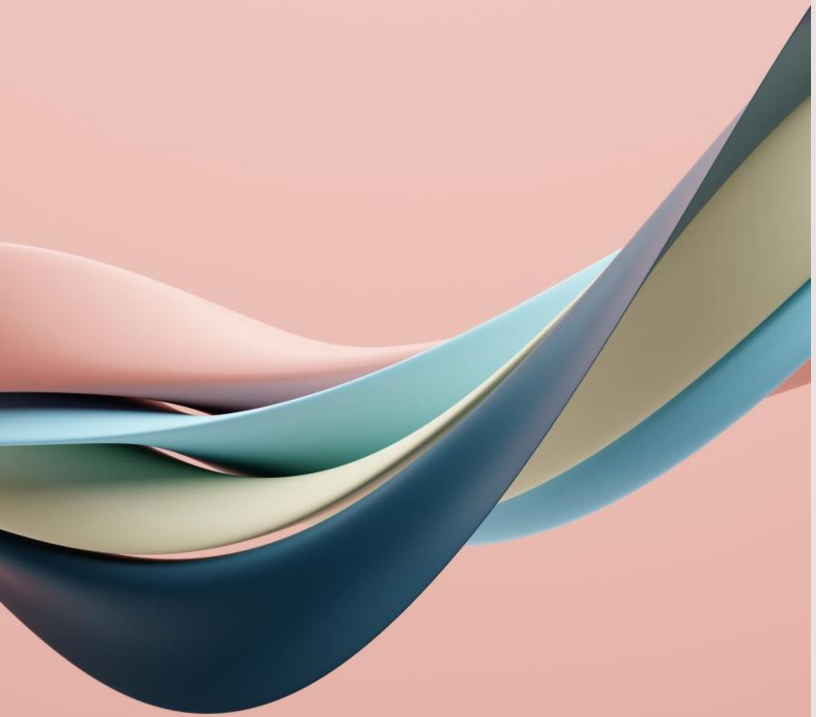


Off-Policy Deep Reinforcement Learning without Exploration

ICML 2019

Outline

- Introduction
- Problem definition
- Batch-Constrained Reinforcement Learning
- Experiment
- Conclusion

Introduction

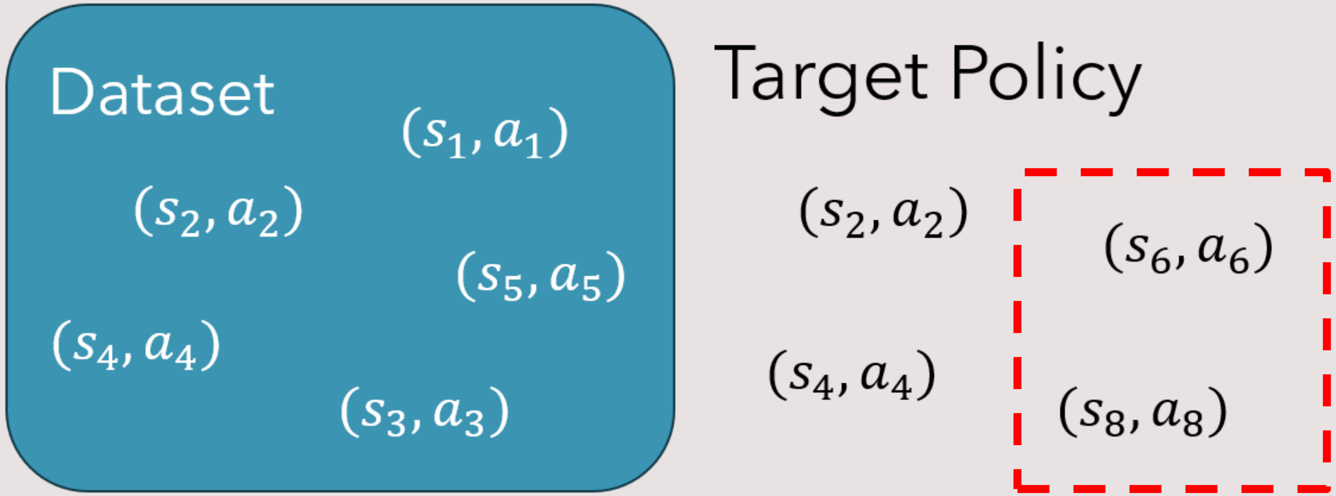
- Batch reinforcement learning :
  - Learning from a fixed dataset without interactions with the environment
  - without restrictions on the quality of the data
  - May occur problem that we call "Extrapolation Error" :
    - Absent Data
    - Model bias
    - Training Mismatch

Introduction

- Batch-constrained reinforcement learning :
  - Agents are trained to
    - maximize reward
    - minimizing the mismatch between the state-action contained in the batch and in the policy.
  - Algorithm in paper : Batch-Constrained deep Q-learning (BCQ)

Problem definition

- Extrapolation Error
  - "Introduced by the mismatch between the dataset and true state-action visitation of the current policy"
  - The target policy selects an unfamiliar action a′ at the next state s′
    - (s′, a′) is unlikely, or not contained, in the dataset



Problem definition

- Absent Data
  - The estimate of $Q_\theta(s', \pi(s'))$ may be arbitrarily bad without sufficient data near $(s', \pi(s'))$
- Model Bias
  - For a stochastic MDP, without infinite state-action visitation, this produces a biased estimate of the transition dynamics
    - We cannot accurately determine the true transition dynamics

$$\mathcal{T}^\pi Q(s,a) \approx \mathbb{E}_{s' \sim \mathcal{B}}[r + \gamma Q(s', \pi(s'))]$$

Problem definition

- Model Bias
  - The expectation is with respect to transitions in the batch $\mathcal{B}$, rather than the true MDP
- Training Mismatch
  - If the distribution of data in the batch does not correspond with the distribution under the current policy
    - Even with sufficient data, the value function may be a poor estimate

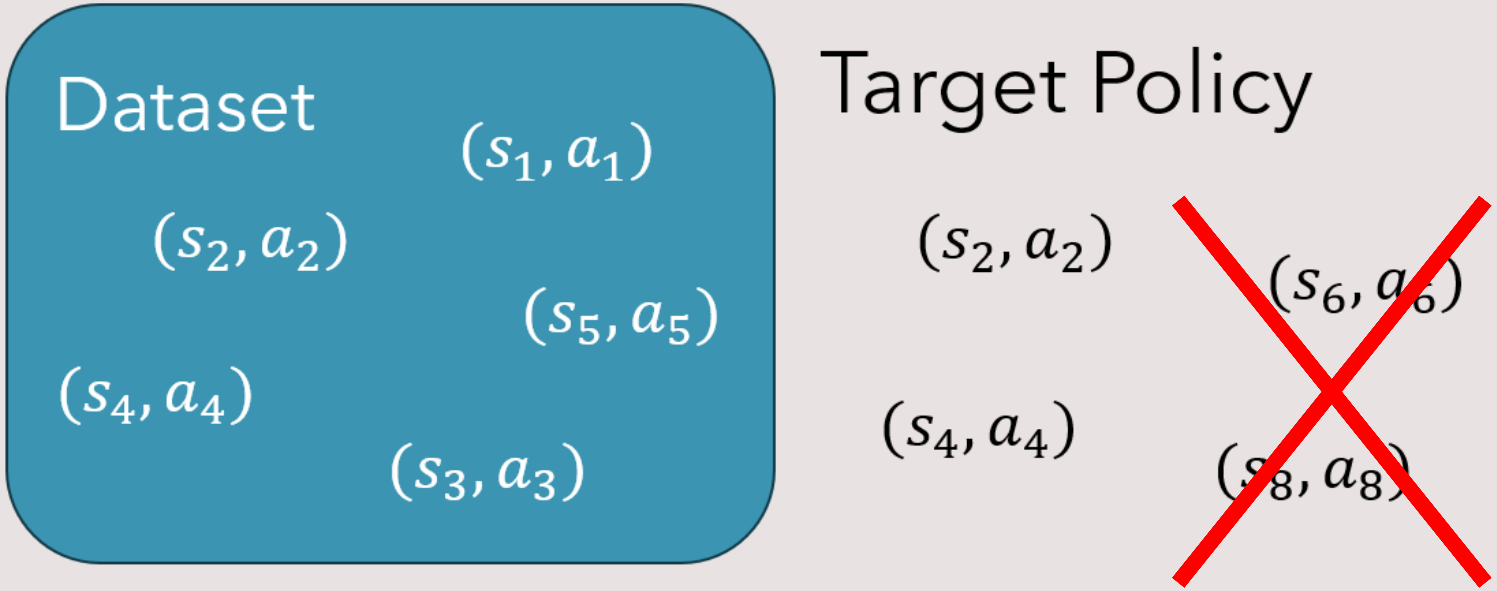
$$\approx \frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s') \in \mathcal{B}} \left|\left| r + \gamma Q_{\theta'}(s', \pi(s')) - Q_{\theta}(s,a) \right|\right|^2$$

Batch-Constrained Reinforcement Learning

- Motivation
  - Current off-policy reinforcement algorithms fail to address extrapolation error
    - Without consideration of the accuracy of the learned value estimate
    - Certain out-of-distribution actions can be extrapolated to higher values
- Idea
  - A policy should induce a similar state-action visitation to the batch
    - Minimize the distance of selected actions to the data in the batch

Batch-Constrained Reinforcement Learning

- Minimize the distance of selected actions to the data in the batch



Choose the nearest action
Ex. (s_4, a_4), (s_5, a_5)

Batch-Constrained Reinforcement Learning

- Definition
  - Experience replay buffer $\mathcal{B}$
  - The MDP $M_{\mathcal{B}}$
    - The value function Q learned with the batch $\mathcal{B}$
    - the same action and state space as the true MDP M
    - An additional terminal state s_{init}
  - The transition probabilities $p_{\mathcal{B}}$

$$p_{\mathcal{B}}(s'|s,a) = \frac{N(s,a,s')}{\sum_{\tilde{s}} N(s,a,\tilde{s})}$$

    - Where $N(s,a,s')$ is the number of times the tuple (s,a,s') is observed in $\mathcal{B}$

Batch-Constrained Reinforcement Learning

- Definition
  - The tabular extrapolation error ϵ_{MDP}

$$\epsilon_{MDP}(s,a) = Q^{\pi}(s,a) - Q^{\pi}_{\mathcal{B}}(s,a)$$

- Goal

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{\boxed{a' s.t. (s',a') \in \mathcal{B}}} Q(s',a'))$$

  - Versus

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma Q(s',\pi(s')))$$

  - We then use the following 4 theorem to prove the feasibility of BCQL

- The tabular extrapolation error ϵ_{MDP}

$$\epsilon_{MDP}(s,a) = Q^{\pi}(s,a) - Q^{\pi}_{\mathcal{B}}(s,a)$$

Batch-Constrained Reinforcement Learning

- Theorem 1.
  - Performing Q-learning by sampling from a batch $\mathcal{B}$ converges to the optimal value function under the MDP $M_{\mathcal{B}}$
    - By the definition of ϵ^{π}_{MDP} and $\epsilon^{\pi}_{MDP} = \sum_s \mu_\pi(s) \sum_a \pi(a|s)|\epsilon_{MDP}(s,a)|$
    - Only $\epsilon^{\pi}_{MDP} = 0$ is required to evaluate a policy π exactly at relevant state-action pairs
    - Denote a policy $\pi \in \prod_{\mathcal{B}}$ as batch-constrained if for all (s,a) where $\mu_\pi(s) > 0$ and $\pi(a|s) > 0$ then $(s,a) \in B$
    - Denote a batch $\mathcal{B}$ as coherent if for all $(\mathrm{s},\mathrm{a},\mathrm{s}') \in \mathcal{B}$ then $\mathrm{s}' \in \mathcal{B}$

Batch-Constrained Reinforcement Learning

- Theorem 2.
  - For a deterministic MDP and all reward functions, $\epsilon_{MDP}^{\pi} = 0$ if and only if the policy π is batch-constrained
  - Furthermore, if $\mathcal{B}$ is coherent, then such a policy must exist if the start state $s_0 \in \mathcal{B}$
    - Reach our goal with the condition that $(s, a) \in \mathcal{B}$

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r + \gamma \max_{\boxed{a' s.t. (s',a') \in \mathcal{B}}} Q(s',a')\right)$$

- Only $\epsilon_{MDP}^{\pi} = 0$ is required to evaluate a policy π exactly at relevant state-action pairs
- Denote a policy $\pi \in \prod_{\mathcal{B}}$ as batch-constrained if for all (s, a) where $\mu_{\pi}(s) > 0$ and $\pi(a|s) > 0$ then $(s, a) \in B$
- Denote a batch $\mathcal{B}$ as coherent if for all $(s, a, s') \in \mathcal{B}$ then $s' \in \mathcal{B}$

Batch-Constrained Reinforcement Learning

- Theorem 3.
  - Given the Robbins-Monro stochastic convergence conditions on the learning rate α, and standard sampling requirements from the environment
  - BCQL converges to the optimal value function Q^*
    - BCQL converges to the optimal batch-constrained policy $\pi^* \in \prod_{\mathcal{B}}$ such that $Q^{\pi^*}(s,a) \geq Q^{\pi}(s,a)$ for all $\pi \in \prod_{\mathcal{B}}$ and $(s,a) \in \mathcal{B}$

Batch-Constrained Reinforcement Learning

- Theorem 4.
  - Given a deterministic MDP and coherent batch $\mathcal{B}$, along with the Robbins-Monro stochastic convergence conditions on the learning rate α and standard sampling requirements on the batch $\mathcal{B}$
  - BCQL converges to $Q_{\mathcal{B}}^{\pi}(s,a)$ where $\pi^*(s) = argmax_{a s.t.(s,a)\in\mathcal{B}} Q_{\mathcal{B}}^{\pi}(s,a)$ is the optimal batch-constrained policy
    - Versus Q-Learning: $\pi(s') = argmax_{a'} Q(s',a')$

Batch-Constrained Reinforcement Learning

- Practical
  - Introduce a conditional variational auto-encoder (VAE)
    - Form a generative model G_{ω} and sample actions from model
  - Introduce a perturbation model $\xi_{\phi}(s, a, \Phi)$
    - An adjustment to an action a in the range $[-\Phi, \Phi]$
    - Increase the diversity of seen actions
  - Introduce a Clipped Double Q-learning
    - Estimate the value by taking the minimum between two Q-networks $\{Q_{\theta_1}, Q_{\theta_2}\}$
    - penalize uncertainty over future states

Batch-Constrained Reinforcement Learning

- Practical
  - Can view generative model G_ω and perturbation model ξ_ϕ as policy network
    - Generated actions should not deviate too far from those in the dataset
    - The other part is responsible for maximizing the cumulative reward

$$\pi(s) = argmax_{a_i+\xi_\phi(s,a_i,\Phi)} Q_\theta\left(s, a_i + \xi_\phi(s, a_i, \Phi)\right), \{a_i \sim G_\omega(s)\}_{i=1}^{n}$$

  - Q-networks take a convex combination of the two values
    - It can be seen as a transition from behavior cloning to running Q-learning

$$r + \gamma \max_{a_i}[\lambda \min_{j=1,2} Q_{\theta_j'}(s', a_i) + (1 - \lambda) \max_{j=1,2} Q_{\theta_j'}(s', a_i)]$$

Batch-Constrained Reinforcement Learning

- Coding

Algorithm 1 BCQ

Input: Batch $\mathcal{B}$, horizon T, target network update rate τ, mini-batch size N, max perturbation Φ, number of sampled actions n, minimum weighting λ.

Initialize Q-networks $Q_{\theta_1}, Q_{\theta_2}$, perturbation network ξ_ϕ, and VAE $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$, with random parameters θ_1, θ_2, ϕ, ω, and target networks $Q_{\theta'_1}, Q_{\theta'_2}$, $\xi_{\phi'}$ with $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$.

for $t = 1$ **to** T **do**

Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$

$\mu, \sigma = E_{\omega_1}(s, a), \quad \tilde{a} = D_{\omega_2}(s, z), \quad z \sim \mathcal{N}(\mu, \sigma)$

$\omega \leftarrow \text{argmin}_\omega \sum (a - \tilde{a})^2 + D_{\text{KL}}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(0, 1))$

Generative model G_ω

Sample n actions: $\{a_i \sim G_\omega(s')\}_{i=1}^n$

Perturb each action: $\{a_i = a_i + \xi_\phi(s', a_i, \Phi)\}_{i=1}^n$

perturbation model ξ_ϕ

Set value target y (Eqn. 13)

Clipped Double Q-learning

$\theta \leftarrow \text{argmin}_\theta \sum (y - Q_\theta(s, a))^2$

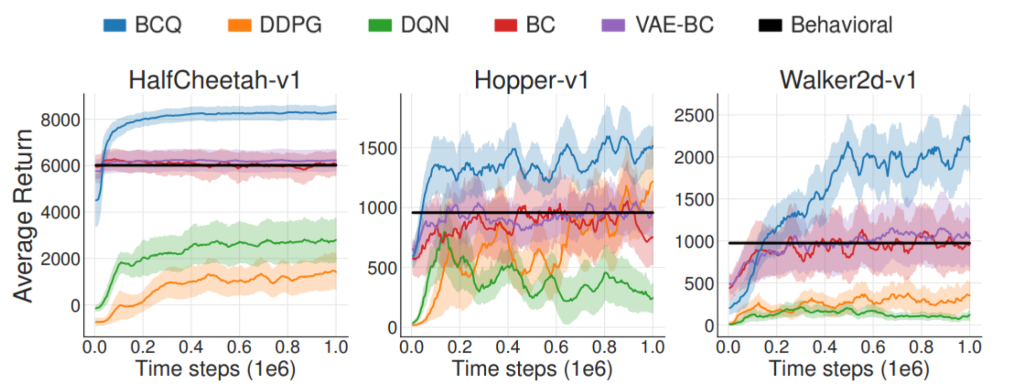
$\phi \leftarrow \text{argmax}_\phi \sum Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), a \sim G_\omega(s)$

Update target networks: $\theta'_i \leftarrow \tau\theta + (1 - \tau)\theta'_i$

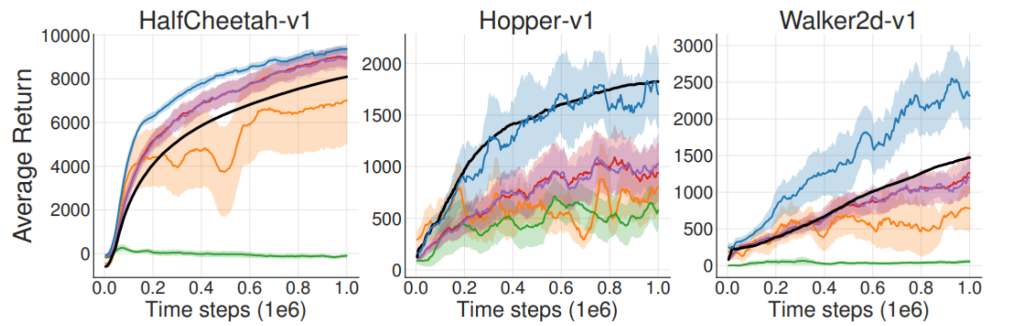
$\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$

end for

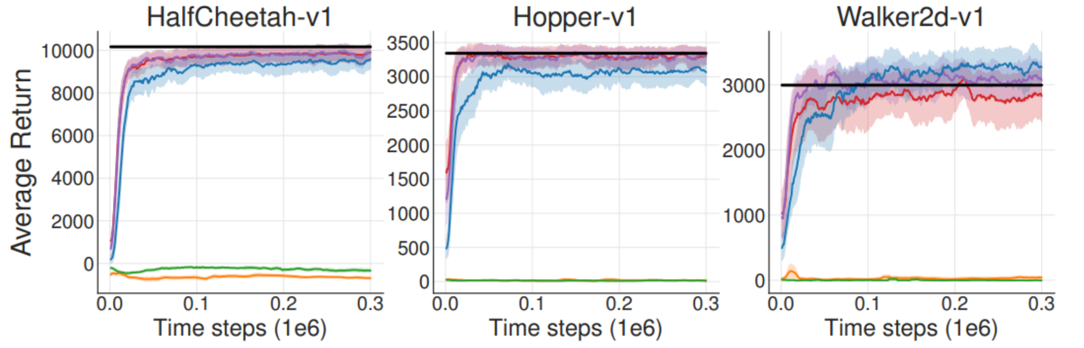
Experiment



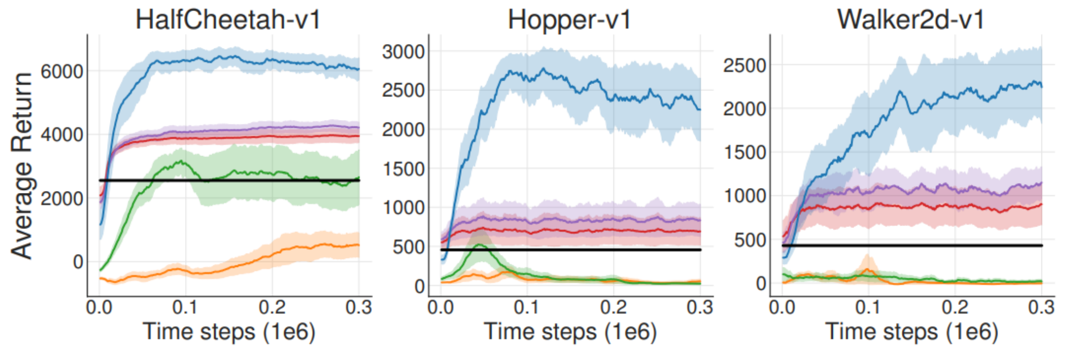
(a) Final buffer performance



(b) Concurrent performance



(c) Imitation performance



(d) Imperfect demonstrations performance

Experiment

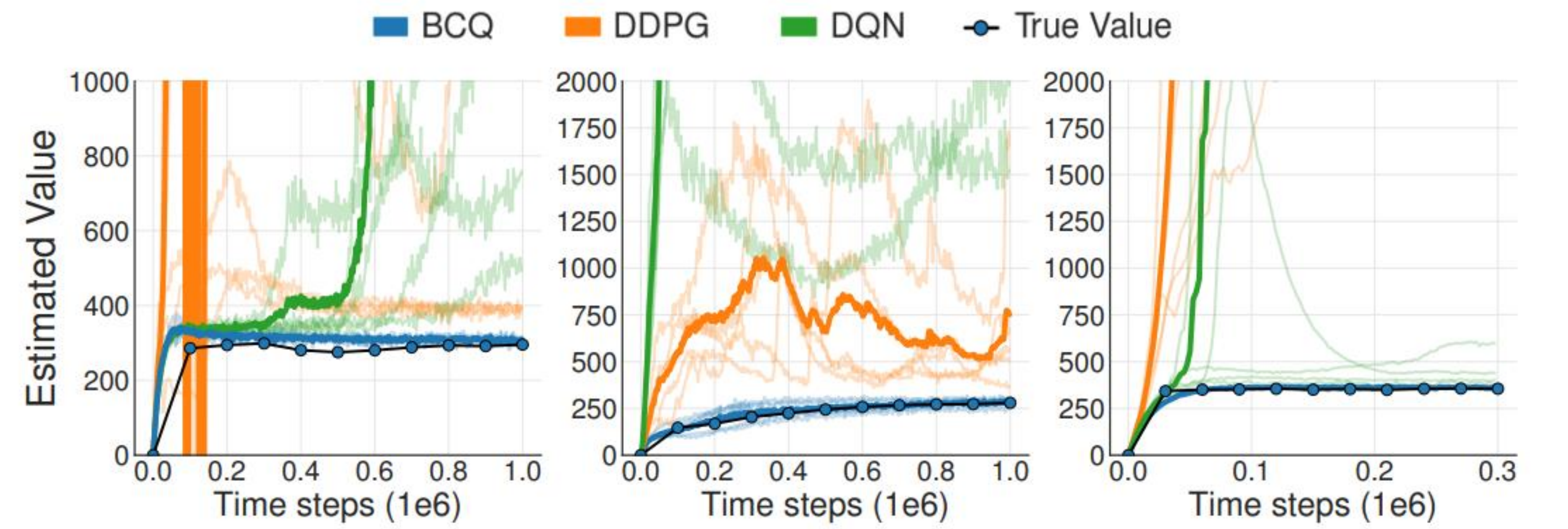
- MuJoCo environments in OpenAI gym
  - *Compare 5 different method in Batch RL:*
    - BCQ (Batch-constrained)
    - DDPG
    - DQN
    - a feed-forward behavioral cloning method (BC)
    - a variant with a VAE (VAE-BC)
  - *4 different batch:*
    - Final Buffer
    - Concurrent
    - Imitation
    - Imperfect demonstrations

Experiment

- Final Buffer:
  - Train a DDPG agent for 1 million time steps, adding Gaussian noise to actions for high exploration, and store all experienced transitions.
- Concurrent:
  - Train the off-policy and behavioral DDPG agents, both agents are trained with the identical dataset.
- Imitation:
  - A trained DDPG agent acts as an expert, and is used to collect a dataset of 1 million transitions
- Imperfect demonstrations:
  - Trained with a batch of 100k transitions collected by an expert policy, with two sources of noise.

Experiment

- BCQ exhibits a highly stable value function in each task



(a) Final Buffer (b) Concurrent (c) Imitation

Conclusion

- Demonstrate a problem in off-policy RL with finite batch data
    - Extrapolation Error

- Present algorithm : Batch-Constrained deep Q-learning (BCQ)
    - Capable of learning from arbitrary batch data, without exploration.