Effective Offline RL Needs Going Beyond Pessimism: Representations and Distributional Shift

Anonymous Author(s)
Affiliation
Address
email

Abstract

Standard off-policy reinforcement learning (RL) methods based on temporal difference (TD) learning generally fail to learn good policies when applied to static offline datasets. Conventionally, this is attributed to distribution shift, where the Bellman backup queries high-value out-of-distribution (OOD) actions for the next time step, which then leads to systematic overestimation. However, this explanation is incomplete, as conservative offline RL methods that directly address overestimation still suffer from stability problems in practice. This suggests that although OOD actions may account for part of the challenge, the difficulties with TD learning in the offline setting are also deeply connected to other aspects such as the quality of representations of learned function approximators. In this work, we demonstrate that merely imposing pessimism is not sufficient for good performance, and demonstrate empirically that regularizing representations actually accounts for a large part of the improvement observed in modern offline RL methods. Building on this insight, we identify concrete metrics that enable effective diagnosis of the quality of the learned representation, and are able to adequately predict performance of the underlying method. Finally, we show that a simple approach for handling representations, without any changing any other aspect of conservative offline RL algorithms, can lead to better performance in several offline RL problems.

1 Introduction

Offline reinforcement learning (RL), combined with powerful deep net function approximators, has the potential for solving decision-making tasks where online interaction is either expensive or unsafe, circumventing a major barrier to the deployment of RL in the real-world. Temporal difference (TD) learning methods, such as Q-learning, provide a natural framework for building offline RL algorithms [30], fitting a parametric value function by sequentially regressing to targets generated from its own previous snapshot using only offline data. However, directly applying TD to a static offline dataset often fails to learn effective policies, as the maximization in the target value computation will find erroneously high-valued out-of-distribution (OOD) actions, resulting in systematic overestimation. A variety of offline RL methods, such as those that apply value conservatism [26, 58] or behavioral constraint [14, 24, 53, 13, 18, 23, 22], have been proposed to address this issue with OOD actions in TD learning by inducing some form of pessimism. While all these methods lead to promising improvement in performance on offline RL tasks, determining why one method for addressing the OOD actions issue is better than another has proven challenging, which in turn makes it difficult to develop insights and guidelines for designing better offline RL algorithms. In fact, in theory, majority of these approaches essentially optimize the very same RL objective subject to a divergence constraint against the behavior policy that generates the data, and

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

would, behave identically in a tabular problem setting. Hence, a natural question to ask is: does the improvement observed from these methods really stem from their ability to induce pessimism?

In this paper, we will show that a significant part of the benefit of offline RL approaches that aim to address OOD actions actually stems from the effect they have on the learned representations, rather than merely from their ability to avoiding overestimation. We first show that the even if we can prevent the value of the learned Q-function at OOD actions from being overestimated, training Q-functions against Bellman targets computed using OOD actions still induces Q-function representations that give rise to poor policy performance, which indicates that overestimation is not sufficient to explain poor performance in offline RL. Second, we empirically demonstrate that an offline RL method that does not apply any pessimism, but only regularizes the representation learned for the dataset and OOD actions to be different using adversarial training, can actually perform quite well. The method we develop resembles the conservative Q-learning (CQL) [26] approach, but crucially only regularizes the representations and not the final Q-values. Our analysis shows that this approach recovers 68% of the performance of CQL, indicating that the performance of CQL, in large part, comes from the implicit regularization obtained by penalizing OOD actions.

Based on this analysis, we propose a metric that evaluates the quality of the representation learned by offline RL methods based on the ability to accurately reconstruct the dataset actions from the learned representation. We demonstrate that comparing this reconstruction error to a dynamic programming approach that does not utilize OOD actions gives us a good measure of representational quality, that is predictive of performance. Finally, we discover that good representations can actually be obtained by a surprisingly simple method: interpolating between TD and supervised learning via an ensemble of N-step returns, similar to TD(λ). We not only find that utilizing an ensemble of N-step returns approach attains better performance, but, more interestingly, we argue that this *cannot* be attributed to standard explanations of a better bias-variance tradeoff.

Our main contributions are to demonstrate, via an extensive empirical study, that merely addressing the OOD action issue in offline RL via pessimism is not sufficient for TD-based offline RL methods, and that the quality of learned representation is crucial for good performance. Our analysis provides guidance on how to measure representational quality, and shows how simple methods such as an ensemble of N-step returns already attain better performance on benchmark tasks from D4RL [12] as a result of improved representational quality. We hope that our analysis provides concrete takeaways for researchers in offline RL and highlights a largely overlooked line of challenges beyond behavior regularization that is crucial in devising more effective and reliable offline RL methods.

2 Related Work

Modern offline RL methods based on Q-learning typically utilize dynamic programming to train a value function together with a mechanism to prevent backing up out-of-distribution (OOD) actions [30]. This can be done by applying an explicit constraint that forces the learned policy to be "close" to the behavior policy under a variety of divergence measures [18, 54, 37, 42, 54, 24, 23, 22, 50, 13], or by directly learning a conservative value function, either via a pessimistic training objective [26, 56, 36, 58] or by utilizing pessimistic bonuses [57, 39, 19, 54] in the backup. Other offline methods include model-based methods [20, 57, 2, 45, 38, 29, 58] that also utilize rollouts under a learned dynamics model to train the value function while also avoiding out-of-distribution actions. While most of these methods differ from each other in implementation details and empirical performance, in theory and in tabular problem settings, most of these methods can be traced back to the same objective that attempts to constrain the policy from choosing OOD actions. It is not entirely clear why one method should work better than another, or how one should go about designing better offline RL methods. In this paper, we show that, to a large extent, the benefits of offline RL methods comes from better representational quality, and how improving representational quality alone can lead to reasonable performance without any form of pessimism.

Prior works have sought to analyze several aspects of the representations induced by TD-based methods with function approximation largely in the standard online RL setting [1, 5, 25, 48, 31, 32] and in the offline RL setting [28, 27]. In the linear setting, [15, 55], study which representations can induce stable convergence of TD and [44, 33] have tried to devise convergent TD methods for arbitrary representations, but these prior works do not attempt to study the effect of pessimism on representations, or how OOD actions affect representations. Recent work [27, 28] study the learning dynamics of Q-learning in an overparameterized setting and observes excessively low-rank and aliased feature representations at the fixed points found by TD-learning. These prior works propose some metrics to evaluate representational quality, and we do evaluate these in our analyses in Section 5, but find that these metrics generally behave well, even though performance can be improved with simple representational regularization. As we show, the metric we propose is more predictive of algorithm performance. Moreover, these prior works do not quite study the interplay between pessimism and representations that we do.

Finally, we note that our proposed approach of utilizing an ensemble of N-step returns is not new. Most notably, it is related to TD(λ) which has been instantiated in various forms [41, 21, 51, 9]. Prior works have also used N-step returns for a fixed value of N in methods that perform off-policy TD learning [49, 17, 10]. Besides the fact that most of these works are based in an online RL setting, the crucial distinction behind these prior works and our paper is that our work goes beyond the standard explanation of bias-variance tradeoff for N-step returns [40], and analyzes N-step returns from a different perspective: improving the quality of learned representations. We emphasize that our goal is not to produce a novel algorithm, but rather to understand the efficacy of different components towards the representations learned by the Q-function.

3 Preliminaries

The RL problem is formally defined by a Markov decision processes (MDPs) defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \mu_0, \gamma)$, where $\mathcal{S}, \mathcal{A}$ denote the state and action spaces, and $T(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, $r(\mathbf{s}, \mathbf{a})$ represent the dynamics and reward function respectively. $\mu_0(s)$ denotes the initial state distribution, and $\gamma \in (0, 1)$ denotes the discount factor. The objective of RL is to learn a policy that maximizes the return (discounted sum of rewards): $\max_\pi J(\pi) := \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t)\sim\pi}[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$. In offline RL, we are provided with an offline dataset, $\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')\}$, of transitions collected using a behavior policy π_β, and our goal is to find the best possible policy only using the given dataset.

Directing training a Q-value function from the offline dataset often suffers from OOD actions [14, 24, 30], and therefore effective offline RL algorithms must enforce some constraint to prevent querying the target Q-function on unseen actions. This constraint could be a behavior constraint, where the learned policy π is constrained to be close to the behavior policy π_β. In this work, we build our analysis on top of conservative Q-learning (CQL) [26], which applies a regularizer $\mathcal{R}(\theta)$ to prevent overestimation of Q-values for OOD actions. $\mathcal{R}(\theta)$ minimizes the Q-values under the policy $\pi(\mathbf{a}|\mathbf{s})$, and counterbalances this term by maximizing the values of the actions in $\mathcal{D}$. Formally:

$$\min_\theta \alpha \left(\mathop{\mathbb{E}}_{\mathbf{s}\sim\mathcal{D}, \mathbf{a}\sim\pi} [Q_\theta(\mathbf{s}, \mathbf{a})] - \mathop{\mathbb{E}}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}} [Q_\theta(\mathbf{s}, \mathbf{a})] \right) + \frac{1}{2} \mathop{\mathbb{E}}_{\substack{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D} \\ \mathbf{a}'\sim\pi}} \left[\left(Q_\theta(\mathbf{s}, \mathbf{a}) - r - \gamma \bar{Q}(\mathbf{s}', \mathbf{a}') \right)^2 \right], \quad (1)$$

where $\bar{Q}$ denotes the target Q-function. On the other hand, training a Q-value function for the behavior policy, that only relies on action samples from the offline dataset is fairly easy and does not suffer from the problem of OOD actions. A standard approach of learning such a Q-function is what we refer to as "offline SARSA" [43], which only queries the action observed in the dataset at the subsequent timestep to compute the Bellman target for training the Q-function. The objective for SARSA can be written as:

$$\min_\theta \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}',\mathbf{a}'\sim\mathcal{D}} \left[\left(Q_\theta(\mathbf{s}, \mathbf{a}) - r - \gamma \bar{Q}(\mathbf{s}', \mathbf{a}') \right)^2 \right]. \quad (2)$$

Since the next step Q-values are computed using dataset actions, it eliminates the need to query Q-function for the values of any OOD actions. In effect, this procedure only relies on supervision observed in the dataset (i.e., actions, the corresponding rewards and the next states) to learn representations. Prior works [28] have argued that avoiding out-of-distribution actions altogether enables SARSA to enjoy benefits of implicit regularization [52, 3] that otherwise may hurt TD learning.

In order to understand representational quality, we focus our analysis on the last layer feature representation $\phi(\mathbf{s}, \mathbf{a})$ learned by the neural network, following the conventions in prior work [8, 28, 27, 31, 32]. These prior works have also attempted to show that certain characteristics of the learned representations $\phi(\mathbf{s}, \mathbf{a})$ of a value network can explain certain pathologies with Q-learning.

4 To What Extent Do OOD Actions Explain the Instability in Offline RL?

Most prior works in offline RL focus on addressing the action distribution shift problem, proposing a wide variety of methods in preventing the policies from taking OOD actions during the training process. However, it remains unclear why different methods for mitigating OOD actions seem to attain significantly different performance, and whether being *better* at preventing OOD actions is actually the key to better results. It therefore seems natural to ask: to what degree is good (or bad) performance of offline RL approaches really dependent on their ability to be pessimistic? In this section, we study this question by performing a controlled empirical study. We perform experiments to investigate both the sufficiency and necessity of being pessimistic and present them next.

4.1 Is Pessimism Sufficient for Good Performance?

While several recent offline RL methods that correct for OOD actions by adding some form of pessimism work well, in most of these approaches, the pessimism-inducing penalty (e.g., value conservatism penalty like in CQL) or constraint (e.g., behavioral constraints) also affects the representation learned by the internal layers of the Q-function (or the policy). In this section, we argue via an empirical study on top of the CQL algorithm that, to a large extent, the benefits of this pessimism-inducing mechanism stem from its impact on the learned representation and not so much from its ability to combat overestimation.

Empirical results showing insufficiency of pessimism. To decouple the effects of pessimism in handling overestimation and representational quality, we train a CQL [26] agent on the hopper-medium-replay-v2 environment from the D4RL [11] suite, and make the following modification: we let the last layer representation $\phi(s, a)$ of the Q-network be updated by the TD-error (second term in Equation 1) and the conservatism regularizer ($\mathcal{R}(\theta)$) is **not** allowed to affect this representation. That said, this regularizer $\mathcal{R}(\theta)$ is allowed to affect the final layer weights of the Q-function. As a result, while the CQL regularizer can still curb overestimation by manipulating the last layer Q-values, it is unable to affect the representations, thereby inhibiting pessimism from providing any representational benefits. For comparison, we also train a regular CQL agent on the same environments. For both runs, we apply the same weight on the conservatism penalty.

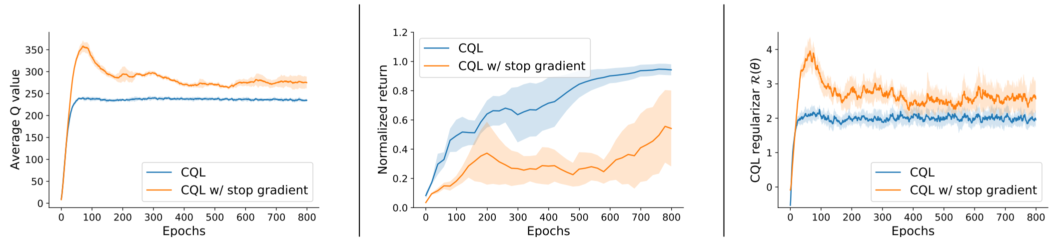

Figure 1: **CQL w/ stop gradient vs CQL** in hopper-medium-replay task. **Left**: CQL w/ stop gradient is able to prevent overestimation and results in non-divergent Q-values. **Middle**: the performance of CQL w/ stop gradient is significantly lower than regular CQL. **Right:** Values of the CQL regularizer are quite comparable between CQL and CQL w/ stop gradient, even though the observed performance is quite different.

As shown in Figure 1, once we prevent the CQL conservatism penalty from affecting the representation, the performance decreases significantly. In the left part of the figure, we see that when the CQL regularizer is not allowed to affect the learned Q-function representations (denoted "CQL w/ stop gradient"), we are still able to attain stable and non-divergent Q-values, thereby avoiding the issues typically observed with standard TD methods. However, CQL w/ stop gradient performs significantly worse than base CQL (Figure 1, middle). As shown in Figure 1 (right), the value of the CQL regularizer (i.e., the amount of pessimism) is still quite comparable in both cases, differing only by about 0.5, which is quite small relative to the average magnitude of the learned Q-values (~ 300), however there is a significant performance difference. This difference indicates that while pessimism might be beneficial in lowering the value of OOD actions, it also contributes significantly to other factors such as representation learning, and this representation learning benefit accounts for much of the improvement from CQL, since without it the method performs much worse.

> **Takeaway 4.1.** *Besides preventing OOD actions, pessimism-inducing mechanisms in offline RL algorithms can also contribute to representation learning, and simply ensuring pessimism, without affecting representations might not be sufficient for good performance.*

4.2 How Much Performance Improvement Do Good Representations Account for?

While the above results suggest that pessimism alone does not account for full performance of of offline RL methods, and the quality of the learned representation has a crucial role to play in determining the performance of value-based offline RL, it is not quite clear how much performance do good representations account for, how much performance is accounted to by other factors and what even a good representation even means. In this section, we attempt to answer this question by construction: we perform an empirical study that completely removes any sort of pessimism, but applies a representational regularizer. We show that it is still possible to obtain reasonable performance if the learned representation is carefully regularized, despite the fact that the method we test has no explicit mechanism for ensuring pessimistic estimates for OOD actions or constraining the policy to remain in-distribution.

Experiment setup. As shown in Equation 1, the CQL regularizer ($\mathcal{R}(\theta)$ in Equation 1) pushes down the Q-value at OOD actions and pushes up the Q-value for in-distribution dataset actions. If this kind of a pessimism penalty truly induces beneficial representational regularization, a nature conjecture is that representations that trained to minimize just the CQL regularizer independently of the TD error must also be useful, and must contain enough information to distinguish dataset actions from OOD actions. On its own, the CQL regularizer (Equation 1) resembles the objective of the discriminator in generative adversarial networks (GAN) [16] which serves a similar function of distinguishing dataset examples from generated examples. Based on this intuition, in the next experiment, we construct an offline RL method that utilizes a GAN objective, but only to train a *separate* linear output head on top of the Q-function network, whereas the Q-values are simply trained to minimize TD error with no form of pessimism whatsoever. A schematic illustration of this approach is shown in Figure 3. Specifically, we adopt the least square GAN [34] objective due to its simplicity and stability. Concretely, let us denote the linear discriminator weight as $\mathbf{w}_d$, then given the Q-network representation $\phi_\theta(\mathbf{s}, \mathbf{a})$, our explicit regularization objective can be written as

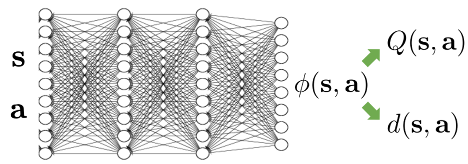

Figure 2: A schematic illustration of our approach for representational regularization that trains a Q-function with an auxiliary discriminator head for distinguishing potentially out-of-distribution policy actions from in-distribution dataset actions.

$$\min_{\theta, \mathbf{w}_d} \; \mathbb{E}_{\mathbf{s}\sim\mathcal{D}, \mathbf{a}\sim\pi}\left[(\phi_\theta(\mathbf{s}, \mathbf{a})^\top \mathbf{w}_d + 1)^2\right] + \mathbb{E}_{\mathbf{s}, \mathbf{a}\sim\mathcal{D}}\left[(\phi_\theta(\mathbf{s}, \mathbf{a})^\top \mathbf{w}_d - 1)^2\right]. \tag{3}$$

We apply this regularization on top of standard off-policy SAC [47], without any form of pessimism, and evaluate the algorithm in the same environment as Section 4.1. For comparison, we also train an nnaïve SAC agent with identical hyperparameters but without this second head.

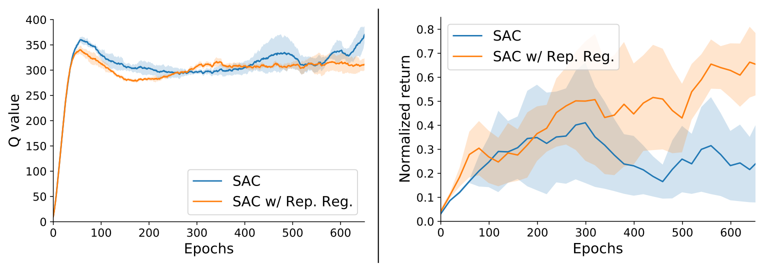

Figure 3: **SAC with representation regularization vs regular SAC** on hopper-medium-replay-v2 task. **Left**: SAC with representation regularization learns similar Q-values to regular SAC. **Right**: Representation regularization significantly improves the performance even without pessimism.

As shown in Figure 3, this modified algorithm can attain reasonable performance, significantly outperforming naïve SAC, despite having no explicit mechanism to ensure pessimism, conservatism, or policy constraints. Since the additional GAN term only influences the last layer representation, its benefits can be attributed entirely to learning better representations. While the method is not as effective as dedicated offline RL approaches such as CQL, this result, together with the experiment from Section 4.1 strongly suggests that representation learning is not only important for offline RL, but it also explains a large fraction of the performance gains for methods such as CQL. This in turn implies that, in designing better offline RL methods, we should put particular emphasis on their effect on representation learning, rather than simply on enforcing pessimism.

Takeaway 4.2. *The ability to learn good representations can explain a large fraction of the performance gains for practical offline RL methods. Explicit regularization techniques that gives good representations can be effective in offline RL, even in the absence of pessimism.*

5 What Constitutes a Good Representation for Offline RL?

Our empirical analysis from the previous section suggests that pessimistic offline RL methods do affect the representations learned by offline RL algorithms such as CQL, and utilizing only the TD error can give rise to representations that fail to adequately distinguish the dataset action from actions from the learned policy. This distinction is crucial: since an offline RL algorithm only observes ground truth supervision only in the form of instantaneous rewards and the subsequent environment state, for dataset actions, the ability to successfully associate the right (long-term) reward with the right dataset action is critical for attaining good performance. Can we formalize this intuition into a diagnostic metric for measuring the "goodness" of the learned representation?

The most natural choice of such a metric inspired by our experimental analysis in Section 4.2 is the accuracy of the separate discriminator head trained to distinguish dataset actions from policy actions. In our preliminary experiments, we find that while a discriminator accuracy near 50% is clearly indicative of poor performance, a reasonable discriminator accuracy (say $\geq 60-70\%$) does not necessarily indicate the absence of any representational issues. This is because while even a somewhat correct representation can attain high accuracy, the representation may still not be rich enough to match the fidelity needed for Q-value estimation. Therefore, we propose to utilize a more complete metric for tracking the extent of action information in the learned representation: we propose to train a non-linear model to reconstruct both the dataset and policy actions from the learned representation $\phi(\mathbf{s}, \mathbf{a})$, and suggest tracking the reconstruction error of this model in aggregate over dataset actions. This metric can be formalized as:

Metric 5.1. *Train a parametric model, $\Delta : \mathbb{S} \times \mathbb{R}^d \rightarrow \mathcal{A}$ on the dataset: $\mathcal{D}_\Delta := \mathcal{D}_\Delta^\pi \cup \mathcal{D}_\Delta^{\pi_\beta}$, where $\mathcal{D}_\Delta^{\pi_\beta} := \{(\mathbf{s}_i, \phi(\mathbf{s}_i, \mathbf{a}_i)), \mathbf{a}_i\}_{i=1}^N$ and $\mathcal{D}_\Delta^\pi := \{(\mathbf{s}_i, \phi(\mathbf{s}_i, \pi(\mathbf{s}_i))), \pi(\mathbf{s}_i)\}_{i=1}^N$. Then, track the error metric:*

$$\mathcal{L}_{\textit{recons}}(\Phi) := \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{s}_i, \mathbf{a}_i) \in \mathcal{D}} ||\mathbf{a}_i - \Delta(\mathbf{s}_i, \phi(\mathbf{s}_i, \mathbf{a}_i))||_2^2 . \tag{4}$$

Since the reconstruction error, $\mathcal{L}_{\text{recons}}(\Phi)$, can take on a range of values, how should we choose values to decide whether a representation is good enough or not? Specifically, what is a baseline value of this quantity that can be considered a "gold standard" for comparison? To identify a good value of this good standard, we seek to intuitively understand how OOD actions would impact the representations learned by a value-based offline RL algorithm. We can do so by utilizing the following informal model of the behavior of neural networks that is implied by several theories of deep learning [3, 4, 46, 7]: sufficiently expressive and overparameterized neural networks are believed to learn the "simplest" function that can fit the training data (i.e., match the actual label on the training datapoints). That is to say that the learned function retains only information about the training data that is absolutely critical for making predictions, and attempts to lose any unnecessary information.

When instantiated in the context of TD-learning, this intuitive model implies that the simplicity of the function approximator would depend on its ability to fit the Bellman constraints on the training data. If several of the actions used to compute Bellman targets are out-of-distribution, in principle, a simpler function approximator can be learned by assigning arbitrary values to them, as Q-values at such actions are hallucinated by the function approximator itself. On the other hand, if all the actions used to produce Bellman targets also appear in the dataset (i.e., these actions also appear on the left hand side of some Bellman constraint), the resulting function approximator is the most constrained, and likely least simple. This implies that a good baseline that can serve as a gold standard for comparing $\mathcal{L}_{\text{recons}}$ is the reconstruction error attained by offline SARSA (Equation 2). This means that closer the value of $\mathcal{L}_{\text{recons}}(\Phi_{\text{offline RL}})$ to $\mathcal{L}_{\text{recons}}(\Phi_{\text{SARSA}})$, the more desirable the learned representation.

Empirical results. To empirically validate the efficacy of our reconstruction error metric, we compute the values of $\mathcal{L}_{\text{recons}}$ for a variety of D4RL [12] tasks and compare them to the values attained by SARSA. Observe in Figure 4 that while in some cases (e.g. kitchen), the reconstruction error for naïve CQL is much larger than SARSA, indicating excessive loss of information about the dataset, in other cases (antmaze and antmaze-heterogeneous), the reconstruction error for naïve CQL is smaller, indicating that CQL hallucinates information about the dataset action. As an additional point of reference, we also plot this metric for an approach that utilizes an N-step Bellman backup with CQL, and observe that this approach attains a value of $\mathcal{L}_{\text{recons}}$ closer to that of SARSA. Furthermore, even though the policies produced by naïve SARSA don't perform well (as confirmed by prior works [6]), the value of $\mathcal{L}_{\text{recons}}$ to that of SARSA, the better the performance of the resulting method. This empirically corroborates our intuition about the efficacy of this metric.

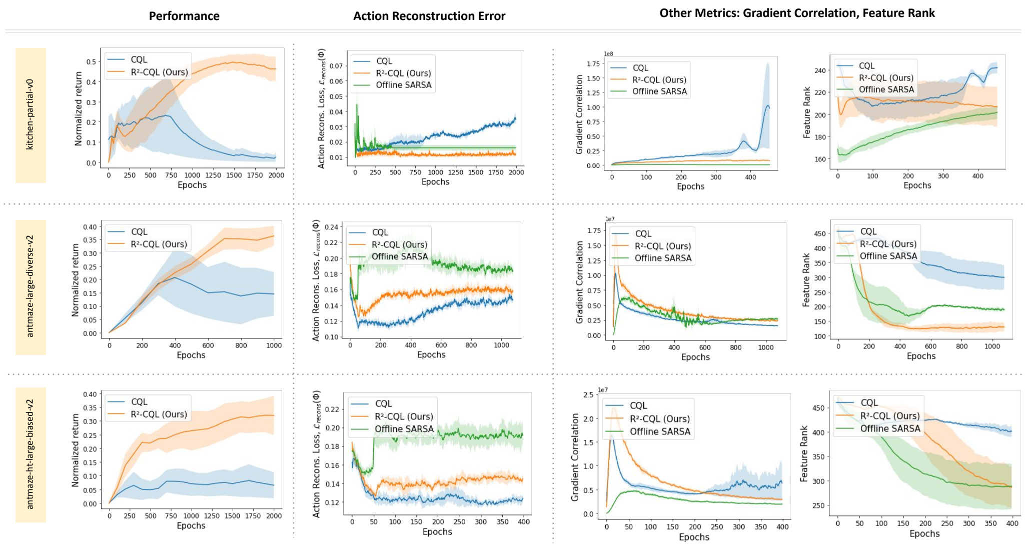

Figure 4: **Performance and metrics of R^2-CQL vs regular CQL, in comparison with SARSA.** Observe that measuring of closeness of the reconstruction error on the dataset actions (Metric 5.1) to the corresponding value for SARSA is able to accurately predict the performance trends, while other prior metrics may not.

Additionally, we also measure the predictive power of existing metrics from prior works such as feature rank penalty [27] and feature dot products [28] in predicting the performance difference between CQL and our approach. While these prior works used extreme values of these metrics (e.g., extremely low rank or extremely large dot products) to diagnose pathologies in TD, our analysis shows that representational issues can still arise when these metrics behave relatively stably (see Figure 4).

Takeaway 5.1. *The closer the value of the reconstruction error metric of an offline RL method based on TD-learning method that utilizes out-of-distribution actions, to that of SARSA, the better we would expect the performance of the learned policy to be.*

6 R^2-CQL: A Simple Approach for Improving Representations For CQL

How can we improve the representations learned by offline RL algorithms? Our analysis above suggests that this would involve constraining the learned representation to be closer to that learned via offline SARSA, which only utilizes dataset actions for which ground truth supervision is available. That is, we wish to devise an approach that can introduce a form of *representational regularization*, which makes the representations closer to that of offline SARSA.

A simple approach that meets these requirements, and imposes a form of representational regularization, is one that utilizes a Bellman backup operator which interpolates between complete bootstrapping and estimating the value for SARSA. To this end, we propose to utilize an ensemble of n-step return estimators in conjunction with offline RL methods, similar to TD(λ) [43]. Concretely, for a given choice of values of $n = \{n_0, n_1, \cdots, n_k\}$, we utilize the following Bellman operator to generate regression targets for TD:

$$\widetilde{\mathcal{B}}^{\pi} Q(\mathbf{s}_0, \mathbf{a}_0) := \frac{1}{k} \sum_{j=1}^{k} \left(\sum_{l=0}^{n_j - 1} \gamma^l r(\mathbf{s}_l, \mathbf{a}_l) + \gamma^{n_j} Q(\mathbf{s}_{n_j}, \mathbf{a}_{n_j}) \right). \tag{5}$$

We will now discuss how we can convert this approach into a practical method for offline RL.

Practical instantiation. Our practical algorithm only modifies the CQL training objective (Equation 1) to now use the Bellman backup operator shown in Equation 5, with no other changes. We inherit the value of α directly from CQL, without tuning it, and do not modify any other hyperparameters. We utilize values of $n = \{1, 3, 5\}$ across all domains. Note that unlike prior methods based on explicit regularization such as the feature rank [28] or dot products [27], our approach does not require any specific hyperparameter to be tuned per domain, highlighting the simplicity of this approach.

Empirical results. We empirically validate our n-step approach by evaluating both the value of $\mathcal{L}_{\text{recons}}$ and the performance across a wide range of offline RL tasks from D4RL [12]. Following the protocol in [28], we present two sets of performance numbers in Table 1: the final performance attained by the algorithm after a fixed number of gradient steps (denoted "Final Performance") and the average performance attained over the course of training (denoted "Average Performance"), which is a measure of the stability of the offline RL algorithm over the course of training. We additionally already presented the value of the reconstruction error on a subset of domains in Figure 4.

Observe that on all the tasks, our approach, R^2-CQL attains a better or comparable performance both measured by the final performance of the algorithm and the average performance across iterations, which demonstrates the stability of training. The gap between naïve CQL and the n-step approach is larger under the average performance metric, indicating that the latter is much more stable. Finally, perhaps unsurprisingly, the representational metrics do indicate that utilizing the mixture of n-step Bellman targets does lead to reconstruction error values closer to that of offline SARSA.

While this simple approach does lead to improvements in performance, perhaps the more important question is *why* does it actually improve performance. Traditionally, in the on-policy setting, the utility of an ensemble of N-step returns via approaches such as TD(λ) [43] or GAE [41] primarily emerges from an ability to better manage a bias-variance tradeoff: by controlling an algorithmic hyperparameter, the bias induced in learning a parametric Q-function can be effectively traded against the variance of a Monte-Carlo return estimator. However, in this case, we utilize N-step returns in an offline setting, with an already pessimistic algorithm (CQL). Since CQL already aims to underestimate the return of the learned policy, we would expect N-step Bellman targets to only be *more* conservative, since they bias the Q-function towards the values of the behavior policy and therefore be more biased than CQL. Typically, this bias issue is solved by utilizing importance corrections [9, 35], but we do not use any such correction. Therefore, not only does R^2-CQL use a high variance Bellman target, but also a more biased one, and yet it outperforms CQL. This again indicates that the representation learning benefits of this approach are likely much more useful towards improving performance despite the bias.

Task	Final Performance		Average Performance	
	CQL	R^2-CQL	CQL	R^2-CQL
kitchen-mixed	0.000 ± 0.000	0.362 ± 0.013	0.085 ± 0.114	0.330 ± 0.098
kitchen-partial	0.138 ± 0.138	0.475 ± 0.075	0.089 ± 0.111	0.414 ± 0.139
kitchen-complete	0.000 ± 0.000	0.025 ± 0.025	0.163 ± 0.143	0.100 ± 0.106
antmaze-medium-play	0.435 ± 0.315	0.670 ± 0.090	0.569 ± 0.200	0.602 ± 0.216
antmaze-medium-diverse	0.680 ± 0.070	0.645 ± 0.045	0.511 ± 0.214	0.538 ± 0.212
antmaze-large-play	0.005 ± 0.005	0.320 ± 0.000	0.098 ± 0.105	0.265 ± 0.104
antmaze-large-diverse	0.095 ± 0.035	0.420 ± 0.010	0.162 ± 0.083	0.303 ± 0.145
antmaze-ht-large	0.090 ± 0.090	0.380 ± 0.160	0.082 ± 0.057	0.283 ± 0.125
antmaze-ht-large-biased	0.000 ± 0.000	0.310 ± 0.190	0.067 ± 0.057	0.302 ± 0.098
antmaze-ht-medium	0.000 ± 0.000	0.320 ± 0.140	0.155 ± 0.118	0.290 ± 0.121
antmaze-ht-medium-biased	0.000 ± 0.000	0.220 ± 0.040	0.126 ± 0.192	0.234 ± 0.083

Table 1: Final and average performance for R^2-CQL and CQL across 7 D4RL tasks and 4 heterogeneous antmaze tasks. All performances are evaluated with 2 random seeds for 1000 epochs. We see that R^2-CQL improves the final and average perfromance over naïve CQL significantly.

7 Discussion and Conclusion

In this paper, we demonstrate that while addressing the overestimation due to OOD actions is important for offline RL, a crucial ,but largely overlooked ,factor for obtaining good performance in value-based offline RL algorithms is good representation quality. We show through extensive empirical results that, perhaps surprisingly, pessimism in practical offline RL algorithms such as CQL contributes to the performance not only as a way to prevent overestimation, but more significantly as a way to induce good representations. We also show that pessimism is not the only way to attain good representations and methods that attain good representations can still work well. Based on this experimental analysis, we propose a practical metric that quantitatively tracks the quality of learned representation, and show that simply utilizing a ensemble of N-step returns to compute Bellman targets can provide a strong representation regularization and thus significantly improve the performance of conservative offline RL algorithm. We hope that our discovery can highlight the importance of representation learning in offline RL, and thus open up new opportunities to devise stronger offline RL methods.

While we provide a practical method R^2-CQL to regularize representations, by no means we claim that it is an optimal method. Therefore a natural step for future work direction is to seek for better ways to understand and improve the quality of learned representations. We believe that such future search has the potential of bringing deep insights and profound influences to the field of offline RL and hope that our analysis sheds light on some of these questions.

References

[1] Joshua Achiam, Ethan Knight, and Pieter Abbeel. Towards characterizing divergence in deep q-learning. *arXiv preprint arXiv:1903.08894*, 2019.

[2] Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. *arXiv preprint arXiv:2008.05556*, 2020.

[3] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018.

[4] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pages 7413–7424, 2019.

[5] Emmanuel Bengio, Joelle Pineau, and Doina Precup. Interference and generalization in temporal difference learning. *arXiv preprint arXiv:2003.06350*, 2020.

[6] David Brandfonbrener, William F Whitney, Rajesh Ranganath, and Joan Bruna. Offline rl without off-policy evaluation. *arXiv preprint arXiv:2106.08909*, 2021.

[7] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[8] Will Dabney, André Barreto, Mark Rowland, Robert Dadashi, John Quan, Marc G Bellemare, and David Silver. The value-improvement path: Towards better representations for reinforcement learning. *arXiv preprint arXiv:2006.02243*, 2020.

[9] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.

[10] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. *arXiv preprint arXiv:2007.06700*, 2020.

[11] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. In *arXiv*, 2020. URL https://arxiv.org/pdf/2004.07219.

[12] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[13] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *arXiv preprint arXiv:2106.06860*, 2021.

[14] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900*, 2018.

[15] Dibya Ghosh and Marc G Bellemare. Representations for stable off-policy reinforcement learning. *arXiv preprint arXiv:2007.05520*, 2020.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[17] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[18] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

[19] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? *arXiv preprint arXiv:2012.15085*, 2020.

[20] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

[21] Hajime Kimura, Shigenobu Kobayashi, et al. An analysis of actor-critic algorithms using eligibility traces: reinforcement learning with imperfect value functions. *Journal of Japanese Society for Artificial Intelligence*, 15(2):267–275, 2000.

[22] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[23] Ilya Kostrikov, Jonathan Tompson, Rob Fergus, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. *arXiv preprint arXiv:2103.08050*, 2021.

[24] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pages 11761–11771, 2019.

[25] Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. *arXiv preprint arXiv:2003.07305*, 2020.

[26] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

[27] Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O9bnihsFfXU.

[28] Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. DR3: Value-Based Deep Reinforcement Learning Requires Explicit Regularization. *arXiv preprint arXiv:2112.04716*, 2021.

[29] Byung-Jun Lee, Jongmin Lee, and Kee-Eung Kim. Representation balancing offline model-based reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=QpNz8r_Ri2Y.

[30] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[31] Clare Lyle, Mark Rowland, Georg Ostrovski, and Will Dabney. On the effect of auxiliary tasks on representation dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 1–9. PMLR, 2021.

[32] Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. *arXiv preprint arXiv:2204.09560*, 2022.

[33] Hamid R. Maei, Csaba Szepesvári, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, 2009.

[34] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[35] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1054–1062, 2016.

[36] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

[37] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[38] Rafael Rafailov, Tianhe Yu, A. Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. *Learning for Decision Making and Control (L4DC)*, 2021.

[39] Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration. *arXiv preprint arXiv:2106.06431*, 2021.

[40] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

[41] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[42] Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

[43] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Second edition, 2018.

[44] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000, 2009.

[45] Phillip Swazinna, Steffen Udluft, and Thomas Runkler. Overcoming model bias for robust offline deep reinforcement learning. *arXiv preprint arXiv:2008.05533*, 2020.

[46] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[47] Kristian Hartikainen George Tucker Sehoon Ha Jie Tan Vikash Kumar Henry Zhu Abhishek Gupta Pieter Abbeel Tuomas Haarnoja, Aurick Zhou and Sergey Levine. Soft actor-critic algorithms and applications. Technical report, 2018.

[48] Hado Van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*, 2018.

[49] Hado van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? *arXiv preprint arXiv:1906.05243*, 2019.

[50] Ziyu Wang, Alexander Novikov, Konrad Żołna, Jost Tobias Springenberg, Scott Reed, Bobak Shahriari, Noah Siegel, Josh Merel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *arXiv preprint arXiv:2006.15134*, 2020.

[51] Paweł Wawrzyński. Real-time reinforcement learning by sequential actor–critics and experience replay. *Neural networks*, 22(10):1484–1497, 2009.

[52] Colin Wei, Jason Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. 2019.

[53] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[54] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[55] Chenjun Xiao, Bo Dai, Jincheng Mei, Oscar A Ramirez, Ramki Gummadi, Chris Harris, and Dale Schuurmans. Understanding and leveraging overparameterization in recursive value estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=shbAgEsk3qM.

[56] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34, 2021.

[57] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

[58] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *arXiv preprint arXiv:2102.08363*, 2021.