PC-PG: Policy Cover Directed Exploration for Provable Policy Gradient Learning

Joint work with Alekh Agarwal, Mikael Henaff, and Sham Kakade

Microsoft Research

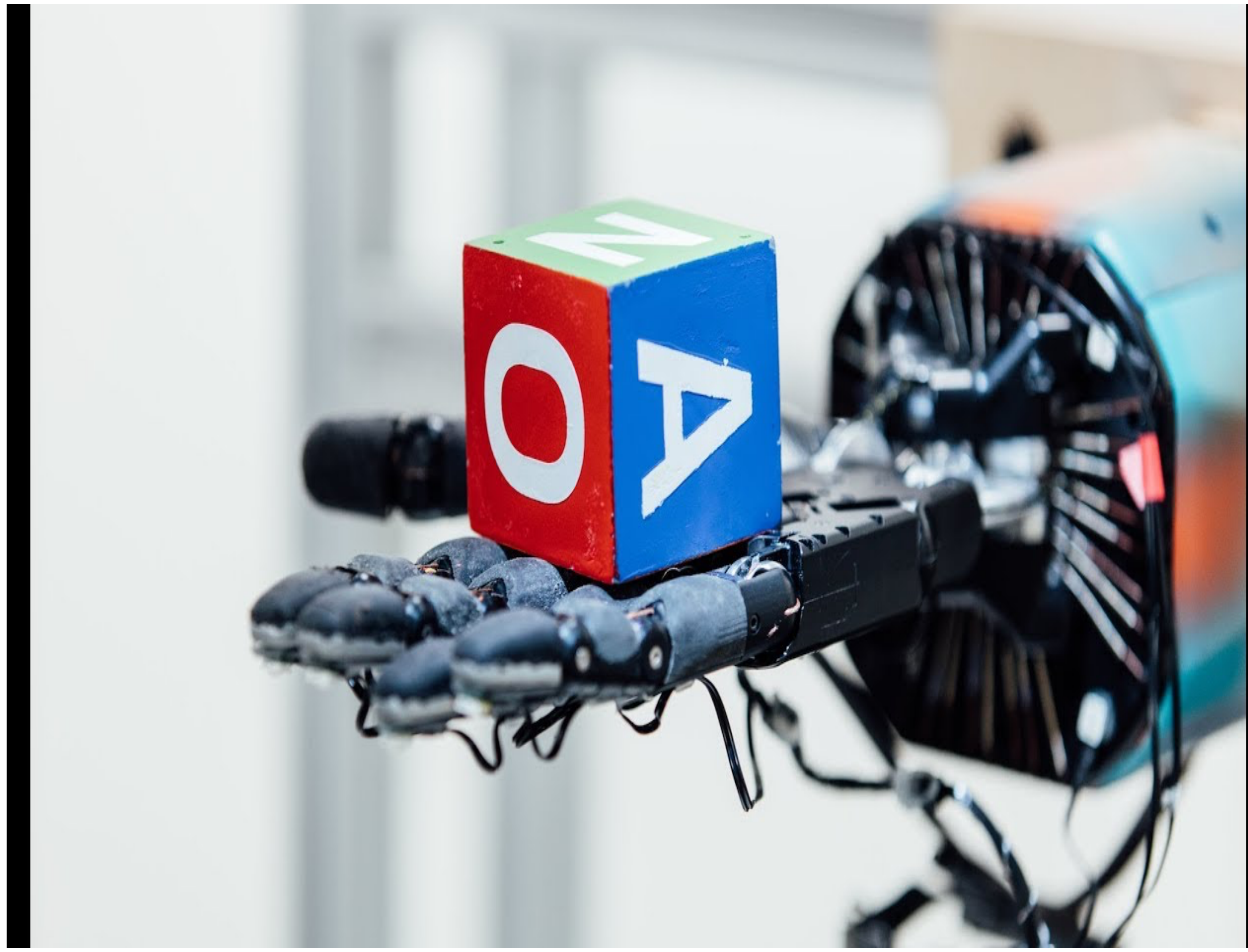
Facebook AI Research

W
UNIVERSITY of WASHINGTON

Policy Optimization

[AlphaZero, Silver et.al, 17]

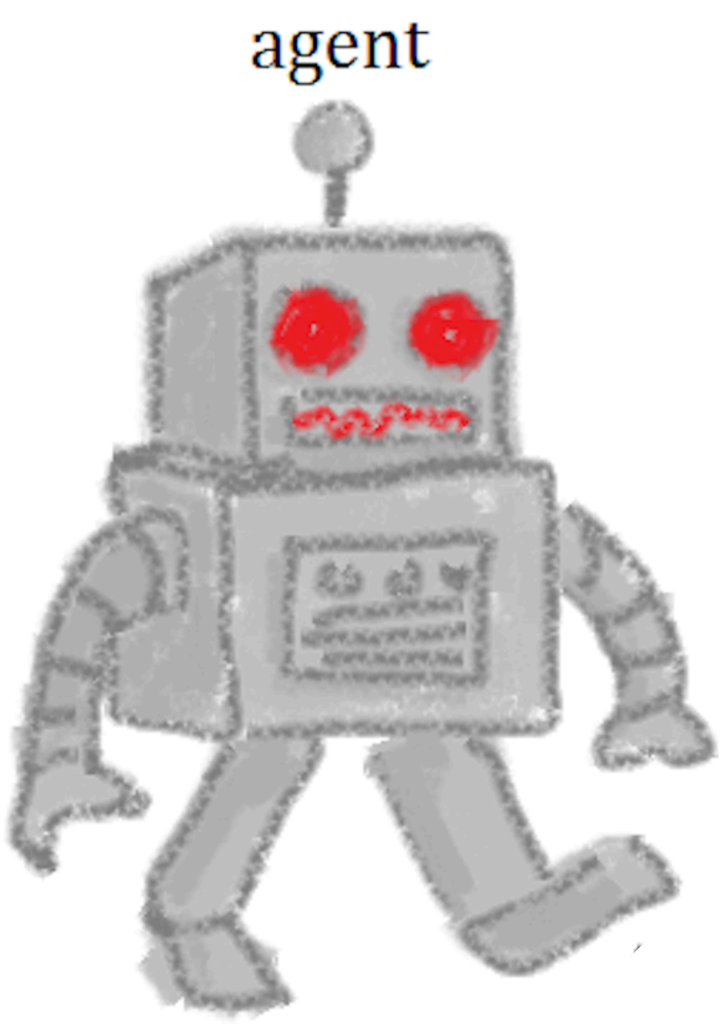
[OpenAI Five, 18]



[OpenAI,19]

Can we design Provably Correct Policy Gradient algorithms?

Infinite Horizon Discounted MDPs



Policy: state to action

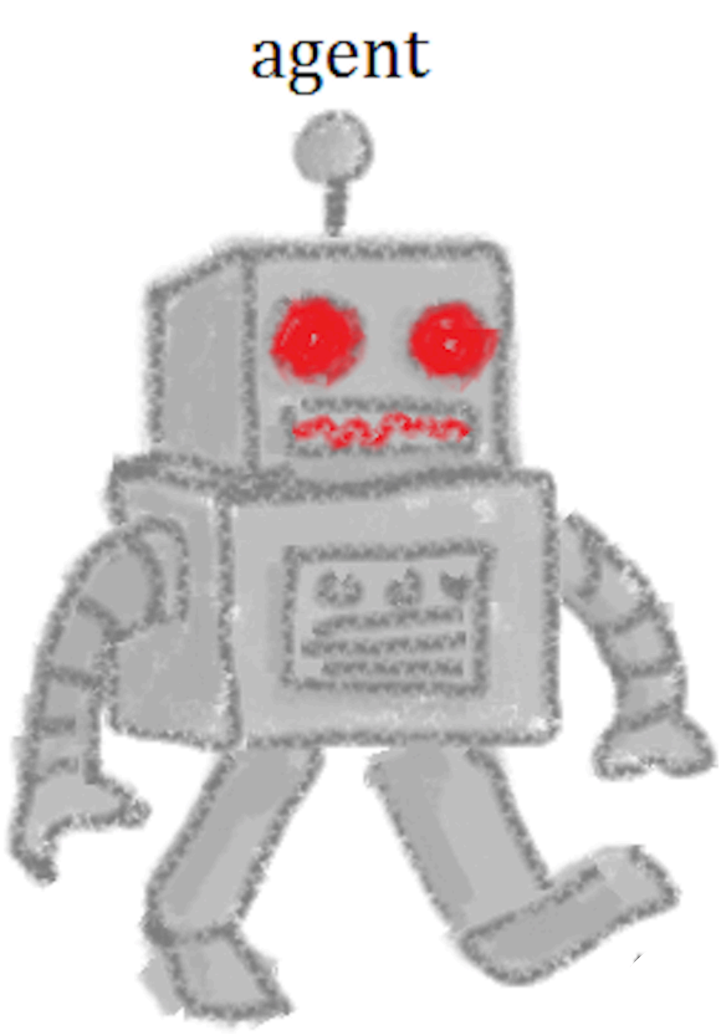
$$\pi(s) \rightarrow a$$

Reward & Next State

$$r(s, a), s' \sim P(\cdot \mid s, a)$$

Objective: $\max_{\pi} \mathbb{E}_{\pi,P} \left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots \right]$

Infinite Horizon Discounted MDPs



Policy: state to action

$$\pi(s) \to a$$

Reward & Next State

$$r(s, a), s' \sim P(\cdot \,|\, s, a)$$

Objective: $\max_{\pi} \mathbb{E}_{\pi,P}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots\right]$

Assume $s_0 \sim \mu_0$ and we can only reset from μ_0

Policy Gradient Methods

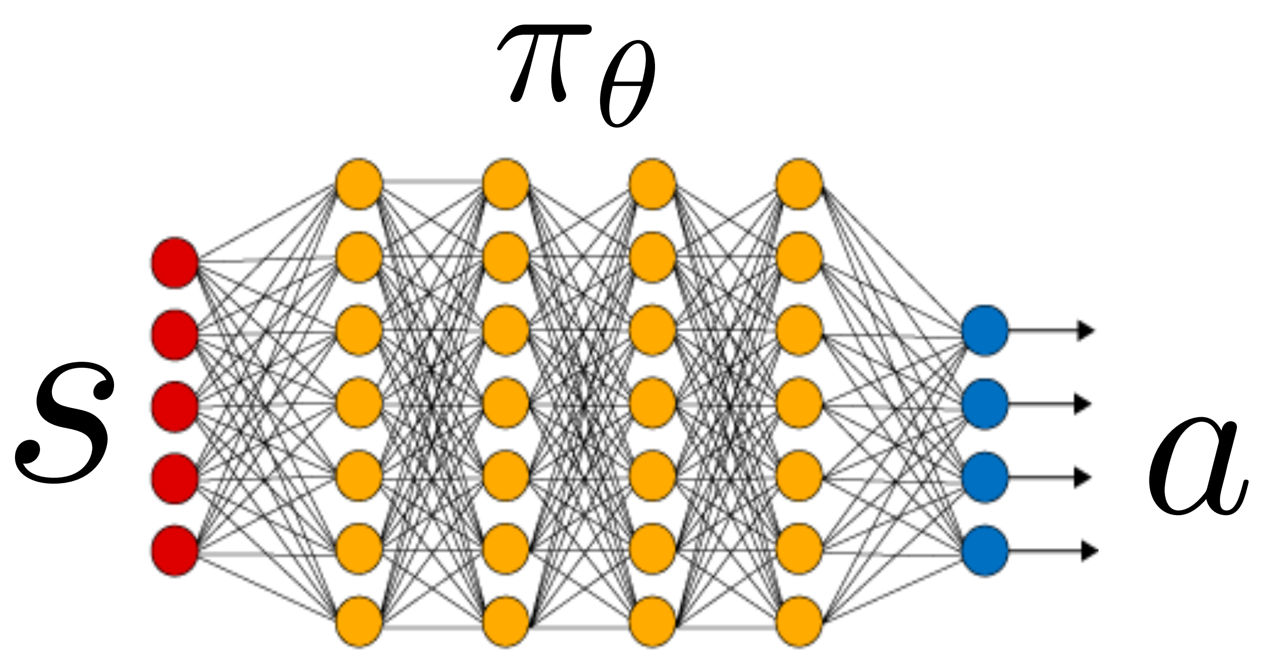
e.g., Reinforce, Natural Policy Gradient, TRPO, PPO:

(Williams 92, Kakade 02, Schulman et al 15, 17)

Policy Gradient Methods

e.g., Reinforce, Natural Policy Gradient, TRPO, PPO:

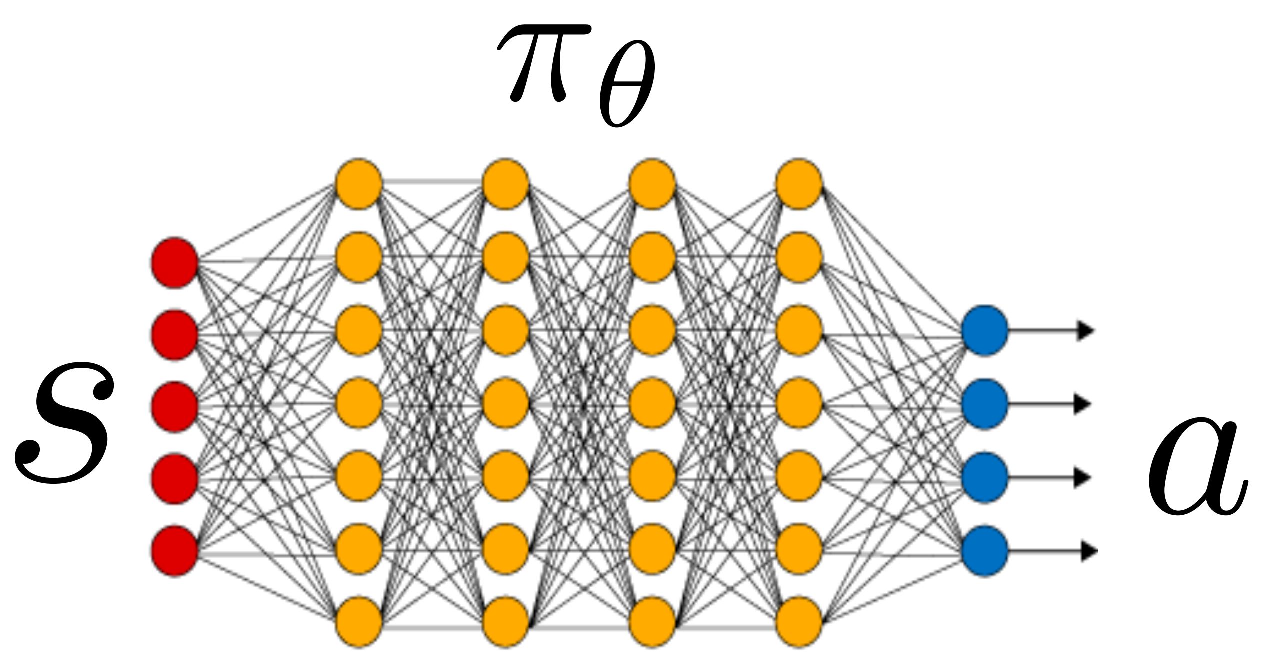
(Williams 92, Kakade 02, Schulman et al 15, 17)



Policy Gradient Methods

e.g., Reinforce, Natural Policy Gradient, TRPO, PPO:

(Williams 92, Kakade 02, Schulman et al 15, 17)

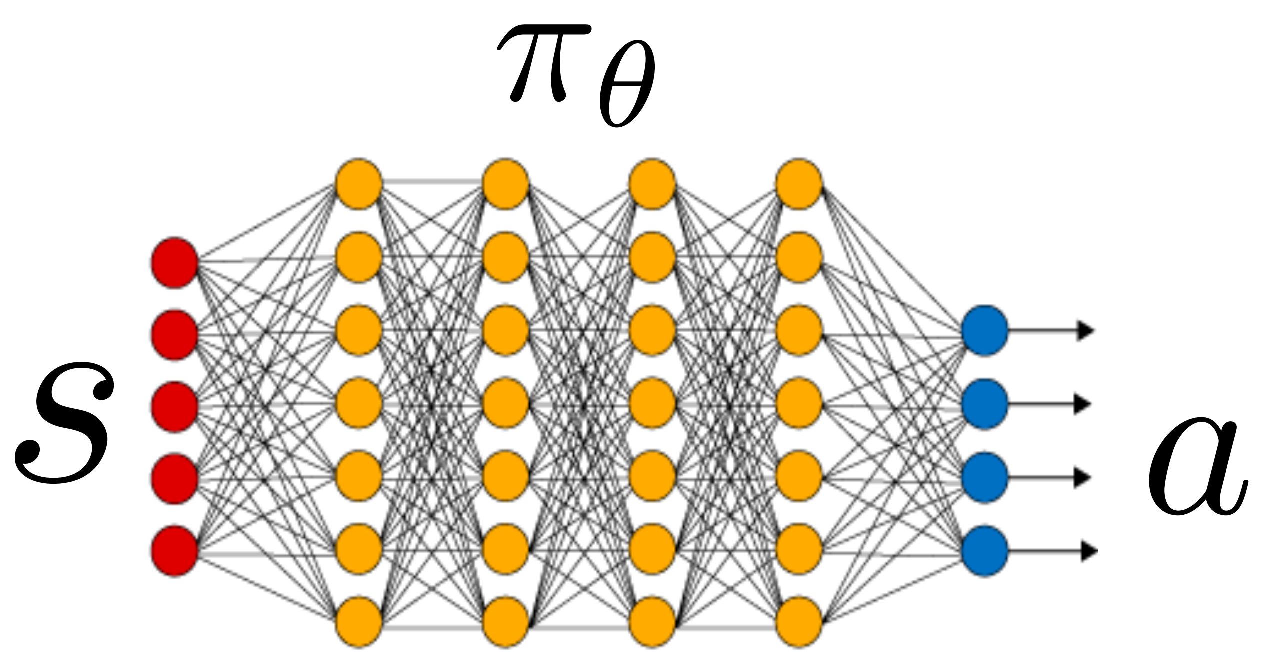


$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots\right]$$

Policy Gradient Methods

e.g., Reinforce, Natural Policy Gradient, TRPO, PPO:

(Williams 92, Kakade 02, Schulman et al 15, 17)



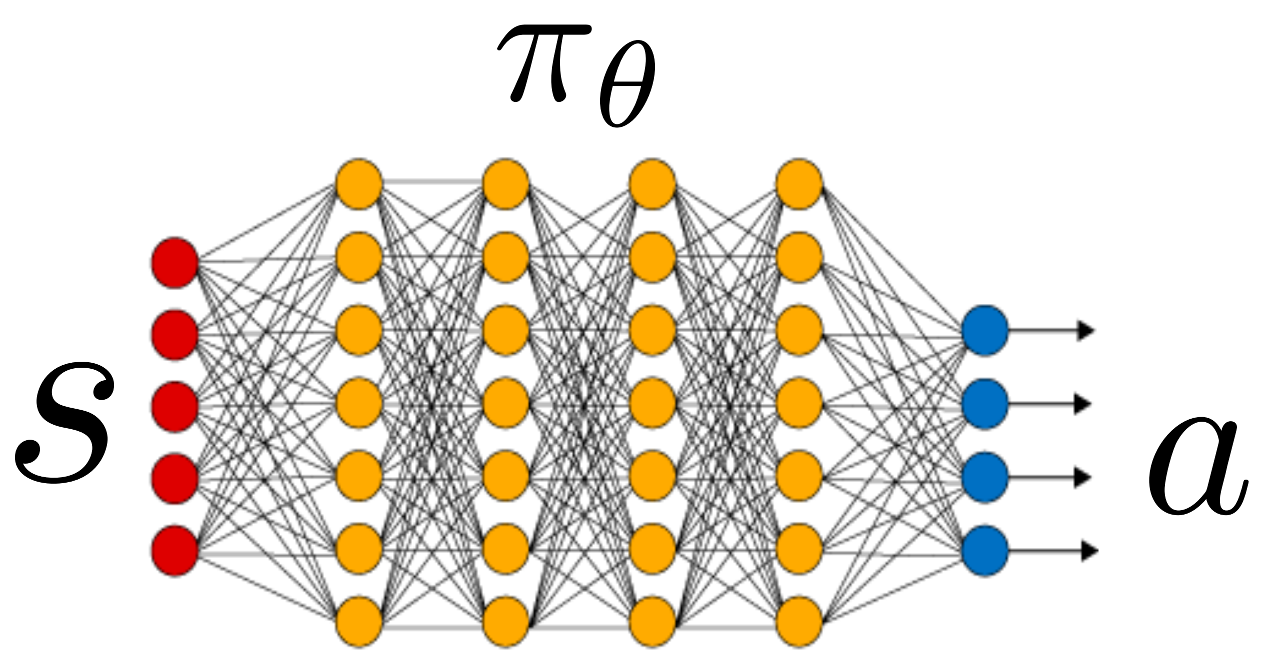
$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots\right]$$

$$\theta = \theta + \eta \nabla_\theta J(\pi_\theta)$$

Policy Gradient Methods

e.g., Reinforce, Natural Policy Gradient, TRPO, PPO:

(Williams 92, Kakade 02, Schulman et al 15, 17)



$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots\right]$$

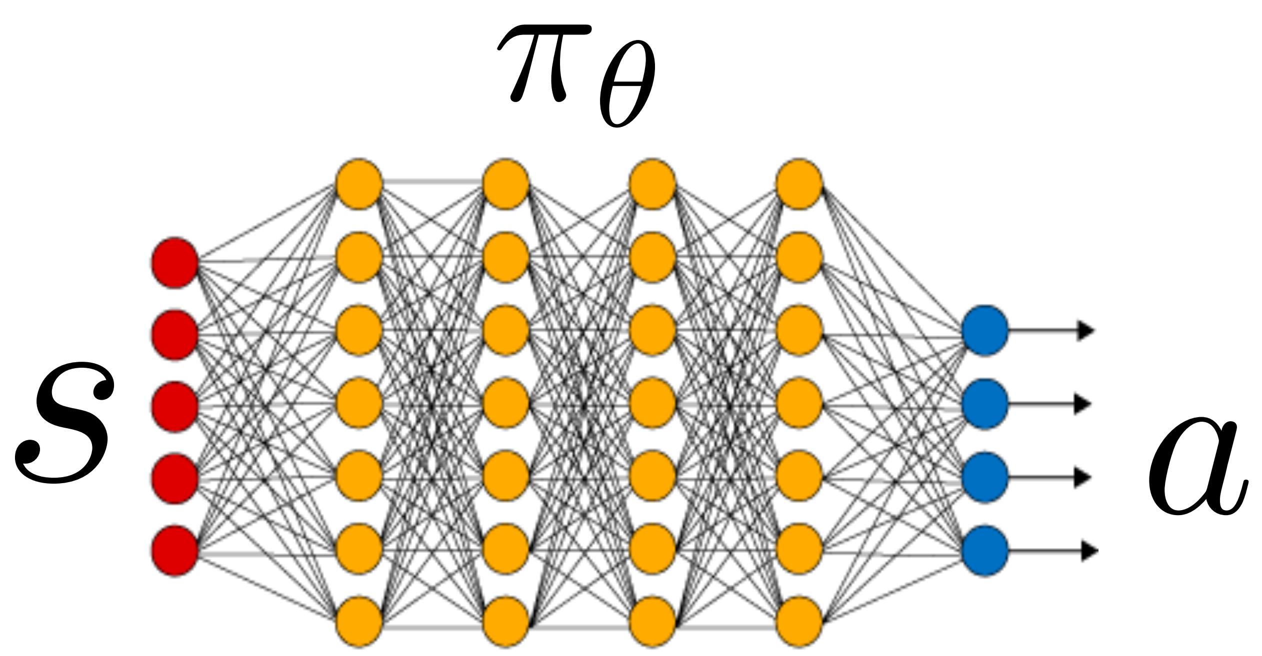
$$\theta = \theta + \eta \nabla_\theta J(\pi_\theta)$$

$$\theta = \theta + \eta F_\theta^{-1} \nabla_\theta J(\pi_\theta)$$

Policy Gradient Methods

e.g., Reinforce, Natural Policy Gradient, TRPO, PPO:

(Williams 92, Kakade 02, Schulman et al 15, 17)



$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots\right]$$

$$\theta = \theta + \eta \nabla_\theta J(\pi_\theta)$$

$$\theta = \theta + \eta F_\theta^{-1} \nabla_\theta J(\pi_\theta)$$

Preconditioning w/ Fisher Information matrix

(TRPO and PPO are variants of it)

Advantages of Policy Gradient Methods

Strong Agnostic guarantee:

Compete to the best policy in the given class: $\widetilde{\pi} \in \Pi$

Advantages of Policy Gradient Methods

Strong Agnostic guarantee:

Compete to the best policy in the given class: $\widetilde{\pi} \in \Pi$

under a "wide" reset distribution (e.g., see Agarwal et al 19)

Advantages of Policy Gradient Methods

Strong Agnostic guarantee:

Compete to the best policy in the given class: $\widetilde{\pi} \in \Pi$

under a "wide" reset distribution (e.g., see Agarwal et al 19)

State distribution of $\widetilde{\pi}$

$$\sup_s \left| \frac{d^{\widetilde{\pi}}(s)}{\mu_0(s)} \right| < \infty$$

Initial/reset dist

Advantages of Policy Gradient Methods

Strong Agnostic guarantee:

Compete to the best policy in the given class: $\widetilde{\pi} \in \Pi$

under a "wide" reset distribution (e.g., see Agarwal et al 19)

State distribution of $\widetilde{\pi}$

$$\sup_s \left| \frac{d^{\widetilde{\pi}}(s)}{\mu_0(s)} \right| < \infty$$

Initial/reset dist

Q-learning, Fitted Q iteration:

Realizability (& Bellman Complete)

$$Q^\star \in \mathcal{Q}$$

Successful Story of PG:

Robot hand manipulation (OpenAI, 19)

Successful Story of PG:

Robot hand manipulation (OpenAI, 19)

Successful Story of PG:

Robot hand manipulation (OpenAI, 19)

Successful Story of PG:

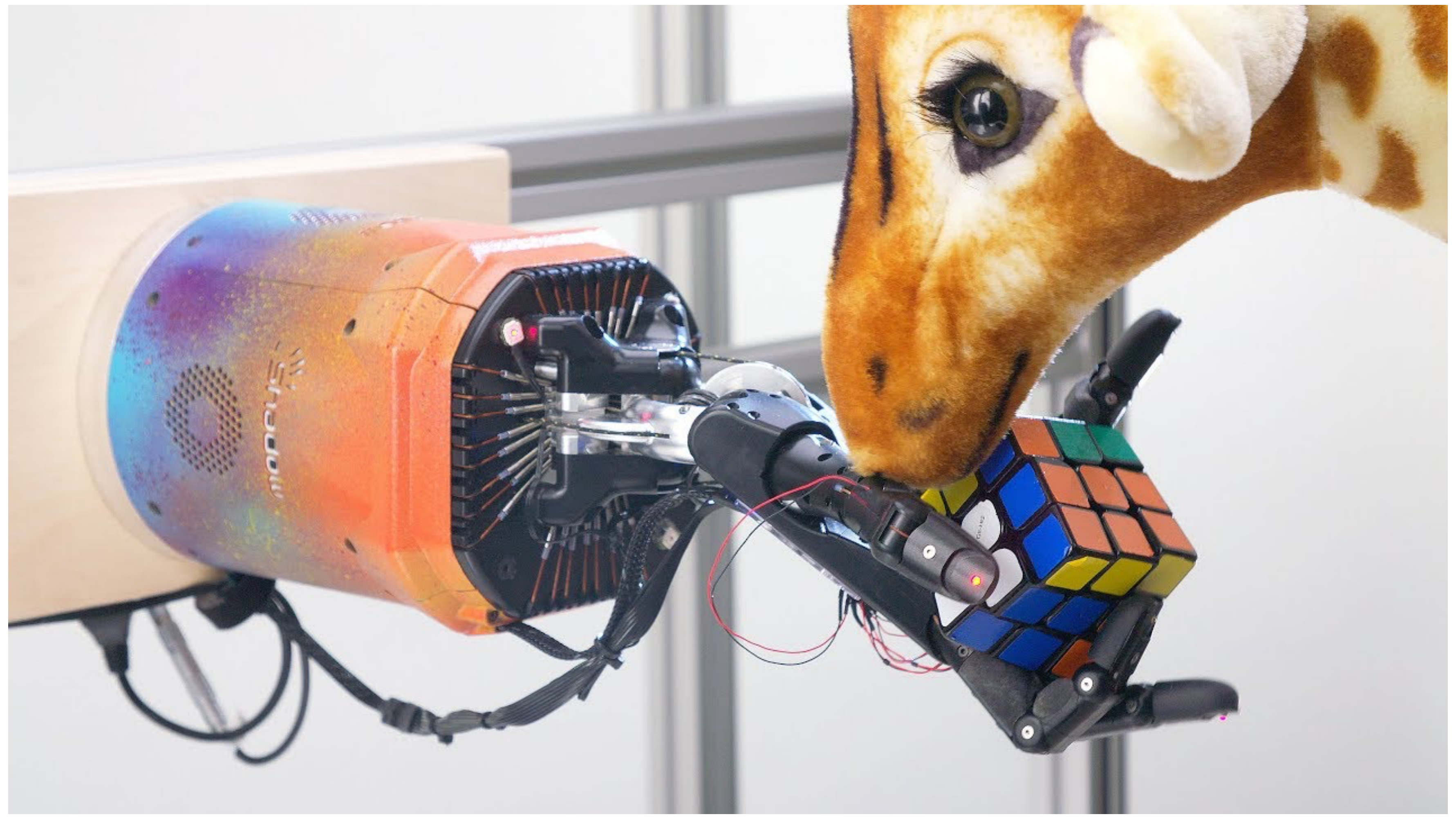
Robot hand manipulation (OpenAI, 19)

A notable technique

Domain Randomization:

Successful Story of PG:

Robot hand manipulation (OpenAI, 19)

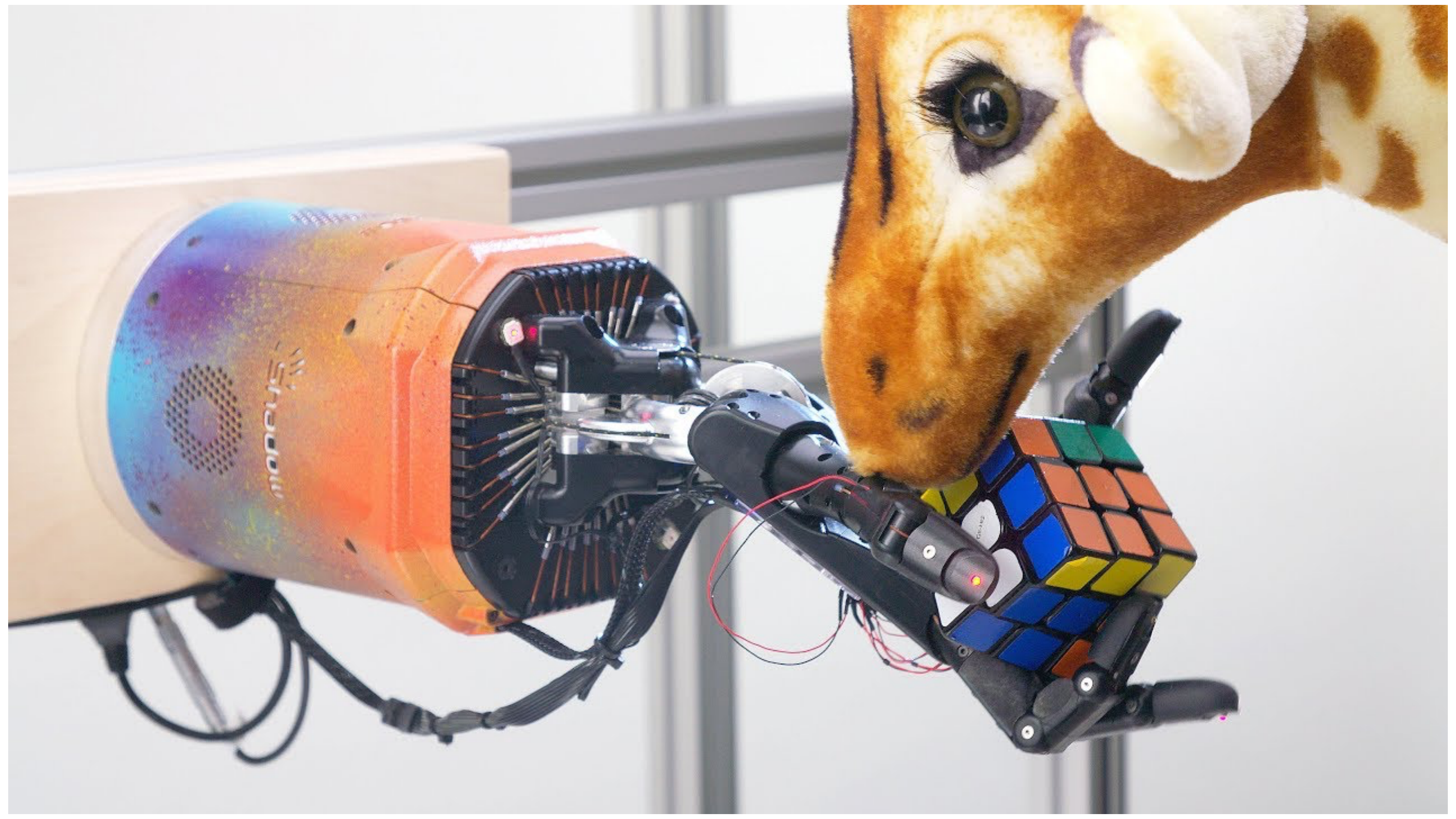


A notable technique

Domain Randomization:

Successful Story of PG:

Robot hand manipulation (OpenAI, 19)



A notable technique

Domain Randomization:

State dist of the best $\widetilde{\pi}$

$$\sup_s \left| \frac{d^{\widetilde{\pi}}(s)}{\mu_0(s)} \right| < \infty$$

Initial/reset dist

Make μ_0 as "wide" as possible !!

But PG Fails if initial condition does not hold, provably

Initialization: s_0 + random walk

But PG Fails if initial condition does not hold, provably

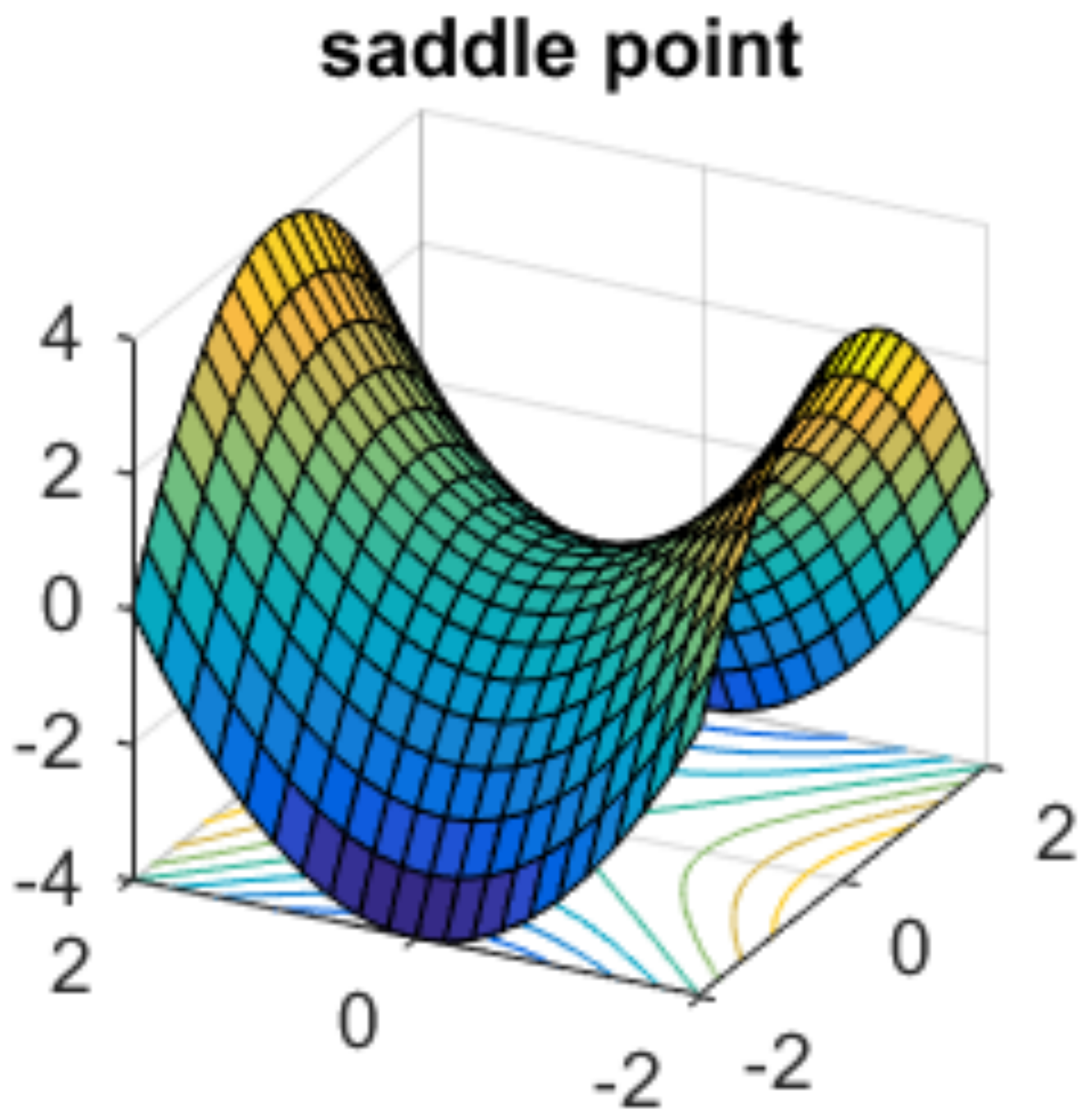
Initialization: s_0 + random walk

Extremely flatten gradient:

magnitude of gradient is exponentially small (even in higher order): 2^{-H}

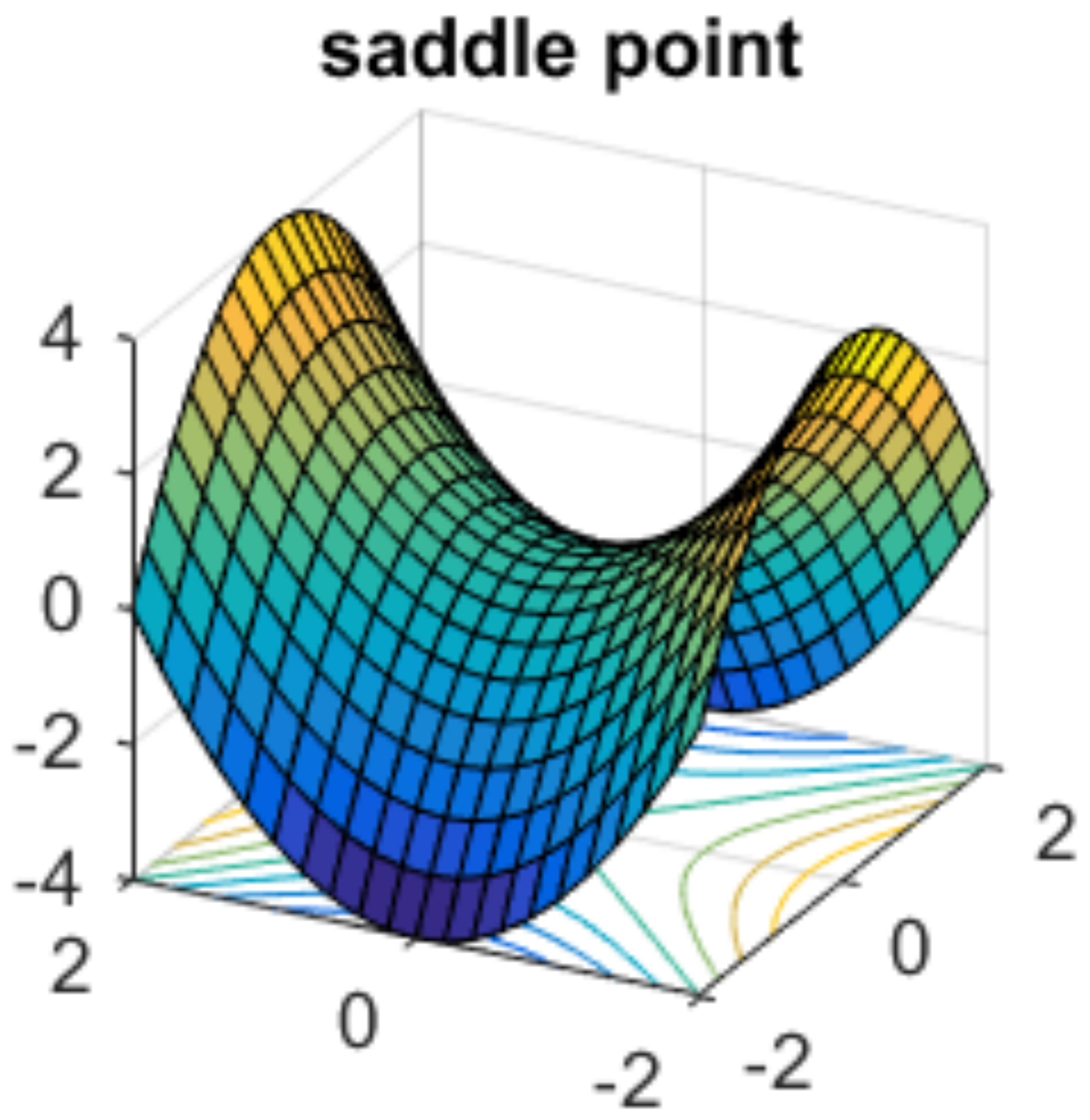
(e.g., see CPI from Kadade & Langford 02, and Agarwal et al 19)

The Optimization Landscape



Supervised Learning

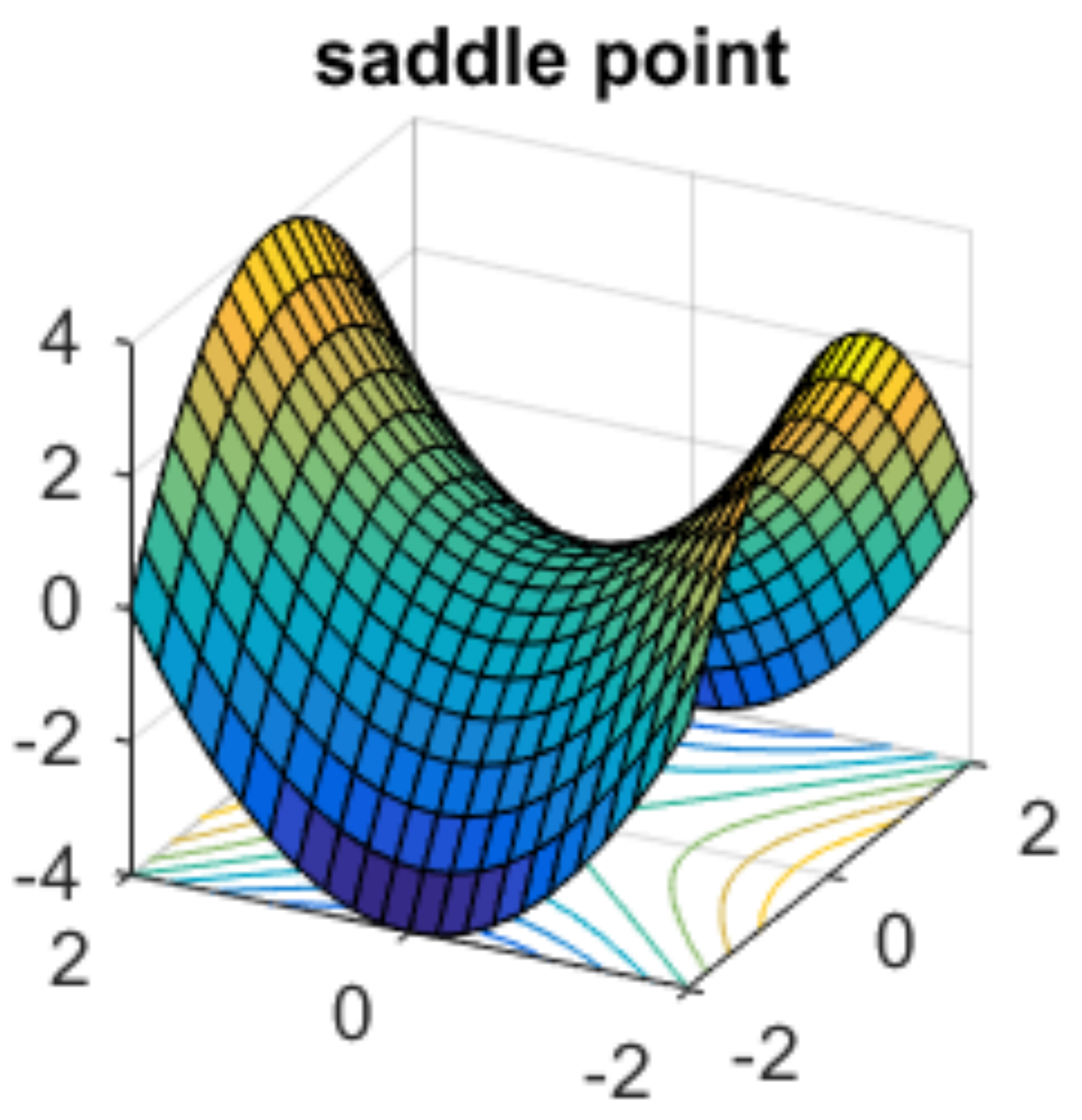
The Optimization Landscape



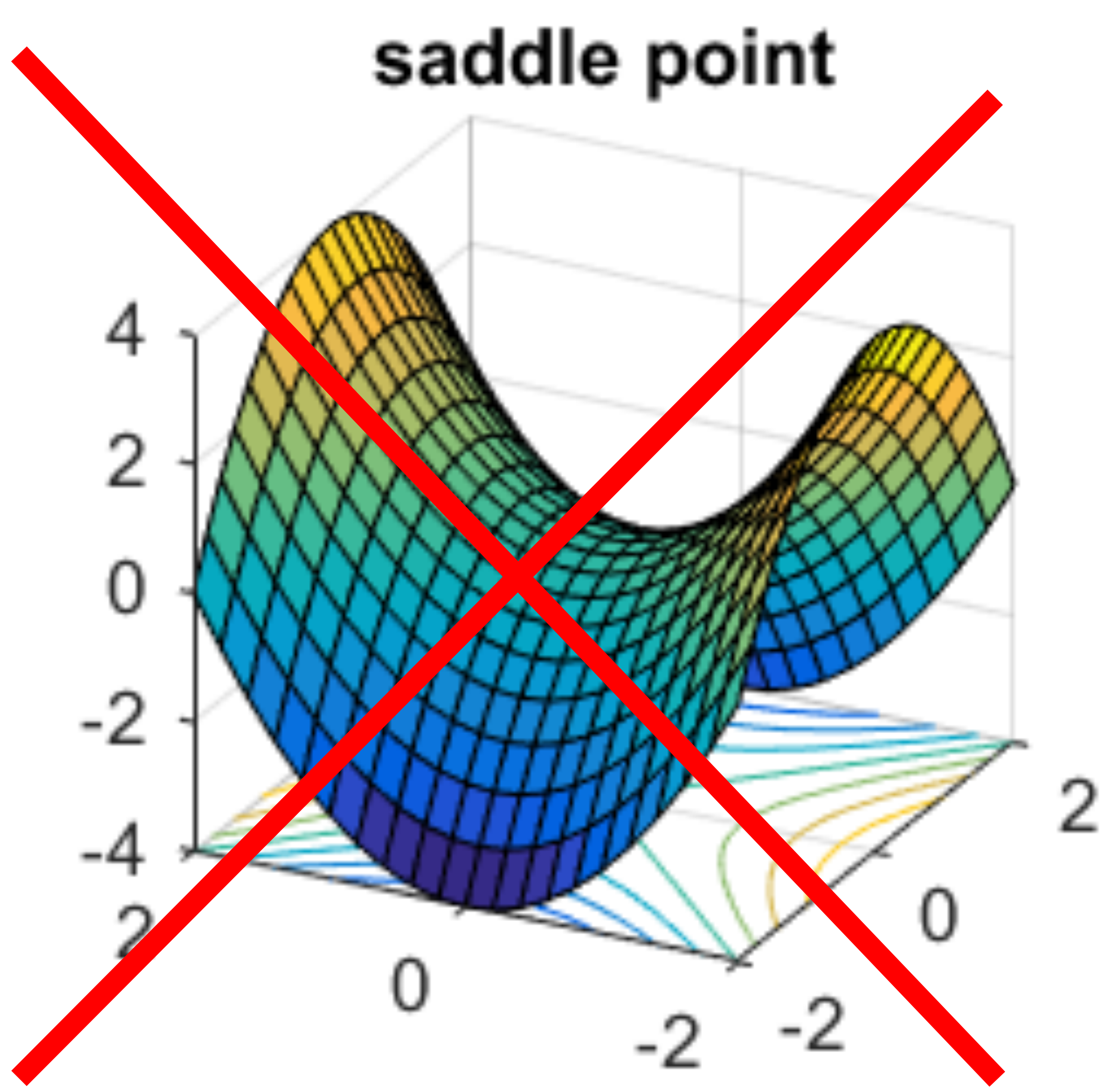
Supervised Learning

- Gradient descent tends to just work
- Not sensitive to initialization
- Saddle point is not a problem

The Optimization Landscape



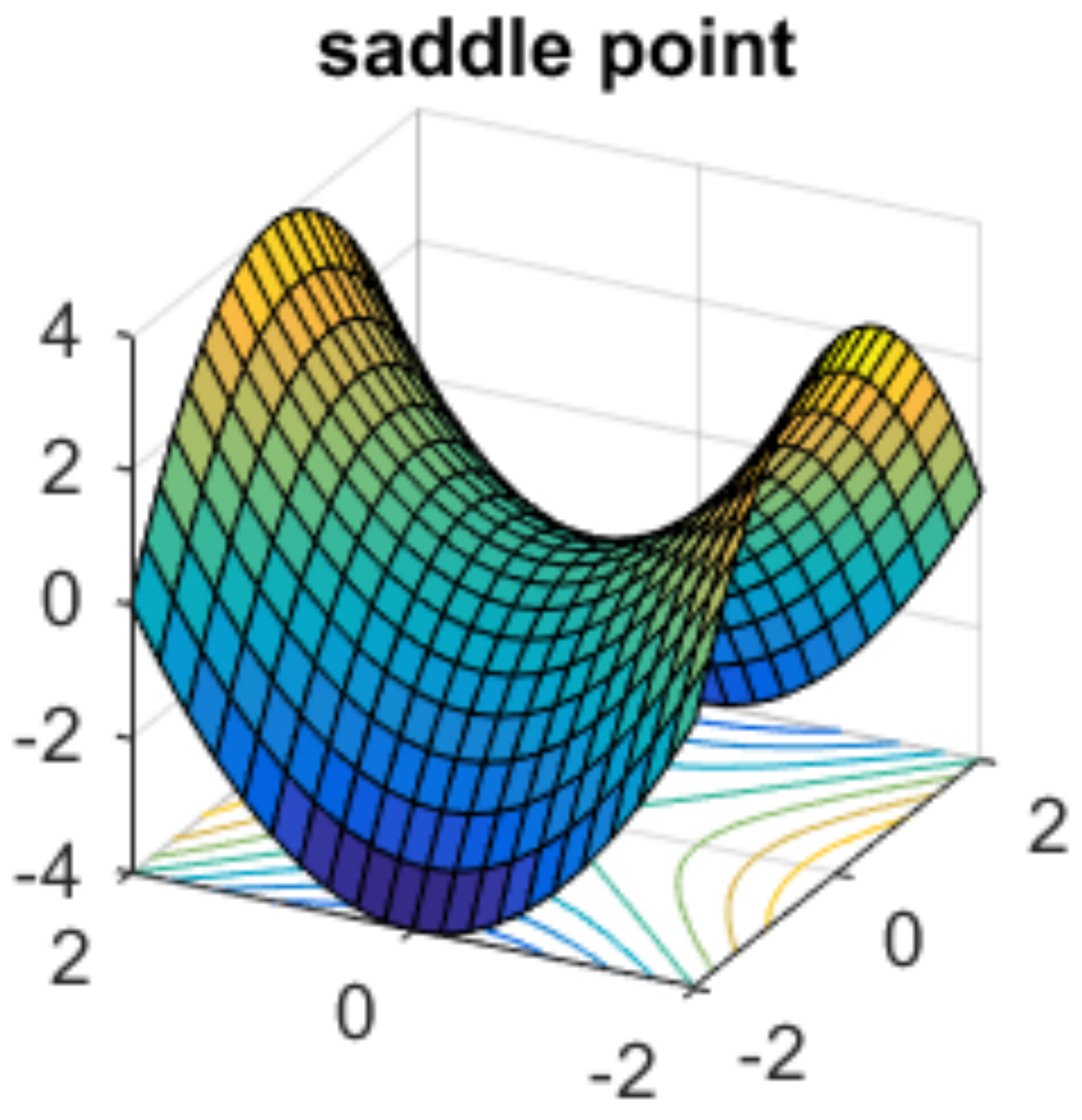
Supervised Learning



Reinforcement Learning:

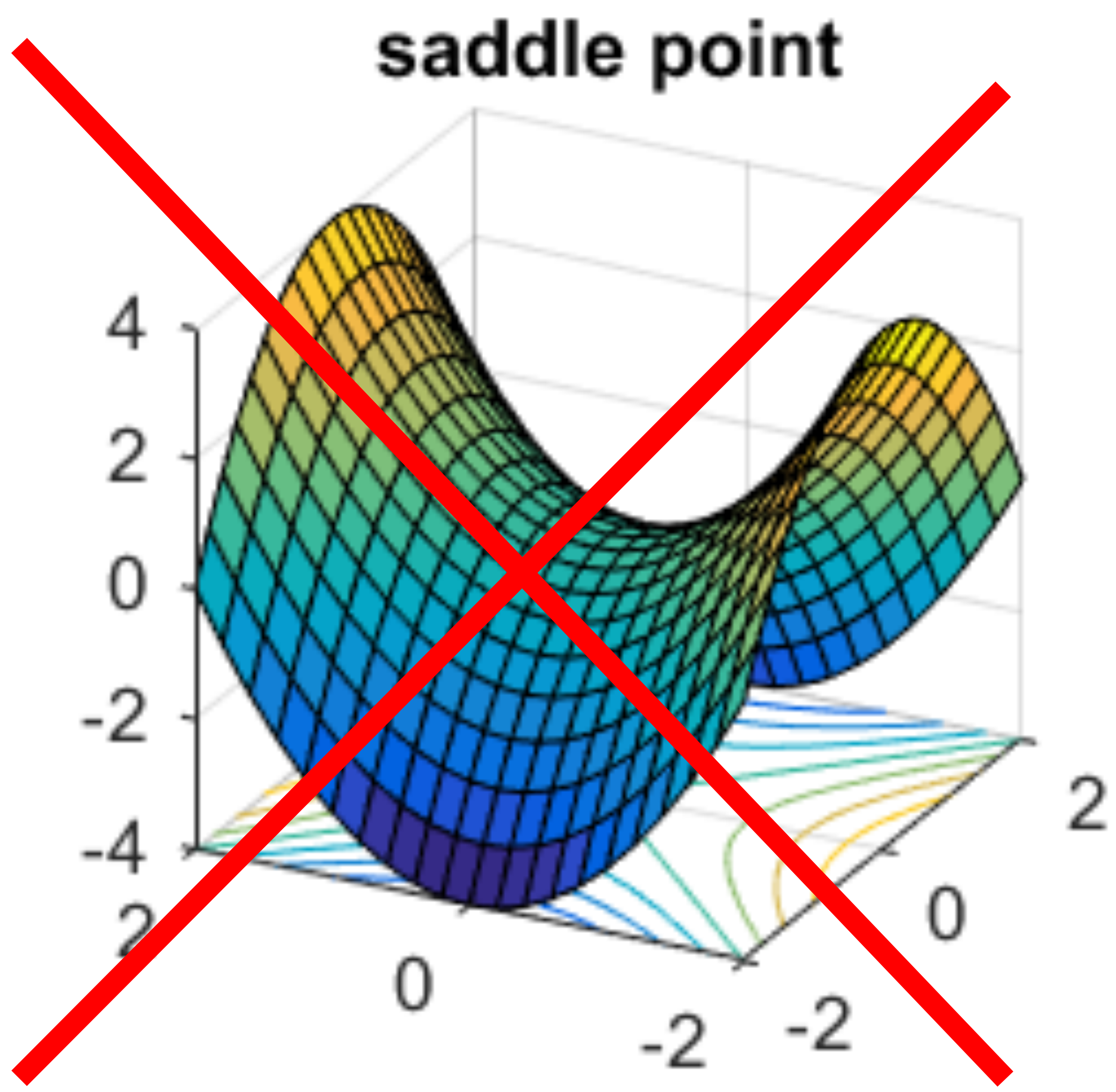
- Gradient descent tends to just work
- Not sensitive to initialization
- Saddle point is not a problem

The Optimization Landscape



Supervised Learning

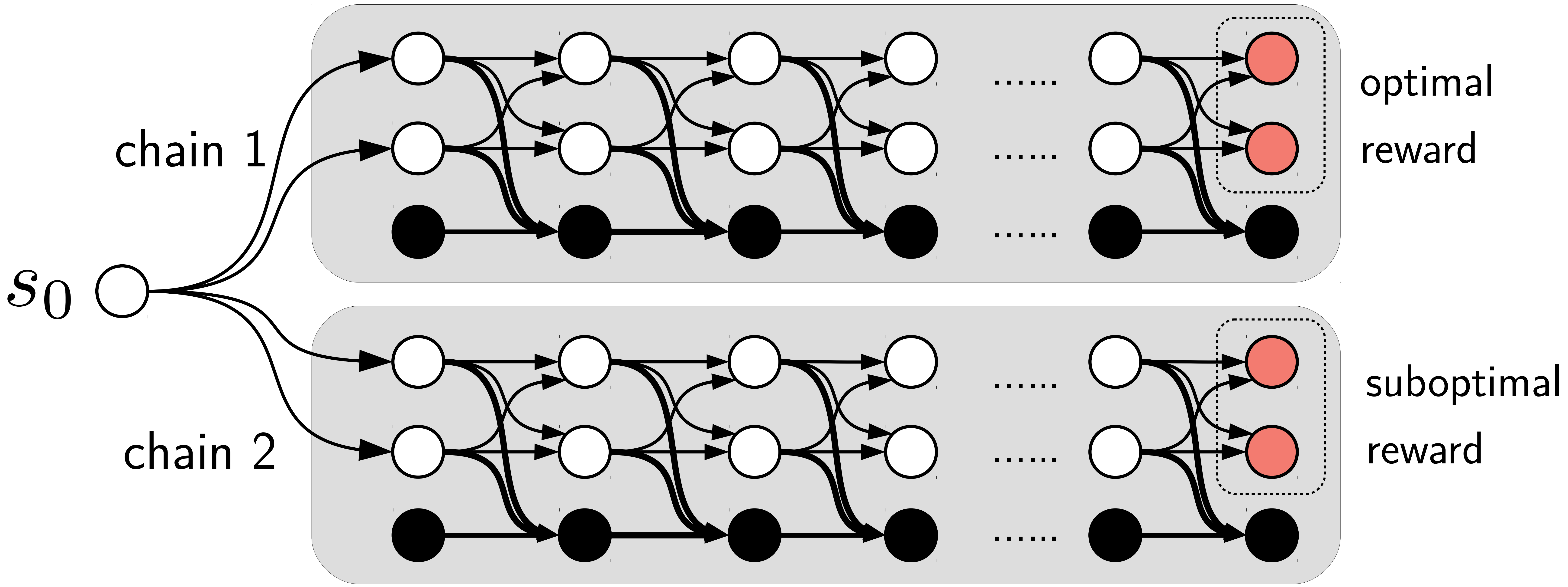
- Gradient descent tends to just work
- Not sensitive to initialization
- Saddle point is not a problem



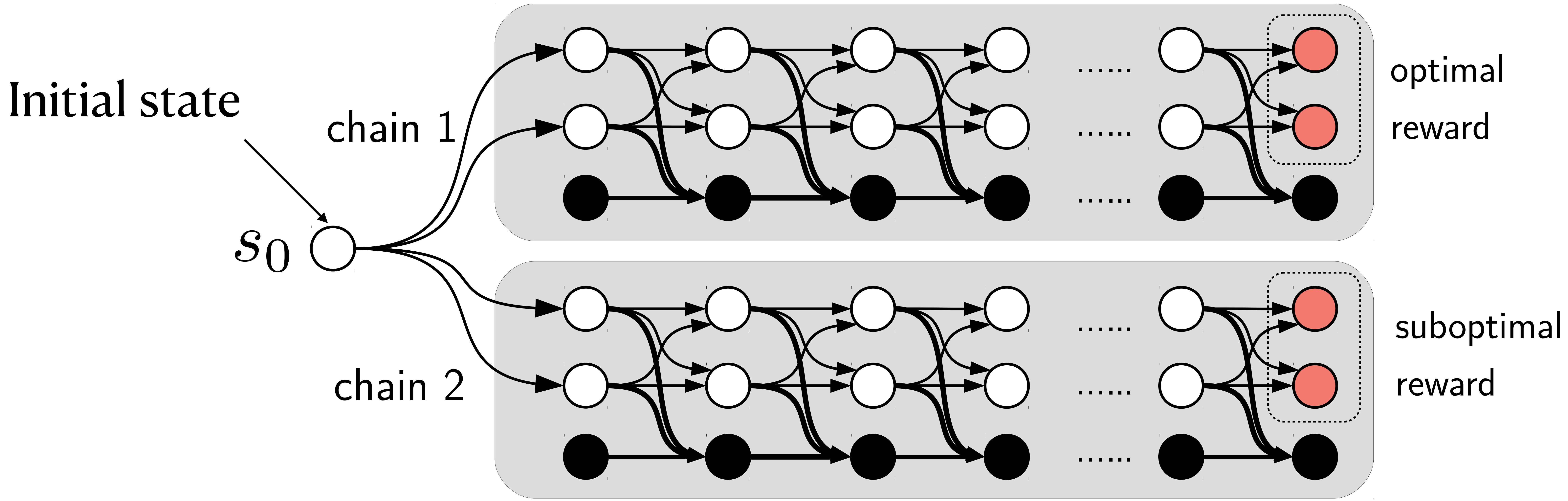
Reinforcement Learning:

- Extremely flatten region even at initialization
- Due to lack of exploration

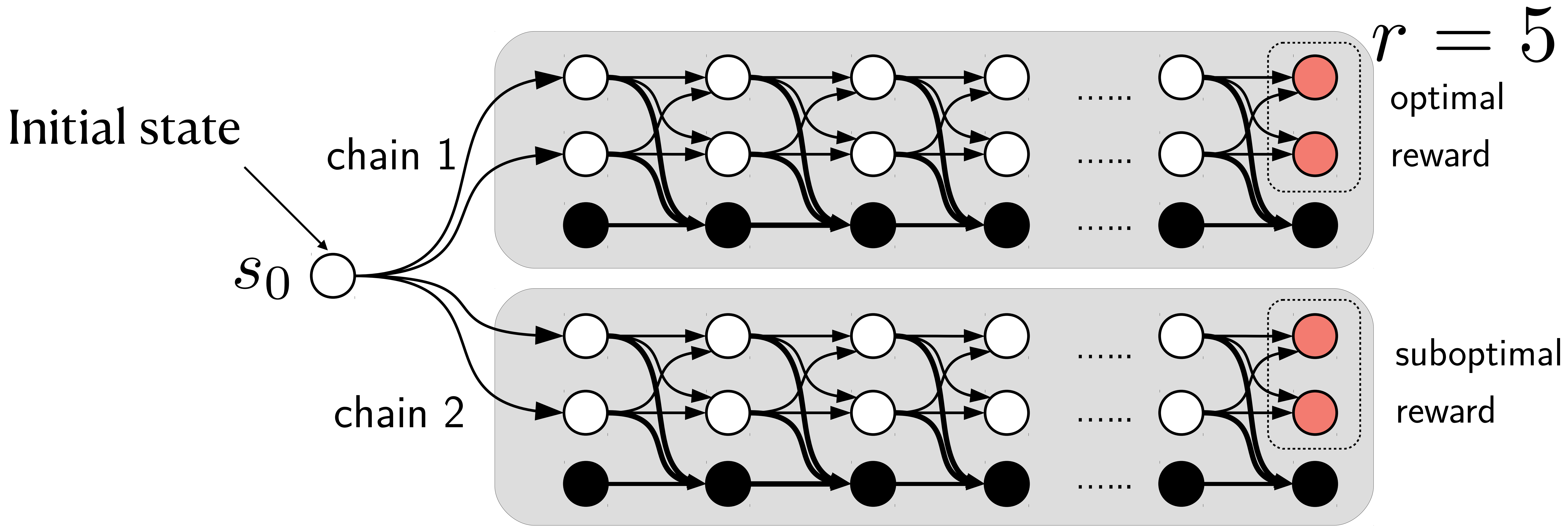
Bidirectional Combination Lock



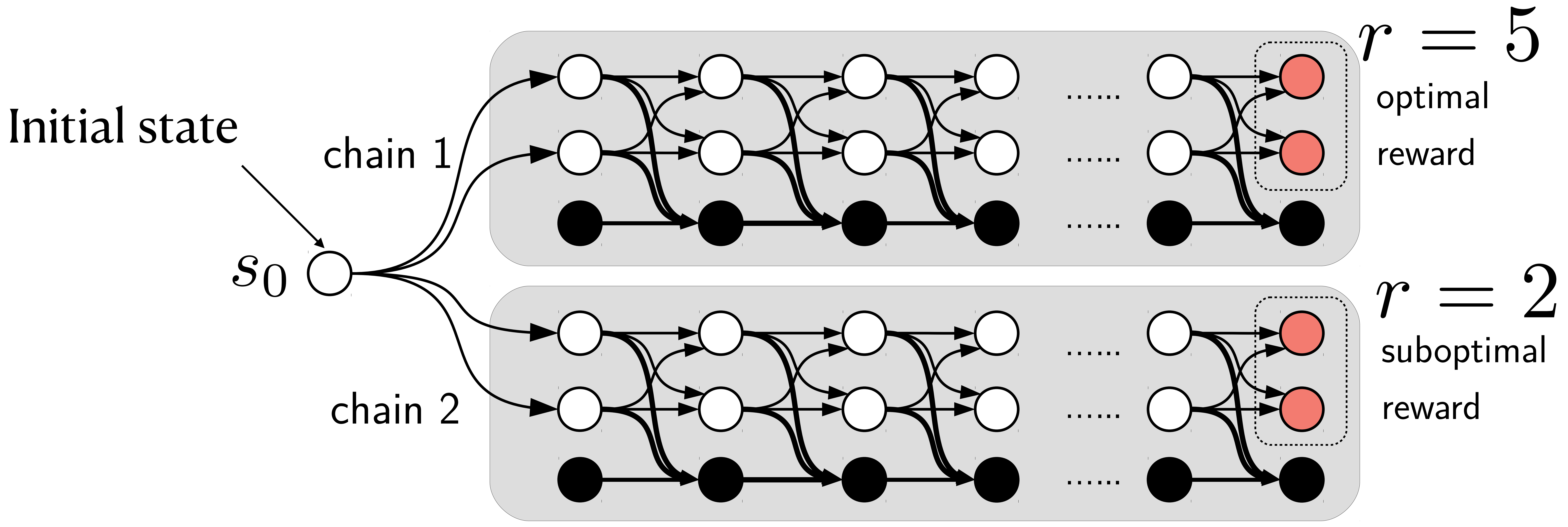
Bidirectional Combination Lock



Bidirectional Combination Lock

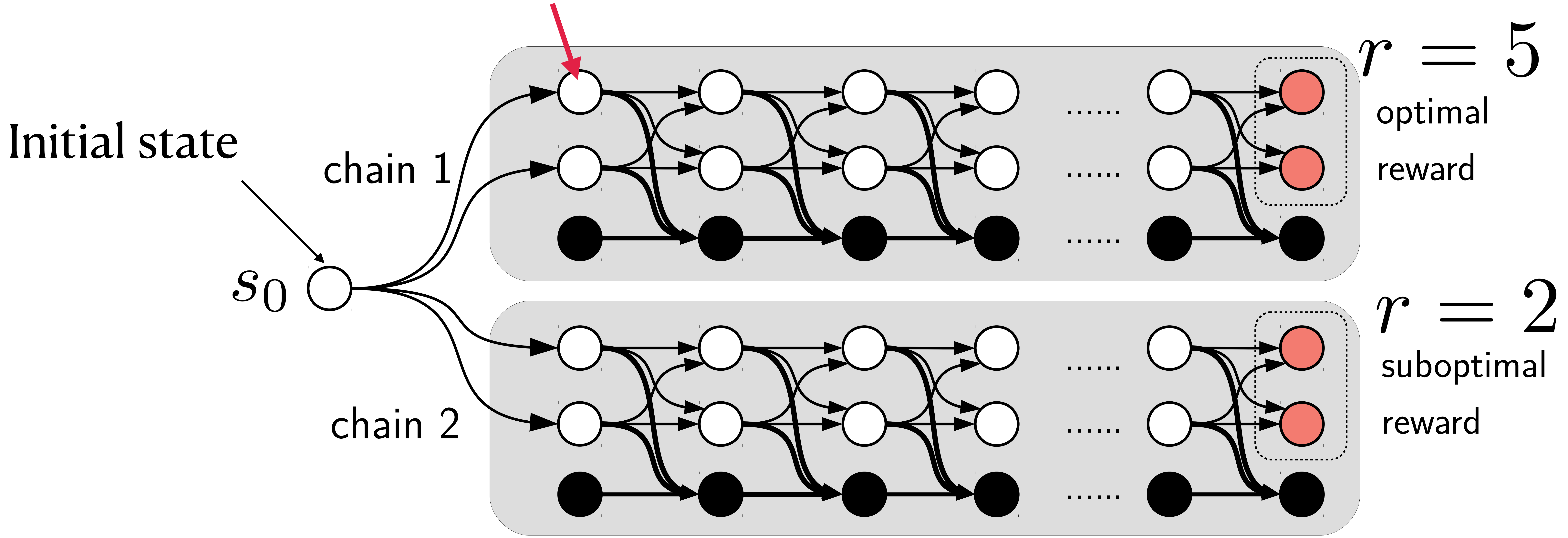


Bidirectional Combination Lock



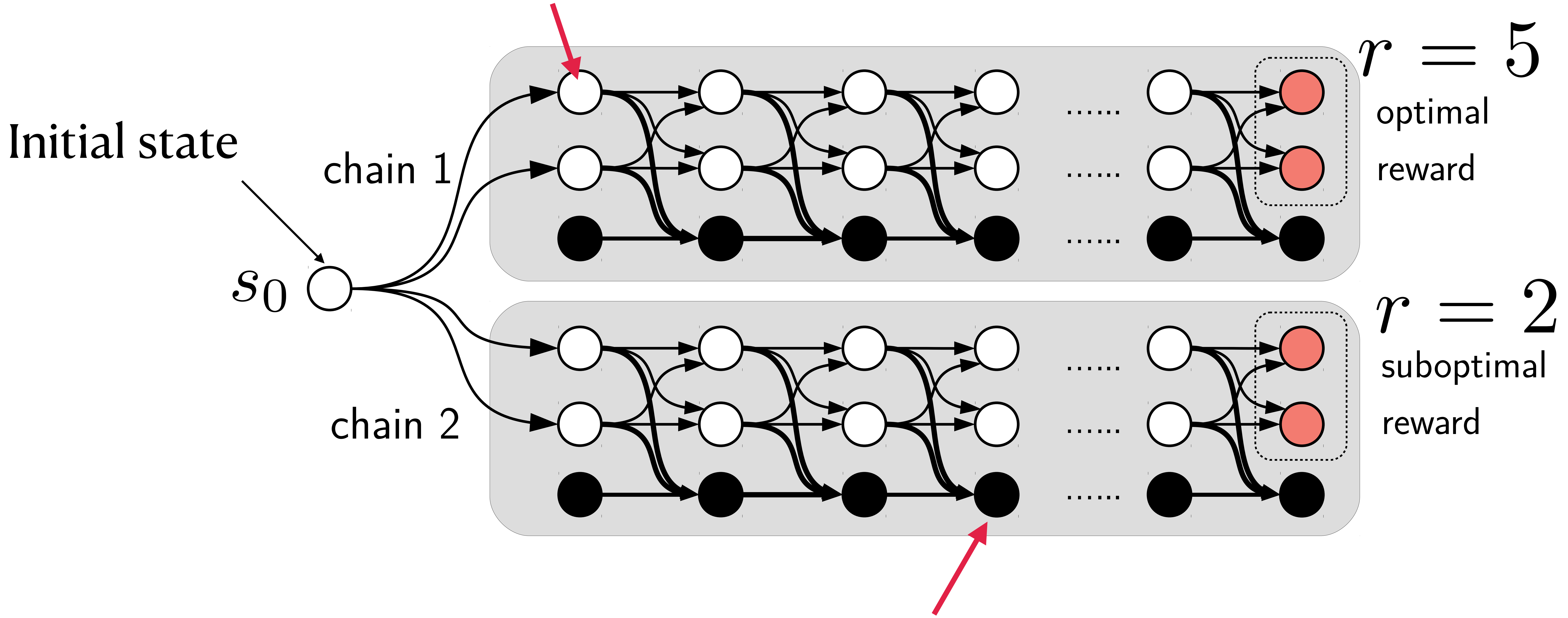
Bidirectional Combination Lock

"survived" state (white): 9 out of 10 actions go to bad state (black)



Bidirectional Combination Lock

"survived" state (white): 9 out of 10 actions go to bad state (black)

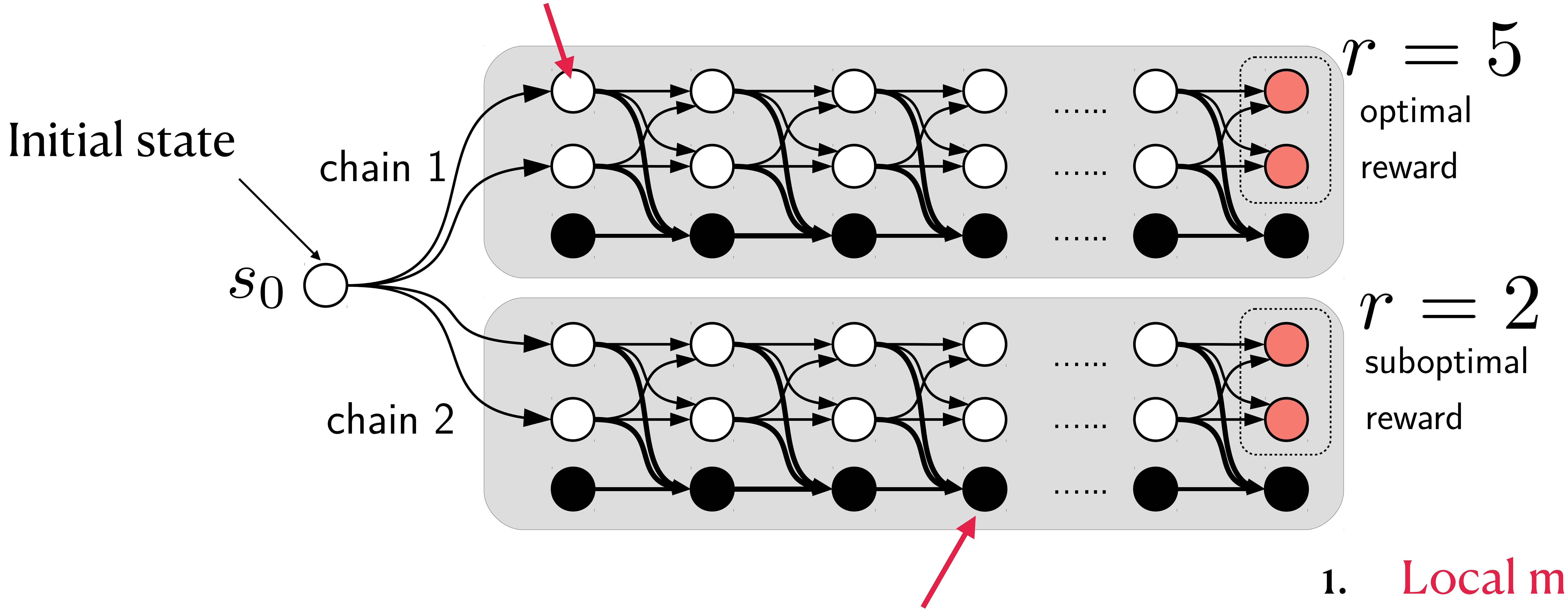


Bad state (cannot recover) has Anti-shaped reward:

$$r = 1/H$$

Bidirectional Combination Lock

“survived” state (white): 9 out of 10 actions go to bad state (black)



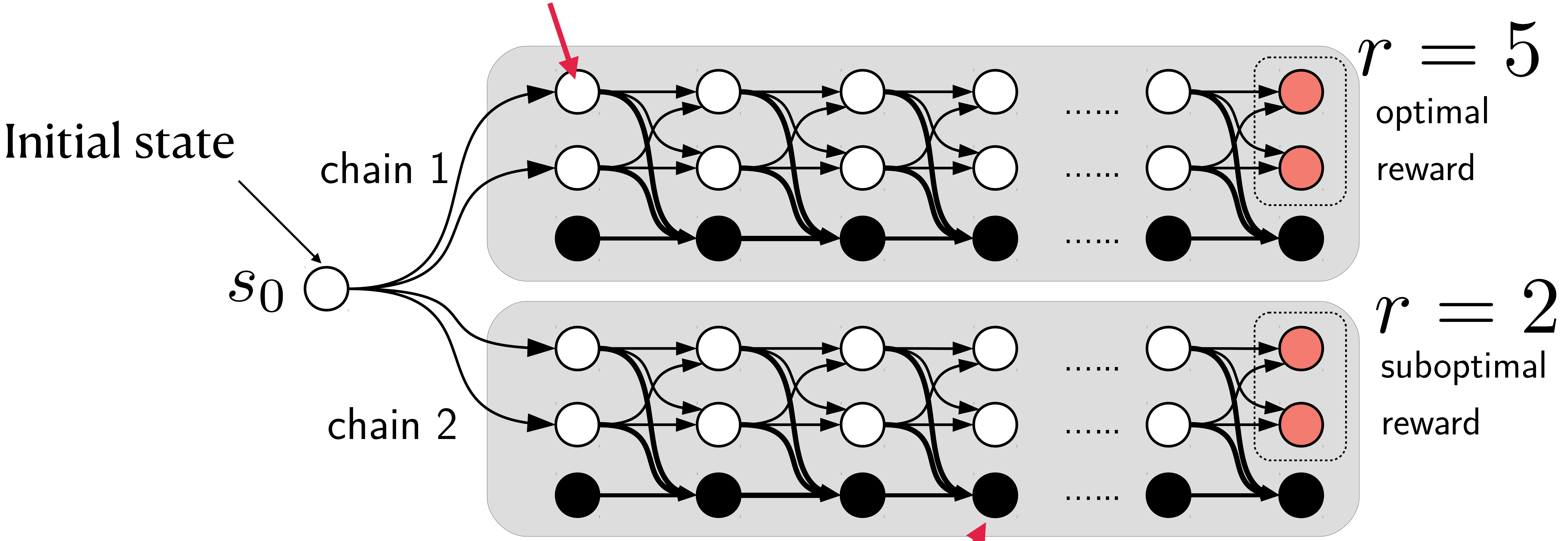
Bad state (cannot recover) has Anti-shaped reward:

$$r = 1/H$$

1. Local minima
2. Forgetting (policy becomes deterministic

Bidirectional Combination Lock

“survived” state (white): 9 out of 10 actions go to bad state (black)



$$\left(\frac{1}{10}\right)^H$$

Bad state (cannot recover) has Anti-shaped reward:

$$r = 1/H$$

1. Local minima
2. Forgetting (policy becomes deterministic

Experiments on Bi-directional Comb Lock

Success Rate (visit the better chain):

Algorithm	Horizon			
	2	5	10	15
PPO	1.0	0.0	0.0	0.0
PPO+RND	0.75	0.40	0.50	0.55
PC-PG	?	?	?	?

PPO+RND: Random Network Distillation [Burda et.al, 19]

Experiments on Bi-directional Comb Lock

Success Rate (visit the better chain):

Algorithm	Horizon			
	2	5	10	15
PPO	1.0	0.0	0.0	0.0
PPO+RND	0.75	0.40	0.50	0.55
PC-PG	?	?	?	?

PPO+RND: Random Network Distillation [Burda et.al, 19]

Experiments on Bi-directional Comb Lock

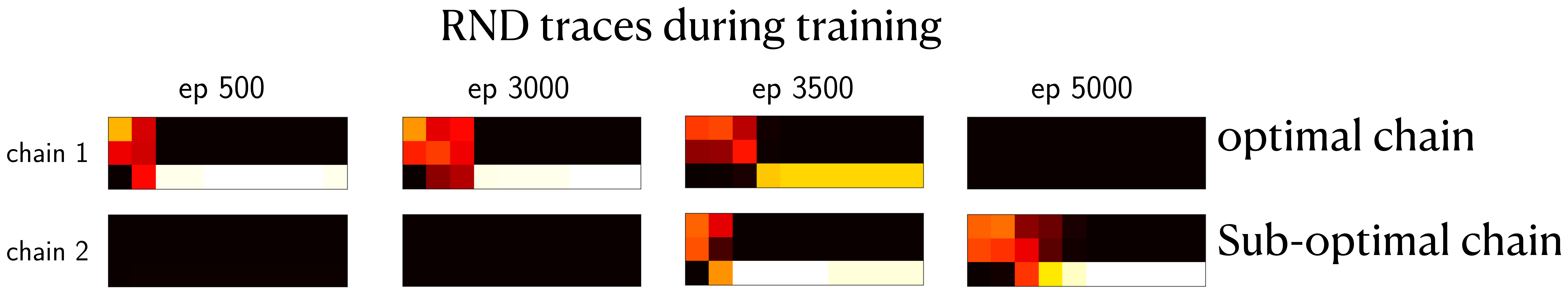
Success Rate (visit the better chain):

Algorithm	Horizon			
	2	5	10	15
PPO	1.0	0.0	0.0	0.0
PPO+RND	0.75	0.40	0.50	0.55
PC-PG	?	?	?	?

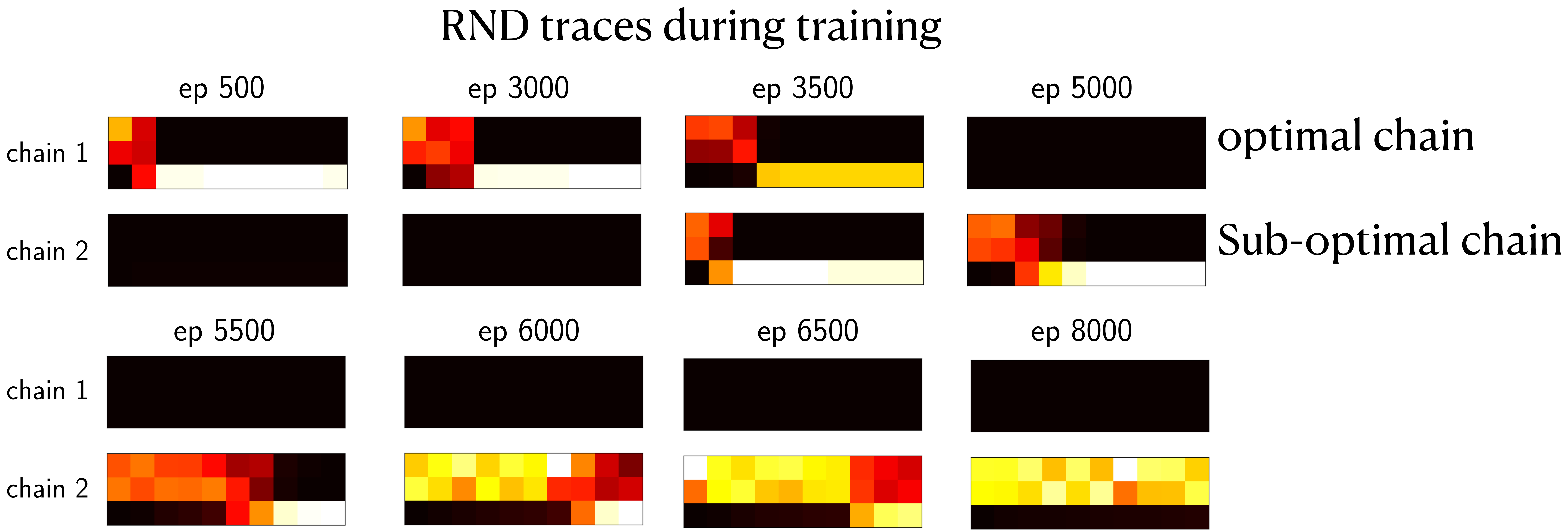
Forgets to visit the other chain !

PPO+RND: Random Network Distillation [Burda et.al, 19]

Experiments on Bi-directional Comb Lock

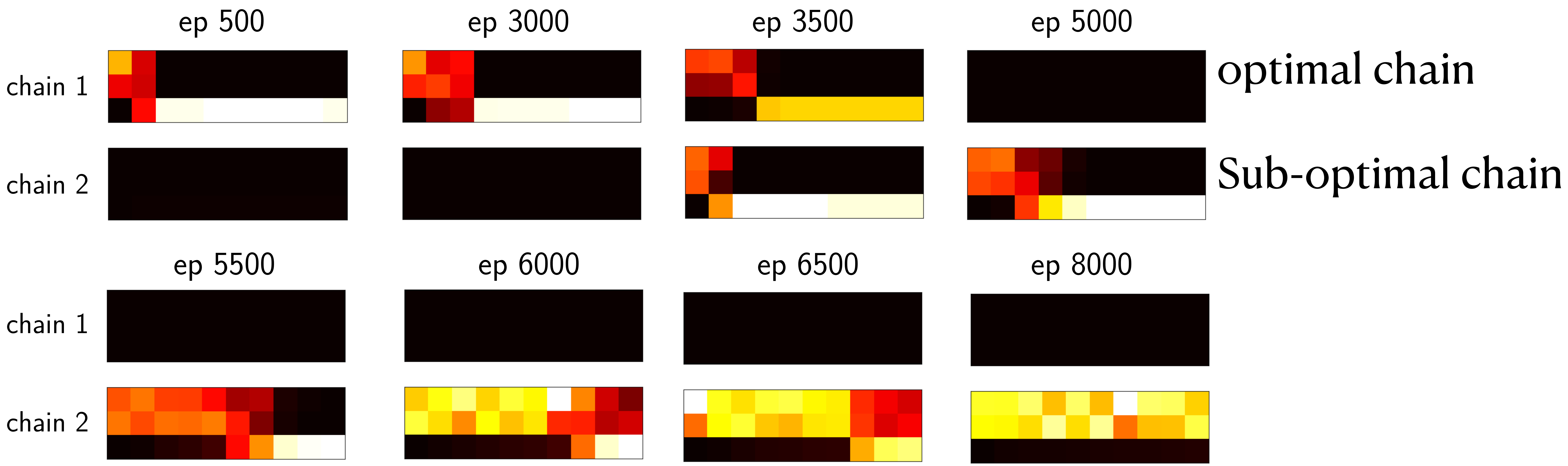


Experiments on Bi-directional Comb Lock



Experiments on Bi-directional Comb Lock

RND traces during training



Policy quickly becomes too deterministic and
forgets to explore the other (better!) chain

Summary of PG methods' common issues

1. Lack ability to explore

1. Catastrophic forgetting (even w/ reward bonus)

Summary of PG methods' common issues

1. Lack ability to explore

1. Catastrophic forgetting (even w/ reward bonus)

Next:

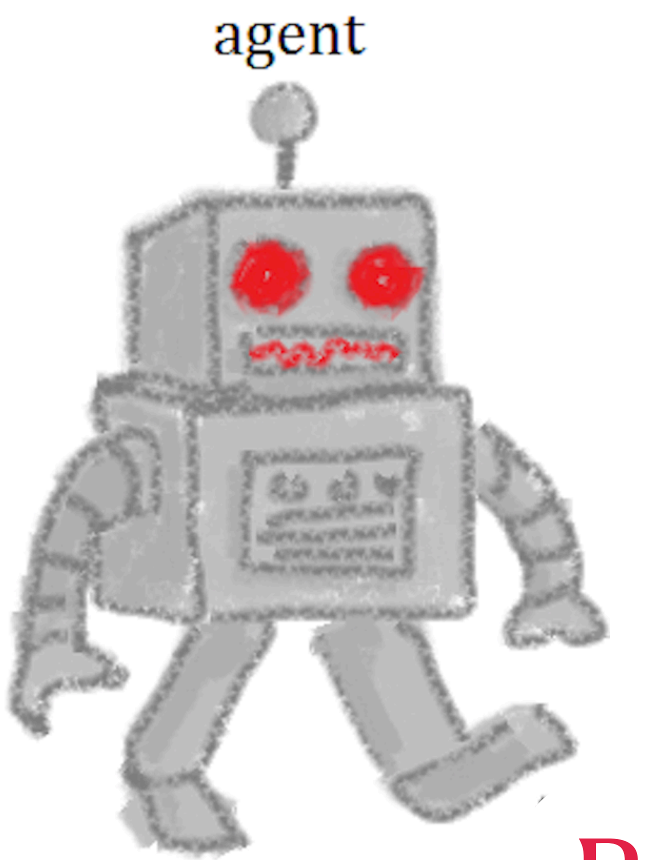
Our Solution: Policy Cover Policy Gradient (PC-PG)

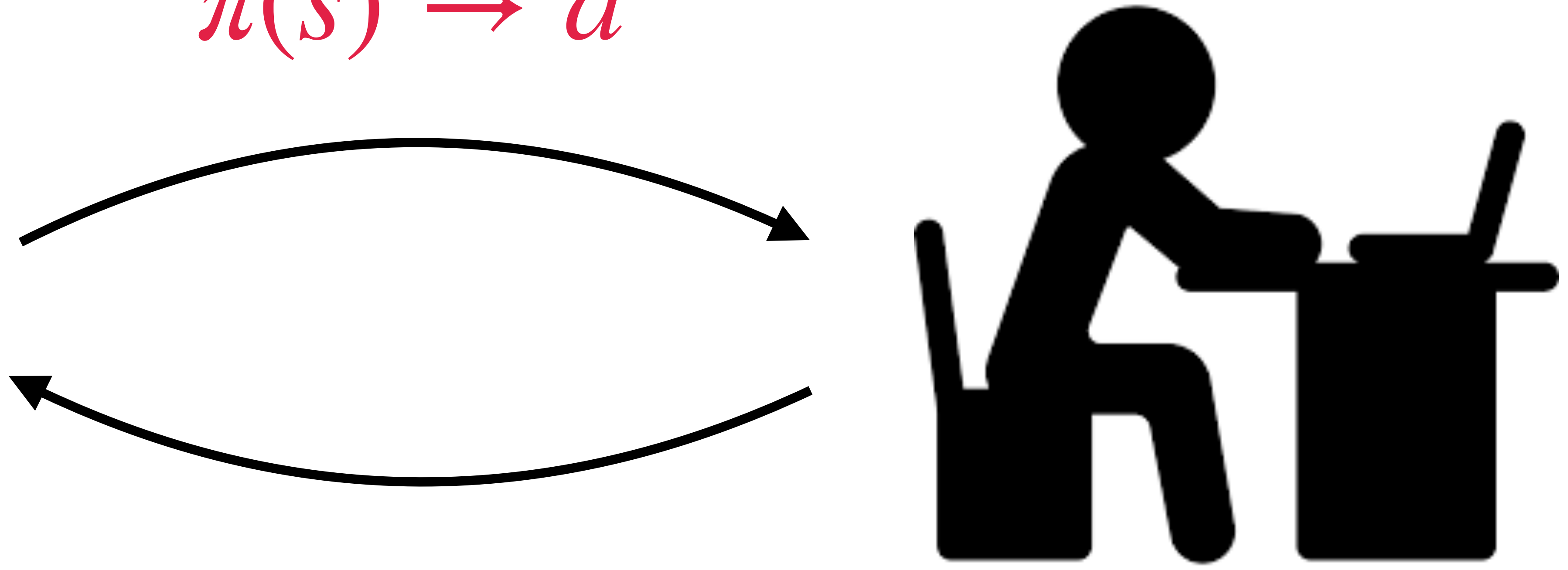
Policy Ensemble + Reward Bonus

Notations

Policy: state to action

$$\pi(s) \rightarrow a$$





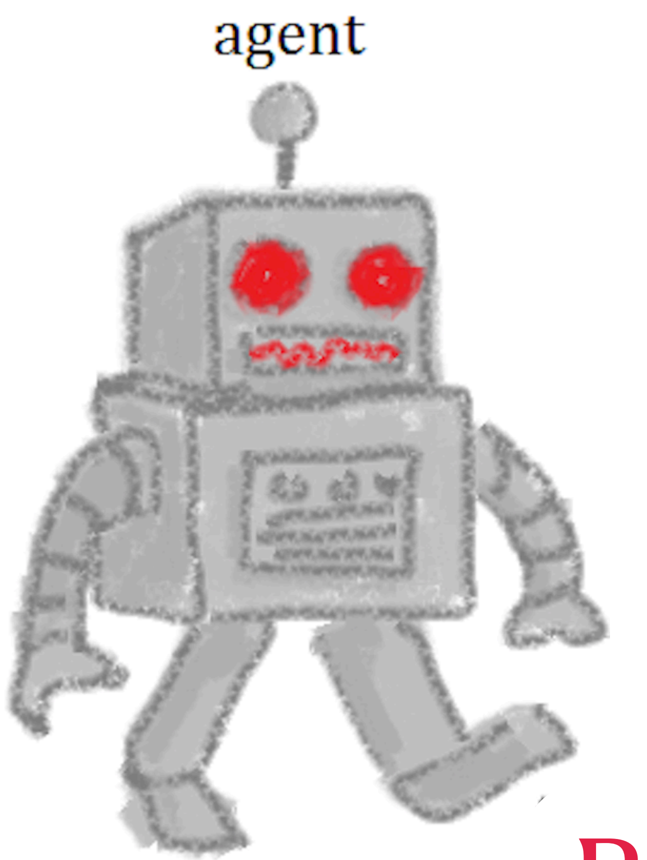
Reward & Next State

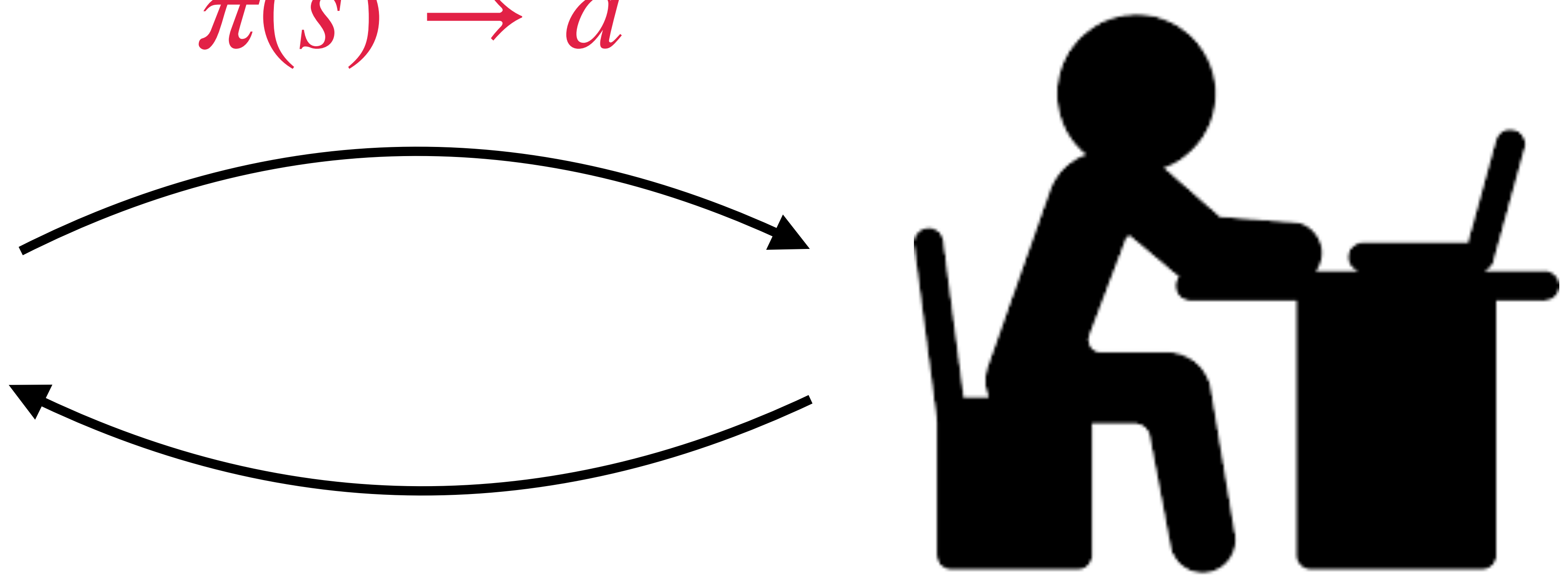
$$r(s, a), s' \sim P(\cdot \mid s, a)$$

Notations

Policy: state to action

$$\pi(s) \rightarrow a$$





Reward & Next State

$$r(s,a), s' \sim P(\cdot \mid s,a)$$

Value and Q function

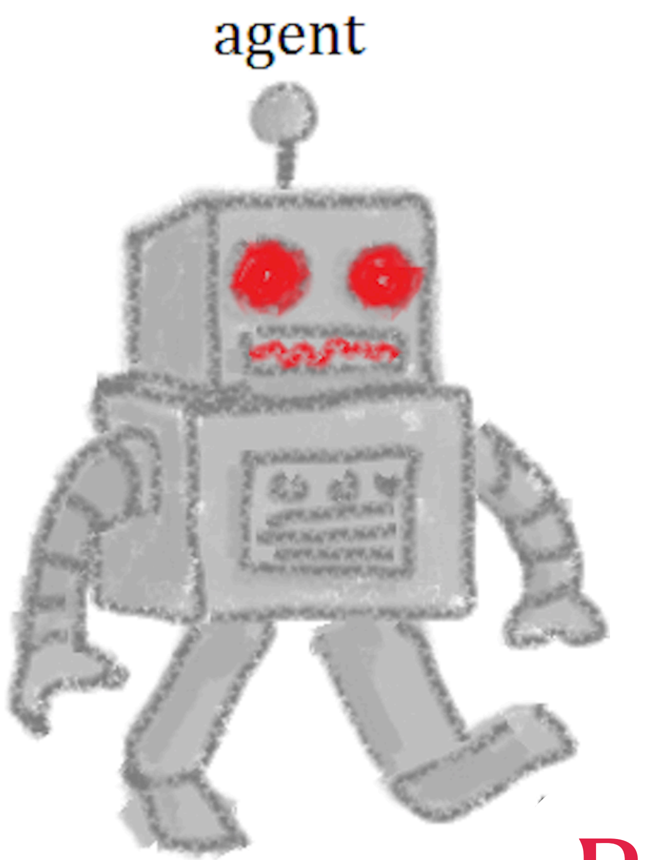
$$V^\pi(s) = \mathbb{E}\left[r(s_0,a_0) + \gamma r(s_1,a_1) + \ldots \mid s_0 = s, a_h \sim \pi(s_h)\right]$$

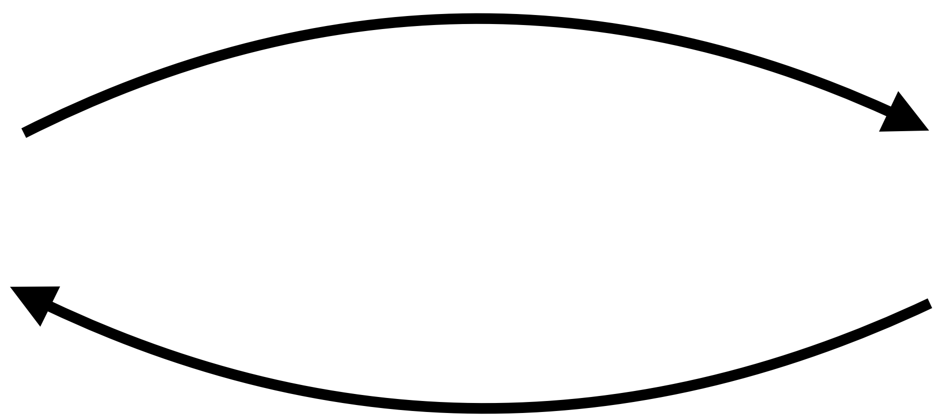
$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} V^\pi(s')$$

Notations

Policy: state to action

$\pi(s) \rightarrow a$





Reward & Next State

$r(s, a), s' \sim P(\cdot \mid s, a)$

Value and Q function

$$V^\pi(s) = \mathbb{E}\left[r(s_0, a_0) + \gamma r(s_1, a_1) + \ldots \mid s_0 = s, a_h \sim \pi(s_h)\right]$$

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot \mid s, a)} V^\pi(s')$$

Policy's state-action distribution

$$d^\pi(s, a) = (1 - \gamma) \sum_{h=0}^{\infty} \gamma^h \mathbb{P}^\pi\left((s_h, a_h) = (s, a)\right)$$

PC-PG: Policy Cover - Policy Gradient

1. **Form Cover:**

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

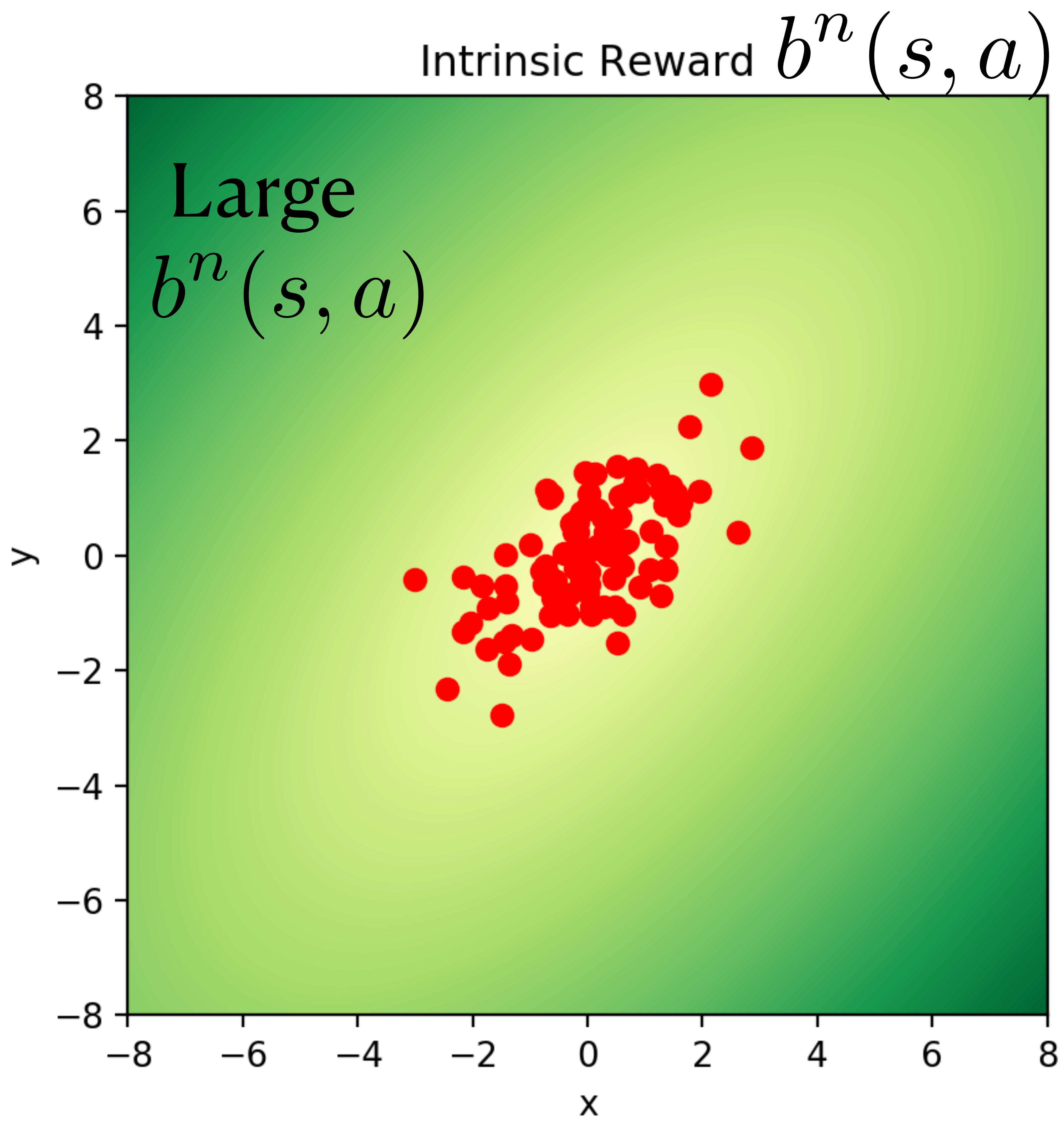
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



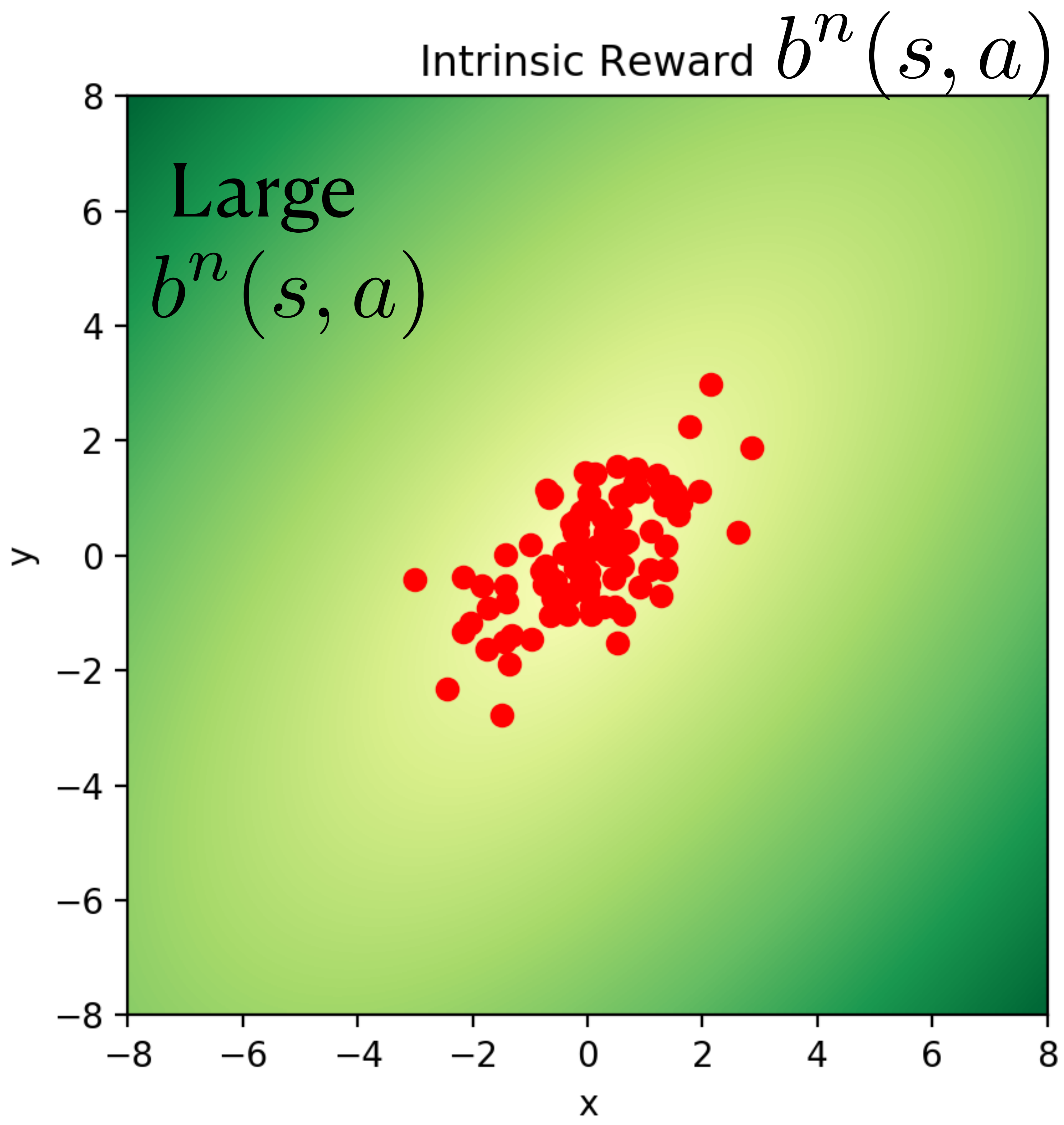
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



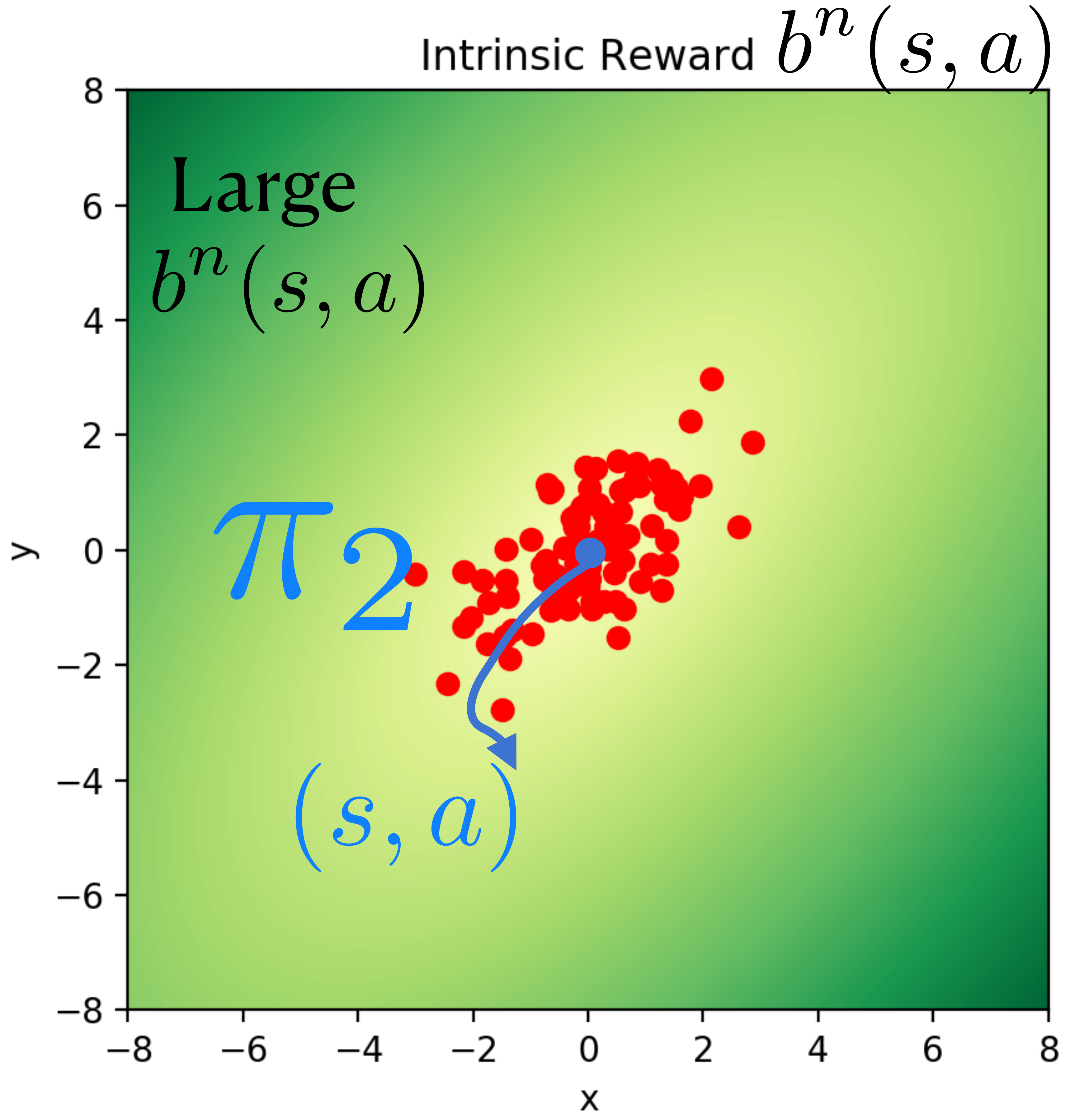
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



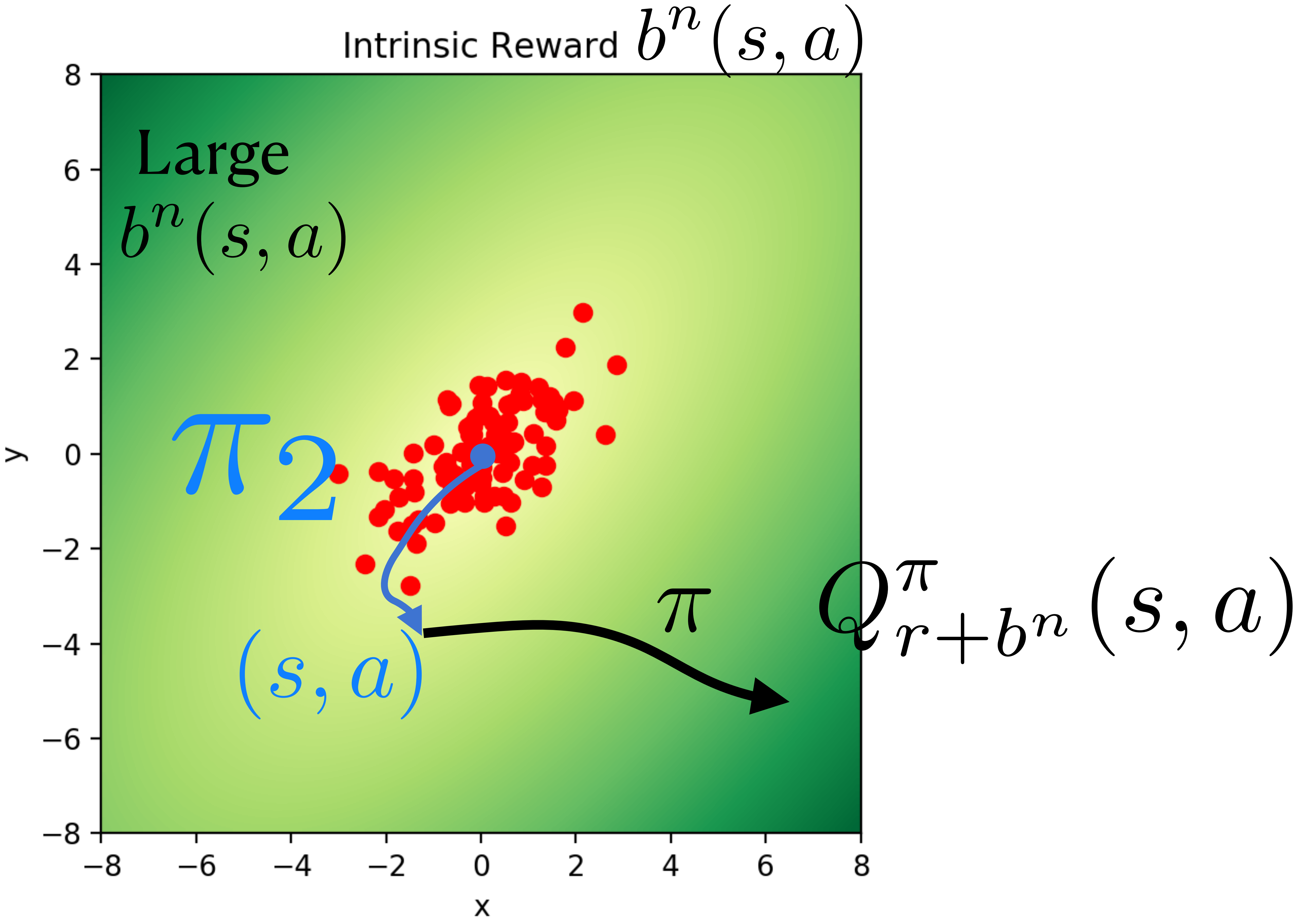
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



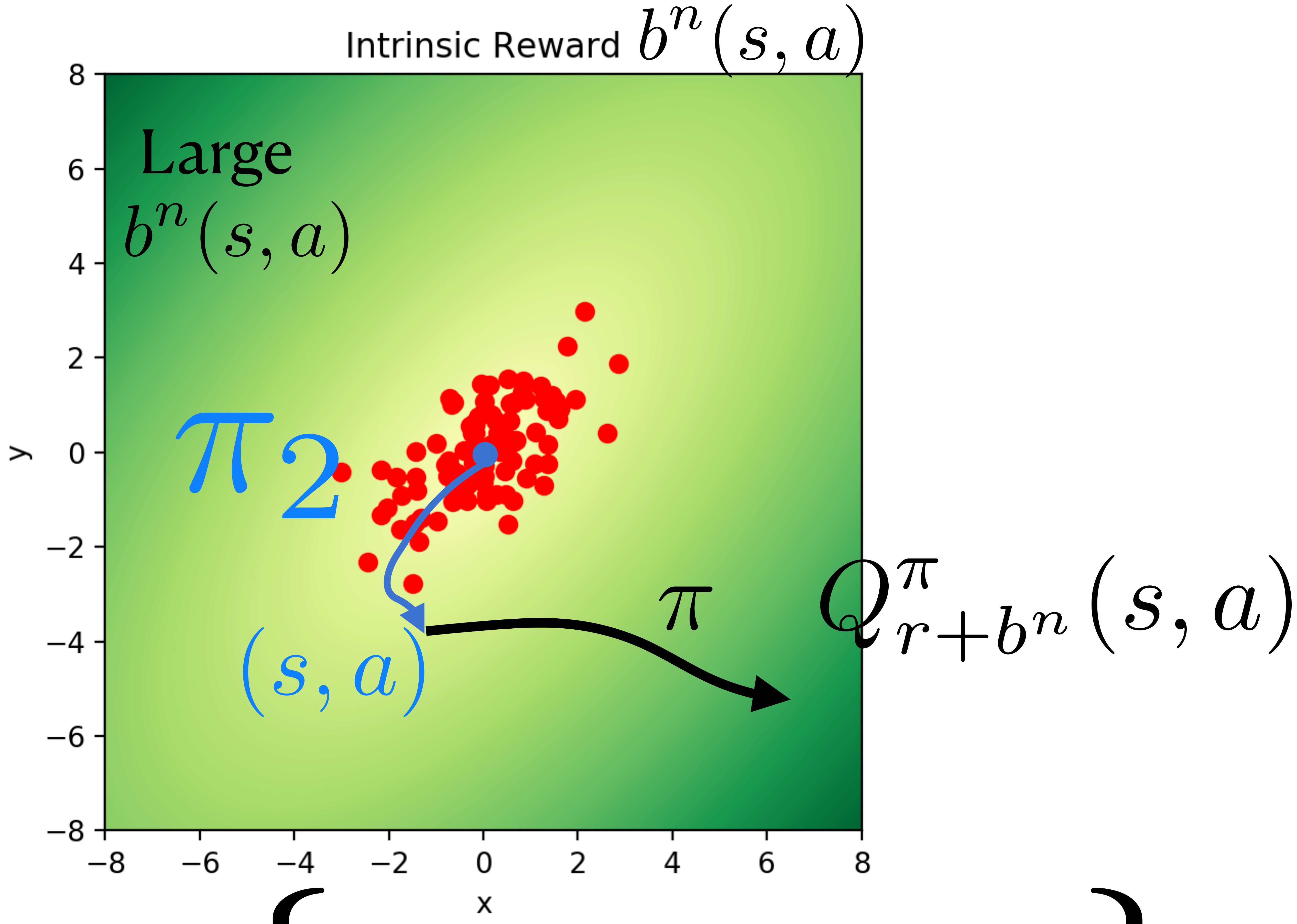
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \dots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



$$\left\{ (s_i, a_i), Q^{\pi}_{r+b^n}(s_i, a_i) \right\}$$

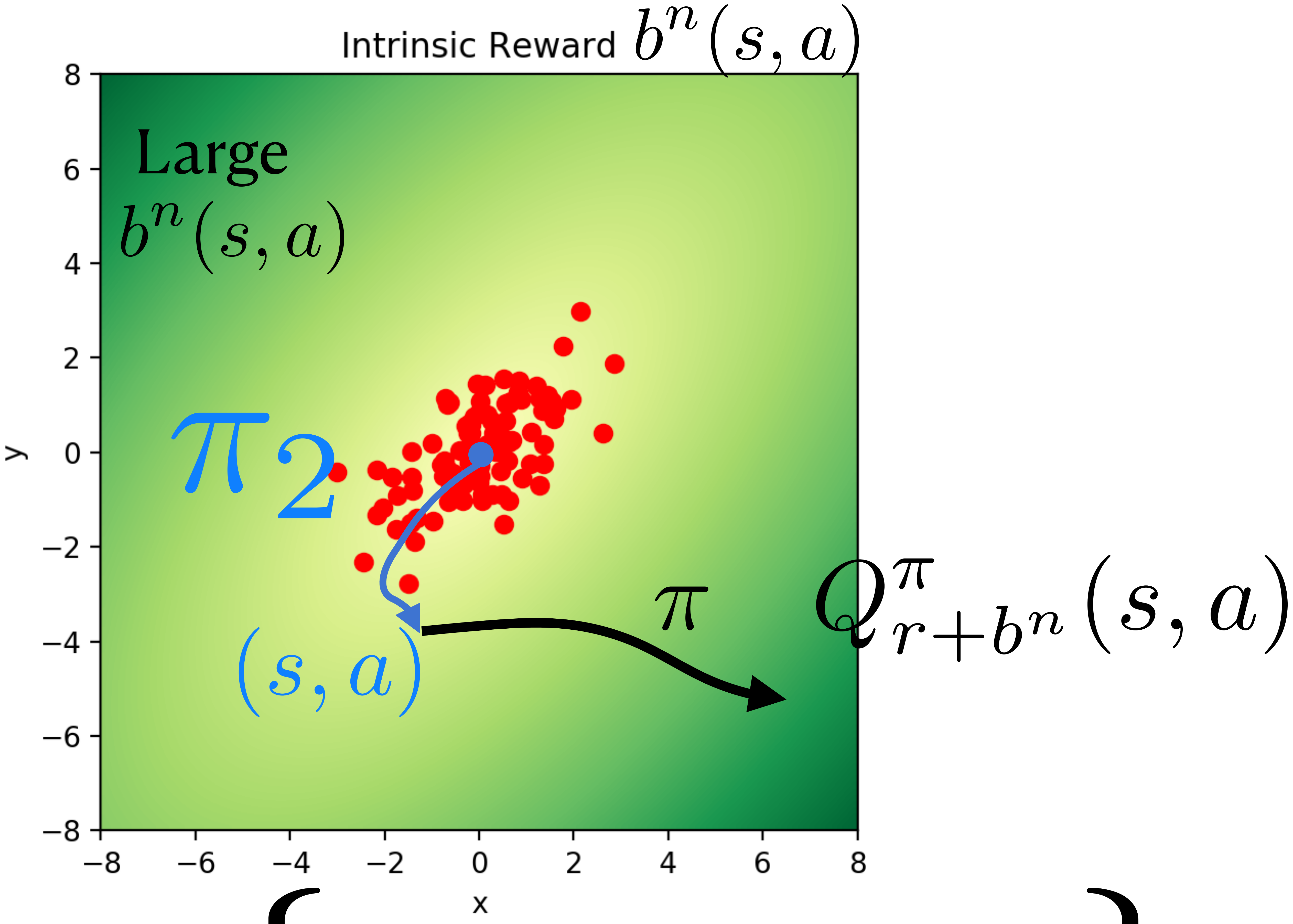
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



$$\left\{(s_i, a_i), Q^\pi_{r+b^n}(s_i, a_i)\right\}$$

$$\hat{f} = \max_{f\in\mathcal{F}} \sum_i \left(f(s_i, a_i) - Q^\pi_{r+b^n}(s_i, a_i)\right)^2$$

3. On-Policy Critic Fit (least square)

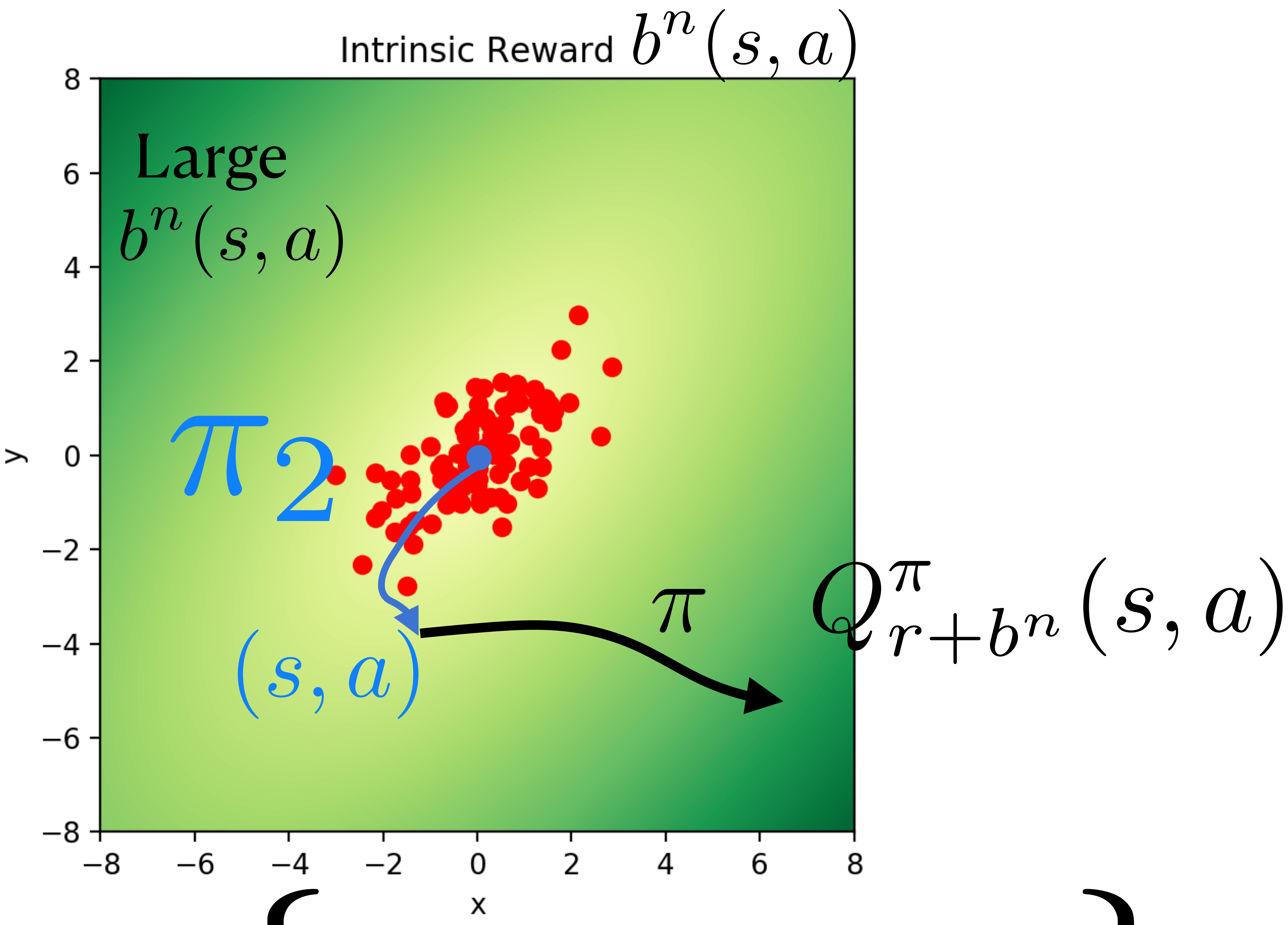
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



$$\pi(s,a) \Leftarrow \pi(s,a)\exp(\eta \hat{f}(s,a))$$

4. Actor NPG update (mirror descent)

$$\hat{f} = \max_{f \in \mathcal{F}} \sum_i \left(f(s_i,a_i) - Q^\pi_{r+b^n}(s_i,a_i)\right)^2$$

3. On-Policy Critic Fit (least square)

$$\left\{(s_i,a_i), Q^\pi_{r+b^n}(s_i,a_i)\right\}$$

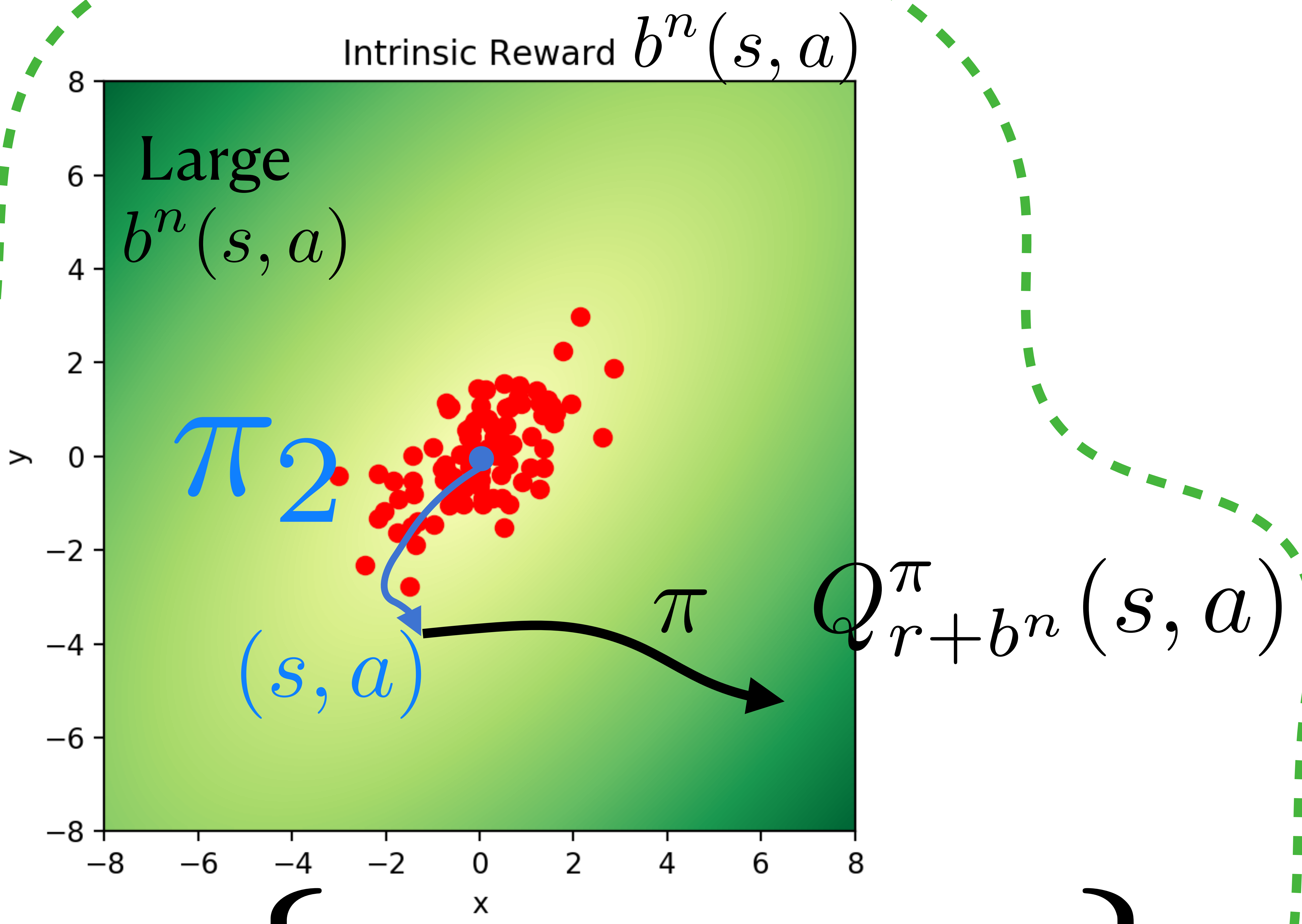
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



$$\left\{ (s_i, a_i),\, Q^{\pi}_{r+b^n}(s_i, a_i) \right\}$$

$$\hat{f} = \max_{f \in \mathcal{F}} \sum_i \left(f(s_i, a_i) - Q^{\pi}_{r+b^n}(s_i, a_i) \right)^2$$

3. On-Policy Critic Fit (least square)

$$\pi(s,a) \Leftarrow \pi(s,a) \exp(\eta \hat{f}(s,a))$$

4. Actor NPG update (mirror descent)

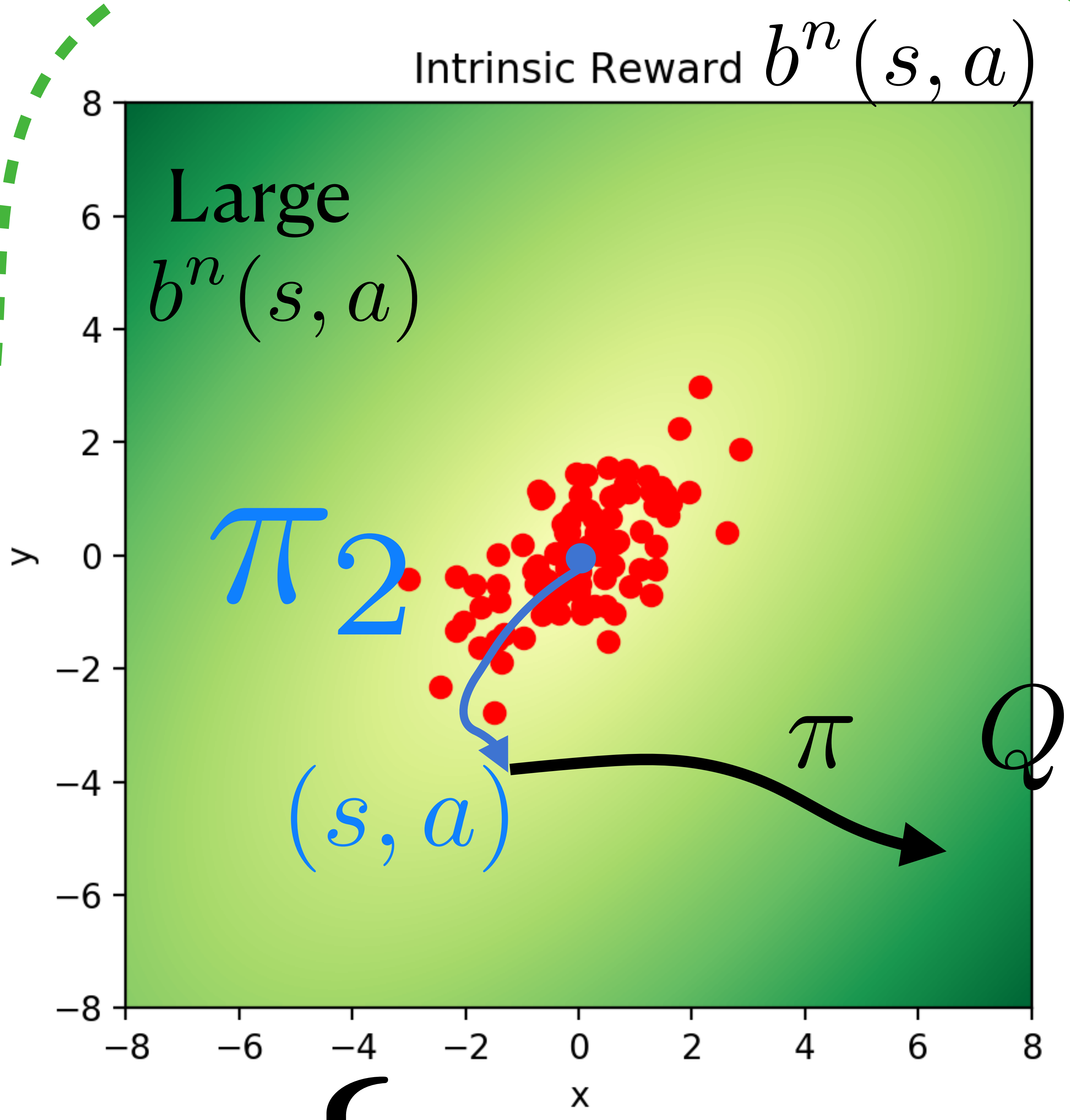
PC-PG: Policy Cover - Policy Gradient

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\rho_n := \sum_{i=1}^{n} d^{\pi^i}/n$$

2. Bonus



$$Q^{\pi}_{r+b^n}(s,a)$$

$$\left\{(s_i, a_i), Q^{\pi}_{r+b^n}(s_i, a_i)\right\}$$

$$\hat{f} = \max_{f \in \mathcal{F}} \sum_i \left(f(s_i,a_i) - Q^{\pi}_{r+b^n}(s_i,a_i)\right)^2$$

3. On-Policy Critic Fit (least square)

$$\pi(s,a) \Leftarrow \pi(s,a)\exp(\eta \hat{f}(s,a))$$

4. Actor NPG update (mirror descent)

5. Append

What objective function PC-PG is trying to optimize?

At episode n: Natural PG is optimizing

$$\mathbb{E}_{s_0, a_0 \sim \rho_n} \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + b^n(s_t, a_t) \right) \right]$$

What objective function PC-PG is trying to optimize?

At episode n: Natural PG is optimizing

$$\mathbb{E}_{s_0, a_0 \sim \rho_n}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + b^n(s_t, a_t)\right)\right]$$

Use the cover and roll-in via policies in the cover

(note we do not start at μ_0)

What objective function PC-PG is trying to optimize?

At episode n: Natural PG is optimizing

$$\mathbb{E}_{s_0,a_0\sim\rho_n}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t)+b^n(s_t,a_t)\right)\right]$$

Use the cover and roll-in via policies in the cover

Bonus based on the cover

(note we do not start at μ_0)

What objective function PC-PG is trying to optimize?

At episode n: Natural PG is optimizing

$$\mathbb{E}_{s_0,a_0\sim\rho_n}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t)+b^n(s_t,a_t)\right)\right]$$

Use the cover and roll-in via policies in the cover

(note we do not start at μ_0)

Bonus based on the cover

No more forgetting!

What objective function PC-PG is trying to optimize?

At episode n: Natural PG is optimizing

$$\mathbb{E}_{s_0,a_0\sim\rho_n}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t)+b^n(s_t,a_t)\right)\right]$$

Use the cover and roll-in via policies in the cover

(note we do not start at μ_0)

No more forgetting!

Bonus based on the cover

No more sparse reward!

PC-PG Specialized to Linear Function Approximation

Use linear function $\theta \cdot \phi(s, a)$ to approximate Q^π

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\Sigma_{\pi_i} = \mathbb{E}_{s,a \sim d_{\pi_i}} \phi(s, a) \phi(s, a)^\top$$

$$\Sigma_n = \sum_{i=1}^{n} \Sigma_{\pi_i} + \lambda I$$

PC-PG Specialized to Linear Function Approximation

Use linear function $\theta \cdot \phi(s, a)$ to approximate Q^π

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\Sigma_{\pi_i} = \mathbb{E}_{s,a\sim d_{\pi_i}} \phi(s,a)\phi(s,a)^\top$$

$$\Sigma_n = \sum_{i=1}^{n} \Sigma_{\pi_i} + \lambda I$$

→

2. Bonus

$$b^n(s,a) = \mathbf{1}\left\{\phi(s,a)^\top \Sigma_n^{-1} \phi(s,a) \geq \beta\right\} / (1-\gamma)$$

PC-PG Specialized to Linear Function Approximation

Use linear function $\theta \cdot \phi(s, a)$ to approximate Q^π

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\Sigma_{\pi_i} = \mathbb{E}_{s,a\sim d_{\pi_i}} \phi(s,a)\phi(s,a)^\top$$

$$\Sigma_n = \sum_{i=1}^{n} \Sigma_{\pi_i} + \lambda I$$

2. Bonus

$$b^n(s,a) = \mathbf{1}\left\{\phi(s,a)^\top \Sigma_n^{-1} \phi(s,a) \geq \beta\right\} / (1-\gamma)$$

Rewarding (s, a) whose feature $\phi(s, a)$ aligns with small eigenvectors

PC-PG Specialized to Linear Function Approximation

Use linear function $\theta \cdot \phi(s,a)$ to approximate Q^π

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\Sigma_{\pi_i} = \mathbb{E}_{s,a\sim d_{\pi_i}} \phi(s,a)\phi(s,a)^\top$$

$$\Sigma_n = \sum_{i=1}^{n} \Sigma_{\pi_i} + \lambda I$$

2. Bonus

$$b^n(s,a) = \mathbf{1}\left\{\phi(s,a)^\top \Sigma_n^{-1} \phi(s,a) \geq \beta\right\}/(1-\gamma)$$

Rewarding (s,a) whose feature $\phi(s,a)$ aligns with small eigenvectors

3. Natural PG:

$$Q^\pi_{r+b^n}(s,a) \approx \theta \cdot \phi(s,a) + b^n(s,a)$$

$$\pi \Leftarrow \pi \exp\left(\eta\left(b^n + \theta \cdot \phi\right)\right)$$

PC-PG Specialized to Linear Function Approximation

Use linear function $\theta \cdot \phi(s,a)$ to approximate Q^π

1. Form Cover:

$$\{\pi_1, \pi_2, \ldots, \pi_n\}$$

$$\Sigma_{\pi_i} = \mathbb{E}_{s,a \sim d_{\pi_i}} \phi(s,a)\phi(s,a)^\top$$

$$\Sigma_n = \sum_{i=1}^{n} \Sigma_{\pi_i} + \lambda I$$

2. Bonus

$$b^n(s,a) = \mathbf{1}\left\{\phi(s,a)^\top \Sigma_n^{-1} \phi(s,a) \geq \beta\right\} / (1-\gamma)$$

Rewarding (s,a) whose feature $\phi(s,a)$ aligns with small eigenvectors

3. Natural PG:

$$Q^\pi_{r+b^n}(s,a) \approx \theta \cdot \phi(s,a) + b^n(s,a)$$

$$\pi \Leftarrow \pi \exp\left(\eta\left(b^n + \theta \cdot \phi\right)\right)$$

PC-PG Specialized to Tabular MDPs

One-hot vector: $\phi(s, a) \in \mathbb{R}^{SA}$

$$\rho_n = \sum_{i=1}^{n} d^{\pi_i}/n$$

$$\Sigma_n = \mathrm{diag}\left(\ldots, \rho_n(s, a), \ldots, \right)$$

PC-PG Specialized to Tabular MDPs

One-hot vector: $\phi(s, a) \in \mathbb{R}^{SA}$

$$\rho_n = \sum_{i=1}^{n} d^{\pi_i}/n$$

$$\Sigma_n = \operatorname{diag}\left(\ldots, \rho_n(s, a), \ldots, \right)$$

Rewarding state that has low probability of being covered

$$\phi(s, a)^\top \Sigma_n^{-1} \phi(s, a) \Leftarrow \frac{1}{\rho_n(s, a)}$$

Well-Specified Setting: Linear MDPs (and Tabular MDPs)

Reward and transition in RKHS:

[RKHS version of the Linear mdp model from Jin et al, 19]

$$r(s,a) = \theta \cdot \phi(s,a), P(\cdot \,|\, s,a) = \mu\phi(s,a)$$

Well-Specified Setting: Linear MDPs (and Tabular MDPs)

Reward and transition in RKHS:

[RKHS version of the Linear mdp model from Jin et al, 19]

$$r(s,a) = \theta \cdot \phi(s,a), P(\cdot \mid s,a) = \mu\phi(s,a)$$

A bellman backup of any $f(s)$ will be linear in $\phi(s,a)$:

$$r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} f(s') = w^\top \phi(s,a)$$

Well-Specified Setting: Linear MDPs (and Tabular MDPs)

Reward and transition in RKHS:

[RKHS version of the Linear mdp model from Jin et al, 19]

$$r(s,a) = \theta \cdot \phi(s,a), P(\cdot \mid s,a) = \mu\phi(s,a)$$

A bellman backup of any $f(s)$ will be linear in $\phi(s,a)$:

$$r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} f(s') = w^\top \phi(s,a)$$

$$V^{\hat{\pi}} \geq V^\star - \epsilon$$

with # of samples

$$\text{poly}(d, \log(A), 1/(1-\gamma), 1/\epsilon)$$

Well-Specified Setting: Linear MDPs (and Tabular MDPs)

Reward and transition in RKHS:

[RKHS version of the Linear mdp model from Jin et al, 19]

$$r(s,a) = \theta \cdot \phi(s,a), P(\cdot \,|\, s,a) = \mu\phi(s,a)$$

A bellman backup of any $f(s)$ will be linear in $\phi(s,a)$:

$$r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} f(s') = w^\top \phi(s,a)$$

$$V^{\hat{\pi}} \geq V^\star - \epsilon$$

with # of samples

$$\text{poly}(d, \log(A), 1/(1-\gamma), 1/\epsilon)$$

Dim of feature

(extendable to RKHS w/ Information Gain)

Comparison to vanilla Natural PG in Linear MDPs

[Agarwal et al 19]

A wide initial distribution:

$$\kappa = 1/\sigma_{\min}\left(\mathbb{E}_{s,a\sim\mu_0}\phi(s,a)\phi(s,a)^\top\right) < \infty$$

Comparison to vanilla Natural PG in Linear MDPs

[Agarwal et al 19]

A wide initial distribution:

$$\kappa = 1/\sigma_{\min}\left(\mathbb{E}_{s,a\sim\mu_0}\phi(s,a)\phi(s,a)^\top\right) < \infty$$

$$V^{\hat{\pi}} \geq V^\star - \epsilon$$

w/ # of samples $\mathrm{poly}(1/\epsilon, 1/(1-\gamma), \ln(A), \kappa)$

Comparison to vanilla Natural PG in Linear MDPs

[Agarwal et al 19]

A wide initial distribution:

$$\kappa = 1/\sigma_{\min}\left(\mathbb{E}_{s,a\sim\mu_0}\phi(s,a)\phi(s,a)^\top\right) < \infty$$

$$V^{\widehat{\pi}} \geq V^\star - \epsilon$$

w/ # of samples $\text{poly}(1/\epsilon, 1/(1-\gamma), \ln(A), \kappa)$

Condition number could be exponential !!

Comparison to vanilla Natural PG in Linear MDPs

[Agarwal et al 19]

A wide initial distribution:

$$\kappa = 1/\sigma_{\min}\left(\mathbb{E}_{s,a\sim\mu_0}\phi(s,a)\phi(s,a)^\top\right) < \infty$$

$$V^{\hat{\pi}} \geq V^\star - \epsilon$$

w/ # of samples $\text{poly}(1/\epsilon, 1/(1-\gamma), \ln(A), \kappa)$

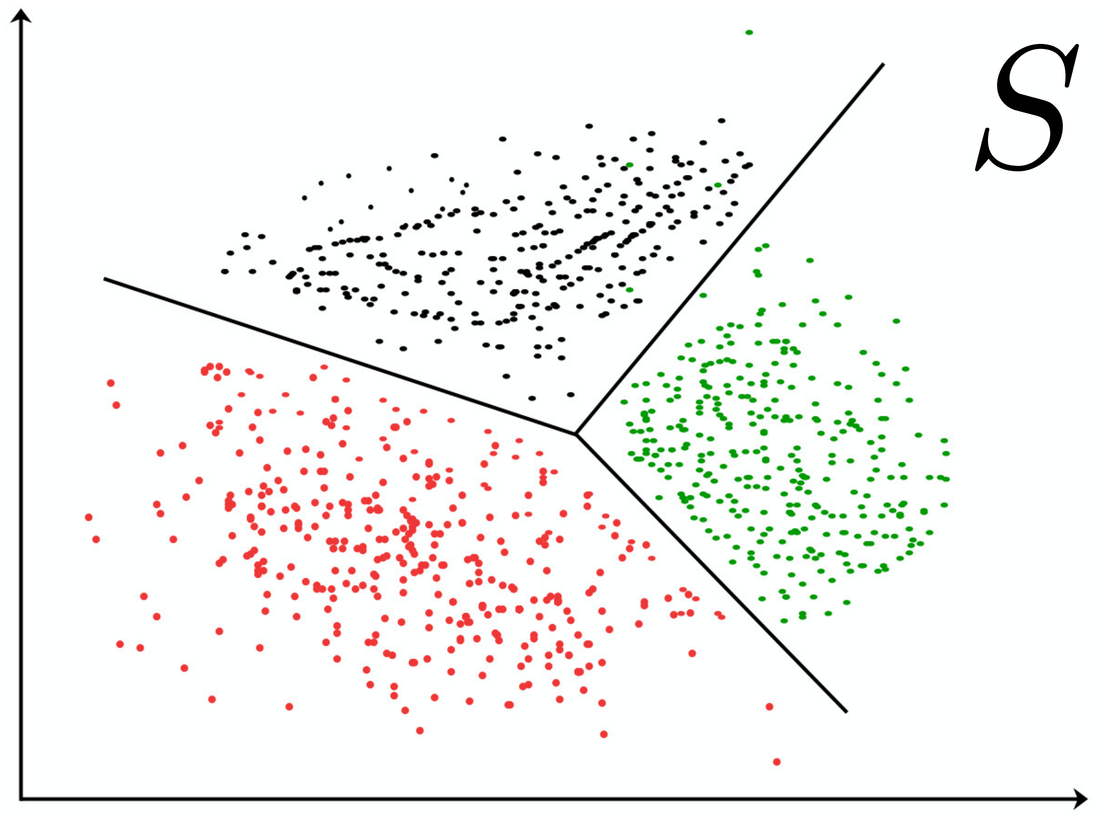
Condition number could be exponential !!

PC-PG eliminates the condition number by actively **exploring** and building **policy cover**

Robustness to Model-Misspecification

Average VS ℓ_∞

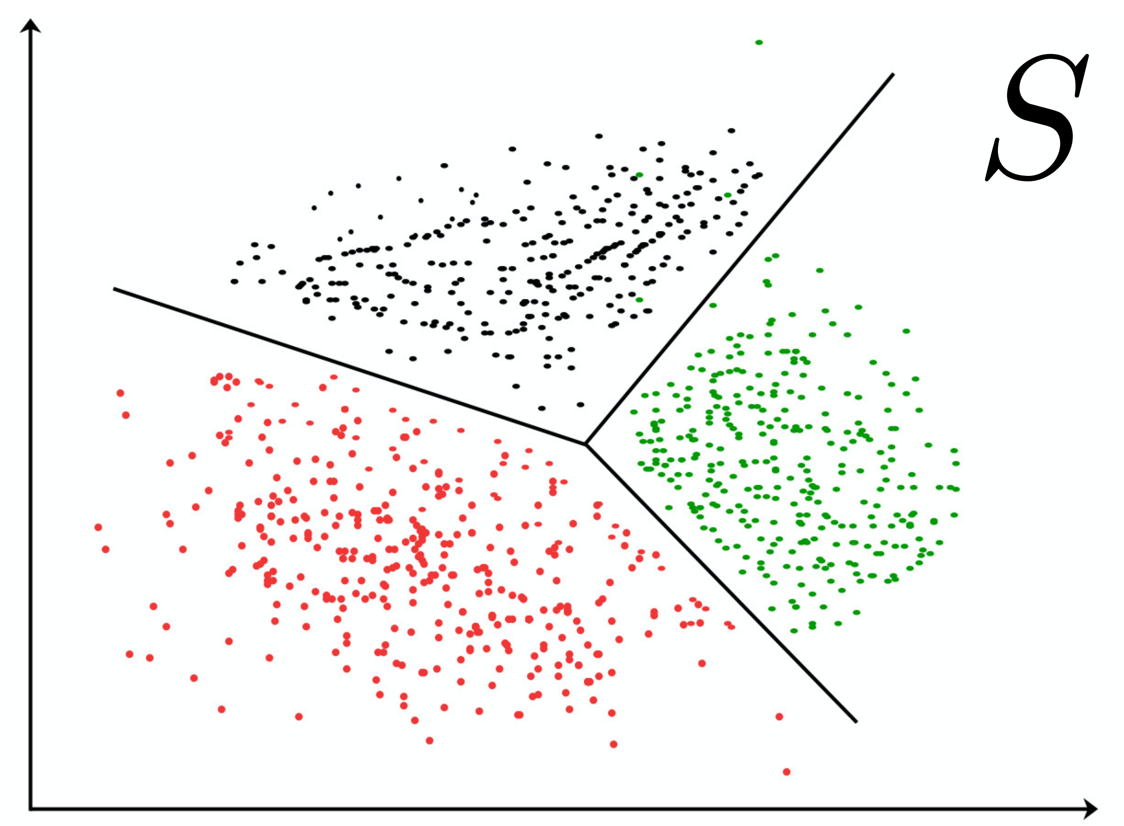
Model-Misspecification from State-Abstraction



$$\phi : S \to Z$$

Group "similar" states (s) into an abstracted state (z)

Model-Misspecification from State-Abstraction



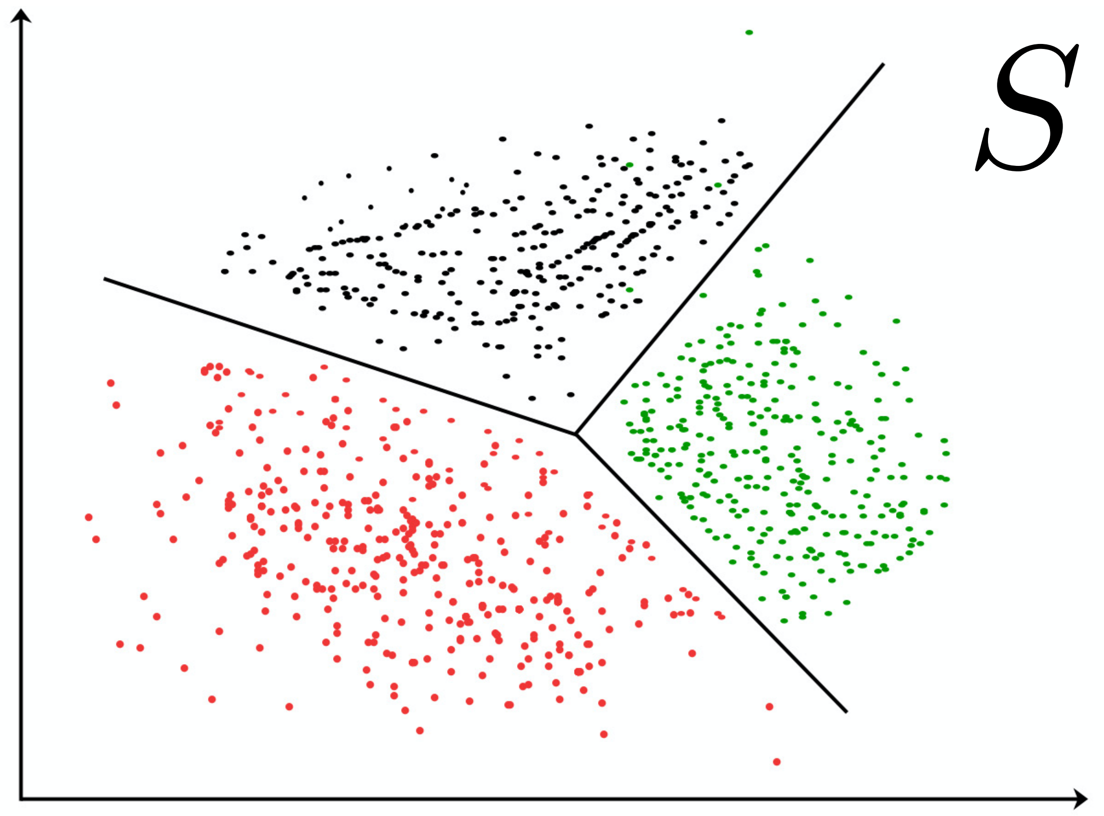
$$\phi : S \rightarrow Z$$

Group "similar" states (s) into an abstracted state (z)

$$|Z| \ll |S|$$

$$\text{poly}(|Z|)$$

Model-Misspecification from State-Abstraction



$$\phi : S \to Z$$

Group "similar" states (s) into an abstracted state (z)

$$|Z| \ll |S|$$

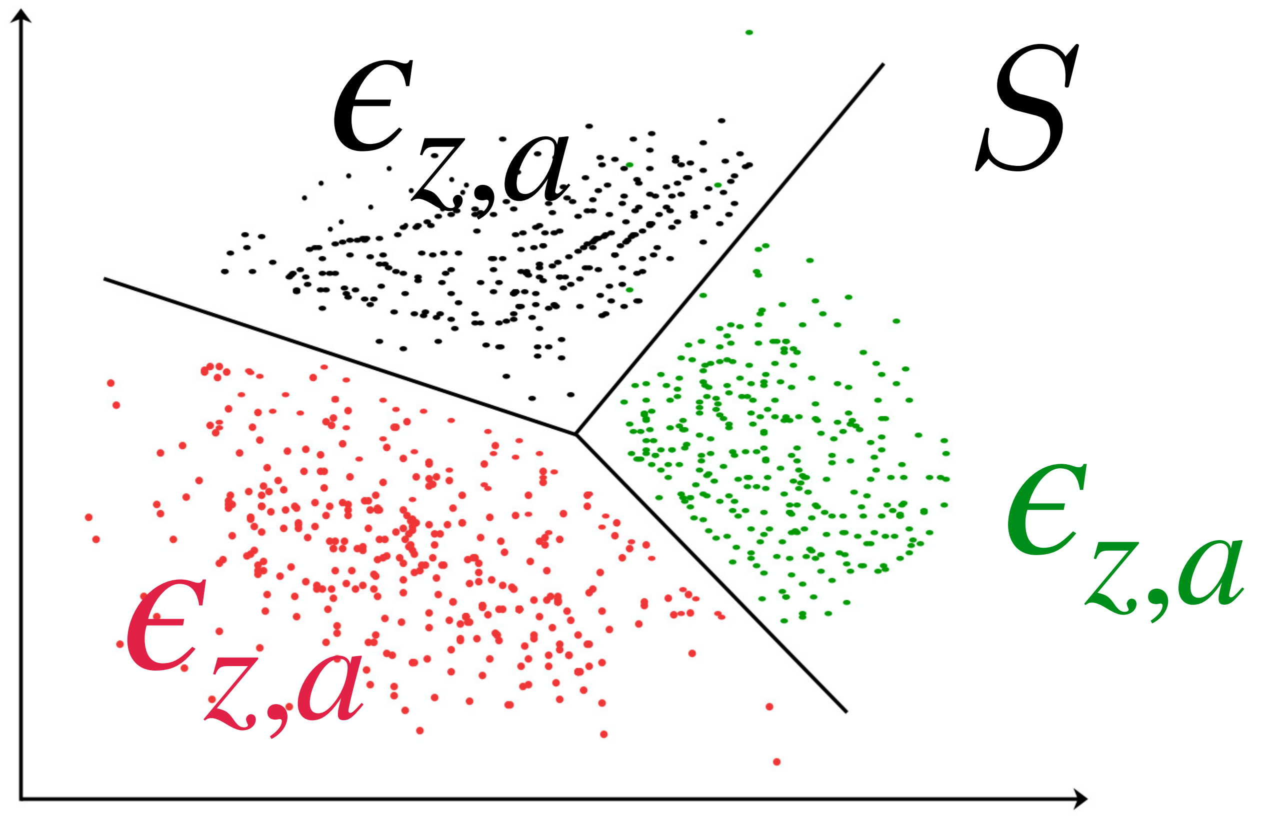
$$\text{poly}(|Z|)$$

Model-misspecification:

$$s, s', \text{ s.t. } \phi(s) = \phi(s')$$

$$\|P(\cdot \mid s, a) - P(\cdot \mid s', a)\|_1 \leq \epsilon_{z,a}$$

Model-Misspecification from State-Abstraction



$\phi : S \to Z$

Group "similar" states (s) into an abstracted state (z)

$|Z| \ll |S|$

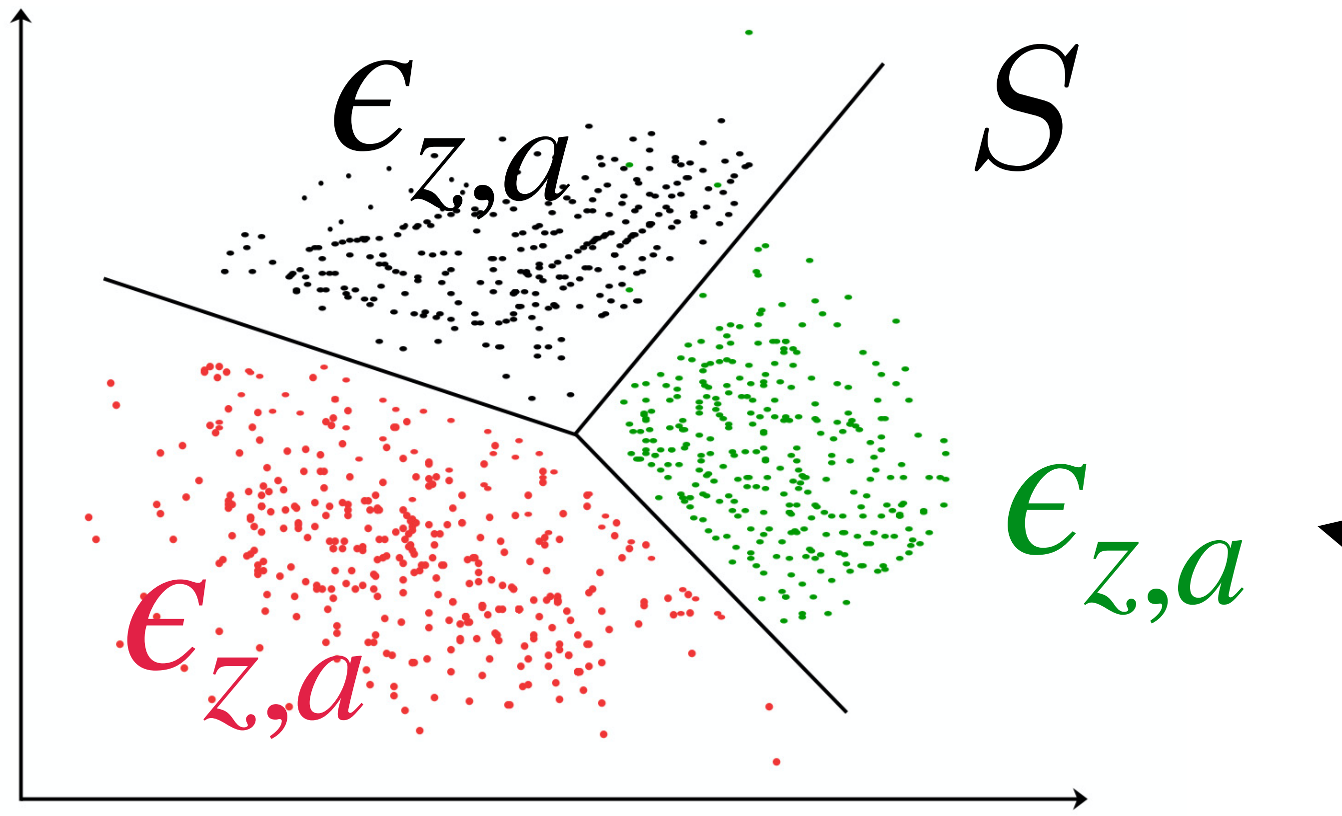
poly($|Z|$)

Model-misspecification:

$$s, s', \text{ s.t. } \phi(s) = \phi(s')$$

$$\|P(\cdot \mid s, a) - P(\cdot \mid s', a)\|_1 \leq \epsilon_{z,a}$$

Model-Misspecification from State-Abstraction



$\phi : S \to Z$

Group "similar" states (s) into an abstracted state (z)

$|Z| \ll |S|$

poly$(|Z|)$

Model-misspecification:

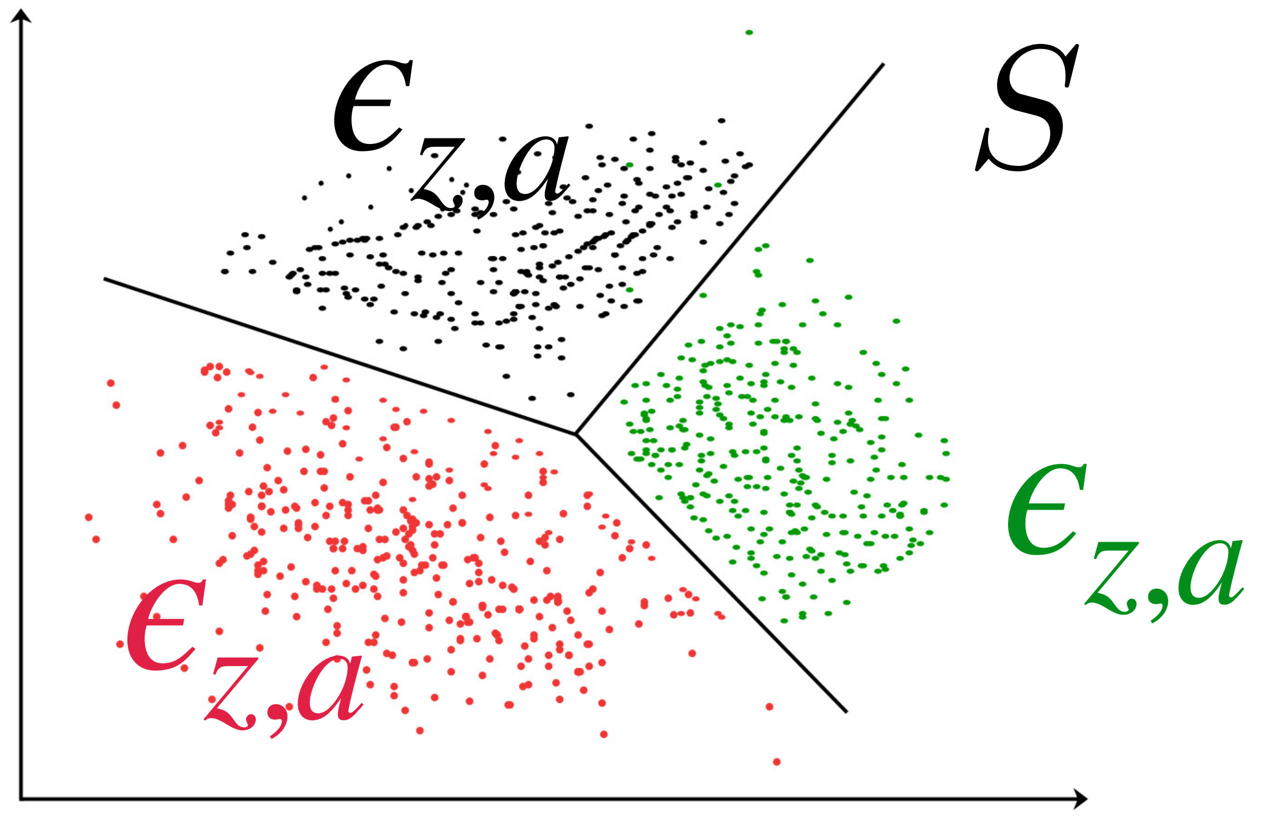
$$s, s', \text{ s.t. } \phi(s) = \phi(s')$$

$$\|P(\cdot \mid s, a) - P(\cdot \mid s', a)\|_1 \leq \epsilon_{z,a}$$

Bellman-backup based (e.g., Q-learning [Dong et al, 19])

$$\text{poly}\left(1/(1-\gamma)\right)\left(\max_{z,a} \epsilon_{z,a}\right)$$

Model-Misspecification from State-Abstraction



$$\phi : S \rightarrow Z$$

Group "similar" states (s) into an abstracted state (z)

$$|Z| \ll |S|$$

$$\text{poly}(|Z|)$$

Model-misspecification:

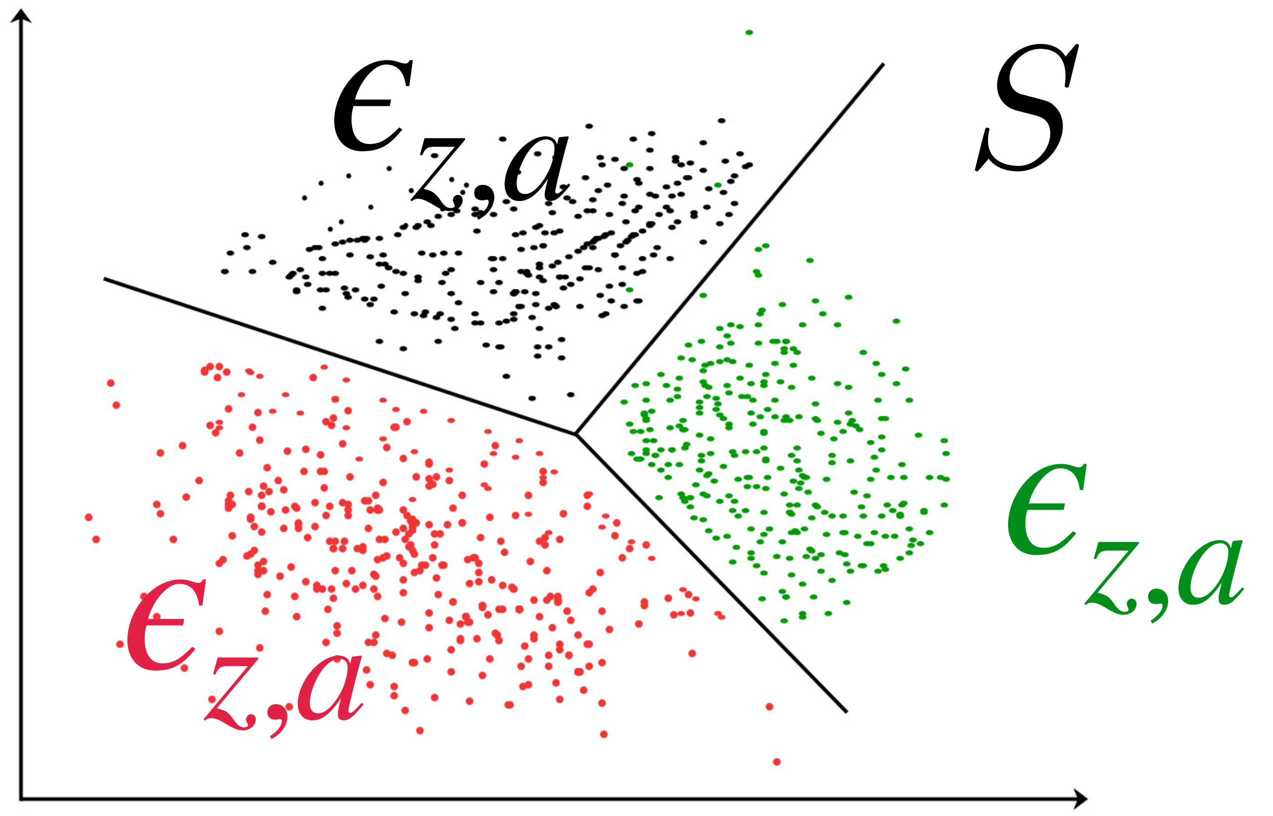
$$s, s', \text{ s.t. } \phi(s) = \phi(s')$$

$$\|P(\cdot \,|\, s, a) - P(\cdot \,|\, s', a)\|_1 \leq \epsilon_{z,a}$$

Bellman-backup based (e.g., Q-learning [Dong et al, 19])

$$\text{poly}\left(1/(1-\gamma)\right)\left(\max_{z,a} \epsilon_{z,a}\right) \longleftarrow \ell_\infty$$

Model-Misspecification from State-Abstraction



$\phi : S \to Z$

Group "similar" states (s) into an abstracted state (z)

$|Z| \ll |S|$

poly$(|Z|)$

Model-misspecification:

$$s, s', \text{ s.t. } \phi(s) = \phi(s')$$

$$\|P(\cdot\,|\,s,a) - P(\cdot\,|\,s',a)\|_1 \le \epsilon_{z,a}$$

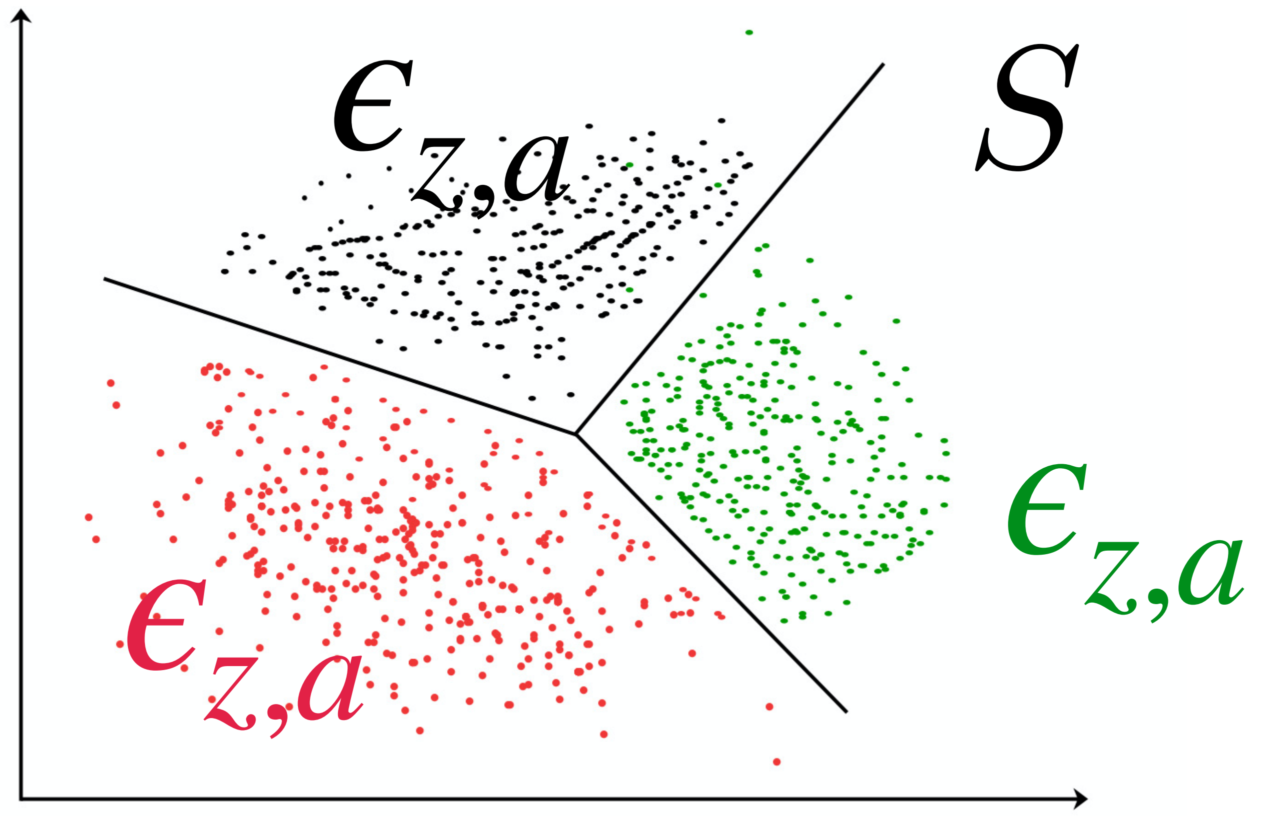
Bellman-backup based (e.g., Q-learning [Dong et al, 19])

$$\text{poly}\left(1/(1-\gamma)\right)\left(\max_{z,a} \epsilon_{z,a}\right) \longleftarrow \ell_\infty$$

PC-PG

$$\text{poly}(1/(1-\gamma))\left(\mathbb{E}_{z,a\sim d^{\tilde{\pi}}}\left[\epsilon_{z,a}\right]\right)$$

Model-Misspecification from State-Abstraction



$\phi : S \to Z$

Group "similar" states (s) into an abstracted state (z)

$|Z| \ll |S|$

poly$(|Z|)$

Model-misspecification:

$$s, s', \text{ s.t. } \phi(s) = \phi(s')$$

$$\|P(\cdot \mid s, a) - P(\cdot \mid s', a)\|_1 \leq \epsilon_{z,a}$$

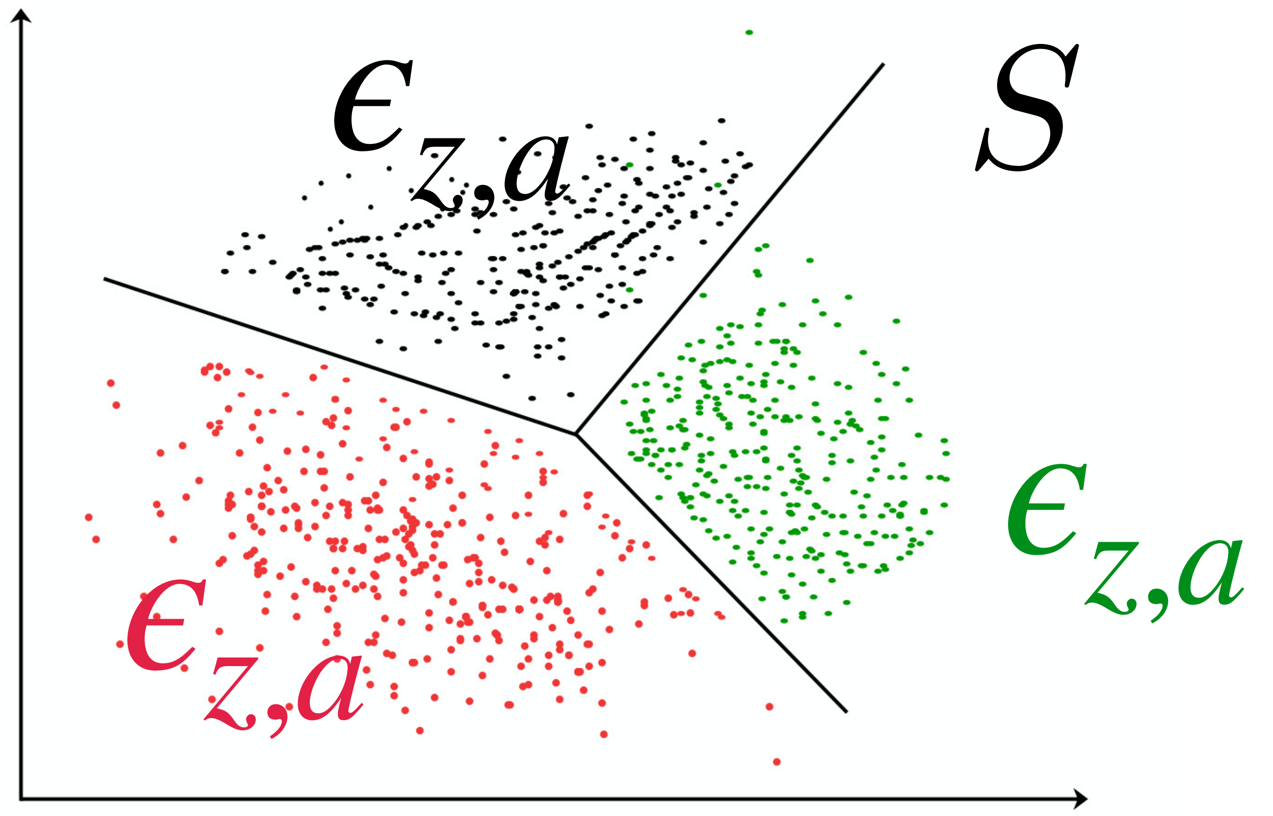
Bellman-backup based (e.g., Q-learning [Dong et al, 19])

$$\text{poly}\left(1/(1-\gamma)\right)\left(\max_{z,a} \epsilon_{z,a}\right) \longleftarrow \ell_\infty$$

PC-PG

$$\text{poly}(1/(1-\gamma))\left(\mathbb{E}_{z,a\sim d^{\tilde{\pi}}}\left[\epsilon_{z,a}\right]\right)$$

Model-Misspecification from State-Abstraction



$$\phi : S \to Z$$

Group "similar" states (s) into an abstracted state (z)

$$|Z| \ll |S|$$

$$\text{poly}(|Z|)$$

Model-misspecification:

$$s, s', \text{ s.t. } \phi(s) = \phi(s')$$

$$\|P(\cdot \mid s, a) - P(\cdot \mid s', a)\|_1 \le \epsilon_{z,a}$$

Bellman-backup based (e.g., Q-learning [Dong et al, 19])

$$\text{poly}\left(1/(1-\gamma)\right)\left(\max_{z,a} \epsilon_{z,a}\right) \leftarrow \ell_\infty$$

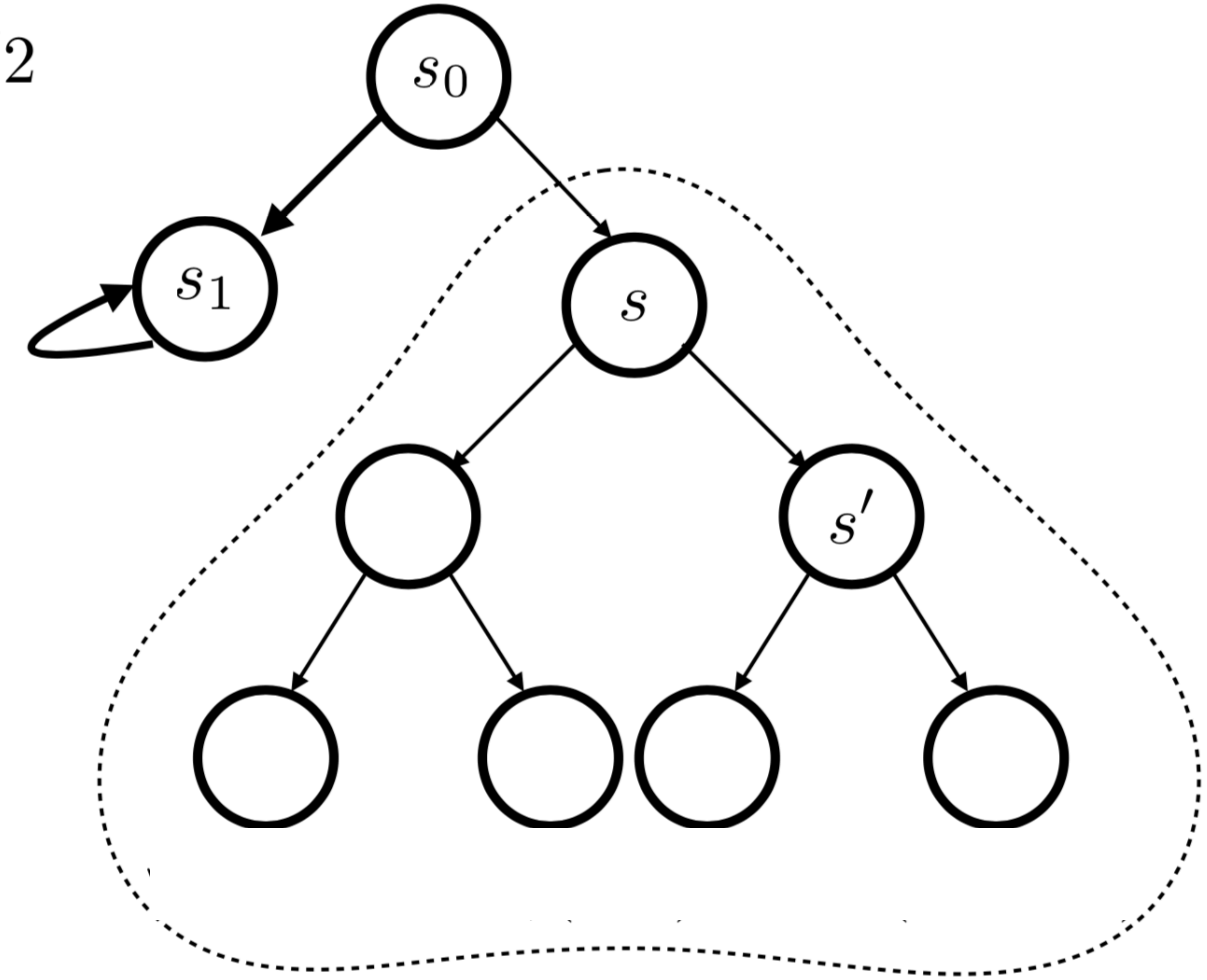
PC-PG

$$\text{poly}(1/(1-\gamma))\left(\mathbb{E}_{z,a\sim d^{\tilde{\pi}}}\left[\epsilon_{z,a}\right]\right)$$

Average over comparator's distribution

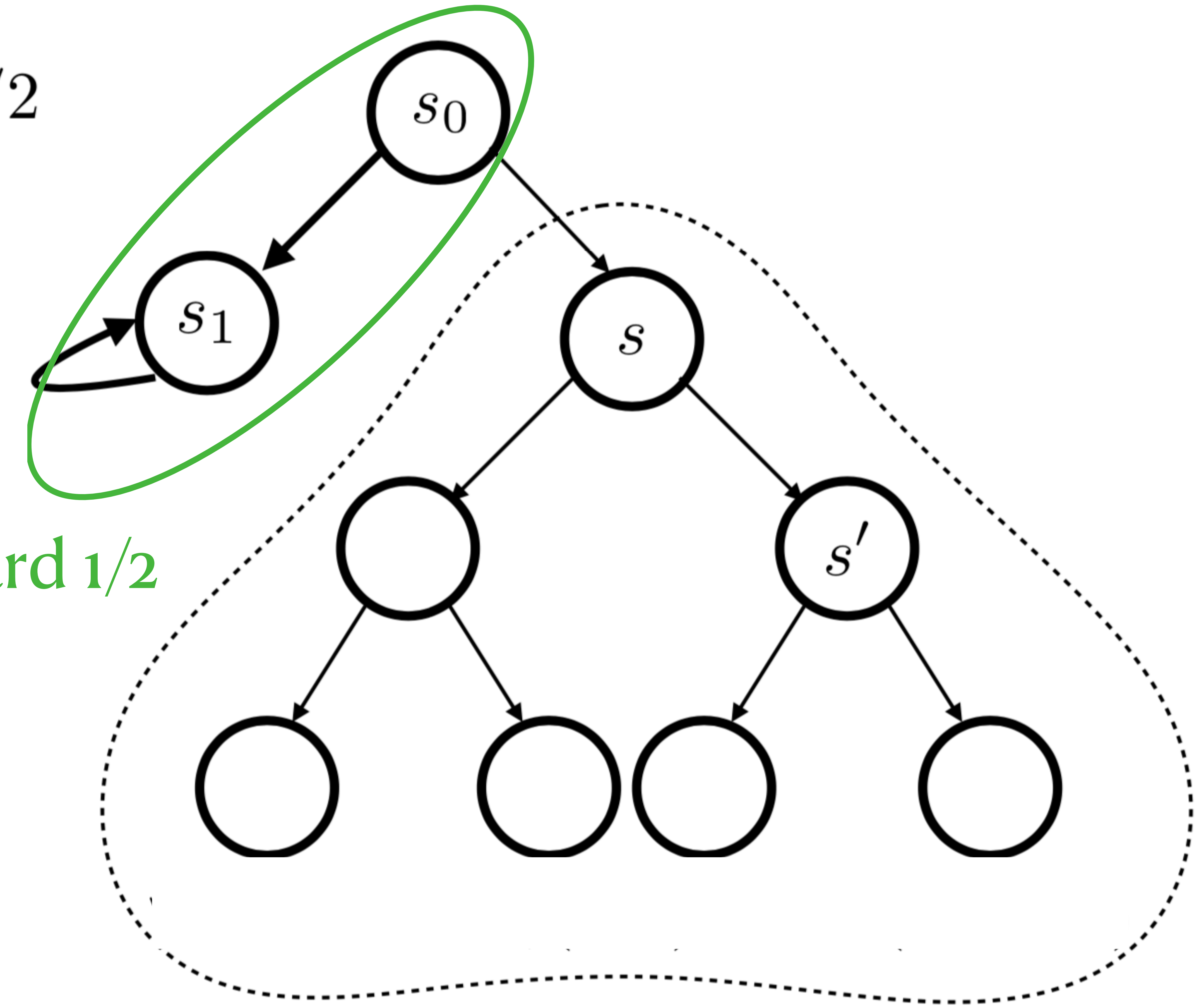
Additional Example w/ linear approximation

$r(s_0, L) = 1/2$



Additional Example w/ linear approximation

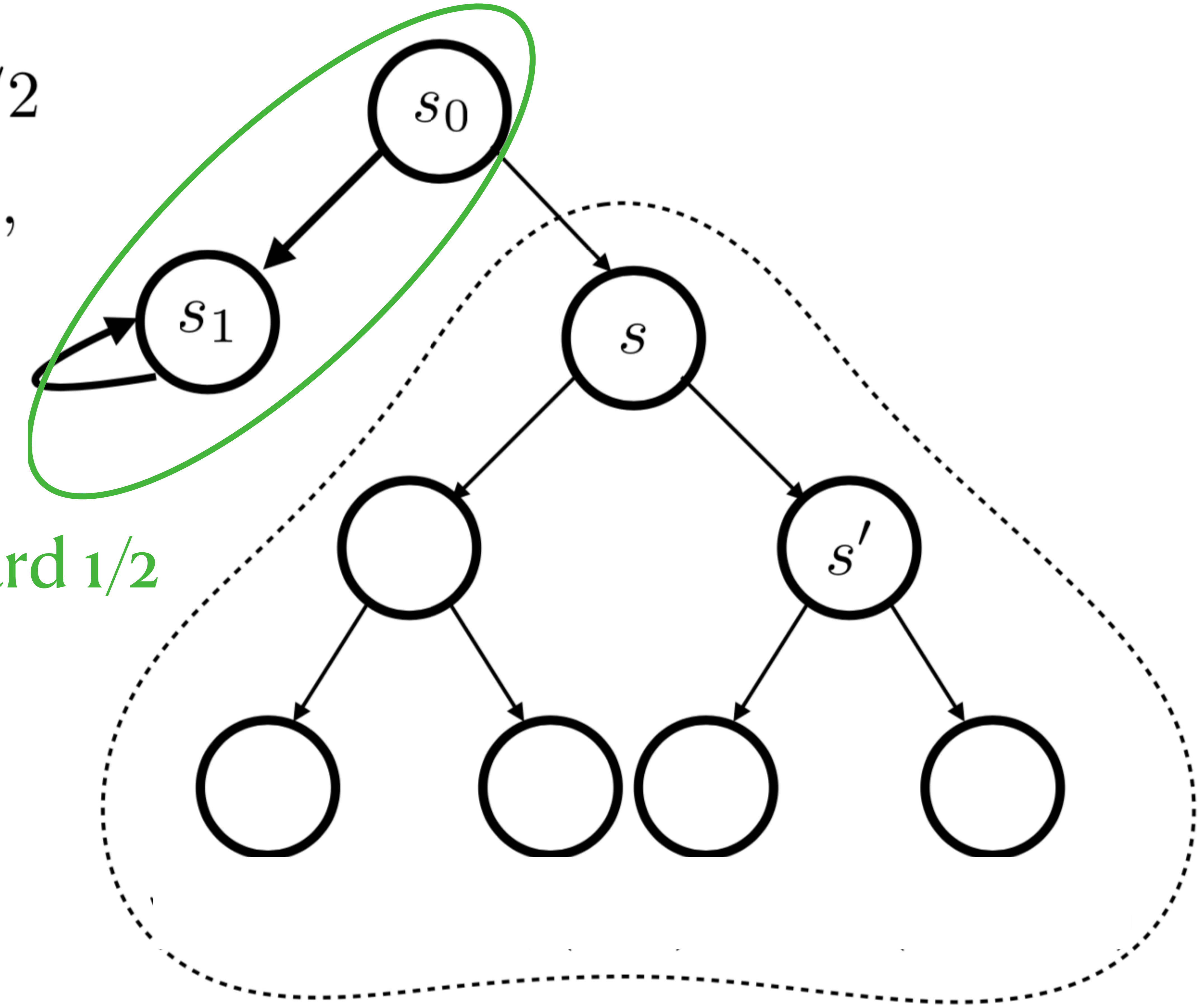
$r(s_0, L) = 1/2$



Path with reward 1/2

Additional Example w/ linear approximation

$$r(s_0, L) = 1/2$$
$$\phi(s_0, L) = e_1,$$



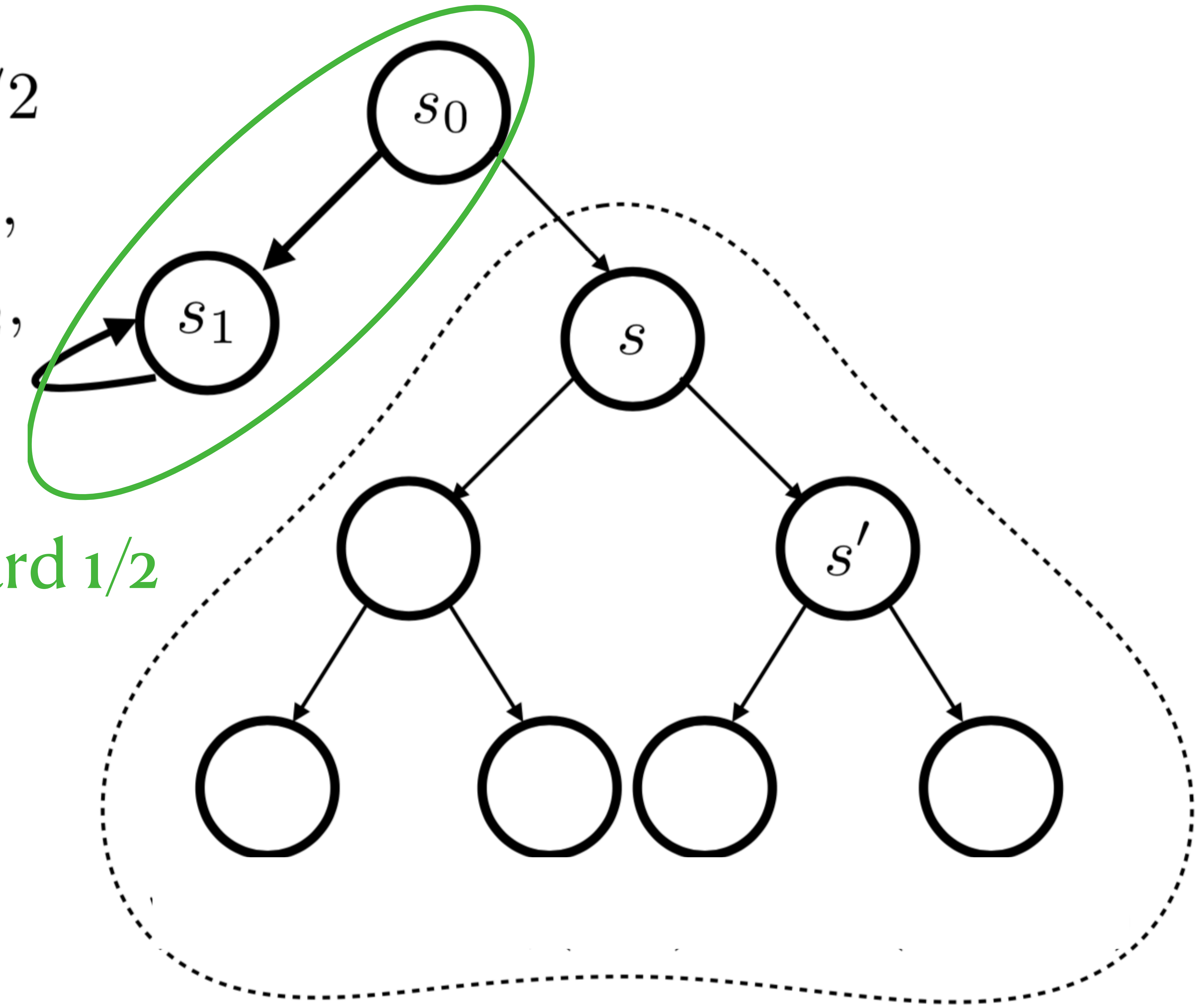
Path with reward 1/2

Additional Example w/ linear approximation

$r(s_0, L) = 1/2$

$\phi(s_0, L) = e_1,$

$\phi(s_0, R) = e_2,$



Path with reward 1/2

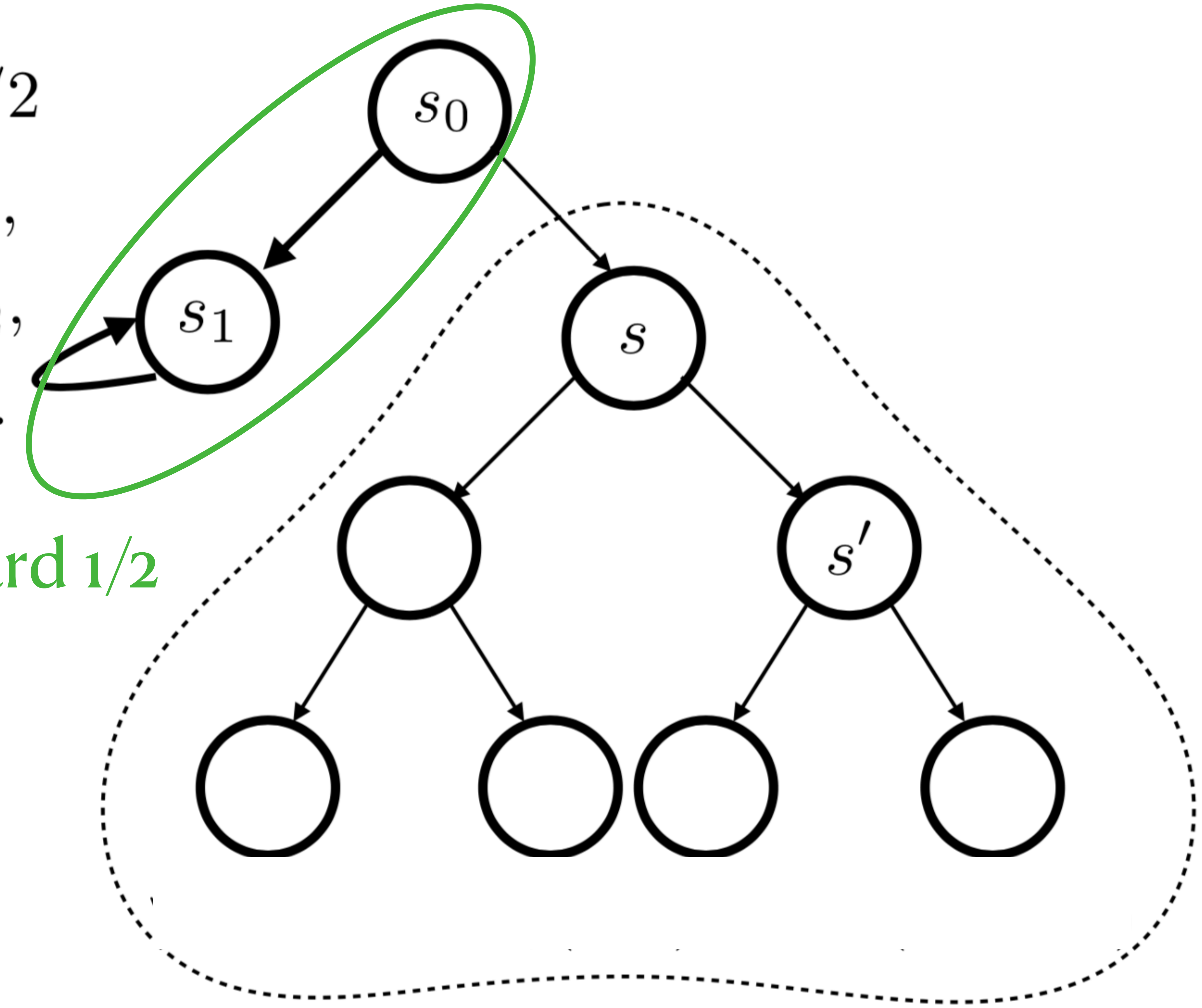
Additional Example w/ linear approximation

$r(s_0, L) = 1/2$

$\phi(s_0, L) = e_1,$

$\phi(s_0, R) = e_2,$

$\phi(s_1, a) = e_3.$

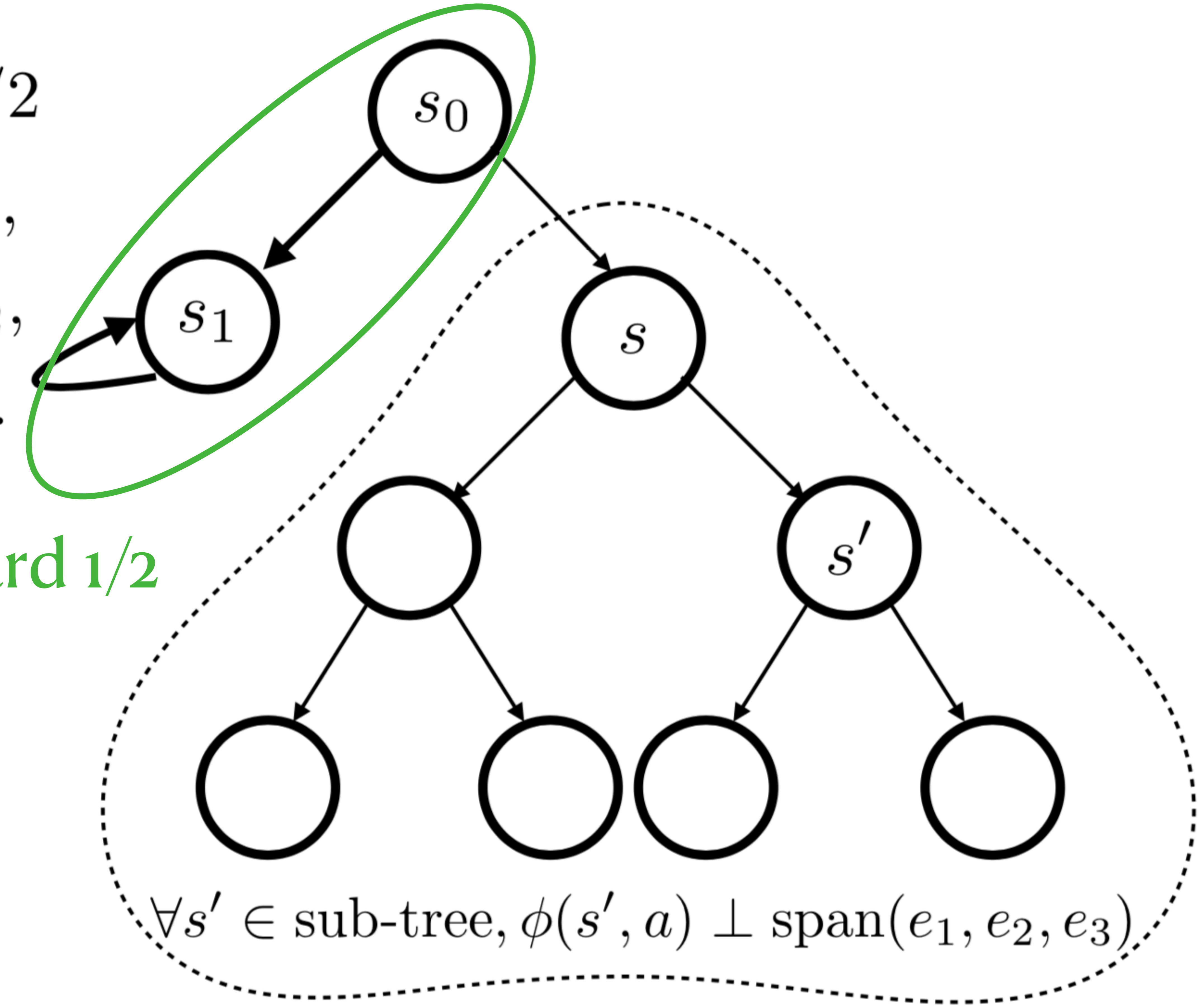


Path with reward 1/2

Additional Example w/ linear approximation

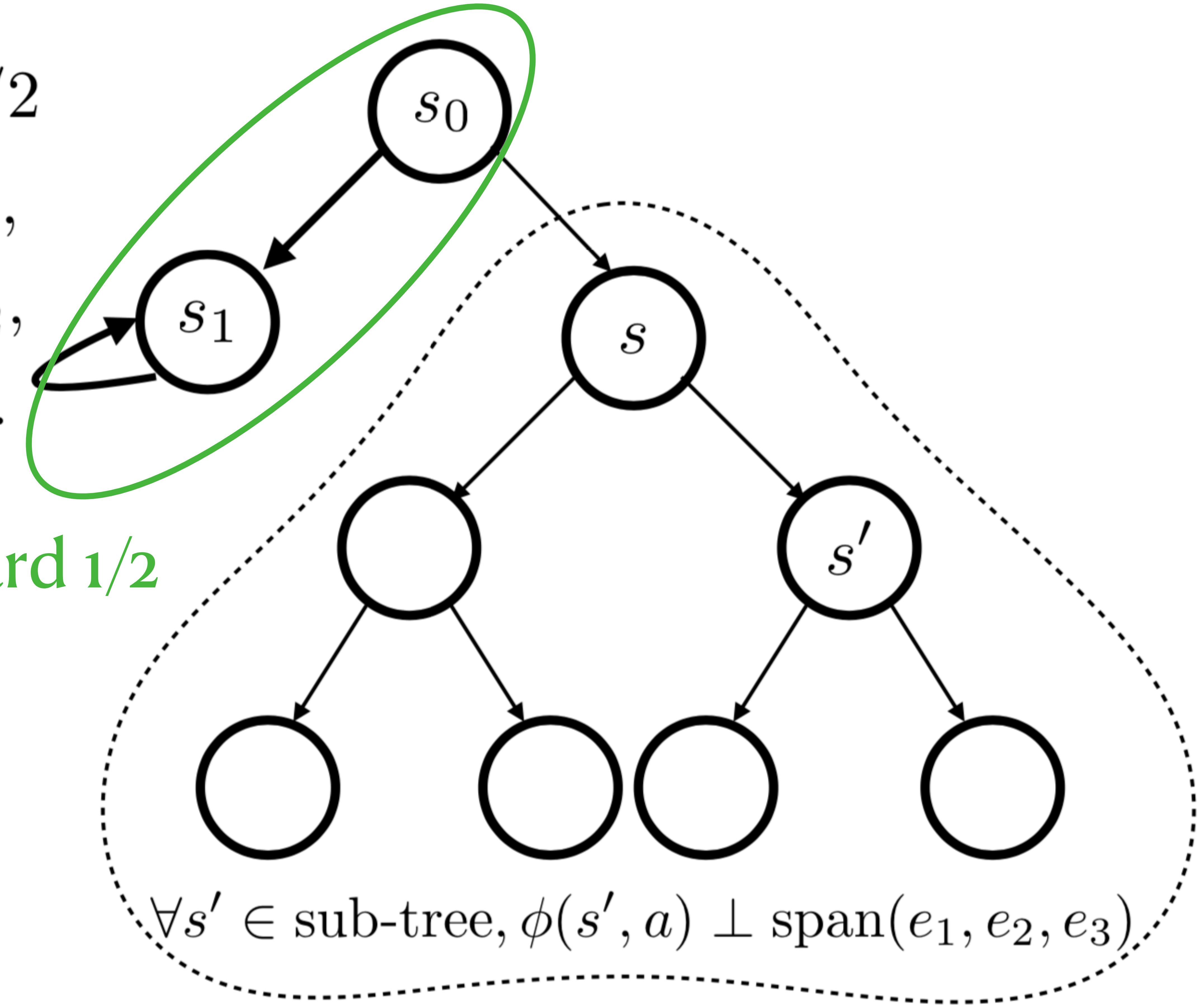
$$r(s_0, L) = 1/2$$
$$\phi(s_0, L) = e_1,$$
$$\phi(s_0, R) = e_2,$$
$$\phi(s_1, a) = e_3.$$

Path with reward 1/2



Additional Example w/ linear approximation

$$r(s_0, L) = 1/2$$
$$\phi(s_0, L) = e_1,$$
$$\phi(s_0, R) = e_2,$$
$$\phi(s_1, a) = e_3.$$

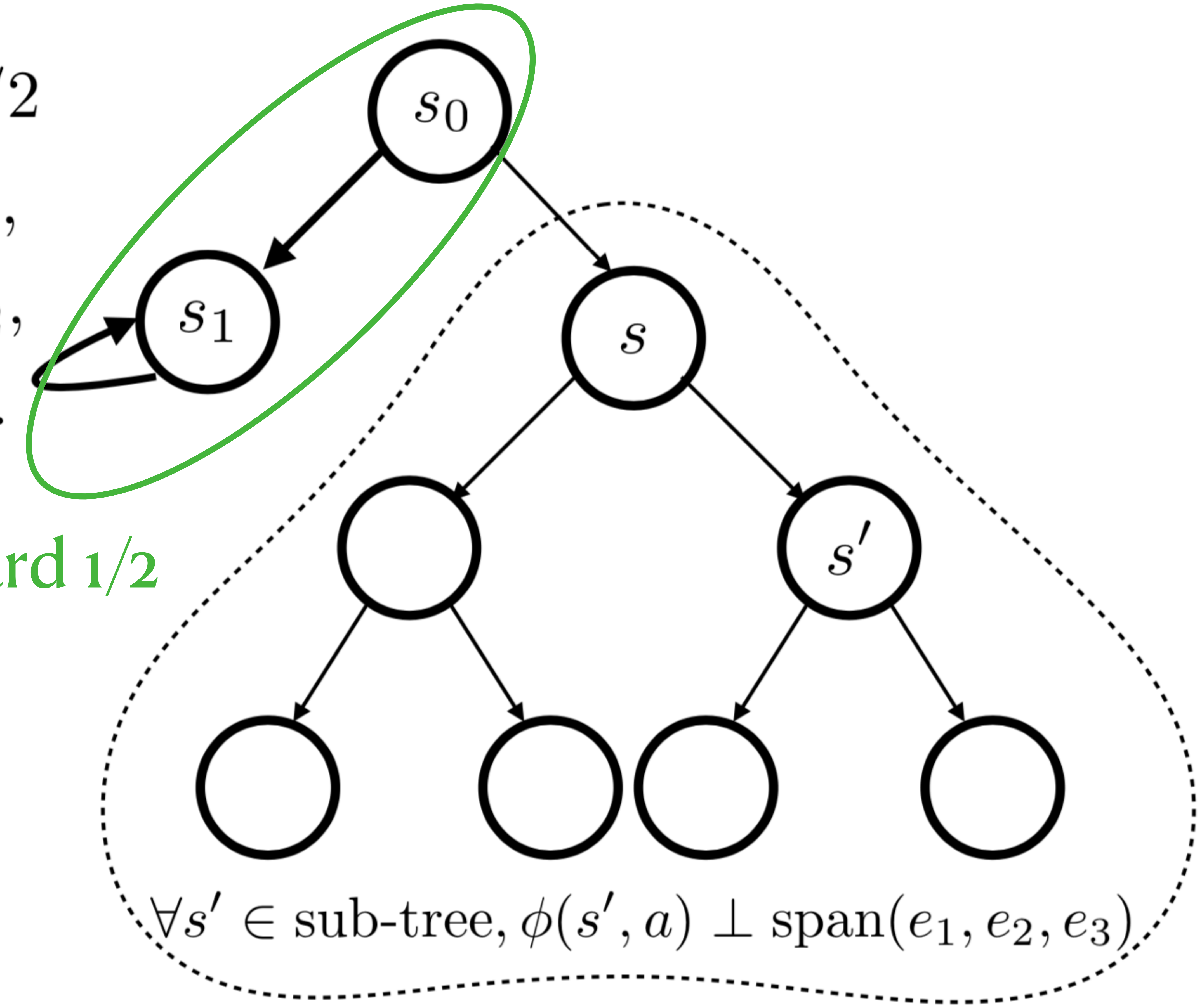


Path with reward 1/2

Features is **arbitrary** in the tree, i.e., model-misspecification can be very **serious**.

Additional Example w/ linear approximation

$$r(s_0, L) = 1/2$$
$$\phi(s_0, L) = e_1,$$
$$\phi(s_0, R) = e_2,$$
$$\phi(s_1, a) = e_3.$$



Path with reward 1/2

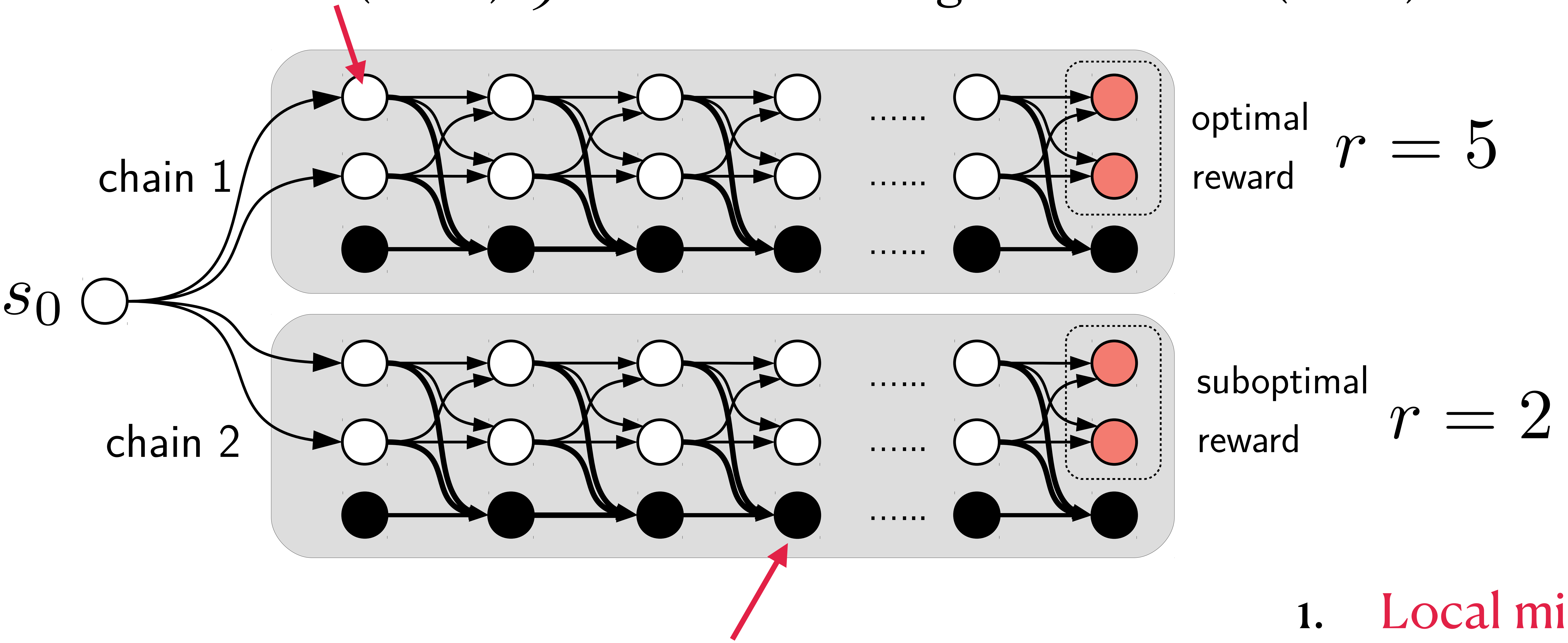
PC-PG is at least as good as the green trajectory

Features is **arbitrary** in the tree, i.e., model-misspecification can be very **serious**.

Experiments

Bidirectional Combination Lock

Good state (white): 9 out of 10 actions go to bad state (black)



Bad state has Anti-shaped reward: $r = 1/H$

1. Local minima
2. Forgetting

Experiments on Bi-directional Comb Lock

Feature vector: binary vector indicating state-action and time step

Policy Opt procedure: PPO w/ NN policy

Bonus: $\phi^\top \Sigma_n^{-1} \phi$

Experiments on Bi-directional Comb Lock

Feature vector: binary vector indicating state-action and time step

Policy Opt procedure: PPO w/ NN policy

Bonus: $\phi^\top \Sigma_n^{-1} \phi$

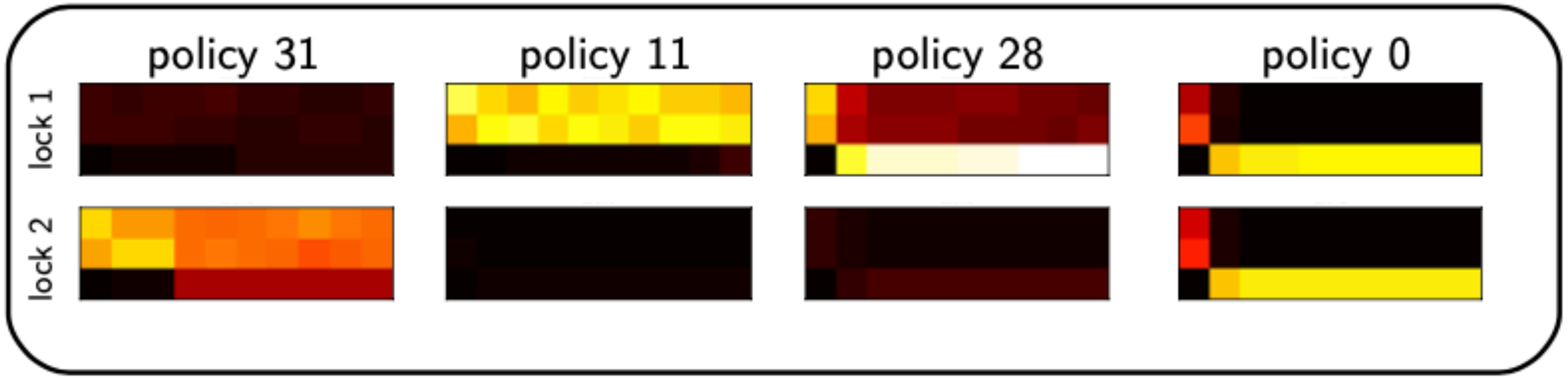
Success Rate (visit two chains):

Algorithm	Horizon			
	2	5	10	15
PPO	1.0	0.0	0.0	0.0
PPO+RND	0.75	0.40	0.50	0.55
PC-PG	1.0	1.0	1.0	1.0

Experiments on Bi-directional Comb Lock

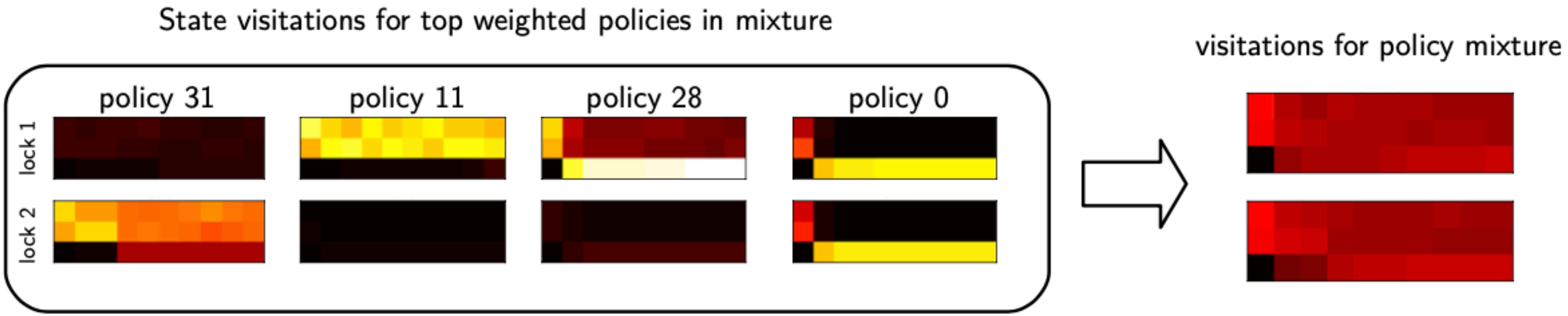
Due to the policy cover PC-PG maintains..

State visitations for top weighted policies in mixture



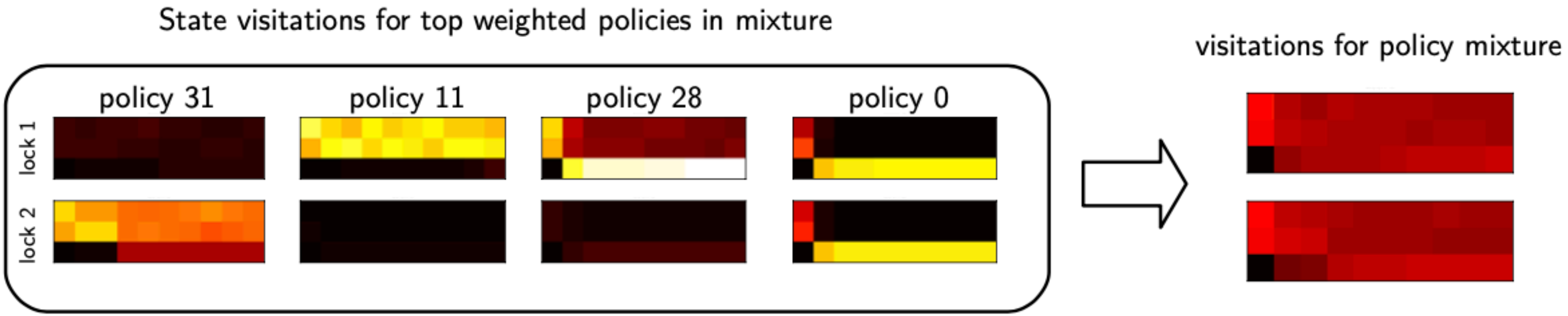
Experiments on Bi-directional Comb Lock

Due to the policy cover PC-PG maintains..



Experiments on Bi-directional Comb Lock

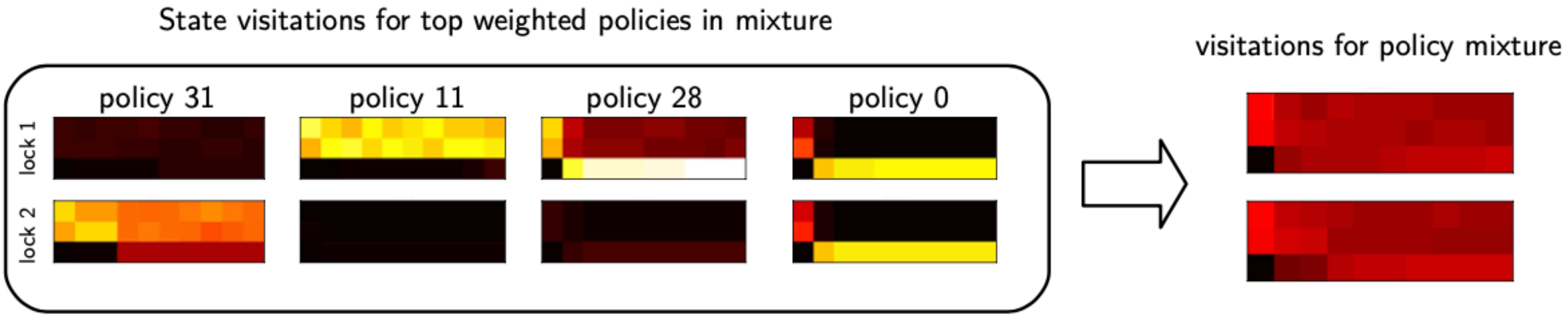
Due to the policy cover PC-PG maintains..



Cover ρ_n near uniformly cover both chains

Experiments on Bi-directional Comb Lock

Due to the policy cover PC-PG maintains..



Cover ρ_n near uniformly cover both chains

PG with ρ_n will succeed!

Reward-Free Explore in Maze

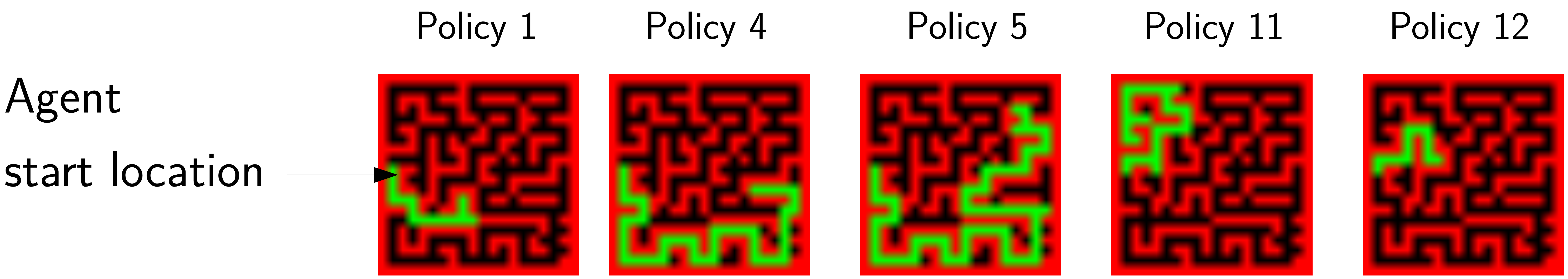
$$r(s, a) = 0$$

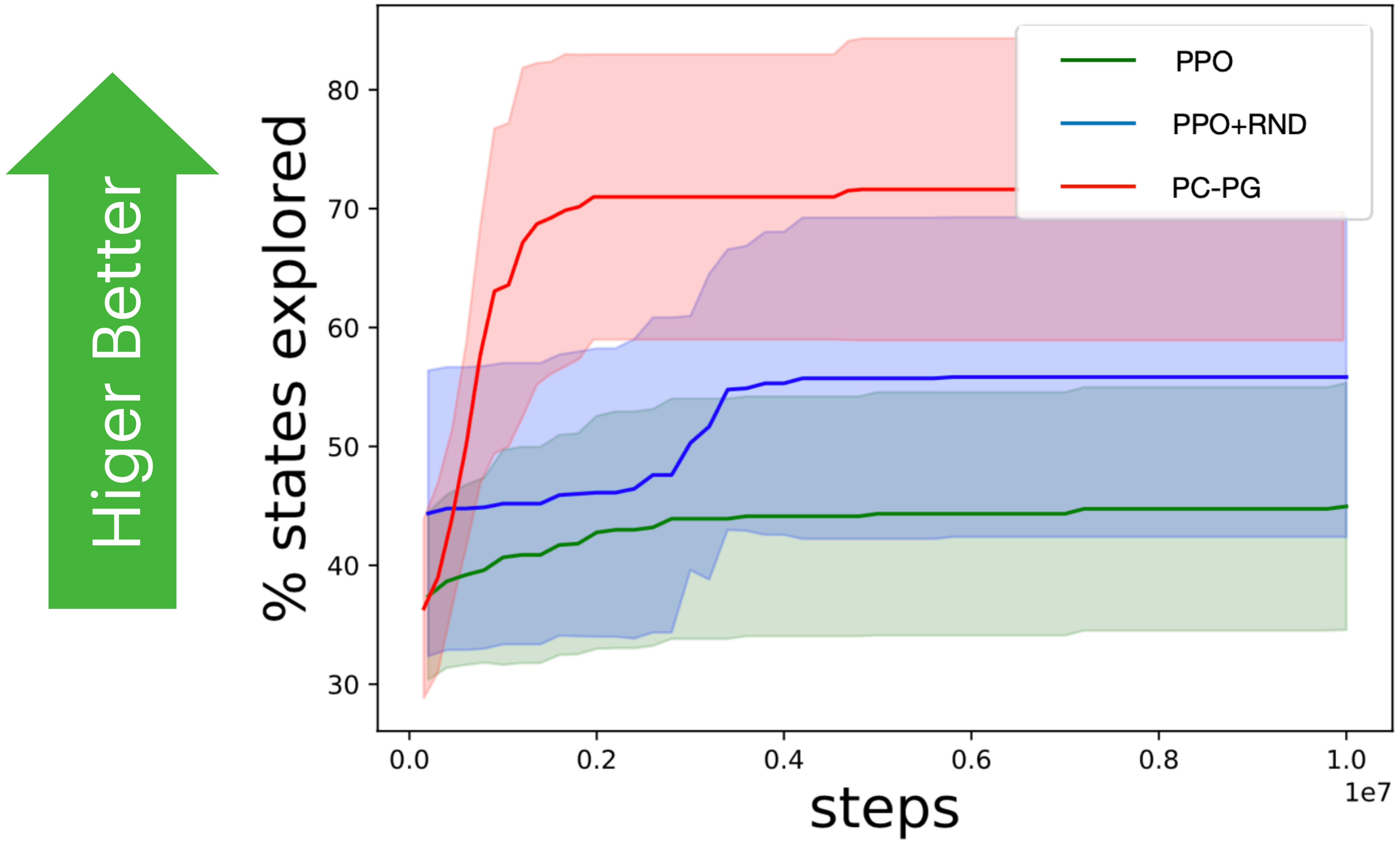
$\phi(s, a)$: Random initialized CovNet

Policy Opt procedure: PPO w/ CovNet-based policy

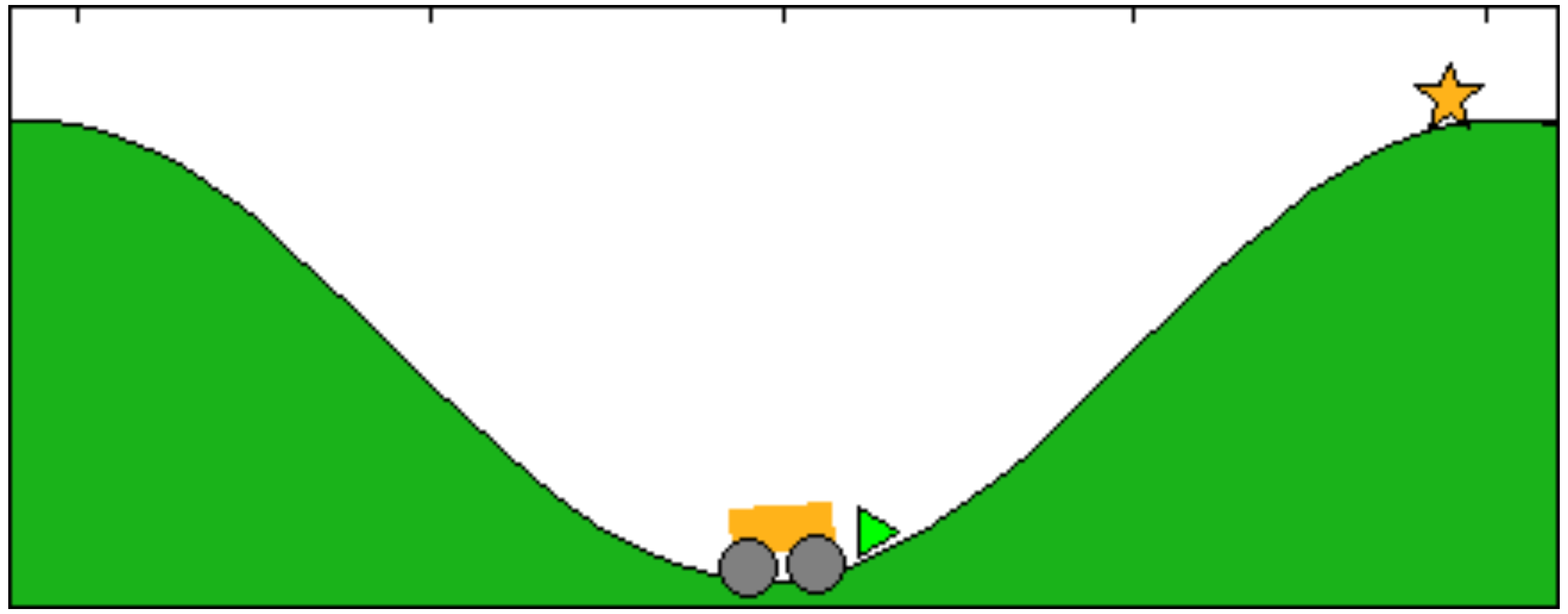
Reward-Free Explore in Maze

Traces of policies in the policy cover:





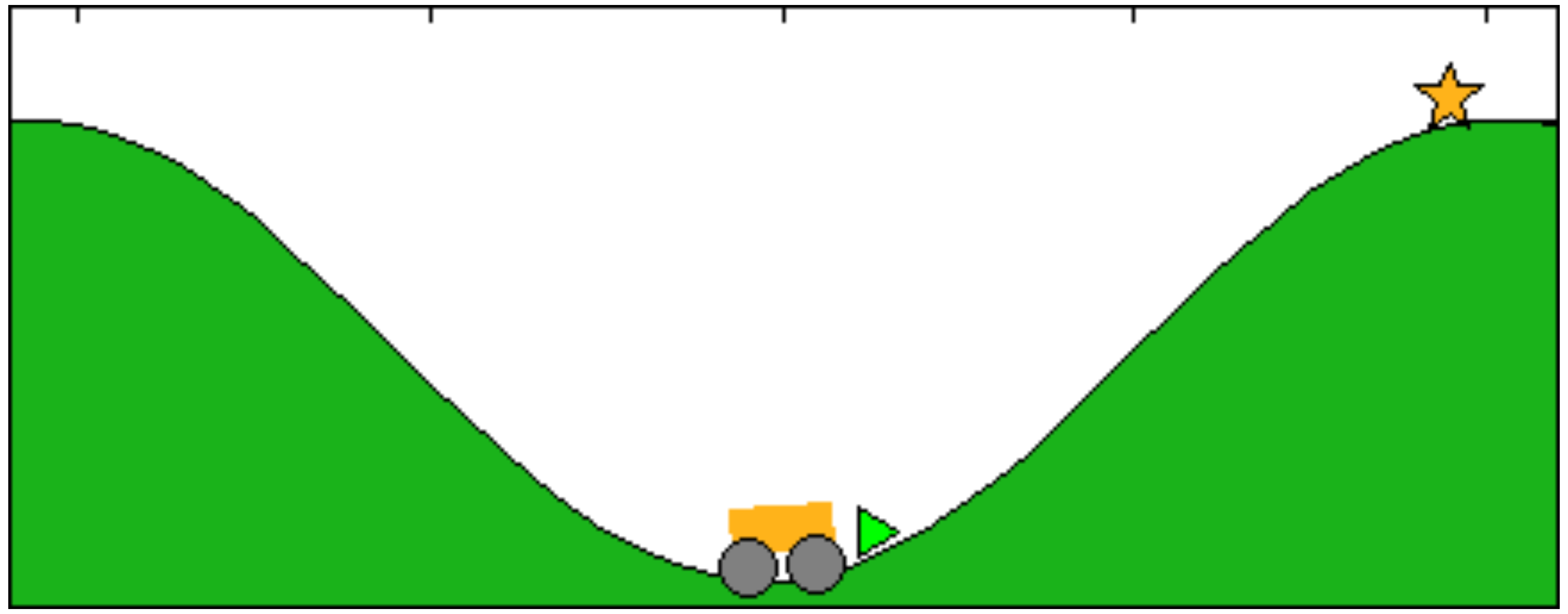
Continuous Control w/ sparse reward

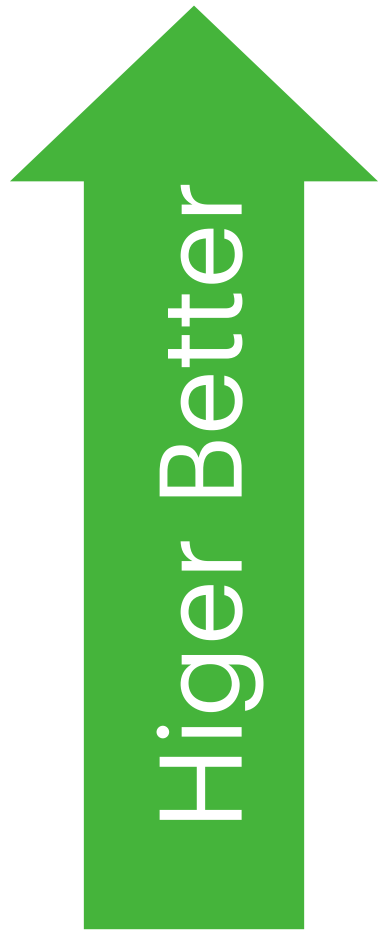


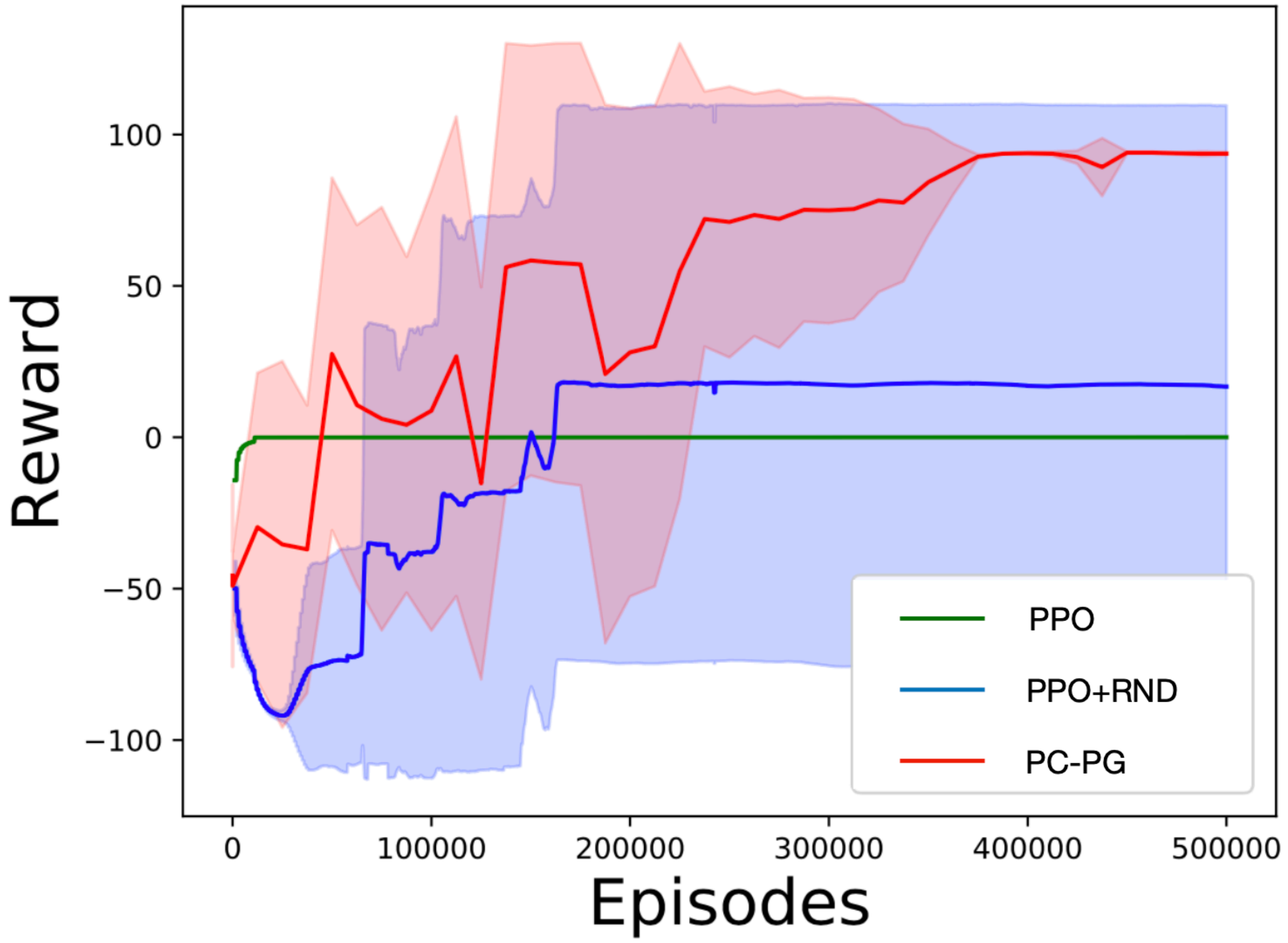
Sparse reward: large reward at the goal;

Anti-shaped reward: penalize control inputs

Continuous Control w/ sparse reward



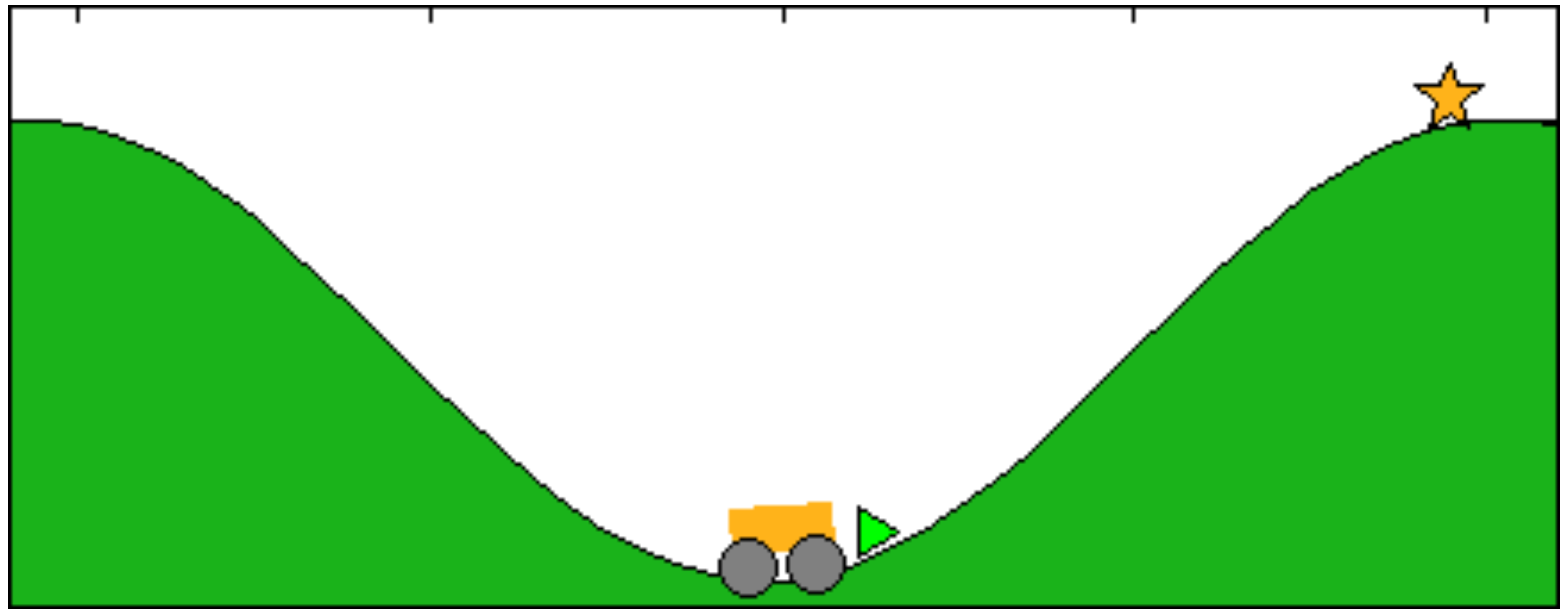


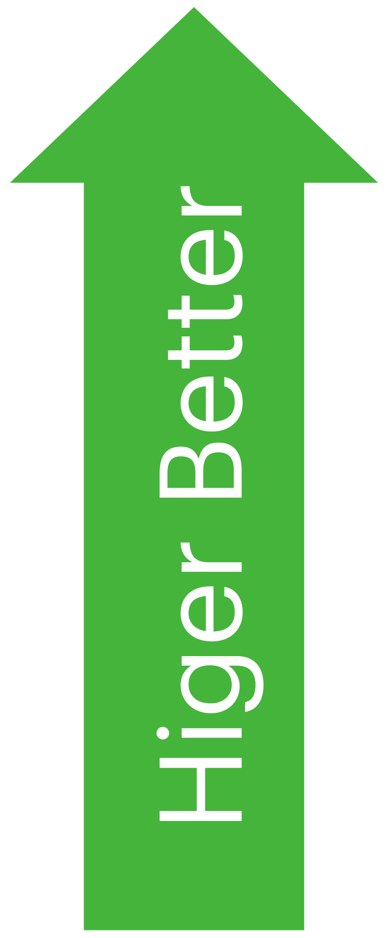


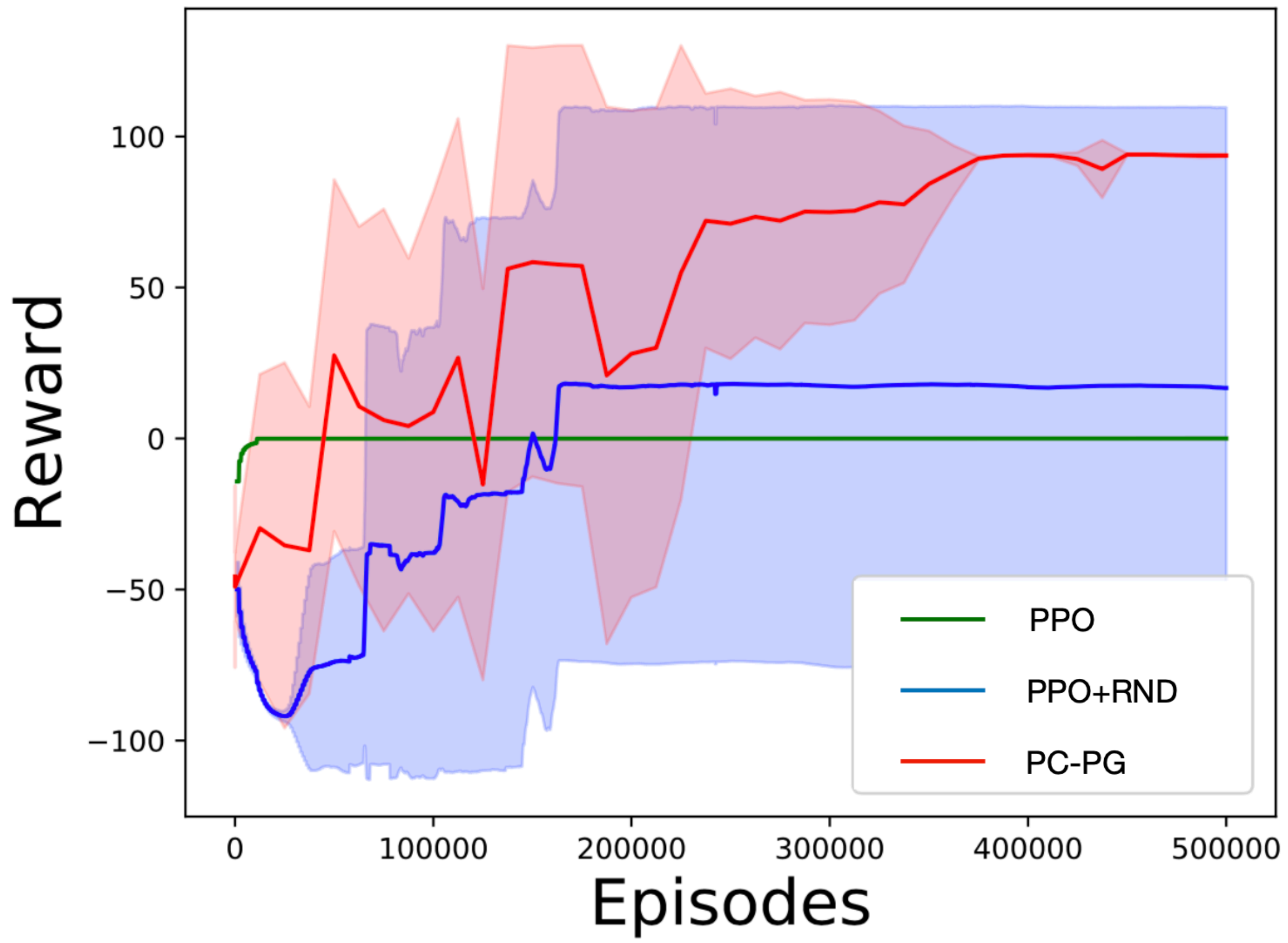
Sparse reward: large reward at the goal;

Anti-shaped reward: penalize control inputs

Continuous Control w/ sparse reward

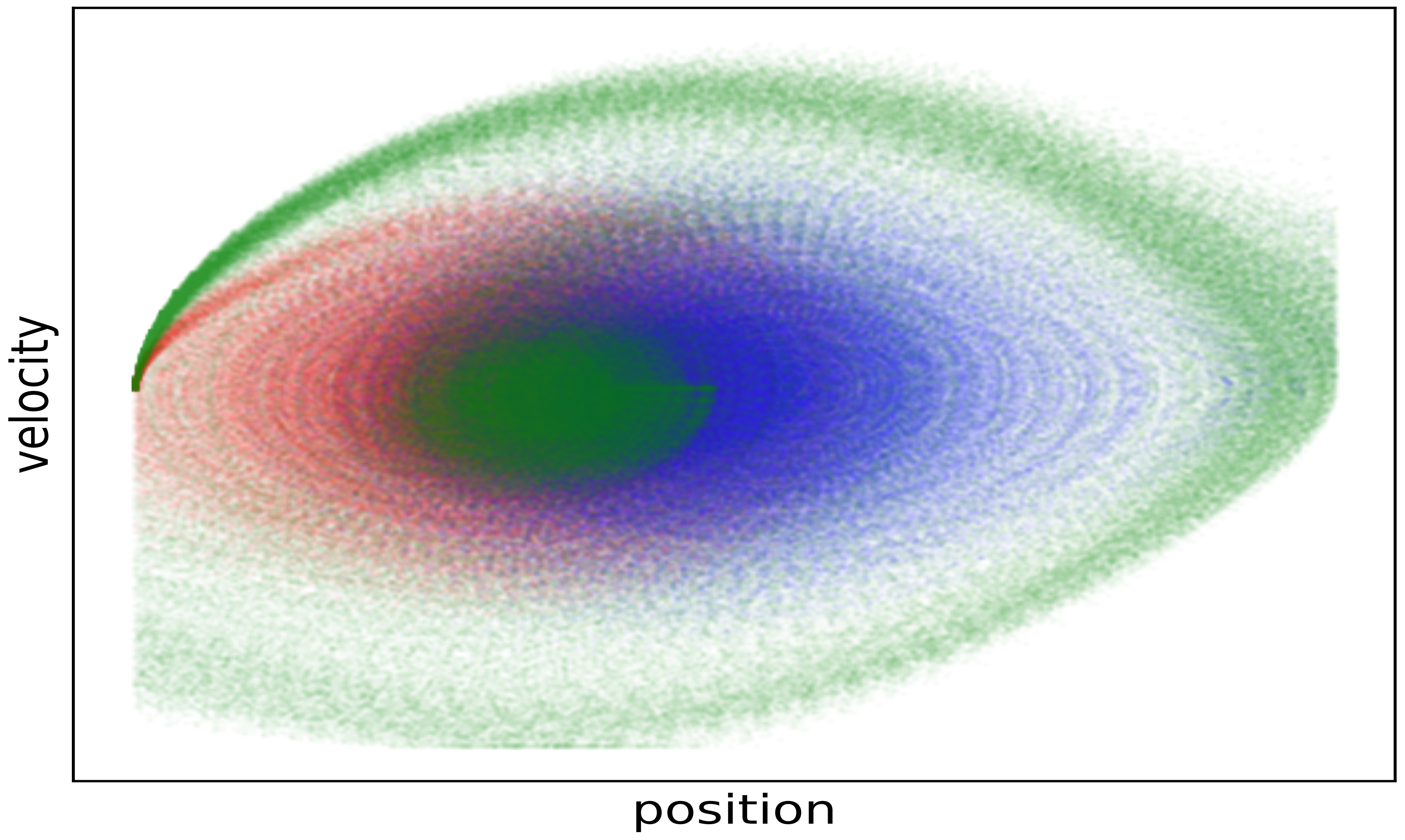






Sparse reward: large reward at the goal;

Anti-shaped reward: penalize control inputs



Conclusion

Conclusion

Strong agnostic results

Average model-misspecification VS ℓ_∞

Conclusion

Strong agnostic results

Average model-misspecification VS ℓ_∞

Polynomial Sample Complexity in well-specified case:

Linear MDPs (RKHS) & Tabular MDPs

Conclusion

Strong agnostic results

Average model-misspecification VS ℓ_∞

Polynomial Sample Complexity in well-specified case:

Linear MDPs (RKHS) & Tabular MDPs

Policy Cover/bonus solve the issue of flatten gradient & Forgetting

Treat policy cover's distribution as the reset distribution for PG

Conclusion

Strong agnostic results

Average model-misspecification VS ℓ_∞

Polynomial Sample Complexity in well-specified case:

Linear MDPs (RKHS) & Tabular MDPs

Policy Cover/bonus solve the issue of flatten gradient & Forgetting

Treat policy cover's distribution as the reset distribution for PG

Flexibility to leverage existing deep learning/RL tools

Vanilla implementation explores 4 to 5 rooms in M-Revenge