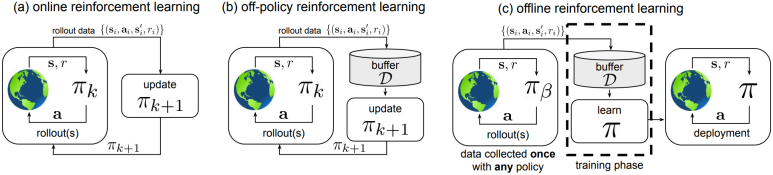
Low Rank structure for Offline Reinforcement Learning

Xinyue (Sherry) Lou

Department of Statistics
University of Chicago

May, 2025



(a) online reinforcement learning
rollout data $\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i)\}$
$\mathbf{s}, r$
π_k
$\mathbf{a}$
rollout(s)
update
π_{k+1}
π_{k+1}
(b) off-policy reinforcement learning
rollout data $\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i)\}$
buffer
$\mathcal{D}$
$\mathbf{s}, r$
π_k
$\mathbf{a}$
rollout(s)
update
π_{k+1}
π_{k+1}
(c) offline reinforcement learning
$\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i)\}$
$\mathbf{s}, r$
π_β
$\mathbf{a}$
rollout(s)
data collected **once**
with **any** policy
buffer
$\mathcal{D}$
learn
π
training phase
$\mathbf{s}, r$
π
$\mathbf{a}$
deployment

Offline Reinforcement Learning

- **Reinforcement Learning**: Agent learns a policy $\pi(a|s)$ to maximize the expected return:

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t, a_t)\right]$$

- **Off-Policy Reinforcement Learning**: Learns $\pi(a|s)$ using data collected by a different behavior policy $\pi_\beta(a|s)$
- **Offline Reinforcement Learning**: Special case of off-policy RL where no new environment interaction is allowed.

$$\text{Goal: } \pi^* = \arg\max_{\pi} \mathbb{E}_{(s,a)\sim d^{\pi}}[Q^{\pi}(s,a)]$$

$$D = \{(s_i, a_i, r_i, s_i')\}_{i=1}^{N} \sim \pi_\beta$$

Q-Function Definition

- **State-Value Function**:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s\right]$$

- **Action-Value Function (Q-Function)**:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a\right]$$

Problem in Offline Reinforcement Learning

- **Distribution Shift**:

$$(s, a) \sim \pi \quad \text{but data from} \quad D = \{(s_i, a_i)\}_{i=1}^{N} \sim \pi_\beta$$

Leads to *extrapolation error* when $Q^\pi(s, a)$ is estimated outside support of π_β.

- **Coverage Assumption**:

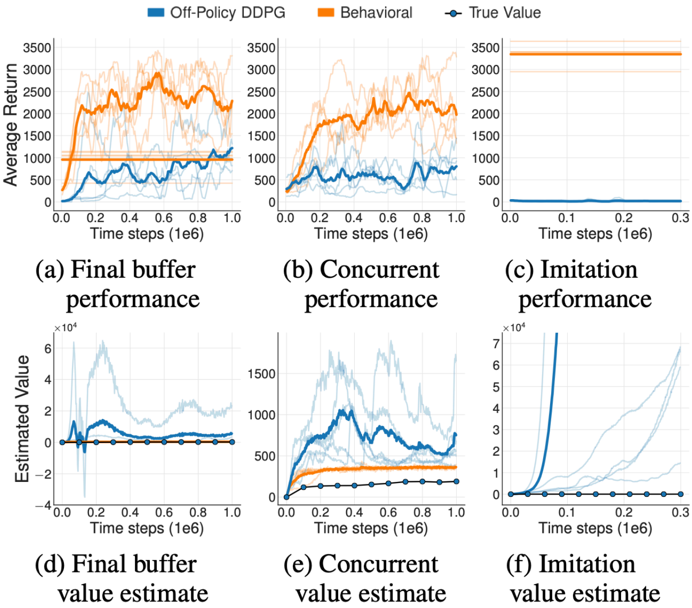
$$\text{Require:} \quad d^\pi(s, a) \ll d^{\pi_\beta}(s, a)$$

If π induces state-action pairs poorly covered by π_β, estimation becomes unreliable.

- **Sample Efficiency**:

$$\text{Offline RL learn from finite dataset} \quad D = \{(s_i, a_i, r_i, s_i')\}_{i=1}^{N}$$

No further interaction $\Longrightarrow$ better algorithms must be more sample efficient.



(a) Final buffer performance

(b) Concurrent performance

(c) Imitation performance

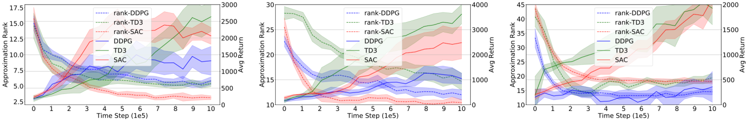
(d) Final buffer value estimate

(e) Concurrent value estimate

(f) Imitation value estimate

Motivation and Justification for Low-Rank Structure

- Low-rank structures are empirically observed and theoretically supported in deep Q-networks.



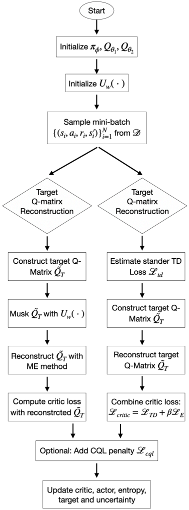
(a) Hopper (b) Walker2d (c) Ant

- Matrix completion methods, under the low-rank assumption, can mitigate extrapolation errors by smoothing out uncertainties in poorly covered regions.

Incorporating low-rank structures into reinforcement learning enhances sample efficiency and can alleviate the strict data coverage requirement.

- Estimating an ϵ-optimal Q-function requires $\Omega\left(\frac{1}{\epsilon^{d_1+d_2+2}}\right)$samples for general Lipschitz functions, but improves to $O\left(\frac{1}{\epsilon^{\max(d_1,d_2)+2}}\right)$ when the optimal Q-function has low rank r and γ is sufficiently small.
  Sample Ecient Reinforcement Learning via Low-Rank Matrix Estimation
- With low-rank MDP, accurate policy evaluation is possible via me, with error bounded by an operator discrepancy $Dis(p,q) = \min_{g:\mathrm{supp}(g)\subseteq\mathrm{supp}(p)} \|g-q\|_{\mathrm{op}}$.
  Matrix Estimation for offline Reinforcement Learning with Low-Rank Structure

Existed Algorithm



Uncertainty Estimator (U_ω):

- **Count-Based (CB)**: $U_{\text{CB}}(s, a) = \frac{1}{N(s,a)}$
- **Bootstrapped-Based (BB)**:

$$U_{\text{BB}}(s, a) = \sqrt{\frac{1}{K} \sum_{i=1}^{K} (Q_i(s, a) - \bar{Q})^2}$$

Loss Construction:

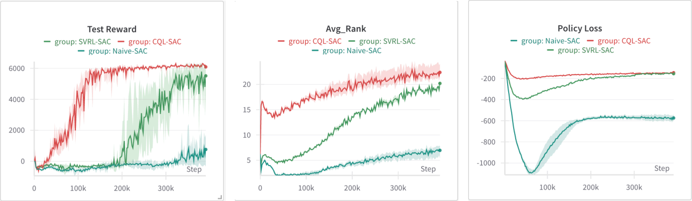
- TD loss: $\mathcal{L}_{\text{TD}} = (Q - y)^2$
- Eval loss: $\mathcal{L}_{\text{Eval}} = (Q - Q_{\text{recon}})^2$
- Total: $\mathcal{L} = \mathcal{L}_{\text{TD}} + \beta \mathcal{L}_{\text{Eval}}$

1. Introduction
2. Literature Review
3. **Experiment Result**
4. Further Investigation and References

Experiment Setting

- **Environment:** We use the `HalfCheetah-v2` locomotion task from the MuJoCo benchmark.
  - 17-dimensional state space, 6-dimensional continuous action space.
  - Goal: maximize forward velocity while maintaining stability.
- **Dataset:** We use the `halfcheetah-medium-v2` dataset from the D4RL benchmark.
  - Collected using a medium-performance policy trained with SAC.
  - Represents moderately good but suboptimal behavior.
- **Algorithm:** Soft Actor-Critic (SAC)
  - Off-policy actor-critic algorithm with entropy regularization.
  - Objective: maximize expected return + entropy to encourage exploration.
  - Uses twin Q-networks, stochastic actor, and soft updates of target critics.

Experiment Result



- **Method:** Structured Value-based RL (SVRL) with random masking and target Q-matrix reconstruction.
- **Findings:** SVRL-SAC consistently outperforms naive SAC and closely approaches the performance of CQL-SAC.
- **Insight:** Although `HalfCheetah` is a high-dimensional continuous control task with weaker low-rank structure, SVRL-SAC demonstrates promising results. Its advantage may become more significant when:
  - The dataset has poor coverage (i.e., limited or biased behavior).
  - The environment exhibits stronger low-rank structure.

Further Investigation

- **Uncertainty-Aware Masking:** Incorporate bootstrapped or count-based uncertainty estimates into the masking strategy to selectively reconstruct more confident Q-value regions.
- **Task and Dataset Diversity:** Extend evaluation to additional benchmark tasks and dataset types.
- **Dynamic Target Fusion:** Currently, we always reconstruct the target Q-matrix via low-rank completion.
  - Future work will explore dynamically weighting the structured signal and original TD target:

$$Q_{\text{target}} = (1 - \lambda_t) Q_{\text{true}} + \lambda_t Q_{\text{recon}}, \quad \lambda_t \in [0, 1]$$

  - This may help balance structure guidance and raw Bellman signal adaptively during training.
- **Rank Dynamics:** Investigate how approximate rank evolves over training, and how it correlates with performance or overfitting.
- **Beyond SoftImpute:** Explore alternative low-rank methods (e.g., nuclear norm, matrix factorization) for more expressive structure modeling.

[1] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *Proceedings of the 36th International Conference on Machine Learning*.

[2] D. Shah, D. Song, Z. Xu, and Y. Yang, "Sample efficient reinforcement learning via low-rank matrix estimation," 2020.

[3] X. Xi, C. L. Yu, and Y. Chen, "Matrix estimation for offline reinforcement learning with low-rank structure," 2023.

[4] T. Sang, H. Tang, J. Hao, Y. Zheng, and Z. Meng, "Uncertainty-aware low-rank q-matrix estimation for deep reinforcement learning," 2021.

[5] Y. Yang, G. Zhang, Z. Xu, and D. Katabi, "Harnessing structures for value-based planning and reinforcement learning," 2020.

[6] S. Dittert, "Cql." `https://github.com/BY571/CQL`, 2021.