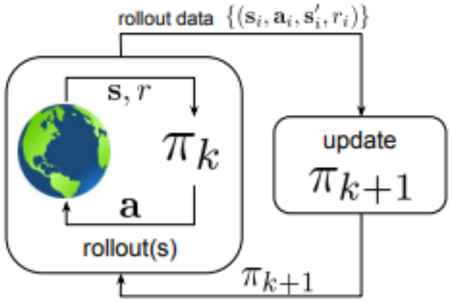
Batch / Offline Reinforcement Learning

With thanks to Emma Brunskill, Scott Fujimoto, Pieter Abeell, George Tucker, Sergey Levine, Bilal Piot, Yuxin Chen, Yuejie Chi

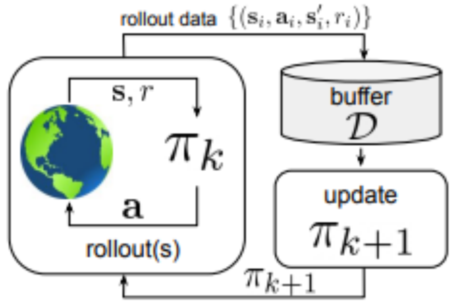
COMP579 Lecture 21, 2024

On-policy vs off-policy vs offline RL

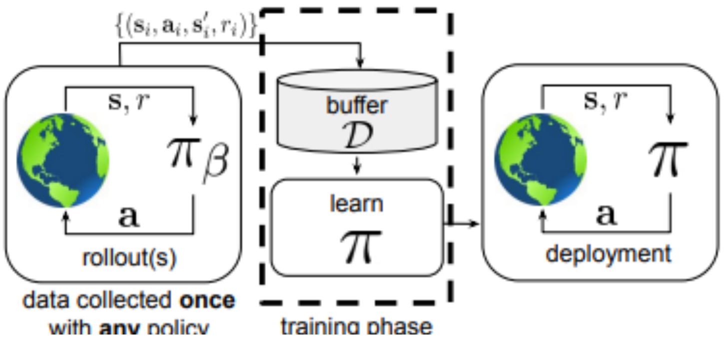
on-policy RL



off-policy RL



offline reinforcement learning



Formally:

$$\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i)\}$$

$\mathbf{s} \sim d^{\pi_\beta}(\mathbf{s})$

$\mathbf{a} \sim \pi_\beta(\mathbf{a}|\mathbf{s})$ ← generally **not** known

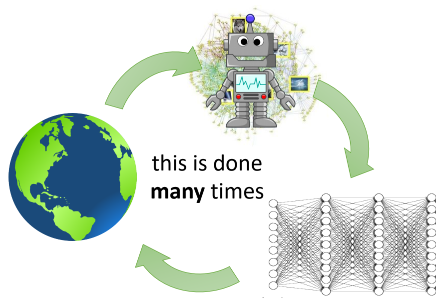
$\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$

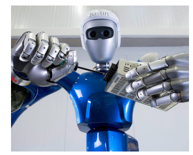
$r \leftarrow r(\mathbf{s}, \mathbf{a})$

RL objective: $\max_\pi \sum_{t=0}^{T} E_{\mathbf{s}_t \sim d^\pi(\mathbf{s}), \mathbf{a}_t \sim \pi(\mathbf{a}|\mathbf{s})}[\gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$

Why is this important?

- Collecting new data may be expensive / infeasible
- We may have access to existing/historical data instead





Problem formulation

A historical dataset $\mathcal{D} = \left\{(s^{(i)}, a^{(i)}, s'^{(i)})\right\}$**:** N independent copies of

$$s \sim \rho^{\mathsf{b}}, \qquad a \sim \pi^{\mathsf{b}}(\cdot \,|\, s), \qquad s' \sim P(\cdot \,|\, s, a)$$

for some state distribution ρ^{b} and behavior policy π^{b}

Goal: given some test distribution ρ and accuracy level ε, find an ε-optimal policy $\widehat{\pi}$ based on $\mathcal{D}$ obeying

$$V^{\star}(\rho) - V^{\widehat{\pi}}(\rho) = \mathop{\mathbb{E}}_{s\sim\rho}\left[V^{\star}(s)\right] - \mathop{\mathbb{E}}_{s\sim\rho}\left[V^{\widehat{\pi}}(s)\right] \leq \varepsilon$$

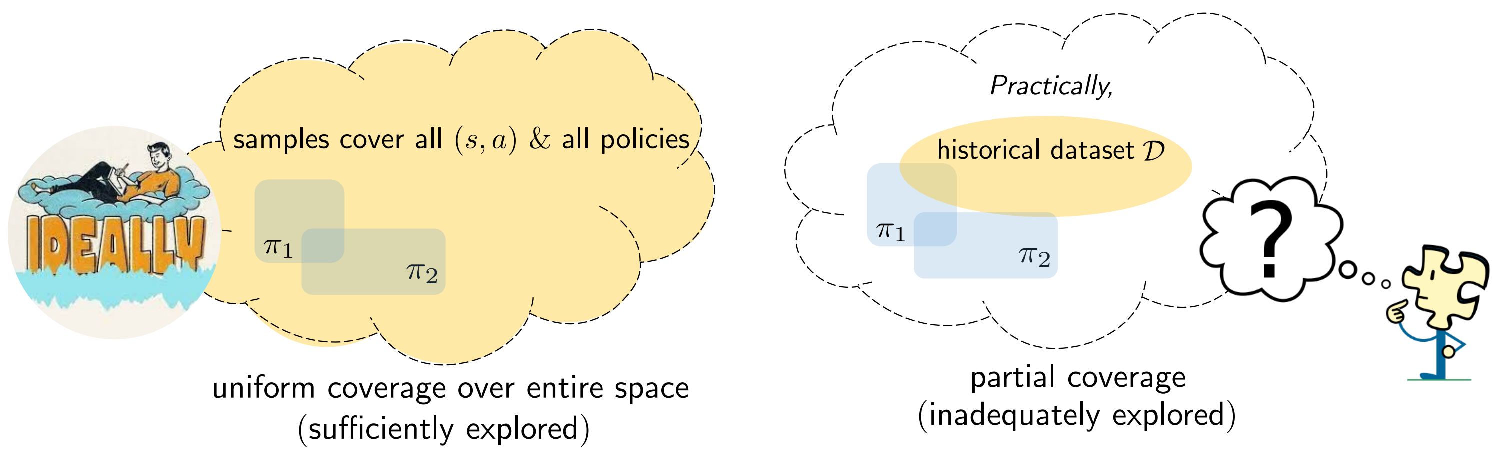
— *in a sample-efficient manner*

Challenges of offline / batch RL (1)

- **Distribution shift**:

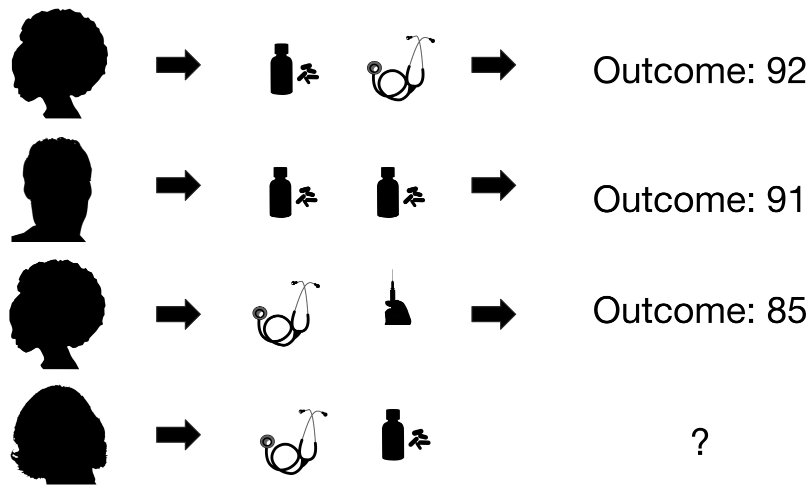
$$\text{distribution}(\mathcal{D}) \neq \text{target distribution under } \pi^\star$$

- **Partial coverage of state-action space**:



Challenges of offline / batch RL (2)

- Data is *censored*: we only observe outcomes for decisions made (and need to generalize from them)



- Need for *counterfactual inference*: what would happen if one would take a different action?
- Often we do not observe rewards, just states and actions!

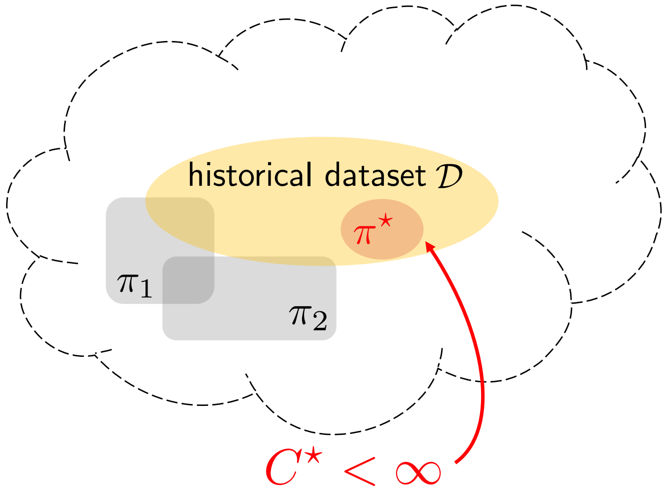
Dataset quality assessment

Single-policy concentrability coefficient

$$C^\star := \max_{s,a} \frac{d^{\pi^\star}(s,a)}{d^{\pi^\mathsf{b}}(s,a)} = \left\| \frac{\textit{occupancy density of } \pi^\star}{\textit{occupancy density of } \pi^\mathsf{b}} \right\|_\infty \geq 1$$

where $d^\pi(s,a) = (1-\gamma)\sum_{t=0}^\infty \gamma^t \mathbb{P}\big((s^t,a^t)=(s,a)\,|\,\pi\big)$

- captures distributional shift
- allows for partial coverage

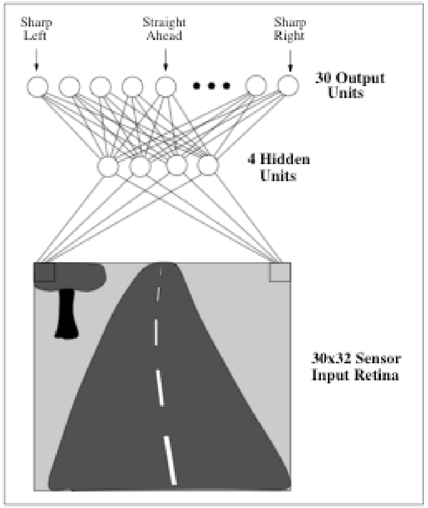


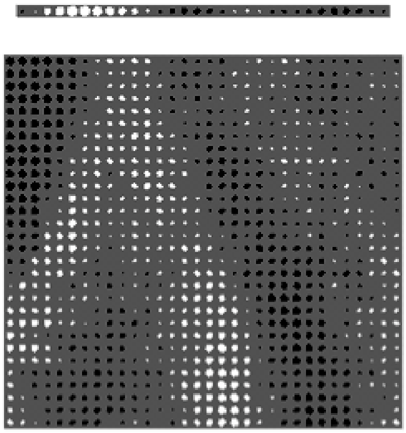
Classes of algorithms

- Behavior cloning (no rewards required)
- Learn a model, use it for model-based RL (LSTD, LSPI)
- Pessimistic algorithms (require rewards)
- Inverse RL (learn reward function from data, use it for RL agent)

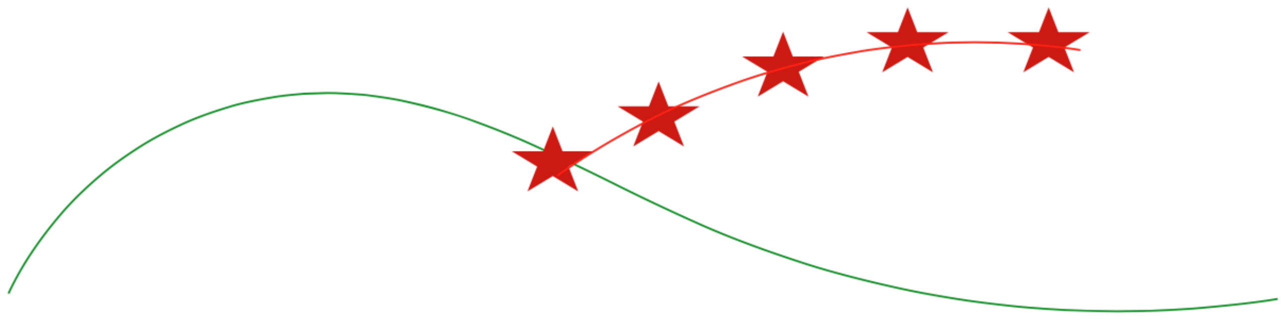
Behavior cloning

- Take dataset $\mathcal{D}$, learn a policy from states to actions
- Often uses a rich policy class (neural net)





Problem: compounding errors



- Error at time t with probability ϵ
- Approximate intuition: $\mathbb{E}[\text{Total errors}]$
  $\leq \epsilon(T + (T-1) + (T-2) \ldots + 1) \propto \epsilon T^2$

One solution: dataset aggregation

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

- Idea: Get more labels of the expert action along the path taken by the policy computed by behavior cloning
- Obtains a stationary deterministic policy with good performance under its induced state distribution

Pessimism in the face of uncertainty

- *Conservative* approach
- Assume that states or state-action pairs not visited are bad
- Use a penalty to avoid the new policy visiting them

Value iteration with lower confidence bounds

Pessimism in the face of uncertainty: penalize value estimate of those (s, a) pairs that were poorly visited [Jin et al., 2021, Rashidinejad et al., 2021]

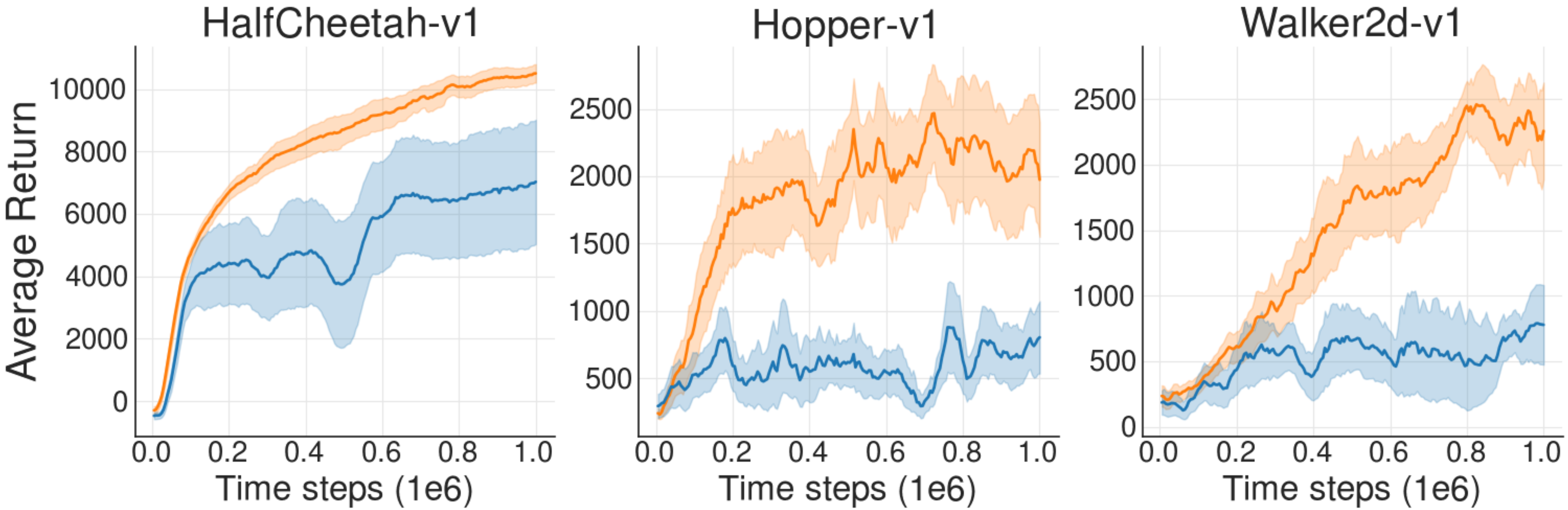
Algorithm: value iteration w/ lower confidence bounds

- compute empirical estimate $\widehat{P}$ of P
- initialize $\widehat{Q} = 0$, and repeat

$$\widehat{Q}(s,a) \;\leftarrow\; \max\Big\{ r(s,a) + \gamma \big\langle \widehat{P}(\cdot \,|\, s,a), \widehat{V} \big\rangle - \underbrace{b(s,a;\widehat{V})}_{\text{Bernstein-style confidence bound}}\,,\; 0 \Big\}$$

for all (s, a), where $\widehat{V}(s) = \max_a \widehat{Q}(s,a)$

Q-learning version exists as well



HalfCheetah-v1
Hopper-v1
Walker2d-v1
Average Return
10000
8000
6000
4000
2000
0
2500
2000
1500
1000
500
0.0
0.2
0.4
0.6
0.8
1.0
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?

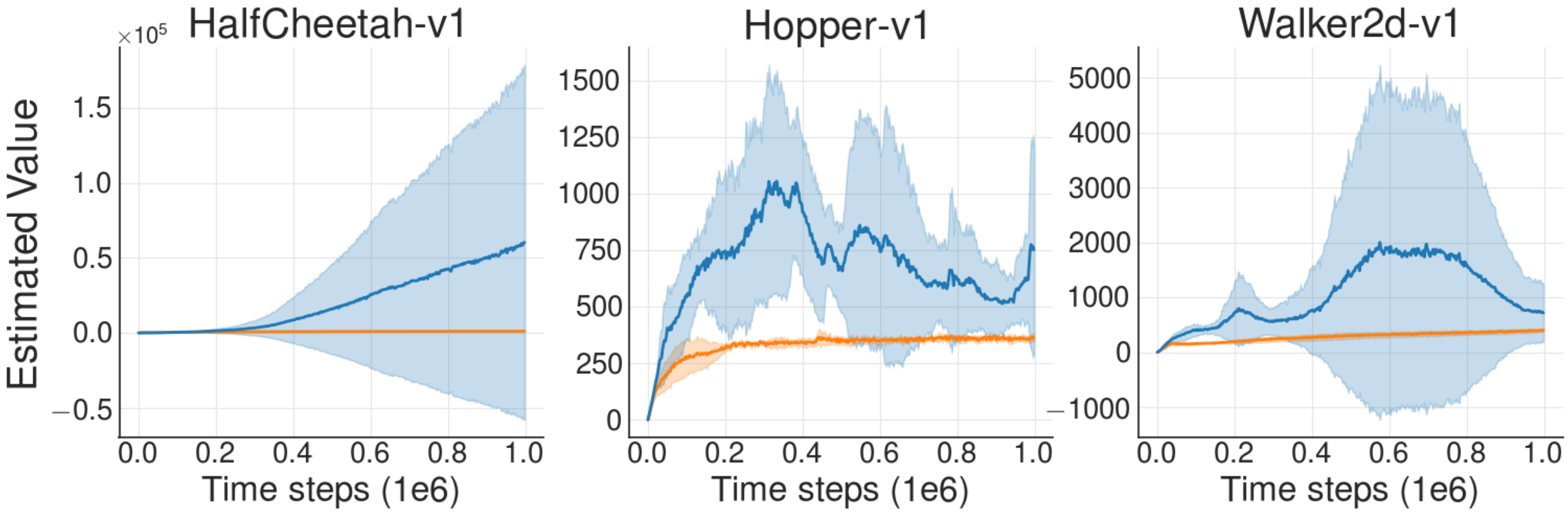
1. Agent orange: Interacted with the environment.
    - Standard RL loop.
    - Collect data, store data in buffer, train, repeat.

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

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

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

Extrapolation Error

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

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

Extrapolation Error

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

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

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

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

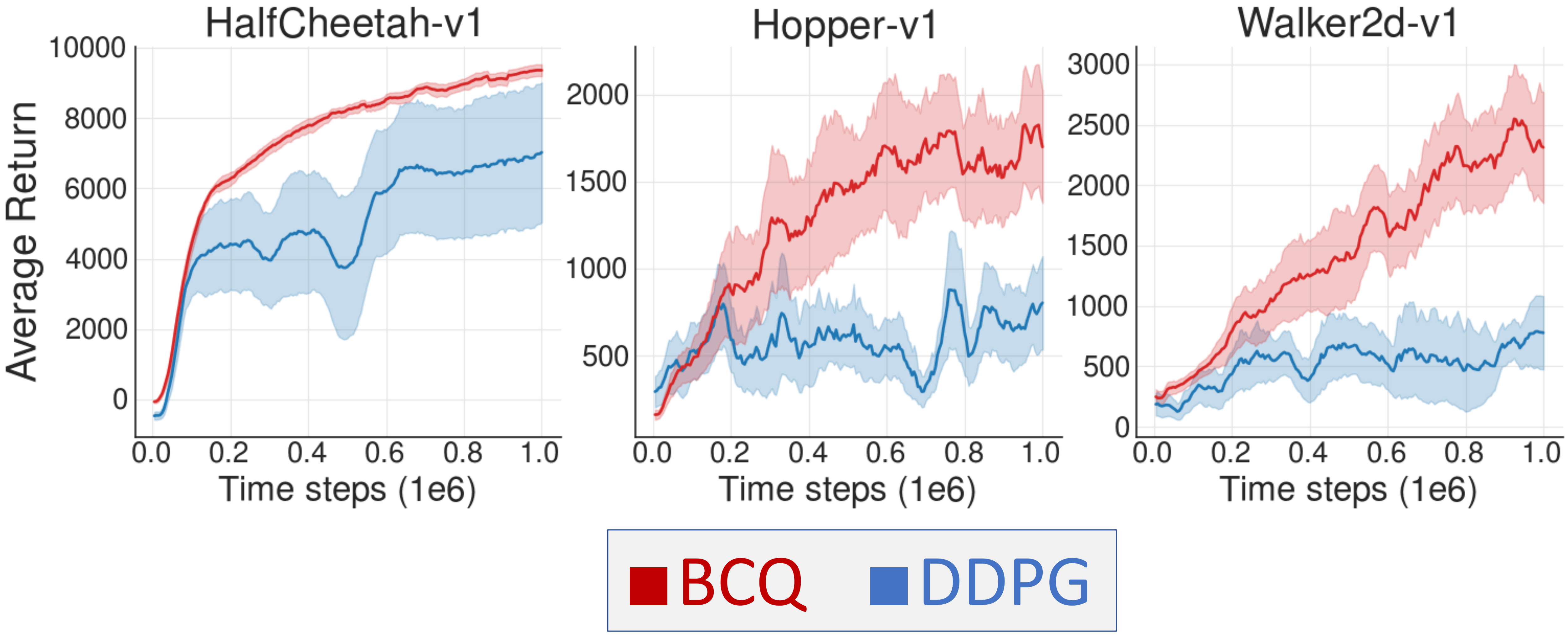
1. a$\sim\pi(s)$ such that $(s, a) \in Dataset$.
2. a$\sim\pi(s)$ such that $\left(s', \pi(s')\right) \in Dataset$.
3. a$\sim\pi(s)$ such that $Q(s, a)$ is maxed.

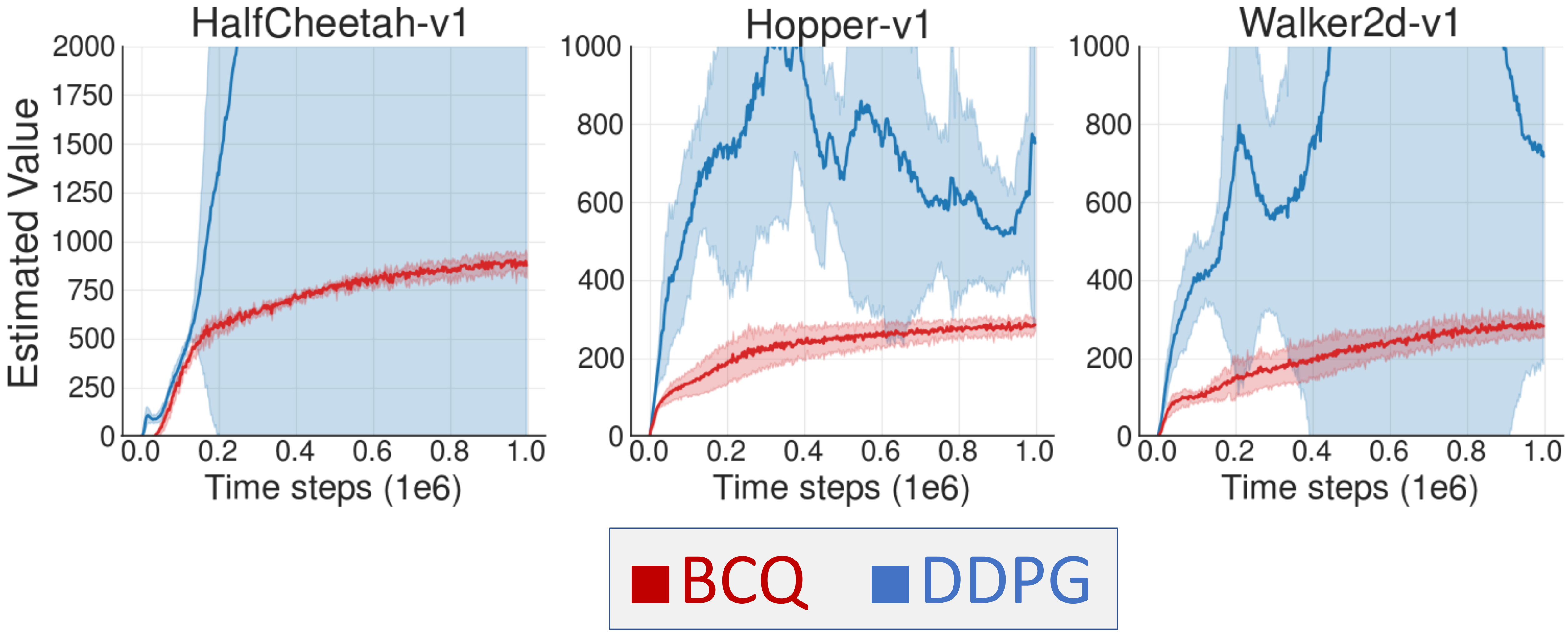
Batch-Constrained Deep Q-Learning (BCQ)

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)

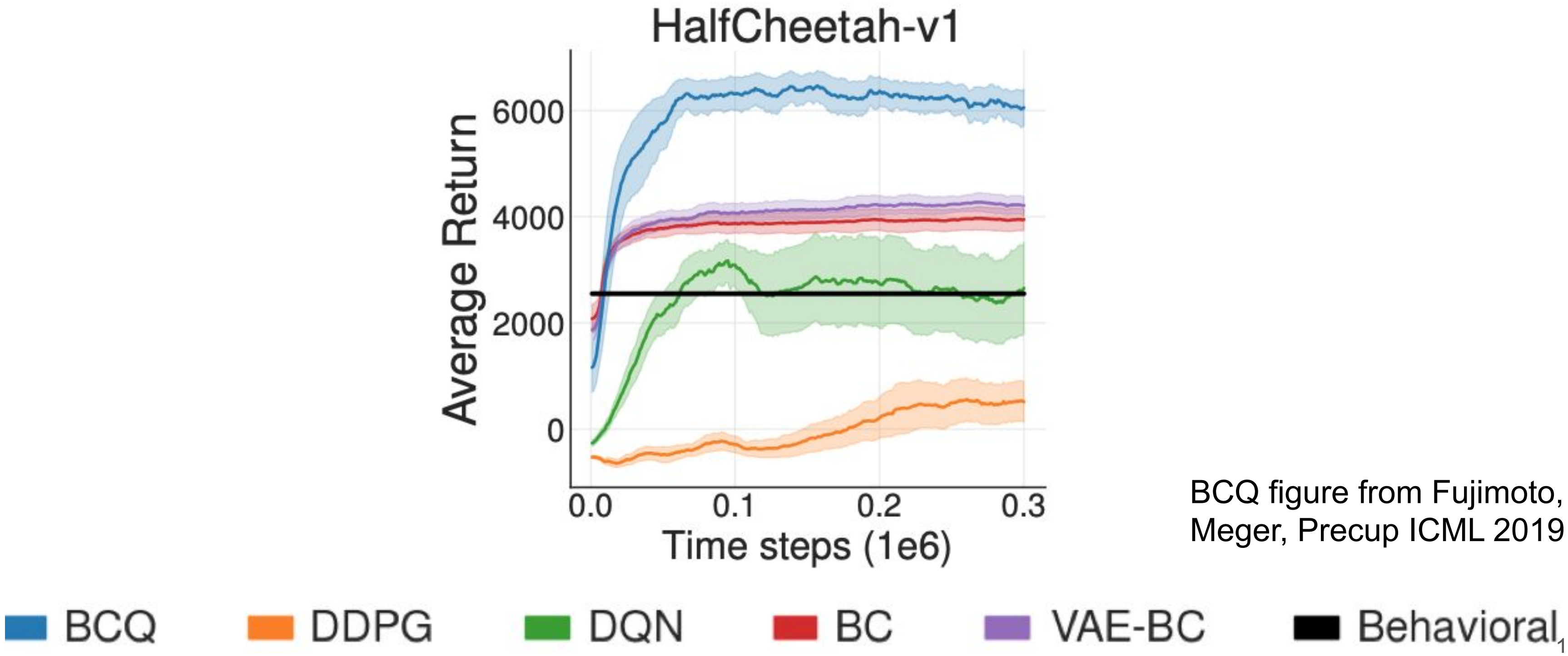
(+ some additional deep RL magic)


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
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
Time steps (1e6)
BCQ
DDPG

BCQ comparison



BCQ figure from Fujimoto, Meger, Precup ICML 2019