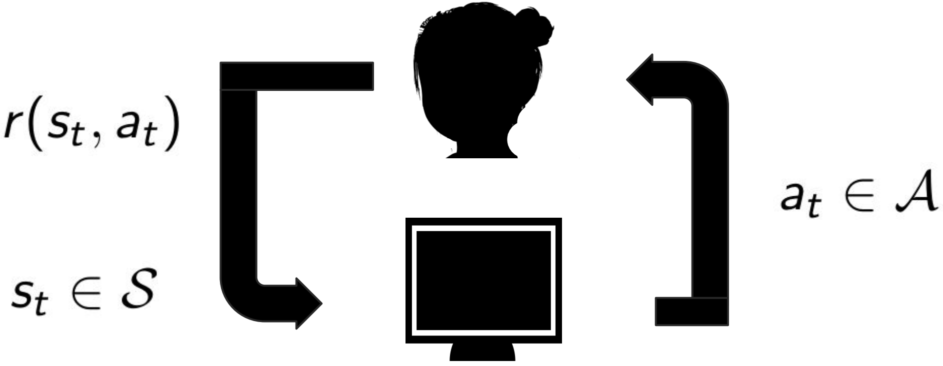
Reinforcement Learning

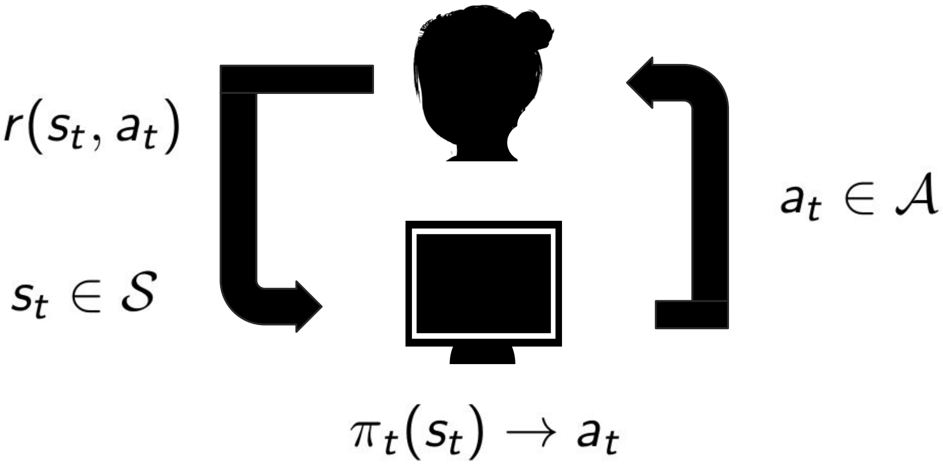


$$\pi_t(s_t) \rightarrow a_t$$

$$\underbrace{V^\pi(s)}_{\text{Value func.}} = \underbrace{r(s, \pi(s))}_{\text{Reward}} + \gamma \sum_{s'} \underbrace{p(s'|s, a)}_{\text{Dynamics}} V^\pi(s')$$

Only observed through samples (experience)

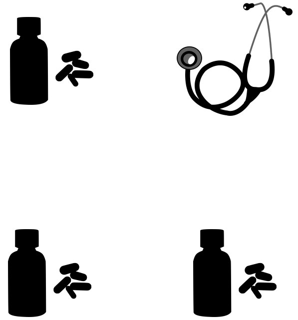
New Topic: Counterfactual / Batch RL



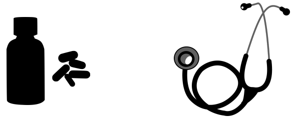
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$

Patient group 1 → Outcome: 92

Patient group 2 → Outcome: 91



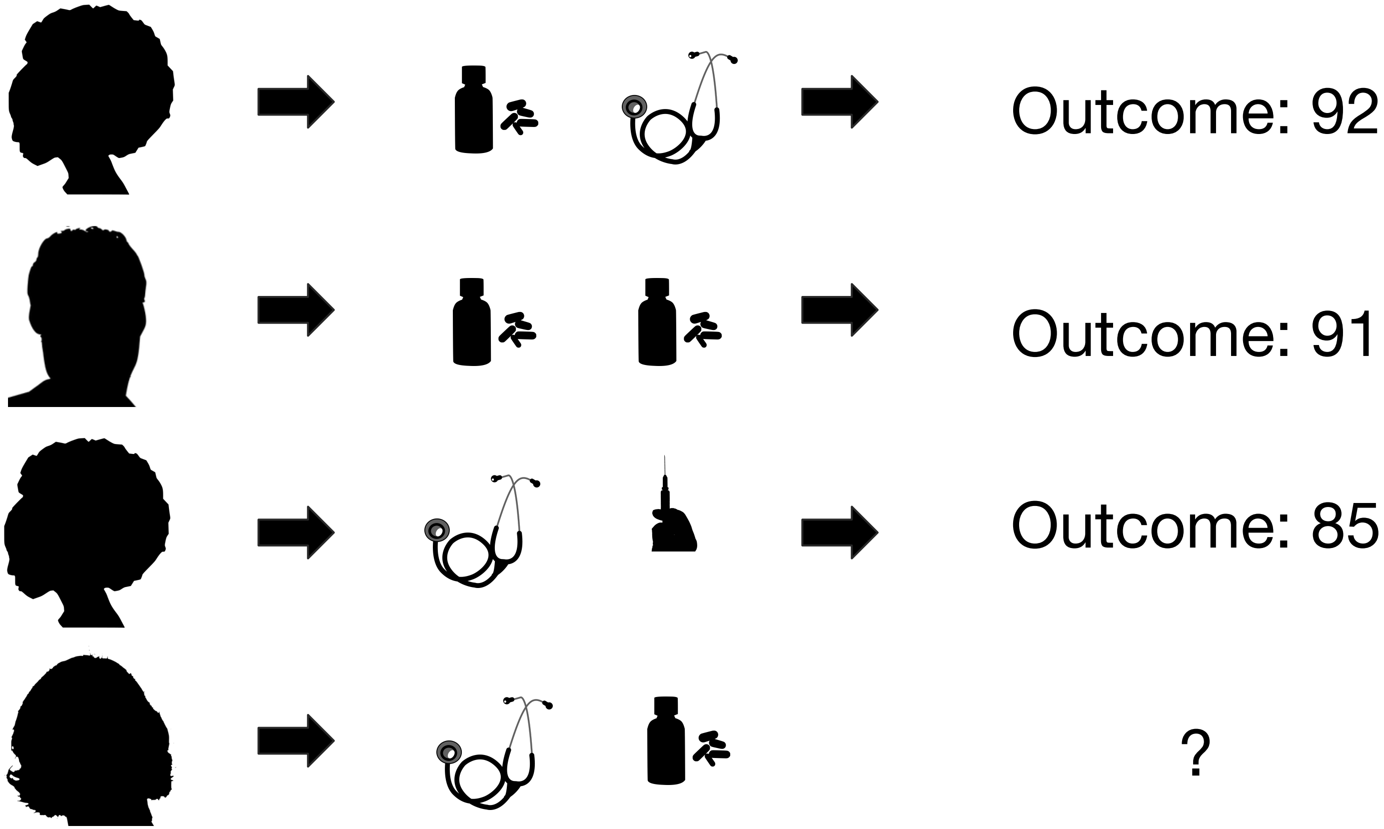
Data Is Censored in that Only Observe Outcomes for Decisions Made

Patient group 1 ➡  ➡ Outcome: 92

Patient group 2 ➡  ➡ Outcome: 91

 ➡ ?

Need for Generalization

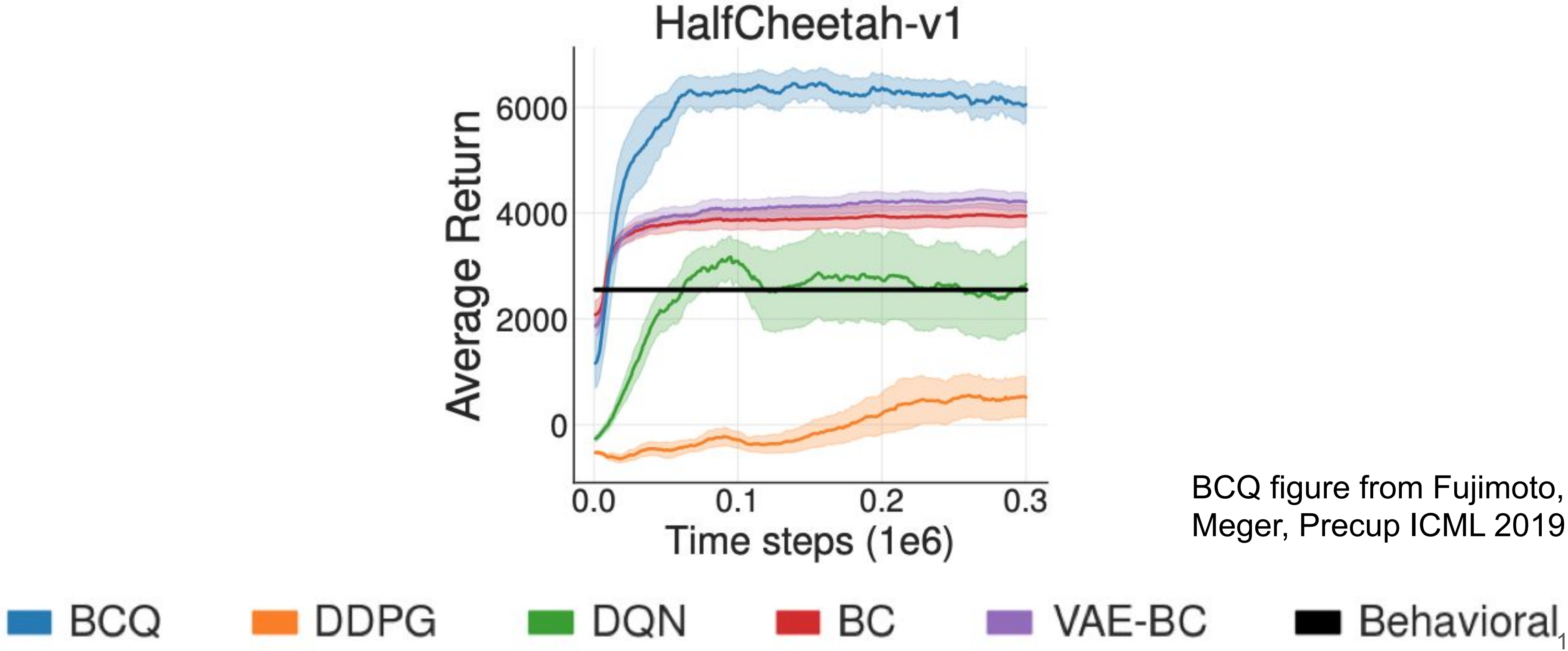


Why Can’t We Just Use Q-Learning?

- Q-learning is an off policy RL algorithm
  - Can be used with data different than the state--action pairs would visit under the optimal Q state action values

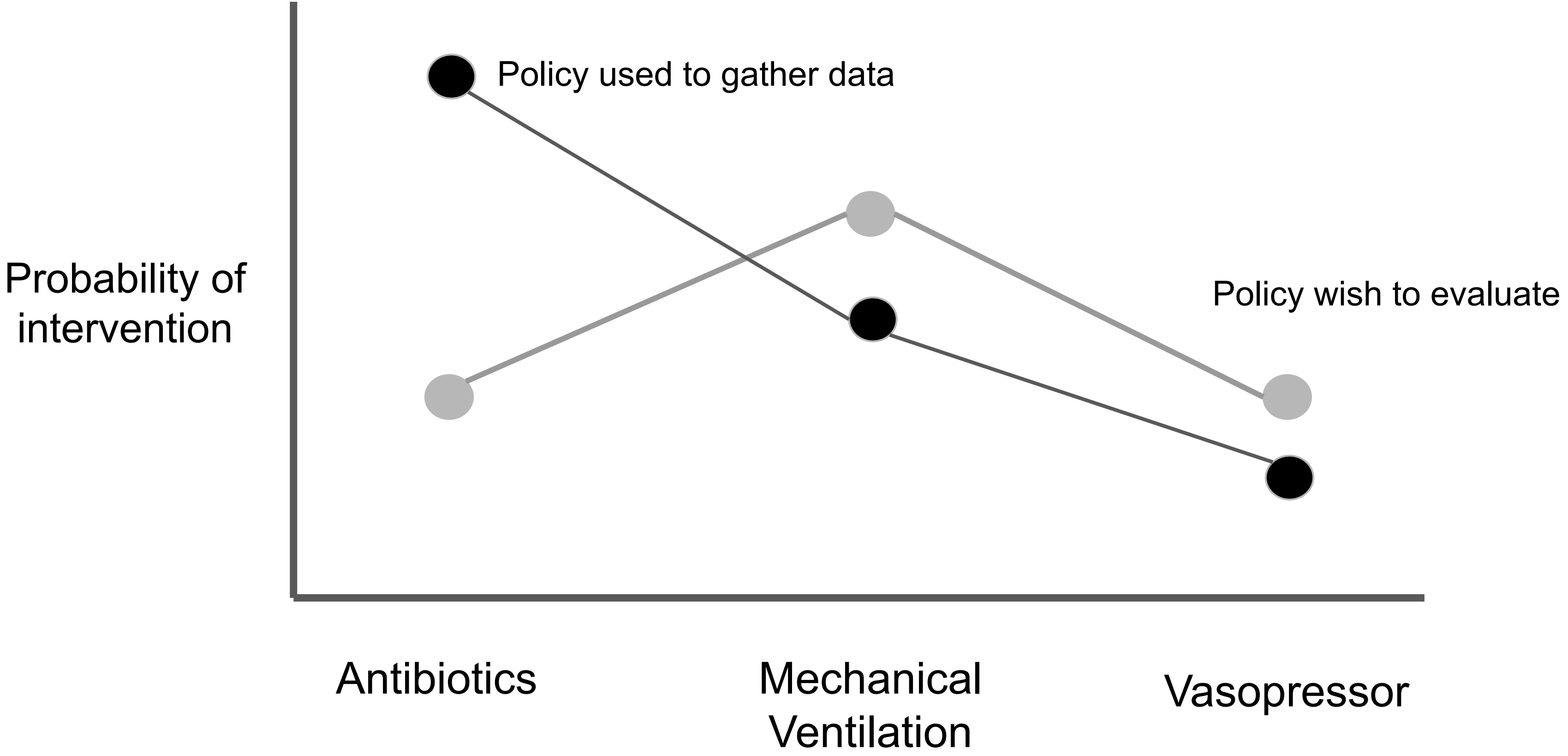
- But deadly triad of bootstrapping, function approximation and off policy, and can fail

Important in Practice

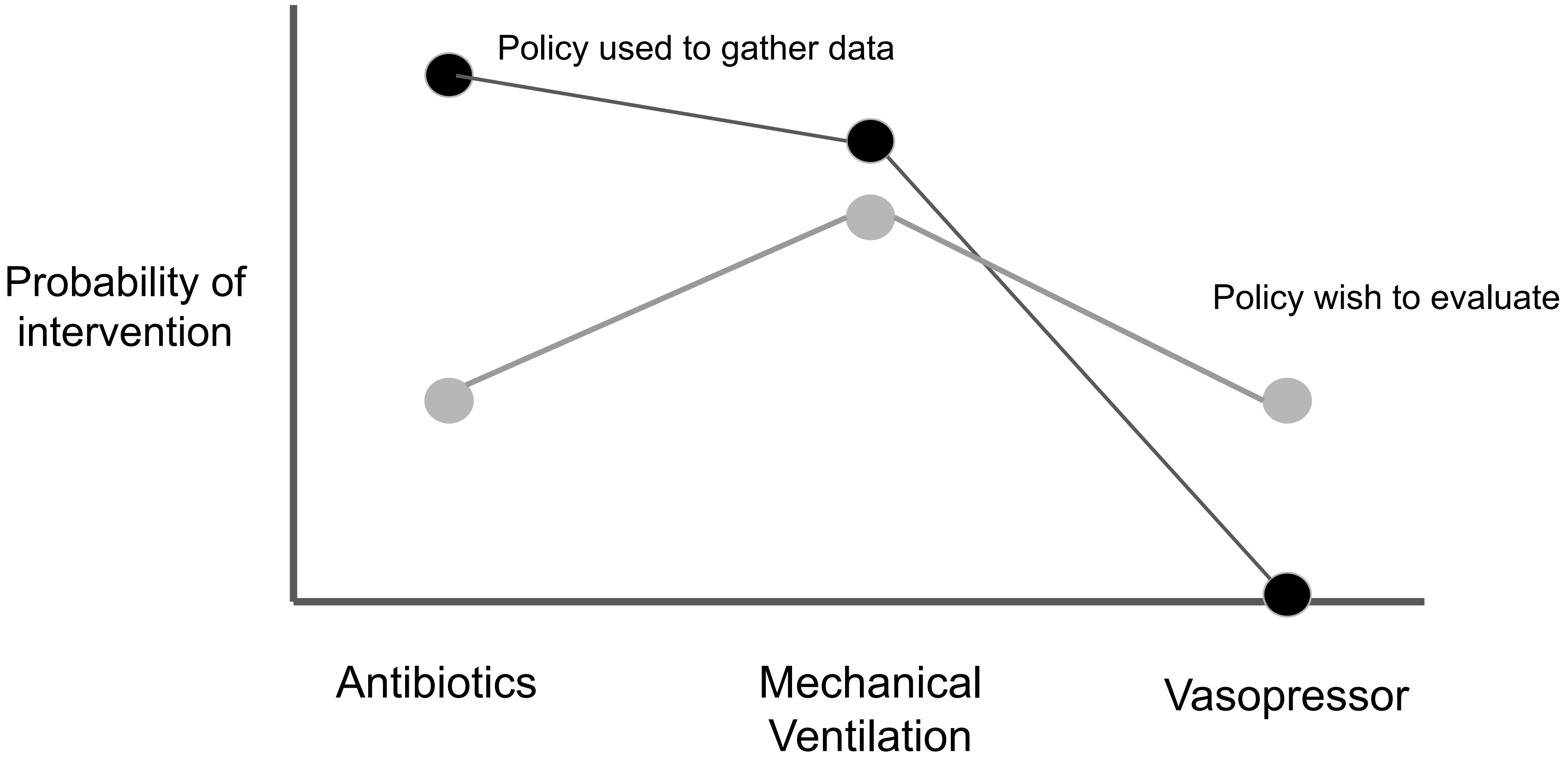


BCQ figure from Fujimoto, Meger, Precup ICML 2019

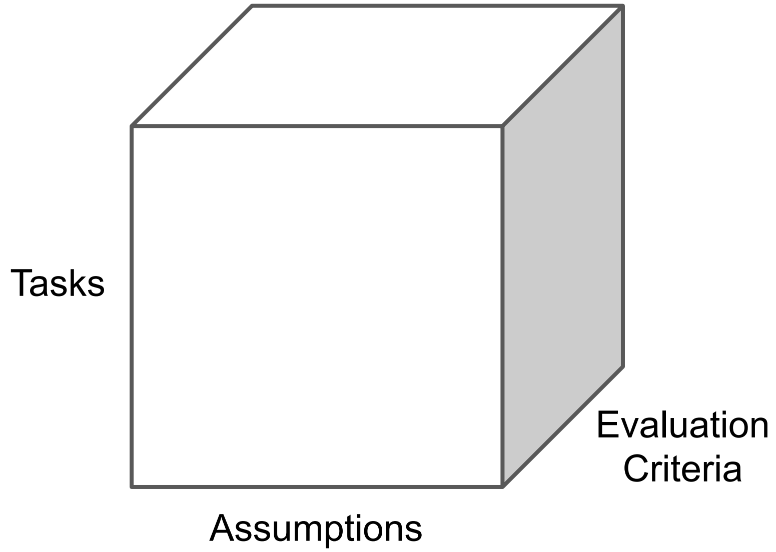
Overlap Requirement: Data Must Support Policy Wish to Evaluate



No Overlap for Vasopressor⇒ Can't Do Off Policy Estimation for Desired Policy



Offline / Batch Reinforcement Learning



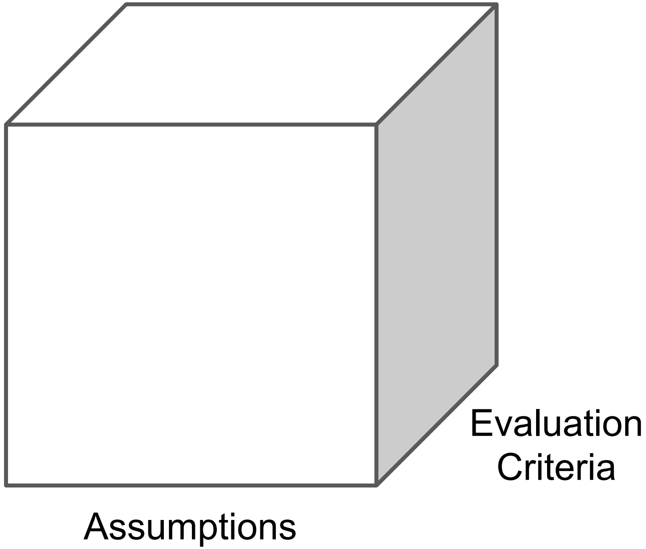
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Common Tasks: Off Policy Evaluation & Optimization

Tasks

$$\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds$$

$$\arg\max_{\pi \in \mathcal{H}_i} \int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds$$



$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \to a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Common Assumptions

- Stationary process: Policy will be evaluated in or deployed in the same stationary decision process as the behavior policy operated in to gather data
- **Markov**
- Sequential ignorability (no confounding)

$$\{Y(A_{1:(t-1)}, a_{t:T}), S_{t'}(A_{1:(t-1)}, a_{t:(t'-1)})\}_{t'=t+1}^{T} \perp\!\!\!\perp A_t \,\Big|\, \mathcal{F}_t$$

- Overlap

$$\forall(s,a)\ \mu_e(s,a) > 0 \quad \rightarrow \mu_b(s,a) > 0$$

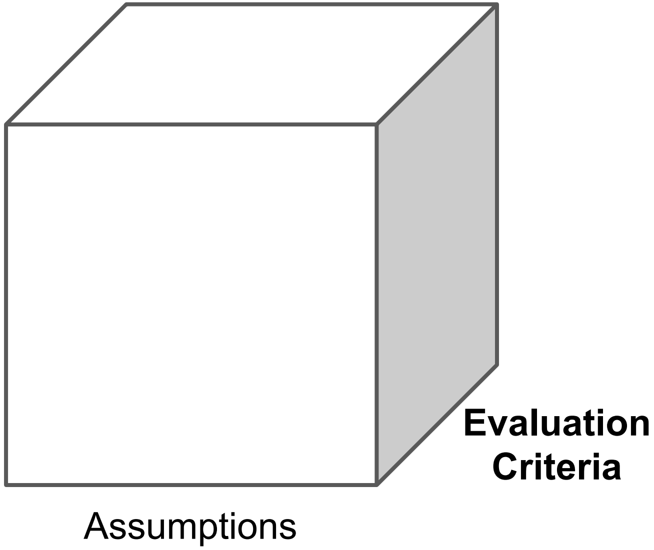
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s,\mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Common Tasks: Off Policy Evaluation & Optimization

Tasks

$$\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D})ds$$

$$\arg\max_{\pi \in \mathcal{H}_i} \int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D})ds$$



$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Off Policy Reinforcement Learning

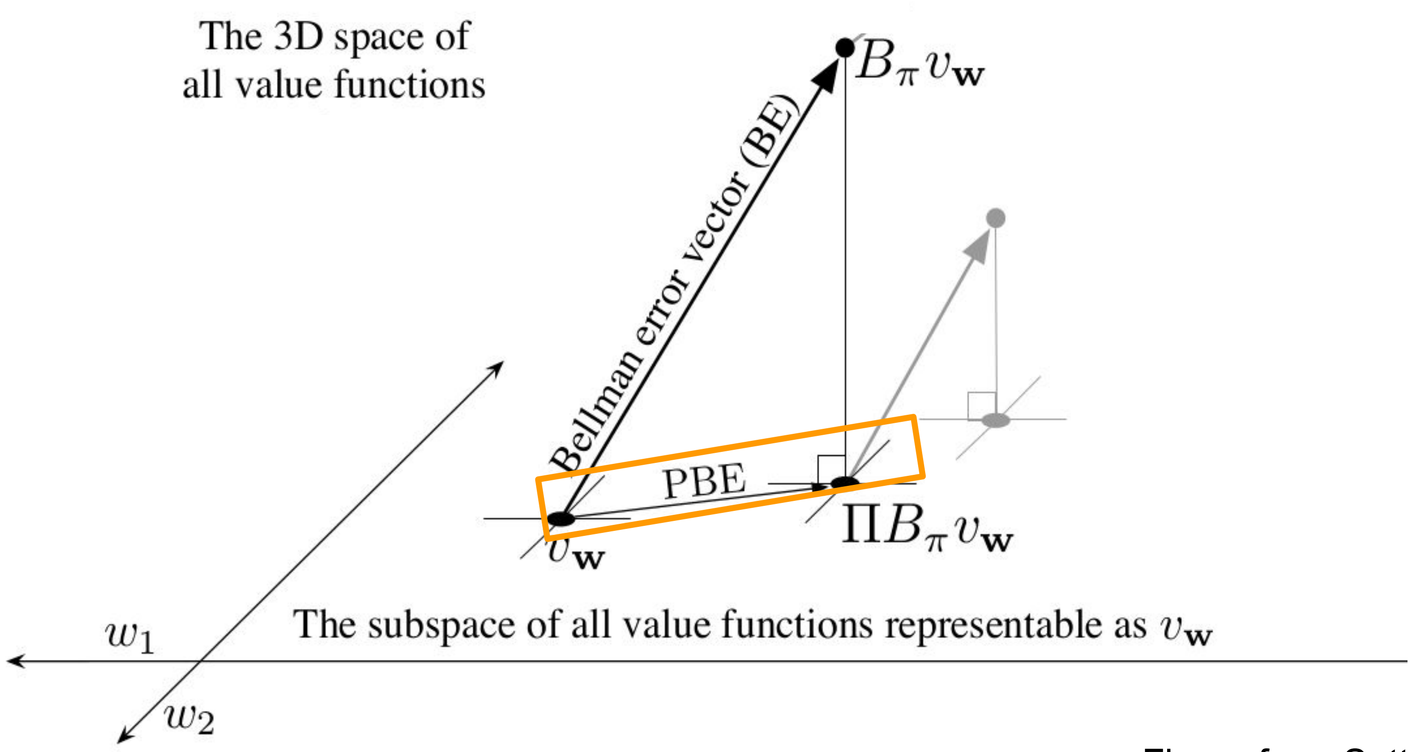
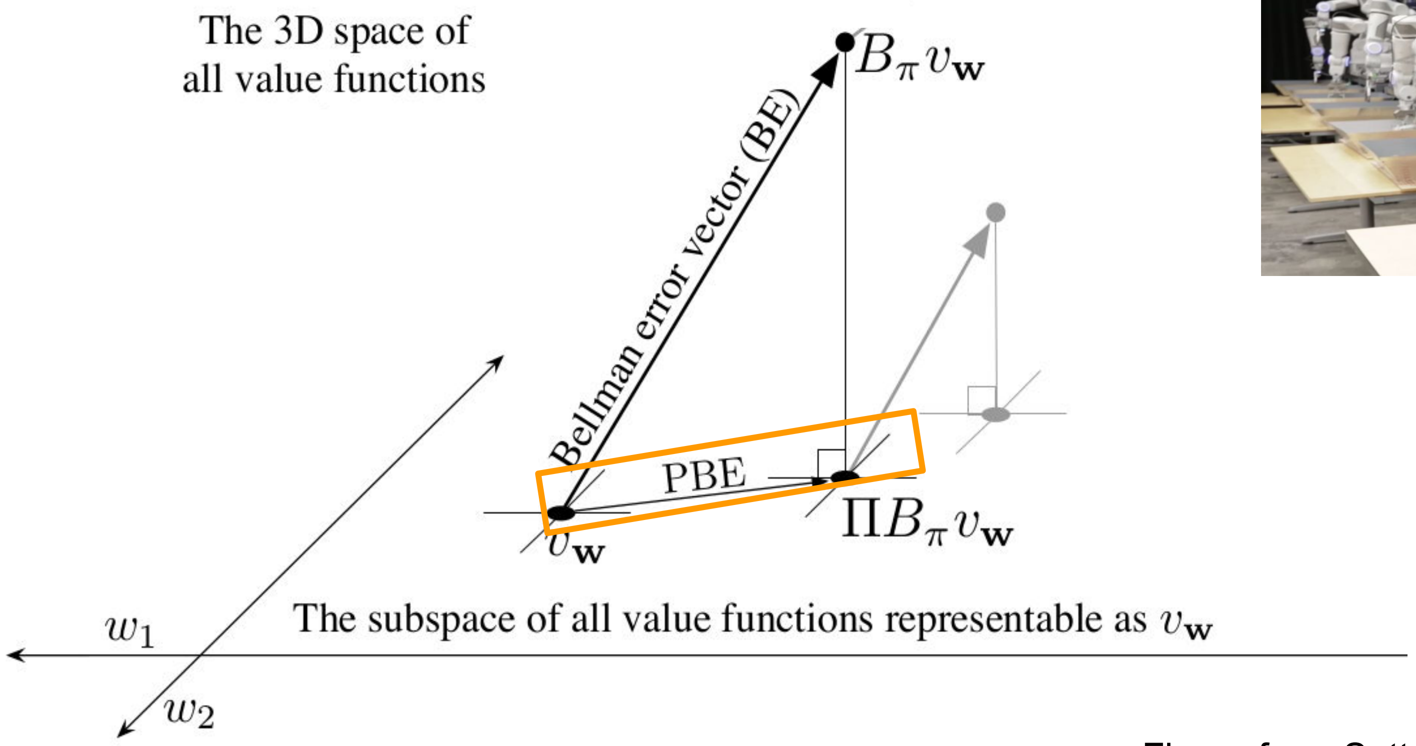


Figure from Sutton & Barto 2018

Off Policy Reinforcement Learning



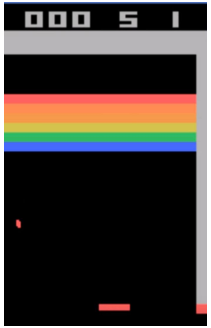


Figure from Sutton & Barto 2018

Batch Off Policy Reinforcement Learning

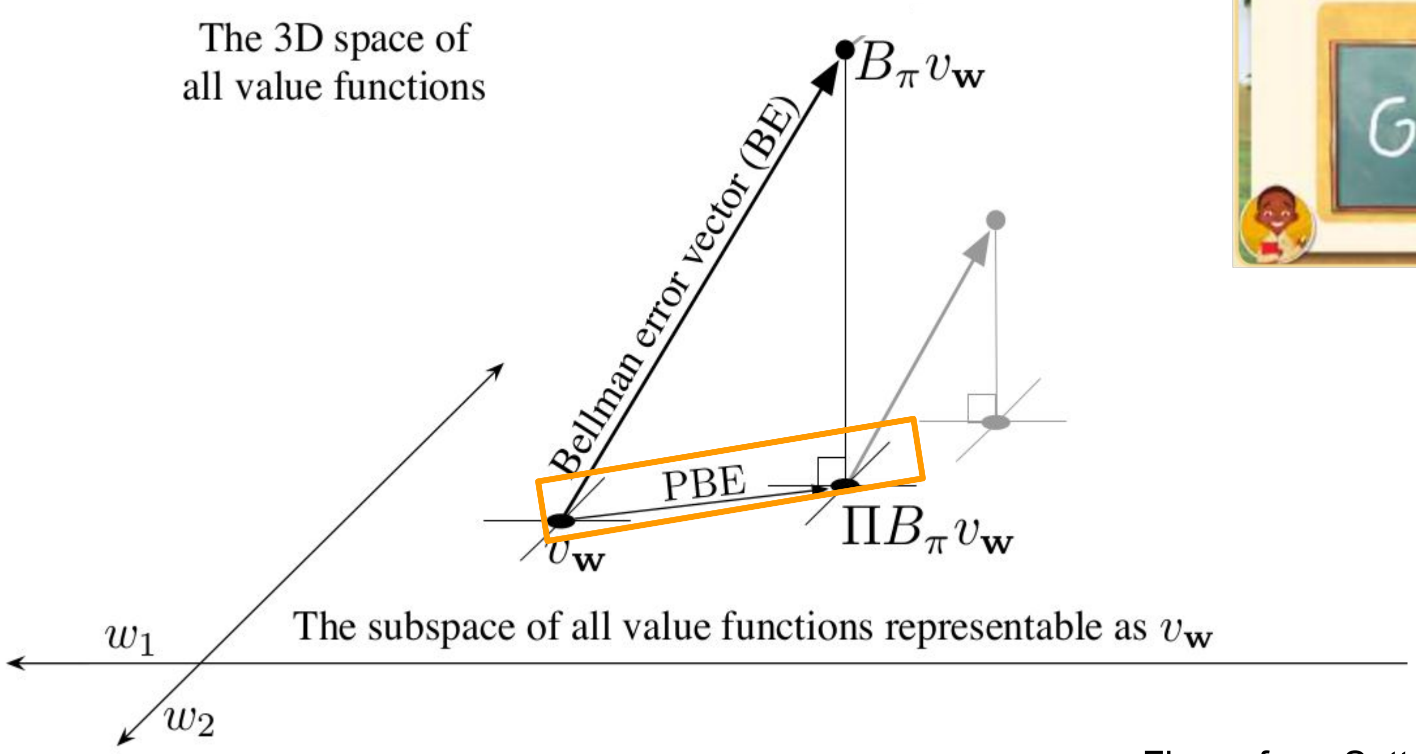


Figure from Sutton & Barto 2018

Batch Off Policy Reinforcement Learning

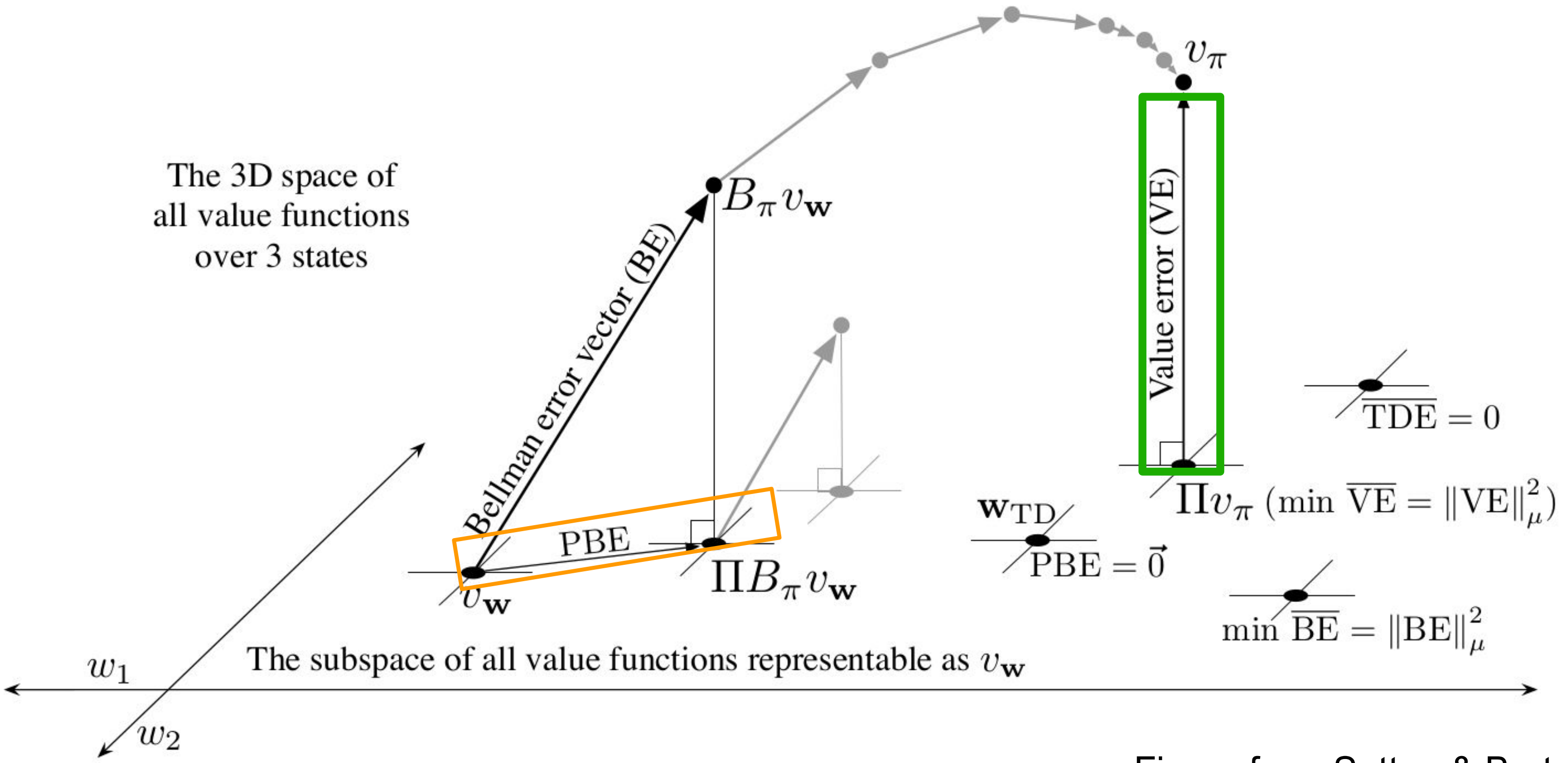


Figure from Sutton & Barto 2018

Common Evaluation Criteria for Off Policy Evaluation

- Computational efficiency
- Performance accuracy

$$\forall \mathcal{D}_i \in \{\mathcal{D}_1 \sim \mathcal{M}_1, \mathcal{D}_2 \sim \mathcal{M}_2, \ldots, \mathcal{D}_K \sim \mathcal{M}_K\} \quad \frac{1}{|\rho|} \sum_{s_0 \in \rho} (\hat{V}^{\pi}_{\mathcal{M}_i}(s_0, \mathcal{D}_i) - V^{\pi}_{\mathcal{M}_i}(s_0))^2$$

$$\lim_{|\mathcal{D}| \to \infty} \frac{1}{|\rho|} \sum_{s_0 \in \rho} \hat{V}^{\pi}(s_0, \mathcal{D}) \to \frac{1}{|\rho|} \sum_{s_0 \in \rho} V^{\pi}(s_0)$$

$$\frac{1}{|\rho|} \sum_{s_0 \in \rho} \hat{V}^{\pi}(s_0, \mathcal{D}) \leq \frac{1}{|\rho|} \sum_{s_0 \in \rho} V^{\pi}(s_0) - f(n, \ldots)$$

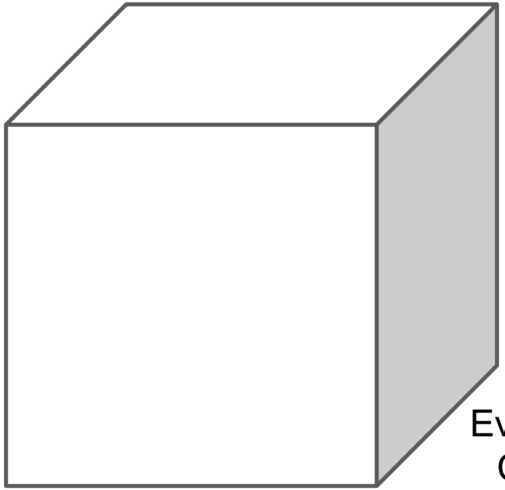
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \to a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Offline / Batch Reinforcement Learning

Tasks

$$\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D})ds$$

$$\arg\max_{\pi \in \mathcal{H}_i} \int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D})ds$$



Evaluation Criteria

- Empirical accuracy
- Consistency
- Robustness
- Asymptotic efficiency
- Finite sample bounds
- Computational cost

Assumptions

- Markov?
- Overlap?
- Sequential ignorability?

$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Batch Policy Optimization: Find a Good Policy That Will Perform Well in the Future

$$\underbrace{\arg\max_{\pi \in \mathcal{H}_i} \quad \max_{\mathcal{H}_i \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}}}_{\text{Policy Optimization}} \quad \underbrace{\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds}_{\text{Policy Evaluation}}$$

$$\mathcal{H} = \mathcal{M}, \mathcal{V}, \mathit{\Pi} \ ?$$

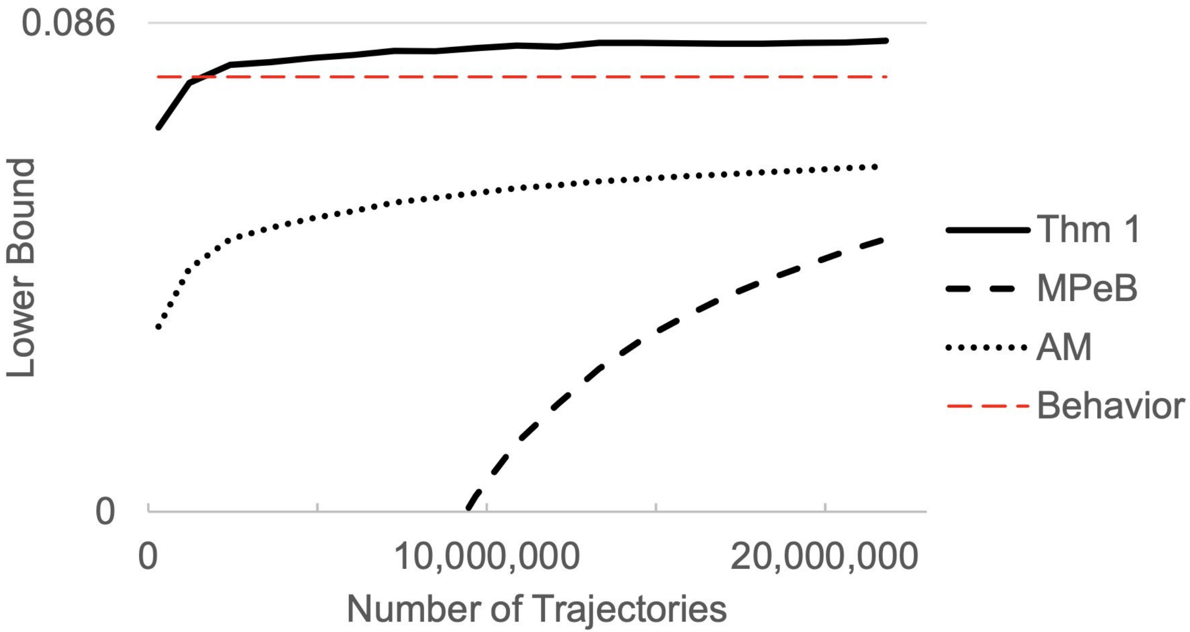
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \to a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Batch Policy Evaluation: Estimate the Performance of a Particular Decision Policy

$$\underbrace{\arg\max_{\pi \in \mathcal{H}_i} \max_{\mathcal{H}_i \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}}}_{\text{Policy Optimization}} \underbrace{\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds}_{\text{Policy Evaluation}}$$

$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Policy Evaluation

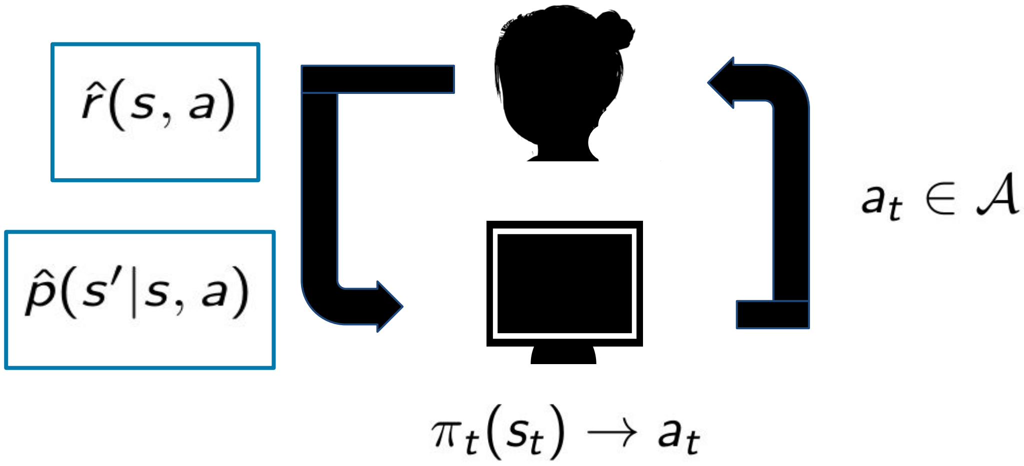


Thomas, Philip, Georgios Theocharous, and Mohammad Ghavamzadeh. "High-confidence off-policy evaluation." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1. 2015.

Outline

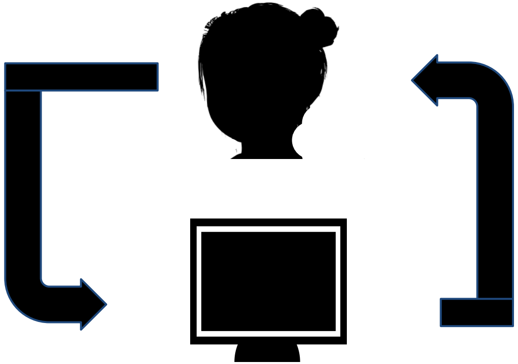
1. Introduction and Setting
2. **Offline batch evaluation using models**
3. Offline batch evaluation using Q functions
4. Offline batch evaluation using importance sampling
5. Safe batch RL

Learn Dynamics and Reward Models from Data



$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \to a$
S_0: Set of initial states
$\hat{V}^\pi(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Learn Dynamics and Reward Models from Data, Evaluate Policy



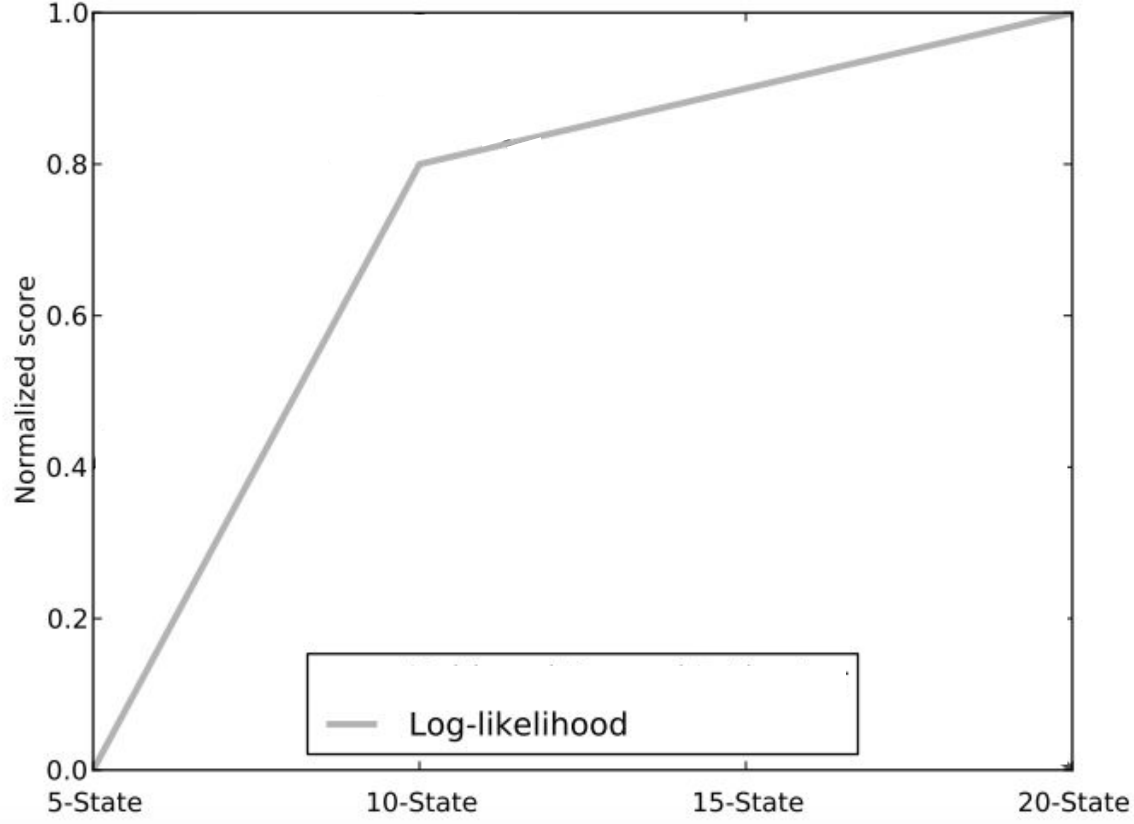
$$V^{\pi} \approx (I - \gamma \hat{P}^{\pi})^{-1} \hat{R}^{\pi}$$

$$P^{\pi}(s'|s) = p(s'|s, \pi(s))$$

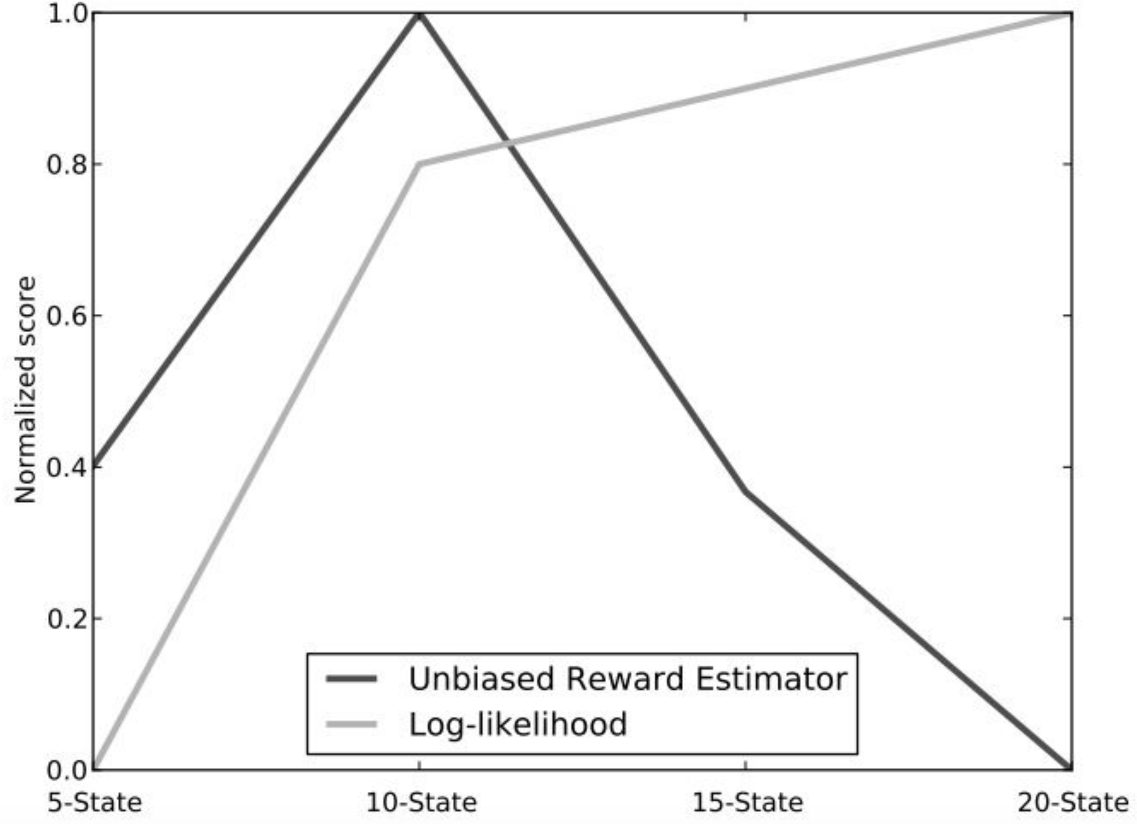
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

- Mannor, Simster, Sun, Tsitsiklis 2007

Better Dynamics/Reward Models for Existing Data (Improve likelihood)



Better Dynamics/Reward Models for Existing Data, May **Not** Lead to Better Policies for Future Use → Bias due to Model **Misspecification**



Mandel, Liu, Brunskill, Popovic AAMAS 2014

Model Free Value Function Approximation: Fitted Q Evaluation

$$\mathcal{D} = (s_i, a_i, r_i, s_{i+1}) \ \forall i$$

$$\tilde{Q}^\pi(s_i, a_i) \quad = \quad r_i + \gamma V_\theta^\pi(s_{i+1})$$

$$\arg\min_\theta \sum_i (Q_\theta^\pi(s_i, a_i) - \tilde{Q}^\pi(s_i, a_i))^2$$

$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \to a$
S_0: Set of initial states
$\hat{V}^\pi(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

- Fitted Q evaluation, LSTD, ...

Algorithm 3 Fitted Q Evaluation: $\texttt{FQE}(\pi, c)$

Input: Dataset $\mathrm{D} = \{x_i, a_i, x'_i, c_i\}_{i=1}^n \sim \pi_{\mathrm{D}}$. Function class F.
Policy π to be evaluated

1: Initialize $Q_0 \in \mathrm{F}$ randomly
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Compute target $y_i = c_i + \gamma Q_{k-1}(x'_i, \pi(x'_i)) \ \forall i$
4: Build training set $\widetilde{\mathrm{D}}_k = \{(x_i, a_i), y_i\}_{i=1}^n$
5: Solve a supervised learning problem:
$$Q_k = \arg\min_{f \in \mathrm{F}} \frac{1}{n} \sum_{i=1}^n (f(x_i, a_i) - y_i)^2$$
6: **end for**

Output: $\widehat{C}^\pi(x) = Q_K(x, \pi(x)) \quad \forall x$

Let's assume we use a DNN for F.

What is different vs DQN?

Le, H., Voloshin, C., & Yue, Y. (2019, May). Batch policy learning under constraints. In *International Conference on Machine Learning*

Model Free Policy Evaluation

- Challenge: still relies on Markov assumption
- Challenge: still relies on models being well specified or have no computable guarantees if there is misspecification

$$d_F^\pi = \sup_{g \in F} \inf_{f \in F} ||f - B^\pi g||_\pi$$

Batch Policy Optimization: Find a Good Policy That Will Perform Well in the Future

$$\underbrace{\arg\max_{\pi \in \mathcal{H}_i} \max_{\mathcal{H}_i \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}}}_{\text{Policy Optimization}} \underbrace{\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds}_{\text{Policy Evaluation}}$$

$$\mathcal{H} = \mathcal{M}, \mathcal{V}, \Pi \text{ ?}$$

$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

- Today will not be a comprehensive overview, but instead highlight some of the challenges involved & some approaches with desirable statistical properties convergence, sample efficiency & bounds

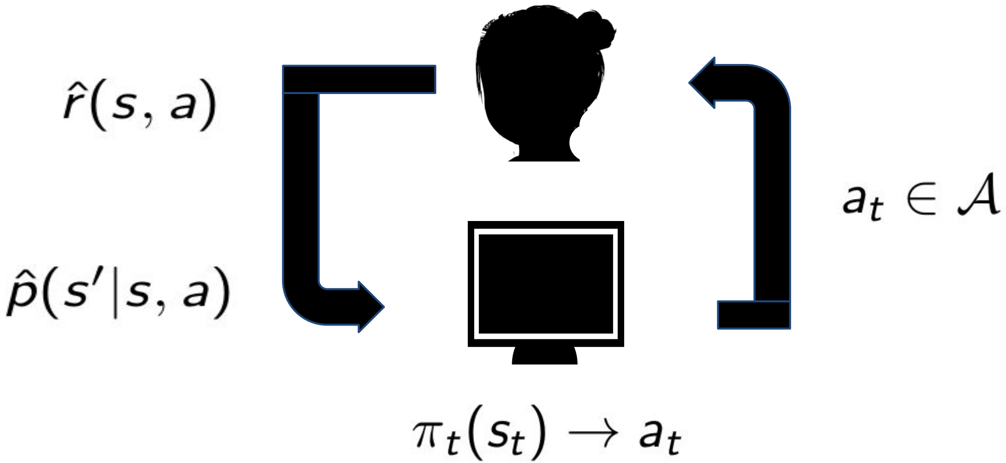
Policy Optimization: Find Good Policy to Deploy

$$\arg\max_{\pi \in \mathcal{H}_i} \max_{\mathcal{H}_i \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}} \int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds$$

$\mathcal{H} = \mathcal{M}, \mathcal{V}, \Pi$?

$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Learn Dynamics and Reward Models from Data, Plan



$$\hat{V}^*(s) = \max_a \hat{r}(s, a) + \gamma \sum_{s'} \hat{p}(s'|s, a)\hat{V}^*(s')$$

Model Free Value Function Approximation: Fitted Q Iteration

$$\mathcal{D} = (s_i, a_i, r_i, s_{i+1}) \ \forall i$$

$$(\mathcal{T} f)(s, a) := R(s, a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V_f(s')]$$

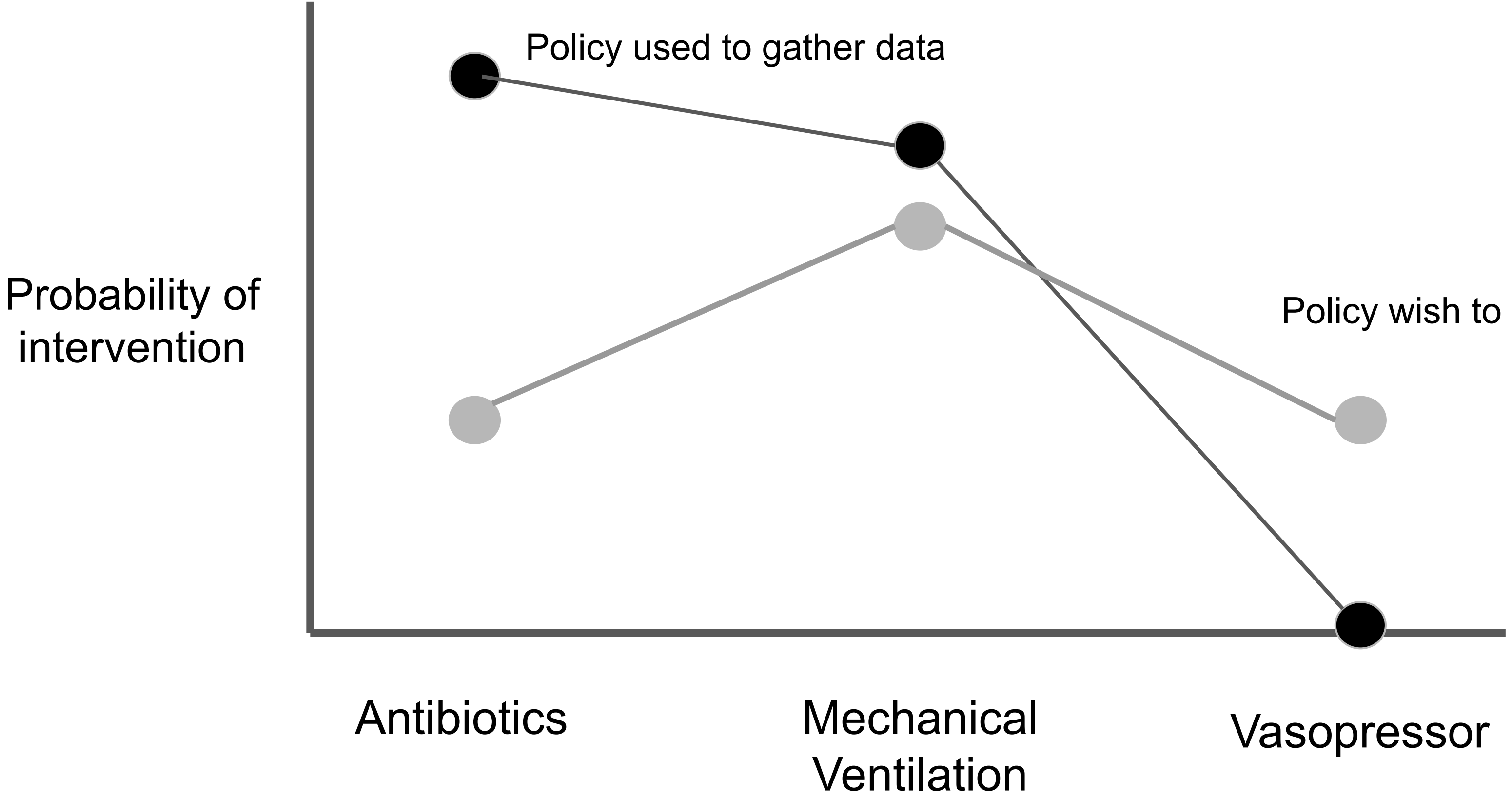
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^\pi(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Standard Assumptions for Off Policy / Counterfactual Estimation & Optimization

- Overlap
  - Have to take all actions that target policy would take
  - In infinite data / finite data
- No confounding

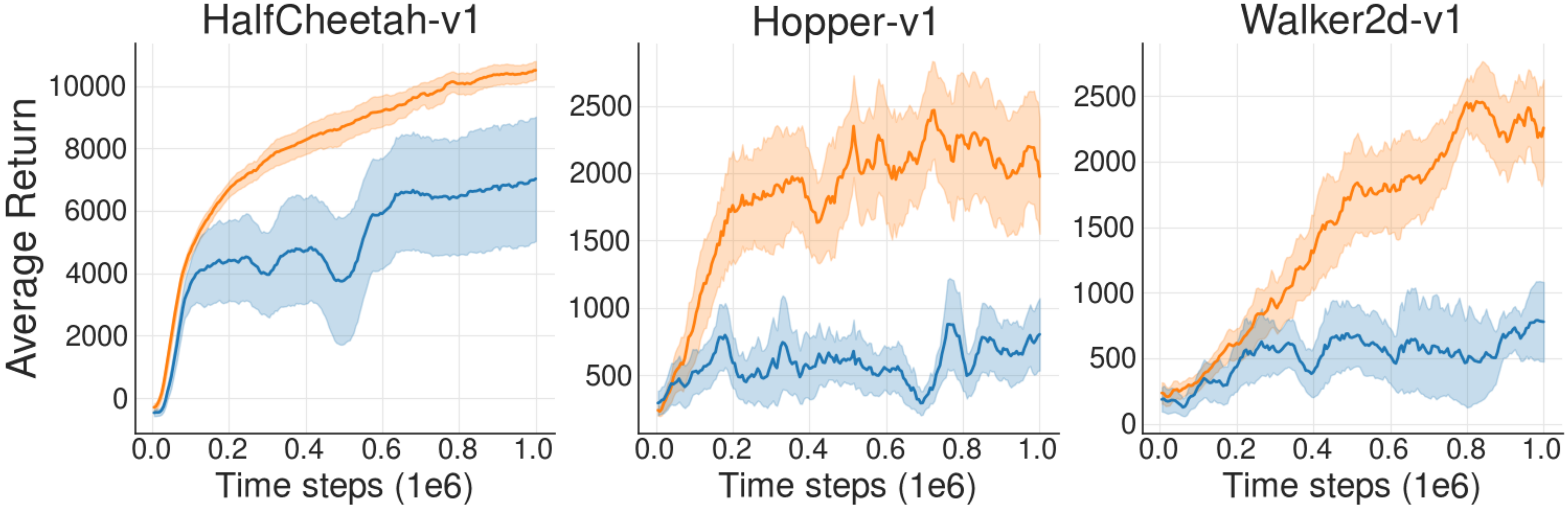
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^\pi(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

No Overlap for Vasopressor⇒ Can't Do Off Policy Estimation for Desired Policy



Limitations of Prior Work

- Typically assume overlap
  - Off policy estimation: for policy of interest
  - Off policy optimization: for all policies including optimal one (see concentrability assumption in batch RL)
- Unlikely to be true in many settings
- Many real datasets don’t include complete random exploration
- Assuming overlap when it’s not there can be a problem:
  - We can end up with a policy with estimated high performance, but actually does poorly when deployed


HalfCheetah-v1
Hopper-v1
Walker2d-v1
Average Return
10000
8000
6000
4000
2000
0
2500
2000
1500
1000
500
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
Time steps (1e6)
Time steps (1e6)
Time steps (1e6)

Surprise!

Agent orange and agent blue are trained with…

1. The **same off-policy algorithm** (DDPG).

2. The **same dataset.**

The Difference?

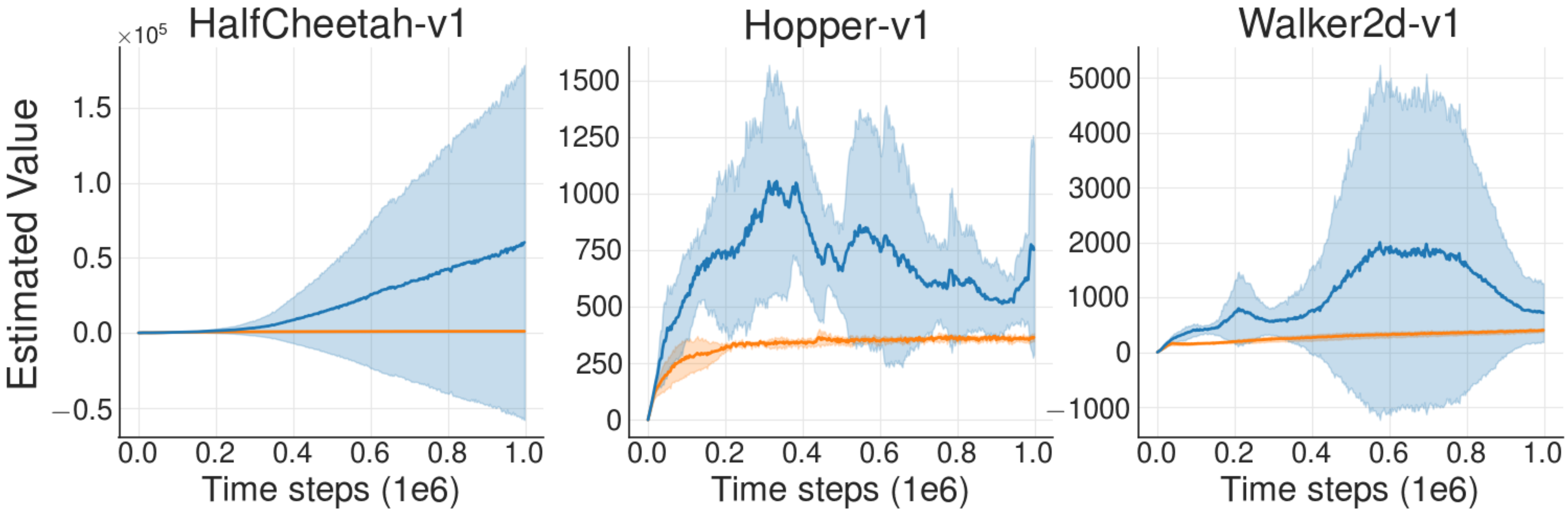
1. Agent orange: Interacted with the environment.
    - Standard RL loop.
    - Collect data, store data in buffer, train, repeat.

2. Agent blue: Never interacted with the environment.
    - Trained with data collected by agent orange concurrently.

1. Trained with the same off-policy algorithm.
2. Trained with the same dataset.
3. One interacts with the environment. One doesn’t.

Off-policy deep RL fails when **truly off-policy**.

Value Predictions



Extrapolation Error

$$Q(s,a) \leftarrow r + \gamma Q(s', a')$$

Extrapolation Error

$$Q(s, a) \leftarrow r + \gamma Q(s', a')$$

GIVEN

GENERATED

Extrapolation Error

$$Q(s,a) \leftarrow r + \gamma Q(s',a')$$

1. $(s,a,r,s') \sim Dataset$
2. $a' \sim \pi(s')$

Extrapolation Error

$$Q(s,a) \leftarrow r + \gamma Q(s',a')$$

$$(s',a') \notin Dataset \rightarrow Q(s',a') = \textbf{bad}$$

$$\rightarrow Q(s,a) \quad = \textbf{bad}$$

Extrapolation Error

$$Q(s,a) \leftarrow r + \gamma Q(s',a')$$

$$(s',a') \notin Dataset \rightarrow Q(s',a') = \mathbf{bad}$$

$$\rightarrow Q(s,a) \quad = \mathbf{bad}$$

Extrapolation Error

$$Q(s,a) \leftarrow r + \gamma Q(s',a')$$

$$(s',a') \notin Dataset \rightarrow Q(s',a') = \mathbf{bad}$$

$$\rightarrow Q(s,a) \quad = \mathbf{bad}$$

Extrapolation Error

Attempting to evaluate π without (sufficient) access to the (s, a) pairs π visits.

Batch-Constrained Reinforcement Learning

Only choose π such that we have access to the (s, a) pairs π visits.

Batch-Constrained Reinforcement Learning

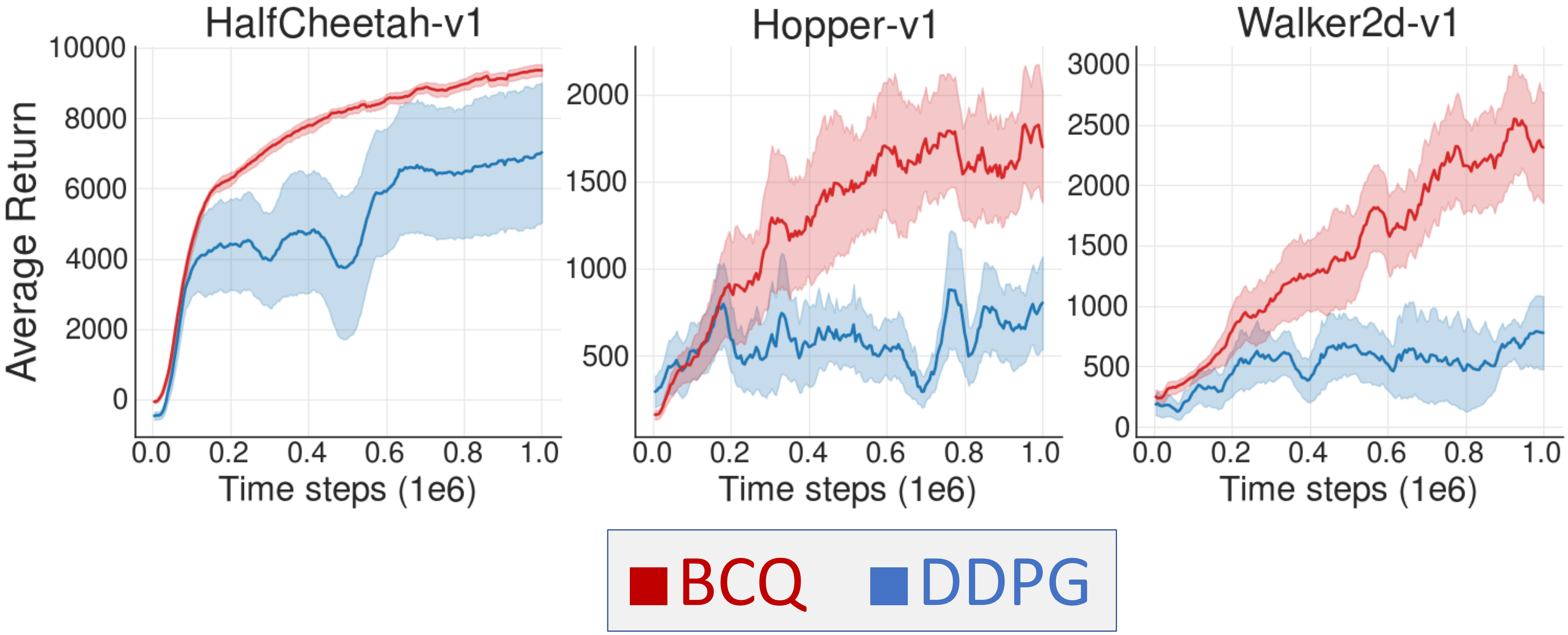
1. a$\sim\pi(s)$ such that $(s, a) \in Dataset$.
2. a$\sim\pi(s)$ such that $\left(s', \pi(s')\right) \in Dataset$.
3. a$\sim\pi(s)$ such that $Q(s, a)$ is maxed.

Batch-Constrained Deep Q-Learning (BCQ)

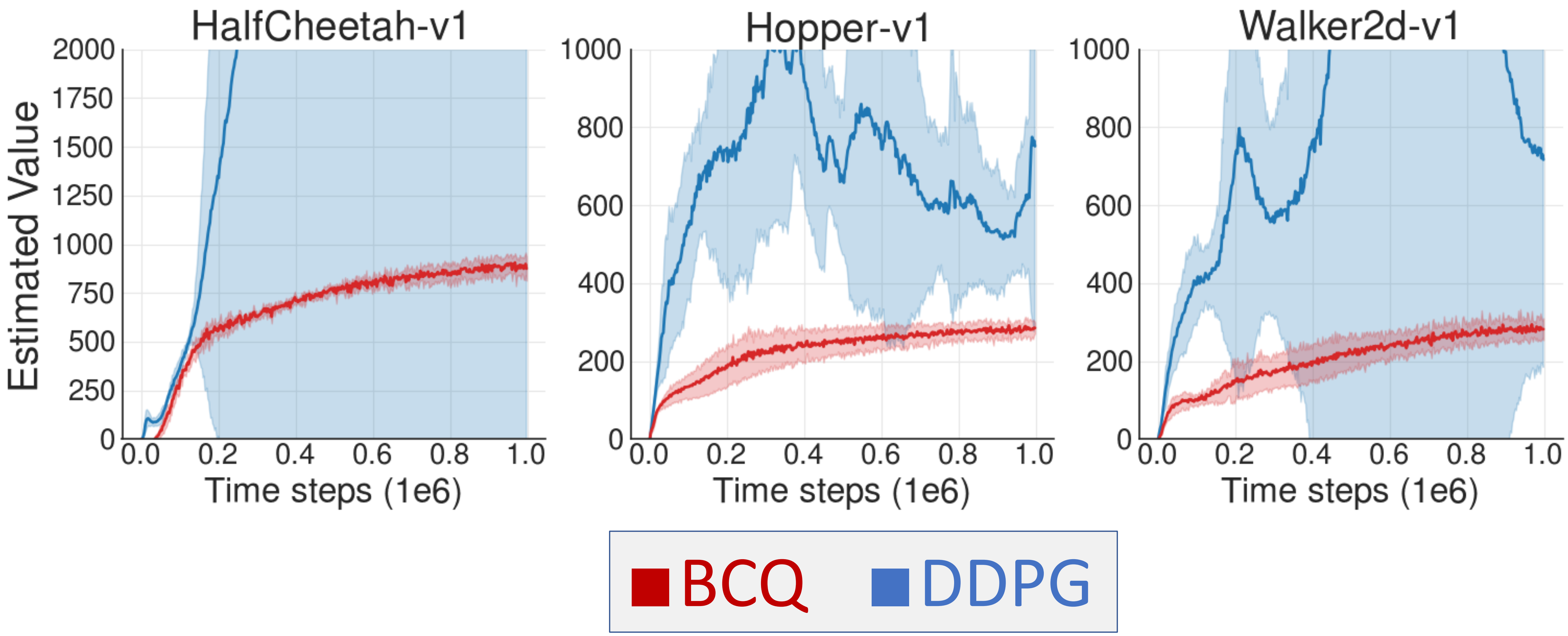
First imitate dataset via generative model:
$G(a|s) \approx P_{Dataset}(a|s)$.

$\pi(s) = \text{argmax}_{a_i} Q\,(s, a_i)$, where $a_i \sim G$
(I.e. select the best action that is likely under the dataset)

(+ some additional deep RL magic)



HalfCheetah-v1
Hopper-v1
Walker2d-v1
Average Return
10000
8000
6000
4000
2000
0
2000
1500
1000
500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
Time steps (1e6)
Time steps (1e6)
Time steps (1e6)
BCQ
DDPG


HalfCheetah-v1
Hopper-v1
Walker2d-v1
Estimated Value
2000
1750
1500
1250
1000
750
500
250
0
800
600
400
200
0.0
0.2
0.4
0.6
0.8
1.0
Time steps (1e6)
BCQ
DDPG