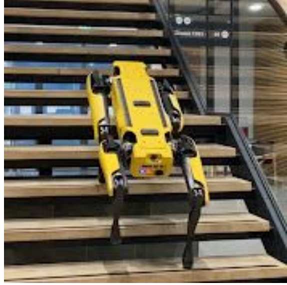
Aalto University



Offline RL, introduction, methods, and challenges

Mohammadreza Nakhaei

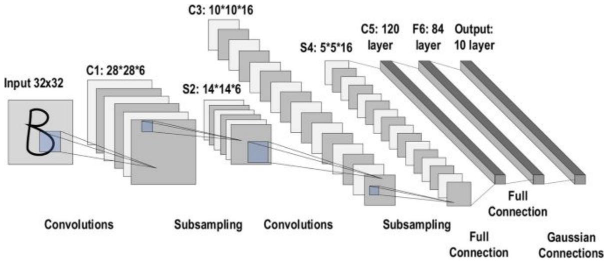
Aalto Robot Learning Lab

November 2023

Introduction

Motivation

Success of Modern ML



Challenges with Online RL:

- Active data collection: expensive, unsafe, unethical, …
- Not perfect simulator, require sim-real transfer

Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. (2021), Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition

What is Offline RL?

Reinforcement Learning with Online Interactions

Offline Reinforcement Learning

Formally:

$$D = \{(s_i, a_i, r_i, s_i')\}$$

$$a_i \sim \pi_\beta(a_i | s_i)$$ **Usually Not Known**

$$s_i' \sim P(s_i' | s_i, a_i)$$

Objective:

Under **fixed, Static** Dataset

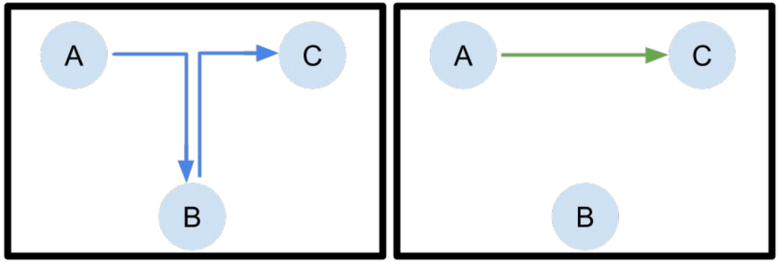
$$J = max_\pi \sum_{t=0}^{T} E_{a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t)}[r(s_t, a_t)]$$

Agarwal, R., Schuurmans, D. & Norouzi, M.. (2020). An Optimistic Perspective on Offline Reinforcement Learning, *ICML*

Offline RL vs Imitation Learning?

Offline RL

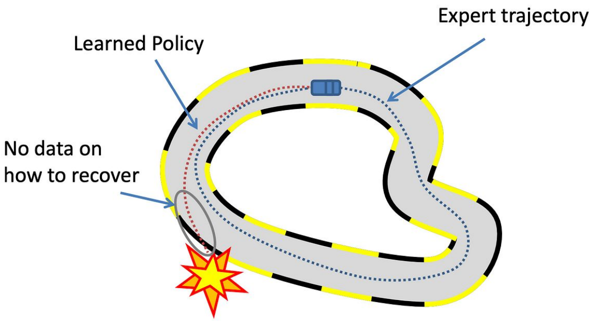
- Reward function
- Sub-optimal noisy demonstrations
- Stitching good behaviors!
- Large datasets



Kumar, A. & Levine, S.. (2020). Offline Reinforcement Learning: From Algorithms to Practical Challenges, *NeurIPS 2020 Tutorial*

Imitation Learning

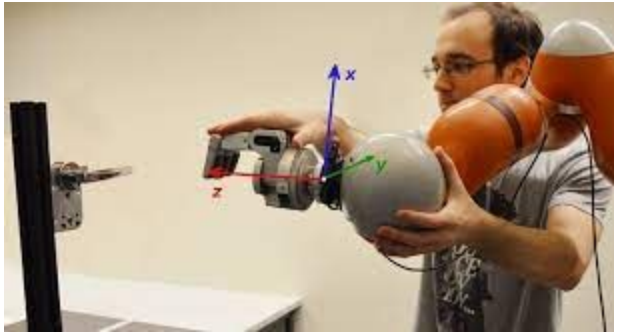
- No reward label
- Expert demonstrations
- Mimic the expert
- Require less data



Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., & Li, H. (2023). End-to-end autonomous driving: Challenges and frontiers

Benefit of Offline RL

- Learning from existing logged data
  - Human operators
  - Hand design controllers/systems
  - RL agents
  - Mixtures of sources
- Data could be shared!
  - Dataset from different task can also be used (reward relabelling)
- Pretraining and fine tuning with RL
  - More sample-efficient than training RL from scratch!
  - Guiding the agent, less exploratory/dangerous behavior



Racca, Mattia, et al. (2016) "Learning in-contact control strategies from demonstration." *IROS*.

Standard Off-policy RL?

Recall: Q-Learning objective:

$$\sum_{(s,a,s')\sim D} ||Q(s,a) - (r + \gamma \max_{a\prime} Q'(s',a'))||^2$$

$$\sum_{(s,a,s')\sim D} ||Q(s,a) - [r + \gamma Q'(s',\pi'(a'|s'))]||^2$$

What could be the problem?

- What if a' is not in the dataset?
- Bootstrapping: Learning from a guess!
- Max operator: select actions with overestimated Q value
- Distribution shift: Learned policy deviates from behavior policy, reaching unfamiliar states!

Agarwal, R., Schuurmans, D. & Norouzi, M.. (2020). An Optimistic Perspective on Offline Reinforcement Learning, International Conference on Machine Learning (ICML).

Methods

Batch Constraints Deep Q Learning (BCQ)

Recall: Q-Learning objective:

$$\sum_{(s,a,s')\sim D} ||Q(s,a) - [r + \gamma Q'(s', \underbrace{\pi'(a'|s')}_{\text{OOD}})]||^2$$

- Constraining the action by conditional generative model
- VAE propose n actions conditions on state (close to behavior policy)
- Evaluate potentials actions with added constrained perturbations (increase diversity)
- Weighted double clipped Q functions:

$$r + \gamma \max_{a_i} \left[\lambda \min_{j=1,2} Q_{\theta'_j}(s', a_i) + (1-\lambda) \max_{j=1,2} Q_{\theta'_j}(s', a_i) \right] \tag{13}$$

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 $\mathcal{B}$
  $\mu, \sigma = E_{\omega_1}(s, a), \quad \tilde{a} = D_{\omega_2}(s, z), \quad z \sim \mathcal{N}(\mu, \sigma)$
  $\omega \leftarrow \text{argmin}_\omega \sum (a - \tilde{a})^2 + D_{\text{KL}}(\mathcal{N}(\mu, \sigma)||\mathcal{N}(0, 1))$
  Sample n actions: $\{a_i \sim G_\omega(s')\}_{i=1}^n$
  Perturb each action: $\{a_i = a_i + \xi_\phi(s', a_i, \Phi)\}_{i=1}^n$
  Set value target y (Eqn. 13)
  $\theta \leftarrow \text{argmin}_\theta \sum (y - Q_\theta(s, a))^2$
  $\phi \leftarrow \text{argmax}_\phi \sum Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), a \sim G_\omega(s)$
  Update target networks: $\theta'_i \leftarrow \tau\theta + (1-\tau)\theta'_i$
  $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
end for

Fujimoto, S., Meger, D., & Precup, D. (2019). Off-policy deep reinforcement learning without exploration. *ICML*

Behavior Regularized Actor Critic (BRAC)

- New objective:

$$\sum_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \left\| Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma E_{\mathbf{a}'\sim\pi_{\text{new}}(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}')\right)\right\|^2$$

$$\pi_{\text{new}} = \arg\max_{\pi} E_{\mathbf{a}'\sim\pi(\cdot|\mathbf{s}')} Q(\mathbf{s}',\mathbf{a}') \text{ s.t. } \pi \text{ close to } \pi_\beta$$

Not known

- Direct method
  - Learn behavior policy in imitation learning style (fit behavior policy)
  - Use direct distance functions such as KL divergence, Wasserstein
  - Easier to implement
- Implicit:
  - Support constraints: $\pi(a|s) \quad \text{if} \quad \pi_b(a|s) \geq \epsilon$
  - Usually use maximum mean discrepancy (MMD) kernel

Wu, Y., Tucker, G., & Nachum, O. (2019). Behavior Regularized Offline Reinforcement Learning.

Behavior Regularized Actor Critic (BRAC)

Especial case: KL Divergence and Gaussian distributions

- Output of the policy and behavior policy are Gaussian
- Objective:

Lagrange Multiplier

$$J = \arg\max_{\theta} E_{s \sim D, a \sim \pi_\theta]}[Q(s,a)] - \lambda KL(\pi_\theta(.\,|s)||\pi_\beta(.\,|s))$$

$$KL(\pi_\theta(.\,|s)||\pi_\beta(.\,|s)) = E_{a \sim \pi_\theta}[\log \pi_\theta(a|s) - \log \pi_\beta(a|s)]$$

$$J = \arg\max_{\theta} E_{s \sim D, a \sim \pi_\theta}[Q(s,a) + \lambda \log \sigma_\theta - \lambda \log \sigma_\beta - \lambda \frac{(a - \mu_\beta)^2}{2\sigma_\beta^2}]$$

- Another way:

$$\bar{r} = r - \lambda KL(\pi_\theta||\pi_\beta)$$

Wu, Y., Tucker, G., & Nachum, O. (2019). Behavior Regularized Offline Reinforcement Learning.

TD3-BC

- **TD3 compared To DDPG**
  - Two critics, take the min for the target value
  - Add noise to the actions selected for the target value

$$y = r + \gamma \min Q_i(s', \pi_{\theta'}(a'|s') + \epsilon)$$

  - Update actor less often than critic

- **TD3-BC for offline RL**
  - Directly encouraging policy to select actions close to behavior policy
  - Normalizing the Q values

$$\lambda = \frac{\alpha}{1/N \sum Q(s_i, a_i)}$$

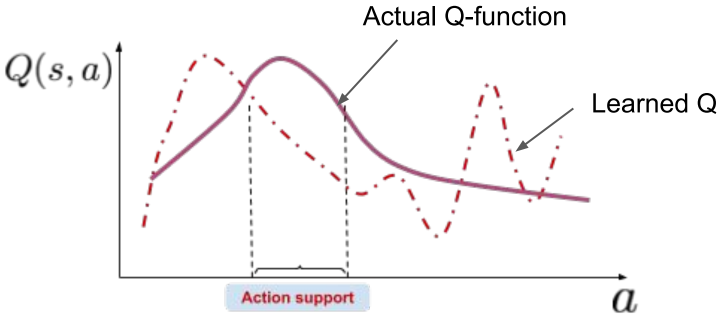
$$\pi = \arg\max E_{(s,a)\sim D}[\lambda Q(s, \pi(s)) - (\pi(s) - a)^2]$$

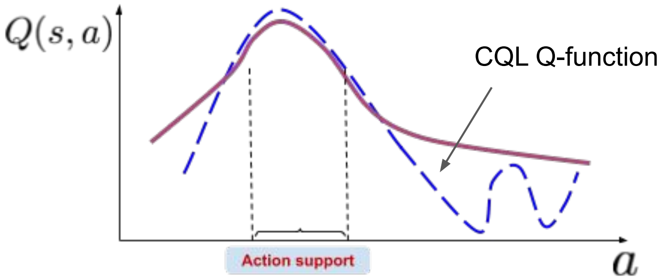
Fujimoto, S., & Gu, S. S. (2021). A minimalist approach to offline reinforcement learning. *NeurIPS*.

Conservative Q Learning (CQL)

- Learn a Q-function, that lower-bounds the policy value, *provably*

$$Q^{\pi}_{CQL}(s_t, a_t) \leq E_{\pi}[\sum_{t'} r(s_{t'}, a_{t'})]$$





$$\min_{Q} \; \alpha \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\log \sum_{\mathbf{a}} \exp(Q(\mathbf{s},\mathbf{a})) - \mathbb{E}_{\mathbf{a}\sim\hat{\pi}_{\beta}(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right]\right] + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q - \hat{\mathcal{B}}^{\pi_k}\hat{Q}^k\right)^2\right]$$

Uniform random actions | From dataset | Bellman Operator Similar to SAC

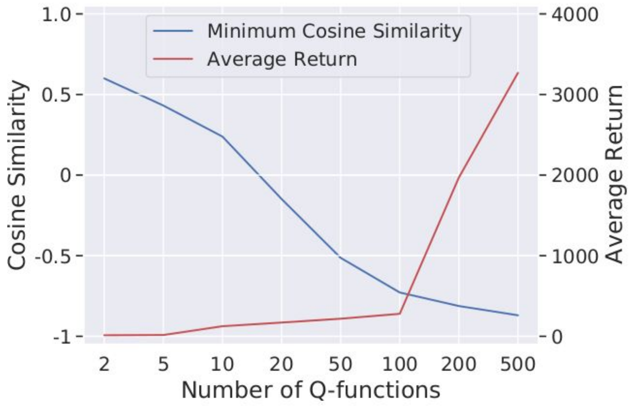
Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative q-learning for offline reinforcement learning.. *NeurIPS*.

Ensemble-Diversified Actor Critic (EDAC)

- Instead of one/two critics, consider N ensemble of them
- Take the minimum value for computing the target (SAC-N)

$$y = r + \gamma \min_{i=0}^{N} Q_i(s', \pi'(a'|s')) - \beta \log \pi(a'|s')$$

- Generally require large number of critics (up to 500).



EDAC: Diversify the critics local structure

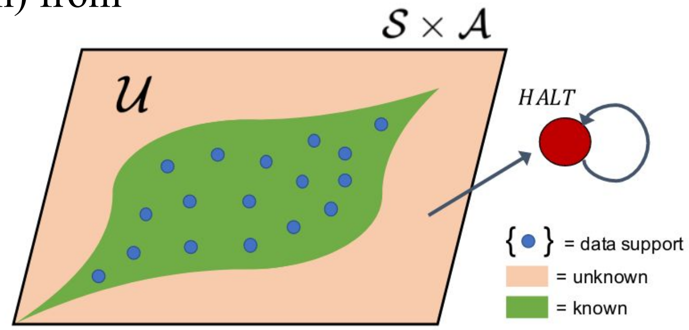
- Diversify the gradient of the critic with respect to the actions
- Reduce the number of critic for the same performance

$$\underset{\phi}{\text{minimize}} \; J_{\text{ES}}(Q_\phi) := \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[\frac{1}{N-1}\sum_{1\le i\ne j\le N} \underbrace{\langle \nabla_{\mathbf{a}} Q_{\phi_i}(\mathbf{s},\mathbf{a}), \nabla_{\mathbf{a}} Q_{\phi_j}(\mathbf{s},\mathbf{a})\rangle}_{\text{ES}_{\phi_i,\phi_j}(\mathbf{s},\mathbf{a})}\right]$$

An, G., Moon, S., Kim, J. H., & Song, H. O. (2021). Uncertainty-based offline reinforcement learning with diversified q-ensemble. *NeurIPS*

Model-based methods based on Uncertainty

1. Estimate transition dynamics $P(s'|s,a)$ (and reward function) from dataset.
2. Apply RL methods in the learned dynamics
   a. Online transitions, no distribution shift



(c)

Problem: Dynamic model estimation is not accurate due to limited data

- Uncertainty Quantification (could be hard).
  - Ensemble of probabilistic networks
- Penalizing reward (MOPO): $\bar{r} = r - \lambda u(s,a)$
- Large negative reward for unknown region (MOREL)

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., ... & Ma, T. (2020). Mopo: Model-based offline policy optimization. *NeurIPS*.
Kidambi, R., Rajeswaran, A., Netrapalli, P., & Joachims, T. (2020). Morel: Model-based offline reinforcement learning, *NeurIPS*

Advantage Weighted Regression (AWR)

Idea: Imitation learning weighted by advantage function

1. Learn value function for the behavior policy: $V^{\pi_\beta}(s) = E_{s\sim D, a_t\sim\pi_\beta(a_t|s_t), s_{t+1\sim P(s_{t+1}|s_t,a_t))}}[\sum_t^T r(s_t, a_t)]$
2. Learn policy: $J = \arg\max_{\pi} E_{s,a\sim D}[\log \pi_\theta(a|s) \exp\left(\frac{1}{\alpha}(R_{s,a} - V^{\pi_\beta}(s))\right)]$

Hyper-parameter   Empirical Return

Advantages

- No OOD action selection
- Easy to implement

Disadvantages

- Monte carlo estimates are noisy (high variance)
- Multi-modality in dataset

Peng, X. B., Kumar, A., Zhang, G., & Levine, S. (2020). Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning.

Advantage Weighted Actor Critic (AWAC)

Idea: Using Actor Critic to learn advantage instead of Monte Carlo samples (Bootstrapping)

1. Learn Q function according to the policy: $$minE_{s,a,s',r\sim D}[||Q(s,a)-(r+\gamma Q'(s',\pi_\theta(a'|s')))||^2]$$
$$A(s,a)=Q(s,a)-Q(s,\pi_\theta(a|s))$$

2. Learn policy: $$J=\arg\max E_{s,a\sim D}[\log\pi(a|s)\exp\left(\frac{A(s,a)}{\alpha}\right)]$$

Advantages

- Policy trained on actions in dataset
- Better advantage estimation

Disadvantages

- OOD actions in advantages
- Multi-modality

Implicit Q Learning (**IQL**): Extension of AWAC; **Sarsa** style value function learning with **expectile regression** Loss

Nair, A., Gupta, A., Dalal, M., & Levine, S. (2020). Awac: Accelerating online reinforcement learning with offline datasets.

Decision Transformer

Idea: Viewing Offline RL as Big sequence

- Predict action condition on a sequence of previous states, actions, and desired returns.
- Transformer architecture (similar to GPT-2, but smaller)

$$\tau = \left(\widehat{R}_1, s_1, a_1, \widehat{R}_2, s_2, a_2, \ldots, \widehat{R}_T, s_T, a_T\right)$$

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., ... & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling, *NeurIPS*

Trajectory Transformer

Idea: Modeling the environment with transformer

- Maximizing log-likelihood of sequences
- Generative multiple possible sequences
- Plan with beam search algorithm
- Select the best action
- Discretized states and actions

Trajectory Transformer

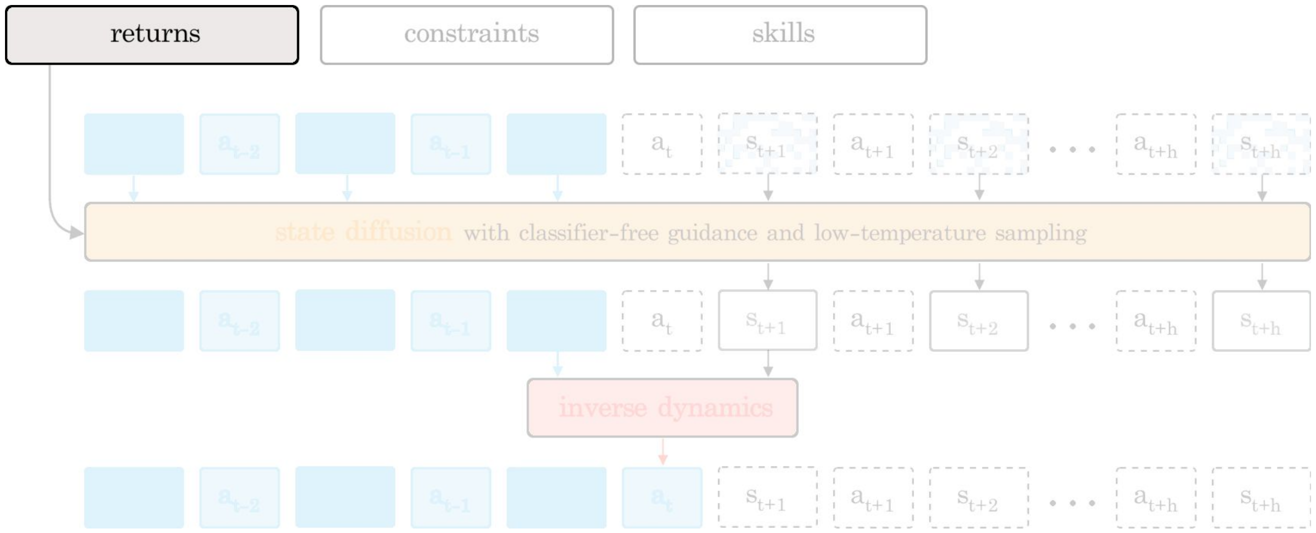
One step MLP Model

Janner, M., Li, Q., & Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *NeurIPS*

Decision Diffusion

Idea: Generating sequence of states using diffusion models.

- Could be conditioned on Returns, constraints and skills
- U-Net architecture with 1d convolutions

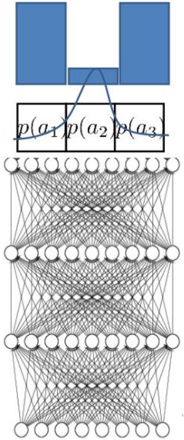


Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., & Agrawal, P. (2023). Is conditional generative modeling all you need for decision-making, ICLR

Challenges

Multi-modal Behavior In Dataset

- Different actions (solutions)
- Data collected from various sources
- More expressive distributions
    - Mixture of Gaussian
    - Latent variable model (VAE)
    - Diffusion models
- Discretization of actions
    - In high-dimensions, almost impractical

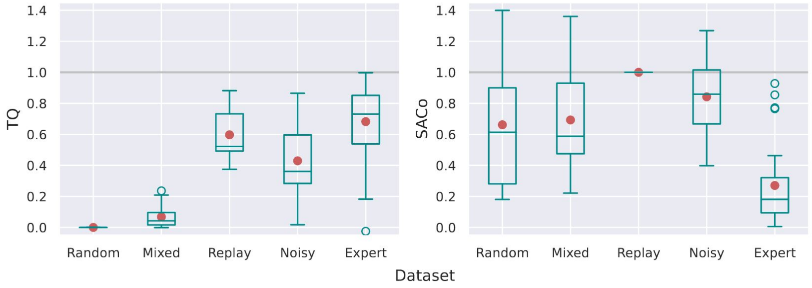


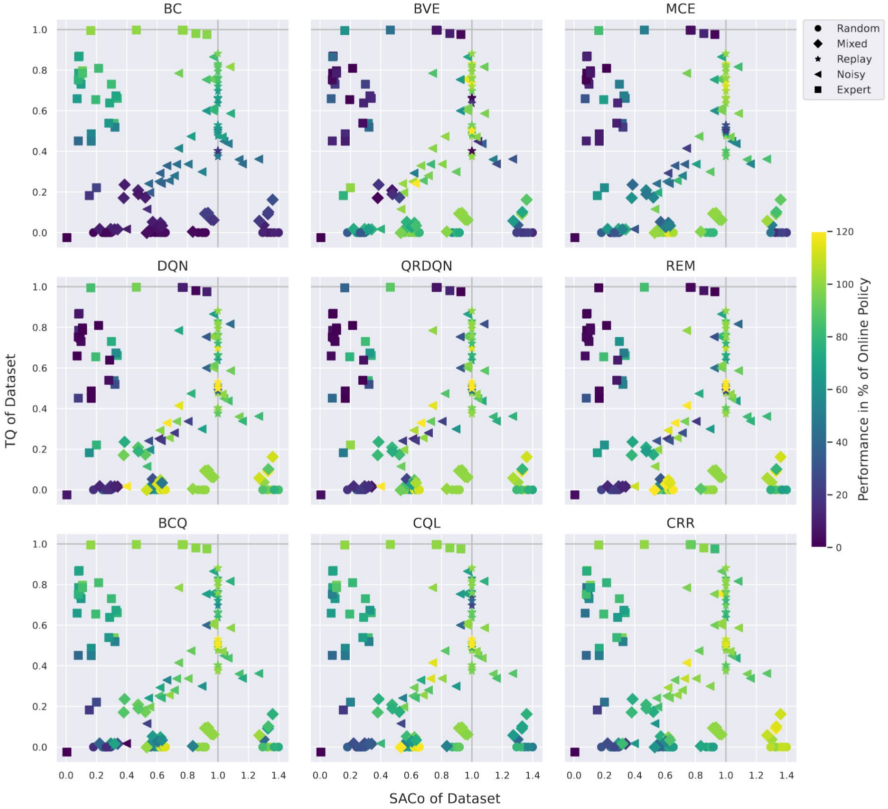
Which Offline RL algorithms suffer from this?

Pictures adopted from Sergey Levine, Deep RL course at Berkeley

Dataset perspective

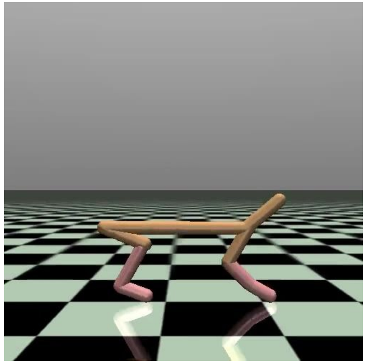
- Define two metrics for dataset:
  - TQ: Exploitive behavior of dataset
  - SACo: Diversity of dataset, based on entropy
- **When to use which algorithm?**

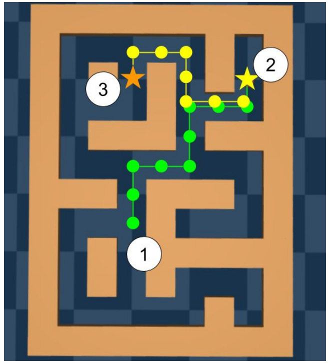




Schweighofer, K., Dinu, M. C., Radler, A., Hofmarcher, M., Patil, V. P., Bitto-Nemling, A., ... & Hochreiter, S. (2022). A dataset perspective on offline reinforcement learning. *In Conference on Lifelong Learning Agents.*

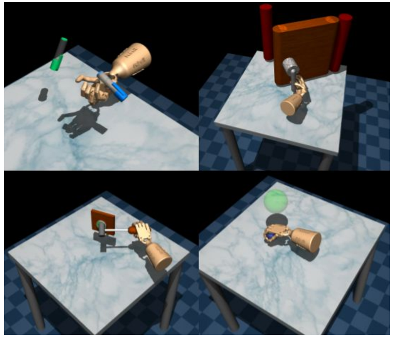
Datasets and Benchmarks





D4RL Benchmark (Mujoco Tasks)

- Random, Medium, Replay, Expert, and mixtures
- Collected by training online RL Agent



NeoRL

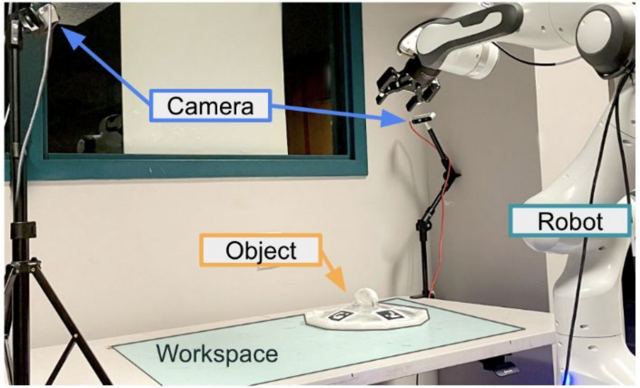
- Learning from conservative and limited data
  - D4RL is too much exploratory (not realistic)
- Comparing with the working policy
- More application including industrial process, finance

Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning.
Qin, R. J., Zhang, X., Gao, S., Chen, X. H., Li, Z., Zhang, W., & Yu, Y. (2022). NeoRL: A near real-world benchmark for offline reinforcement learning, *NeurIPS*

Realistic Offline RL

- Used practical real-world data:
  - 6500 trajectories over 800 robot hours
  - Hardware noise and delays
- Can combination of datasets be beneficial?

Agent	Train Data	Test Task slide	Test Task lift	Test Task PnP
BC	in-domain	0.681 ± 0.147	0.823 ± 0.177	0.818 ± 0.185
	slide	0.681 ± 0.147	0.582 ± 0.058	0.612 ± 0.083
	slide+lift	0.595 ± 0.127	0.580 ± 0.053	0.605 ± 0.120
	slide+lift+PnP	0.610 ± 0.137	0.609 ± 0.079	0.640 ± 0.144
MOREL	in-domain	0.629 ± 0.160	0.678 ± 0.186	0.750 ± 0.197
	slide	0.629 ± 0.160	0.606 ± 0.063	0.744 ± 0.174
	slide+lift	0.616 ± 0.146	0.726 ± 0.184	0.636 ± 0.173
	slide+lift+PnP	0.715 ± 0.134	**0.896 ± 0.133**	0.753 ± 0.181
AWAC	in-domain	0.732 ± 0.113	0.863 ± 0.149 *	0.735 ± 0.175 *
	slide	0.732 ± 0.113	0.638 ± 0.055	0.770 ± 0.111*
	slide+lift	0.734 ± 0.110 *	0.899 ± 0.149	0.813 ± 0.121
	slide+lift+PnP	0.644 ± 0.144 *	0.728 ± 0.200 *	0.758 ± 0.188 *
IQL	in-domain	**0.767 ± 0.065**	0.880 ± 0.149	0.601 ± 0.228
	slide	0.767 ± 0.065	0.258 ± 0.033	0.810 ± 0.107
	slide+lift	0.704 ± 0.141	0.863 ± 0.166	**0.842 ± 0.114**
	slide+lift+PnP	0.643 ± 0.143	0.684 ± 0.158	0.833 ± 0.183



Zhou, G., Ke, L., Srinivasa, S., Gupta, A., Rajeswaran, A., & Kumar, V. (2023). Real world offline reinforcement learning with realistic data source. *ICRA*

Offline Evaluation (Model Selection)

How to make sure learned policy is good?

- There is no general, reliable offline method!

Using real world:

- Accurate, but can be expensive/risky, not completely offline!

Simulators

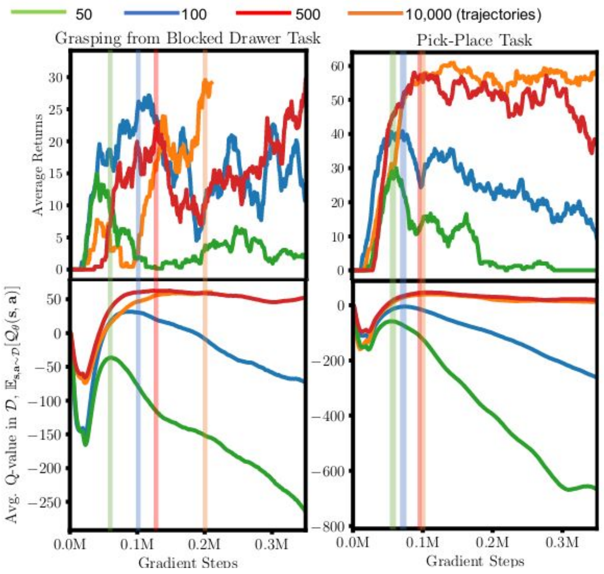
- Might be helpful for comparing policies, but building a good simulator can be hard

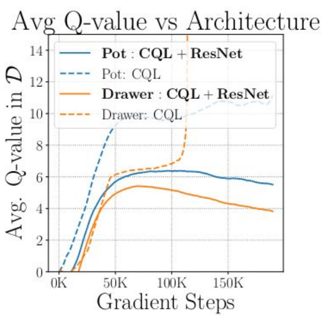
Heuristics

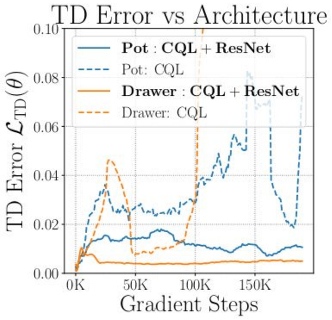
- Easy and cheap, but not reliable and general purpose

Workflow for conservative Offline RL

- Provide metrics and condition for overfitting and underfitting in CQL
- Overfitting:
  - **Expected average Q values of dataset** decreasing after increasing, also low value
  - Add capacity decreasing regularizer (VIB)
- Underfitting:
  - **High values in TD errors**, compare with higher capacity model (regularizer), if TD errors decrease significantly, underfitting
  - Add capacity decreasing regularizer (DR3)
  - Larger models (ResNet)







Kumar, A., Singh, A., Tian, S., Finn, C., & Levine, S. (2022). A workflow for offline model-free robotic reinforcement learning, *CoRL*

Take Away from Today

- Difference between offline and online RL
- Importance of offline RL
- Distribution shift
- The basic idea behind some methods
- Some of the current limitation of offline RL