

# EEE448/548 Reinforcement Learning & Dynamic Programming

## Offline RL

Ayşe Selin CİN , Berk Kaan ELMAS , Burak Eren ÖZCAN , Melih Kutay YAĞDERELİ , Tuna SAYGIN

### 1. Introduction

Offline reinforcement learning is a methodology of learning the optimal policy by only learning from an entirely fixed dataset. This methodology has a significant role when exploration and data collection is not easily available. Additionally, simulations can be problematic as it may not reflect the real life world scenarios. Hence, offline RL tries to learn a policy or make decision according to a prior behavior policy. However, the static dataset of not interacting with the environment cause distributional shift which is the essential problem of offline RL and it is the reason why it performs poorly on rl. This problem will be explained in detail in section 2. In section 3, seminal works that shaped offline RL will be evaluated in terms of their motivation, strength and weaknesses. In section 4, state of the art methodology and research going on this area will be evaluated. Additionally, open research areas will be evaluated and the report will be concluded.

Offline reinforcement learning (RL) studies how to learn an optimal policy purely from a fixed dataset of past interactions, without any further environment roll-outs. Formally, we consider a Markov decision process as  $\{S, A, P, R, \gamma\}$  and within a static dataset  $D = \{s_i, a_i, r_i, s_{i+1}\}_{i=1}^N$  collected under a behavior policy under  $\pi_\beta(a|s)$ . The objective is to optimize the expected return

$$J(\pi) = E_{T \sim p_\pi} \left[ \sum_{i=0}^{\infty} \gamma^i r_i \right]$$

where  $p_\pi$  denotes trajectories generated by  $\pi$ . Offline RL is critical when online exploration is costly, unsafe, or when simulators poorly approximate real-world dynamics. This report begins, in Section 2, by clearly defining the distributional-shift problem that arises in offline settings. Section 3 surveys the seminal works that founded the field, analyzing their motivations, strengths, and weaknesses. Section 4 reviews state-of-the-art algorithms and open research directions, before concluding with future challenges and opportunities.

### 2. Problem Description

The core challenge in offline RL is distributional shift: the mismatch between the state-action distribution  $\rho_\beta(s, a)$  under the behavior policy and the distribution  $\rho_\pi(s, a)$  induced by the learned policy. Since the dataset  $D$  provides samples only from  $\rho_\beta$ , any Bellman backup

$$T^\pi Q(s, a) = R(s, a) + \gamma E_{s' \sim p(|s, a)} [Q(s', a)]$$

applied at  $(s, a)$  values outside the support of  $\rho_\beta$  must extrapolate, leading to extrapolation error. Concretely, when  $\pi(a|s) > 0$  but  $\pi_\beta(a|s) \approx 0$ , estimates of  $Q(s, a)$  are uncontrolled by data. These errors propagate through successive backups—so-called bootstrapping—amplifying estimation bias and often causing learned policies to diverge or collapse to trivial behaviors. In practical terms, distributional shift undermines offline RL in safety-critical domains such as healthcare (where patient treatment logs form  $D$ ), autonomous driving (sensor logs under human control), finance (historical trading data), and industrial control systems (SCADA logs). Key sub-challenges include coverage where we need to ensure  $\text{supp}(\rho_\pi) \subset \text{supp}(\rho_\beta)$  to decrease the distribution shift error. Additionally, how to estimate value function in the unexplored states by  $\pi_\beta$  still faces challenges.

These problems and concept of offline RL is not only theoretical but it has lots of different application areas. As discussed in the section 1, offline RL is essential for healthcare, finance and autonomous driving where the data collection is unsafe, and impractical. That's why the distributional shift problem of the offline RL may cause problematic as there may be unexplored, and not sampled states which in turn lead to incorrect decision making policy in a critical environment.

### 3. Seminal Works

#### **Seminal Work 1:Ernst et al. (2005) Tree-Based Batch Mode Reinforcement Learning[1]**

**Tree-Based Batch Mode Reinforcement Learning** is found by Ernst et al.[1] which is one of the first nominal papers after q learning that converts rl objective of maximizing expected return to a loss function that can be learnable through supervised learning methods like tree regressions or ensemble methods and this paper further improved through the thought of deep fitted q iteration.

Fitted Q-Iteration(FQI) was originally motivated by the desire to turn the Bellman optimality equation into a purely supervised-learning problem. At each iteration, one computes Bellman backup as regression target:

$$y_i = r_i + \gamma \max_a [Q_{k-1}(s'_i, a'_i)]$$

for each transition  $(s_i, a_i, r_i, s'_i)$  in the fixed batch, and then uses batch regression to fit a function  $Q_k(s, a) \approx y_i$ . This reframing leverages powerful, off-the-shelf regressors (e.g. regression trees or kernels) and sidesteps incremental, on-line updates. The updates are done by the mean squared error formulation of minimization

$$Q_k = \operatorname{argmin}_{f \in F} \left[ \sum_{i=1}^N (f(s_i, a_i) - y_i^{(k)})^2 \right]$$

This formulation leverages any supervised learner—trees, kernels, or MLPs—to handle continuous or high-dimensional spaces and fully reuse a static dataset without further environment interaction. FQI's global updates curb the instability of single-sample Q-learning, and under ideal function classes it provably converges.

Ernst et al. (2005) evaluated FQI on five continuous control benchmarks (“Left or Right,” “Car on the Hill,” “Acrobot Swing-Up,” “Bicycle Balancing,” and “Bicycle Riding”) using two metrics—policy score (average Monte Carlo return) and Bellman residual (mean squared Bellman error)—and compared five tree-based regressors: KD-Tree, Pruned CART, Tree Bagging (50 unpruned CART trees), Extra-Trees (50 extremely randomized trees with KKK random splits per node), and Totally

Randomized Trees (one random split per node). They found that ensembles dramatically outperform single trees, with Extra-Trees achieving the lowest residuals and highest returns (Tree Bagging close behind), that pruning is vital for standalone CART but unnecessary for ensembles, that performance plateaus around M=50M=50M=50 trees, that freezing tree structure (KD-Tree, Totally Randomized) guarantees convergence by avoiding per-iteration rebuilds, and that doubling dataset size consistently lowers residuals and modestly boosts scores, with both metrics stabilizing within 5–10 iterations.

<b>Strengths</b>	<b>and</b>	<b>Weaknesses:</b>
------------------	------------	--------------------

One major strength of Fitted Q-Iteration (FQI) is that it allows offline Reinforcement Learning (RL), meaning it can learn from previously collected data without needing to interact with the environment again. This is especially important in situations where collecting new data is difficult, expensive, or risky, such as in healthcare, autonomous driving, or finance. By using data that was already gathered, FQI can still learn effective policies. The method works by using batch regression techniques, which are more stable compared to traditional Q-learning, where updates happen based on single samples. This stability makes FQI useful when it's necessary to learn from fixed datasets. Another advantage of FQI is that it uses simple, off-the-shelf tools like decision trees, which makes it easy to apply and computationally efficient. These tools are not complicated and can handle many types of problems, making FQI accessible and useful for practical situations.

However, FQI has some limitations. It doesn't perform well when the state space is large or continuous. This is because the method needs to break the state space into smaller parts to use regression techniques, which can be challenging in environments with many possible states or continuous variables. Also, although FQI works well with decision trees, it doesn't handle more complex methods like neural networks very well. When FQI is combined with neural networks or other function approximators, it can become unstable, much like traditional Q-learning. Another problem with FQI is that, while it improves the stability of learning compared to earlier methods, it still doesn't solve the issue of errors that occur when the data does not cover all possible states and actions. This can cause problems, especially when the data is incomplete or doesn't represent every situation that might happen in the real world. Despite these weaknesses, FQI was an important breakthrough in offline RL. It helped push the field forward by showing that you can use supervised learning methods to handle RL problems, and it laid the foundation for later improvements in offline RL techniques.

### **Seminal Work 2: Precup et al. (2001) Off-Policy Temporal-Difference Learning with Function Approximation [2]**

Precup et al. (2001) tackle the longstanding instability of off-policy TD methods—such as Q-learning—when combined with linear function approximation. Motivated by counterexamples (e.g. Baird's "star" MDP) showing divergence even in simple tabular settings, they derive the first stable off-policy policy-evaluation algorithm that converges w.p.1 under linear approximation. Their key insight is to marry  $\text{TD}(\lambda)$  eligibility-trace updates with per-decision importance sampling so that each bootstrapped target is unbiased for a fixed target policy  $\pi$ , even though data are generated by a different behavior policy  $b$ .

At each time step  $t$ , they form the one-step TD error

$$\delta_t = r_{t+1} + \gamma \rho_{t+1} \theta_t^T \varphi(a_{t+1}, s_{t+1}) - \theta_t^T \varphi(a_t, s_t)$$

Where Q function defined as  $Q(s, a) = \theta^T \varphi(a, s)$  and  $\rho_{t+1} = \frac{\pi(a_t | s_t)}{b(a_t | s_t)}$  is importance weights of importance sampling. This equation is used and eligibility trace is updated as  $e_t = \rho_t (\lambda \gamma e_{t-1} + \varphi(a_t, s_t))$  which will then be used to update weights by  $\theta_t = \theta_{t-1} + \alpha \delta_t e_t$ .

Under standard stochastic-approximation conditions—bounded rewards, proper policies ensuring coverage, diminishing step sizes, and a bounded-variance condition on the  $\rho$ -products—they prove that  $\theta_t$  converges w.p. 1 to  $\theta_\infty$ , whose linear Q-function satisfies

$$\|Q_{\theta_\infty} - Q^\pi\|_{D_\pi} \leq \frac{1}{1-\beta} \min_\theta \|Q_\theta - Q^\pi\|_{D_\pi}$$

Empirically on an  $11 \times 11$  gridworld, IS-TD( $\lambda$ ) quickly drives  $\theta$  to its true values and exhibits dramatically lower steady-state error than the naïve full-episode IS method. This methodology incorporates value approximation methods and importance sampling into a robust method. Even though it is off policy paper it is important paper as importance sampling is used for many algorithms such as Conservative Q learning and Implicit Q-learning.

<b>Strengths</b>	<b>and</b>	<b>Weaknesses:</b>
One of the main strengths of this work is that it tackles a real-world problem, not just a theoretical issue. It addresses something that actually happens when trying to use off-policy learning with function approximation. Before this paper, people didn't really understand why off-policy learning with function approximation often didn't work well. It was like a mystery, and many people struggled to apply it in real situations. This paper not only explained why it was failing but also offered a solution using importance sampling. The authors showed, step by step, how importance sampling can fix the problem by adjusting for the differences between the behavior policy and the target policy. They also provided the math behind the approach, which helped make it easier to understand and apply. What's really helpful is that they proved importance sampling works better than not using any corrections at all, making it a practical solution for real-world applications in reinforcement learning.		

However, while the importance sampling technique is an important improvement, it's not perfect. One of the problems with it is that when the target policy is very different from the behavior policy, the ratio used in importance sampling can become extremely large. This causes something called "high variance," which means that the learning updates can become very unstable. In simple terms, the updates bounce around too much, which makes the learning process slower and can even cause it to fail or behave unpredictably. This is still a problem that wasn't completely solved in the paper. Additionally, the paper mostly focuses on linear function approximation, which was a common method at the time. However, today, many people use more advanced methods, such as neural networks, to handle more complex problems. These modern methods were not fully explored in this paper. Despite these issues, the paper is still very valuable. It provided a major contribution to the field by showing how off-policy methods could be made more reliable. It also helped make offline reinforcement learning (RL) more practical, especially in situations where the data cannot be changed, such as when learning from fixed datasets. This paper laid the foundation for many future advances in the area, making it an essential step forward in the field.

### **Seminal Work 3: Jiang and Li (2016) Doubly Robust Off-policy Value Evaluation for Reinforcement Learning [3]**

Jiang and Li wrote this paper in 2016 to solve the problem of off-policy evaluation. That means trying to guess how good a policy is, without actually using it in the environment. This is really important for offline RL, where you only have old data and can't test a new policy safely. Before this, people were using importance sampling or model-based methods to do the evaluation, but both had problems. IS is unbiased but has super high variance, and model-based methods are low variance but biased if the model is wrong.

So they proposed a new method called Doubly Robust (DR) estimation. It mixes both ideas. You use the model to guess the return, and then use IS to correct the guess. If either the model or the importance weights are right, then the final estimate is still okay. This makes it more stable than either method on its own. The methodology is implemented in three parts for each collected prior

trajectories with length of H. Fit a  $\hat{Q}$  function from the existing trajectories. Then, we need to define per-step ratio which is similar to importance sampling.

$$\rho_t = \frac{\pi_{target}(a_t|s_t)}{\pi_{behavior}(a_t|s_t)}$$

Then using the per-step ratio and initialization of  $V_{DR}(0) = 0$  calculate  $V_{DR}(t)$  as

$V_{DR}(t) = \sum_a \pi_{target}(a|s_t)\hat{Q}(s_t, a) + \rho_t[r_t + \gamma V_{DR}(t-1) - \hat{Q}(s_t, a_t)]$  where  $s_t$ ,  $a_t$  and  $r_t$  are the state action and reward for a trajectory at timestep t. The term  $\rho_t[r_t + \gamma V_{DR}(t-1) - \hat{Q}(s_t, a_t)]$  is a correction term that Decrease the bias if estimation of  $\hat{Q}$  is incorrect and  $\rho_t$  is correct which makes the evaluation more robust.

By using  $\hat{V}_{DR} = \frac{1}{N} \sum_{i=1}^N V_{DR}^{(i)}$   $(H) = \frac{1}{N} \sum_{i=1}^N V_{DR}^{(i)}$  and  $\sigma_{dr} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\hat{V}_{DR} - V_{DR}^{(i)})^2 / \sqrt{N}}$ , find the best policy that maximizes the lower confidence bound which is defined as

$$LCB(\pi_k) = \hat{V}_{DR} - C\sigma_{dr}(\pi_k)$$

And the policy that maximizes LCB is the target policy that we want. In benchmarks (Mountain Car, Sailing, KDD-98), DR-LCB accepted good policies faster and rejected bad ones just as reliably as IS, achieving both aggressive improvement and safety in fully offline RL.

The paper gave both math proofs and experiments showing that DR is a better way to evaluate policies when you have a fixed dataset. It's now used in many offline RL papers that need to measure how well a new policy might do before actually trying it out.

<b>Strengths</b>	<b>and</b>	<b>Weaknesses:</b>
A major strength of this paper is that it solved one of the biggest problems in offline reinforcement learning (RL)—how to safely evaluate a policy without causing harm. The method it introduced, known as the <i>DR method</i> (Doubly Robust), is clever because it doesn't completely depend on either the model or the importance weights alone. Instead, it combines both in a way that helps balance out mistakes made by one part. This is especially important for real-world applications like healthcare, where testing a bad policy could have serious or even dangerous consequences. The paper does a good job of explaining the method clearly, providing both theoretical insights and practical experimental results. This makes it easier to understand and apply, especially in high-risk areas where safety is a concern.	and	Weaknesses:

However, there are some weaknesses. If both the model and the importance weights are inaccurate, the estimates produced by the DR method will also be unreliable. This is a problem because it assumes that even if one part of the method is wrong, the other part can compensate for it. But if both parts are wrong, then the whole estimate fails. Another issue is that the method assumes we can get a reasonably good estimate of the value function directly from the data. In real-world environments, especially complex ones, this is not always the case, and the paper doesn't go into detail on how to handle situations where it's difficult to get an accurate estimate. Additionally, when the importance weights are extremely large, the DR method can still suffer from high variance, making the learning process unstable or slow. Despite these limitations, compared to other methods available at the time, this approach was a significant improvement. It provided a much-needed step forward in making offline RL safer and more reliable, and it has been very useful for research in the field.

## 4. State-of-the-Art Research

### State-of-the-Art Work 1: Fujimoto et al. (2019) — Off-Policy Deep Reinforcement Learning without Exploration [4]

Fujimoto et al. (2019) proposed Batch-Constrained Q-Learning (BCQ) to address a fundamental problem in offline reinforcement learning: extrapolation error. When using off-policy deep RL algorithms like DQN or DDPG with a fixed dataset, learned Q-functions tend to assign overly optimistic values to actions that are not well-represented in the data. This leads the policy to favor out-of-distribution (OOD) actions and ultimately results in poor decision-making. BCQ aims to mitigate this by restricting the learned policy to select only those actions that are likely under the behavior policy that generated the dataset.

The core idea behind BCQ is to combine three components: a generative model to approximate the action distribution, a perturbation model to refine candidate actions, and a Q-network for evaluation. Specifically, BCQ learns a conditional generative model  $G_w(a|s)$ , typically implemented as a variational autoencoder (VAE), to generate plausible actions given a state. At evaluation time,  $M$  candidate actions  $\{a_1, a_2, \dots, a_M\}$  are sampled from  $G$ , and each action is perturbed slightly using a small transformation function  $\xi\phi(s, a)$ , producing:

$$a'_j = a_j + \xi\phi(s, a_j)$$

The final policy selects the action with the highest estimated Q-value:

$$\pi(s) = \operatorname{argmax}_{j \in \{1, \dots, M\}} Q_\theta(s, a'_j)$$

The Q-function  $Q_\theta$  is trained using a standard Bellman regression objective, where the target value is:

$$y_i = r_i + \gamma \cdot Q_\theta(s'_i, \pi(s'_i))$$

The loss minimized over the dataset  $D$  is:

$$L(\theta) = (1/N) \sum_{i=1}^N (Q_\theta(s_i, a_i) - y_i)^2$$

To further reduce overestimation bias, BCQ employs Clipped Double Q-learning, using two Q-networks and taking the minimum of their outputs during target calculation.

Empirical evaluations on continuous control benchmarks (e.g., MuJoCo environments) show that BCQ outperforms standard off-policy methods and remains robust even when the dataset is limited or collected under suboptimal policies. It laid foundational groundwork for subsequent methods like CQL and BRAC.

Strengths	and	Weaknesses:
BCQ's strength lies in its principled response to extrapolation error, one of the central challenges in offline RL. By constraining the learned policy to remain close to the support of the training data, BCQ avoids unsafe or unsupported actions that lead to overestimated Q values. The use of a generative model enables flexible sampling of plausible actions, while the perturbation model introduces just enough variability to enable local exploration. This combination allows BCQ to maintain a delicate balance between safety and performance. Furthermore, the modular design consisting of a VAE for action generation, a perturbation network, and a Q-network makes BCQ extensible and adaptable for other settings. Its empirical results validate its robustness across several tasks and dataset types.		

However, BCQ also introduces complexity and new sources of potential failure. The performance of the method depends heavily on the quality of the generative model  $G_w$ . If the VAE fails to capture the true action distribution, the policy may be overly conservative or even degenerate. Hyperparameter tuning is also non-trivial: the number of candidate actions  $M$ , the strength of perturbation, and the architecture of the VAE all influence performance. In addition, training three networks (VAE, perturbation model, and Q-network) increases computational cost and implementation complexity compared to simpler methods like CQL. Another limitation is that by design, BCQ avoids exploring actions outside the dataset entirely even in situations where limited extrapolation could yield better policies. This conservative nature can result in suboptimal performance when the dataset is narrow or biased. Nonetheless, BCQ was one of the first practical offline RL methods to explicitly constrain policy behavior and has inspired a significant body of followup work.

### **State-of-the-Art Work 2 : Kumar et al. (2020) — Conservative Q-Learning (CQL) [5]**

Conservative Q-Learning (CQL), introduced by Kumar et al. (2020), addresses one of the most pressing issues in offline reinforcement learning—extrapolation error. In offline settings, the agent cannot interact with the environment to collect new data. Instead, it must learn entirely from a fixed dataset generated by a prior behavior policy. This setup leads to a distribution mismatch between the state-action pairs observed in training and those evaluated during policy learning. Standard Q learning often overestimates the value of actions that are rarely or never seen in the dataset, known as out of distribution (OOD) actions, causing unreliable and potentially harmful policies.

To mitigate this, CQL proposes modifying the standard Q-learning loss function to penalize high Q-values for OOD actions. This is done by introducing a conservative regularization term alongside the Bellman error. The full loss function is given by:

$$L(Q) = \alpha \times [E_{\{(s, a) \sim \mu\}} Q(s, a) - E_{\{(s, a) \sim \pi_\beta\}} Q(s, a)] + (1/2) \times E_{\{(s, a, s') \sim D\}} [(Q(s, a) - B^\pi Q(s, a))^2]$$

In this expression,  $\mu(a|s)$  represents a sampling distribution that includes a broader range of actions, often uniformly sampled.  $\pi_\beta$  is the behavior policy that generated the dataset, and  $B^\pi$  is the Bellman operator. The hyperparameter  $\alpha$  controls the degree of conservatism applied during learning. By penalizing large Q values for unlikely actions while maintaining the standard Bellman update, CQL learns value functions that act as conservative lower bounds. As a result, the learned policies avoid relying on unsupported or risky actions. Experimental results show that CQL achieves strong performance on benchmark suites such as D4RL, including Atari games, MuJoCo control tasks, and Adroit robotic manipulation. It often outperforms earlier offline RL algorithms such as BCQ and BRAC, particularly in environments with poor data coverage or suboptimal demonstrations.

<b>Strengths</b>	<b>and</b>	<b>Weaknesses:</b>
One of the core strengths of CQL lies in its simplicity and ease of adoption. It does not require additional models or complex architectures but instead modifies the existing Q learning objective with a well-motivated regularization term. This makes it compatible with standard deep RL implementations and computationally more efficient than methods that rely on generative models or divergence estimation. CQL is also grounded in strong theoretical guarantees. The conservative penalty ensures that Q-values are not overestimated in regions of the state-action space that are poorly covered by the dataset, which stabilizes learning and helps avoid unsafe policy behaviors. Empirically, CQL performs reliably across both continuous and discrete tasks, making it broadly applicable.		

However, the method also has notable limitations. Choosing the right value of the conservatism coefficient  $\alpha$  is crucial if  $\alpha$  is too high, the policy becomes excessively conservative and may ignore infrequent but valuable actions. If  $\alpha$  is too low, the Q-function can still suffer from overestimation. Furthermore, evaluating the expectation over the sampling distribution  $\mu$  can be computationally intensive, particularly in high-dimensional or continuous action spaces. This may require approximate

sampling or assumptions that limit scalability. Finally, while CQL effectively addresses extrapolation error, it does not solve other key challenges in offline RL, such as sample inefficiency or generalization to unseen states. These areas remain open for future research and improvement.

### **State-of-the-Art Work 3: Wu et al. (2019) — BRAC+: Improved Behavior Regularized Actor Critic for Offline Reinforcement Learning [6]**

A major challenge in Offline RL is the distributional shift between the behavior policy and the learned policy, which can result in overestimation of out-of-distribution actions and propagate errors through Bellman updates. To mitigate this, prior works use behavior regularization, encouraging the learned policy to stay close to the behavior policy using divergence measures such as KL divergence or Maximum Mean Discrepancy (MMD).

Behavior Regularized Actor Critic (BRAC) solves the offline RL problem by adding a constraint to the policy optimization step. The goal is to maximize expected Q-values under the learned policy, while ensuring the learned policy stays close to the behavior policy, which generated the dataset. This is formulated as:

$$E[s] \sim D[Q(s, \pi\theta)] \quad E[s] \sim D[D(\pi\theta(\cdot|s) \| \pi_b(\cdot|s))] < \epsilon$$

This paper proposes BRAC+, an enhanced version of the Behavior Regularized Actor Critic (BRAC) algorithm. It highlights the weaknesses in existing methods, such as MMD's failure under multi-modal distributions and the high variance in KL divergence estimation. BRAC+ introduces an analytical upper bound on KL divergence for stable regularization and adds a gradient penalty to ensure Q-value convergence. Experiments show BRAC+ improves performance significantly on the D4RL benchmark -a standardized set of datasets and environments designed to evaluate and compare Offline RL algorithms- outperforming prior methods by 40–87% and state-of-the-art approaches by 6%.

Kernel MMD was introduced to offline RL as a way to regularize the learned policy so that it stays close to the behavior policy. MMD measures the distance between two distributions. The squared MMD between the learned policy and the behavior policy is computed as:

$$MMD^2_k(\pi(\cdot|s), \pi_b(\cdot|s)) = E_{x,x'}[\pi(\cdot|s) [K(x, x')]] - 2 \cdot E_{x\pi(\cdot|s), y\sim\pi_b(\cdot|s)}[K(x, y)] + E_{y,y'}[\pi_b(\cdot|s) [K(y, y')]]$$

This approach can effectively regularize learned policies when the dataset is collected from a single-modal behavior policy. However, in multi-modal settings, MMD may fail to penalize out-of-distribution actions. This is because the learned policy might minimize MMD by placing its probability mass between modes, which results in low-density actions that were rarely or never taken in the dataset which is precisely the kind of behavior offline RL tries to avoid. As a result, in such cases, MMD can encourage out-of-distribution actions rather than discourage them, leading to performance degradation despite the presence of a regularization term.

KL Divergence is a widely used approach to measure the difference between the two policies using Kullback-Leibler (KL) divergence. KL divergence quantifies how one probability distribution diverges from another and is formally defined as:

$$D_{\text{KL}}(P || Q) = \int_{x \sim \mathcal{X}} P(x) \cdot \log [P(x) / Q(x)] dx$$

However, estimating KL divergence from samples introduces high variance and can be computationally expensive, especially when the behavior policy is implicit or when large batches of actions must be evaluated.

Analytical KL divergence is used to address the high variance and computational inefficiency of sampling-based KL divergence estimation, an analytical upper bound on the KL divergence between the learned policy and the behavior policy was proposed.

Assuming the behavior policy is modeled using a Conditional Variational Autoencoder (CVAE) with latent variable  $Z$ , the Evidence Lower Bound (ELBO) provides a lower bound on the log-likelihood of actions:

$$\log \pi_b(a | s) \geq E_{\{z \sim q(z | s, a)\}} [\log p(a | s, z)] - D_{KL}(q(z | s, a) || p(z))$$

This leads to an upper bound on the KL divergence:

$$\begin{aligned} D_{KL}(\pi_\theta(a | s) || \pi_b(a | s)) &= E_{\{a \sim \pi_\theta\}} [\log (\pi_\theta(a | s) / \pi_b(a | s))] \\ &\leq E_{\{a \sim \pi_\theta\}} [\log \pi_\theta(a | s) - E_{\{z \sim q(z | s, a)\}} [\log p(a | s, z)] + D_{KL}(q(z | s, a) || p(z))] \end{aligned}$$

While behavior regularization constrains the learned policy to remain close to the behavior policy, it is not sufficient on its own to prevent Q-function overestimation during offline RL. To address this, Gradient Penalized Policy Evaluation in BRAC+ introduces a gradient penalty to the Q-function objective. The penalty is based on the norm of the gradient of the Q-function with respect to the action, scaled by a divergence-aware weighting function. This penalizes large gradients in regions where the learned policy deviates significantly from the behavior policy.

$$\begin{aligned} \text{minimize over } \psi: \quad & E_{\{(s, a, s', r) \sim \mathcal{D}\}} [(Q\psi(s, a) - [r \gamma \cdot E_{\{a' \sim \pi_\theta\}} [Q\psi'(s', a')]]^2)] \\ & + \lambda \cdot E_{\{a'' \sim \pi(\cdot | s)\}} [\|\nabla_{a''} Q\psi(s, a'')\|^2] \cdot f(D_{KL}(\pi_\theta || \pi_b)) \end{aligned}$$

This penalty ensures the existence of a local maximizer, stabilizing training and promoting convergence.

### Strengths and Weaknesses:

Despite improvements from behavior regularization and gradient penalty, these methods remain insufficient in addressing all offline RL challenges, particularly due to their ignorance of the state distribution. The author argues that such methods can fail in scenarios where the dataset consists of a mix of expert and suboptimal trajectories. For example, if expert demonstrations visit valuable states not encountered by the low-quality behavior policy, a regularized approach will tend to imitate the behavior policy, and may not explore or generalize toward high-value regions of the state space. This results in compounding errors, as the learned policy lacks coverage over important parts of the environment and is unable to reliably combine suboptimal experiences into a superior policy.

### State-of-the-Art Work 4: Gulcehre et al. (2021) — Regularized Behavior Value Estimation (R-BVE) [7]

Standard Q-learning methods aim to estimate the optimal  $Q^*$  with SARSA update. Consequently, applying policy improvement and policy evaluation steps. The main problem with that method is that the policy improvement involves the Bellman max-operator, which amplifies approximation errors when applied to one of the out-of-distribution actions and can lead to divergence. R-BVE breaks this loop by learning only the behavior policy's value function  $Q^{\pi_B}$  during training and enforcing a ranking constraint that penalizes extrapolated values. Policy improvement is deferred in the training process, and only one policy improvement step is performed with greedy updates at deployment. Empirically, R-BVE attains state-of-the-art performance on the RL Unplugged Atari suite and demonstrates robustness on bsuite and DeepMind Lab benchmarks. The R-BVE consists of two main parts: Behavior Value Estimation (BVE) and Ranking Regularization (R).

Behavior value estimations aim to eliminate the dependency of  $\pi$  on  $Q_\theta$  and remove the policy improvement step in the training. This is done by applying the given equation to transitions.

$$\mathcal{L}_{\theta'}(\theta) = \mathbb{E}_{(s,a,r,s',a') \sim \pi} (Q_\theta(s,a) - r - \gamma Q_{\theta'}(s',a'))^2$$

At deployment, BVE applies exactly one greedy improvement to the bound extrapolation error.

Ranking Regularization after estimating Q applies a squared-hinge margin  $\nu$  that penalizes any unobserved action whose predicted value exceeds that of the dataset actions, weighted by each transition's normalized return

$$w(s) = \exp((V^{\pi_B}(s) - \mathbb{E}_{s \sim \mathcal{D}}[V^{\pi_B}(s)])/\beta)$$

$$\mathcal{C}(\theta) = w(s) \sum_{i=0, i \neq t}^{|\mathcal{A}|} \max(Q_\theta(s, a_i) - Q_\theta(s, a_t) + \nu, 0)^2$$

#### Strengths and Weaknesses:

In practice, R-BVE handles the bootstrapping error that offline Q-learning suffers from highly by eliminating policy improvement during training. Instead, the Q-network is trained exclusively on logged transitions, so its estimates remain close to the support of the behavior data and cannot explode on unseen actions. This action dramatically improves stability as seen in Atari and Bsuit experiments. On top, ranking regularization forces the values of actions observed in high-return trajectories to exceed those of all others by a fixed margin, focusing learning on samples originally in the data set. Because both changes reside purely in the loss function, R-BVE can be implemented on existing architecture with no required modification, making R-BVE straightforward to integrate into any deep Q-learning codebase. R-BVE yields state-of-the-art results (median normalized scores of 109 % on the RL Unplugged Atari suite) and robust performance on bsuite and DeepMind Lab, even when data is scarce or noisy.

That said, R-BVE also presents new trade-offs. It depends on two hyperparameters (the margin  $\nu$  and scale  $\beta$ ) that must be tuned per domain to avoid under-/over-regularizing. The ranking regularizer similarly employs a discrete action set, and extending it to continuous spaces would require action sampling or relaxation methods, making implementation and optimization harder. Although adequate in numerous scenarios, the one greedy boost of the model at deployment potentially underuses pervasive suboptimal behavior policies, hence capping attainable returns. Additionally, although negative sampling lowers incremental cost, using a hinge loss on all discrete actions still incurs prohibitive overhead in settings characterized by large action spaces.

## 5. Open Problems and Future Research Directions

Despite its success, the proposed techniques do not address all significant issues. Offline reinforcement learning deployments in real world environments often face challenges due to low quality data and outliers due to noise. Logs from uncontrolled environments can have mislabeled transitions and sparse outliers, which misleads value estimates and hinder the policy's learning. Future work should aim to construct robust training objectives and frameworks that automatically detect and downweight or correct erroneous samples. Incorporating ideas from robust statistics, such as adaptive M-estimators, trimmed-mean updates, or consensus-based filtering, into offline One significant restriction is the inefficient use of available offline trajectories. Many current algorithms do not take advantage of auxiliary information or exploit the action-state space's inherent structures. More advanced representation learning techniques that obtain task specific embeddings from offline data can improve sample efficiency. Meta learning approaches, which take advantage

of similar tasks to obtain transferable prior knowledge, enable adaptation to new environments with low additional data requirements. Also, the uninterpretable nature of most offline RL policies creates difficulties in validating, auditing, or certifying them. There is a need to build methods for extracting interpretable summaries of policies using tools like decision trees, program synthesis, or attention mechanisms highlighting key state features. Other tools that measure a trained policy's dependence on specific inputs or visualize its decision boundaries can be important for building user trust and meeting regulatory requirements. These recommendations could significantly improve the performance of offline RL in real-world scenarios.

## 6. Conclusion

Offline reinforcement learning enables agents to learn from fixed datasets without interacting with the environment, making it very useful in domains where real-time environment interaction is expensive, dangerous, or infeasible. A key challenge is the distributional shift between the behavior and learned policies, which can lead to overestimation and poor generalization. Recent methods like BCQ, CQL, and BRAC+ address this through behavior regularization and conservative updates. Among them, BRAC+ shows strong performance on the D4RL benchmark by balancing policy learning with distributional constraints. While progress is evident, challenges remain –especially in handling limited or low-quality data.

## 7. References

- [1] T. Ernst, D. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, pp. 503-523, 2005.
- [2] D. Precup, R. S. Sutton, and S. Singh, "Off-policy temporal difference learning with function approximation," *Proceedings of the 18th International Conference on Machine Learning*, pp. 417-424, 2001.
- [3] Y. Jiang and L. Li, "Doubly robust off-policy evaluation for reinforcement learning," *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, pp. 1756-1765, 2016.
- [4] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019
- [5] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems*, 2020
- [6] C. Zhang, S. R. Kuppannagari, and V. K. Prasanna, "BRAC+: Improved behavior regularized actor critic for offline reinforcement learning," *arXiv preprint arXiv:2110.00894*, 2021. [Online]. Available: <https://arxiv.org/abs/2110.00894>
- [7] C. Gulcehre et al., "Regularized behavior value estimation," *arXiv preprint arXiv:2103.09575*, 2021. [Online]. Available: <https://arxiv.org/abs/2103.09575>