ROER: Regularized Optimal Experience Replay

Changling Li
lichan@student.ethz.ch
ETH Zurich

Zhang-Wei Hong
zwhong@mit.edu
Massachusetts Institute of Technology

Pulkit Agrawal
pulkitag@mit.edu
Massachusetts Institute of Technology

Divyansh Garg
divgarg@stanford.edu
Stanford University

Joni Pajarinen
joni.pajarinen@aalto.fi
Aalto University

Abstract

Experience replay serves as a key component in the success of online reinforcement learning (RL). Prioritized experience replay (PER) reweights experiences by the temporal difference (TD) error empirically enhancing the performance. However, few works have explored the motivation of using TD error. In this work, we provide an alternative perspective on TD-error-based reweighting. We show the connections between the experience prioritization and occupancy optimization. By using a regularized RL objective with $f-$divergence regularizer and employing its dual form, we show that an optimal solution to the objective is obtained by shifting the distribution of off-policy data in the replay buffer towards the on-policy optimal distribution using TD-error-based occupancy ratios. Our derivation results in a new pipeline of TD error prioritization. We specifically explore the KL divergence as the regularizer and obtain a new form of prioritization scheme, the regularized optimal experience replay (ROER). We evaluate the proposed prioritization scheme with the Soft Actor-Critic (SAC) algorithm in continuous control MuJoCo and DM Control benchmark tasks where our proposed scheme outperforms baselines in 6 out of 11 tasks while the results of the rest match with or do not deviate far from the baselines. Further, using pretraining, ROER achieves noticeable improvement on difficult Antmaze environment where baselines fail, showing applicability to offline-to-online fine-tuning. Code is available at https://github.com/XavierChanglingLi/Regularized-Optimal-Experience-Replay.

1 Introduction

Deep reinforcement learning (RL) have shown wide applications in various domains (Mnih et al., 2015; Levine et al., 2016; Koert et al., 2019; Li et al., 2022; Hong et al., 2024). One key factor for its success is the integrated structure of experience replay (Zhang & Sutton, 2017). Experience replay (Lin, 1992) allows RL algorithms to use collected experience to compute updates for the current policy. It significantly increases the data efficiency and allows RL to be applied to fields where online data collection is expensive. On the other hand, sampling from experience replay buffer breaks the temporal correlations among experiences and stabilizes the gradient update (Mnih et al., 2013). However, past work shows that not all samples are equally informative in updating policy (Katharopoulos & Fleuret, 2018). To enhance the performance, techniques of weighted experience replay (Schaul et al., 2015; Kumar et al., 2020a; Liu et al., 2021; Sinha et al., 2022) are proposed to perform importance sampling and shape the distribution of the data in the replay buffer.

Among the proposed reweighting frameworks, prioritized experience replay (PER) is most commonly utilized for its simplicity and empirically good performance (Hessel et al., 2018). PER attempts to accelerate learning by assigning experiences with the temporal-difference (TD) error to enable higher sampling frequency for transitions with high error. However, PER inherits several shortcomings. First, experience replay reuses experiences from the past iterations to update the current policy. The resulted distribution shift between the data distribution of the replay buffer and the distribution of the current policy can cause incorrect TD error estimations which is detrimental to the performance of PER. On the other hand, it has been empirically shown that staying on policy (Schulman et al., 2015) or maintaining an on-policy sample distribution can be beneficial to the performance (Sutton & Barto, 2018; Fu et al., 2019; Novati & Koumoutsakos, 2019). Second, even though the motivation of using TD error is intuitive, limited works have explored the theoretical foundation (Fujimoto et al., 2020; Lahire et al., 2021).

In this work, we revisit the prioritization scheme and attempt to tackle the aforementioned problems. We provide a new perspective on the TD error prioritization by making connection to the occupancy optimization. We leverage the dual function of the regularized RL objective with f-divergence regularizer between off-policy and on-policy data distributions (Nachum et al., 2019b) and show that an optimal solution (occupancy ratio) to the objective is obtained by shifting the off-policy distribution towards the on-policy optimal distribution which results in a TD error prioritization. The form of TD error prioritization is closely associated with the regularized objective which implies that using simple TD error alone may not work best for every RL objectives. On the other hand, introducing regularizer into the objective penalizes TD-error estimation when the distribution of the data from the replay buffer differs too much from the distribution induced by the current policy and thus, gives a smaller priority to mitigate the bias induced by the distribution shift. Together, our derivation provides an alternative perspective on PER and results in a new pipeline of TD-error-based prioritization scheme whose form depends on the choice of the regularizer. Similar to PER, the new framework can be easily integrated with existing RL algorithms by using an additional value network with the regularized objective.

We specifically focus on KL-divergence as a regularizer and derive its corresponding objective. From this objective, we obtain a new form of prioritized experience replay, the regularized optimal experience replay (ROER). We combine our proposed ROER with Soft Actor-Critic (SAC) (Haarnoja et al., 2018) algorithm and evaluate on continuous control MuJoCo and DM control benchmark tasks. ROER outperforms baselines in 6 out of 11 tasks while the rest match with or do not deviate far from the best performance. Especially, ROER shows performance improvements on environments where PER and LaBER (Lahire et al., 2021) fails. Further evaluation on the value estimations shows that the performance improvement of ROER attributes to the more accurate value estimation by mitigating the underestimation bias of SAC with double critics (Li et al., 2021; Zhou et al., 2022) and thus ROER can obtain or converge to the optimal solutions much faster than baselines. Further, we consider the setting of online with pretraining and ROER achieves noticeable improvement on difficult Antmaze environment whereas the baselines fail, showing the applicability to offline-to-online fine-tuning.

2 Preliminaries

Online RL. Online reinforcement learning concerns optimizing an agent's policy in a Markov decision process (MDP) (Puterman, 2014). The MDP is defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$ where $\mathcal{S}$ and $\mathcal{A}$ represent the state space and action space respectively, $P(s'|s,a)$ denotes the dynamic model, $r(s,a)$ the reward function, ρ_0 the initial state distribution, and $\gamma \in (0,1)$ the discount factor. The agent's behavior is described by its policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$. The performance of a given policy can be measured by the state-action value function $Q^\pi(s,a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)|s_0 = s, a_0 = a, s_{t+1} \sim P(\cdot|s_t,a_t), a_t \sim \pi(\cdot|s_t)]$. The corresponding value function is $V^\pi(s) := \mathbb{E}[Q^\pi(s,a)|a \sim \pi(\cdot|s)]$. The goal is to learn a policy that maximizes the γ-discounted expected cumulative return (Sutton & Barto, 2018):

$$\max_{\pi} \mathcal{J}_P(\pi) := (1-\gamma)\mathbb{E}_{s_0 \sim \rho_0, a_0 \sim \pi(\cdot|s_0)}[Q^\pi(s_0,a_0)] \tag{1}$$

For a fixed policy π, we can rewrite the expected return in terms of its state-action distribution (Wang et al., 2007; Puterman, 2014) as

$$\max_{\pi} \mathcal{J}_D(\pi) := \mathbb{E}_{(s,a)\sim d^{\pi}}[r(s,a)] \tag{2}$$

where $d^{\pi}(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr[s_t = s, a_t = a | s_0 \sim \rho_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)]$.

In actor critic methods, one alternates between updating policy π (the actor) and Q-approximator Q_θ (the critic)(Konda & Tsitsiklis, 1999). The policy updates according to the policy gradient theorem (Sutton et al., 1999) as

$$\frac{\partial}{\partial \pi}\mathcal{J}_P(\pi) = \mathbb{E}_{(s,a)\sim d^{\pi}}[Q^{\pi}(s,a)\nabla \log \pi(a|s)] \tag{3}$$

The critic is learned via TD learning based on Bellman equation (Bellman, 1966) $\mathcal{B}^{\pi}Q^{\pi} := r(s,a) + \gamma \mathbb{E}_{s',a'}Q^{\pi}(s',a')$ where $\mathcal{B}^{\pi}$ denotes the expected Bellman operator. Given some experience replay buffer $\mathcal{D}$ collected in the same MDP but by potentially different policies, the Q-approximator is learned via a variation of the following form

$$\min_{Q_\theta} \mathcal{J}(Q_\theta) := \frac{1}{2}\mathbb{E}_{(s,a)\sim\mathcal{D}}[(\mathcal{B}^{\pi}Q_\theta - Q_\theta)(s,a)^2]. \tag{4}$$

In practice, we generally cannot access the true target value $B^{\pi}Q_\theta$ and thus, we use an estimation $\hat{B}^{\pi}Q_\theta$ to fit $Q_\theta(s,a)$.

Prioritization in Experience Replay. Prioritization in experience replay applies weighted sampling to the experiences by assigning weights to individual state-action which is equivalent to the weighted objective. We define the weight for a experience with state s and action a as $w(s,a)$ which is positive. Then, under the sampling distribution $d \in P(\mathcal{S}\times\mathcal{A})$, we have the weighted learning objective:

$$\min_{Q_\theta} \mathcal{J}(Q_\theta) := \frac{1}{2}\mathbb{E}_d[w(s,a)(\mathcal{B}^{\pi}Q_\theta - Q_\theta)(s,a)^2]. \tag{5}$$

In practice, $w(s,a)$ can be in various forms such as likelihood (Sinha et al., 2022). TD error is the most commonly considered and it forms prioritized experience replay (PER) which samples transitions proportional to their TD errors (Schaul et al., 2015). We denote TD error as δ and at time step t, it is defined as

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t). \tag{6}$$

Even though PER shows heuristically good results, the motivation of using TD error is under explored. In addition, using the TD error estimated by the Q-function induced by the current policy can be sub-optimal as the estimation of TD error can be inaccurate especially on the states that are infrequently visited under the current policy.

3 Experience Prioritization as Occupancy Optimization

The goal of using prioritization is to accelerate learning and obtain an optimal policy which induces an optimal on-policy distribution. We reverse this process and motivate our formulation by the problem of obtaining an optimal policy by finding the optimal on-policy distribution d^*, given access to an off-policy distribution $d^{\mathcal{D}}$. Here, d^* is unknown and we assume to only have samples from $d^{\mathcal{D}}$ which is the distribution of the experience replay buffer. For an MDP with a reward function r, there exists a unique d^*. We consider the following regularized objective with an f-divergence regularizer to include $d^{\mathcal{D}}$ (Nachum et al., 2019a)

$$\max_{d^{\mathcal{D}}} \mathcal{J}_{D,f}(d^*, d^{\mathcal{D}}) := \mathbb{E}_{(s,a)\sim d^*}[r(s,a)] - \beta D_f\left(d^* \| d^{\mathcal{D}}\right) \tag{7}$$

where $\beta > 0$ and D_f denotes the f-divergence induced by a convex function f:

$$D_f(d^*||d^{\mathcal{D}}) = \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f(w_{*/\mathcal{D}}(s,a))] \tag{8}$$

where $w_{*/\mathcal{D}} := \frac{d^*(s,a)}{d^{\mathcal{D}}(s,a)}$. The regularizer encourages conservative estimation and serves as a penalty when the off-policy distribution deviates too much from the on-policy distribution. Note that the strength of regularization can be controlled by β. The above objective $\mathcal{J}$ is maximized for $d^{\mathcal{D}} = d^*$, where it becomes the unconstrained RL problem.

The regularized objective (Equation 7) can be transformed to the following dual problem (Nachum et al., 2019b; Nachum & Dai, 2020). Let $x : S \times A \to \mathbb{R}$. We have the dual function of $\mathcal{J}$:

$$\tilde{\mathcal{J}}_{D,f}(d^*, d^{\mathcal{D}}) = \min_x \mathbb{E}_{(s,a)\sim d^*}[r(s,a)] + \beta \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f_*(x(s,a))] - \beta \mathbb{E}_{(s,a)\sim d^*}[x(s,a)]] \tag{9}$$

where f_* is the convex conjugate of f. Note that the optimal $x^*(s,a)$ w.r.t the dual objective satisfies $f'_*(x^*) = d^*/d^{\mathcal{D}}$. We apply the change of variable. Let $Q(s,a) - \gamma V^*(s') = -\beta x(s,a) + r(s,a)$ where $Q(s,a)$ is a fixed point of a variant of Bellman equation (Nachum et al., 2019a) and $\gamma V^*(s') + r(s,a) = \mathcal{B}^* Q(s,a)$. We obtain the new objective which is independent of d^*:

$$\tilde{\mathcal{J}}_{D,f}(d^*, d^{\mathcal{D}}) = \min_Q \ \beta \cdot \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[f_*\left((\mathcal{B}^* Q(s,a) - Q(s,a))/\beta\right)\right] + (1-\gamma)\mathbb{E}_{s_0\sim\mu_0, a_0\sim\pi^*(s_0)}\left[Q\left(s_0, a_0\right)\right] \tag{10}$$

Using $\delta_Q := \mathcal{B}^* Q(s,a) - Q(s,a)$ to denote the TD error, we obtain a solution Q^* to the objective satisfying:

$$f'_*(\delta_{Q^*}/\beta) = d^*/d^{\mathcal{D}}, \tag{11}$$

which gives the TD-error based occupancy ratio between the optimal distribution and the current distribution. We point out two key observations:

- Using the property of convex conjugate: $f'_*(f'(x)) = x$ and $f''(x) \geq 0$, we can rewrite $\mathcal{B}^* Q^*(s,a) - Q^*(s,a) = \beta f'(d^*/d^{\mathcal{D}})$. By absorbing the term $\beta \mathbb{E}_{(s,a)\sim d^*}[x(s,a)]]$ of the dual objective into the reward, we have that Q^* is the optimal Q-function for the augmented reward $\tilde{r} = r - \beta f'(d^*/d^{\mathcal{D}})$.
- When $d^* = d^{\mathcal{D}}$, as $f'(1) = 0$, Q^* is the optimal Q-function to the reward r and solves the unregularized RL problem of maximizing r.

Thus, in theory, the above problem has a unique saddle point solution where $d^{\mathcal{D}} = d^*$ and Q^* is the optimal Q-function, which can be found by shaping $d^{\mathcal{D}}$ towards d^* using the following weighting formulation:

$$d^* = f'_*(\delta_{Q^*}/\beta) \cdot d^{\mathcal{D}}. \tag{12}$$

We include the details of derivation in Appendix A. In practice, we have a changing distribution of $d^{\mathcal{D}}$ for online reinforcement learning due to the collection of the new data and update to the policy. However, we solve the optimization problem in many steps. We show that the distribution can still asymptotically converge to the optimal d^* by empirically showing that our proposed method mitigates the value underestimation bias of Soft-Actor Critic with double q-learning and converges to the true value in section 5.2.

4 Regularized Optimal Experience Replay

In this section, we discuss our choice of using Kullback-Leibler (KL) divergence as the regularizer and proceed to the practical implementation of the prioritization scheme with the KL-divergence regularizer which forms our proposed method, the regularized optimal experience replay (ROER). Other forms of f-divergence can also be suitable candidates and we provide further discussions in Appendix B.

4.1 KL Divergence as Regularizer

f-divergence consists of numerous forms and past works have explored the application of it in the policy update rules of RL (Belousov & Peters, 2017; Kumar et al., 2020b). Particularly, many works focus on KL-divergence as it improves the efficiency and performance of RL algorithms such as Trust Region Policy Optimization (Schulman et al., 2015) and Maximum a Posteriori Policy Optimization (Abdolmaleki et al., 2018). Theoretical exploration also shows the advantage of KL regularization (Vieillard et al., 2020). Thus, we consider KL-divergence as the regularizer in our formulation for it penalizing the off-policy distribution being too far from the on-policy distribution and the later-on derived objective.

Recall that the function of KL-divergence has the form $f(x) = x\log(x)$ and its convex conjugate has the form $f_*(y) = e^y - 1$. Let $y = (\mathcal{B}^*Q(s,a) - Q(s,a))/\beta$, we follow the derivation in section 3 and obtain the following objective:

$$\min_Q \ \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[e^{(\mathcal{B}^*Q(s,a)-Q(s,a))/\beta}\right] - \mathbb{E}_{(s,a,s')\sim d^{\mathcal{D}}}\left[\mathcal{B}^*Q(s,a) - Q(s,a)\right] - 1 \tag{13}$$

We note that this objective is reminiscent to the loss function of Extreme-Q learning (Garg et al., 2023) which leverages Extreme Value Theory to avoid computing Q-values using out of distribution actions and thus, mitigate the estimation error. This allows for obtaining a more accurate TD error for priority calculation. We note that our method differs from extreme q-learning as we only uses this loss to obtain TD error to shape the data distribution towards an optimal on-policy distribution. Using this objective, the occupancy ratio has the form

$$d^*/d^{\mathcal{D}} = f'_*(\delta_Q/\beta) = e^{\delta_Q/\beta} \tag{14}$$

which gives our proposed regularized optimal experience replay formulation.

4.2 Practical Implementation

Algorithm 1 Actor Critic with Regularized Optimal Experience Replay

```
1: Initialize $Q_\theta$, $\pi_\psi$, value network $V_\phi$, training start step $\tau$
2: Let $\mathcal{D}$ be the empty replay buffer or filled with offline data with $d(s,a) = 1$
3: for step $t$ in $1, ..., N$ do
4:     Update (s, a, r, s') to $\mathcal{D}$ with $d(s,a) = 1$
5:     if t $\geq \tau$ then
6:         Update $d(s,a)$ with $d'$ from Eq. 16
7:         Train $Q_\theta$ with $J(Q_\theta)$ from Eq. 5 using $d(s,a)$ as $w(s,a)$
8:         Train $V_\phi$ with $\mathcal{L}(V)$ from Eq. 15
9:         Update $\pi_\psi$
10:    end if
11: end for
12: return $Q^*, \pi^*$
```

Note that the form of occupancy ratio is derived from a regularized objective which can be different from the objective of the applied algorithm. For smooth integration to the existing algorithms, we propose to incorporate a separate value network using the regularized objective for TD error estimation and priority calculation. The above KL divergence gives the value network the following objective of the ExtremeV loss (Garg et al., 2023)

$$\mathcal{L}(\mathcal{V}) = \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[e^{(\mathcal{B}^*Q(s,a)-V(s))/\beta}\right] - \mathbb{E}_{(s,a,s')\sim d^{\mathcal{D}}}\left[Q(s,a) - V(s)\right] - 1. \tag{15}$$

We then use the TD error obtained from the value network to calculate the priority. Since the distribution of $d^{\mathcal{D}}$ is changing, we consider a stable convergence and solve the optimization problem in many steps. We introduce a convergence parameter λ and formulate the following priority update function

$$d' = [\lambda e^{\delta_{Q^*}/\beta} + (1-\lambda)] \cdot d^{\mathcal{D}} \text{ with } \lambda \in (0,1]. \tag{16}$$

where $d^{\mathcal{D}}$ is the current priority of the samples and d' is the updated priority (used as w). In an online setting, we start by assigning each sample in replay buffer $\mathcal{D}$ with priority $d = 1$ and use the above update function to update the priority after each Q-iteration step. The loss temperature β here controls the scale of TD-error and thus, the scale of the priority. We note that exponential function is sensitive to outliers. Thus, we use mean normalization and clip the exponential of TD error and the priority to control the range and avoid outliers. The general procedure of our approach is summarized in Algorithm 1 and more implementation details are listed in Appendix C. The proposed priority update function slowly improves the current distribution d' towards the optimal policy distribution d^*, and ultimately maximizes the objective $\mathcal{J}$.

5 Experimental Evaluation

We combine our proposed prioritization scheme ROER with Soft-Actor Critic (Haarnoja et al., 2018) algorithm for evaluation. We compare our method with two state-of-art prioritization schemes namely uniform experience replay (UER) and the initial TD error prioritized experience replay (PER) (Schaul et al., 2015), and one additional baseline namely large batch experience replay (LaBER) (Lahire et al., 2021) across a wide set of MuJoCo continuous control tasks interfaced through OpenAI Gym (Brockman et al., 2016) and DM Control tasks (Tunyasuvunakool et al., 2020) in an online setting. Additionally, we consider a suite of more difficult environment Antmaze with pretraining using the data from D4RL (Fu et al., 2020) to show that ROER can achieve good performance in settings where both UER and PER fail. To allow for reproducibility, we use the original set of tasks without modification to the environment or rewards. For a fair comparison between baselines and our approach, our implementations are all based on JAXRL (Kostrikov, 2021).

Compared to the initial PER, even though our proposed method ROER has four more hyperparameters namely the architecture of value network, loss temperature (β), Gumbel loss clip (Grad Clip), and maximum exponential of TD-error clip (Max Exp Clip), we note that β and Grad Clip are not new and they come from the objective of Extreme Q-Learning. Grad Clip is shown to affect the results lightly and the value network can use the default parameters as the critic network. A set of values works well for multiple environments. We provide more details of implementation, ablations and hyperparameters in Appendix C.

5.1 Online

Env	SAC	SAC+PER	SAC+LaBER	SAC+ROER (ours)
Ant-v2	1153.1 ± 335.5	1654.1 ± 342.9	1006.0 ± 546.0	$\mathbf{2275.5} \pm 598.6$
HalfCheetah-v2	9017.4 ± 172.5	9240.4 ± 276.5	7962.8 ± 304.5	$\mathbf{10695.5} \pm 183.4$
Hopper-v2	2813.0 ± 481.2	2937.7 ± 334.3	2330.8 ± 514.3	$\mathbf{3010.2} \pm 299.0$
Humanoid-v2	5026.8 ± 154.1	4993.4 ± 198.0	5000.9 ± 319.5	$\mathbf{5257.0} \pm 153.2$
Walker2d-v2	$\mathbf{4344.3} \pm 177.7$	4003.9 ± 318.7	4033.1 ± 375.7	4328.5 ± 311.4
Fish-swim	247.7 ± 59.6	234.6 ± 63.6	178.3 ± 49.9	$\mathbf{301.9} \pm 54.9$
Hopper-hop	134.4 ± 34.2	$\mathbf{147.2} \pm 31.3$	146.7 ± 29.8	125.7 ± 35.2
Hopper-stand	521.1 ± 120.1	384.7 ± 94.9	475.5 ± 111.0	$\mathbf{798.5} \pm 89.2$
Humanoid-run	130.3 ± 21.7	116.3 ± 18.7	$\mathbf{144.8} \pm 18.1$	137.3 ± 12.3
Humanoid-stand	733.4 ± 53.9	765.0 ± 38.8	$\mathbf{827.8} \pm 40.9$	691.6 ± 57.8
Quadruped-run	761.2 ± 89.4	606.2 ± 114.7	$\mathbf{796.3} \pm 82.6$	772.1 ± 77.7

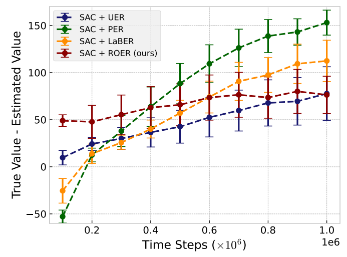
Table 1: Average evaluation performance attained over the last 10 evaluations over 1 million time steps for MuJoCo and DM Control tasks. Average performance and 95% confidence interval ($\pm$) are attained over 20 random seeds. Maximum average value for each task is highlighted as bold

In the online setting, the empirical results demonstrate that our proposed ROER outperforms state-of-the-arts on 6 out of 11 continuous control tasks in terms of the average evaluation while do not deviate far from the baselines for the rest 3 environments as shown in Table 1. We find that ROER with SAC achieves noticeable improvement on HalfCheetah-v2, Ant-v2, Humanoid-v2 from MuJoCo and Fish-Swim, Hopper-stand from DM Control. Especially for Hopper-stand environment, our proposed ROER with SAC nearly doubles the performance of UER or PER with SAC. We attribute the improvements to the more accurate TD error estimation using a separate value network with divergence regularized objective and the associated priority update form. Within the five underperformed tasks, ROER obtains a similar performance as the UER in Walker2d-v2 and outperforms PER and LaBER.

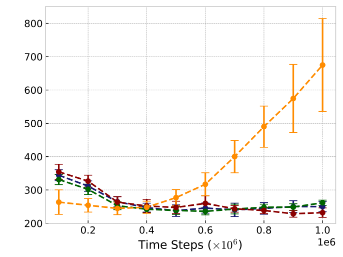
In contrast, PER only shows slight improvement on limited number of continuous control tasks compared to UER including HalfCheetah-v2, Hopper-v2, Hopper-hop, and Humanoid-stand. In many tasks, PER even worsens the performance such as Walker2d, Hopper-stand, Humanoid-run and Quadruped-run whereas our proposed method can maintain a similar or achieve much better performance. We consider the reasons for PER failing to be the biased priority induced by the inaccurate TD error estimation and the less stable priority update scheme. We note that LaBER achieves better results in Humanoid-run, Humanoid-stand and Quadruped-run but in the cost of much longer training time due to the larger batch required by the algorithm. The hyperparameter selection for LaBER can be found in Appendix C.

We notice that all three prioritization schemes under perform in Ant-v2. Our proposed ROER with SAC, even though achieves higher average performance in Ant-v2, has a very large confidence interval. This requires additional tuning to hyperparameters of SAC and training steps. We keep our current results for a fair comparison across tasks. Evaluation curves and additional discussion of results can be found in Appendix D.

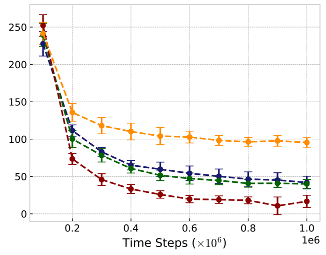
5.2 Value Estimation Analysis



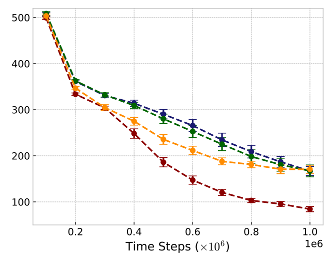
(a) Ant-v2



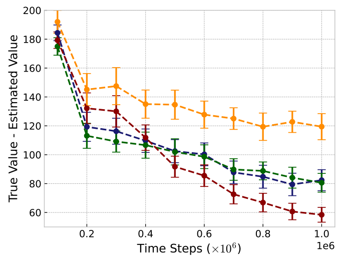
(b) HalfCheetah-v2



(c) Hopper-v2



(d) Humanoid-v2



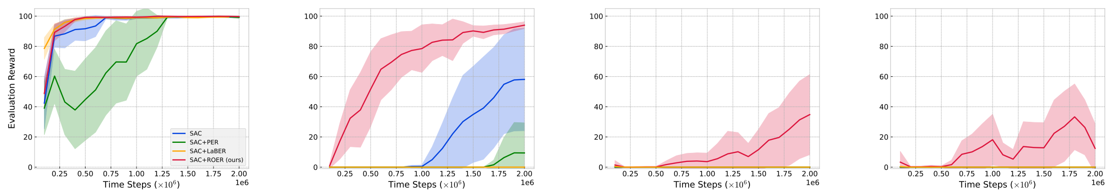
(e) Walker2d-v2

Figure 1: Measuring underestimation bias in the value estimates of SAC, SAC with PER and SAC with ROER of continuous control tasks in MuJoCo by the difference between the true values and the value estimates. True value is obtained by Monte Carlo returns. Value estimates and true values are averaged over 20 random seeds and the error bar represents 95% confidence interval.

The better performance can be empirically confirmed by the faster convergence to the true value. Besides the derivation that shows our proposed prioritization scheme results in the optimal solution, ROER also demonstrates empirically more accurate value estimation and faster convergence than the baselines. We compare the value estimation with the true value for each algorithm trained online over tasks in MuJoCo as shown in Fig. 1. SAC with double critics tends to underestimate the value (Li et al., 2021; Zhou et al., 2022). Compared to the baselines, ROER reduces the underestimation biases and converge to the true values much faster especially in Hopper-v2, Humanoid-v2 and Walker2d-v2. We note that ROER reaches the true value in reward saturated cases such as Hopper-v2 while the baselines still show the underestimation bias. This result serves as an empirical evidence that our proposed prioritization scheme reshapes the replay buffer towards the optimal on-policy distribution and results in the optimal Q^* which is the solution to the objective.

5.3 Online with Pretraining

Our proposed ROER prioritization scheme can benefit from pretraining using offline data and show significant performance improvement over more difficult environment Antmaze-Umaze and Antmaze-Medium as revealed in Table 2. We recognize that using the average performance over the last 200 evaluations may not be the most suitable metric here due to the sparsity of rewards and the difficulty of the environments. Thus, we also include the learning curves to illustrate the results as in Fig. 2. We found that SAC with ROER can obtain good performance at a very early stage in Antmaze-Umaze environment using both Antmaze-Umaze-v2 dataset and Antmaze-Umaze-diverse-v2 dataset. Especially with Antmaze-Umaze-diverse-v2 dataset, SAC with ROER achieved a significantly better performance compared to the state-of-art prioritization schemes while PER and LaBER are shown to be detrimental to the learning process. We also note that for a more difficult environment Antmaze-Medium, our proposed method can obtain rewards at an early stage and shows improvement over training steps as in Fig. 2c and Fig. 2d. In contrast, SAC with UER, SAC with PER and SAC with LaBER completely fail to obtain any reward signal. This implication is crucial to improve training efficiency and safety by using offline data and correct the distribution to obtain a good performance online. It shows the potential applicability of our method in offline-to-online finetuning.



(a) Umaze-V2 (b) Umaze-Diverse-V2 (c) Medium-Play-V2 (d) Medium-Diverse-V2

Figure 2: Learning curves for the Antmaze tasks in Gym-Robotics with data from D4RL. Curves are averaged over 10 random seeds, where the shaded area represents the standard error of the average evaluation.

Env	SAC	SAC+PER	SAC+LaBER	SAC+ROER (ours)
antmaze-umaze-v2	99.6 ± 0.4	99.6 ± 0.5	99.45 ± 0.5	$\mathbf{99.9} \pm 0.2$
antmaze-umaze-diverse-v2	57.8 ± 22.7	9.5 ± 13.5	0.0	$\mathbf{92.7} \pm 1.9$
antmaze-medium-play-v2	0.0	0.0	0.0	$\mathbf{31.3} \pm 17.3$
antmaze-medium-diverse-v2	0.0	0.0	0.0	$\mathbf{26.3} \pm 14.0$

Table 2: Average evaluation performance attained over the last 200 evaluations over $2e6$ time steps after Antmaze environments. Average performance and 95% confidence interval ($\pm$) are attained over 10 random seeds.

6 Related Work

Our approach builds upon regularized RL objective and weighted experience replay.

Regularized RL. Regularization is commonly utilized in offline reinforcement learning to constrain the behavior policy and action selection (Kumar et al., 2020b; Wu et al., 2019; Kumar et al., 2019). Other works have considered regularized Q-function of the behavior policy (Shi et al., 2023) and state-action value offset (Kostrikov et al., 2021). Adapting such regularizers in online setting can achieve more stable performance (Fujimoto et al., 2019; Schulman et al., 2015). Maximizing the regularizer as a way to encourage exploration also shows improvement in performance and forms the framework of max entropy RL (Ziebart et al., 2008; Haarnoja et al., 2017; 2018). Our work builds upon the line of work that utilizes the dual function of the regularized objective (Belousov & Peters, 2017; Nachum & Dai, 2020; Nachum et al., 2019b) which allows to express the max-return optimization by an expectation over an arbitrary behavior-agnostic and off-policy data distribution. We extend this approach and formulate the prioritization scheme that allows the data distribution in replay buffer gradually converge to the optimal distribution which gives the optimal Q-function. Theoretical analysis of the regularized RL shows that despite its non-convexity, this problem has zero duality gap and can be solved exactly in the dual domain (Geist et al., 2019; Neu et al., 2017; Paternain et al., 2019).

Weighted experience replay. Experience replay is crucial to the success of deep RL for improving the data efficiency by using off-policy data (Lin, 1992; Hessel et al., 2018). Various frameworks have been proposed to change the sampling strategy to achieve superior performance than uniform sampling. Prioritized experience replay (PER) weights the experiences by their TD errors and shows empirical improvement when applying to deep RL (Schaul et al., 2015; Fujimoto et al., 2020). However, few works have explored the theoretical motivation of using TD-error based reweighting scheme. Lahire et al. (2021) suggest that PER can be considered as an importance sampling scheme using approximated per-sample gradient norms and prioritizing stochastic gradient descent variance reduction. Our work, on the other hand, uses dual function of the regularized RL objective to provide an alternative perspective on TD-error-based prioritization. Other considerations of prioritization scheme include loss value (Hessel et al., 2018), accuracy of the TD-error estimation (Sinha et al., 2022), regret minimization (Liu et al., 2021), and leveraging neural network for experience selection (Zha et al., 2019). We note that Kumar et al. (2020a) shares similarity to our work in correcting the replay buffer towards optimal distribution. However, they consider optimizing corrective feedback while our work builds on dual function of regularized RL objective. Another work that shares slight similarity with our method is ReF-ER (Novati & Koumoutsakos, 2019) where they ignore the updates from experiences that deviates significantly from the current policy. Our work focuses on penalizing the TD errors of the samples that deviate from the current policy which leads to smaller priority instead of completely ignoring those experiences. In addition, our proposed method forms a new pipeline of TD-error-based prioritization scheme.

7 Conclusion

By leveraging the regularized RL objective and its dual function, we propose a new pipeline of TD-error-based prioritization scheme that is more robust towards distribution shift between off-policy data and current policy. By considering KL-divergence as the reuglarizer, we formulated a new prioritized experience replay, namely regularized optimal experience replay (ROER). Our proposed ROER when applied to SAC empirically demonstrates the ability of mitigating the underestimation bias and shows faster convergence to the true value. It outperforms baselines in 6 out of 11 continuous control tasks in the online setting and significantly improves the performance in Antmaze with pretraining. However, we recognize the tuning of additional hyperparameters can limit the application. Future work can explore an adaptive loss temperature to dynamically adjust the strength of the regularization. Additionally, it would be valuable to extend the application of the proposed method to offline setting and further explore the applicability to offline-to-online fine tuning.

Limitations. Although our method provides theoretical motivations and empirically shows performance improvement and convergence to the optimal solutions over various environments, we lack theoretical guarantees to ensure the convergence. Few works have provided the theoretical ground of analyzing convergence using f-divergence regularizer (Paternain et al., 2019). More exploration is required to understand the convergence in online setting.

Acknowledgments

This research was supported in part by Quanta Computer Inc. and ONR MURI under Grant Number N00014-22-1-2740. We acknowledge access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium. J. Pajarinen was partly supported by Research Council of Finland (345521,353198).

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Boris Belousov and Jan Peters. f-divergence constrained policy improvement. *arXiv preprint arXiv:1801.00056*, 2017.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pp. 2021–2030. PMLR, 2019.

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.

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

Scott Fujimoto, David Meger, and Doina Precup. An equivalence between loss functions and non-uniform sampling in experience replay. *Advances in neural information processing systems*, 33: 14219–14230, 2020.

Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent rl without entropy. *arXiv preprint arXiv:2301.02328*, 2023.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169. PMLR, 2019.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

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*, pp. 1861–1870. PMLR, 2018.

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 *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zhang-Wei Hong, Idan Shenfeld, Tsun-Hsuan Wang, Yung-Sung Chuang, Aldo Pareja, James Glass, Akash Srivastava, and Pulkit Agrawal. Curiosity-driven red-teaming for large language models. *arXiv preprint arXiv:2402.19464*, 2024.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018.

Dorothea Koert, Joni Pajarinen, Albert Schotschneider, Susanne Trick, Constantin Rothkopf, and Jan Peters. Learning intention aware online adaptation of movement primitives. *IEEE Robotics and Automation Letters*, 4(4):3719–3726, 2019.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Ilya Kostrikov. JAXRL: Implementations of Reinforcement Learning algorithms in JAX, 10 2021. URL https://github.com/ikostrikov/jaxrl.

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

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.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. *Advances in Neural Information Processing Systems*, 33: 18560–18572, 2020a.

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, 2020b.

Thibault Lahire, Matthieu Geist, and Emmanuel Rachelson. Large batch experience replay. *arXiv preprint arXiv:2110.01528*, 2021.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Sicen Li, Qinyun Tang, Yiming Pang, Xinmeng Ma, and Gang Wang. Balancing value underestimation and overestimation with realistic actor-critic. *arXiv preprint arXiv:2110.09712*, 2021.

Ying Li, Changling Li, Jiyao Chen, and Christine Roinou. Energy-aware multi-agent reinforcement learning for collaborative execution in mission-oriented drone networks. In *2022 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–9. IEEE, 2022.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8:293–321, 1992.

Xu-Hui Liu, Zhenghai Xue, Jingcheng Pang, Shengyi Jiang, Feng Xu, and Yang Yu. Regret minimization experience replay in off-policy reinforcement learning. *Advances in Neural Information Processing Systems*, 34:17604–17615, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019a.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *ArXiv*, abs/1912.02074, 2019b.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Guido Novati and Petros Koumoutsakos. Remember and forget for experience replay. In *International Conference on Machine Learning*, pp. 4851–4860. PMLR, 2019.

Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

Henning Petzka, Asja Fischer, and Denis Lukovnicov. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, 2014.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Laixi Shi, Robert Dadashi, Yuejie Chi, Pablo Samuel Castro, and Matthieu Geist. Offline reinforcement learning with on-policy q-function regularization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 455–471. Springer, 2023.

Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. Experience replay with likelihood-free importance weights. In *Learning for Dynamics and Control Conference*, pp. 110–123. PMLR, 2022.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020.

Tao Wang, Michael Bowling, and Dale Schuurmans. Dual representations for dynamic programming and reinforcement learning. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 44–51. IEEE, 2007.

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

Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Experience replay optimization. *arXiv preprint arXiv:1906.08387*, 2019.

Shangtong Zhang and Richard S Sutton. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*, 2017.

Haibin Zhou, Zichuan Lin, Junyou Li, Qiang Fu, Wei Yang, and Deheng Ye. Revisiting discrete soft actor-critic. *arXiv preprint arXiv:2209.10081*, 2022.

Brian D. Ziebart, Andrew L. Maas, J. Bagnell, and A. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.

A Derivation Details

In this section, we provide the detailed derivation of our method for completeness. We reference Nachum et al. (2019b) for the derivation. We start by using the regularized max-return objective with divergence term between the on-policy optiaml distribution d^* and off-policy distribution $d^{\mathcal{D}}$

$$\max_\pi \mathcal{J}_{D,f}(\pi) := \mathbb{E}_{(s,a)\sim d^*}[r(s,a)] - \beta D_f(d^* || d^{\mathcal{D}}), \tag{17}$$

where $\beta > 0$ and D_f denotes the f-divergence induced by a convex function f:

$$D_f(d^*||d^{\mathcal{D}}) = \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f(w_{*/\mathcal{D}}(s,a))], \tag{18}$$

where $w_{*/\mathcal{D}} := \frac{d^*(s,a)}{d^{\mathcal{D}}(s,a)}$.

We then transform the f-divergence to its variational form using a dual function $x : S \times A \to \mathbb{R}$ that is bounded which gives the following expressions

$$\begin{aligned}\tilde{\mathcal{J}}_{D,f}(\pi, x) :&= \min_x \mathbb{E}_{(s,a)\sim d^*}[r(s,a)] + \beta \cdot \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f_*(x(s,a))] - \beta \cdot \mathbb{E}_{(s,a)\sim d^*}[x(s,a)] \\ &= \min_x \mathbb{E}_{(s,a)\sim d^*}[r(s,a) - \beta \cdot x(s,a)] + \beta \cdot \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f_*(x(s,a)]. \end{aligned} \tag{19}$$

Here, f_* is the convex conjugate of f. Recall the definition of convex conjugate: the convex conjugate of $f(x)$ is defined as $f_*(x) = \sup_{x\in \mathrm{dom} f}\{\langle y, x\rangle - f(x)\}$, where $\langle y, x\rangle$ denotes the dot product (Boyd & Vandenberghe, 2004).

To eliminate the dependence on d^*, we use change of variables and let $Q(s,a) - \gamma V^*(s') = -\beta x(s,a) + r(s,a)$. Applying the change of variable to Eq.19, we obtain:

$$\begin{aligned}\mathcal{J}_{D,f}(\pi, Q) := \min_Q \mathbb{E}_{(s,a)\sim d^*}&[r(s,a) + Q(s,a) - \gamma V^*(s') - r(s,a)] \\ &+ \beta \cdot \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f_*(\gamma V^*(s') - Q(s,a) + r(s,a))/\beta]. \end{aligned} \tag{20}$$

Note that $B^*Q(s,a) = r(s,a) + \gamma V^*(s')$. We simplify the above as

$$\mathcal{J}_{D,f}(\pi, Q) := \min_Q \mathbb{E}_{(s,a)\sim d^*}[Q(s,a) - \gamma V^*(s')] + \beta \cdot \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f_*(B^*Q(s,a) - Q(s,a))/\beta]. \tag{21}$$

Since $x(s,a)$ is bounded and $\gamma < 1$, $Q(s,a)$ is also bounded. Define

$$\Omega_t(s) := \Pr(s = s_t | s \sim \Omega, a_k \sim \pi^*(s_k), s_{k+1} \sim P(\cdot|s_k, a_k) \text{ for } 0 \le k \le t), \tag{22}$$

as the state visitation probability at step t following policy π^* (Nachum et al., 2019a). Then by telescoping, we have the following process for $\mathbb{E}_{(s,a)\sim d^*}[Q(s,a) - \gamma V^*(s')]$ (denoting this term as $*$)

$$\begin{aligned}
* &= \mathbb{E}_{(s,a)\sim d^*}[Q(s,a) - \gamma \mathbb{E}_{s'\sim P(\cdot|s,a), a'\sim\pi^*(s')}[Q(s',a')]] \\
&= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{E}_{s\sim\Omega_t, a\sim\pi^*(s)}[Q(s,a) - \gamma\mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi^*(s')}[Q(s',a')]] \\
&= (1-\gamma)\sum_{t=1}^{\infty}\gamma^t \mathbb{E}_{s\sim\Omega_t, a\sim\pi^*(s)}[Q(s,a)] - (1-\gamma)\sum_{t=1}^{\infty}\gamma^{t+1}\mathbb{E}_{s\sim\Omega_t,a\sim\pi^*(s)}[Q(s,a)] \\
&= (1-\gamma)\mathbb{E}_{s\sim\Omega, a\sim\pi^*(s)}[Q(s,a)]. \qquad (23)
\end{aligned}$$

Applying the above result to the dual objective, we obtain the final objective:

$$\mathcal{J}_{D,f}(\pi, Q) := \min_Q \beta\cdot\mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}[f_*(B^*Q(s,a) - Q(s,a))/\beta] + (1-\gamma)\mathbb{E}_{s_0\sim\mu_0, a_0\sim\pi^*(s_0)}[Q(s_0,a_0)] \quad (24)$$

which completes the derivation.

B Other divergence

In this section, we firstly show the connection between the dual objective and the actor-critc objective. Then we give another consideration of regularizer which results in a different form of prioritization.

B.1 Derivation Details of ROER

KL-divergence has the form $f(x) = x\log(x)$ and its convex conjugate has the form $f_*(y) = e^y - 1$. Let $y = (\mathcal{B}^*Q(s,a) - Q(s,a))/\beta$, we follow the derivation in section 3 and obtain the following dual objective:

$$\min_Q \ \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[e^{(\mathcal{B}^*Q(s,a)-Q(s,a))/\beta}\right] + (1-\gamma)\mathbb{E}_{s_0\sim\mu_0}\left[V^*(s_0)\right] - 1 \quad (25)$$

which can be expanded to:

$$\min_Q \ \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[e^{(\mathcal{B}^*Q(s,a)-Q(s,a))/\beta}\right] + \mathbb{E}_{(s,a,s')\sim d^{\mathcal{D}}}\left[V^*(s) - \gamma V^*(s')\right] - 1. \quad (26)$$

Recall that $\gamma V^*(s') = \mathcal{B}^*Q(s,a) - r(s,a)$. We substitute the expression of $\gamma V^*(s')$ back to the above objective and obtain

$$\min_Q \ \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[e^{(\mathcal{B}^*Q(s,a)-Q(s,a))/\beta}\right] + \mathbb{E}_{(s,a,s')\sim d^{\mathcal{D}}}\left[V^*(s) - \mathcal{B}^*Q(s,a) + r(s,a)\right] - 1, \quad (27)$$

and we can further simplify the expression and obtain

$$\min_Q \ \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[e^{(\mathcal{B}^*Q(s,a)-Q(s,a))/\beta}\right] - \mathbb{E}_{(s,a,s')\sim d^{\mathcal{D}}}\left[\mathcal{B}^*Q(s,a) - Q(s,a)\right] - 1 \quad (28)$$

which completes the derivation. This objective corresponds to ExtremeQ loss as in Garg et al. (2023). It has a corresponding ExtremeV loss in the following form:

$$\mathcal{L}(\mathcal{V}) = \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[e^{(\mathcal{B}^*Q(s,a)-V(s))/\beta}\right] - \mathbb{E}_{(s,a,s')\sim d^{\mathcal{D}}}\left[Q(s,a) - V(s)\right] - 1 \quad (29)$$

which is the objective of our value network.

B.2 Connection to Actor Critic

Recall that the dual function of the regularized RL objective with the change of variable has the following form

$$\mathcal{J}_{D,f}(\pi, Q) = \min_{Q} \ \beta \cdot \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}} \left[f_* \left((\mathcal{B}^* Q(s,a) - Q(s,a))/\beta\right)\right] + (1-\gamma)\mathbb{E}_{s_0\sim\mu_0, a_0\sim\pi^*(s_0)} \left[Q\left(s_0, a_0\right)\right] \tag{30}$$

We consider a convex function of the form $f(x) = \frac{1}{2}x^2$. Its convex conjugate has the same form as itself $f_*(y) = \frac{1}{2}y^2$. Let $y := \mathcal{B}^*Q(s,a) - Q(s,a)$. The above function can be expressed as below:

$$\mathcal{J}_{D,f}(\pi, Q) = \min_{Q} \ \frac{1}{2\beta} \cdot \mathbb{E}_{(s,a)\sim d^{D}} \left[\left(\mathcal{B}^* Q(s,a) - Q(s,a)\right)^2\right] + (1-\gamma)\mathbb{E}_{s_0\sim\mu_0, a_0\sim\pi^*(s_0)} \left[Q\left(s_0, a_0\right)\right] \tag{31}$$

which transforms the off-policy actor-critic to an on-policy actor-critic by introducing the second term. This unifies the two separate objectives of value and policy into a single objective and both functions are trained with respect to the same off-policy objective (Nachum et al., 2019b).

B.3 Pearson χ^2 Divergence

A variety of f-divergence can be suitable candidates for the dual objective and prioritization derivation. Here, we provide a list of f-divergences $f(x)$, its corresponding convex conjugates $f_*(y)$ and the potential priority forms $f'_*(y)$ in Table 3. We note that the forms presented are theoretical forms and they may vary when applying to the RL objectives.

Divergence	$f(x)$	$f_*(y)$	$f'_*(y)$
KL	$x \log x$	$e^y - 1$	d^y
Reverse KL	$-\log x$	$-\log(1-y)$	$\frac{1}{1-y}$
Pearson χ^2	$\frac{1}{2}(x-1)^2$	$\frac{1}{2}y^2 + y$	$y+1$
Neyman χ^2	$\frac{(x-1)^2}{2x}$	$-\sqrt{1-2y}+1$	$\frac{1}{\sqrt{1-2y}}$
Total variation	$\frac{1}{2}\|x-1\|$	y	1
Squared Hellinger	$2(\sqrt{x}-1)^2$	$\frac{2y}{2-y}$	$\frac{4}{(2-y)^2}$

Table 3: List of f-divergence functions $f(x)$, convex conjugates $f_*(y)$ and the potential priority forms $f'_*(y)$.

We use Pearson χ^2 divergence as an example as the resulting objective has a particular implication. Pearson χ^2 divergence has the form $f(x) = \frac{1}{2}(x-1)^2$ and its convex conjugate has the form $f_*(y) = \frac{1}{2}y^2 + y$. Again, let $y := \mathcal{B}^*Q(s,a) - Q(s,a)$. We can obtain the following dual objective:

$$\min_{Q} \ \frac{1}{2\beta} \cdot \mathbb{E}_{(s,a)\sim d^{D}} \left[\left(\mathcal{B}^* Q(s,a) - Q(s,a)\right)^2\right] + \mathbb{E}_{(s,a)\sim d^{D}} \left[\mathcal{B}^* Q(s,a) - Q(s,a)\right] + (1-\gamma)\mathbb{E}_{s_0\sim\mu_0} \left[V^*\left(s_0\right)\right] \tag{32}$$

Using $\gamma V^*(s') + r(s,a) = \mathcal{B}^*Q(s,a)$, the objective can be further simplified to:

$$\min_{Q} \ \frac{1}{2\beta} \cdot \mathbb{E}_{(s,a)\sim d^{D}} \left[\left(\mathcal{B}^* Q(s,a) - Q(s,a)\right)^2\right] + \mathbb{E}_{(s,a)\sim d^{D}} \left[V^*(s) - Q(s,a)\right] \tag{33}$$

We note that this corresponds to the learning objective of conservative Q-learning (Kumar et al., 2020b). This objective is optimized when $d^*/d^D = f'_*(\delta^*_Q/\beta) = \delta_Q/\beta + 1$ which gives a new form of priority calculation. We address that even though this form is almost identical to PER, the source of δ is different. We derive this priority form from the conservative Q-learning objective and thus, it requires the value network to use the corresponding loss function. In addition, the loss temperature β here also controls the scale of the TD error and the strength of the regularizer which we expect to give better performance than the naive PER.

C Experiments

In this section, we provide details of the implementation and the experiments with further discussion on hyperparameter ablation and selection.

C.1 Experimental Details

Environment

In the online setting, our agents are evaluated in MuJoCo via OpenAI gym interface using the v2 environments (Brockman et al., 2016) and DM Control tasks (Tunyasuvunakool et al., 2020). For the MuJoCo environment, we do not modify or preprocess the state space, action space and reward function for easy reproducibility. For the DM Control tasks, we adapt to the gym environment interface. In the online with pretraining settings, our agents are evaluated in the environment with D4RL datasets (Fu et al., 2020). We shaped the reward of Antmaze by subtracting 1 as suggested in Kumar et al. (2020b) which shows to largely benefit the performance in Antmaze. Each environment runs for a maximum of 1000 time steps which is the default setting or until a termination state is reached.

Value estimation

Value estimates are averaged over mini-batches of 256 and sampled every 2000 iterations. The true value is estimated by sample 256 state-action pairs from the replay buffer and compute the discounted return by running the episode following the current policy until termination.

Reward Evaluation

In the online setting, we evaluate the current policy over 10 episodes for every 5000 training steps. The evaluation reward takes the average over the 10 episodes. In the online with pretraining setting, we evaluate the current policy over 100 episodes for every 10000 training steps due to the difficulty of the environments. The evaluation reward takes the average over the 100 episodes.

Algorithm implementation

We base our implementation of SAC off Kostrikov (2021). It uses one target critic, double critics, a single actor network and a single network for temperature adjustment for maximum entropy. We add an additional value network with extreme q-learning loss for ROER TD error estimation and priority calculation. We use the default hyperparameters and network architectures for the SAC algorithms for all of our experiments. The hyperparameters and the network architecture are shown in Table 4.

Parameter	Value
optimizer	Adam
learning rate	3×10^{-3}
actor, critic, and value network arch	(256, 256)
non-linearity	ReLU
value network noise	0.1
batch size	256
buffer size	1,000,000 , 2,000,000(antmaze)
discount	0.99
target smoothing coefficient	5×10^{-3}
gradient penalty coefficient	1

Table 4: Hyperparameters of SAC and network architecture.

We adapt the code in Lahire et al. (2021) into JAX implementation for LaBER. We adapt the proposed method in Fujimoto et al. (2020) for PER implementation and uses the loss adjusted version of PER as it shows similar or even better performance than the original implementation. Loss adjusted PER (LAP) uses Huber loss for critic objective which has the following form (Huber, 1992)

$$\mathcal{L}_{\text{Huber}}(\delta(i)) = \begin{cases} 0.5\delta(i)^2 & \text{if } |\delta(i)| \leq k \\ k(|\delta(i)| - 0.5k) & \text{otherwise} \end{cases} \tag{34}$$

where k is the bound for the Huber loss transformation and the default value is 1. We also uses Huber loss for our proposed ROER in the same idea as LAP. For SAC with uniform experience replay, we keep the original loss form which uses mean square loss. To stabilize the performance of all algorithms used in this study, we apply an additional gradient penalty to the critic loss inspired by Petzka et al. (2017) which penalizes gradient with norm great than 1. The penalty κ has the following form

$$\kappa = \max(\|\nabla Q_\theta - 1\|, 0)^2 \tag{35}$$

We note ROER uses exponential function in the formulation and it is sensitive to outliers. Thus, we lower clip the immediate weight with value 1 and use batch mean normalization on the immediate weight $e^{\delta_Q/\beta}$ as following

$$e^{\delta_Q/\beta}_{\text{normalized}} = \frac{e^{\delta_Q/\beta}}{\bar{e}^{\delta_Q/\beta}} \tag{36}$$

where $\bar{e}^{\delta_Q/\beta}$ denotes the batch mean. To further stabilize the performance and prevent the outliers, we clip the immediate weight and add minimum clip on the final priority which are further discussed in the following section of ablation study and hyperparameter selections.

C.2 Hyper-parameter Selection

Online

For the value of parameter of PER, we use Fujimoto et al. (2020) as a reference for MuJoCo environment where they show that a weight scale $\alpha = 0.4$ works best for the set of tasks. As to DM Control, we search over the set $[0.1, 0.2, 0.4, 0.6, 0.8]$ for individual task. The final choice of the value for each task is shown in Table 5.

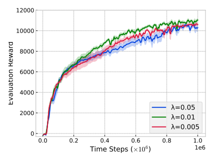
For the value of parameter large batch of LaBER, we search over the set $[768, 1024, 1280, 1536]$ for individual task in both MuJoCo and DM Control environment. The final choice of the value for each task is shown in Table 5.

For our proposed method ROER, we have 5 parameters, namely convergence rate (λ), gumbel loss clip (Grad Clip), loss temperature (β), immediate weight clip (Max Exp Clip), and minimum priority clip (Min Clip) that require tuning. However, we discover that only β and the clip range requires tuning for each specific environment while we can use a set of values for the rest. We use the set of value used in Garg et al. (2023) as a reference and search over the set $[0.005, 0.01, 0.05]$ for λ, $[5, 7, 10]$ for the Grad Clip, $[0.4, 1, 4]$ for β, $[25, 50, 100]$ for the Max Exp Clip, and $[1, 5, 10]$ for the Min Priority Clip. The final choice of the value for each task is shown in Table 5. We take HalfCheetaah-v2 from MuJoCo and Hopper-Stand from DM Control as examples to show the hyper-parammeter ablations for the online experiments. We show the effect of each parameter by fixing the rest as the default values. The default parameter values for HalfCheetah-v2 is $\lambda = 0.01, \text{Grad Clip} = 7, \beta = 4, \text{Max Exp Clip} = 50, \text{Min Clip} = 10$. The default parameter values for Hopper-Stand is $\lambda = 0.01, \text{Grad Clip} = 7, \beta = 1, \text{Max Exp Clip} = 100, \text{Min Clip} = 1$. The performance comparisons of the two task for varying each parameter are plotted in Fig. 3 and Fig. 4 respectively.

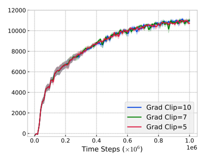
According to Fig. 3a and Fig. 4a, $\lambda = 0.01$ achieves the best performance for the two environments. We note that a too big λ can results in divergence as in Fig. 4a where $\lambda = 0.05$ achieves bad performance due to the too quick priority update which results in numerical instability. A small λ gives stable convergence but too small λ may slow down the convergence in some cases as in Fig. 4a. We discover that generally $\beta = 0.01$ works well across a wide domain and we use this value for all environments in our study.

			ROER			PER		LaBER
Env	λ	β	Grad Clip	Min Priority Clip	Max Exp Clip	α	Min Priority Clip	Large Batch
Ant-v2	0.01	1	7	10	100	0.4	1	1280
HalfCheetah-v2	0.01	4	7	10	50	0.4	1	1024
Hopper-v2	0.01	0.4	7	10	100	0.4	1	1536
Humanoid-v2	0.01	4	7	10	50	0.4	1	768
Walker2d-v2	0.01	4	7	10	50	0.4	1	1024
Fish-swim	0.01	1	7	1	100	0.4	1	768
Hopper-hop	0.01	1	7	1	50	0.6	1	1024
Hopper-stand	0.01	1	7	1	100	0.2	1	1280
Humanoid-run	0.01	4	7	1	100	0.4	1	1536
Humanoid-stand	0.01	4	7	1	100	0.4	1	1280
Quadruped-run	0.01	1	7	10	100	0.4	1	1024

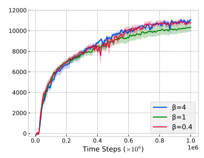
Table 5: Hyperparameters for ROER, PER and LaBER in online setting



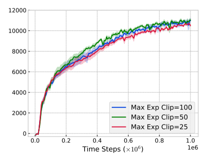
(a) λ



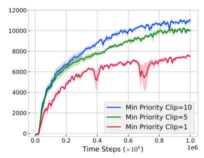
(b) Grad Clip



(c) β

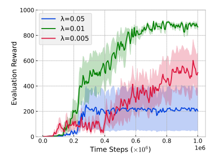


(d) Max Exp Clip

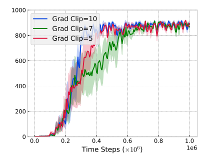


(e) Min Clip

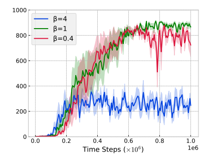
Figure 3: Convergence rate (λ), Gumbel loss clip (Grad Clip), loss temperature (β), Maximum exponential clip (Max Exp Clip), and minimum priority clip (Min Clip) Ablation for HalfCheetah-v2 over 5 random seeds. One parameter is changing while the rest are fixed. The default combination is [0.01, 7, 4, 50, 10] which is the set used in our final results. All curves are smoothed with Savitzky–Golay filter for visual clarity. The shaded region represents standard error which is favored in this case to separate the curves.



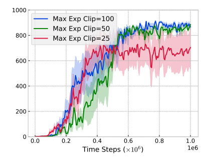
(a) λ



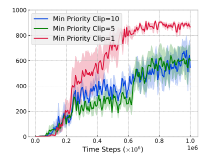
(b) Grad Clip



(c) β



(d) Max Exp Clip



(e) Min Clip

Figure 4: Convergence rate (λ), Gumbel loss clip (Grad Clip), loss temperature (β), Maximum exponential clip (Max Exp Clip), and minimum priority clip (Min Clip) Ablation for Hopper-stand over 5 random seeds. One parameter is changing while the rest are fixed. The default combination is [0.01, 7, 1, 10, 1] which is the set used in our final results. All curves are smoothed with Savitzky–Golay filter for visual clarity. The shaded region represents standard error which is favored in this case to separate the curves.

Grad Clip is initially leveraged to prevent the outliers in gumbel loss of extreme q-learning (Garg et al., 2023). In our case, we found that varying its value has negligible effect on the final result. As shown in Fig. 3b and Fig. 4b, all three values give similar performance. We use 7 as the value for all environments in our study.

The loss temperature β controls the strength of the penalization from the KL regularizer on the distributions between the on-policy data and off-policy data in replay buffer. It also scales the TD error and thus affect the value of the priority. Theoretically, a small β is beneficial for datasets with lots of random noisy actions and far from the on-policy distributions while a high β works well for datasets close to the on-policy distribution. In practice, we note that using a fixed value requires tuning for specific environments due to the different dynamics. $\beta = 4$ works well for most tasks in

MuJoCo while $\beta = 1$ works well for most tasks in DM Control. The affect of β is more obvious and easier to interpret in online with pretraining setting which is discussed in the following subsection. We recognize the difficulty of deciding the value for β and a solution to this can be using adaptive loss temperature. As the off-policy data in replay buffer converges to the on-policy distribution, we increase the value of β.

The Max Exp Clip and Min Priority Clip serve to prevent outliers and control the range of priority distribution. In most cases, a Max Exp Clip of value 50 or 100 works well. Min Clip requires tuning for each environment but we find Min Priority Clip = 10 works well for all MuJoCo tasks and Min Priority Clip = 1 works well for most of DM Control tasks except quadruped-run which uses value 10. A lower Max Exp Clip and a higher Min Priority Clip reduce the range of distribution and tend to stabilize the performance but may give sub-optimal performance. In addition, a higher Min Priority Clip can also prevent experience forgetting as the downweighted samples do not differ too much from the rest.

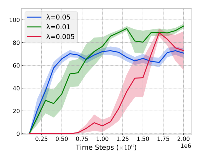
Online with Pretraining

To select the appropriate value of α for PER, we searched over $[0.1, 0.2, 0.4, 0.6, 0.8, 1]$. To select the appropriate value of large batch for LaBER, we search over $[768, 1024, 1280, 1536]$. The final choices for each environment are listed in Table 6.

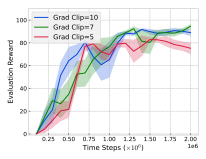
To select the appropriate parameter values for ROER, we searched over the same sets of values for λ, Grad Clip, Max Exp Clip, and Min Priority Clip as in the online setting. For the loss temperature β, we consider a slightly larger set $[0.4, 0.8, 2, 4]$ to better demonstrate the effect of β. Similar to the online setting, we also found that a set of values work well across different environments and the final choices are shown in Table 6. We take Antmaze-Umaze with pretraining using antmaze-umaze-diverse-v2 dataset as an example to ablate the hyperparameters. While varying the value of one parameter, we fix the rest as the default values. The default parameter values for antmaze-umaze-diverse-v2 is $\lambda = 0.01, \text{Grad Clip} = 7, \beta = 0.4, \text{Max Exp Clip} = 50, \text{Min Clip} = 1$.

	ROER					PER		LaBER
Env	λ	β	Grad Clip	Min Priority Clip	Max Exp Clip	α	Priority Clip	Large Batch
antmaze-umaze-v2	0.01	0.4	7	10	50	0.1	1	1024
antmaze-umaze-diverse-v2	0.01	0.4	7	1	50	0.4	1	1024
antmaze-mediumm-play-v2	0.01	0.4	7	1	50	0.4	1	1536
antmaze-medium-diverse-v2	0.01	0.4	7	1	50	0.4	1	1024

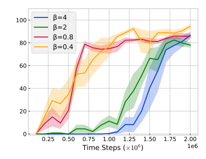
Table 6: Hyperparameters for ROER, PER, and LaBER in online with pretraining setting



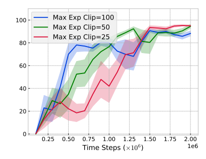
(a) λ



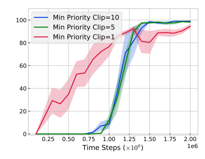
(b) Grad Clip



(c) β



(d) Max Exp Clip



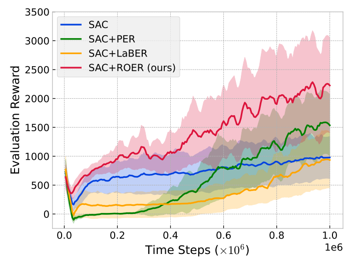
(e) Min Clip

Figure 5: Gumbel loss clip (Grad Clip), loss temperature (β), and minimum priority clip (Min Clip) Ablation for Antmaze with Antmaze-umaze-diverse-v2 dataset over 5 random seeds. One parameter is changing while the rest are fixed. The default combination is [0.01, 7, 0.4, 50, 1] which is the set used in our final results. The shaded region represents standard error which is favored in this case to separate the curves.

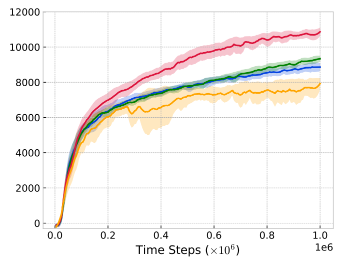
Similar to the online setting, $\lambda = 0.01$ shows good performance and the value of Grad Clip only slightly affect the final performance. We choose the values for the two parameters same as before. Varying β shows that a smaller value results in early performance improvement while bigger value only shows improvement after the on-policy data overweight in the replay buffer as demonstrated in Fig. 5c. This confirms our previous discussion that a small β is beneficial for datasets that differ from the on-policy distribution while a high β favors datasets close to the on-policy distribution. The behavior of varying Max Exp Clip and Min Priority Clip also corresponds to our previous discussion that a smaller range of priority distribution gives a stable but potentially sub-optimal performance while a wider range can benefit the agent to explore and potentially learn faster.

D Additional Results

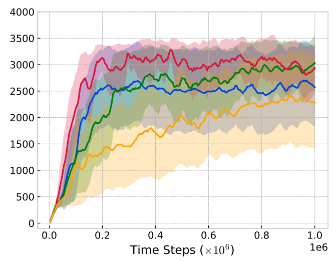
In this section, we present the additional results and discussions of our experiments. The learning curves in the online setting are shown in Fig. 6 for tasks in MuJoCo and Fig. 7 for tasks in DM Control. Besides the better performance, we note that our proposed ROER also shows faster improvement in Ant-v2, HalfCheetah-v2, Hopper-v2, Humanoid-v2, Fish-Swim and Hopper-Stand. This implies that ROER can obtain better data efficiency than the baselines. We additionally evaluate our proposed ROER in comparison to baselines in MuJoCo over 3 million steps to better illustrate the advantage of our proposed method as shown in Fig. 8. ROER shows consistently better performance as in the evaluation over 1 million steps and outperforms baselines in Ant-v2, HalfCheetah-v2, and Humanoid-v2 with very little or without overlapping shaded region. We did not include LaBER for this comparison due to the long training time it takes. We note that the performance of ROER gets worse in Hopper-v2 for longer steps as it reaches the reward saturation very early and the extra training can be harmful for the policy update due to over-fitting and additional updates on Q-functions with weights that causes loss explosion.



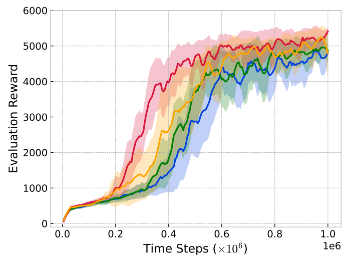
(a) Ant-v2



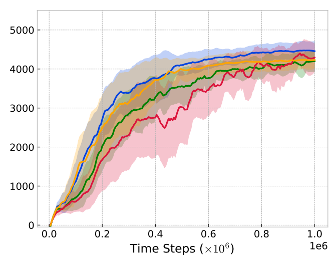
(b) HalfCheetah-v2



(c) Hopper-v2

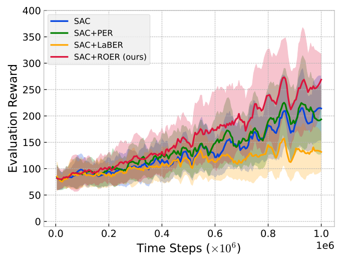


(d) Humanoid-v2

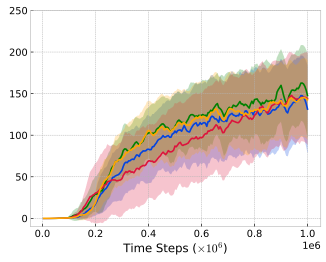


(e) Walker2d-v2

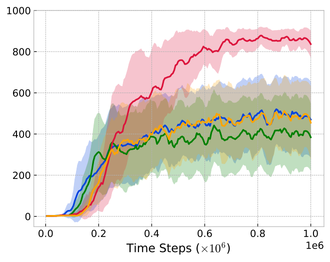
Figure 6: Learning curves for continuous control tasks in MuJoCo over 1 million steps. Curves are averaged over 20 random seeds, where the shaded area represents the 95% confidence interval of the average evaluation. All curves are smoothed with Savitzky–Golay filter for visual clarity.



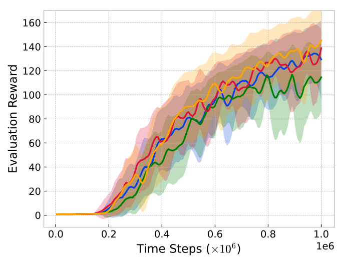
(a) Fish-Swim



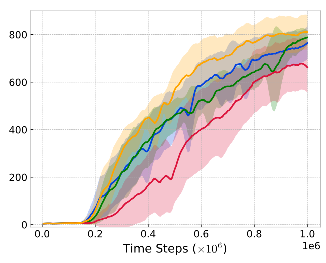
(b) Hopper-Hop



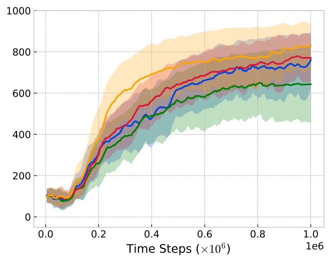
(c) Hopper-Stand



(d) Humanoid-Run

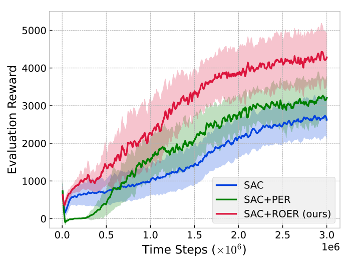


(e) Humanoid-Stand

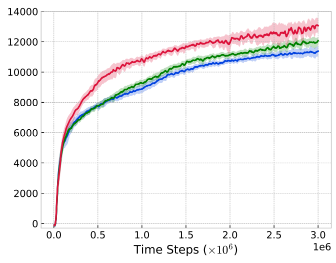


(f) Quadruped-Run

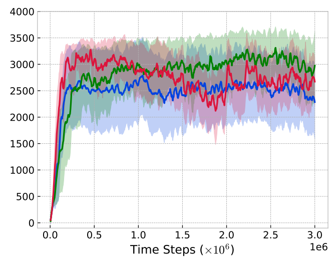
Figure 7: Learning curves for continuous control tasks in DM Control over 1 million steps. Curves are averaged over 20 random seeds, where the shaded area represents the 95% confidence interval of the average evaluation. All curves are smoothed with Savitzky–Golay filter for visual clarity.



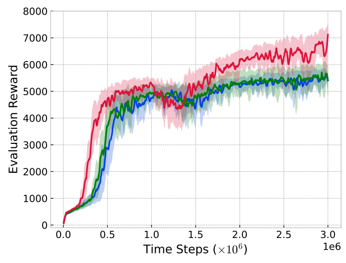
(a) Ant-v2



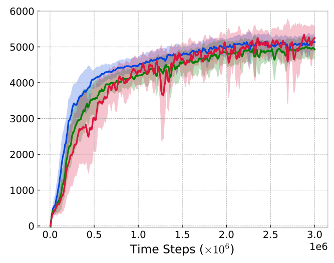
(b) HalfCheetah-v2



(c) Hopper-v2



(d) Humanoid-v2



(e) Walker2d-v2

Figure 8: Learning curves for continuous control tasks in MuJoCo over 3 million steps. Curves are averaged over 10 random seeds, where the shaded area represents the 95% confidence interval of the average evaluation. All curves are smoothed with Savitzky–Golay filter for visual clarity.