Mildly Conservative Regularized Evaluation for Offline Reinforcement Learning

Haohui Chen and Zhiyong Chen

***Abstract*—Offline reinforcement learning (RL) seeks to learn optimal policies from static datasets without further environment interaction. A key challenge is the distribution shift between the learned and behavior policies, leading to out-of-distribution (OOD) actions and overestimation. To prevent gross overestimation, the value function must remain conservative; however, excessive conservatism may hinder performance improvement. To address this, we propose the mildly conservative regularized evaluation (MCRE) framework, which balances conservatism and performance by combining temporal difference (TD) error with a behavior cloning term in the Bellman backup. Building on this, we develop the mildly conservative regularized Q-learning (MCRQ) algorithm, which integrates MCRE into an off-policy actor–critic framework. Experiments show that MCRQ outperforms strong baselines and state-of-the-art offline RL algorithms on benchmark datasets.**

***Index Terms*—Offline reinforcement learning, Actor-critic, Value overestimation, Q-function regularization, Bellman backup**

I. Introduction

Reinforcement learning (RL) has achieved success across various domains. In classical online RL, agents learn optimal policies through real-time interaction. However, in real-world settings, continuous interaction is often impractical due to data collection challenges, safety concerns, and high costs. Offline RL, also known as batch RL [1], addresses this by learning policies from static datasets generated by unknown behavior policies, eliminating the need for environment interaction. This makes offline RL suitable for real-world applications without simulators, such as energy optimization [2], robotics [3], and recommendation systems [4].

A key challenge in offline RL is the distribution shift between the learned policy and the behavior policy, as the latter often fails to sufficiently cover the state–action space. As a result, directly applying off-policy online RL algorithms in offline settings typically yields poor performance [5]. The evaluation of out-of-distribution (OOD) actions introduces extrapolation errors, which can be amplified through bootstrapping, leading to significant overestimation bias [6].

In critic regularization, the regularization term is typically integrated directly into the critic loss. The core idea is to mitigate overestimation of OOD actions by regularizing the Q-function or state value function, thereby addressing distribution shift. However, overly conservative regularization can lead to excessively low Q-values [7], limiting the agent's ability to explore and exploit, and exposing the actor to inaccurate Q-values, ultimately slowing convergence or leading to suboptimal policies [8]. The mildly conservative regularized evaluation (MCRE) proposed in this paper, a form of critic regularization, highlights how over-conservatism impedes policy improvement by analyzing the gap between the learned and true Q-values, as further supported by ablation experiments.

In actor regularization, policy update stability is often achieved by constraining the target policy to remain close to the behavior policy distribution. However, excessive regularization can hinder exploration, preventing the target policy from fully utilizing the Q-function and impairing convergence. Over-conservatism poses challenges even when learning from expert datasets, regardless of whether regularization is applied to the critic or actor [9]. Therefore, conservatism should remain mild [10].

Temporal difference (TD) error is a fundamental mechanism for aligning value estimates with temporal targets, guiding policy optimization through iterative feedback. However, in offline RL, distribution shift and limited data coverage introduce compounding biases and OOD actions, which are amplified through TD updates. To address this, the proposed MCRE framework integrates TD error with a behavior cloning term to improve value estimation and reduce the impact of distribution shift and OOD actions. MCRE combines two complementary components: TD error regularization, which refines value estimates via temporal feedback, and behavior cloning, which anchors the policy to the dataset's empirical action distribution. Together, they form a mildly conservative regularization term.

This dual mechanism ensures that TD updates operate within a policy-constrained regime, suppressing OOD actions while preserving their role as corrective signals. Crucially, MCRE imposes only a mild constraint on the TD error, allowing the target policy to deviate slightly from the behavior policy and thus avoid over-conservatism. By embedding this constraint into the standard Bellman backup, MCRE enables effective policy optimization without incurring performance degradation due to excessive conservatism.

The primary contributions of this paper are summarized as follows.

1) We propose a novel framework that integrates TD error with a behavior cloning term within the standard Bellman backup. This unified design balances value estimation and policy conservatism, with the behavior cloning term constraining policy updates to the support of the behavior distribution, effectively suppressing OOD actions.

Haohui Chen is with the School of Automation, Central South University, Changsha 410083, China (e-mail: haohuichen@csu.edu.cn).

Zhiyong Chen is with the School of Engineering, University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: zhiyong.chen@newcastle.edu.au).

Corresponding author: Zhiyong Chen.

2) We theoretically prove that MCRE converges under both sampling and no-sampling error conditions. In both cases, the gap between the learned and true Q-value and state-value functions is effectively bounded. Furthermore, we analyze policy suboptimality in offline RL, showing that the difference between the learned suboptimal policy and the true optimal policy is also upper-bounded.

3) We provide a theoretical analysis of the Q-function learned by MCRE under both sampling and no-sampling error, showing that stronger constraints lead to over-conservatism, increasing the gap between the learned and true Q-functions and degrading performance. This finding is supported by ablation studies, which reveal a positive correlation between conservatism in MCRE and the strength of its constraints.

4) Based on MCRE, we propose a novel and efficient offline RL algorithm, mildly conservative regularized Q-learning (MCRQ), which integrates an offline actor-critic framework to effectively mitigate the challenges of over-conservatism and distribution shift. Experimental results demonstrate that MCRQ outperforms baselines and state-of-the-art (SOTA) algorithms on various MuJoCo tasks in the D4RL benchmark.

The remainder of the paper is organized as follows. Section II reviews related work and the motivation behind MCRE. Section III introduces the necessary preliminaries. Section IV details the MCRE framework and provides its theoretical analysis. Section V presents the practical implementation of the MCRQ algorithm. Section VI reports experimental results on D4RL tasks and compares MCRQ with baselines and recent SOTA algorithms to demonstrate its superior performance. Finally, Section VII concludes the paper and discusses potential future directions.

II. Related Work

Offline RL algorithms that mitigate over-conservatism and distribution shift can be classified into critic and actor regularization methods [11], as discussed below.

A. Critic Regularization

Critic regularization typically imposes a penalty on the TD target or critic loss to make the learned value function closer to the true one. Fakoor et al. [12] proposed continuous doubly constrained (CDC), a method for batch RL that adds two novel batch-RL regularizers to the standard off-policy actor critic (AC) algorithm, for mitigating the overestimation bias caused by distribution shift. To reduce high variance observed in multi-step evaluation, Brandfonbrener et al. [13] developed a one-step framework, which implements constrained policy improvement. Fujimoto et al. [14] enhanced the traditional twin delayed deep deterministic policy gradient (TD3) by integrating behavior cloning to alleviate the impact of distribution shift. To mitigate the optimistic Q-values caused by iterative errors during policy optimization, Kumar et al. [15] proposed conservative Q-learning (CQL), which aims to learn a conservative Q-function. To alleviate the computational burden of CQL, Kostrikov et al. [16] replaced the KL divergence in CQL with the fisher divergence. Kostrikov et al. [17] introduced implicit Q-learning (IQL), an on-policy SARSA-style offline RL algorithm that uses advantage-weighted behavior cloning to avoid querying OOD actions. Bai et al. [18] introduced pessimistic bootstrapping for offline RL (PBRL), which applies pessimistic updates to Q-functions. Huang et al. [19] proposed an offline actor critic framework with behavior value regularization, which helps mitigate over-optimistic Q-values and reduce bias.

B. Actor Regularization

Actor regularization usually works through ensuring target policy stays within the distribution of behavior policy. However, this constraint tends to be overly conservative because it requires the target policy to choose an action similar to the behavior policy with a given state. This restriction may affect the performance of the target policy, especially if the behavior policy is not optimal [20]. Actor regularization methods generally add constraints on actor loss [11]. Fujimoto et al. [21] proposed batch-constrained deep Q-learning (BCQ), which employs a generative model to generate only previously observed actions. However, Kumar et al. [22] pointed out that BCQ is overly aggressive, and developed the bootstrapping error accumulation reduction (BEAR), an innovative constraint-based approach specifically designed to break the chain of error propagation in Q-learning updates. Wu et al. [23] proposed uncertainty weighted actor critic (UWAC), which utilizes Monte Carlo dropout for uncertainty estimation, significantly reducing computational overhead. Nair et al. [24] proposed advantage weighted actor critic (AWAC), which utilizes offline data and supports online fine-tuning. Ran et al. [20] proposed policy regularization with dataset constraint (PRDC), which utilizes OOD actions as guidance for policy optimization when updating a policy in a given state. PRDC imposes a milder constraint, yet still helps avoid OOD actions.

Although critic and actor regularization methods have made significant progress in mitigating OOD actions and distribution shift, critic regularization may lead to excessive pessimism, especially with OOD actions, thus hindering policy optimization. Similarly, actor regularization may limit the flexibility of a policy by constraining the target policy to remain within the distribution of behavior policy, which may restrict overall performance. These overly conservative constraints can lead to convergence to a local optimum or even cause the policy to diverge, thereby hindering further policy improvement. To alleviate these problems, we propose the MCRE framework, which reduces the risk of OOD actions and prevents performance degradation due to over-conservatism.

III. Preliminaries

The RL paradigm formalized as a Markov decision process (MDP) is characterized by the tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \rho_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces respectively, $\mathcal{P}$ defines the state transition dynamics, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the bounded reward function with $\|r\|_\infty \leq r_{\max}$, ρ_0 specifies the initial state distribution, $\gamma \in [0, 1)$ is the discount factor.

During environment interaction at timestep t, a deterministic target policy $\pi \in \Pi : \mathcal{S} \to \mathcal{A}$ generates an action $a_t =$

$\pi(s_t)$, which leads to a transition to the next state according to $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ and yields an instantaneous scalar reward r_t. The γ-discounted trajectory return is formulated as $R_t^\gamma = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$. For a policy π, it is defined as

$$\Gamma(\pi) = \mathbb{E}[R_0^\gamma]. \tag{1}$$

The objective in RL is to derive a Bellman-optimal mapping π_* that maximizes $\Gamma(\pi)$

$$\pi_* = \arg\max_\pi \Gamma(\pi). \tag{2}$$

The state value function (V-function) is $V^\pi(s) = \mathbb{E}[R_0^\gamma \mid s_0 = s]$, and the state-action value function (Q-function) is $Q^\pi(s,a) = \mathbb{E}[R_0^\gamma \mid s_0 = s, a_0 = a]$. For deterministic policies, $V^\pi(s) = Q^\pi(s,a)$. Consequently, both are the unique fixed points of the Bellman equations:

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[Q^\pi(s', \pi(s'))] \tag{3}$$

$$V^\pi(s) = r(s,\pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,\pi(s))}[V^\pi(s')]. \tag{4}$$

The Bellman backup for obtaining the corresponding Q-function is defined as

$$(\mathcal{T}^\pi Q)(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[Q(s', \pi(s'))], \tag{5}$$

which is also the actual Bellman operator. Thus, we can rewrite (3) as $Q^\pi(s,a) = (\mathcal{T}^\pi Q^\pi)(s,a)$.

Given a dataset $\mathcal{D} = \{(s,a,r,s')\}$ consisting of tuples from trajectories collected under some unknown behavior policies, the Q-function is updated according to the following rule:

$$\hat{Q}_{k+1} = \arg\min_Q \mathbb{E}_{(s,a)\sim\mathcal{D}}[(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) - Q(s,a)]^2, \tag{6}$$

where $\hat{\mathcal{T}}^\pi$ denotes an empirical Bellman operator, which approximates the Bellman operator in (5) using a finite number of samples. Specifically,

$$(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) = r + \gamma \mathbb{E}_{s' \sim \hat{\mathcal{P}}(\cdot|s,a)}[\hat{Q}_k(s', \pi(s'))], \tag{7}$$

where $\hat{\mathcal{P}}$ is the empirical state transition probability. The expectation $\mathbb{E}_{(s,a)\sim\mathcal{D}}$ in (6) is computed as the empirical mean over the samples in $\mathcal{D}$, and is subsequently abbreviated as $\mathbb{E}$ for notational simplicity. Accordingly, the policy improvement step is defined as:

$$\pi_{k+1} = \arg\max_\pi \mathbb{E}[\hat{Q}_{k+1}(s, \pi(s))]. \tag{8}$$

Under the offline RL paradigm, agents infer the optimal policy from a fixed dataset $\mathcal{D}$ generated by an unknown behavior policy, denoted as π_β. Online off-policy algorithms are susceptible to distribution shift because $a' \sim \pi(s')$ used in the Bellman backup (7) may not be sampled from the behavior policy π_β, since a' can fall outside the distribution of π_β owing to the disparity between the distributions of π and π_β, which significantly disrupts the training process [21].

During policy evaluation, the Bellman operator employs actions drawn from the evaluation policy to compute state-action value updates. In contrast, the Q-function is exclusively updated on behavior policy-generated action samples derived from the static dataset. This discrepancy leads to extrapolation errors, causing the off-policy AC algorithm to produce inaccurately high Q-values for OOD actions [14], [25].

IV. Mildly Conservative Regularized Evaluation

This section introduces the MCRE framework and analyzes its properties.

A. Framework of MCRE

TD error assesses the accuracy of the current Q estimate relative to the TD target [26]. When the current estimate $Q(s_t, \pi(s_t))$ becomes an unbiased approximation of the true value $Q^\pi(s,a)$, TD error satisfies zero under expectation. The TD error reflects not only the discrepancy between two time steps but also, more importantly, the discrepancy between the estimate $Q(s_t, \pi(s_t))$ and the true state-action value $Q^\pi(s,a)$. The core mechanism of TD learning is correcting $Q(s_t, \pi(s_t))$ through newly obtained information [27]. However, under the offline RL setting, $Q(s_t, \pi(s_t))$ might already deviate significantly from $Q^\pi(s,a)$ due to distribution shift or limited coverage of the dataset. Consequently, TD error itself could also be biased.

Unlike prior works, we propose a mechanism to constrain deviations of the learned Q-function from its true value by quantifying the discrepancy between the current Q-estimate and the TD error. However, inaccuracies in TD error may arise due to policy mismatch between $\pi(s)$ and $\pi_\beta(s)$. To mitigate this, we introduce a behavior cloning term, which encourages the learned actions to align with those in the dataset and helps avoid querying OOD actions. Furthermore, by integrating TD error as a correction term into the standard Bellman backup, the proposed MCRE framework jointly suppresses distribution shift and OOD actions inherent to offline RL.

In contrast to (6), a new Q-function update rule is given by

$$\hat{Q}_{k+1} = \arg\min_Q \mathbb{E}[(\hat{\mathcal{Z}}^\pi \hat{Q}_k)(s,a) - Q(s,a)]^2, \tag{9}$$

where $(\hat{\mathcal{Z}}^\pi \hat{Q}_k)(s,a)$ is the modified empirical Bellman operator, defined as

$$\begin{aligned}(\hat{\mathcal{Z}}^\pi \hat{Q}_k)(s,a) =& (1-\upsilon)(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) \\ &+ \upsilon(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a) - \gamma \mathcal{I}^\pi(s,a)\end{aligned} \tag{10}$$

which consists of three components: the empirical Bellman operator defined in (7), the empirical TD Bellman operator defined as

$$\begin{aligned}(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a) = r + \gamma \mathbb{E}_{s' \sim \hat{\mathcal{P}}(\cdot|s,a)}\big[&\hat{Q}_k(s', \pi(s')) \\ &-\big(r + \gamma \hat{Q}_k(s', \pi(s')) - \hat{Q}_k(s, \pi(s))\big)\big],\end{aligned} \tag{11}$$

and the behavior cloning term

$$\mathcal{I}^\pi(s,a) = \omega(\pi(s) - a)^2.$$

Here, $\upsilon \in [0,1]$ and $\omega \geq 0$ are hyperparameters that modulate the weights of their respective components.

Applying the modified empirical Bellman operator (10) in the update rule (9) is referred to as MCRE. The concept of mild conservativeness is described below. The empirical TD Bellman operator (11) can be rewritten as

$$\begin{aligned}(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a) =& (\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) \\ &- \gamma\big((\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) - \hat{Q}_k(s, \pi(s))\big).\end{aligned}$$

When $(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) > \hat{Q}_k(s,\pi(s))$, indicating an overestimation, $(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a)$ yields a lower value than $(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a)$. Conversely, when $(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) < \hat{Q}_k(s,\pi(s))$, indicating an underestimation, $(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a)$ yields a higher value. In both cases, $(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a)$ is drawn closer to $\hat{Q}_k(s,\pi(s))$. This adjustment effect becomes more pronounced as the discount factor γ approaches 1.

As γ approaches 1, encouraging agents to prioritize long-term rewards over short-sighted decisions [27], $(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a)$ tends to align more closely with $\hat{Q}_k(s,\pi(s))$. This calibration is further justified by the regularization term $\mathcal{I}^\pi(s,a)$, which promotes alignment with $\hat{Q}_k(s,\pi(s))$ only when the corresponding state-action pairs are well supported by the offline dataset.

In offline RL, conservatism refers to avoiding the overestimation of Q-values for OOD actions, thereby mitigating extrapolation errors. CQL [15] addresses this by minimizing the maximum Q-value produced by the target policy while maximizing the Q-value derived from the behavior policy. SVR [28] enforces the Q-values of all OOD actions to equal the minimum Q-value observed in the offline dataset. Both methods adopt overly conservative strategies.

In contrast, MCRE introduces a mildly conservative mechanism via $(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a)$, which pulls the Q-function toward $\hat{Q}_k(s,\pi(s))$. Here, $\pi(s)$ is itself regularized through $\mathcal{I}^\pi$, ensuring it remains close to the dataset distribution. Consequently, if $(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a)$ becomes overly optimistic or overly conservative, this formulation introduces a corrective adjustment toward more reliable value estimates. Rather than aggressively suppressing Q-values in unsupported regions, MCRE promotes reliability through targeted regularization. Thus, compared to CQL and SVR, the conservatism imposed by MCRE is mild, offering a balance between caution and expressiveness in value estimation.

The iterative update rule for $\hat{Q}_{k+1}$ can be derived by equating the gradient of the loss function in (9) to zero

$$\frac{\partial}{\partial Q(s,a)} \mathbb{E}[(\hat{\mathcal{Z}}^\pi \hat{Q}_k)(s,a) - Q(s,a)]^2 = 0,$$

which yields the following closed-form expression:

$$\hat{Q}_{k+1}(s,a) = (\hat{\mathcal{Z}}^\pi \hat{Q}_k)(s,a). \tag{12}$$

B. Convergence Analysis of MRCE

The operator $\hat{\mathcal{T}}^\pi$ in (10) is the same as that used in MOAC, where its contraction property has already been established [29]. The key distinction in the MCRE setting lies in the incorporation of the additional term $(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s,a)$. Accordingly, we focus on establishing the contraction property of the operator $\hat{\mathcal{H}}^\pi$. It follows that the overall operator $\hat{\mathcal{Z}}^\pi$ also satisfies the contraction property, thereby guaranteeing the convergence of the Q-function update rule. The result is summarized in the following theorem.

Theorem 1. *The MCRE operator $\hat{\mathcal{Z}}^\pi$ is a contraction mapping in the $\mathcal{L}_\infty$ norm, given any initial Q-value function, and the sequence $\hat{Q}_0(s,a), \hat{Q}_1(s,a), \ldots$ generated by the iterative update rule* (12) *converges to a fixed point.*

Proof. We begin by proving that $\hat{\mathcal{Z}}^\pi$ is a contraction mapping. For any two distinct Q-functions $\hat{Q}_1(s,a)$ and $\hat{Q}_2(s,a)$, it follows from the definition (10) that

$$\begin{aligned} &\left\|(\hat{\mathcal{Z}}^\pi \hat{Q}_1)(s,a) - (\hat{\mathcal{Z}}^\pi \hat{Q}_2)(s,a)\right\|_\infty \\ \le&(1-\upsilon)\big\|\big(\hat{\mathcal{T}}^\pi \hat{Q}_1\big)(s,a) - \big(\hat{\mathcal{T}}^\pi \hat{Q}_2\big)(s,a)\big\|_\infty \\ &+ \upsilon\big\|\big(\hat{\mathcal{H}}^\pi \hat{Q}_1\big)(s,a) - \big(\hat{\mathcal{H}}^\pi \hat{Q}_2\big)(s,a)\big\|_\infty. \end{aligned} \tag{13}$$

The contraction property of $\hat{\mathcal{T}}^\pi$ is given by [29] as follows:

$$\begin{aligned} &\big\|(\hat{\mathcal{T}}^\pi \hat{Q}_1)(s,a) - (\hat{\mathcal{T}}^\pi \hat{Q}_2)(s,a)\big\|_\infty \\ \le&\gamma \max_{s,a} \big|\hat{Q}_1(s,a) - \hat{Q}_2(s,a)\big|. \end{aligned} \tag{14}$$

Next, we perform the following calculation for $\hat{\mathcal{H}}^\pi$:

$$\begin{aligned} &\left\|(\hat{\mathcal{H}}^\pi \hat{Q}_1)(s,a) - (\hat{\mathcal{H}}^\pi \hat{Q}_2)(s,a)\right\|_\infty \\ =&\gamma\left\|(1-\gamma)\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,a)}\Big[\hat{Q}_1(s',\pi(s')) - \hat{Q}_2(s',\pi(s'))\Big]\right. \\ &\left. - \Big[\hat{Q}_1(s,\pi(s)) - \hat{Q}_2(s,\pi(s))\Big]\right\|_\infty \\ \le&(\gamma-\gamma^2)\max_{s'}\Big|\hat{Q}_1(s',\pi(s')) - \hat{Q}_2(s',\pi(s'))\Big| \\ &+ \gamma\max_{s}\Big|\hat{Q}_1(s,\pi(s)) - \hat{Q}_2(s,\pi(s))\Big| \\ \le&(2\gamma-\gamma^2)\max_{s,a}\big|\hat{Q}_1(s,a) - \hat{Q}_2(s,a)\big|. \end{aligned} \tag{15}$$

Then, substituting (14) and (15) into (13) yields

$$\begin{aligned} &\|(\hat{\mathcal{Z}}^\pi \hat{Q}_1)(s,a) - (\hat{\mathcal{Z}}^\pi \hat{Q}_2)(s,a)\|_\infty \\ \le&(\gamma+\upsilon\gamma-\upsilon\gamma^2)\max_{s,a}|\hat{Q}_1(s,a) - \hat{Q}_2(s,a)|. \end{aligned} \tag{16}$$

Since $(\gamma+\upsilon\gamma-\upsilon\gamma^2) \in [0,1)$ for $\gamma \in [0,1)$ and $\upsilon \in [0,1]$, it follows that $\hat{\mathcal{Z}}^\pi$ is a contraction mapping under the $\mathcal{L}_\infty$ norm.

This contraction property guarantees the presence and uniqueness of a fixed point $\hat{Q}_*^\pi$ satisfying $\hat{Q}_*^\pi = (\hat{\mathcal{Z}}^\pi \hat{Q}_*^\pi)$. The convergence of the sequence generated by the iterative update rule (12) is rigorously characterized by $\|\hat{Q}_k - \hat{Q}_*^\pi\|_\infty \le (\gamma + \upsilon\gamma - \upsilon\gamma^2)^k \|\hat{Q}_0 - \hat{Q}_*^\pi\|_\infty$. This result establishes a geometric convergence rate, ensuring that $\hat{Q}_k$ converges uniformly to $\hat{Q}_*^\pi$ as $k \to \infty$. □

C. Error Analysis of MCRE Value Estimates

When there is no sampling error, the empirical expectation $\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,a)}$ is replaced by the true expectation $\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}$, and $\hat{\mathcal{T}}^\pi$ is replaced by $\mathcal{T}^\pi$ in (10) and (11). Under this setting, we define the corresponding modified actual Bellman operator $\mathcal{Z}^\pi$ and the actual TD Bellman operator $\mathcal{H}^\pi$. We first conduct an error analysis of the Q-function obtained via MCRE using these actual operators. Specifically, our goal is to characterize an upper bound on the discrepancy between the learned Q-function and the true value function. Moreover, we also characterize the discrepancy between the V-function obtained through MCRE and the true value.

The results in the following two theorems hold when υ is sufficiently small in (10) to ensure that $\hat{\mathcal{T}}^\pi$ is not dominated;

specifically, when $\upsilon < (1-\gamma)/(\gamma^2+\gamma)$, or equivalently, $1-\gamma-\upsilon\gamma^2-\upsilon\gamma > 0$.

Theorem 2. *(Absence of Sampling Error) For $(s,a) \in \mathcal{D}$, the learned Q-function $Q_*^\pi(s,a)$, obtained from* (12) *without sampling error, and the true Q-function $Q^\pi(s,a)$ satisfy*

$$\|Q_*^\pi(s,a) - Q^\pi(s,a)\|_\infty \leq \frac{\gamma \max_{s,a} \mathcal{I}^\pi(s,a)}{1-\gamma-\upsilon\gamma^2-\upsilon\gamma}. \tag{17}$$

Moreover, the learned V-function $V_*^\pi(s)$ exactly matches the true V-function $V^\pi(s)$, i.e., $V_*^\pi(s) = V^\pi(s)$.

Proof. Since $Q_*^\pi = (\mathcal{Z}^\pi Q_*^\pi)$, as established in Theorem 1, and based on (10) without sampling error, we have

$$\begin{aligned} Q_*^\pi(s,a) =& (1-\upsilon)(\mathcal{T}^\pi Q_*^\pi)(s,a) \\ &+ \upsilon(\mathcal{H}^\pi Q_*^\pi)(s,a) - \gamma \mathcal{I}^\pi(s,a). \end{aligned}$$

According to (3), (5) and (11), we obtain

$$\begin{aligned} &\Big|Q_*^\pi(s,a) - Q^\pi(s,a)\Big| \\ =&\Bigg| r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[(1-\upsilon) Q_*^\pi(s',\pi(s')) \\ &+ \upsilon\Big(Q_*^\pi(s',\pi(s')) - \big(r(s,a) + \gamma Q_*^\pi(s',\pi(s')) \\ &- Q_*^\pi(s,\pi(s))\big)\Big) - \mathcal{I}^\pi(s,a)\Big] - r(s,a) \\ &- \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[Q^\pi(s',\pi(s'))\Big]\Bigg|. \end{aligned} \tag{18}$$

The calculation then proceeds as follows:

$$\begin{aligned} &\Big|Q_*^\pi(s,a) - Q^\pi(s,a)\Big| \\ \leq& \gamma \Bigg| \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[Q_*^\pi(s',\pi(s')) - Q^\pi(s',\pi(s'))\Big] \\ &+ \upsilon \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[-\gamma Q_*^\pi(s',\pi(s')) - r(s,a) \\ &+ Q_*^\pi(s,\pi(s))\Big]\Bigg| + \gamma \max_{s,a} \mathcal{I}^\pi(s,a) \\ =& \gamma \Bigg| \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[Q_*^\pi(s',\pi(s')) - Q^\pi(s',\pi(s'))\Big] \\ &+ \upsilon \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[-\gamma\big(Q_*^\pi(s',\pi(s')) - Q^\pi(s',\pi(s'))\big) \\ &+ Q_*^\pi(s,\pi(s)) - Q^\pi(s,a)\Big]\Bigg| + \gamma \max_{s,a} \mathcal{I}^\pi(s,a) \\ \leq& \gamma \Bigg| \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[Q_*^\pi(s',\pi(s')) - Q^\pi(s',\pi(s'))\Big]\Bigg| \\ &+ \upsilon\gamma \Bigg| \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big[-\gamma\big(Q_*^\pi(s',\pi(s')) - Q^\pi(s',\pi(s'))\big)\Big]\Bigg| \\ &+ \upsilon\gamma \Bigg| Q_*^\pi(s,\pi(s)) - Q^\pi(s,a)\Bigg| + \gamma \max_{s,a} \mathcal{I}^\pi(s,a) \\ \leq& (\gamma + \upsilon\gamma^2 + \upsilon\gamma) \max_{s,a} \Big|Q_*^\pi(s,a) - Q^\pi(s,a)\Big| \\ &+ \gamma \max_{s,a} \mathcal{I}^\pi(s,a). \end{aligned} \tag{19}$$

Under the condition $1-\gamma-\upsilon\gamma^2-\upsilon\gamma > 0$, the bound in (17) follows directly from (19).

Next, since π is a deterministic policy, and using a similar calculation as in (18), we have

$$\begin{aligned} &V_*^\pi(s) = Q_*^\pi(s,\pi(s)) \\ =& r(s,\pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,\pi(s))} \Big[(1-\upsilon) V_*^\pi(s') \\ &+ \upsilon\Big(V_*^\pi(s') - \big(r(s,\pi(s)) + \gamma V_*^\pi(s') - V_*^\pi(s)\big)\Big)\Big] \\ =& (1-\upsilon\gamma) r(s,\pi(s)) + \upsilon\gamma V_*^\pi(s) \\ &+ \big(\gamma - \upsilon\gamma^2\big) \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,\pi(s))} \big[V_*^\pi(s')\big]. \end{aligned} \tag{20}$$

By rearranging terms and solving for $V_*^\pi(s)$ under the condition $1-\upsilon\gamma > 0$, we obtain the recursive relationship:

$$V_*^\pi(s) = r(s,\pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,\pi(s))}[V_*^\pi(s')]. \tag{21}$$

This shows $V_*^\pi(s)$ is the solution of (4), and hence $V_*^\pi(s) = V^\pi(s)$. The theorem is thus proved. □

In practical offline RL settings where sampling error is present, the actual operators are typically inaccessible. However, when the dataset $\mathcal{D}$ provides sufficient coverage of the state-action space, their empirical counterparts serve as reasonable approximations. In what follows, we conduct an error analysis of the Q-function obtained via MCRE under this empirical scenario.

To rigorously characterize the discrepancy between the empirical and true operators, we introduce the following assumption on the deviation between $\hat{\mathcal{P}}(s' \mid s,a)$ and $\mathcal{P}(s' \mid s,a)$.

Assumption 1. *[15], [29]: For any state-action pair $(s,a) \in \mathcal{D}$, the ℓ_1-norm deviation between the empirical and actual state transition dynamics satisfies the following bound with probability larger than $1-\delta$:*

$$|\hat{\mathcal{P}}(s' \mid s,a) - \mathcal{P}(s' \mid s,a)| \leq \frac{c_p}{\sqrt{\mathcal{D}_c}} \tag{22}$$

where $\delta \in (0,1)$, $c_p > 0$ is a constant, and $\mathcal{D}_c \neq 0$ denotes the count of state-action pairs in $\mathcal{D}$. When $\mathcal{D}_c = 0$, the bound holds trivially if $\delta \geq 2r_{\max}/(1-\gamma)$.

Under Assumption 1, the gap between $(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a)$ and $(\mathcal{T}^\pi \hat{Q}_k)(s,a)$ for any given policy π satisfies [11]

$$\begin{aligned} &\big|(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s,a) - (\mathcal{T}^\pi \hat{Q}_k)(s,a)\big| \\ =& \gamma\big|\sum_{s'}(\hat{\mathcal{P}}(s' \mid s,a) - \mathcal{P}(s' \mid s,a))\hat{Q}_k(s',\pi(s'))\big| \\ \leq& \frac{\gamma c_p r_{\max}}{(1-\gamma)\sqrt{\mathcal{D}_c}}. \end{aligned} \tag{23}$$

We now enhance Theorem 2 to the following version.

Theorem 3. *(Presence of Sampling Error) For $(s,a) \in \mathcal{D}$, the learned Q-function $\hat{Q}_*^\pi(s,a)$, obtained from* (12) *under Assumption 1, and the true Q-function $Q^\pi(s,a)$ satisfy*

$$\|\hat{Q}_*^\pi(s,a) - Q^\pi(s,a)\|_\infty \leq \frac{\gamma \max_{s,a} \mathcal{I}^\pi(s,a)}{1-\gamma-\upsilon\gamma^2-\upsilon\gamma}$$

$$+\frac{\gamma c_p r_{\max}}{(1-\gamma-\upsilon\gamma^2-\upsilon\gamma)(1-\gamma)\sqrt{\mathcal{D}_c}}. \tag{24}$$

Moreover, the learned V-function $\hat{V}^{\pi}_{*}(s)$ and the true V-function $V^{\pi}(s)$ satisfy

$$\|\hat{V}^{\pi}_{*}(s)-V^{\pi}(s)\|_{\infty}\le\frac{\gamma c_p r_{\max}}{(1-\gamma)^2\sqrt{\mathcal{D}_c}}. \tag{25}$$

Proof. The proof is similar to that of Theorem 2. Specifically, the calculation in (19) becomes

$$\begin{aligned}
&\left|\hat{Q}^{\pi}_{*}(s,a)-Q^{\pi}(s,a)\right|\\
\le&\gamma\left|\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,a)}\left[\hat{Q}^{\pi}_{*}(s',\pi(s'))\right]-\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\left[Q^{\pi}(s',\pi(s'))\right]\right|\\
&+\upsilon\gamma\left|\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,a)}\left[-\gamma\big(\hat{Q}^{\pi}_{*}(s',\pi(s'))-Q^{\pi}(s',\pi(s'))\big)\right]\right|\\
&+\upsilon\gamma\left|Q^{\pi}_{*}(s,\pi(s))-Q^{\pi}(s,a)\right|+\gamma\max_{s,a}\mathcal{I}^{\pi}(s,a)\\
\le&(\gamma+\upsilon\gamma^2+\upsilon\gamma)\max_{s,a}\left|\hat{Q}^{\pi}_{*}(s,a)-Q^{\pi}(s,a)\right|\\
&+\gamma\max_{s,a}\mathcal{I}^{\pi}(s,a)+\gamma\Big|\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,a)}[Q^{\pi}(s',\pi(s'))]\\
&-\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[Q^{\pi}(s',\pi(s'))]\Big|,
\end{aligned} \tag{26}$$

where an additional term is incurred compared with (19). According to (23), this additional term is bounded as follows:

$$\begin{aligned}
&\gamma\left|\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,a)}[Q^{\pi}(s',\pi(s'))]-\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[Q^{\pi}(s',\pi(s'))]\right|\\
\le&\gamma\Big|\sum_{s'}\big(\hat{\mathcal{P}}(s'\mid s,a)-\mathcal{P}(s'\mid s,a)\big)Q^{\pi}(s',\pi(s'))\Big|\\
\le&\frac{\gamma c_p r_{\max}}{(1-\gamma)\sqrt{\mathcal{D}_c}}.
\end{aligned} \tag{27}$$

By substituting (27) into (26), we obtain

$$\begin{aligned}
&\max_{s,a}\left|\hat{Q}^{\pi}_{*}(s,a)-Q^{\pi}(s,a)\right|\\
\le&(\gamma+\upsilon\gamma^2+\upsilon\gamma)\max_{s,a}\left|\hat{Q}^{\pi}_{*}(s,a)-Q^{\pi}(s,a)\right|\\
&+\gamma\max_{s,a}\mathcal{I}^{\pi}(s,a)+\frac{\gamma c_p r_{\max}}{(1-\gamma)\sqrt{\mathcal{D}_c}}.
\end{aligned} \tag{28}$$

Under the condition $1-\gamma-\upsilon\gamma^2-\upsilon\gamma>0$, the bound in (24) follows directly from (28).

Next, from (21), we have

$$\hat{V}^{\pi}_{*}(s)=r(s,\pi(s))+\gamma\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,\pi(s))}[\hat{V}^{\pi}_{*}(s')]. \tag{29}$$

Combining (29) and (4) gives

$$\begin{aligned}
&\left|\hat{V}^{\pi}_{*}(s)-V^{\pi}(s)\right|\\
=&\gamma\left|\mathbb{E}_{s'\sim\hat{\mathcal{P}}(\cdot|s,\pi(s))}\left[\hat{V}^{\pi}_{*}(s')\right]-\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,\pi(s))}\left[V^{\pi}(s')\right]\right|,
\end{aligned} \tag{30}$$

which is established in Theorem 6 of [29] and it follows that the bound (25) holds. The theorem is thus proved. □

Theorems 2 and 3 characterize the discrepancy between the learned Q-function and the true Q-function under settings with and without sampling error. These results indicate that a stronger behavior cloning term $\mathcal{I}^{\pi}$, or a larger modification parameter υ in the Bellman operator, both introduced by MCRE, lead to looser bounds on the Q-function error. This highlights an inherent trade-off: while these components are essential for enforcing mild conservatism, their magnitudes must be carefully chosen to avoid compromising the theoretical bound on the discrepancy. Additionally, the theorems also quantify the corresponding gap between the learned and true V-functions under the same conditions.

D. Analysis of the Policy Derived from MCRE

Sampling error and value function bias inherent in offline actor–critic (AC) with MCRE introduce a performance discrepancy between the learned policy $\hat{\pi}*$ and the optimal policy π, as defined in (2). Specifically, $\hat{\pi}*$ and π satisfy

$$\hat{\pi}_{*}=\operatorname{argmax}_{\pi}\hat{V}^{\pi}_{*}(s),\forall s\in\mathcal{D}, \tag{31}$$

$$\pi_{*}=\operatorname{argmax}_{\pi}V^{\pi}(s),\forall s\in\mathcal{D}. \tag{32}$$

The suboptimality of $\hat{\pi}_{*}$ is defined as $\Pi(\hat{\pi}_{*})=\Gamma(\hat{\pi}_{*})-\Gamma(\pi_{*})$, where $\Gamma(\cdot)$ represents the performance metric defined in (1). This suboptimality can equivalently be rewritten as

$$\Pi(\hat{\pi}_{*})=\mathbb{E}[V^{\hat{\pi}_{*}}(s_0)-V^{\pi_{*}}(s_0)], \tag{33}$$

where s_0 is sampled from the initial state distribution ρ_0. To rigorously bound $\Pi(\hat{\pi}_{*})$, we begin by introducing the following assumption.

Assumption 2. *[29] The reward function $r(s,a)$ satisfies $|r(s,\hat{a}_1)-r(s,\hat{a}_2)|\le\ell\|\hat{a}_1-\hat{a}_2\|_{\infty}$, $\forall s\in\mathcal{S}$, $\hat{a}_1,\hat{a}_2\in\mathcal{A}$, where $\ell>0$ is an unknown constant. The action a is bounded by a constant, such that $\|a\|_{\infty}\le a_{\max}$, $\forall a\in\mathcal{A}$.*

Based on this assumption, we establish the following two theorems.

Theorem 4. *(Absence of Sampling Error) Using the V-functions in Theorem 2, and under Assumption 2, the performance difference between the learned suboptimal policy $\hat{\pi}_{*}$ and the optimal policy π_{*} satisfies*

$$\begin{aligned}
&\|\Pi(\hat{\pi}_{*})\|_{\infty}\le\frac{2\ell a_{\max}}{1-\gamma}\\
&+\frac{2\gamma r_{\max}\max_s \mathrm{TV}\big(\mathcal{P}(\cdot\mid s,\hat{\pi}_{*}(s)),\mathcal{P}(\cdot\mid s,\pi_{*}(s))\big)}{(1-\gamma)^2}.
\end{aligned} \tag{34}$$

where TV *denotes the total variation distance between probability distributions [30].*

Proof. By (33), we have

$$\begin{aligned}
\left|\Pi(\hat{\pi}_{*})\right|=&\left|\mathbb{E}\left[V^{\hat{\pi}_{*}}(s_0)-V^{\pi_{*}}(s_0)\right]\right|\\
\le&\max_{s}\left|V^{\hat{\pi}_{*}}(s)-V^{\pi_{*}}(s)\right|
\end{aligned} \tag{35}$$

Moreover,

$$\begin{aligned}
&|V^{\hat{\pi}_{*}}(s)-V^{\pi_{*}}(s)|\\
\le&|V^{\hat{\pi}_{*}}_{*}(s)-V^{\hat{\pi}_{*}}(s)|+|V^{\hat{\pi}_{*}}_{*}(s)-V^{\pi_{*}}(s)|
\end{aligned} \tag{36}$$

According to Theorem 2, we have $|V^{\hat{\pi}_{*}}_{*}(s)-V^{\hat{\pi}_{*}}(s)|=0$. To further analyze $|V^{\hat{\pi}_{*}}_{*}(s)-V^{\pi_{*}}(s)|$, we proceed with the following calculation:

$$\left|V^{\hat{\pi}_{*}}_{*}(s)-V^{\pi_{*}}(s)\right|$$

$$
\begin{aligned}
=&\Bigg| r(s, \hat{\pi}_*(s)) - r(s, \pi_*(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, \hat{\pi}_*(s))} \left[V_*^{\hat{\pi}_*}(s') \right] \\
&- \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, \pi_*(s))} \left[V^{\pi_*}(s') \right] \Bigg| .
\end{aligned}
$$

Under Assumption 2, we have

$$
\begin{aligned}
&\left| V_*^{\hat{\pi}_*}(s) - V^{\pi_*}(s) \right| \\
\le& 2\ell a_{\max} + \gamma \Bigg| \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, \hat{\pi}_*(s))} \left[V_*^{\hat{\pi}_*}(s') - V^{\hat{\pi}_*}(s') \right] \\
&+ \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, \hat{\pi}_*(s))} \left[V^{\hat{\pi}_*}(s') - V^{\pi_*}(s') \right] \\
&+ \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, \hat{\pi}_*(s))} \left[V^{\pi_*}(s') \right] - \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, \pi_*(s))} \left[V^{\pi_*}(s') \right] \Bigg| \\
\le& 2\ell a_{\max} + \gamma \max_s \left| V^{\hat{\pi}_*}(s) - V^{\pi_*}(s) \right| \\
&+ \frac{2\gamma r_{\max} \max_s \mathrm{TV}\big(\mathcal{P}(\cdot \mid s, \hat{\pi}_*(s)), \mathcal{P}(\cdot \mid s, \pi_*(s))\big)}{1-\gamma} \quad (37)
\end{aligned}
$$

Substituting (37) into (36), and subsequently into (35), completes the proof of (34). □

Theorem 5. *(Presence of Sampling Error) Using the V-functions in Theorem 3, and under Assumption 2, the performance difference between the learned suboptimal policy $\hat{\pi}_*$ and the optimal policy π_* satisfies*

$$
\begin{aligned}
\|\Pi(\hat{\pi}_*)\|_\infty \le& \frac{2\ell a_{\max}}{1-\gamma} + \frac{(1+\gamma)\gamma c_p r_{\max}}{(1-\gamma)^3 \sqrt{\mathcal{D}_c}} \\
&+ \frac{2\gamma r_{\max} \max_s \mathrm{TV}\big(\hat{\mathcal{P}}(\cdot \mid s, \hat{\pi}_*(s)), \mathcal{P}(\cdot \mid s, \pi_*(s))\big)}{(1-\gamma)^2} . \quad (38)
\end{aligned}
$$

Proof. The proof follows a similar structure to that of Theorem 4. The only difference is that $V_*^{\hat{\pi}_*}$ is replaced by$\hat{V}_*^{\hat{\pi}_*}$, and the equality $|\hat{V}_*^{\hat{\pi}_*}(s) - V^{\hat{\pi}_*}(s)| = 0$ no longer holds in the presence of sampling error. Instead, this term is bounded as shown in (25). This additional non-zero bound accounts for the extra term in (38), compared to (34). □

V. The MCRQ Algorithm

We introduce a novel offline algorithm, MCRQ, which is built upon the off-the-shelf off-policy online RL algorithm TD3 [31]. The core idea of MCRQ is to incorporate MCRE into the TD target y under the AC setting. Similar to TD3, MCRQ consists of two critic networks, Q_{θ_1} and Q_{θ_2}, along with an actor network, π_ϕ, where θ_1, θ_2, and ϕ represent the network parameters. The associated target critic networks are $Q_{\theta'_1}$ and $Q_{\theta'_2}$, and the target actor network is $\pi_{\phi'}$, where θ'_1, θ'_2, and ϕ' represent the target network parameters. The action a' is defined as

$$
a' = \pi_{\phi'}(s') + \epsilon, \quad (39)
$$

where ϵ denotes added exploration noise.

The three components of MCRE, as defined in (10) and under the AC setting, are used to construct the TD target y in MCRQ. The term

$$
y_1 = r + \gamma \min\left(Q_{\theta'_1}(s', a'), Q_{\theta'_2}(s', a')\right)
$$

corresponds to the empirical Bellman operator $(\hat{\mathcal{T}}^\pi \hat{Q}_k)(s, a)$, and adopts the same TD target formulation as TD3. The term

$$
\begin{aligned}
y_2 =& r + \gamma \bigg(\max\left(Q_{\theta'_1}(s', a'), Q_{\theta'_2}(s', a')\right) \\
&- \Big(r + \gamma \min\left(Q_{\theta'_1}(s', a'), Q_{\theta'_2}(s', a')\right) \\
&- \max\left(Q_{\theta_1}(s, \pi_\phi(s)), Q_{\theta_2}(s, \pi_\phi(s))\right) \Big) \bigg)
\end{aligned}
$$

corresponds to the empirical TD Bellman operator $(\hat{\mathcal{H}}^\pi \hat{Q}_k)(s, a)$. Here, the first $\max$ operator is used to optimistically estimate $Q_{\theta'_i}$ at s', for $i = 1, 2$, while the second $\max$ estimates Q_{θ_i} at s. Both are used to provide a balanced evaluation against the $\min$ operator. The behavior cloning term is calculated as $\mathcal{I}^\pi = \omega(\pi_\phi(s) - a)^2$, which constrains the policy $\pi_\phi(s)$ to stay close to the behavior policy. This constraint helps mitigate the effects of distributional shift and OOD actions, while ensuring that Q_{θ_i} remains supported by the offline dataset.

The TD target y of MCRQ is formally defined as

$$
y = (1 - \upsilon) y_1 + \upsilon y_2 - \gamma \mathcal{I}^\pi . \quad (40)
$$

The loss function for each critic network, parameterized by θ_i, is defined as

$$
\mathbb{L}(\theta_i) = \mathbb{E}[y - Q_{\theta_i}(s, a)]^2 . \quad (41)
$$

Through Theorems 2 and 3, we can drive $\pi_\phi(s)$ to approximate the actions in $\mathcal{D}$, thereby reducing the gap between the learned Q-function and the true one. Consequently, the difference between the action taken by the policy $\pi_\phi(s)$ and the selected action a is treated as a penalty. Thus, the loss function for the actor network $\pi_\phi(s)$ is given by

$$
J(\phi) = -\mathbb{E}[\lambda Q_{\theta_1}(s, \pi_\phi(s)) - (\pi_\phi(s) - a)^2] \quad (42)
$$

where $\lambda = \frac{\alpha}{N^{-1} \sum_{(s,a) \in \mathcal{D}} |Q_{\theta_1}(s,a)|}$ balances the contributions of Q-function and policy penalty for $\alpha \ge 0$. The MCRQ algorithm is summarized in Algorithm 1. A structural comparison between MCRQ and baseline algorithms is discussed below.

BCQ, TD3_BC, CQL, IQL, and MCRQ each employ two critic networks with corresponding target networks. Among them, TD3_BC, CQL, IQL, and MCRQ incorporate an actor network and its associated target network. In contrast, BCQ uses a perturbation network, rather than directly generating actions, which is also accompanied by a target network. It models the behavior policy with a variational auto-encoder (VAE), generating n perturbed candidate actions. These perturbed actions are used to compute the TD target. A feature of BCQ is its convex combination of the two Q-network outputs, weighting the minimum value more heavily. The actor is updated using perturbed VAE-generated actions.

TD3_BC utilizes clipped double Q-learning (CDQ) for critic updates, whereas MCRQ incorporates CDQ and additionally introduces a maximization operator. While MCRQ shares the same actor loss as TD3_BC, it further mitigates distributional shift and reduces the discrepancy between the learned and true Q-functions.

CQL minimizes the maximum Q-value produced by the target policy while maximizing the Q-value derived from the

Algorithm 1: Mildly Conservative Regularized Q-learning (MCRQ)

```
Initialize critic networks $Q_{\theta_1}, Q_{\theta_2}$ and actor network $\pi_\phi$ with random parameters $\theta_1, \theta_2, \phi$, and initialize target networks with parameters $\theta_1'$, $\theta_2'$, $\phi'$.
Initialize target network smoothing parameter $\tau$ and actor network update frequency $d$.
Initialize offline dataset $\mathcal{D}$.
for $t$ = 1 to $T$ do
    Sample $\{(s_t, a_t, r_t, s_{t+1}')\}_{t=1}^{N} \sim \mathcal{D}$
    Calculate $a'$ with (39)
    Calculate $y$ with (40)
    Update critic $\theta_i$ by minimizing (41)
    if $t$ mod $d$ then
        Update actor $\phi$ by minimizing (42)
        Update target networks:
        $\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i'$
        $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
    end if
end for
```

behavior policy. It also regularizes the target policy to stay close to the behavior policy, reducing the impact of OOD actions. The actor network is updated using the soft actor-critic methodology.

IQL is a SARSA-style in-sample algorithm that computes the TD target using actions within the data distribution. It uses expectile regression to address the limitations of the traditional maximization operator. As large target values may stem from lucky transitions rather than a single optimal action, the value network is trained with expectile regression for TD updates. The actor is optimized via policy extraction.

In contrast to BCQ, which perturbs VAE-sampled actions, MCRQ refines critic updates directly, avoiding a perturbation model. Unlike TD3_BC, which relies solely on a behavior cloning penalty, MCRQ explicitly reduces the gap between the learned and true Q-functions. Compared to CQL, which strictly minimizes the maximum Q-value of the target policy while constraining it close to the behavior policy, MCRQ strikes a balance between conservatism and value learning through TD-guided updates. MCRQ differs significantly from IQL, which computes TD target using in-sample action from the dataset via expectile regression.

VI. Experiments

To evaluate the performance of MCRQ, we conduct experiments using datasets for deep data-driven RL (D4RL) [32] on three MuJoCo [33] benchmark tasks from OpenAI Gym [34]: HalfCheetah, Hopper, and Walker2d. This benchmark includes five distinct dataset categories: random (r), medium (m), medium-replay (m-r), medium-expert (m-e), and expert (e), resulting in a comprehensive set of 15 datasets. We compare MCRQ against several recent and competitive baseline algorithms, including BEAR [22], UWAC [23], BC, CDC [12], AWAC [24], BCQ [21], OneStep [13], TD3_BC [14], CQL [15], IQL [17], and PBRL [18]. These methods represent a diverse set of paradigms in offline RL.

The experimental results for BCQ, TD3_BC, CQL, and IQL are obtained using the original implementations provided by the respective authors, and the implementation of BC is adapted from the TD3_BC codebase. Additionally, the results for BEAR, UWAC, AWAC, and OneStep are taken from Table 1 in [28]. The results for CDC and PBRL are sourced from Table 1 in their respective original papers [12] and [18]. The hyperparameter settings for MCRQ follow those used in TD3_BC, except for υ, ω, and α. The values of these parameters are listed in Table I.

TABLE I Hyperparameter setup.

Dataset	HalfCheetah			Hopper			Walker2d		
	υ	ω	α	υ	ω	α	υ	ω	α
r	0.0	2.5	25.0	0.0	0.0	20.0	0.3	2.0	15.0
m	0.0	0.0	25.0	0.0	2.0	10.0	0.0	1.0	5.0
m-r	0.1	0.0	25.0	0.0	1.0	20.0	0.0	2.0	10.0
m-e	0.2	2.0	2.5	0.0	2.0	2.5	0.0	1.0	5.0
e	0.2	0.5	2.5	0.3	1.5	2.5	0.0	2.5	5.0

A. Results on D4RL Datasets

We first evaluate the performance of MCRQ by comparing it with baseline algorithms on the D4RL datasets. Experiments were conducted using five random seeds and network initializations to ensure fair comparisons. Default environment settings were used to maintain a level playing field. Each dataset was run for 1 million steps in the offline RL setting. To assess policy performance, online evaluations were performed every 5,000 steps by interacting with the environment to collect real-time rewards. During each evaluation, the agent was tested over 10 episodes with different random seeds, and the average reward across these episodes was reported as the final score.

The experimental results are shown in Fig. 1, which compares BC, BCQ, TD3_BC, CQL, IQL, and MCRQ. Solid curves represent the mean performance across evaluations, while shaded regions indicate one standard deviation over five runs. A sliding window of five was applied to smooth the curves. Table II presents the normalized average scores from the last ten evaluations, comparing baselines with MCRQ. Row-wise normalization is applied so that the 15 scores for each algorithm are scaled between 0 and 1. We summarize our key observations below.

1) Random datasets: MCRQ outperforms most algorithms by a significant margin, and by a slight margin compared to CDC on halfcheetah-random, PBRL on hopper-random, and BEAR on walker2d-random. As shown in Fig. 1, MCRQ demonstrates rapid improvement and ultimately achieves the highest performance in (a) halfcheetah-random and (f) hopper-random. While PBRL and CDC use critic ensembles for uncertainty estimation, MCRQ relies on double critics, offering substantially higher computational efficiency than PBRL.

2) Medium datasets: MCRQ outperforms all algorithms on halfcheetah-medium and hopper-medium, with slightly lower performance than PBRL on walker2d-medium. As shown

TABLE II Normalized average scores for MCRQ and baseline algorithms.

Task Name	BEAR	UWAC	BC	CDC	AWAC	BCQ	OneStep	TD3_BC	CQL	IQL	PBRL	MCRQ
halfcheetah-r	0.00	0.00	0.00	0.99	0.15	0.00	0.00	0.35	0.55	0.45	0.35	1.00
hopper-r	0.05	0.00	0.00	0.49	0.26	0.20	0.12	0.22	0.23	0.19	0.97	1.00
walker2d-r	0.95	0.14	0.09	0.53	0.00	0.36	0.51	0.11	0.30	0.27	0.60	1.00
halfcheetah-m	0.04	0.00	0.02	0.21	0.31	0.29	0.45	0.34	0.25	0.33	0.86	1.00
hopper-m	0.02	0.00	0.07	0.22	0.21	0.12	0.85	0.20	0.28	0.23	0.57	1.00
walker2d-m	0.00	0.84	0.74	0.92	0.93	0.85	0.95	0.94	0.91	0.91	1.00	0.95
halfcheetah-m-r	0.05	0.03	0.00	0.56	0.57	0.21	0.44	0.57	0.60	0.54	0.58	1.00
hopper-m-r	0.36	0.00	0.04	0.41	0.59	0.42	0.97	0.57	0.85	0.73	1.00	0.82
walker2d-m-r	0.00	0.20	0.15	0.20	0.87	0.48	0.67	0.92	0.89	0.88	0.86	1.00
halfcheetah-m-e	0.06	0.00	0.35	0.33	0.43	1.00	0.63	0.97	0.96	0.99	0.97	1.00
hopper-m-e	0.09	0.00	0.11	0.64	0.84	0.89	0.97	0.85	0.79	0.83	1.00	0.81
walker2d-m-e	0.00	0.83	0.85	0.54	0.98	0.97	0.99	0.98	0.95	1.00	0.98	0.99
halfcheetah-e	0.71	0.73	0.72	0.03	0.00	0.80	0.42	1.00	0.93	0.97	0.69	0.98
hopper-e	0.00	1.00	0.99	0.86	0.98	0.87	0.94	1.00	0.97	0.96	1.00	0.97
walker2d-e	0.75	0.83	0.83	0.00	0.89	0.82	0.92	0.89	0.87	1.00	0.82	0.93

in Fig. 1, MCRQ exhibits rapid performance gains in (b) halfcheetah-medium and (g) hopper-medium, ultimately maintaining the highest performance among all methods.

3) Medium-replay datasets: MCRQ achieves the best performance on halfcheetah-medium-replay and walker2d-medium-replay. On hopper-medium-replay, its performance is comparable to IQL but lower than OneStep, CQL, and PBRL. As shown in Fig. 1, MCRQ shows rapid improvement in (c) halfcheetah-medium-replay and ultimately maintains the highest score among all algorithms.

4) Medium-expert datasets: MCRQ achieves performance comparable to BCQ on halfcheetah-medium-expert and performs competitively on hopper-medium-expert and walker2d-medium-expert.

5) Expert datasets: On halfcheetah-expert, MCRQ ranks second, just behind TD3_BC. On hopper-expert, its performance is slightly lower than UWAC, TD3_BC, and PBRL. On walker2d-expert, MCRQ performs slightly below IQL.

Fig. 2 presents a box plot based on 15 normalized average scores for each algorithm, derived from Table II. In this figure, orange lines denote the mean, with the upper quartile defined as the mean plus the variance and the lower quartile as the mean minus the variance. The horizontal lines at the bottom and top correspond to the minimum and maximum normalized average scores, respectively, while the whiskers extend to these extremes. It statistically shows that MCRQ achieves the highest mean among all evaluated algorithms, along with the narrowest box, indicating the smallest variance. Furthermore, MCRQ exhibits the shortest whiskers and the highest minimum score, demonstrating its strong competitiveness.

B. KL Divergence Comparison

KL divergence is used to evaluate the similarity between the target policy and the behavior policy, with a lower value indicating closer alignment. We compare the KL divergence of BCQ, TD3_BC, CQL, IQL, and MCRQ across five halfcheetah datasets. The results are shown in Fig. 3 and Table III, where the table reports the average of the last ten KL divergence values.

TABLE III Comparison of KL divergence in Fig. 3.

Algorithm	BCQ	TD3_BC	CQL	IQL	MCRQ
halfcheetah-r	12.48	6.78	0.56	3.51	2.10
halfcheetah-m	0.11	0.03	0.03	0.03	0.07
halfcheetah-m-r	0.11	0.16	0.06	0.11	0.16
halfcheetah-m-e	0.77	0.03	0.03	0.03	0.03
halfcheetah-e	0.16	0.06	0.03	0.05	0.06

In halfcheetah-random, CQL appears most conservative, while MCRQ exhibits a lower KL divergence than BCQ, TD3_BC, and IQL. This suggests that MCRQ better aligns with the low-quality behavior data, avoiding excessive deviation from the behavior policy. In contrast, BCQ shows a significantly higher KL divergence, which corresponds with its poor performance (0.00) on halfcheetah-random, as seen in Table II. For the remaining datasets, all algorithms maintain low KL divergence, indicating that their target policies remain closely aligned with the behavior policy.

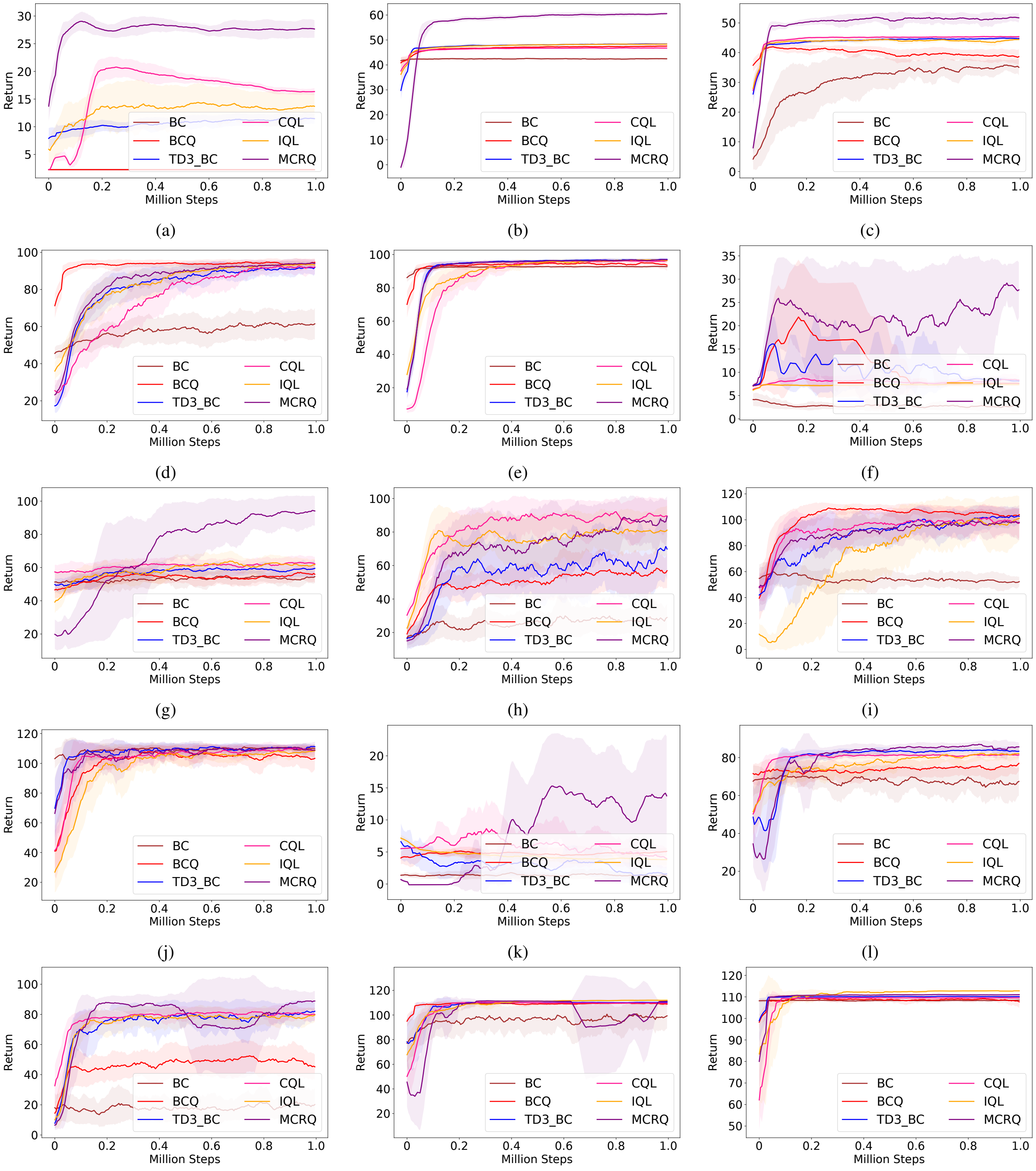
C. Ablation Experiments

We conduct ablation experiments on halfcheetah-medium and halfcheetah-expert, as shown in Fig. 4. The corresponding analysis and discussion are presented below.

1) Impact of ω: As ω increases, MCRQ becomes more conservative, causing the target policy's actions to more closely align with those in the offline dataset. On the selected datasets, MCRQ's performance consistently declines when ω is set to {2.0, 2.5}, regardless of variations in α and υ.

2) Impact of υ: When α and ω are fixed, MCRQ exhibits relatively stable performance across different values of υ in halfcheetah-medium, indicating that changes in υ have minimal impact on performance. In contrast, on halfcheetah-expert, MCRQ is more sensitive to υ when α is set to {10.0, 15.0, 20.0, 25.0}.

3) Impact of α: In subfigures (a)-(f), when υ and ω are fixed, with $\omega \in \{0.0, 0.5, 1.0, 1.5\}$, the performance of MCRQ consistently improves as α increases. However, in subfigures (g)-(l), MCRQ becomes more sensitive to changes in α. Notably, when $\alpha \in \{10.0, 15.0, 20.0, 25.0\}$, the performance deteriorates significantly, regardless of the values of υ and ω.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

(m)

(n)

(o)

Fig. 1. Normalized score curves across D4RL benchmarks. (a) halfcheetah-random-v2, (b) halfcheetah-medium-v2, (c) halfcheetah-medium-replay-v2, (d) halfcheetah-medium-expert-v2, (e) halfcheetah-expert-v2, (f) hopper-random-v2, (g) hopper-medium-v2, (h) hopper-medium-replay-v2, (i) hopper-medium-expert-v2, (j) hopper-expert-v2, (k) walker2d-random-v2, (l) walker2d-medium-v2, (m) walker2d-medium-replay-v2, (n) walker2d-medium-expert-v2, and (o) walker2d-expert-v2.

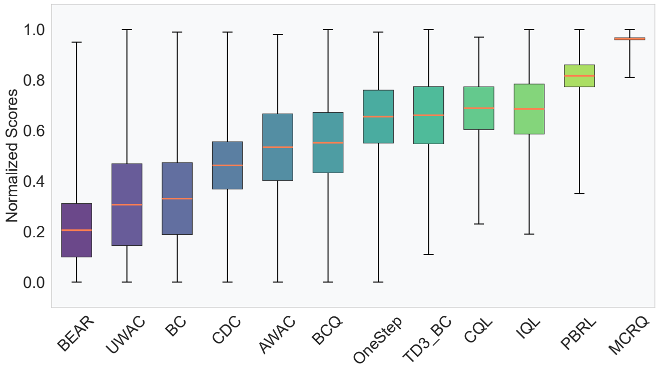


Fig. 2. Box plot with minimum, maximum, mean, and variance of 15 normalized average scores for MCRQ and baseline algorithms.

D. Comparative Visualization of OOD Actions

To evaluate the ability to constrain OOD actions, we compare the action distributions generated by the learned policies of different algorithms with the action distribution of the offline dataset. Specifically, we train BCQ, TD3_BC, and MCRQ, and collect 51,200 samples by running each trained policy separately. For comparison, we also randomly sample 51,200 actions from the offline dataset. Fig. 5 (a)-(c) show the action distributions in halfcheetah-random with t-SNE, while Fig. 5 (d)-(f) depict the distributions in halfcheetah-medium-replay. The corresponding analysis and discussion are provided below.

BCQ generates almost entirely OOD actions, indicating poor adaptation to halfcheetah-random. TD3_BC also produces predominantly OOD actions, with samples clustering unilaterally relative to those in the dataset. In contrast, MCRQ more closely aligns its action distribution with the dataset, though slight deviations from the distributional centroid remain. This suggests that while MCRQ outperforms BCQ and TD3_BC in approximating the dataset distribution, it does not fully replicate it. Notably, since the random dataset is generated by stochastic policies containing both low and high quality actions with inherent uncertainty, strict adherence to its distribution may lead to suboptimal performance. On halfcheetah-medium-replay, all three algorithms align well with the offline dataset.

E. Comparison with SOTA Algorithms

We further compare MCRQ with recent SOTA algorithms to demonstrate its superiority. The selected algorithms include f-DVL [6], CGDT [35], DD [36], DStitch [37], ODC [38], CSVE [39], ACT [40], MISA [41], O-DICE [42], MOAC [29], and ORL-RC [11]. These algorithms incorporate various design features, including critic regularization, imitation learning, decision transformers, and diffusion models.

We selected the medium, medium-replay, and medium-expert datasets to evaluate performance across various tasks. The results for each algorithm are sourced from their original papers. The same row-wise normalization used in Table II is also applied in Table IV. As shown in Table IV, MCRQ does not achieve the best performance on every dataset. However, Fig. 6 shows that MCRQ achieves the highest mean performance among all compared SOTA algorithms, using the same visualization format as Fig. 2. In contrast, f-DVL exhibits the smallest variance but the lowest mean. It is reasonable to observe that different algorithms exhibit varying strengths and weaknesses across environments; thus, the choice of algorithm should be tailored to the specific environment.

F. Computational Efficiency

Table V presents the training times of BCQ, TD3_BC, CQL, IQL, and MCRQ on halfcheetah datasets. All experiments are conducted on a machine equipped with a single AMD EPYC 9004 series processor, 384 GB DDR5 RAM, and two NVIDIA RTX 4090 GPUs (each with 24 GB VRAM), running Ubuntu 20.04. MCRQ has a slightly longer training time than TD3_BC. However, compared to BCQ, CQL, and IQL, it significantly improves training efficiency by minimizing computational overhead while maintaining strong performance.

VII. Conclusion

In this paper, we proposed MCRE to address distribution shift and OOD actions in offline RL. Our theoretical analysis shows that MCRE converges in both the presence and absence of sampling error. Moreover, the estimated Q-function, state value function, and the resulting suboptimal policy are all shown to be controllable under both conditions. Building on MCRE, we introduced MCRQ, an effective offline RL algorithm, and provided detailed implementation and experimental results on the D4RL benchmark. The experiments demonstrate that MCRQ outperforms both baseline and SOTA offline RL algorithms. Future work will explore extending the MCRE framework to other RL paradigms.

References

[1] S. Lange, T. Gabel, and M. Riedmiller, “Batch reinforcement learning,” in *Reinforcement Learning: State-of-the-Art*. Berlin, Germany: Springer, 2012, pp. 45–73.

[2] J. Hu, X. Li, X. Li, Z. Hou, and Z. Zhang, “Optimizing reinforcement learning for large action spaces via generative models: Battery pattern selection,” *Pattern Recognit*, vol. 160, p. 111194, Apr. 2025.

[3] S. Liu, Y. Liu, L. Hu, Z. Zhou, Y. Xie, Z. Zhao, W. Li, and Z. Gan, “Diffskill: Improving reinforcement learning through diffusion-based skill denoiser for robotic manipulation,” *Knowl Based Syst*, vol. 300, p. 112190, Sep. 2024.

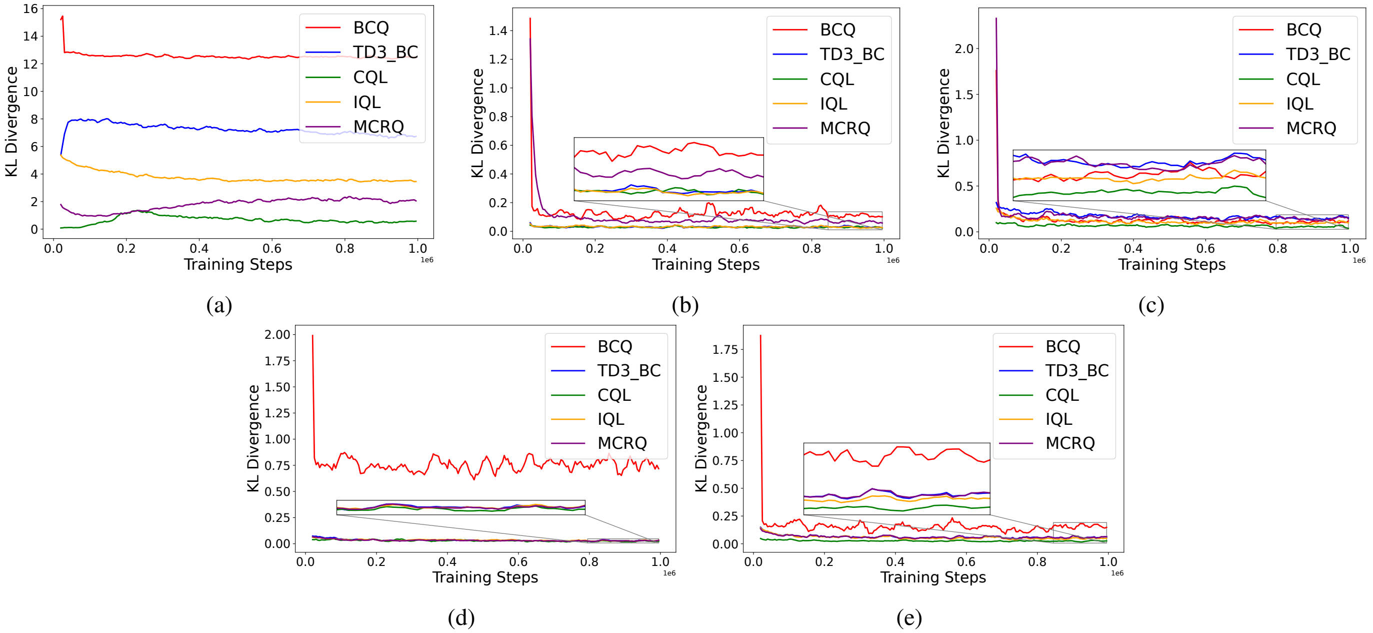
[4] R. Zhang, T. Yu, Y. Shen, and H. Jin, “Text-based interactive recommendation via offline reinforcement learning,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 10, 2022, pp. 11 694–11 702.

[5] S. Rezaeifar, R. Dadashi, N. Vieillard, L. Hussenot, O. Bachem, O. Pietquin, and M. Geist, “Offline reinforcement learning as anti-exploration,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 7, 2022, pp. 8106–8114.

[6] H. Sikchi, Q. Zheng, A. Zhang, and S. Niekum, “Dual RL: Unification and new methods for reinforcement and imitation learning,” in *Proc. Int. Conf. Learn. Represent.*, 2024.

[7] D. Zhang, B. Lyu, S. Qiu, M. Kolar, and T. Zhang, “Pessimism meets risk: Risk-sensitive offline reinforcement learning,” in *Proc. Int. Conf. Mach. Learn.*, vol. 235, 2024, pp. 59 459–59 489.

[8] J. Rao, C. Wang, M. Liu, J. Lei, and W. Giernacki, “Isfors-mix: Multi-agent reinforcement learning with importance-sampling-free off-policy learning and regularized-softmax mixing network,” *Knowl Based Syst*, vol. 309, p. 112881, Jan. 2025.



(a)

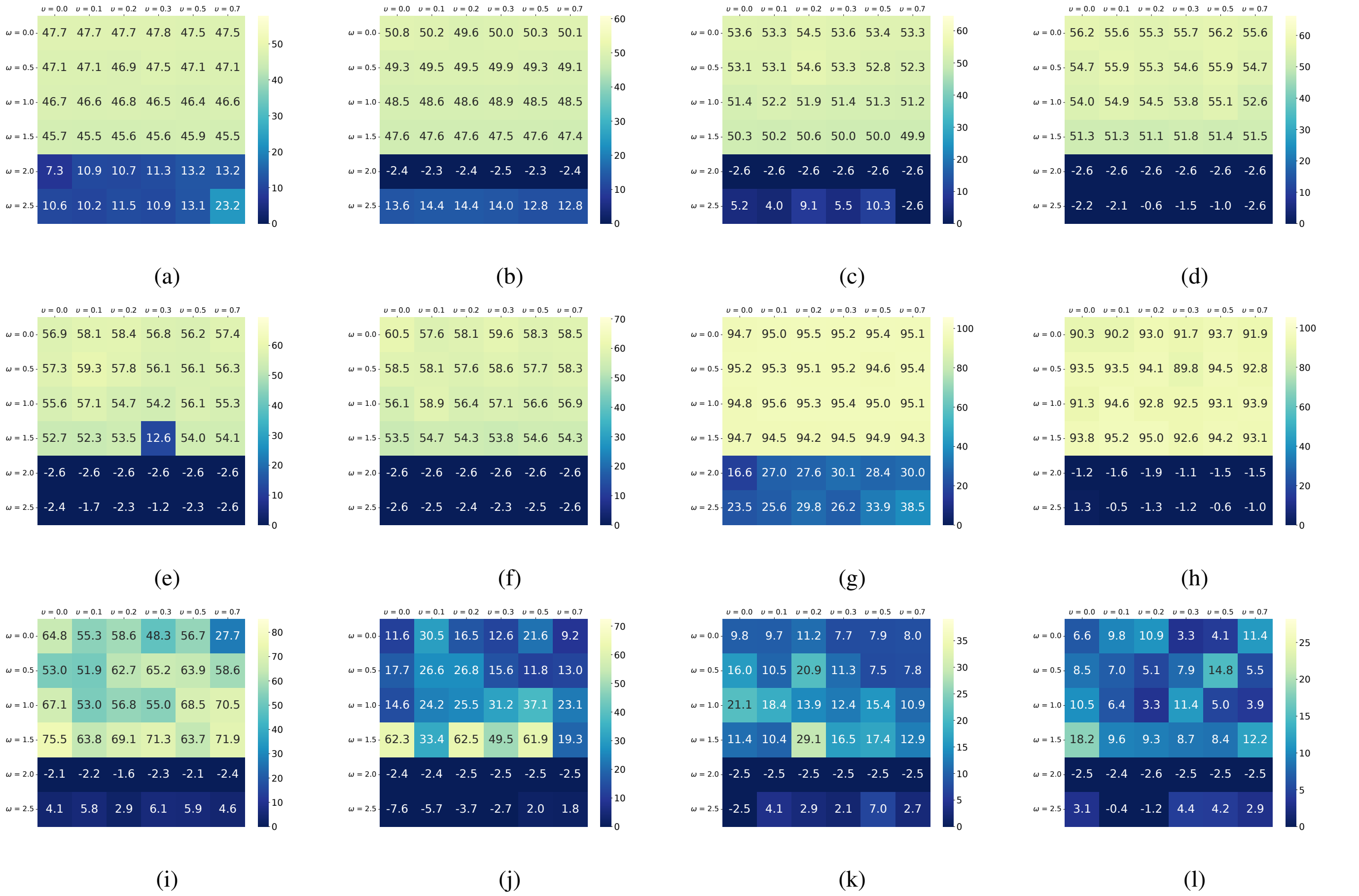
(b)

(c)

(d)

(e)

Fig. 3. Comparison of KL divergence across five halfcheetah datasets. (a) halfcheetah-random-v2, (b) halfcheetah-medium-v2, (c) halfcheetah-medium-replay-v2, (d) halfcheetah-medium-expert-v2, and (e) halfcheetah-expert-v2.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

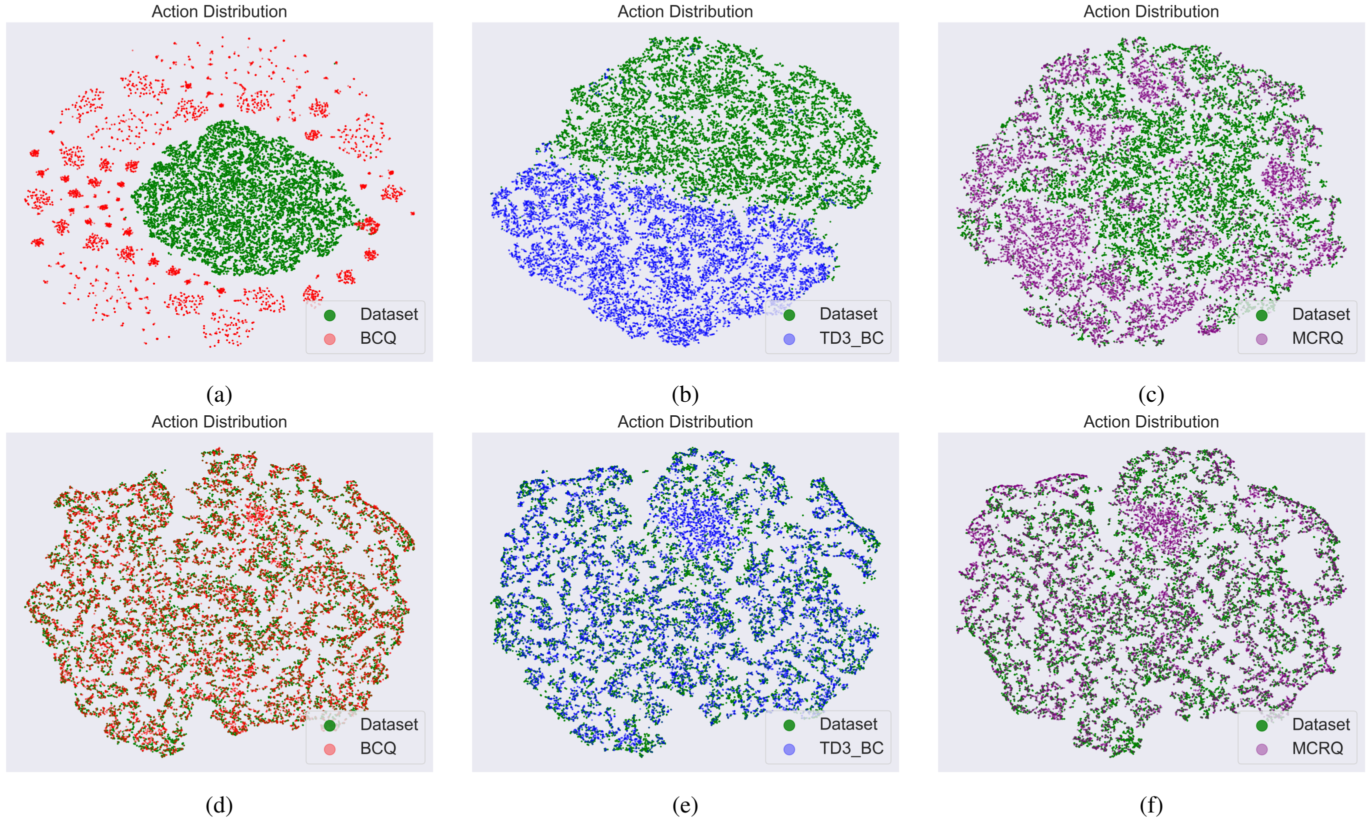
(i)

(j)

(k)

(l)

Fig. 4. Ablation study on υ, ω, and α in MCRQ conducted on halfcheetah-medium and halfcheetah-expert. Subfigures (a)-(f) show the results on halfcheetah-medium with α values of {2.5, 5.0, 10.0, 15.0, 20.0, 25.0}, while (g)-(l) present the corresponding results on halfcheetah-expert using the same set of α values. All values in the figure represent the mean performance.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Policy action visualization. (a), (b), and (c) show the action outputs of BCQ, TD3_BC, and MCRQ compared with the actions within the halfcheetah-random. (d), (e), and (f) show the action outputs of the same algorithms compared with the actions within the halfcheetah-medium-replay. The points represent the action outputs projected onto a 2-D plane using the t-SNE algorithm.

TABLE IV Normalized average scores for for MCRQ and SOTA algorithms.

Algorithm	f-DVL	CGDT	DD	DStitch	ODC	CSVE	ACT	MISA	O-DICE	MOAC	ORL-RC	MCRQ
halfcheetah-m	0.26	0.00	0.35	0.42	0.03	0.31	0.35	0.25	0.25	0.65	0.70	1.00
hopper-m	0.10	1.00	0.52	0.00	0.91	0.99	0.20	0.19	0.70	0.63	0.60	0.92
walker2d-m	0.22	0.00	0.36	0.46	0.15	0.44	0.19	0.53	0.62	0.81	1.00	0.68
halfcheetah-m-r	0.36	0.07	0.00	0.36	0.20	1.00	0.24	0.41	0.31	0.72	0.73	0.84
hopper-m-r	0.90	0.68	1.00	0.00	0.71	0.59	0.92	0.93	0.95	0.90	0.88	0.35
walker2d-m-r	0.37	0.65	0.56	0.99	0.75	0.65	0.00	0.89	0.81	1.00	0.97	0.96
halfcheetah-m-e	0.46	0.73	0.40	0.99	0.86	0.67	1.00	0.85	0.68	0.02	0.00	0.77
hopper-m-e	0.00	0.73	0.95	0.71	0.94	0.04	0.93	0.85	0.90	0.45	0.43	0.26
walker2d-m-e	0.18	0.11	0.00	0.31	0.02	0.04	1.00	0.13	0.44	0.58	0.67	0.49

TABLE V Comparison of training time (single seed run).

Algorithm	BCQ	TD3_BC	CQL	IQL	MCRQ
Time (hours)	21.09	12.85	34.21	19.31	14.05

[9] Q. Yang, S. Wang, Q. Zhang, G. Huang, and S. Song, "Hundreds guide millions: Adaptive offline reinforcement learning with expert guidance," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16 288–16 300, Nov. 2024.

[10] J. Lyu, X. Ma, X. Li, and Z. Lu, "Mildly conservative q-learning for offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 1711–1724.

[11] L. Huang, B. Dong, and W. Zhang, "Efficient offline reinforcement learning with relaxed conservatism," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5260–5272, Aug. 2024.

[12] R. Fakoor, J. W. Mueller, K. Asadi, P. Chaudhari, and A. J. Smola, "Continuous doubly constrained batch reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 11 260–11 273.

[13] D. Brandfonbrener, W. Whitney, R. Ranganath, and J. Bruna, "Offline rl without off-policy evaluation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 4933–4946.

[14] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20 132–20 145.

[15] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1179–1191.

[16] I. Kostrikov, R. Fergus, J. Tompson, and O. Nachum, "Offline reinforcement learning with fisher divergence critic regularization," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 5774–5783.

[17] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv preprint arXiv:2110.06169*, 2021.

[18] C. Bai, L. Wang, Z. Yang, Z.-H. Deng, A. Garg, P. Liu, and Z. Wang, "Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[19] L. Huang, B. Dong, W. Xie, and W. Zhang, "Offline reinforcement

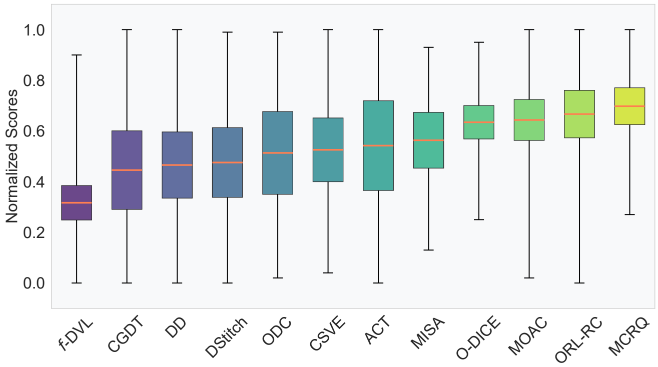


Fig. 6. Box plot with minimum, maximum, mean, and variance of 9 normalized average scores for MCRQ and SOTA algorithms.

learning with behavior value regularization," *IEEE Trans. Cybern.*, vol. 54, no. 6, pp. 3692–3704, Jun. 2024.

[20] Y. Ran, Y.-C. Li, F. Zhang, Z. Zhang, and Y. Yu, "Policy regularization with dataset constraint for offline reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 28 701–28 717.

[21] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 2052–2062.

[22] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy q-learning via bootstrapping error reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[23] Y. Wu, S. Zhai, N. Srivastava, J. M. Susskind, J. Zhang, R. Salakhutdinov, and H. Goh, "Uncertainty weighted actor-critic for offline reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 11 319–11 328.

[24] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[25] D. Tarasov, V. Kurenkov, A. Nikulin, and S. Kolesnikov, "Revisiting the minimalist approach to offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 11 592–11 620.

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[27] S. Zhao, *Mathematical Foundations of Reinforcement Learning*. Springer Nature Press and Tsinghua University Press, 2025.

[28] Y. Mao, H. Zhang, C. Chen, Y. Xu, and X. Ji, "Supported value regularization for offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 40 587–40 609.

[29] L. Huang, B. Dong, J. Lu, and W. Zhang, "Mild policy evaluation for offline actor–critic," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17 950–17 964, Dec. 2024.

[30] D. A. Levin and Y. Peres, *Markov chains and mixing times*. American Mathematical Soc., 2017, vol. 107.

[31] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1–15.

[32] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.

[33] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control." in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, Oct. 2012, pp. 5026–5033.

[34] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[35] Y. Wang, C. Yang, Y. Wen, Y. Liu, and Y. Qiao, "Critic-guided decision transformer for offline reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 14, 2024, pp. 15 706–15 714.

[36] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision making?" in *Proc. Int. Conf. Learn. Represent.*, 2023.

[37] G. Li, Y. Shan, Z. Zhu, T. Long, and W. Zhang, "Diffstitch: Boosting offline reinforcement learning with diffusion-based trajectory stitching," in *Proc. Int. Conf. Mach. Learn.*, vol. 235, 2024, pp. 28 597–28 609.

[38] J. Kim, S. Lee, W. Kim, and Y. Sung, "Decision convformer: Local filtering in metaformer is sufficient for decision making," in *Proc. Int. Conf. Learn. Represent.*, 2024.

[39] L. Chen, J. Yan, Z. Shao, L. Wang, Q. Lin, S. Rajmohan, T. Moscibroda, and D. Zhang, "Conservative state value estimation for offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 35 064–35 083.

[40] C.-X. Gao, C. Wu, M. Cao, R. Kong, Z. Zhang, and Y. Yu, "Act: Empowering decision transformer with dynamic programming via advantage conditioning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 11, 2024, pp. 12 127–12 135.

[41] X. Ma, B. Kang, Z. Xu, M. Lin, and S. Yan, "Mutual information regularized offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 19 058–19 072.

[42] L. Mao, H. Xu, W. Zhang, and X. Zhan, "ODICE: Revealing the mystery of distribution correction estimation via orthogonal-gradient update," in *Proc. Int. Conf. Learn. Represent.*, 2024.