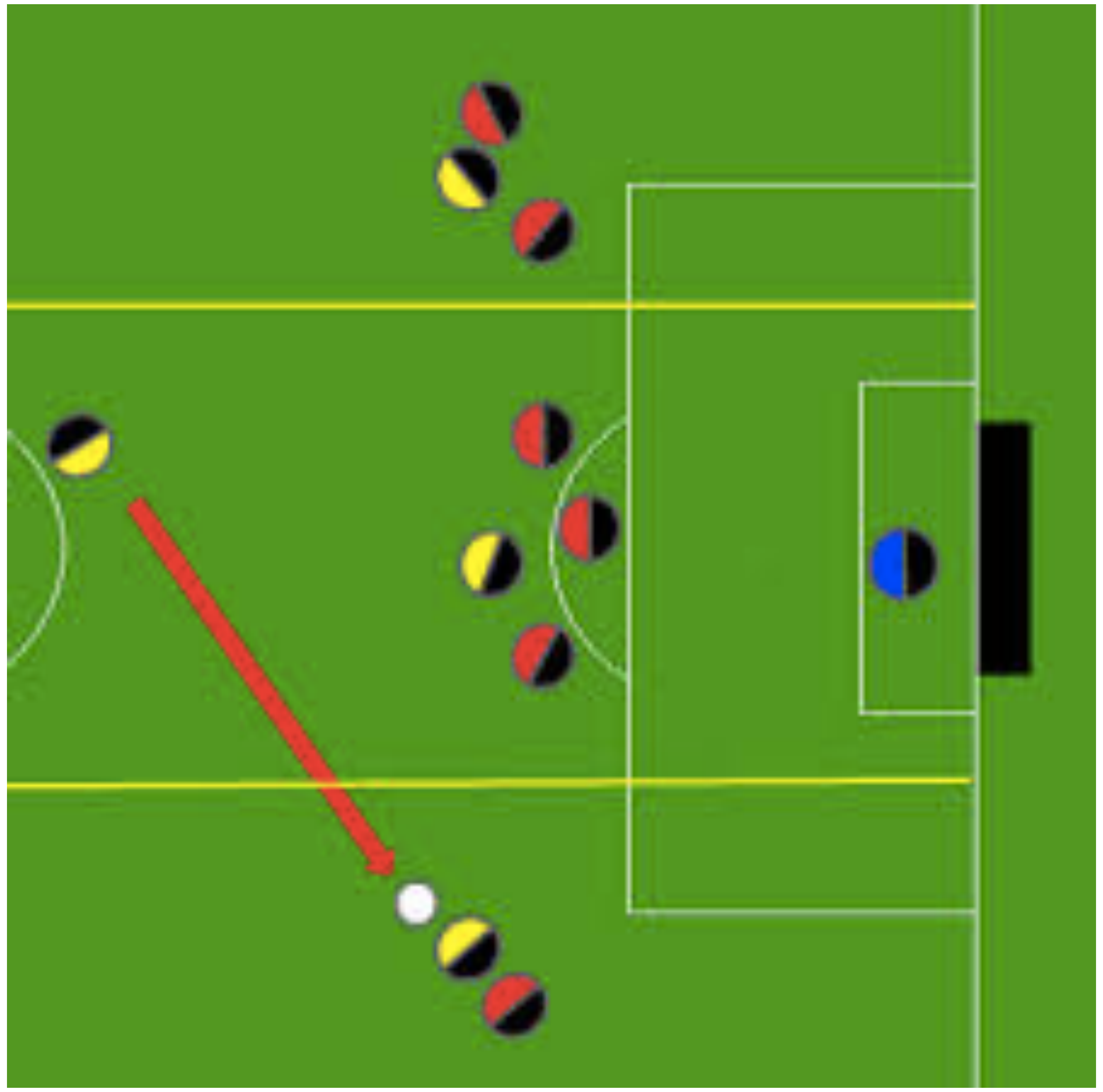


Josiah Hanna

Learning in an abstract simulation



Learning in an abstract simulation



Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

Optimal sampling for regression importance sampling

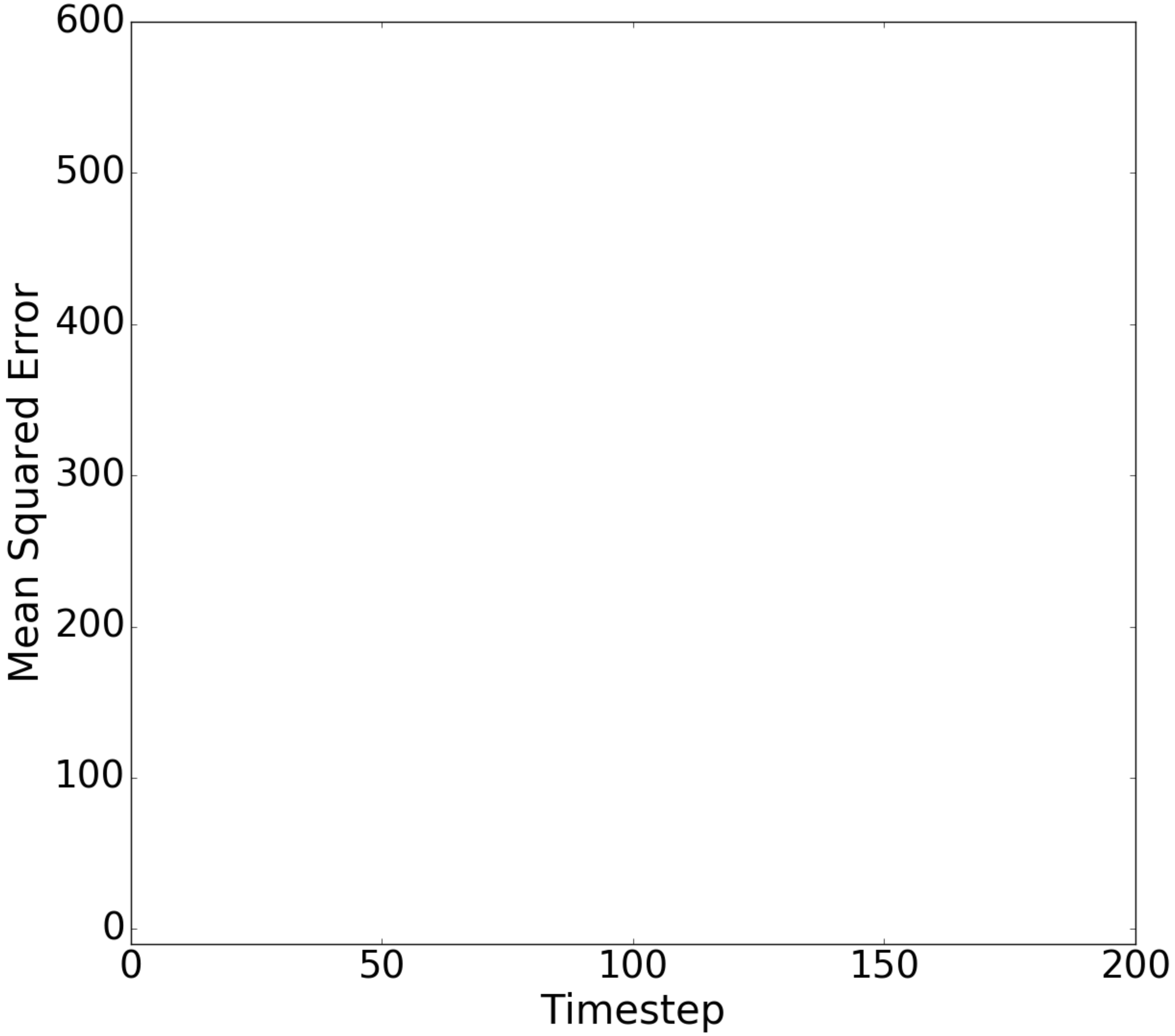
$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

50-armed bandit with stochastic rewards.

Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

50-armed bandit with stochastic rewards.

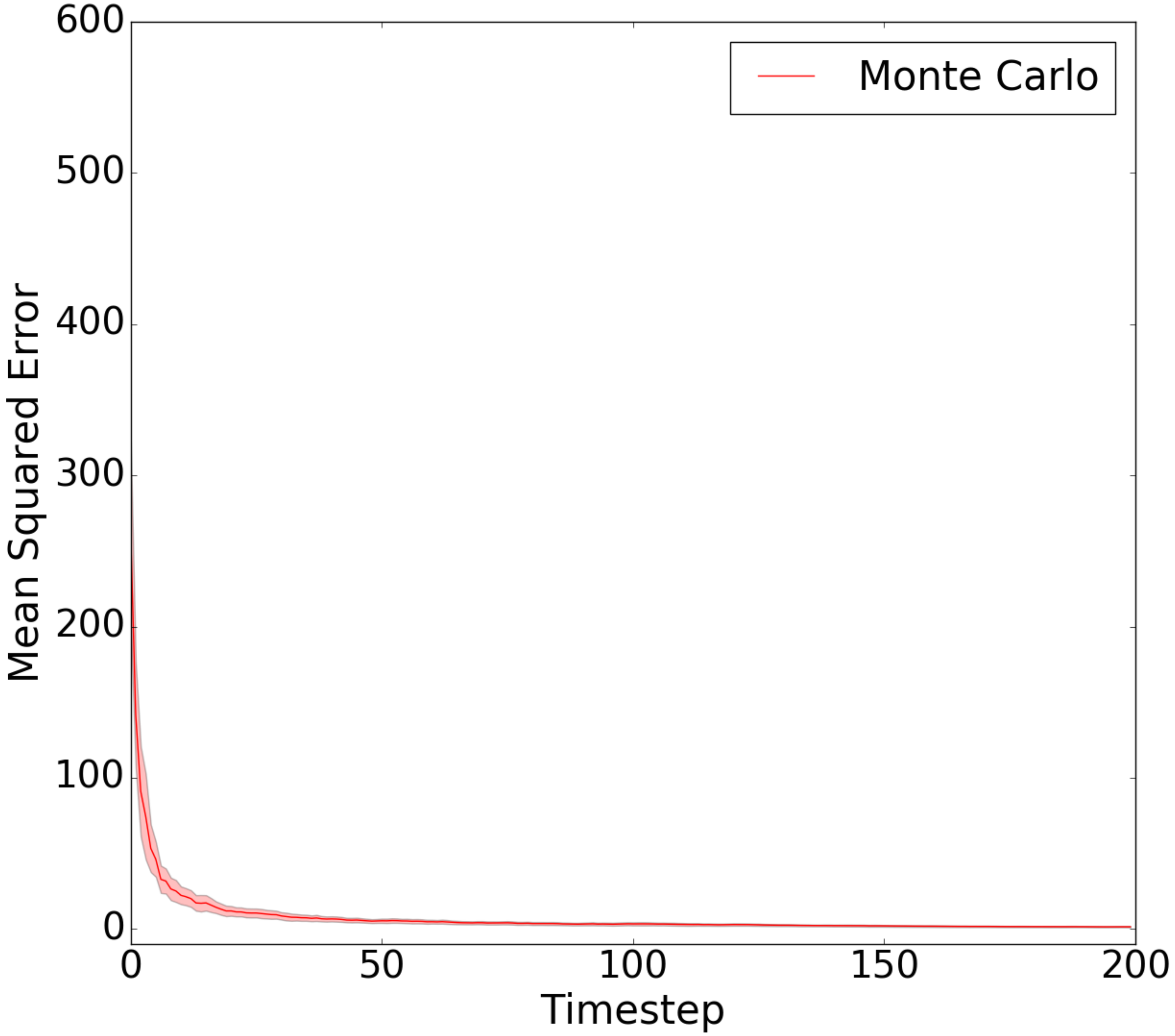


Josiah Hanna

Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L}\frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right)\times\left(\sum_{t=0}^{L}R_t\right)$$

50-armed bandit with stochastic rewards.

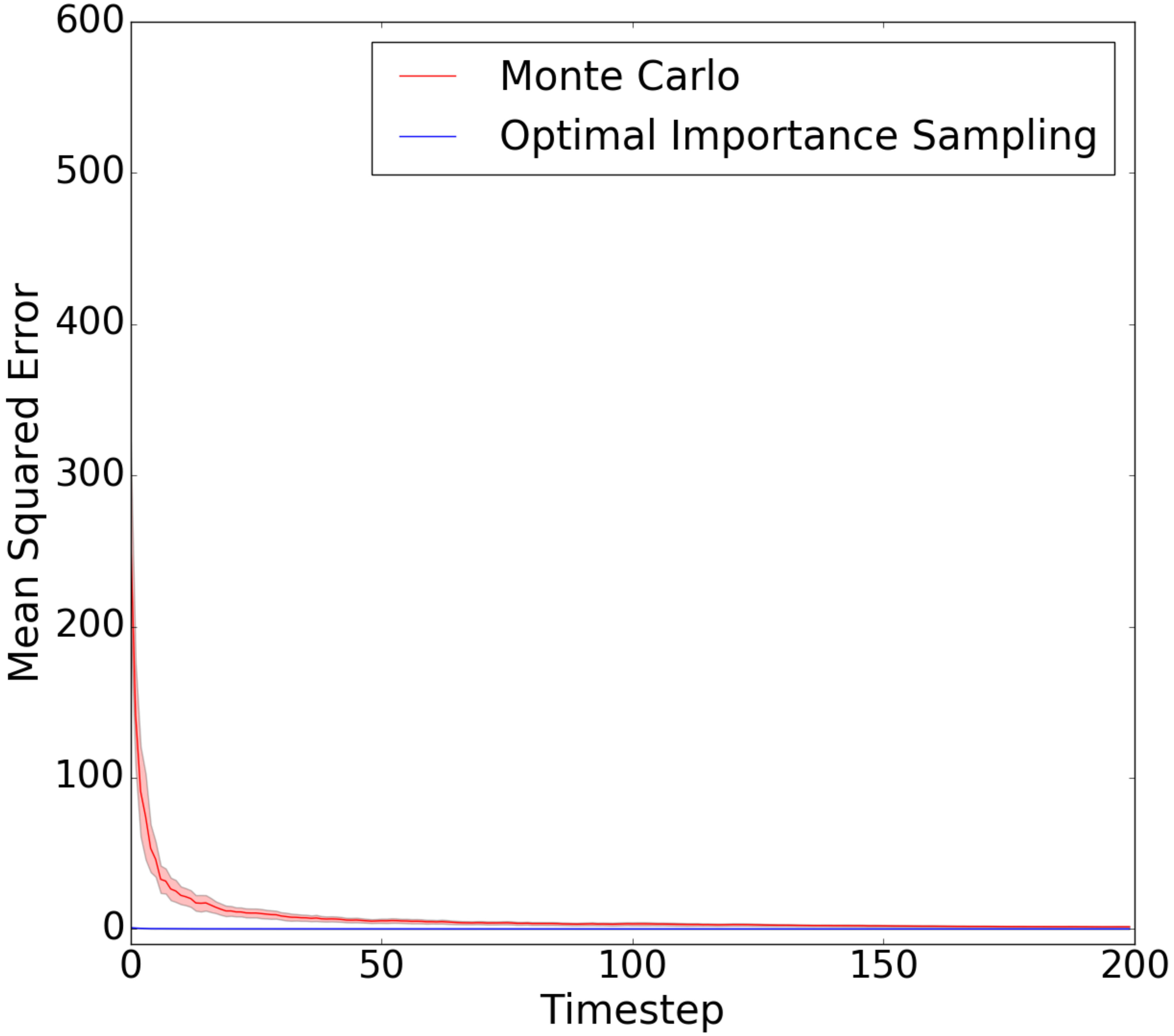


Josiah Hanna

Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

50-armed bandit with stochastic rewards.

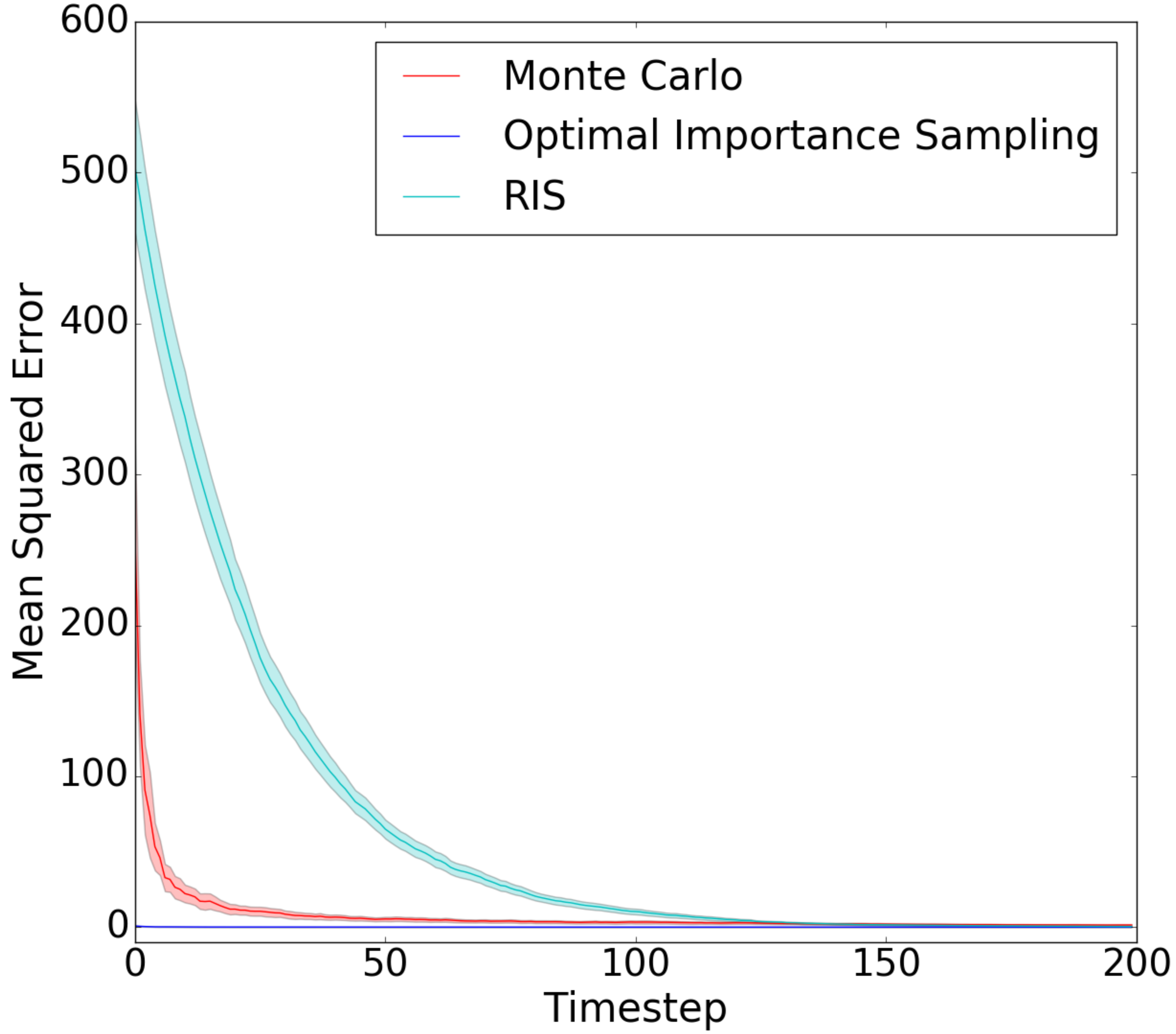


Josiah Hanna

Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

50-armed bandit with stochastic rewards.



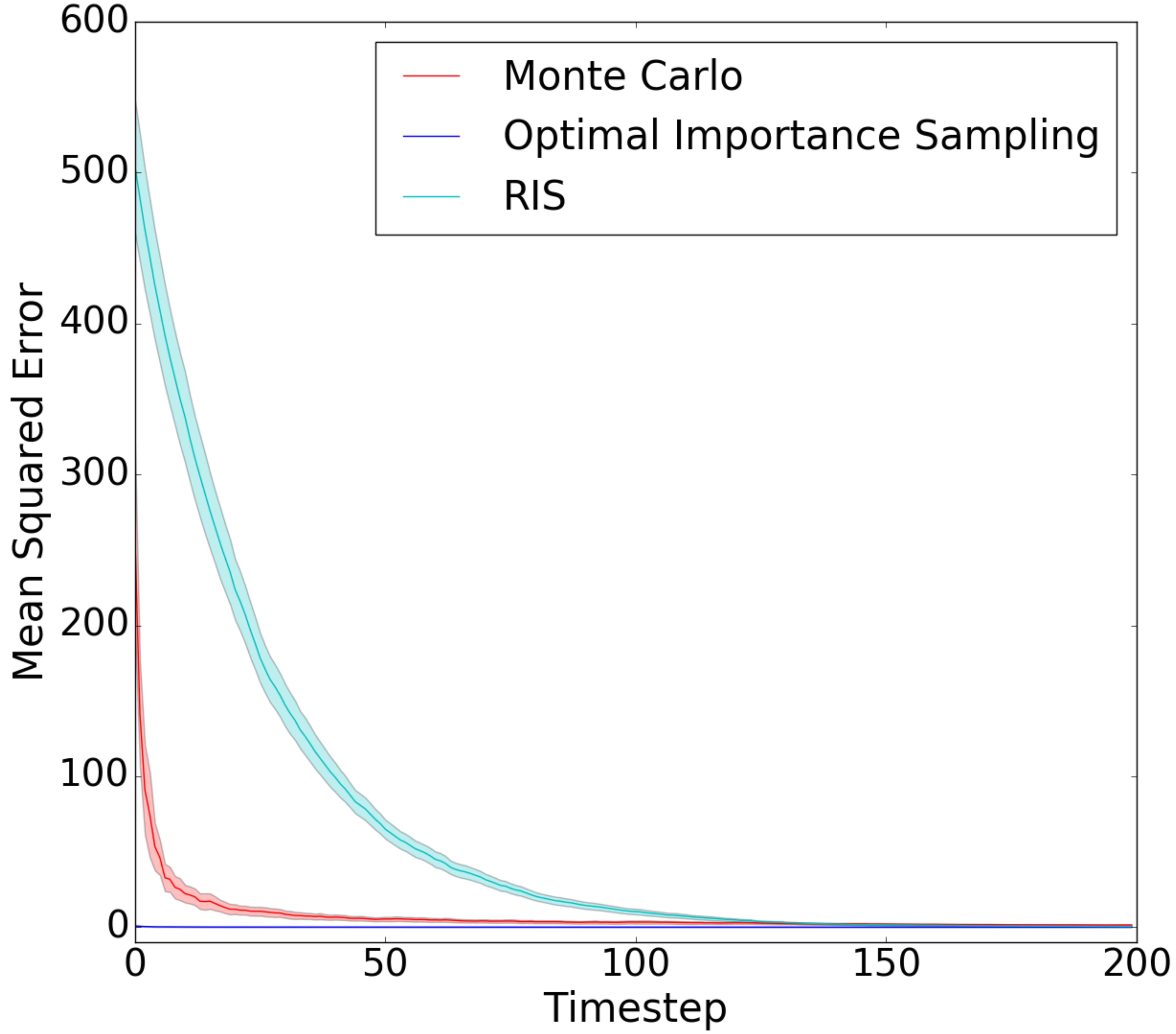
Josiah Hanna

Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L}\frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right)\times\left(\sum_{t=0}^{L}R_t\right)$$

50-armed bandit with stochastic rewards.

RIS needs to observe every arm!



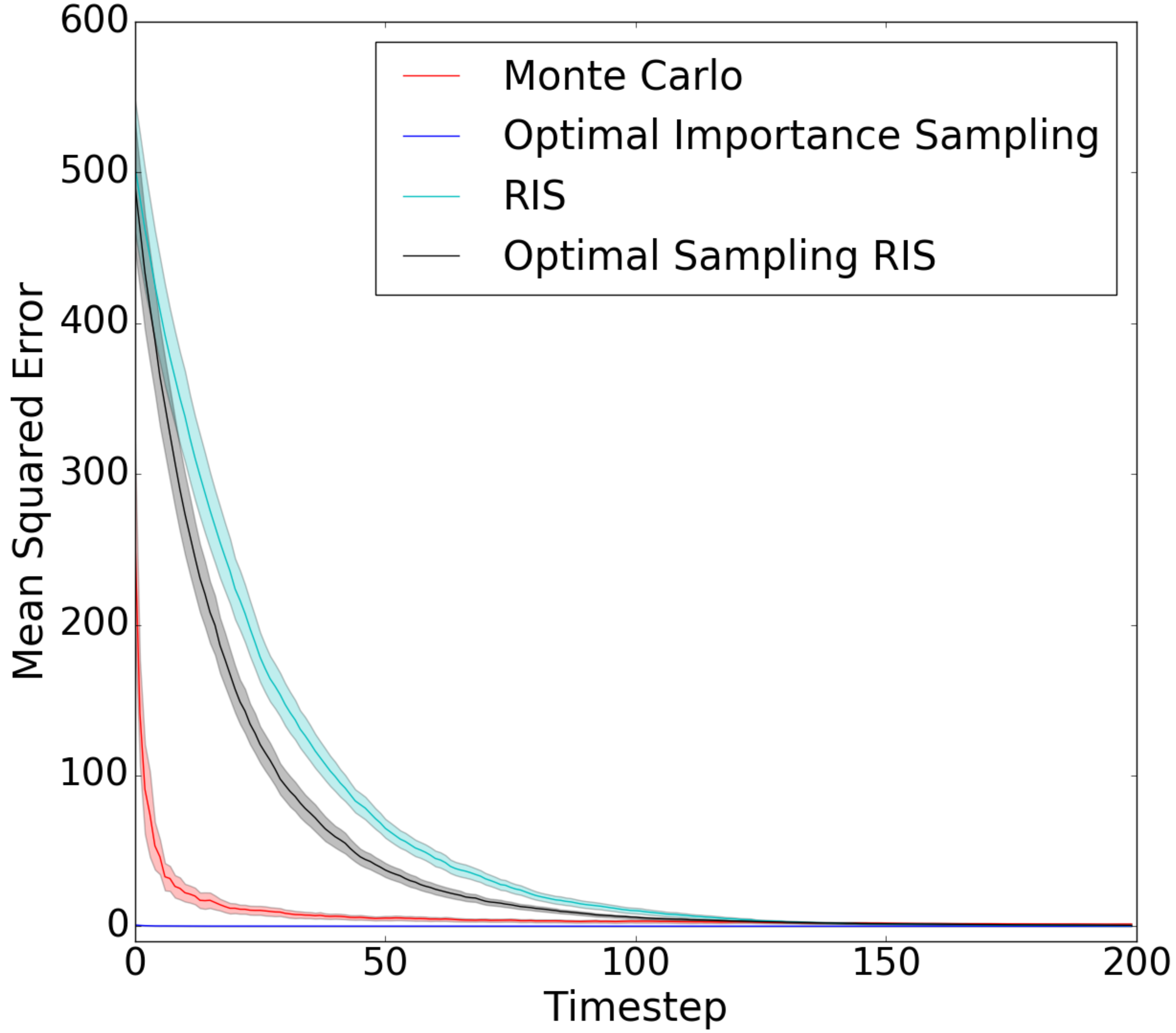
Josiah Hanna

Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

50-armed bandit with stochastic rewards.

RIS needs to observe every arm!



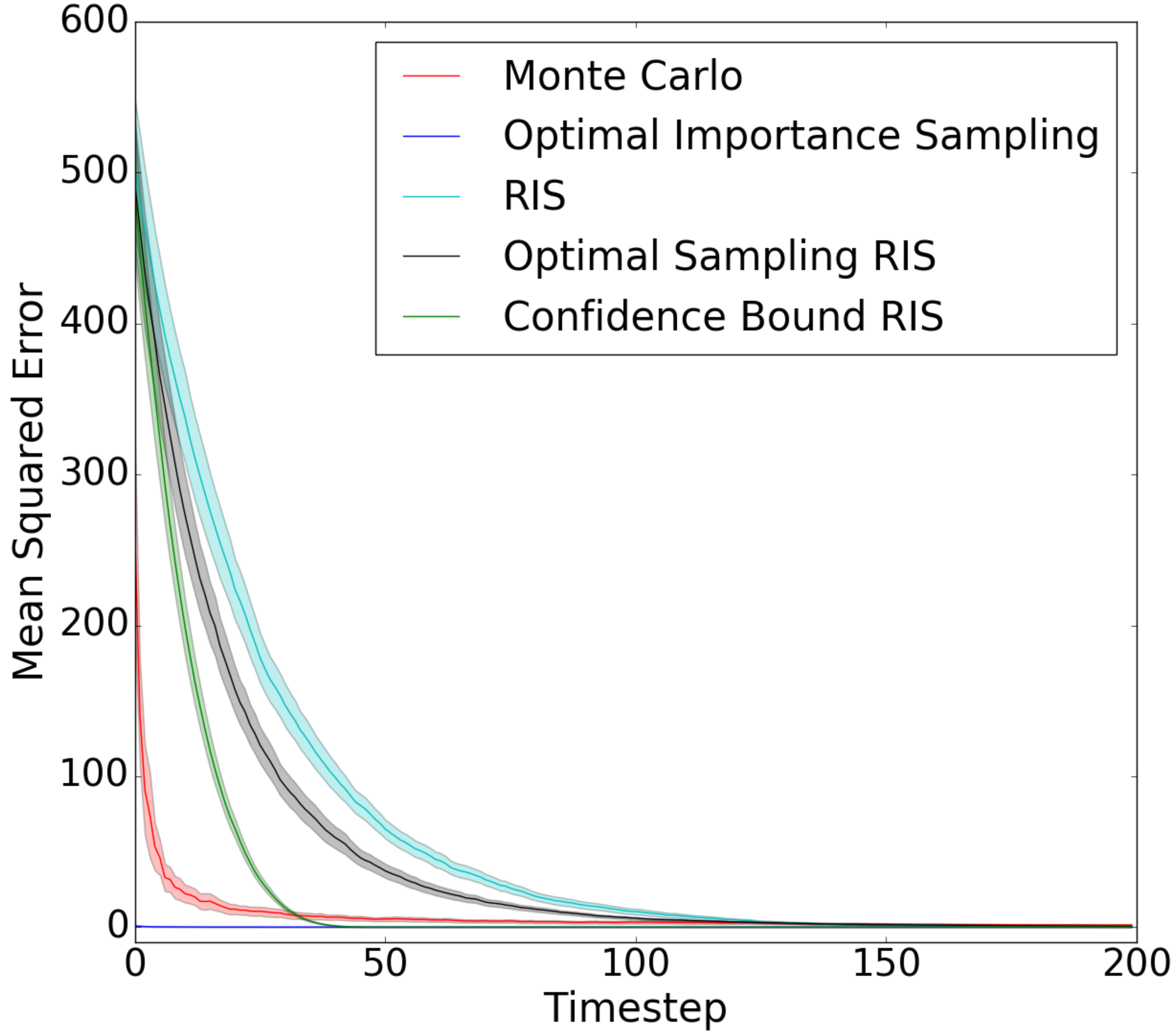
Josiah Hanna

Optimal sampling for regression importance sampling

$$\left(\prod_{t=0}^{L} \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}\right) \times \left(\sum_{t=0}^{L} R_t\right)$$

50-armed bandit with stochastic rewards.

RIS needs to observe every arm!



Josiah Hanna

Value function learning with RIS and BPG

Value function learning with RIS and BPG

$$v_{t+1}(S_t) \leftarrow v_t(S_t) + \alpha(U_t - v_t(S_t))$$

Value function learning with RIS and BPG

$$v_{t+1}(S_t) \leftarrow v_t(S_t) + \alpha(U_t - v_t(S_t))$$

$$U_t = \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}(R_t + v_t(S_{t+1}))$$

Value function learning with RIS and BPG

$$v_{t+1}(S_t) \leftarrow v_t(S_t) + \alpha(U_t - v_t(S_t))$$

$$U_t = \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}(R_t + v_t(S_{t+1}))$$

Collecting data:

Value function learning with RIS and BPG

$$v_{t+1}(S_t) \leftarrow v_t(S_t) + \alpha(U_t - v_t(S_t))$$

$$U_t = \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}(R_t + v_t(S_{t+1}))$$

Collecting data:

What is optimal behavior policy with changing value function?

Value function learning with RIS and BPG

$$v_{t+1}(S_t) \leftarrow v_t(S_t) + \alpha(U_t - v_t(S_t))$$

$$U_t = \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}(R_t + v_t(S_{t+1}))$$

Collecting data:

What is optimal behavior policy with changing value function?

Weighting data:

Josiah Hanna

Value function learning with RIS and BPG

$$v_{t+1}(S_t) \leftarrow v_t(S_t) + \alpha(U_t - v_t(S_t))$$

$$U_t = \frac{\pi(A_t|S_t)}{\pi_b(A_t|S_t)}(R_t + v_t(S_{t+1}))$$

Collecting data:

What is optimal behavior policy with changing value function?

Weighting data:

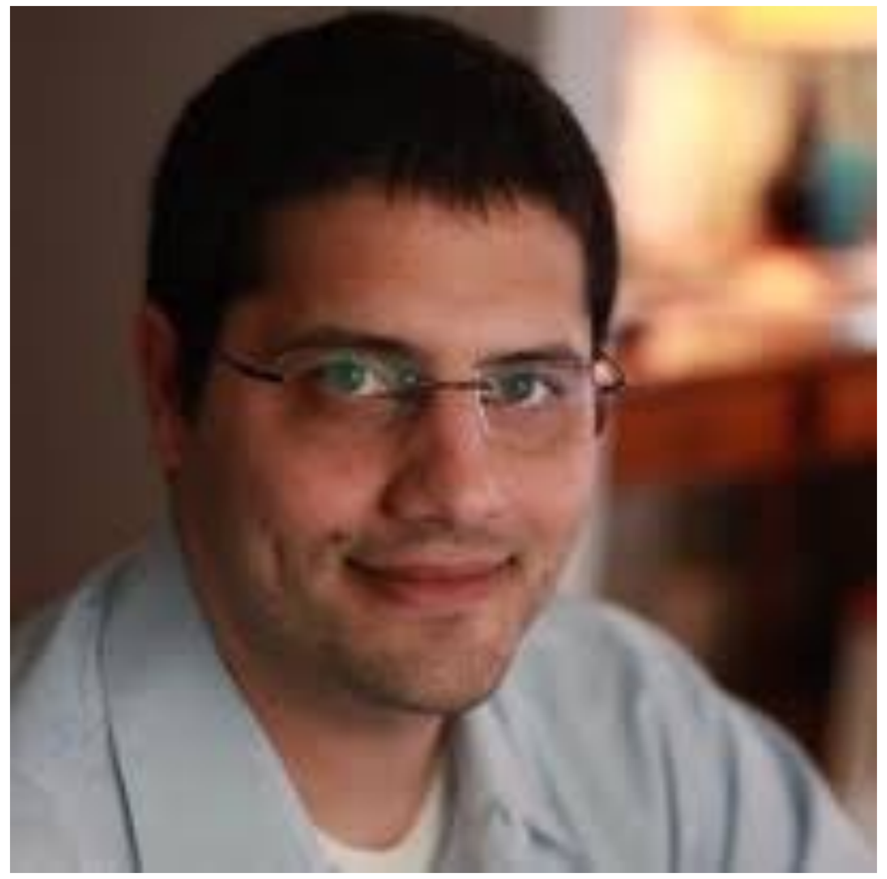
How to estimate behavior policy during online learning?

Josiah Hanna

Acknowledgments


Peter Stone

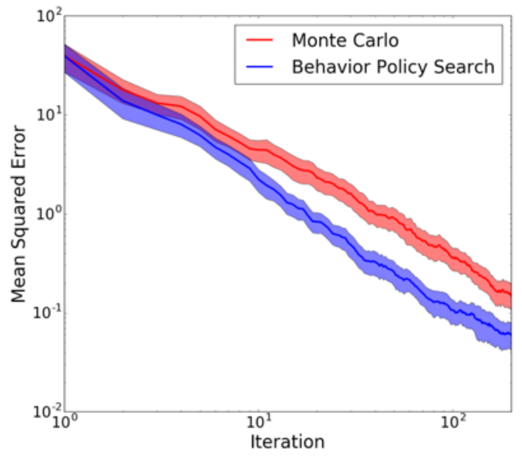

Scott Niekum


Phil Thomas

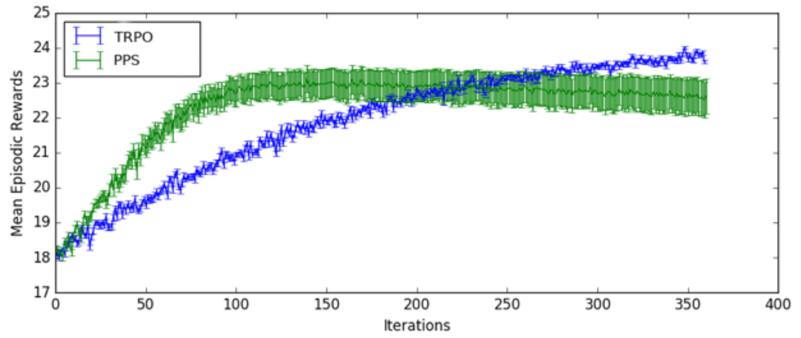

Xiang Gu

How can a reinforcement learning agent leverage off-policy and simulated data to evaluate and improve upon the expected performance of a policy?

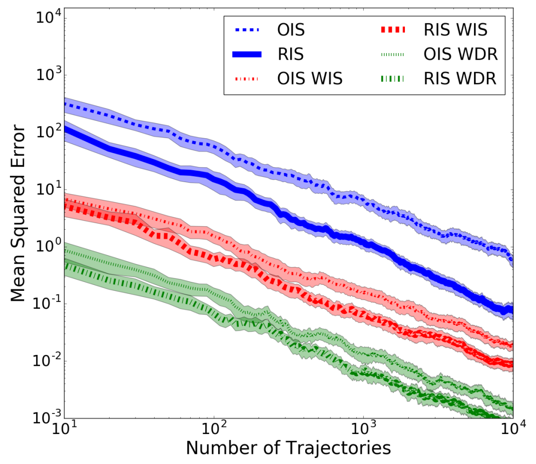
How should an RL agent collect off-policy data?



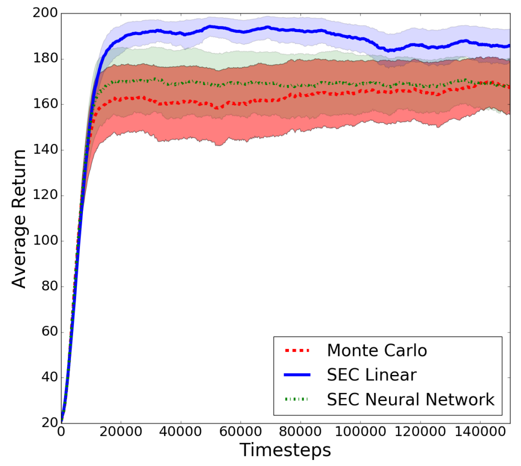
ICML 2017
AAAI SS 2018



How should an RL agent weight off-policy data?

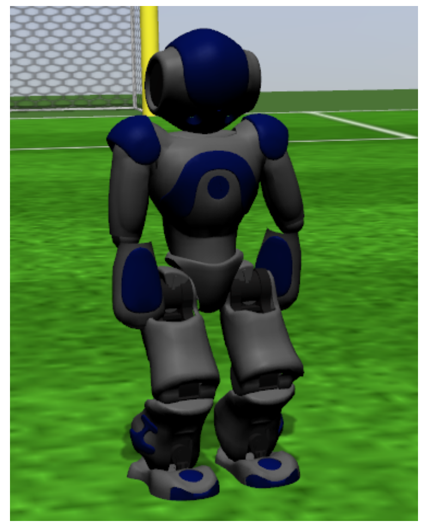


AAMAS 2019
ICML 2019



How can an RL agent use simulated data?

AAAI 2017



How can an RL agent combine simulated and off-policy data?

AAMAS 2017