Adaptive Reward Shifting Based on Behavior Proximity for Offline Reinforcement Learning

Zhe Zhang[1,2] , **Xiaoyang Tan**[1,2*]

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
{zhangzhe, x.tan}@nuaa.edu.cn

Abstract

One of the major challenges of the current offline reinforcement learning research is to deal with the distribution shift problem due to the change in state-action visitations for the new policy. To address this issue, we present a novel reward shifting-based method. Specifically, to regularize the behavior of the new policy at each state, we modify the reward to be received by the new policy by shifting it adaptively according to its proximity to the behavior policy, and apply the reward shifting along opposite directions for in-distribution actions and the ones not. In this way we are able to guide the learning procedure of the new policy itself by influencing the consequence of different actions explicitly, helping it to achieve a better balance between behavior constraints and policy improvement. Empirical results on the popular D4RL benchmarks show that the proposed method obtains competitive performance compared to the state-of-art baselines.

1 Introduction

The success of modern deep reinforcement learning (RL) mainly relies on a large number of online interactions with the environment. This characteristic limits its broader applications to real-world scenes, where online data may be costly and dangerously collected, *e.g.*, in such fields as healthcare [Yu *et al.*, 2021a], autonomous driving [Grigorescu *et al.*, 2020], and so on. Offline RL instead attempts to address this issue by learning from a fixed dataset collected by the behavior policy in advance. This involves learning from behaviors generated by a policy different than the new policy. Unfortunately, the direct employment of the common off-policy strategy often fails to achieve the same level of performance as in the online setting [Fujimoto *et al.*, 2019; Levine *et al.*, 2020].

The *extrapolation error* from out-of-distribution (OOD) actions is generally thought of as the main reason responsible for the aforementioned performance degradation [Fujimoto *et al.*, 2019]. More specifically, the *distribution shift* between the learned policy and behavior policy may easily lead to the overestimated Q-value of OOD actions and such error compounds, leading to potentially dangerous consequences. Hence it's important to prevent the agent from taking overestimated OOD actions. One way for this is to enforce the learned policy to stay close to the behavior policy, *e.g.*, by imposing regularizations on the actor or critic learning. Specifically, actor regularization usually [Wu *et al.*, 2019; Kumar *et al.*, 2019] restricts the actor from being updated within a small range of the behavior policy via policy constraints. Nevertheless, these constraints lack direct regularization on critic updates to prevent the propagation of extrapolation errors. To address this issue, various critic regularization schemes have been proposed in previous literature to regularize value function, either through penalizing Q-values for OOD actions [Kumar *et al.*, 2020; Yu *et al.*, 2021b] or through only selecting in-sample actions to construct the bootstrapped targets [Fujimoto *et al.*, 2019].

An alternative but less studied strategy to address the above issue is through reward shifting, a specific mechanism for reward shaping. For example, [Sun *et al.*, 2022] show that adding a constant positive shift to the reward of the in-sample data is beneficial to avoid taking OOD actions in offline RL. Intuitively, in this setting, the positive shift has the effect of initializing the policy pessimistically [Sun *et al.*, 2022], hence encouraging the agent to exploit in-distribution data more frequently than the OOD data. One main advantage of reward shifting lies in its capability to influence the consequence of different actions explicitly, which, compared with the aforementioned regularization-based methods, essentially allows us to have some direct control over the learning procedure itself. This highlights the usefulness of reward shifting for offline RL, but many problems remain unsolved and more exploration is needed, *e.g.*, in seeking practical schemes to set the reward shifting for more efficient and stable offline RL.

Inspired by this, we propose a novel reward shifting method for offline RL, which utilizes two opposite reward shifting schemes to reduce OOD behaviors and alleviate the consequent extrapolation error. Specifically, we introduce a positive reward shifting for in-distribution data, seeking more exploitation for the observed data. Meanwhile, a negative shift will be added to the reward function of OOD data for less exploration of itself. We dub this proposed method Bi-direction Reward Shifting (BRS) to emphasize the fact that two opposite reward shifting schemes are adopted. To achieve

[*] Corresponding author

a better balance between the OOD behavior constraint and the policy improvement, we further present a Proximity-based BRS (PBRS) method, which adjusts the magnitude of reward shifting adaptively for each state according to the proximity of the learned policy to behavior policy. Finally, we evaluate the performance on the popular D4RL benchmark and show that our method is competitive compared with the SOTA baselines.

2 Related Work

Current researches on offline RL usually tackle the extrapolation error issue by regularizing the learning of the actor or critic. Specifically, policy constraint methods introduce different regularization terms into the actor objective in order to keep close to the behavior policy. Previous work either adds the policy constraint term into the policy optimization objective [Wu *et al.*, 2019; Fujimoto and Gu, 2021] or directly updates the policy using the closed-form solution to the joint objective comprised of the policy improvement and the policy constraint [Peng *et al.*, 2019; Siegel *et al.*, 2020]. As for the critic regularization methods, these works usually learn a conservative value function to mitigate the overestimation issue of OOD actions. To achieve this, conservative Q-learning (CQL) [Kumar *et al.*, 2020] directly minimizes the Q-values of OOD samples besides the TD-error objective. Some other studies [Kumar *et al.*, 2019; Fujimoto *et al.*, 2019] choose to construct the bootstrapped target using the maximum within in-sample actions while not the one over the whole action space. In practice, these additional regularization terms usually need to be designed and finetuned carefully.

Instead of the actor and critic regularization methods, research on reward engineering is also considered to address the issues in offline RL. Some related studies utilize different prior information to learn the additional supplementary reward. For example, Mezghani et al. focus on the goal-conditioned offline RL and propose to learn a distance function used to shape a dense reward function via self-supervised learning [Mezghani *et al.*, 2022], while several works [Konyushkova *et al.*, 2020; Cabi *et al.*, 2020] attempt to learn reward functions from offline data with human preference or few human annotations. Different from the above studies that focus on incomplete reward information in offline RL, some other methods aim to address the common extrapolation error issue by modifying the original reward function. Specifically, MOPO [Yu *et al.*, 2020] reshapes the reward with the uncertainty of ensemble dynamic models, and TD3-CVAE [Rezaeifar *et al.*, 2022] subtracts an anti-exploration bonus defined by the reconstruction error from the reward function. Both methods utilize the reward penalty on possible OOD data to reshape reward functions, which are expected to regularize the OOD behavior. Our work is related to TD3-CVAE and MOPO in the sense of reward shaping, but in contrast, our method can not only add opposite bonuses for different data but also adjust the bonus along with the policy learning process, while MOPO and TD3-CVAE add the bonus absolutely dependent on the property of pretrained model and may be affected by some abnormal actions values.

3 Priliminaries

In this paper, we choose to solve RL problem within the Markov Decision Process (MDP) framework specified by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, where $\mathcal{S}, \mathcal{A}$ denote the state and action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ and $r : \mathcal{S} \times \mathcal{A} \rightarrow [R_{\min}, R_{\max}]$ represent the Markov transition probability function and reward function respectively, and $\gamma \in (0, 1)$ is the discounted factor. The goal of RL is to learn a policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ that can maximize the corresponding expected discounted cumulative return: $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] = \mathbb{E}_{s\sim d^\pi, a\sim\pi}\left[r(s, a)\right]$, where d^π is the stationary state distribution. To evaluate the quality of policy, the Bellman operator is usually used to estimate its Q-value via bootstrapping:

$$\mathcal{T}^\pi Q(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim P, a'\sim\pi}\left[Q(s', a')\right] \qquad (1)$$

For the online RL problem, one can conduct the policy evaluation after the online data is collected by the current policy, where the estimation error can be corrected through this online trial and error.

3.1 Offline Reinforcement Learning

Different from online RL setting, offline RL [Levine *et al.*, 2020] is required to evaluate and learn the policy based on a fixed dataset $\mathcal{D} = \{s_i, a_i, r_i, s_{i+1}\}_{i=1}^{N}$ generated in advance by the unknown behavior policy π_β. Traditional off-policy methods [Lillicrap *et al.*, 2015; Mnih *et al.*, 2015] usually fail in this setting due to the *distribution shift* between the learned policy and behavior policy π_β. This is because the OOD action a' rarely visited by policy π_β may be easily overestimated and chosen to construct the bootstrapped target using Eq.(1), and thus accumulate and propagate extrapolation errors in estimated value function.

3.2 Reward Shifting for Reinforcement Learning

Reward shifting is a special case of reward shaping method [Laud, 2004; Ng *et al.*, 1999] that uses a linear transformation. In particular, given a MDP $\mathcal{M}$, the original reward function r is replaced with its linear form: $r' = k \cdot r + b, \forall k > 0, b \in \mathbb{R}$. This linear transformation will not change the optimal policy and the optimal Q-value before and after this change also satisfies a linear transformation:

$$\begin{aligned} &Q^*_{k,b}(s, a) = k \cdot Q^*(s, a) + \frac{b}{1-\gamma} \\ \Rightarrow &\pi^*(s) = \arg\max_{a\in\mathcal{A}} Q^*_{k,b}(s, a) = \arg\max_{a\in\mathcal{A}} Q^*(s, a) \end{aligned} \qquad (2)$$

Where $Q^*_{k,b}$ and Q^* represent the optimal Q-values using transformed reward r' and original reward r respectively. Previous work [Sun *et al.*, 2022] provides a key insight into the different effects of reward shifting. For convenience, they fixed the scaling factor $k = 1$ and studied how the opposite bias b affects the reinforcement learning in different settings:

a) Online RL: The online RL prefers a negative reward shifting, *i.e.*, $b < 0$, which may lead to the relatively optimistic initialization. This is because the values of visited state-action pairs may be negatively shifted lower than the initialization. And in subsequent interactions, the greedy policy will be prone to select under-explored actions.

b) Offline RL: In contrast, according to Eq.(2), a positive bias $b > 0$ will lead to universally optimistic optimal values (positively shifted by $\frac{b}{1-\gamma}$). Since only the observed offline data is updated with the optimistic target, this will enlarge the value gap with the unobserved data (*i.e.*, OOD data) that has relatively pessimistic initialization. The larger gap can lead to more exploitation on observed data when choosing the action greedily, which is beneficial to avoid taking OOD actions in offline RL.

In this paper, we also keep the fixed $k = 1$ and utilize the different reward shifting with the positive or negative bias b.

4 Method

In this section, we first introduce a simple Bi-direction Reward Shifting (BRS) method for offline RL, which utilizes the opposite reward shifting for training data in and out of distribution, achieving the anti-exploration on OOD data. Then, to balance the OOD behavior constraint and policy improvement, we further present an adaptive version of BRS, which adjusts the reward shifting for each state according to the proximity of the learned policy to the behavior policy. Besides, we show a close connection of our BRS to the representative conservative-based method. Finally, we detail the practical algorithm with some specific designs.

4.1 Bi-direction Reward Shifting for Offline RL

As mentioned above, different RL settings prefer different reward shifting, *i.e.*, a positive reward bias $b^+ > 0$ favors the more exploitation of the given dataset in offline RL, while a negative reward bias $b^- < 0$ is beneficial for optimistic exploration in online RL.

In this paper, we focus on the offline RL setting, which suffers from the extrapolation error and has to avoid OOD behavior. Inspired by the effects of reward shifting, we propose to apply the positive reward shifting to the provided offline data. Mathematically, we define the iterative operator of in-distribution transitions (s, a, s', r) sampled from $\mathcal{D}$ by:

$$\mathcal{T}_{\text{in}} Q_k(s, a) = r(s, a) + b^+ + \gamma \mathbb{E}_{a \sim \pi} [Q_k(s', a')] \quad (3)$$

The positive bias $b^+ > 0$ added to the reward function of sampled data from buffer $\mathcal{D}$ is expected to lead to more exploitation of in-distribution data.

As for the unobserved data, *i.e.*, $(s, a) \notin \mathcal{D}$, we add a negative shift $b^- < 0$ into its reward function, which can result in less exploration and queries on out-of-distribution data and mitigate its potential overestimation issue. Similarly, we define the iterative operator of OOD data as:

$$\mathcal{T}_{\text{ood}} Q_k(s, a) = Q_k(s, a) + b^- \quad (4)$$

Note that since the offline dataset doesn't contain the out-of-distribution data, we have no information about its reward and transitioned state. So we use the current approximated Q-value $Q_k(s, a)$ as the pseudo value target, which is dependent on the generalization of the learned value function.

Both the more exploitation of in-distribution data and the less exploration of OOD data will make the learned policy avoid the OOD behavior as much as possible (called *anti-exploration* on OOD actions[Rezaeifar *et al.*, 2022]). Specifically, TD3-CVAE[Rezaeifar *et al.*, 2022] achieves this anti-exploration by subtracting a bonus from the reward function, while our method adds opposite reward biases for different data. Besides, the subtracted bonus for each action in TD3-CVAE depends on the reconstruction error itself. In contrast, our method determines the bias by the policy difference for each state (introduced in the next section), which may mitigate the effects of individual abnormal action values. Due to the opposite reward shifting imposed on different data, we dub this method **B**i-direction **R**eward **S**hifting (BRS).

4.2 Proximity-based BRS

The proposed BRS method utilizes the opposite reward shifting to achieve the anti-exploration of OOD actions. Note that the magnitude of reward shifting controls the strength of this anti-exploration. This means that, according to the above definitions in Eq.(3) and Eq.(4), the BRS method will impose the same strength of anti-exploration across all states via two constant reward biases b^+ and b^-. However, it may not be the best choice - *e.g.*, for some states where π and π_β behave very differently, a higher magnitude of reward shifting is preferred as this will help to increase the strength of anti-exploration by restricting the learned policy π from performing OOD actions. On the other hand, for those states where π has already been trained to behave similarly with π_β, the same amount of high magnitude would instead make both policies become over-consistent and hinder the performance improvement of target policy π. In such cases, a lower magnitude of reward shifting is obviously desired.

Based on the above observations, we propose a **P**roximity-based **BRS** (PBRS) method that adjusts the shifted reward adaptively for each state according to the proximity between the learned policy and behavior policy as the following:

$$b_k(s) = c \cdot g\left(D_f\left[\pi_k(\cdot|s) \| \pi_\beta(\cdot|s)\right]\right) \quad (5)$$

where $c > 0$ is a scaling coefficient for this adaptive reward shifting, and the f-divergence between π_k and π_β, $D_f\left[\pi_k(\cdot|s) \| \pi_\beta(\cdot|s)\right]$, is used to measure the proximity of the learned policy to behavior policy at state s. $g(\cdot)$ is a monotonically increasing function which satisfies that $g(0) \geq 0$.

Note that the proposed function $b_k(s)$ finally returns an adaptive non-negative shifted reward for each state at any k-th iteration, and we apply this state-wise reward shifting for the training data in and out of distribution along opposite directions in PBRS:

$$\begin{aligned} & Q_{k+1}(s, a) \\ = & \begin{cases} r(s, a) + \gamma \mathbb{E}_{a \sim \pi} [Q_k(s', a')] + b_k(s) & (s, a) \sim \mathcal{D} \\ Q_k(s, a) - b_k(s) & (s, a) \notin \mathcal{D} \end{cases} \end{aligned} \quad (6)$$

The motivation behind Eq.(6) is to utilize a larger reward shifting to regularize OOD actions at those states where a significant difference exists between both π_k and π_β. In contrast, if π_k approaches to π_β at some states, the decreasing $b_k(s)$ is beneficial to obtain more policy improvements within a broader range by relaxing the behavior constraint. We finally expect to achieve a balance between the behavior constraint and the policy improvement for each state by an adaptive strength of anti-exploration for OOD actions.

4.3 A Link to CQL

The conservative-based methods in offline RL follow the pessimistic principle and aim to learn a conservative Q-value function to mitigate the overestimation issue for OOD actions so as to reduce the extrapolation error. Taking the representative CQL algorithm [Kumar *et al.*, 2020] as an example, although CQL derives its original critic objective from the perspective of the Q-value regularization, we show that CQL is also equivalent to a special case of our BRS method. Recalling the definition of critic objective in CQL:

$$\begin{aligned}\min_Q J_{\text{cql}}(Q) = &\frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a)-\mathcal{T}^{\pi_k}Q_k(s,a)\right)^2\right]\\ &+\alpha\left(\mathbb{E}_{\substack{s\sim\mathcal{D},\\ a\sim\mu(a|s)}}[Q(s,a)]-\mathbb{E}_{\substack{s\sim\mathcal{D},\\ a\sim\pi_\beta(a|s)}}[Q(s,a)]\right)\end{aligned} \tag{7}$$

Where the actions sampled by the special policy $\mu(a|s)$[1] can be viewed as OOD data. Note that, besides minimizing the Bellman error, the minimization of the Q-value regularization term helps to push down the Q-values for actions from μ, and pull up the Q-values for in-distribution data simultaneously. Considering the derivative of Eq.(7), we can rewrite its objective:

$$\begin{aligned}&\nabla J_{\text{cql}}(Q)\\ =&\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a)-\mathcal{T}^{\pi_k}Q_k(s,a)\right)\cdot\nabla Q(s,a)\right]\\ &+\alpha\left(\mathbb{E}_{\substack{s\sim\mathcal{D},\\ a\sim\mu(a|s)}}[\nabla Q(s,a)]-\mathbb{E}_{\substack{s\sim\mathcal{D},\\ a\sim\pi_\beta(a|s)}}[\nabla Q(s,a)]\right)\\ =&\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a)-\mathcal{T}^{\pi_k}Q_k(s,a)-\alpha\right)\cdot\nabla Q(s,a)\right]\\ &+\mathbb{E}_{\substack{s\sim\mathcal{D},\\ a\sim\mu(a|s)}}\left[\left(Q_k(s,a)-Q_k(s,a)+\alpha\right)\cdot\nabla Q(s,a)\right]\\ =&\frac{1}{2}\nabla\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a)-(\mathcal{T}^{\pi_k}Q_k(s,a)+\alpha)\right)^2\right]\\ &+\frac{1}{2}\nabla\mathbb{E}_{\substack{s\sim\mathcal{D},\\ a\sim\mu(a|s)}}\left[\left(Q(s,a)-(Q_k(s,a)-\alpha)\right)^2\right]\Bigg|_{Q=Q_k}\end{aligned} \tag{8}$$

Where we use a fact that all in-distribution data in $\mathcal{D}$ is sampled from the unknown behavior policy π_β. Note that if we treat the actions sampled by μ as OOD actions, the above Eq.(8) indicates that CQL is a special BRS method under the conditions that the opposite shifted rewards are $b^+ = \alpha$ and $b^- = -\alpha$,

And according to the original analysis (Theorem 3.4 in [Kumar *et al.*, 2020]), compared with the Bellman operator, CQL can expand the difference in expected Q-values under behavior policy π_β and μ which is beneficial to prevent over-optimistically erroneous OOD actions when choosing a large enough α. However, a large α may also become problematic in some cases, especially when the given offline dataset doesn't contain the optimal actions a^* for some states. We provide the analysis of this case in the following Theorem:

Theorem 1. *When using CQL algorithm in discrete action space, assume that the corresponding optimal action a^* belongs to OOD actions for any state s,* i.e.*,* $\pi_\beta(a^*|s) = 0$*, and $\forall a \in \mathcal{A}^s_{\text{in}} = \{a : \pi_\beta(a|s) > 0|s\}$ satisfies that $\max_{a\in\mathcal{A}^s_{\text{in}}} |Q_k(s,a) - Q_{k+1}(s,a)| < \alpha$, then the probability of a^* induced by Boltzmann policy will decrease,* i.e.*,* $\pi_{k+1}(a^*|s) < \pi_k(a^*|s)$.

We refer to the Appendix.A for its detailed proof. Theorem 1 shows that a large α will lead to the probability decreasing of a^* earlier, which may hinder the policy improvement and learn a suboptimal policy finally. This suggests the necessity of adaptive reward shifting for different states, which is important to balance the OOD actions constraints and the policy improvement, especially for the non-expert datasets.

4.4 Practical Algorithm

In the previous sections, we propose the general framework of the PBRS method, while the practical algorithm still involves some special designs. In this paper, we build our algorithm upon the popular off-policy RL algorithm, Soft Actor-Critic (SAC) [Haarnoja *et al.*, 2018], and provide more details about the specific choices here.

Loss functions. According to the above definition, we should construct the Q-value targets for in-distribution and out-of-distribution data separately. We directly sample in-distribution tuple (s, a, s', r) from the offline dataset $\mathcal{D}$. As for OOD data, we sample OOD states from in-distribution dataset $\mathcal{D}$ and sample OOD actions for these OOD states by following the current learned policy π. It's reasonable to sample OOD data in this way since in continuous action space, the sampled OOD actions by learned policy π are almost impossible to be included in offline dataset $\mathcal{D}$ except π is the same with π_β. Combining both the in-distribution value target and OOD target in Eq.(6), we get the final loss function of the critic objective:

$$\begin{aligned}\mathcal{L}(Q) = &\mathbb{E}_{s\sim\mathcal{D},a\sim\pi}\left[\left(Q(s,a)-(Q_k(s,a)-b_k(s))\right)^2\right]\\ &+\mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[\left(Q-(r(s,a)+b_k(s)+\gamma Q_k(s',a'))\right)^2\right]\end{aligned} \tag{9}$$

After updating the critic objective, we learn the actor by maximizing the SAC-style policy optimization objective:

$$\mathcal{L}(\pi) = \mathbb{E}_{s\sim\mathcal{D},a\sim\pi}\left[\min_{i=1,2} Q^i(s,a) - \tau\log\pi(\cdot|s)\right] \tag{10}$$

Where τ is the coefficient of the entropy term and the minimum of double Q functions $\min_{i=1,2} Q^i(s,a)$ is used to address the overestimation issue following the SAC algorithm. Both the critic and actor are parameterized by the neural network and updated by the above objectives.

Behavior policy. The calculation of adaptive reward shifting term $b_k(s)$ requires the information of behavior policy π_β. Thus, we need to fit a behavior policy model using the offline dataset $\mathcal{D} = \{s_i, a_i\}_{i=0}^N$. Though previous studies

[1] μ is assumed to match the state-marginal of behavior policy, *i.e.*, $\mu(s,a) = d^{\pi_\beta}(s)\mu(a|s)$, avoiding unseen states. And μ is learned by an adversarial training objective to maximize the critic objective.

Algorithm 1 Proximity-based BRS

```
Input: Dataset $\mathcal{D} = \{s_i, a_i, r_i, s_{i+1}\}_{i=0}^{K}$.
initialize behavior policy $\pi_\beta$, learned policy network $\pi_\theta$,
    Q-function network $Q_\psi$.
for t=0, 1, $\cdots$, M do
    Sample mini-batch samples $(s, a) \sim \mathcal{D}$.
    Update $\pi_\beta$ using Eq.(11).
end for
for k=0, 1, $\cdots$, N do
    Sample mini-batch samples $(s, a, r, s') \sim \mathcal{D}$.
    Calculate adaptive reward shifting $b_k(s)$ using Eq.(5)
    Update $Q_\psi$ by Eq.(9).
    Update $\pi_\theta$ by Eq.(10).
end for
```

often adopt the conditional variational autoencoder (CVAE) [Sohn *et al.*, 2015; Rezaeifar *et al.*, 2022] or the Mixture of Gaussians [Kostrikov *et al.*, 2021] as the density estimator to model the behavior policy π_β, in this paper, we found that a simple single Gaussian model is sufficient to achieve good performance. So we choose to train a Gaussian policy model using the maximum likelihood estimation (MLE) objective:

$$\pi_\beta = \arg\max_{\pi_\theta} \mathcal{L}_{\mathrm{MLE}}(\pi_\theta) = \mathbb{E}_{s,a\sim\mathcal{D}}\left[\log \pi_\theta(s, a)\right] \qquad (11)$$

To be consistent with the learned policy, we can further add a policy entropy term $-\tau \log \pi_\theta(a|s)$ to the above Eq.(11).

Adaptive reward shifting. When computing the adaptive reward shifting term $b_k(s)$ according to Eq.(5), we tried both Maximum Mean Discrepancy and KL-divergence as the proximity measurement of both policies and found that the latter can achieve a better result on broader tasks. We estimate this KL-divergence using the Monte Carlo sampling method: $\hat{D}_{\mathrm{KL}}[\pi\|\pi_\beta](s) \approx \frac{1}{N}\sum_{i=1}^{N} \log \pi(a_i|s)/\log \pi_\beta(a_i|s)$ and replace $\hat{D}_{\mathrm{KL}}(s)$ by $[\hat{D}_{\mathrm{KL}}(s)]_0^u$, where the clipping function $[\cdot]_l^u$ is used to avoid the numerical issue and ensure the non-negativity of KL-divergence. We fix $u = 10$ for practical runs. As for the monotonically increasing function $g(\cdot)$, we found both the softplus function $g(x) = \log(1 + e^x)$ and the simple identity function $g(\cdot) = \mathbb{I}(\cdot)$ could lead to similar performance improvements. We show the results of using softplus function in the main text and supplement the results of using the identity function in the Appendix. Finally, we summarize the overall algorithm in Algorithm 1

5 Experiments

In this section, to verify the feasibility and effectiveness of our proposed method, we first empirically evaluate the performance on the D4RL benchmark [Fu *et al.*, 2020] and compare it with several strong baselines. Furthermore, we empirically analyze some critical properties to confirm the rationality of our motivation.

5.1 Experiment Setup

Baselines. In the D4RL benchmark, we compare our PBRS method with some current state-of-the-art methods. These representative algorithms attempt to solve the existing issues in offline RL from different perspectives:

- **UWAC** [Wu *et al.*, 2021] that reweights the TD-error according to the uncertainty estimation;
- **TD3-CVAE** [Rezaeifar *et al.*, 2022] that penalizes the rewards of OOD actions via a reconstruction error;
- **MOPO** [Yu *et al.*, 2020] that imposes reward penalty via the uncertainty of dynamic models;
- **CQL** [Kumar *et al.*, 2020] that learns a conservative Q function by minimizing the Q-values of OOD actions;
- **TD3-BC** [Fujimoto and Gu, 2021] that regularizes the policy update via a simple BC constraint;
- **IQL** [Kostrikov *et al.*, 2022] that learns the V-value function so as to avoid the queries on OOD actions.

Datasets and implementation. We choose to conduct experiments on 12 MuJoCo locomotion tasks made up of all combinations of three environments (HalfCheetah, Hopper, Walker2d) across four different types of collected datasets (medium, medium-replay, medium-expert, expert). Since we adopt the latest bug-fixed '-v2' dataset for empirical comparison, we reimplement the CQL that evaluates the performance on '-v0' datasets in the original paper. As for the other baselines, we take their reported results from their paper directly. Note that the coefficient c used in our adaptive reward shifting term $b_k(s)$ controls the strength of anti-exploration. Due to the significant differences between different types of datasets, we tune this coefficient for each task over a small set $c \in \{0.3, 0.5, 0.7, 0.9\}$, and the corresponding ablation study will be provided later. Besides, we also evaluate our method on the more challenging AntMaze domains and provide the complete results as well as more experimental details in the Appendix.B.

5.2 Performance Comparison

We summarize and compare the average normalized score of our PBRS with all mentioned baselines in Table 1, where 0 represents a random policy and 100 corresponds to an expert policy. All the scores are averaged over the final 10 evaluations after one million training steps. We provide the complete learning curves in Appendix.B. We can see that our method can significantly outperform these strong baselines across most tasks. Especially in 'medium' and 'medium-replay' datasets that include many suboptimal samples, our method can suppress the most baselines with a large margin. Even for the high-quality datasets ('medium-expert' and 'expert'), our PBRS can still achieve competitive performance except on the 'HalfCheetah-medium-expert' task. Compared with TD3-CVAE and MOPO that add the negative bonus to reward functions, our method achieves superior performance on many tasks except a few ones ('HalfCheetah-medium' and 'Halfcheetah-medium-replay'), which verifies the effectiveness of the opposite reward shifting.

Note that our proposed method can achieve remarkable performance improvements over the CQL algorithm on 11 out of 12 datasets. Since the CQL method can be seen as a special case of our method using constant reward shifting, we attribute these inferior results to the rigid strength of

Task Name	UWAC	TD3-CVAE	MOPO	CQL	TD3+BC	IQL	PBRS
HalfCheetah-m	42.2±0.4	43.2±0.4	**73.1±2.4**	49.1±0.1	48.3±0.3	47.4±0.2	58.2±0.5
Hopper-m	50.9±4.4	55.9±11.4	38.3±34.9	67.5±2.4	59.3±4.2	66.3±5.7	**75.4±1.8**
Walker2d-m	75.4±3.0	68.2±18.7	41.2±30.8	83.1±0.6	83.7±2.1	78.3±8.7	**88.5±0.8**
HalfCheetah-m-r	35.9±3.7	45.3±0.4	**69.2±1.1**	45.5±0.2	44.6±0.5	44.2±1.2	49.4±0.2
Hopper-m-r	25.3±1.7	46.7±17.9	32.7±9.4	95.5±0.8	60.9±18.8	94.7±8.6	**102.3±0.7**
Walker2d-m-r	23.6±6.9	15.4±7.8	73.7±9.4	82.5±2.1	81.8±5.5	73.9±7.1	**88.9±1.0**
HalfCheetah-m-e	42.7±0.3	86.1±9.7	70.3±21.9	74.5±6.4	**90.7±4.3**	86.7±5.3	66.5±7.8
Hopper-m-e	44.9±8.1	**111.6±2.3**	60.6±32.5	104.6±2.2	98.0±9.4	91.5±14.3	109.4±1.7
Walker2d-m-e	96.5±9.1	84.9±20.9	77.4±27.9	109.5±0.3	110.1±0.5	109.6±1.0	**111.7±0.4**
HalfCheetah-e	92.9±0.6	-	81.3±21.8	98.2±1.3	96.7±1.1	95.0±0.5	**102.3±1.0**
Hopper-e	110.5±0.5	-	62.5±29.0	107.7±2.4	107.8±7.0	109.4±0.5	**111.8±0.3**
Walker2d-e	108.4±0.4	-	62.4±3.2	109.4±0.1	110.2±0.3	109.9±1.2	**111.9±0.4**
Total	749.2	-	742.7	1027.1	992.1	1006.9	1076.3

Table 1: Normalized score comparison of all mentioned methods above on D4RL benchmark. All results are evaluated on '-v2' datasets, where m = medium, m-r = medium-replay, m-e = medium-expert, e = expert. Except for some results taken from the corresponding paper, we report the mean and standard deviation of score performance over four random seeds and bold the highest results.

anti-exploration for OOD data, which may lead to too much or too little regularization on OOD actions at some specific states, while our proposed adaptive reward shifting mechanism can adjust this strength according to the necessity of behavior constraint at each state, *i.e.*, the proximity between the learned and behavior policy and the effectiveness is verified by the outperformance of our PBRS.

5.3 Property Analysis

Adaptive reward shifting variation. To demonstrate the effectiveness of our adaptive reward shifting, we record the variation of $b_k(s)$ in the training process. Specifically, we record the mean and standard deviation of $b_k(s)$ for 'Walker2d-medium' and 'Walker2d-medium-expert' datasets. All the results are shown in the **top row** of Figure 1. We can see that the mean of $b_k(s)$ (left subplot) will decrease sharply at the beginning of training due to the warm-up of behavior cloning, and when starting training following our PBRS, the mean will gradually decrease and converge to a small level as the training process. This suggests that the learned policy by our PBRS would approach the behavior policy and the corresponding decrease in $b_k(s)$ can mitigate the potential excessive regularization on OOD actions, which is considered beneficial for policy improvement. As for the standard deviation of $b_k(s)$, the right subplot implies that the proximity to behavior policy varies significantly among different states, though this variation will decrease as the training process. These results demonstrate the necessity to adjust the strength of reward shifting at each state for more robust behavior constraints.

OOD actions regularization. In this part, to compare the strength of OOD action regularization, we compute the distance between the actions selected by the learned policy and the ones from the dataset. Mathematically, this action distance is formulated by $\mathbb{E}_{(s,a)\sim\mathcal{D},\hat{a}\sim\pi_\theta(\cdot|s)}\left[\|\hat{a}-a\|_2^2\right]$. We choose to estimate the action distance using a batch of samples from both datasets and plot the histograms of distance in the **middle row** of Figure 1. In particular, we compare

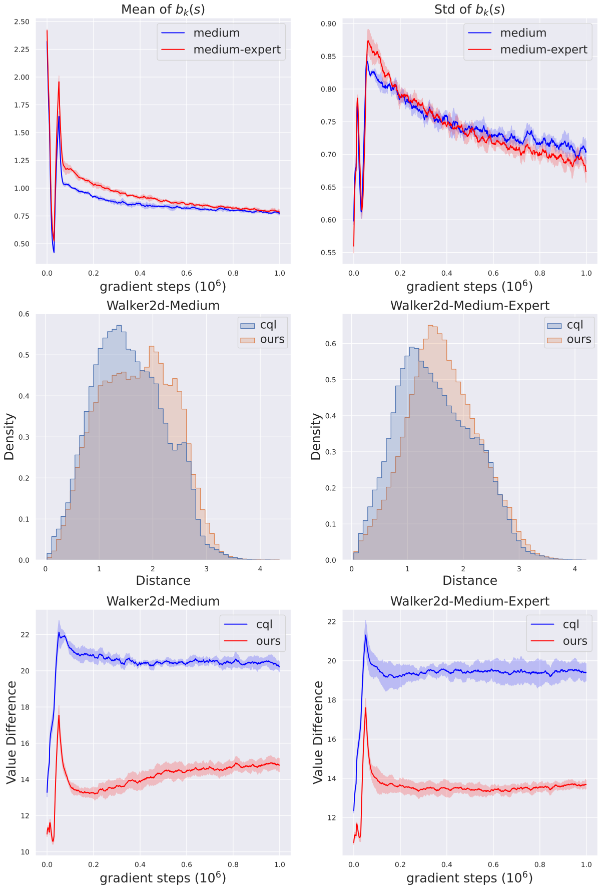

Figure 1: Property comparisons on both 'Walker2d-medium' and 'Walker2d-medium-expert' datasets. **top row:** Variation curves of mean and standard deviation about the adaptive reward shifting $b_t(s)$; **middle row:** Histograms of the distance between the actions from the learned policy and actions from the datasets; **bottom row:** Value difference between in-distribution and OOD actions.

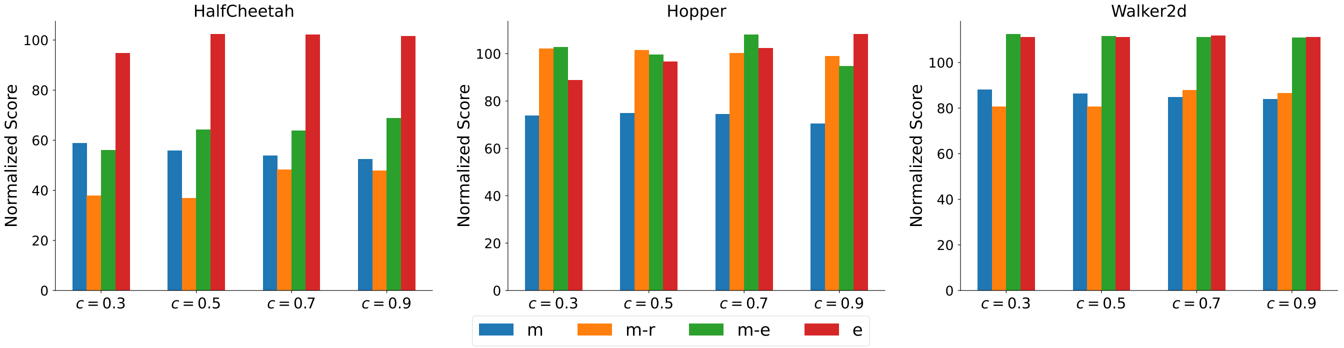

Figure 2: Performance comparison on the scaling coefficient candidates $c = \{0.3, 0.5, 0.7, 0.9\}$ over all the combinations of environments (HalfCheetah, Hopper, Walker2d) and datasets (m = medium, m-r =medium-replay, m-e = medium-expert, e = expert).

the action distance for CQL and our PBRS. The results show that our method owns a larger action distance than the CQL method; this suggests that the conservative-based method imposes a stronger OOD action regularization, leading to a more similar behavior with the dataset. While our PBRS can choose more diverse actions, which is thought of as the reason to achieve more policy improvement and better performance.

Q-value difference. Conservative Q-values of OOD actions play an important role in mitigating the potential overestimation issue, which could lead to the extrapolation error. In fact, these conservative OOD action values would enlarge the Q-value gaps between in-distribution and out-of-distribution actions. And the larger gaps could decrease the probability that OOD actions are chosen. So we further record and compare the value difference between different actions. Specifically, we use $\mathbb{E}_{(s,a)\sim\mathcal{D}}\left[Q(s,a) - \mathbb{E}_{a_{\text{ood}}}[Q(s,a_{\text{ood}})]\right]$ to estimate the value difference between in-distribution and out-of-distribution actions. And we sample actions from the mixture of a uniform action distribution and the learned policy as the approximation of a_{ood}. We compare the estimated value difference for the conservative-based method and our PBRS, and the results are shown in **bottom row** of Figure 1. From this, we can see that our method owns a smaller value difference between data in and out of distribution. However, the outperformance achieved by our PBRS method implies that the sufficiently conservative Q-values of OOD actions are not necessary to achieve better performance via OOD behavior regularization. And our proposed adaptive reward shifting is a reasonable and promising attempt to control the conservative Q-function adaptively.

Ablation study. According to the definition of Eq.(5), in addition to the policy distance, the adaptive reward shifting is also related to the scaling coefficient c. Though c doesn't adjust the reward shifting at each state, it still determines the magnitude of adaptive reward shifting and affects the performance trained on different datasets. In this section, we evaluate and compare the performance over all the combinations of three environments and four datasets, using different coefficients within the candidate set $c = \{0.3, 0.5, 0.7, 0.9\}$. We demonstrate the results in Figure 2 and provide the learning curves in Appendix. We can see that low-quality datasets, *e.g.*, 'medium' dataset, usually favor a small scaling coefficient (blue histograms), while for the high-quality ones, *e.g.*, 'expert' dataset, a large c is preferred to achieve better performance (red histograms). This indicates that a high-quality dataset with a relatively higher scale coefficient (larger c value) generally leads to better results, implying that more behavior constraints are necessary if the offline dataset is generated with a better behavior policy. We think both the scaling coefficient and the proximity-based adaptation are complementary because the former controls the overall strength of anti-exploration of OOD actions, and the latter can further refine it for each state.

6 Conclusion

As a linear transformation case of reward shaping, the simple reward shifting method is considered in this paper to control the exploitation and exploration effects through different reward shifting constants. Specifically, we propose to apply two opposite reward shifting schemes for in-distribution and out-of-distribution actions respectively, resulting in more exploitation of in-distribution data and less exploration of OOD data. However, the constant reward shifting for all states may ignore the balance between OOD behavior regularization and policy improvement. To address this issue, we propose a proximity-based adaptive mechanism to adjust the strength of reward shifting with respect to different states in the training process. The empirical results on the D4RL benchmark verify the feasibility and effectiveness of the proposed method.

For future studies, one interesting avenue is to consider more fine-grained reward shifting. For example, shifting the reward adaptively for each state-action pair so as to control the action regularization more precisely. Another interesting direction is to extend this adaptive reward shifting method to unobserved states for better policy generalization.

Acknowledgments

This work is partially supported by National Key R&D program of China (2021ZD0113203) and National Science Foundation of China (61976115). We would also like to thank the anonymous reviewers, for offering thoughtful comments and helpful advice on this work.

References

[Cabi *et al.*, 2020] Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott E. Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerík, Oleg Sushkov, David Barker, Jonathan Scholz, Misha Denil, Nando de Freitas, and Ziyu Wang. Scaling data-driven robotics with reward sketching and batch reinforcement learning. In *Robotics: Science and Systems XVI, Virtual Event / Corvalis, Oregon, USA, July 12-16, 2020*, 2020.

[Fu *et al.*, 2020] 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.

[Fujimoto and Gu, 2021] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[Fujimoto *et al.*, 2019] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[Grigorescu *et al.*, 2020] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[Konyushkova *et al.*, 2020] Ksenia Konyushkova, Konrad Zolna, Yusuf Aytar, Alexander Novikov, Scott Reed, Serkan Cabi, and Nando de Freitas. Semi-supervised reward learning for offline reinforcement learning. *arXiv preprint arXiv:2012.06899*, 2020.

[Kostrikov *et al.*, 2021] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pages 5774–5783. PMLR, 2021.

[Kostrikov *et al.*, 2022] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[Kumar *et al.*, 2019] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[Kumar *et al.*, 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[Laud, 2004] Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[Levine *et al.*, 2020] 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.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Mezghani *et al.*, 2022] Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric, and Karteek Alahari. Learning goal-conditioned policies offline with self-supervised reward shaping. In *CoRL-Conference on Robot Learning*, 2022.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Ng *et al.*, 1999] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

[Peng *et al.*, 2019] 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.

[Rezaeifar *et al.*, 2022] Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8106–8114, 2022.

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

[Sohn *et al.*, 2015] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[Sun *et al.*, 2022] Hao Sun, Lei Han, Rui Yang, Xiaoteng Ma, Jian Guo, and Bolei Zhou. Optimistic curiosity exploration and conservative exploitation with linear reward shaping. In *Advances in Neural Information Processing Systems*, 2022.

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

[Wu *et al.*, 2021] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua M. Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11319–11328. PMLR, 2021.

[Yu *et al.*, 2020] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[Yu *et al.*, 2021a] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[Yu *et al.*, 2021b] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.