Information Sciences 675 (2024) 120736

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Controlling estimation error in reinforcement learning via Reinforced Operation

Yujia Zhang, Lin Li, Wei Wei *, Xiu You, Jiye Liang

Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, Shanxi, China

ARTICLE INFO

Keywords:
Reinforcement learning
Value estimation
Estimation bias
Estimation variance
Reinforced operation

ABSTRACT

In Value-based and Actor-Critic reinforcement learning (RL) methods, both inaccuracy and instability of value estimation will detrimentally affect their performance. Some typical RL methods, such as Maxmin Q-learning and QMD3, are plagued by the underestimation problem while failing to trade off the estimation bias and variance jointly. We propose the Reinforced Operation (RO) to address these shortcomings, which selects the closest median among multiple Q-function. RO is applicable to any model-free RL method. Theoretically, we introduce the Mean Square Error (MSE) to jointly analyze the estimation bias and variance of value estimation methods. We also demonstrate the superiority of RO in MSE reduction and give an upper bound for the estimation bias of value estimation methods with arbitrary distribution to guide the calculation of the estimation bias of value estimation methods. Based on RO, we propose the variants of Q-learning and TD3, Reinforced Q-learning (RQ) and Reinforced Delayed Deep Deterministic policy gradient (RD3), respectively, to tackle different tasks. We empirically demonstrate that our method can reduce estimation error and achieve superior performance on discrete and continuous benchmark tasks.

1. Introduction

One of the cornerstones of optimal policy acquisition in reinforcement learning (RL) is the accurate estimation of the state-action value function (Q-function). An inaccurate estimate of state-action value yields estimation bias, whether overestimated or underestimated, inducing the agent to produce wrong actions, slowing down training, and leading to inferior results [1]. Also, the inaccurate estimation can affect the extension of RL to robot [2,3], optimal control [4,5], and other domains. Q-learning [6] is widespread for discrete tasks, owing to its simple and readily accepted way of updating the Q-function using the highest state-action value the agent believes it can gain from subsequent state. Unfortunately, this maximization operation leads to overestimation bias, especially when using function approximators. In practice, most real-world problems involve high-dimensional inputs, which demand expressive and flexible nonlinear function approximators [7]. By employing nonlinear function approximators, the initial workload of RL is significantly decreased. Deep Q-Network (DQN) [8,9] performs well in most arcade learning environments (ALE) due to the incorporation of a powerful nonlinear approximation technique Deep Neural Network [10,11] on Q-learning. Due to inaccurate

* Corresponding author.
E-mail addresses: 342564535@qq.com (Y. Zhang), lilynn1116@sxu.edu.cn (L. Li), weiwei@sxu.edu.cn (W. Wei), youxiu@sxu.edu.cn (X. You), ljy@sxu.edu.cn (J. Liang).

https://doi.org/10.1016/j.ins.2024.120736
Received 25 April 2023; Received in revised form 29 December 2023; Accepted 8 May 2024

Available online 13 May 2024

function estimation or maximization operation, DQN also suffers from the overestimation mentioned above, which seriously affects the quality of the learned policy [12–14].

Recent studies have proposed diverse solutions to alleviate the overestimation issue in state-action value estimation. Techniques such as bias-correction and regularizing terms enhance accuracy [15,16]. Weighted Q-learning [17] and Softmax Q-learning [18] achieve more precise estimations using Gaussian approximation and the Softmax Bellman operator, respectively. Ensemble DQN and Averaged-DQN (ADQN) [19] address overestimation bias through averaging, reducing estimation variance. However, they fail to fully eliminate the overestimation bias, as the combination of overestimated Q-functions still has a positive bias, especially under the infinite action-value function setting. Double Q-learning [20] and Double DQN [21] address this problem with a decoupling operation, but this can lead to underestimation bias and is not suitable for Actor-Critic methods. Weighted Double Q-learning [22] uses two independent estimators for more accurate action value approximation, addressing both overestimation and underestimation biases. Despite these advancements, a gap remains in guiding agents to effectively determine or control estimation bias and variance, both key factors in estimation error.

Actor-Critic approaches suffer from the same overestimation issue, like DDPG [23], which inherently adopts the maximization operation. Following the DDPG work, the overestimation bias is reduced by additional modifications and extensions to the basic algorithm. Fujimoto et al. [24] introduce the minimization operation, alias the clipping operation, to reduce the overestimation bias by selecting the minimum from a pair of Q-function, and proposes the corresponding TD3 algorithm. Maxmin Q-learning (MQL) [25] and REDQ [26] also employ the minimization operation. However, the underestimation bias brought by the minimization operation also adversely affects the policy quality, and there is a lack of theoretical guarantee on how to jointly control the estimation bias and variance.

Some works aim to balance the underestimation and the overestimation bias. Jiang et al. [27] introduce a technique that involves selecting high-value action candidates to estimate the maximum expected action value, effectively mitigating underestimation bias in their approach. PRAG [28] enhances critic learning through the strategic periodic integration of action gradients, optimizing target values and navigating around local optima. GD3 [29] introduces an innovative generalized-activated weighting mechanism, leveraging non-decreasing functions, also known as activation functions, as weights to refine value estimation and substantially reduce bias. TADD [30] balances underestimation and overestimation biases via a weighting operation. QMQ and QMD3 [31] address underestimation bias and boost estimation stability through the quasi-median operation (QMO). DARC [32] brings innovation to RL by employing double actors to reduce estimation biases in DDPG and TD3, improving value estimation and exploration, and stabilizing value estimates via critic network regularization. Nevertheless, these techniques necessitate extra hyperparameters for control magnitude adjustment and may not be universally applicable. Persistent underestimation bias leads to significant estimation errors, compromising agent interaction efficiency.

A series of works on the Q-function aim to find the accurate state-action value, leading to an efficient Q-function update. Existing studies still suffer from a significant underestimation bias and do not provide a comprehensive joint analysis of estimation bias and estimation variance, both components of estimation error. Thus, this paper is motivated to develop an accurate estimation method and give a joint analysis of estimation bias and estimation variance. The following are summaries of the paper's contributions:

- We design the Reinforced Operation (RO), an operation applicable for any model-free RL method, which can reduce estimation error, while the state-action value obtained by RO is robust to outliers.
- Theoretically, we innovatively introduce the Mean Square Error (MSE) to integrally analyze the estimation bias and estimation variance of different operations. Further, we investigate the upper bound of estimation bias of value estimation methods without any distribution assumption.
- Based on the proposed RO, we construct Reinforced Q-learning (RQ) and Reinforced Delayed Deep Deterministic policy gradient (RD3) to cope with discrete and continuous action tasks, respectively.
- We evaluate our algorithm on six toy discrete action tasks and eight MuJoCo continuous action tasks across ten random seeds. Extensive experiments demonstrate that our methods outperform the most advanced ones.

2. Preliminaries

We formalize the usual RL framework as a Markov Decision Process (MDP), $(\mathcal{S}, \mathcal{A}, p, p_0, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action space, p: $\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ is the transition distribution, along with p_0, representing the initial state distribution, $r:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$ is the reward, and $\gamma\in[0,1]$ is the discount factor. At time t, with a given $s_t\in\mathcal{S}$, the agent takes $a_t\in\mathcal{A}$ and then transitions to the s_{t+1} according to its policy $\pi_\phi(s_t|a_t)$ with parameter ϕ, receiving r_t. Finding an optimal policy π_ϕ that maximizes $R_t=\sum_{t'=t}^{T}\gamma^{t'-t}r_{t'}$ starting from s_0 is the eventual purpose of RL.

Given s_0, a_0 and π, the critic or Q-function is

$$Q^\pi(s,a)=\mathbb{E}^\pi\left[\sum_{t=0}^{T}\gamma^t r_t|s=s_0,a=a_0\right].$$

The optimal $Q^\pi(s,a)$ is $Q^*(s,a)=\max_\pi Q^\pi(s,a)$, and the optimal policy π is $\pi^*(s)\in\operatorname{argmax}_a Q^*(s,a)$.

2.1. Value-based methods

For the Value-based methods, such as Q-learning and MQL, the temporal difference [33] can be applied to learn $Q^{\pi}(s,a)$. And the Bellman equation [34] establishes a recursive connection between (s,a) and (s',a'):

$$Q^{\pi}(s,a) = r + \gamma \mathbb{E}_{s',a'}\left[Q^{\pi}(s',a')\right], a' \sim \pi(s').$$

The loss function of a differentiated function approximator with parameter θ is

$$L(\theta) = \mathbb{E}_{s,a,r,s'\sim\mathcal{B}}\left[(y - Q_{\theta}(s,a))^2\right], \tag{1}$$

where $y^{QL} = r + \gamma \max_{a'\in\mathcal{A}} Q_{\theta}(s',a')$ and $y^{MQL} = r + \gamma \max_{a'\in\mathcal{A}} Q^{min}_{\theta}(s',a')$ in Q-learning and MQL, respectively, Q^{min}_{θ} indicates the selection of the minimum value from $Q_{\theta_i}(s,a)$ for $i = 1,\cdots,n$. And $\mathcal{B}$ is the replay buffer.

2.2. Actor-Critic methods

The Value-based RL methods will no longer work for the continuous control settings, while the Actor-Critic RL methods can handle these tasks. DDPG [23], as a typical Actor-Critic RL method, achieves relatively fast learning of sub-optimal policy owing to its deterministic policy gradient [35]. DDPG uses a deep neural network parameterized by ϕ, acting as an output policy actor, and the update rule is

$$\nabla_{\phi} J(\phi) = \mathbb{E}_{s\sim p_{\pi}}\left[\nabla_a Q^{\pi}_{\theta}(s,a)|_{a=\pi_{\phi}(s)} \nabla_{\phi}\pi_{\phi}(s)\right].$$

θ is updated the same as Equation (1), with different that $a = \pi_{\phi}(s)$ and $y^{DDPG} = r + \gamma Q'_{\theta'}(s', \pi_{\phi'}(s'))$, where θ' and ϕ' denote the parameter of the target networks.

TD3 [24] is a variant of DDPG that utilizes two critics with parameters θ_1 and θ_2, respectively. The actor parameter ϕ of TD3 is updated by

$$\nabla_{\phi} J(\phi) = \mathbb{E}_{s\sim p_{\pi}}\left[\nabla_a Q^{\pi}_{\theta_1}(s,a)|_{a=\pi_{\phi}(s)} \nabla_{\phi}\pi_{\phi}(s)\right]. \tag{2}$$

For $i = 1,2$, θ_i is updated in the following way:

$$L(\theta_i) = \mathbb{E}_{s,a,r,s'\sim\mathcal{B}}\left[(y - Q_{\theta_i}(s, \pi_{\phi}(s)))^2\right], \tag{3}$$

where $y^{TD3} = r + \gamma \min_{i=1,2} Q'_{\theta'_i}(s', \pi_{\phi'})$, and $\min_{i=1,2} Q'_{\theta_i}$ represents the minimization operation, also known as clipping operation.

TADD [30] alleviates the underestimation bias of TD3 by introducing the hyperparameter β:

$$y^{TADD} = r + \gamma(\beta \min_{i=1,2} Q'_{\theta'_i}(s', \pi_{\phi'}) + (1-\beta)Q''),$$

where $Q'' = \frac{1}{2}(Q'_{\theta'_3}(s', \pi_{\phi'}) + Q'_{\theta'_4}(s', \pi_{\phi'}))$ denotes the averaging operation.

QMD3 [31] updates the critic by selecting the state-action value which is smaller than as well as closest to the median. The target value of QMD3 is

$$y^{QMD3} = r + \gamma Q^{QMO}_{\theta'}(s', \pi_{\phi'}),$$

where $Q^{QMO}_{\theta'}(s', \pi_{\phi'})$ is the selection of $Q_{(\lfloor \frac{n}{2} \rfloor)}(n > 3)$ from given critics.

2.3. Estimation MSE in RL

Estimation bias is the deviation of the mathematical expectation of the estimated values from the true value Q^{true}. Suppose there are n estimates $\hat{Q}_i$, let $Z_i = \hat{Q}_i - Q^{true}$ be the individual estimation bias.

The estimation bias arises from two factors. One is the approximator, such as a neural network, is inaccurate. The other is the maximization or minimization operation. When there is no estimation bias, that is, $E[Z_i] = 0$, one should also be required that the estimation variance of state-action value is as small as possible. If the estimation bias exists, we should consider the estimation bias and the estimation variance jointly. Thus we introduce the MSE to measure the goodness of the estimated value. The MSE is

$$\begin{aligned} MSE &= E[\hat{Q}_i - Q^{true}]^2 \\ &= E[(\hat{Q}_i - E[\hat{Q}_i]) + (E[\hat{Q}_i] - Q^{true})]^2 \\ &= Var[\hat{Q}_i] + (E[Z_i])^2, \end{aligned} \tag{4}$$

where $Var[\hat{Q}_i]$ denotes the estimation variance, and $(E[Z_i])^2$ denotes the square of the estimation bias.

It is easy to see that MSE is a direct measure of estimation error that combines estimation bias and estimation variance. Unbiased and stable estimation of the state-action value helps reduce MSE.

Algorithm 1 RQ algorithm.

Initialize $Q_{\theta_1}, \cdots, Q_{\theta_n}$, and s_0
Initialize $\mathcal{B}$
while *interacting with the Environment* **do**
  $Q_\theta^{RO}(s,a) \leftarrow \underset{i=1,\cdots,n}{\mathrm{RO}} Q_{\theta_i}(s,a)$
  Select a based on $Q_\theta^{RO}(s,a)$ using ϵ-greedy
  Take a, and store (s,a,r,s') in $\mathcal{B}$
  for $t = 1$ to T **do**
    Sample a mini-batch (s,a,r,s') from $\mathcal{B}$
    $y \leftarrow r + \gamma \max_{a' \in \mathcal{A}} Q_\theta^{RO}(s',a')$
    $Q_i(s,a) \leftarrow Q_i(s,a) + \alpha[y - Q_i(s,a)]$
  end for
  $s \leftarrow s'$
end while

3. The proposed method

In this section, we first design an operation that can be applied to any model-free RL method, namely Reinforced Operation (RO), which can effectively alleviate the underestimation problem. Then, to handle different types of tasks, based on the RO, we propose novel modified versions of DQN and TD3, Reinforced Q-learning (RQ) and Reinforced Delayed Deep Deterministic policy gradient (RD3), respectively. In addition, we innovatively introduce MSE to analyze the estimation bias and estimation variance synthetically. We also theoretically demonstrate the superiority of RQ and RD3 in controlling the estimation error. Further, we theoretically investigate the upper bound of estimation bias of value estimation methods with the arbitrary distribution.

3.1. Reinforced Operation

Typically, the averaging or weighting operation is applied to several state-action values to produce a more precise target value, but the overestimation issue persists [24,25]. The minimization operation is proposed to counteract the overestimation problem, but this operation leads to the underestimation problem and is vulnerable to outliers [31]. The QMO operation is outlier independent and further alleviates the underestimation, but it still suffers from the underestimation problem and underutilizes the multiple Q-function. To address the limitations of the above methods, we endeavor to design an estimation method that has low MSE, insensitive to outliers, and is capable of making full use of multiple Q-function. Then, we propose a new method called Reinforced Operation (RO).

Suppose that $Q_1, Q_2, \cdots, Q_n$ are $n (n > 1)$ state-action values. The corresponding order state-action values are the $Q_{(i)}$ ranked in non-decreasing order. The smallest state-action value is $Q_{(1)}$, the second smallest is $Q_{(2)}$, $\cdots$, and, finally, the largest is denoted by $Q_{(n)}$. The target value obtained by RO is

$$Q^{RO} = \begin{cases} (Q_{(\lfloor \frac{n}{2} \rfloor)} + Q_{(\lceil \frac{n}{2} \rceil)})/2, & n = 3,5,7,\cdots \\ Q_{(\frac{n}{2})}, & n = 2,4,6,\cdots. \end{cases}$$

And The RO critic is defined as

$$Q^{RO}(s,a) = \underset{i=1,\cdots,n}{\mathrm{RO}} Q_{\theta_i}(s,a),$$

where $\underset{i=1,\cdots,n}{\mathrm{RO}}$ means taking RO on $Q_{\theta_i}(s,a)$.

In particular, when $n = 2$, RO degenerates to minimization operation, which means choosing $Q_{(1)}$. And when $n = 4,6,8,\cdots$, RO degenerates to QMO, all of which are the optimal target value calculation methods under the current n. Although QMO and RO share similarities, QMO can easily cause significant underestimation bias when $n = 3,5,7,\cdots$, which in turn causes large estimation error. Moreover, QMO only selects one Q-value, which underutilizes the Q-function. In contrast, RO can be applied to any number of Q-function and can reduce estimation error. Compared to other methods, RO utilizes the ranking information of all Q_i and cherry-picks quasi-median and median that are interference-free from outliers. And RO can be applied to any standard model-free RL method. In Section 3.3, we will analyze the effectiveness of RO in reducing estimation error.

To exploit the RO specifically, we develop the Reinforced Q-learning (RQ) and the Reinforced Delayed Deep Deterministic policy gradient (RD3) to handle different types of tasks based on Q-learning and TD3, respectively, as shown in Algorithm 1 and 2. In our experiments, RQ straightforwardly uses the tricks of DQN [9] to better match DQN. We described the update process of RD3 in Fig. 1 to explain the generality of our method. Moreover, to fully understand the difference between the method in this paper and other methods, we visualize several forward propagation processes of the Value-based methods MQL and RQ, and the Actor-Critic methods TADD and RD3 in Fig. 2.

3.2. Convergence of Reinforced Q-learning

This section demonstrates that RQ can converge to $Q^*(s,a)$ in the finite MDP setting, as Theorem 1. Suppose there are five value estimates Q^A, Q^B, Q^C, Q^D, Q^E if $n = 5$. We set the optimal action a^* of next state is

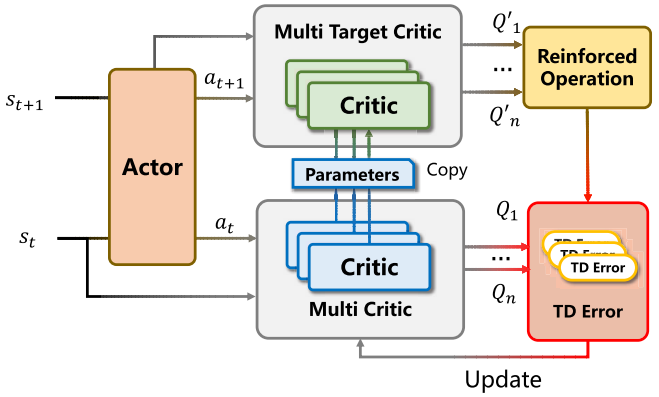


Fig. 1. The structure of the RD3. The target value is selected by the RO, and then multiple TD Errors are calculated with the state-action values output by multiple critics. Every critic is updated by the corresponding TD Error.

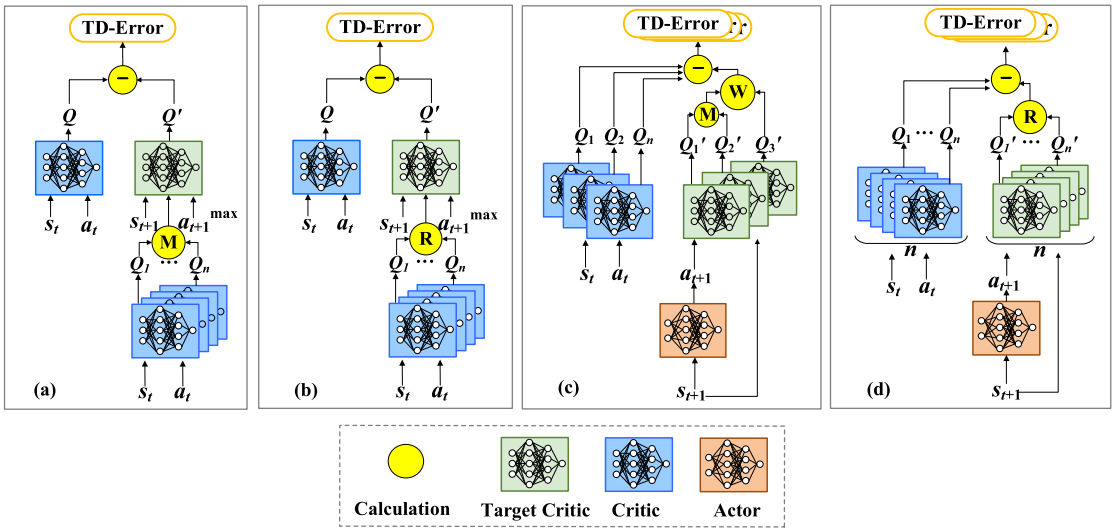


Fig. 2. Diagram of forward propagation processes of MQL (a), RQ (b), TADD (c), and RD3 (d). "-", "W", "M", "R" denote the subtraction, weighting, minimization and reinforced operation, respectively.

Algorithm 2 RD3 algorithm.

Initialize $Q_{\theta_1}, \cdots, Q_{\theta_n}$, π_ϕ, $\theta'_1 \leftarrow \theta_1, \cdots, \theta'_n \leftarrow \theta_n$, and $\phi' \leftarrow \phi$
Initialize $\mathcal{B}$
for $t = 1$ to T **do**
  Select $a \sim \pi_\phi(s) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$, store (s, a, r, s') in $\mathcal{B}$, and sample a mini-batch (s, a, r, s') from $\mathcal{B}$
  $a' \leftarrow \pi_{\phi'}(s')$+clip$(\epsilon, -c, c), \epsilon \sim \mathcal{N}(0, \sigma)$
  $y \leftarrow r + \gamma\, Q^{RO}_{\theta'}(s', \pi_{\phi'}(s'))$
  Update θ_i by Equation (3)
  if t mod d **then**
    Update ϕ by Equation (2)
    $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$, $\phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i$
  end if
end for

$$a^* = \underset{a}{\text{argmax}} Q^A(s', a).$$

At each time step, execute optimal action a^* at next state, then we have

$$y = r + \gamma \text{RO}(Q^A, Q^B, Q^C, Q^D, Q^E).$$

The update formulas of Q^A, Q^B, Q^C, Q^D, Q^E are as follows:

$$Q^A(s, a) = Q^A(s, a) + \alpha_t(y - Q^A(s, a)),$$

$$Q^B(s,a) = Q^B(s,a) + \alpha_t(y - Q^B(s,a)),$$
$$Q^C(s,a) = Q^C(s,a) + \alpha_t(y - Q^C(s,a)),$$
$$Q^D(s,a) = Q^D(s,a) + \alpha_t(y - Q^D(s,a)),$$
$$Q^E(s,a) = Q^E(s,a) + \alpha_t(y - Q^E(s,a)),$$

where α_t denotes the learning rate. Next, we give Lemma 1 to help us prove that RQ can converge to $Q^*(s,a)$ under the updating method above.

Lemma 1. *Consider a stochastic process* $(\zeta_t, \triangle_t, F_t)(t = 0, 1, 2, \cdots)$*, where* $\zeta_t, \triangle_t, F_t : X \to \mathbb{R}$ *satisfy the equation*:

$$\triangle_{t+1}(x_t) = (1 - \zeta_t(x_t))\triangle_t(x_t) + \zeta_t(x_t)F_t(x_t).$$

Let P_t *be a sequence of increasing* σ*-fields such that* ζ_0 *and* $\triangle_0$ *are* P_0*-measurable, and* $\zeta_t, \triangle_t$*, and* F_{t-1} *are* P_t*-measurable. Assume that the following hold*:

1. *The set* X *is finite.*
2. $\zeta_t(x_t) \in [0,1], \sum_t \zeta_t(x_t) = \infty, \sum_t(\zeta_t(x_t))^2 < \infty$ *with probability* 1 *and* $\zeta(x) = 0, \forall x \neq x_t$.
3. $\|E[F_t|P_t]\| \leq \kappa \|\triangle_t\| + c_t$*, where* $\kappa \in [0,1)$*, and* c_t *converges to* 0 *with probability* 1.
4. $Var[F_t(x_t)|P_t] \leq K(1 + \kappa\|\triangle_t\|)^2$*, where* K *is a constant,* $\|\cdot\|$ *represents the maximum norm.*

Then $\triangle_t$ *can converge to* 0 *with probability* 1.

Theorem 1. *Given the following conditions:*

1. *With the finite setting of MDP, each state-action pair is sampled infinitely from the lookup table.*
2. $\gamma \in [0,1)$.
3. *Both* Q^A, Q^B, Q^C, Q^D *and* Q^E *receive an infinite number of updates.*
4. $\alpha \in [0,1], \sum_t \alpha_t(s,a) = \infty, \sum_t(\alpha_t(s,a))^2 \leq \infty$ *with probability* 1*, and* $\alpha_t(s,a) = 0, \forall(s,a) \neq (s_t,a_t)$.
5. $Var[r(s,a)] \leq \infty, \forall s, a$.

Then, RQ will converge to Q^* *with probability 1.*

Proof. Set P_t is

$$\{Q_0^A, Q_0^B, Q_0^C, Q_0^D, Q_0^E, s_0, a_0, \alpha_0, r_1, s_1, \cdots, s_t, a_t\},$$

and $X = S \times A, \triangle_t = Q_t^A - Q^*, \zeta_t = \alpha_t$. Note that conditions 1 and 4 of the Lemma 1 hold by conditions 1 and 5 of the Theorem 1, respectively. Condition 2 of the Lemma 1 holds by the condition 4 of the Theorem 1 along with our selection of $\zeta_t = \alpha_t$. For simplicity, we set $Q_t^A = Q_t^A(s_{t+1}, a^*)$, and the other Q-functions are similar. Execute optimal action a^* at next state, we have:

$$\begin{aligned}\triangle_{t+1} &= (1 - \alpha_t)(Q_t^A(s_t,a_t) - Q^*(s_t,a_t)) \\ &\quad + \alpha_t(r_t + \gamma \mathrm{RO}(Q_t^A, Q_t^B, Q_t^C, Q_t^D, Q_t^E) - Q^*(s_t,a_t)) \\ &= (1 - \alpha_t)\triangle_t(s_t,a_t) + \alpha_t F_t,\end{aligned}$$

where $a^* = \underset{a}{\mathrm{argmax}} Q^A(s_{t+1}, a)$. And F_t is defined as

$$\begin{aligned}F_t &= r_t + \gamma \mathrm{RO}(Q_t^A, Q_t^B, Q_t^C, Q_t^D, Q_t^E) - Q^*(s_t,a_t) \\ &= r_t + \gamma \mathrm{RO}(Q_t^A, Q_t^B, Q_t^C, Q_t^D, Q_t^E) - Q^*(s_t,a_t) \\ &\quad + \gamma Q_t^A(s_{t+1}, a^*) - \gamma Q_t^A(s_{t+1}, a^*) \\ &= F_t^Q + c_t,\end{aligned}$$

where $c_t = \gamma \mathrm{RO}(Q_t^A, Q_t^B, Q_t^C, Q_t^D, Q_t^E) - \gamma Q_t^A(s_{t+1}, a^*)$, and $F_t^Q = r_t + \gamma Q_t^A(s_{t+1}, a^*) - Q^*(s_t, a_t)$. It is clear that $E[F_t^Q|P_t] \leq \gamma\|\triangle_t\|$, then condition 3 of the Lemma 1 holds if c_t converges to 0 with probability 1.

Let $y = r_t + \gamma \mathrm{RO}(Q_t^A, Q_t^B, Q_t^C, Q_t^D, Q_t^E)$, and let $\triangle_t^{BA} = Q_t^B(s_t,a_t) - Q_t^A(s_t,a_t)$, $\triangle_t^{CA} = Q_t^C(s_t,a_t) - Q_t^A(s_t,a_t)$, $\triangle_t^{DA} = Q_t^D(s_t,a_t) - Q_t^A(s_t,a_t)$, $\triangle_t^{EA} = Q_t^E(s_t,a_t) - Q_t^A(s_t,a_t)$, then c_t converges to 0 if $\triangle_t^{BA}$, $\triangle_t^{CA}$, $\triangle_t^{DA}$, and $\triangle_t^{EA}$ converge to 0. The update of $\triangle_t^{BA}$ is the sum of Q^A and Q^B at time t:

$$\triangle_{t+1}^{BA} = \triangle_t^{BA} + \alpha_t(y - Q_t^B(s_t,a_t) - (y - Q_t^A(s_t,a_t))) = (1 - \alpha_t)\triangle_t^{BA}.$$

The update of $\triangle_t^{CA}$ is the sum of Q^A and Q^C at time t:

$$\triangle_{t+1}^{CA} = (1-\alpha_t)\triangle_t^{CA}.$$

$\triangle_t^{DA}$ and $\triangle_t^{EA}$ are updated in a similar way:

$$\triangle_{t+1}^{DA} = (1-\alpha_t)\triangle_t^{DA},$$
$$\triangle_{t+1}^{EA} = (1-\alpha_t)\triangle_t^{EA}.$$

Clearly, $\triangle_t^{BA}, \triangle_t^{CA}, \triangle_t^{DA}$, and $\triangle_t^{EA}$ will converge to 0, which means that we satisfy condition 3 of the Lemma 1. So $Q_t^A(s_t,a_t)$ will converge to $Q_t^*(s_t,a_t)$. Let $\triangle_t = Q_t^B - Q^*, \triangle_t = Q_t^C - Q^*, \triangle_t = Q_t^D - Q^*$, and $\triangle_t = Q_t^E - Q^*$, we can derive $Q_t^B(s_t,a_t), Q_t^C(s_t,a_t), Q_t^D(s_t,a_t)$ can converge $Q_t^*(s_t,a_t)$. For other numbers of Q-functions, we just repeat the above arguments. Thus, we prove the Theorem 1, then the RQ can converge to $Q^*(s,a)$. □

Next, we theoretically analyze the MSE of RO and other operations to demonstrate the superiority of RO.

3.3. Estimation MSE reduction via Reinforced Operation

In this section, we first investigate the underestimation issue resulting from using the existing operations, such as minimization operation [24–26], weighting operation [30], and quasi-median operation [31]. We show the negative impact of these operations and then justify the effectiveness of RO in reducing the MSE of estimated values.

Let $Q_{(i)}$ denote the order state-action value arranged in non-decreasing order for $Q_1, Q_2, \cdots, Q_n$ that are independently and identically distributed in $[\lambda-\mu, \lambda+\mu](\mu \gg \lambda > 0)$, λ is the overestimation bias of Q-function and μ denotes the range of positive bias [31]. $Z_{(k)} = \hat{Q}_{(k)} - Q^{true}$ is the k-th estimation bias. In our analysis of the MSE across various methods, we introduce Theorem 2. This theorem underscores the effectiveness of RO in reducing underestimation bias, a crucial aspect of estimation error.

Theorem 2. *Using $Z(\cdot)$ to represent the estimation bias of a given method. Under the default parameter settings as outlined in the original document, we present the following comparative analysis of estimation biases among various methods:*

$$Z(MQL) \le Z(TD3) = Z(REDQ) < Z(TADD) < Z(QMO)$$
$$\le Z(RO) \le 0 < Z(DDPG)$$

Proof. From [25,30,31], we have:

$$E[Z_{(k)}] = \frac{(2k-n-1)\mu}{n+1} + \lambda. \tag{5}$$

The estimation bias, marked by λ for both median and averaging operations, suggests an overestimation problem, adversely affecting the agent's policy learning capabilities. Consequently, we establish that $Z(DDPG) = \lambda$.

Methods such as MQL, TD3, and REDQ primarily use a minimization strategy by choosing the smallest state-action value. Their estimation bias, quantified as $-\frac{1}{3}\mu + \lambda < 0$ for $n = 2, k = 1$, indicates a tendency towards underestimation. MQL increases the number of critics, and as the number of critics grows, $Z(MQL) = \frac{1-n}{n+1}\mu + \lambda < 0$, which is evidently a monotonically decreasing function. The underestimation in MQL can worsen with an increase in n. Even at the minimum number of critics ($n = 2$), there's a significant underestimation bias.

TADD, by adding an averaging operation to the minimization process, modifies the estimation bias to $-\frac{19}{60}\mu + \frac{1}{20}\lambda < 0$, somewhat mitigating the underestimation bias. For $n > 3$, the estimation bias of QMO is $\frac{n\lambda+\lambda-\mu}{n+1}$ for $n = 4, 6, 8, \cdots$, and it becomes $\frac{n\lambda+\lambda-2\mu}{n+1}$ for $n = 5, 7, 9, \cdots$. QMO still encounters the underestimation problem with smaller n. For larger n, QMO alleviates underestimation but slows down policy learning.

The estimation bias of RO is $\frac{n\lambda+\lambda-\mu}{n+1}$ regardless of whether n is odd or even. RO is more accurate than other methods in this regard. Based on the aforementioned analysis, we arrive at the following conclusion:

$$Z(MQL) \le Z(TD3) = Z(REDQ) < Z(TADD) < Z(QMO) \tag{6}$$
$$\le Z(RO) \le 0 < Z(DDPG) \quad \square$$

Relative to other approaches, RO exhibits a lesser degree of underestimation bias, a trend that holds irrespective of the value of n. Notably, as n escalates, $Z(RO)$ gradually approaches zero, reflecting a trend towards more precise estimations.

The magnitude of the estimation bias is related to the experimental setting. Therefore, we use the normalized bias Z_i/Q^{true} to eliminate the influence of the environment setting, and the Q^{true} denotes the discounted cumulative return according to the current policy computed by the Monte Carlo.

To verify the estimation accuracy of RO, we first design a bias random generation experiment to randomly generate 10,000 points in the $[\lambda-\mu, \lambda+\mu]$ interval and then employ averaging operation (ADQN), minimization operation (TD3, MQL, REDQ),

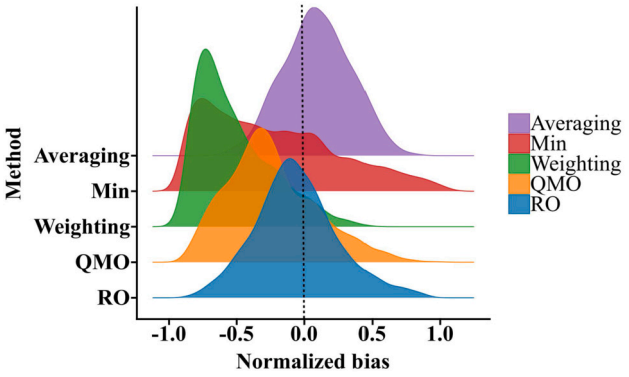
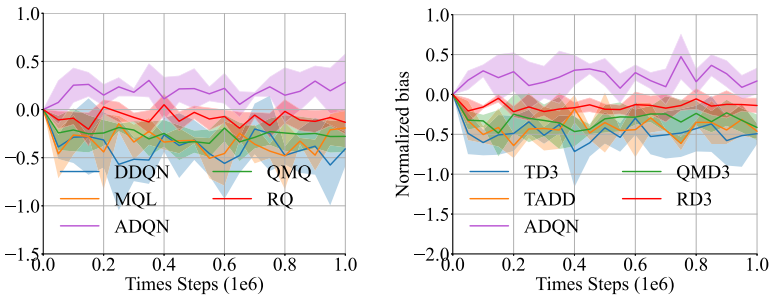


Fig. 3. Visualization of distribution changes of different methods. $\lambda = 0.1\mu > 0$. The closer the concentration of the distribution is to 0, the more accurate estimation is, and the flatter the curve is, the more unstable the estimation is.



(a) Space Invaders-v0 (b) Ant-v3

Fig. 4. Estimation bias curves on two different types of tasks. The bold lines depict the mean estimation bias over ten trials, and the darkened region stands for one standard deviation.

weighting operation (TADD), QMO (QMQ, QMD3) and RO (RQ, RD3) to obtain the corresponding normalized bias, as shown in Fig. 3. Specifically, the average operation has an overall left-skewed distribution with a large positive bias. The other four methods have an overall right-skewed distribution, in which the weighting and minimization operations have a larger negative bias, while QMO and RO have a smaller negative bias. It is noticed that the estimation bias of RO is the lowest, and the estimation is more stable than other methods. This also verifies the above analysis.

In Fig. 4, we measure the normalized bias of value estimation methods on the discrete action task Space Invaders-v0 and the continuous control task Ant-v3 when $n = 5$. It can be found that the average operation (ADQN) has a large positive bias, and the minimization operation (MQL, TD3), decoupling operation (DDQN) and QMO (QMQ, QMD3) have a negative bias. The weighting operation mitigates the negative bias of the minimization operation, but still suffers from a certain negative bias. Our method is able to gain more precise estimates than competing methods, and it has a smaller estimation variance than those methods, which indicates that RQ and RD3 are resistant to random initial conditions. It is worth mentioning that the test on the real environment in Fig. 4 is consistent with the results of the large sample experiment in Fig. 3, which further validates our previous analysis and the superiority of our methods in this paper.

Based on the above analysis, the RO operation is competitive with other methods regardless of the number of Q-functions. Further, the underestimation bias decreases as n increases, which may help to reduce the estimation MSE. Furthermore, we theoretically investigate the MSE of RO in Theorem 3 and then compare the MSE of value estimation methods to illustrate the superiority of our method.

Theorem 3. *When $n = 3, 5, 7, \cdots$, the MSE of RO is*

$$MSE(RO) = \frac{n^2 + 4n + 3}{(n+1)^2(n+2)}\mu^2 - \frac{2\mu\lambda}{n+1} + \lambda^2, \tag{7}$$

and when $n = 4, 6, 8, \cdots$, the MSE of RO changed to

$$MSE(RO) = \frac{n^2 + 3n + 2}{(n+1)^2(n+2)}\mu^2 - \frac{2\mu\lambda}{n+1} + \lambda^2. \tag{8}$$

Proof. When $n = 4, 6, 8, \cdots$, from [31], we know that

$$Var(RO) = \frac{n^2\mu^2 + 2n\mu^2}{(n+1)^2(n+2)}.$$

From Equation (4) and Equation (5), we have:

$$\begin{aligned} MSE(RO) &= \frac{n^2+2n}{(n+1)^2(n+2)}\mu^2 + (-\frac{1}{n+1}\mu + \lambda)^2 \\ &= \frac{n^2+3n+2}{(n+1)^2(n+2)}\mu^2 - \frac{2\mu\lambda}{n+1} + \lambda^2. \end{aligned}$$

When $n = 3, 5, 7, \cdots$, the state-action value obtained after RO consists of two parts, $Q_{(\lfloor \frac{n}{2} \rfloor)}$ and $Q_{(\lceil \frac{n}{2} \rceil)}$, which are not independent of each other. Then the estimation bias is

$$\frac{1}{2}(\hat{Q}_{(\lfloor \frac{n}{2} \rfloor)} + \hat{Q}_{(\lceil \frac{n}{2} \rceil)}) - Q^{true} = \frac{1}{2}Z_{(\lfloor \frac{n}{2} \rfloor)} + \frac{1}{2}Z_{(\lceil \frac{n}{2} \rceil)}.$$

And the estimation variance is

$$\frac{1}{4}Var(Z_{(\lfloor \frac{n}{2} \rfloor)}) + \frac{1}{4}Var(Z_{(\lceil \frac{n}{2} \rceil)}) + \frac{1}{2}Cov(Z_{(\lfloor \frac{n}{2} \rfloor)}, Z_{(\lceil \frac{n}{2} \rceil)}),$$

where $Cov(\cdot)$ is the covariance.

From [36], we know the probability distribution function (pdf) of $Z_{(k)}$ is

$$p_{(k)}(z_i) = \frac{n!p(z_i)}{(k-1)!(n-k)!}(F(z_i))^{k-1}(1 - F(z_i))^{n-k},$$

where $p(z_i)$ is pdf of z_i, $F(z_i)$ is the cumulative distribution function (cdf). Specifically:

$$p(z_i) = \begin{cases} \frac{1}{2\mu}, & z \in [\lambda - \mu, \lambda + \mu] \\ 0, & z \in else, \end{cases}$$

$$F(z_i) = \begin{cases} 0, & z_i \in (-\infty, \lambda - \mu] \\ \frac{z_i - \lambda + \mu}{2\mu}, & z_i \in (\lambda - \mu, \lambda + \mu] \\ 1, & z_i \in (\lambda + \mu, \infty). \end{cases}$$

It is difficult to directly compute the variance of $\frac{1}{2}Z_{(\lfloor \frac{n}{2} \rfloor)} + \frac{1}{2}Z_{(\lceil \frac{n}{2} \rceil)}$. To facilitate the calculation, we introduced a random variable $Y_i = \frac{Z_i - \lambda + \mu}{2\mu}$ which is independent of Z_i for $i = 1, \cdots, n$. It is easy to deduce that Y_i is identically uniformly distributed in $[0, 1]$ for $i = 1, \cdots, n$, the pdf and cdf of Y_i are as follows:

$$p(y_i) = \begin{cases} 1, & y_i \in [0, 1] \\ 0, & y_i \in else, \end{cases}$$

$$F(y_i) = \begin{cases} 0, & y_i \in (-\infty, 0] \\ y_i, & y_i \in (0, 1] \\ 1, & y_i \in (1, \infty). \end{cases}$$

From [31,36], we have:

$$E[Y_{(k)}] = \frac{k}{n+1}, Var[Y_{(k)}] = \frac{k(n-k+1)}{(n+1)^2(n+2)}, \tag{9}$$

and for $0 \le y_i \le y_j$, we have:

$$\begin{aligned} p_{Y(i),Y(j)}(y_i, y_j) = & m(i,j)[F(y_i)]^{i-1}[F(y_i) - F(y_j)]^{j-i-1} \cdot \\ & [1 - F(y_i)]^{n-j} f_Y(y_i) f_Y(y_j), \end{aligned}$$

where $m(i, j)$ denotes $\frac{n!}{(i-1)!(j-i-1)!(n-j)!}$. Then we have:

$$E[Y_{(i)}Y_{(j)}] = m(i,j) \int_0^1 y_i f_Y(y_i) \left[F(y_i)\right]^{i-1} dy_i \cdot$$

$$\int_{y_i}^1 y_j f_Y(y_j) \left[F(y_j) - F(y_i)\right]^{j-i-1} \left[1 - F(y_j)\right]^{n-j} dy_j$$

$$
\begin{aligned}
&= m(i,j)\int_0^1 y^j dy \int_y^1 x(x-y)^{j-i-1}(1-x)^{n-j}dx \\
&= m(i,j)\int_0^1 y^j dy \int_0^{1-y} (1-t)[(1-y)-t]^{j-i-1}t^{n-j}dt \\
&= m(i,j)\sum_{k=0}^{j-i-1}(-1)^{j-i-1-k}C_{j-i-1}^k \int_0^1 y^i(1-y)^k dy \int_0^{1-y}(1-t)t^{n-i-1-k}dt \\
&= m(i,j)\sum_{k=0}^{j-i-1}(-1)^{j-i-1-k}C_{j-i-1}^k \left[\frac{B(i+1,n-i+1)}{n-i-k} - \frac{B(i+1,n-i+2)}{n-i-k+1}\right] \\
&= \frac{i(n-i)!}{(n+2)(n+1)(j-i-1)!(n-j)!}(j+1)\sum_{k=0}^{j-i-1}\frac{(-1)^{j-i-1-k}C_{j-i-1}^k}{n-i-k}
\end{aligned}
$$

We note that:

$$\sum_{j=0}^{n}\frac{(-1)^j}{a+j}C_n^j = \frac{n!}{a(a+1)\cdots(a+n)}.$$

Then:

$$
\begin{aligned}
E\left[Y_{(i)}Y_{(j)}\right] &= \frac{i(j+1)}{(n+2)(n+1)}\frac{(n-i)!}{(j-i-1)!(n-j)!}\cdot \\
&\quad \frac{(j-i-1)!}{(n-i)(n-i-1)\cdots(n-j+1)} \\
&= \frac{i(j+1)}{(n+2)(n+1)}.
\end{aligned} \tag{10}
$$

Following Equation (9) and Equation (10), we have:

$$Cov(Y_{(i)},Y_{(j)}) = \frac{i(j+1)}{(n+2)(n+1)} - \frac{i}{n+1}\frac{j}{n+1} = \frac{i(n+1-j)}{(n+2)(n+1)^2}. \tag{11}$$

Based on Equation (9) and Equation (11), we have:

$$Var\left(Y_{(i)}+Y_{(j)}\right) = \frac{(j+2i)(n+1-j)+in+i-i^2}{(n+2)(n+1)^2}.$$

Then:

$$Var\left(Z_{(j)}+Z_{(i)}\right) = \frac{(j+2i)(n-j+1)+in+i-i^2}{(n+2)(n+1)^2}4\mu^2.$$

Then we can derive the estimation variance by adopting the RO:

$$Var\left(\frac{1}{2}Z_{(\lfloor\frac{n}{2}\rfloor)}+\frac{1}{2}Z_{(\lceil\frac{n}{2}\rceil)}\right) = \frac{n^2+3n+1}{(n+2)(n+1)^2}\mu^2.$$

This implies that when $n = 3,5,7,\cdots$,

$$MSE(RO) = \frac{n^2+4n+3}{(n+1)^2(n+2)}\mu^2 - \frac{2\mu\lambda}{n+1} + \lambda^2. \quad \square$$

Further, we analyze the MSE of value estimation methods in Theorem 4.

Theorem 4. *The comparison of MSE of value estimation methods is*

$$
\begin{aligned}
&MSE(RD3) \le MSE(QMD3) \le MSE(TADD) \le MSE(MQL) \\
&\qquad = MSE(TD3) = MSE(REDQ).
\end{aligned}
$$

Proof. According to Equation (5) and the proof of Theorem 3, we know that the MSE of minimization operation (MO) and QMO ($n = 5,7,9,\cdots$) are

$$MSE(MO) = \frac{n^3+n+2}{(n+1)^2(n+2)}\mu^2 + \frac{2-2n}{n+1}\mu\lambda + \lambda^2,$$
$$MSE(QMO) = \frac{n^2+6n+5}{(n+1)^2(n+2)}\mu^2 - \frac{4}{n+1}\mu\lambda + \lambda^2.$$

Since $\mu \gg \lambda > 0$, we can simplify the calculation by removing the term containing λ above. When $n = 4, 6, 8, \cdots$, it is obvious that $n^3+n+2-n^2-3n-2 > 0$, it implies that $MSE(RD3) = MSE(QMD3) \leq MSE(MQL) = MSE(TD3) = MSE(REDQ)$. When $n = 3, 5, 7, \cdots$, clearly $n^2+4n+3 \leq n^2+6n+5 \leq n^3+n+2$, then we have:

$$MSE(RD3) \leq MSE(QMD3) \leq MSE(TADD) \leq MSE(MQL)$$
$$= MSE(TD3) = MSE(REDQ). \quad \square$$

Given this, we can deduce that our approach yields an MSE that is lower than that of other approaches, irrespective of the number of Q-function. Although the estimation variance of our method is slightly higher than that of QMD3 at $n = 3, 5, 7, \cdots$, the estimation bias of our method is much lower, resulting in an overall lower MSE, which is the reason for the ultimate better performance of our method, and we will demonstrate it in the subsequent experiments. Further, we give Corollary 1 to show the relationship between the MSE of the RO and Q-function number n.

Corollary 1. *The MSE of RO decreases as n increases, and* $MSE(RO) = 0$ *when* $n \to +\infty$.

Proof. We also simplify the calculation by removing the term containing λ of Equation (7) and Equation (8). Then we have:

$$\lim_{n\to\infty} \frac{n^2+4n+3}{(n+1)^2(n+2)} = \lim_{n\to\infty} \frac{2}{6n+8} = 0,$$
$$\lim_{n\to\infty} \frac{n^2+3n+2}{(n+1)^2(n+2)} = \lim_{n\to\infty} \frac{2}{6n+8} = 0.$$

It is easy to see that $(n^2+4n+3)/[(n+1)^2(n+2)]$ and $(n^2+3n+2)/[(n+1)^2(n+2)]$ are both monotonically decreasing functions. As a result, the MSE of RO will get lower as n increases. $\square$

Moreover, in Theorem 5, we give an upper bound on the estimation bias of RO without any distribution assumption.

Theorem 5. *For an arbitrary distribution of estimation bias z with the same mean* ξ *and variance* σ^2*, respectively. When* $n = 3, 5, 7, \cdots$*, an upper bound on RO is*

$$\sup_{z\sim(\xi,\sigma^2)} E[RO] \leq \xi + \frac{\sigma}{2}\left(\sqrt{\frac{n-3}{n+3}} + \sqrt{\frac{n-1}{n+1}}\right),$$

when $n = 4, 6, 8, \cdots$*,*

$$\sup_{z\sim(\xi,\sigma^2)} E[RO] \leq \xi + \sigma\sqrt{\frac{n-2}{n+2}}.$$

Proof. We can derive from [36–38] that the $E[z_{(k)}]$ is upper bounded by:

$$\min_z \left(z + \frac{n}{2(n-k+1)}\left[\xi - z + \sqrt{(\xi-z)^2+\sigma^2}\right]\right), \tag{12}$$

where $z^* = \xi + \frac{(2k-n-2)\sigma}{2\sqrt{(k-1)(n-k+1)}}$, substituting z^* into Equation (12) yields:

$$\sup_{z\sim(\xi,\sigma^2)} E[z_{(k)}] \leq \xi + \sigma\sqrt{\frac{k-1}{n-k+1}}.$$

When $n = 3, 5, 7, \cdots$, RO select $(Q_{(\lfloor \frac{n}{2} \rfloor)} + Q_{(\lceil \frac{n}{2} \rceil)})/2$, then the upper bound of RO is

$$\sup_{z\sim(\xi,\sigma^2)} \frac{1}{2}\left(E[z_{(\lfloor \frac{n}{2} \rfloor)}] + E[z_{(\lceil \frac{n}{2} \rceil)}]\right) \leq \xi + \sigma\left(\sqrt{\frac{n-3}{n+3}} + \sqrt{\frac{n-1}{n+1}}\right),$$

and when $n = 4, 6, 8, \cdots$, it becomes:

$$\sup_{z\sim(\xi,\sigma^2)} (E[z_{(\lfloor \frac{n}{2} \rfloor)}] \leq \xi + \sigma\sqrt{\frac{n-2}{n+2}}. \quad \square$$

The results of Theorem 5 are distribution-free without explicit distribution assumptions, and the analysis of the upper bound on the estimation bias can guide to find operations that can effectively reduce the estimation bias. Based on the proof of Theorem 5, we can obtain the upper bound on the estimation bias of the value estimation methods. For an exact upper bound of RO, just simply bring the specific mean and variance. For example, if the bias z is independently and identically distributed in $[\lambda-\mu, \lambda+\mu]$, then the upper bound on the estimation bias of RO is $\lambda + \frac{\mu}{3}\sqrt{\frac{3(n-2)}{n+2}}$ when $n = 4, 6, 8, \cdots$.

4. Experiments

In this section, we empirically evaluate RQ and RD3 on extensive different types of tasks. We use ten random seeds in our tests to ensure that our comparisons are valid and trustworthy. And we plot the average return as bold lines and half standard deviation as the darkened region. All experiments are conducted on servers with Intel I9-10850K and NVIDIA RTX3070.

4.1. Discrete action tasks

To evaluate RQ, We select six tasks from Gym [39], PLE [40], and MinAtar [41] to evaluate RQ. We compare our RQ with QMQ [31], MQL [25], ADQN [19], DDQN [21] and DQN [8]. The hyper-parameters and settings of network are maintained in line with [25]. The number of critics for RQ, QMQ, and ADQN is 5, for MQL is 2, and for DQN is 1.

Fig. 5 (a)-(f) show the smoothed learning curves on six benchmark environments. We find that RQ outperforms the comparison algorithms in the final performance on all tasks, and RQ can achieve a favorable policy with fewer time steps. Moreover, the standard deviation (height of the shaded area) of RQ is lower than that of the comparison algorithms, indicating that the decrease in estimation MSE significantly impacts performance enhancement. Generally, the ranking of the final performance of different methods is matched with Theorem 4. DQN and ADQN find inferior policy due to the overestimation problem, QMQ, MQL, and DDQN have clear underestimation bias. RQ alleviates the underestimation problem of QMQ, MQL, therefore, the final policy is better than the other algorithms.

Further analysis involves contrasting the performance trajectories of RQ, QMQ, and MQL for $n = 5, 7, 9$, as illustrated in Fig. 5 (g) and (h). As n escalates, both RQ and QMQ exhibit improved final performances, corroborating Corollary 1's assertions. Notably, RQ demonstrates more pronounced gains, attributable to its marked reduction in MSE. While MQL may attain slightly enhanced final performance at higher n values, it notably hinders early learning phases, a trend consistent with findings from [31] and [25]. Theoretically, increased n values diminish estimation bias in RQ and QMQ but exacerbate MQL's underestimation bias, thus rationalizing MQL's delayed early learning with higher n values.

4.2. Continuous action tasks

For the continuous action tasks, we select eight MuJoCo tasks [42]. For convenience, we abbreviate InvertedPendulum and InvertedDoublePendulum as IP and IDP. We first compare our RD3 with QMD3 [31], TADD [30], TD3 [24], and DDPG [23]. The hyper-parameters and network settings of RD3 are precisely equivalent to those of QMD3, TD3, and TADD for all tasks. The number of critics n for RD3 and QMD3 is 5, for TADD is 4, for TD3 is 2, and for DDPG is 1.

Fig. 6 represents the smoothed learning curves on the eight MuJoCo tasks. Fig. 6 demonstrates that, compared to other methods, although RD3 has a slight underestimation bias that does not propagate through the gradient, RD3 can achieve comparable or superior performance on all continuous tasks while maintaining similar convergence speeds. In particular, for the challenging benchmarks, such as Hopper-v3 and Humanoid-v3, RD3 can achieve a significantly higher averaged return than the state-of-the-art (SOTA) method QMD3. Due to overestimation bias, DDPG performs poorly on most tasks, which aligns with our earlier findings. We also notice that TD3 and TADD do not work well on challenging tasks, such as Ant-v3. Furthermore, the standard deviation of RD3 and QMD3 is lower than that of the other methods.

In Fig. 7, the average return of RD3 is tested for $n = 5, 7, 10, 15$. RD3 consistently learns robust and superior policies for various values of n. Furthermore, an increase in n correlates with RD3 achieving progressively better policies, aligning with the insights of Corollary 1. Additionally, our method does not appreciably increase computational costs relative to QMD3 when utilizing an equivalent number of critics.

In Tables 1 and 2, RD3 is compared with advanced methodologies such as DARC, PRAG, and GD3, each significantly enhancing value estimation accuracy through the use of double actors, activation functions, and action gradients, respectively. This comparison also includes high performers QMD3 and TADD from Fig. 6. Specifically, hyperparameters such as the weighting coefficient in DARC, the index term in GD3, and the action gradient regularization parameter in PRAG are maintained consistently with their original specifications in the respective literature. The scores for each algorithm are normalized by dividing them by TD3 scores, with larger values denoting superior performance across tasks. For Reacher-v2, given that the final returns are negative, we express performance as the inverse ratio to TD3. Additionally, a comparison of computational time relative to TD3 for these methods is provided in Tables 1.

We observe that RD3 outperforms other techniques across most tasks in both average and optimal performance metrics. This marked advancement can be largely credited to RD3's enhanced capability in minimizing estimation errors. Moreover, compared to state-of-the-art methods like DARC, QMD3, and TADD, our approach significantly elevates TD3's performance with acceptable training expenses.

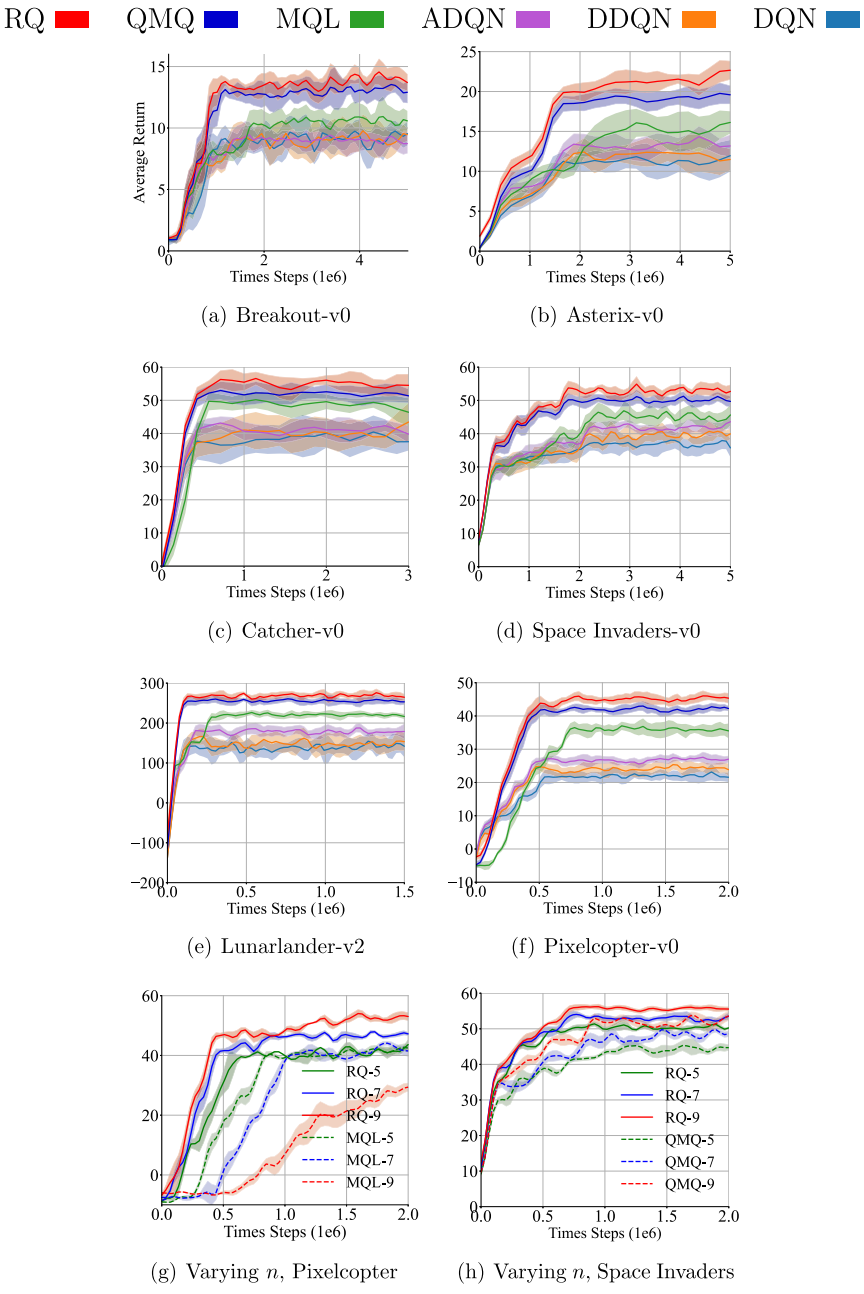


Fig. 5. Learning curves on discrete action tasks.

5. Conclusion

In this work, we present the Reinforced Operation, which can yield accurate and stable state-action value and be applied to any model-free RL method. Based on Reinforced Operation, we offer an extension of Q-learning and TD3, Reinforced Q-learning and Reinforced Delayed Deep Deterministic policy gradient to tackle discrete and continuous action tasks, respectively. In addition, we innovatively introduce the MSE to analyze the estimation error directly and jointly analyze the estimation bias and variance of value estimation methods. We theoretically demonstrate our method's advantage in reducing the estimation MSE. Furthermore, we give the upper bound of the estimation bias of value estimation methods with arbitrary distribution assumptions. We conduct a wide range of experiments on different types of tasks, and the results show that the proposed method significantly surpasses SOTA methods.

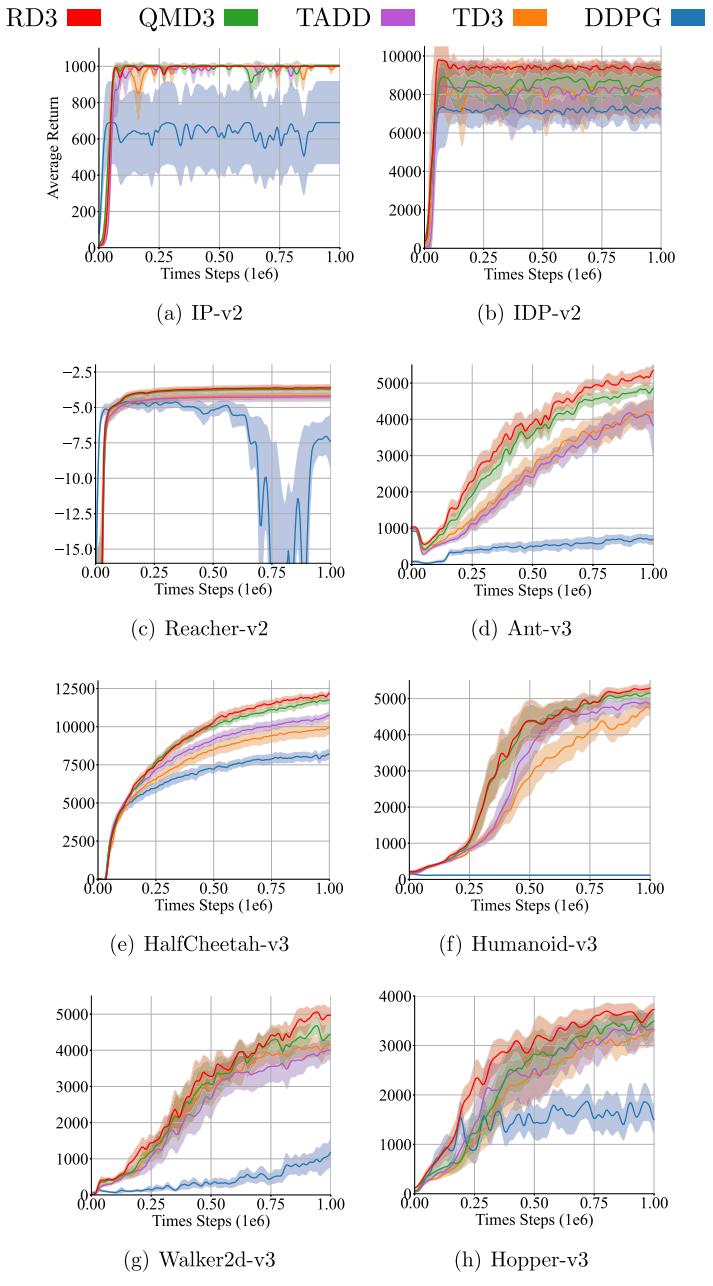


Fig. 6. Learning curves on continuous control tasks.

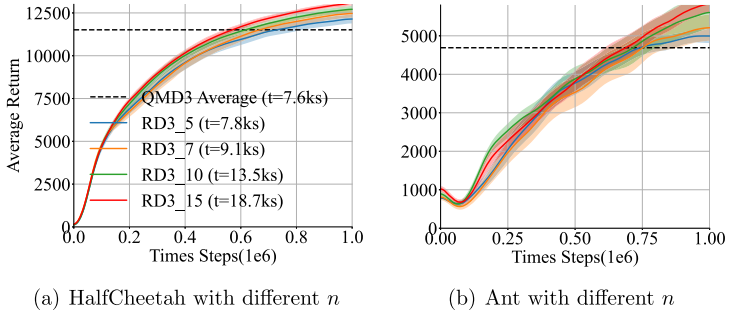


Fig. 7. The learning curves with varying n. The average return of last ten evaluation of QMD3 is shown as the black dotted line.

Table 1
Benchmark results of the maximum average return of one million time steps. The optimal value for each task is bolded.

Task	RD3	DARC	GD3	PRAG	QMD3	TADD
IP	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
IDP	**1.09**	1.03	1.02	1.01	1.03	1.01
Reacher	**1.17**	1.09	1.10	1.13	1.09	0.98
Hopper	**1.15**	1.08	1.02	0.99	1.06	1.08
HalfCheetah	**1.26**	1.18	1.16	1.12	1.21	1.10
Walker2d	**1.29**	1.13	1.10	1.02	1.17	1.04
Ant	1.36	**1.39**	1.23	1.21	1.21	1.00
Humanoid	**1.13**	1.07	1.02	1.01	1.07	1.03
Mean Score	**1.18**	1.12	1.08	1.06	1.10	1.03
Computation Cost	2.76	2.58	7.04	**1.83**	2.69	2.52

Table 2
Benchmark results of the average return of last ten evaluation. The optimal value for each task is bolded.

Task	RD3	DARC	GD3	PRAG	QMD3	TADD
IP	**1.08**	1.06	1.05	1.05	1.02	1.03
IDP	**1.18**	1.08	1.08	1.08	1.13	1.04
Reacher	**1.24**	1.05	1.05	1.02	1.16	1.05
Hopper	**1.14**	1.07	1.06	1.05	1.10	1.01
HalfCheetah	**1.26**	1.11	1.09	1.07	1.19	1.11
Walker2d	**1.18**	1.09	1.01	0.95	1.01	0.93
Ant	1.27	**1.29**	1.08	1.06	1.16	0.98
Humanoid	**1.16**	1.09	1.07	1.03	1.11	1.07
Mean Score	**1.19**	1.10	1.06	1.04	1.11	1.03

CRediT authorship contribution statement

Yujia Zhang: Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Lin Li:** Supervision, Writing – review & editing. **Wei Wei:** Funding acquisition, Project administration, Supervision, Writing – review & editing. **Xiu You:** Resources, Writing – review & editing. **Jiye Liang:** Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Nos. 62276160, 62373232), and the Natural Science Foundation of Shanxi Province, China (No. 202203021211294).

References

[1] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[2] Z. Yao, J. Yu, J. Zhang, W. He, Graph and dynamics interpretation in robotic reinforcement learning task, Inf. Sci. 611 (2022) 317–334.
[3] Y. Matsuo, Y. LeCun, M. Sahani, D. Precup, D. Silver, M. Sugiyama, E. Uchibe, J. Morimoto, Deep learning, reinforcement learning, and world models, Neural Netw. (2022).
[4] R. Zhu, L. Li, S. Wu, P. Lv, Y. Li, M. Xu, Multi-agent broad reinforcement learning for intelligent traffic light control, Inf. Sci. 619 (2023) 509–525.
[5] J. Deng, S. Sierla, J. Sun, V. Vyatkin, Offline reinforcement learning for industrial process control: a case study from steel industry, Inf. Sci. 632 (2023) 221–231.
[6] C.J. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292.
[7] X. Xu, L. Zuo, Z. Huang, Reinforcement learning algorithms with function approximation: recent advances and applications, Inf. Sci. 261 (2014) 1–31.
[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint, arXiv:1312.5602, 2013.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[10] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[11] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: NeurIPS, 2012.
[12] S. Thrun, A. Schwartz, Issues in using function approximation for reinforcement learning, in: Proceedings of the Fourth Connectionist Models Summer School, 1993, pp. 255–263.
[13] I. Szita, A. Lőrincz, The many faces of optimism: a unifying approach, in: ICML, 2008, pp. 1048–1055.
[14] A.L. Strehl, L. Li, M.L. Littman, Reinforcement learning in finite mdps: pac analysis, J. Mach. Learn. Res. 10 (11) (2009) 2413–2444.
[15] D. Lee, B. Defourny, W.B. Powell, Bias-corrected q-learning to control max-operator bias in q-learning, in: ADPRL, 2013, pp. 93–99.
[16] R. Fox, A. Pakman, N. Tishby, Taming the noise in reinforcement learning via soft updates, in: UAI, 2016, pp. 202–211.
[17] C. D'Eramo, M. Restelli, A. Nuara, Estimating maximum expected value through Gaussian approximation, in: ICML, 2016, pp. 1032–1040.
[18] Z. Song, R. Parr, L. Carin, Revisiting the softmax bellman operator: new benefits and new perspective, in: ICML, 2019, pp. 5916–5925.
[19] O. Anschel, N. Baram, N. Shimkin, Averaged-dqn: variance reduction and stabilization for deep reinforcement learning, in: ICML, 2017, pp. 176–185.
[20] H. van Hasselt, Double q-learning, in: NIPS, 2010, pp. 2613–2621.
[21] H. van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double q-learning, in: AAAI, 2016, pp. 2094–2100.
[22] Z. Zhang, Z. Pan, M.J. Kochenderfer, Weighted double q-learning, in: IJCAI, 2017, pp. 3455–3461.
[23] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, in: ICLR, 2016.
[24] S. Fujimoto, H. van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: ICML, 2018, pp. 1582–1591.
[25] Q. Lan, Y. Pan, A. Fyshe, M. White, Maxmin q-learning: controlling the estimation bias of q-learning, in: ICLR, 2020.
[26] X. Chen, C. Wang, Z. Zhou, K.W. Ross, Randomized ensembled double q-learning: learning fast without a model, in: ICLR, 2021.
[27] H. Jiang, J. Xie, J. Yang, Action candidate based clipped double q-learning for discrete and continuous action tasks, in: AAAI, vol. 35, 2021, pp. 7979–7986.
[28] X. Li, Z. Qiao, A. Gong, J. Lyu, C. Yu, J. Yan, X. Li, Prag: periodic regularized action gradient for efficient continuous control, in: PRICA, 2022, pp. 106–119.
[29] J. Lyu, Y. Yang, J. Yan, X. Li, Value activation for bias alleviation: generalized-activated deep double deterministic policy gradients, Neurocomputing 518 (2023) 70–81.
[30] D. Wu, X. Dong, J. Shen, S.C.H. Hoi, Reducing estimation bias via triplet-average deep deterministic policy gradient, IEEE Trans. Neural Netw. Learn. Syst. 31 (11) (2020) 4933–4945.
[31] W. Wei, Y. Zhang, J. Liang, L. Li, Y. Li, Controlling underestimation bias in reinforcement learning via quasi-median operation, in: AAAI, vol. 36, 2022, pp. 8621–8628.
[32] J. Lyu, X. Ma, J. Yan, X. Li, Efficient continuous control with double actors and regularized critics, in: AAAI, vol. 36, 2022, pp. 7655–7663.
[33] G. Tesauro, et al., Temporal difference learning and td-gammon, Commun. ACM 38 (3) (1995) 58–68.
[34] R. Bellman, A Markovian decision process, J. Math. Mech. (1957) 679–684.
[35] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, in: ICML, 2014, pp. 387–395.
[36] H.A. David, H.N. Nagaraja, Order Statistics, John Wiley & Sons, 2004.
[37] D. Bertsimas, K. Natarajan, C.-P. Teo, Tight bounds on expected order statistics, Probab. Eng. Inf. Sci. 20 (4) (2006) 667–686.
[38] D. Bertsimas, K. Natarajan, C.-P. Teo, Probabilistic combinatorial optimization: moments, semidefinite programming, and asymptotic bounds, SIAM J. Optim. 15 (1) (2004) 185–209.
[39] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, CoRR, 2016.
[40] N. Tasfi, Pygame learning environment, https://github.com/ntasfi/PyGame-Learning-Environment. (Accessed 12 April 2016), 2016.
[41] K. Young, T. Tian, Minatar: an atari-inspired testbed for more efficient reinforcement learning experiments, arXiv preprint, arXiv:1903.03176, 2019.
[42] E. Todorov, T. Erez, Y. Tassa, Mujoco: a physics engine for model-based control, in: IROS, 2012, pp. 5026–5033.