

False Correlation Reduction for Offline Reinforcement Learning

Zhihong Deng, Zuyue Fu, Lingxiao Wang, Zhuoran Yang,
Chenjia Bai, Tianyi Zhou, Zhaoran Wang, Jing Jiang

Abstract—Offline reinforcement learning (RL) harnesses the power of massive datasets for resolving sequential decision problems. Most existing papers only discuss defending against out-of-distribution (OOD) actions while we investigate a broader issue, the false correlations between epistemic uncertainty and decision-making, an essential factor that causes suboptimality. In this paper, we propose falSe COrrelation REduction (SCORE) for offline RL, a practically effective and theoretically provable algorithm. We empirically show that SCORE achieves the SoTA performance with 3.1x acceleration on various tasks in a standard benchmark (D4RL). The proposed algorithm introduces an annealing behavior cloning regularizer to help produce a high-quality estimation of uncertainty which is critical for eliminating false correlations from suboptimality. Theoretically, we justify the rationality of the proposed method and prove its convergence to the optimal policy with a sublinear rate under mild assumptions.

Index Terms—Offline reinforcement learning, false correlation, uncertainty estimation.

1 INTRODUCTION

OFFLINE reinforcement learning (RL) aims to learn the optimal policy from a pre-collected dataset without interacting with the environment. Theoretically, experience replay allows for the direct application of off-policy RL algorithms to this setting, but they perform poorly in practice [1], [2]. Many research works attribute this problem to out-of-distribution (OOD) actions [1], [3], [4]. Because the offline setting disallows learning by trial and error, an agent may "exploit" OOD actions to attack the value estimator, resulting in a highly suboptimal learned policy. Although this provides an intuitive explanation, the relationship between OOD actions and the suboptimality of the learned policy is ambiguous and lacks a mathematical description.

As opposed to this, false correlation is a rigorously defined concept by mathematically decomposing suboptimality. Due to the insufficient data coverage, a correlation exhibits between epistemic uncertainty and decision-making. As a result, the agent is biased towards suboptimal policies that look good only by chance. Such false correlations can be generated not only by OOD actions but also by insufficient coverage over the state space (or,

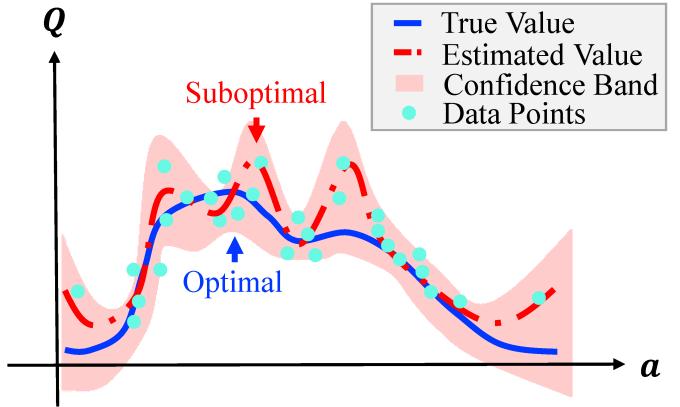


Figure 1. An example of false correlation: the epistemic uncertainty is correlated with the value, making a suboptimal action with high uncertainty appear to be better than the optimal one.

• Z. Deng, J. Jiang are with Centre for Artificial Intelligence, FEIT, University of Technology Sydney, NSW 2007, Australia (Email: zhihong.deng@student.uts.edu.au, jing.jiang@uts.edu.au).

• Z. Fu, L. Wang, Z. Wang are with the Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, USA (zuyuefu2022@u.northwestern.edu, lingxiaowang2022@u.northwestern.edu, zhaoran-wang@gmail.com).

• Z. Yang is with the Department of Statistics and Data Science, Yale University, 24 Hillhouse Avenue, New Haven, CT 06511, USA (zryang1993@gmail.com).

• C. Bai is with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China (e-mail: baichenjia@pjlab.org.cn).

• T. Zhou is with the Department of Computer Science and UMIACS, University of Maryland, College Park, MD 20742, USA (Email: tianyi.david.zhou@gmail.com).

Manuscript received November xx, 2022; revised xx xx, xxxx.

equivalently, OOD states). Moreover, in-distribution samples have differences in uncertainty, and those with higher uncertainty may subtly raise suboptimality when the agent greedily pursues the maximum estimated value.

According to recent theoretical studies, pessimism in the face of uncertainty can solve this problem [5], [6]. An example of false correlation is shown in Figure 1, where the estimated value function encourages the agent to pick a suboptimal action while the lower confidence bound (pessimism) recovers the optimal one. Unfortunately, this approach fails in practice [1], [7] due to the inability to obtain high-quality estimations of uncertainty.

In this paper, we propose falSe COrrelation REduction (SCORE) for offline RL, a practically effective and theoretically provable algorithm. Specifically, SCORE introduces an annealing behavioral cloning regularizer on top of pessimism in the face of uncertainty. This regularizer drives the agent to concentrate on the dataset distribution during the initial stages of training, producing

high-quality uncertainty estimates. In the later phases, SCORE gradually decays the regularization weight towards zero, avoiding the bias towards the behavioral policy. On the theoretical side, we generalize the conclusion in [5] to the context of infinite-horizon regularized MDP and incorporate the policy optimization process, showing SCORE converges to the optimal policy at a sublinear rate. Importantly, this result does not rely on strong assumptions regarding access to the exact value of OOD samples [8] or sufficient coverage over the sample space [9], [10]. Furthermore, SCORE eliminates the need to sample and calculate the target value of OOD data, which distinguishes it from the approach presented in [8].

Through extensive experiments on the D4RL benchmark, we demonstrate that SCORE is not only theoretically sound but also obtains promising empirical results across various data settings. Furthermore, the simplicity of SCORE brings approximately a 3.1x speedup over the previous SoTA method.

2 PRELIMINARIES

In this section, we first formalize the offline RL problem, and analyze the performance gap between the optimality policy and a learned policy. Then we show how pessimism can eliminate false correlations from the suboptimality.

2.1 The Offline RL Problem

Consider a MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$, where \mathcal{S} and \mathcal{A} represent the state space and the action space respectively. $P: \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is the Markov transition function, $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, and $d_0: \mathcal{S} \rightarrow [0, 1]$ is the initial distribution of states. In offline RL, the agent is given a static dataset $\mathcal{D} = \{(s^{(i)}, a^{(i)}, s'^{(i)}, r^{(i)})\}_{i=1}^N$ collected by the behavioral policy π_β . Suppose that $d^\pi(s, a)$ denotes the discounted state-action distribution of a policy π . We have $(s^{(i)}, a^{(i)}) \sim d^{\pi_\beta}(\cdot, \cdot)$, $s'^{(i)} \sim P(\cdot | s^{(i)}, a^{(i)})$, and $r^{(i)} = R(s^{(i)}, a^{(i)})$. Then the goal of offline RL is to search for a policy $\pi: \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ that maximizes the expected cumulative reward $\mathcal{J}(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t)]$ given a static dataset \mathcal{D} , where expectation $\mathbb{E}_\pi[\cdot]$ is taken with respect to $s_0 \sim d_0(\cdot)$, $a_t \sim \pi(\cdot | s_t)$, and $s_{t+1} \sim P(\cdot | s_t, a_t)$. With a slight abuse of notation, we refer to \mathcal{D} as the dataset distribution.

2.2 Suboptimality Decomposition

In offline RL, the samples are drawn from a fixed distribution \mathcal{D} instead of the environment. Therefore, in value iteration, the standard Bellman optimality operator \mathcal{B} gets replaced by its empirical counterpart $\widehat{\mathcal{B}}$ where the transition probabilities and rewards are estimated by the sample average in \mathcal{D} . Since the dataset only covers partial information of the environment, the agent would be learning with bias. In this paper, we formalize such bias for any action-value function $Q: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ as follows:

$$\iota(s, a) = \mathcal{B}Q(s, a) - \widehat{\mathcal{B}}Q(s, a). \quad (1)$$

Since $\iota(s, a)$ characterizes the error arising from insufficient information about the environment in knowledge and gradually converges to zero as we learn more about the state-action pair (s, a) (including its long-term effects), we refer to it as the *epistemic error*. In the ideal case, the dataset accurately mirrors the environment, i.e., $\widehat{\mathcal{B}} = \mathcal{B}$, resulting in zero epistemic error. The agent can learn the optimal policy offline just like in the

online setting. However, this is almost impossible in real-world domains. In general, the dataset only contains limited information and the epistemic error persists throughout the learning process.

According to [5], the suboptimality of a learned policy $\widehat{\pi}$, i.e., the performance gap between $\widehat{\pi}$ and the optimal policy π^* , can be decomposed mathematically into three different components:

$$\begin{aligned} \text{SubOpt}(\widehat{\pi}; s_0) &= V^{\pi^*}(s_0) - V^{\widehat{\pi}}(s_0) \\ &= - \underbrace{\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\widehat{\pi}} [\iota(s_t, a_t) | s_0]}_{\text{(i): False Correlation}} + \underbrace{\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi^*} [\iota(s_t, a_t) | s_0]}_{\text{(ii): Intrinsic Uncertainty}} \\ &\quad + \underbrace{\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi^*} [\langle \widehat{Q}(s_t, \cdot), \pi^*(\cdot | s_t) - \widehat{\pi}(\cdot | s_t) \rangle_A | s_0]}_{\text{(iii): Optimization Error}}, \end{aligned} \quad (2)$$

where s_0 is the initial state, \widehat{Q} is an estimated Q function, and $V^\pi(s) = \langle Q^\pi(s, \cdot), \pi(\cdot | s) \rangle$ is the state-value function. It is straightforward that the suboptimality of the optimal policy π^* is zero and lower indicates a better policy. Term (ii) in equation 2 is proved to arise from the information-theoretic lower bound and thus is impossible to eliminate. Meanwhile, term (iii) is non-positive as long as the policy $\widehat{\pi}$ is greedy with respect to the estimated action-value function \widehat{Q} . Therefore, controlling term (i) is the key to reducing suboptimality in offline RL. Next, we explain how pessimism could help address this issue.

2.3 Pessimism

Let $\widehat{Q}: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ represents an action value estimator based on the dataset. We first define an uncertainty quantifier U with confidence $\xi \in (0, 1)$ as follows.

Definition 2.1 (ξ -Uncertainty Quantifier). $U: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a ξ -uncertainty quantifier with respect to the dataset distribution \mathcal{D} if the event

$$\mathcal{E} = \left\{ |\widehat{\mathcal{B}}\widehat{Q}(s, a) - \mathcal{B}\widehat{Q}(s, a)| \leq U(s, a), \forall (s, a) \in \mathcal{S} \times \mathcal{A} \right\} \quad (3)$$

satisfies $\Pr(\mathcal{E} | \mathcal{D}) \geq 1 - \xi$.

In Definition 2.1, U measures the uncertainty arising from approximating $\mathcal{B}\widehat{Q}$ with $\widehat{\mathcal{B}}\widehat{Q}$, where \mathcal{B} is the true Bellman optimality operator and $\widehat{\mathcal{B}}$ is the empirical Bellman operator. We remark that $\widehat{\mathcal{B}}$ can be constructed implicitly by treating $\widehat{\mathcal{B}}\widehat{Q}: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ as a whole. When $\mathcal{B}\widehat{Q}$ and $\widehat{\mathcal{B}}\widehat{Q}$ differ by a large amount, U should be large, while when the two quantities are sufficiently close, U can be very small or even zero. Based on this definition, Jin et al. [5] construct a pessimistic Bellman operator as follows:

$$\widehat{\mathcal{B}}^-\widehat{Q}(s, a) := \widehat{\mathcal{B}}\widehat{Q}(s, a) - U(s, a). \quad (4)$$

According to Definition 2.1, $\widehat{\mathcal{B}}^-\widehat{Q}(s, a) \leq \mathcal{B}\widehat{Q}(s, a)$ holds for all state-action pairs with a high probability, i.e., the Q-value derived from equation 4 lower bounds the one derived from the standard Bellman operator \mathcal{B} . In other words, equation 4 provides a pessimistic estimation. Replacing the empirical Bellman operator $\widehat{\mathcal{B}}$ in equation 1 with the pessimistic Bellman operator, it holds that:

$$0 \leq \iota(s, a) = \mathcal{B}\widehat{Q}(s, a) - \widehat{\mathcal{B}}^-\widehat{Q}(s, a) \leq 2U(s, a). \quad (5)$$

Since the epistemic error $\iota(s, a)$ is non-negative in this case, term (i) in equation 2 only reduces the suboptimality. As a result,

pessimism eliminates false correlations. Meanwhile, the suboptimality is now upper-bounded by $\sum_{t=0}^{\infty} 2\gamma^t \mathbb{E}_{\pi^*}[U(s, a) | s_0]$, so the remaining issue is finding a sufficiently small ξ -uncertainty quantifier that satisfies Definition 2.1.

3 FALSE CORRELATION REDUCTION FOR OFFLINE RL

In this section, we begin with a motivating example in Section 3.1 to demonstrate the universality of false correlations. We then introduce a practical algorithm named SCORE in Section 3.2. In Section 3.3, we further analyze the theoretical properties of the proposed algorithm.

3.1 A Motivating Example

Consider an autonomous vehicle navigating through a complex urban environment, aiming to optimize its route for efficient travel. During data collection, the vehicle successfully reaches a specific location during rush hour, receiving a high return for the fast completion. However, this success was merely due to a lucky sequence of green lights rather than an optimal policy. When an RL agent learns from the dataset and strives to maximize return, it may be prone to bias towards this suboptimal route, leading to inefficient and frustrating experiences in real-world deployment. This example illustrates how false correlations can significantly influence offline learning and highlights the critical need to address this issue for practical success.

3.2 False Correlation Reduction

Based on Definition 2.1 and equation 5, it is straightforward that $U(s, a) = |\hat{\mathcal{B}}\hat{Q}(s, a) - \mathcal{B}\hat{Q}(s, a)|$ achieves the tightest bound, i.e., U accurately portrays the epistemic uncertainty arising from approximating \mathcal{B} using $\hat{\mathcal{B}}$. Note that the Bellman operator acts on the value function, so U must consider the input state-action pair's missing information in predicting its long-term effects. Besides, since the state and action spaces are enormous in real-world domains, using value function approximators (such as deep neural networks) is necessary. In this case, we can neither derive an analytical solution of the epistemic uncertainty like in linear MDPs [5] nor evaluate it by maintaining a counter or a pseudo-counter for all state-action pairs.

Estimating epistemic uncertainty with deep neural networks is an important research topic. One of the most popular approaches is the bootstrapped ensemble method [11], [12]. Each ensemble member is trained on a different version of data generated by a bootstrap sampling procedure. This approach provides a general and non-parametric way to approximate the Bayesian posterior distribution, so the standard deviation of multiple estimations can be regarded as a reasonable estimation of the uncertainty. Previous works mainly use uncertainty as a bonus to promote efficient exploration in online RL [11], [13], [14], [15], while we utilize uncertainty as a penalty to reduce false correlations. Next, we include this technique into generalized policy iteration to develop a complete RL algorithm.

Policy Evaluation. To implement the pessimistic Bellman operator practically, we maintain M independent critics $\{Q_{\theta_i}\}_{i=1}^M$ and their corresponding target networks $\{Q_{\theta'_i}\}_{i=1}^M$ in the policy

Algorithm 1 False Correlation Reduction for offline RL

```

1: Initialize critic networks  $\{Q_{\theta_i}\}_{i=1}^M$  and actor network  $\pi_\phi$ ,  

   with random parameters  $\{\theta_i\}_{i=1}^M, \phi$   

2: Initialize target networks  $\{\theta'_i\}_{i=1}^M \leftarrow \{\theta_i\}_{i=1}^M, \phi' \leftarrow \phi$   

3: Initialize replay buffer with the dataset  $\mathcal{D}$   

4: for  $t = 1$  to  $T$  do  

5:   Sample  $n$  transitions  $(s, a, s', r)$  from  $\mathcal{D}$   

6:    $a' \leftarrow \pi_{\phi'}(s') + \epsilon, \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma^2), -c, c)$   

7:   for  $i = 1$  to  $M$  do  

8:     Update  $\theta_i$  to minimize equation 6.       $\triangleright$  Pessimism  

9:   end for  

10:  if  $t \% d = 0$  then  

11:    Update  $\phi$  to maximize equation 7.  

12:    Update target networks:  

13:     $\theta'_i \leftarrow \tau\theta'_i + (1 - \tau)\theta_i, \phi' \leftarrow \tau\phi' + (1 - \tau)\phi$ .  

14:  end if  

15:  if  $t \% d_{bc} = 0$  then  

16:     $\lambda = \gamma_{bc} \cdot \lambda$   

17:  end if  

18: end for

```

evaluation step. Samples are drawn uniformly from the offline dataset, and the learning objective of each critic Q_{θ_i} is as follows,

$$\mathcal{L}(Q_{\theta_i}) = \mathbb{E}_{s, a, s', r \sim \mathcal{D}, a' \sim \pi(\cdot | s')} \left[(Q_{\theta_i}(s, a) - y_i)^2 \right], \quad (6)$$

$$y_i = r + \gamma Q_{\theta'_i}(s', a') - \beta u(s, a),$$

where β is a hyperparameter that controls the strength of the uncertainty penalty, and $u(\cdot, \cdot)$ is computed as the standard deviation of $\{Q_{\theta_i}\}_{i=1}^M$. From the Bayesian perspective, the output of the critic ensemble forms a distribution over the Q-value that approximates its posterior. As a result, the standard deviation quantifies the uncertainty in beliefs, i.e., equation 6 implements the ξ -uncertainty quantifier U in equation 4 using the bootstrapped ensemble method. Remark that, in the theoretical algorithm for linear MDPs [5], the uncertainty quantifier and the penalty coefficient are in closed forms. All samples participate in the calculation simultaneously and are used only once, which is almost impossible to implement, especially when dealing with massive datasets collected from complex dynamics. To make the proposed method compatible with large data sets, we optimize the Q networks using mini-batch gradient descent. The samples are used repeatedly in varying order. The uncertainty estimator is unstable at the early stage as it is derived from networks trained from scratch, so penalizing Q with a small quantity (controlled by β) is preferable. While most existing approaches use the smallest Q-value as the target value to avoid overestimation, equation 6 updates each critic Q_{θ_i} toward its corresponding target network $Q_{\theta'_i}$. As a result, equation 6 guarantees temporal consistency and passes the uncertainty over time [11], [13].

Policy Improvement. Designing an uncertainty-based algorithm for offline RL is a non-trivial task. We attribute the failure of previous work [1], [7], [16] to the inability to obtain high-quality uncertainty estimations, which, as pointed out in Section 2.3, is the key to addressing the false correlation issue. Empirically, the bootstrapped ensemble method alone cannot produce such estimation. To this end, we propose using the following objective

function for policy π_ϕ in the policy improvement step:

$$\mathcal{L}(\pi_\phi) = \mathbb{E}_{s,a \sim \mathcal{D}} \left[\min_i Q_{\theta_i}(s, \pi_\phi(s)) - \lambda \|\pi_\phi(s) - a\|_2^2 \right]. \quad (7)$$

The behavior cloning loss $\|\pi_\phi(s) - a\|_2^2$ functions as a regularization term. While the ensemble networks initially struggle to accurately quantify epistemic uncertainty, this term guides the policy to stay near the dataset distribution. It prevents the agent from "exploiting" OOD data to attack the critics. Meanwhile, it helps the critics to be sufficiently familiar with the dataset distribution to deliver high-quality uncertainty estimations. In particular, we gradually decrease the regularization coefficient λ during the training process. The regularizer provides a good initialization at the beginning. Later in the training process, the regularization effect becomes weaker and weaker, and the pessimistic Q-values gradually dominate the policy objective. This way, SCORE returns to a pure uncertainty-based method that reliably implements the pessimism principle, avoiding the bias towards the behavioral policy. Alternatively, we can understand this regularizer from the optimization perspective [17]. Directly maximizing the value function is challenging, and behavior cloning lowers the difficulty of the optimization problem at the early stage. As the training process proceeds, the regularization effect decreases, so the objective function gradually returns to the original problem, i.e., maximizing the pessimistic value function. The complete algorithm is summarized in Algorithm 1.

3.3 Theoretical Results

In this section, we theoretically analyze the proposed algorithm and show that it achieves a sublinear rate of convergence. We begin with the notion of regularized MDP.

Regularized MDP. For any behavior policy π_0 , based on the definition of the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$, we introduce its regularized counterpart $\mathcal{M}_\lambda = (\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0, \lambda)$, where λ is the regularization parameter. Specifically, for any policy π in \mathcal{M}_λ , the regularized state-value function V_λ^π and the regularized action-value function Q_λ^π are defined as

$$\begin{aligned} V_\lambda^\pi(s) &= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \cdot (r(s_t, a_t) \right. \\ &\quad \left. - \lambda \cdot \log(\pi(\cdot | s_t) / \pi_0(\cdot | s_t))) \mid s_0 = s \right], \\ Q_\lambda^\pi(s, a) &= r(s, a) + \gamma \cdot \mathbb{E}_{s' \sim P(\cdot | s, a)} [V_\lambda^\pi(s')], \\ &\quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}, \end{aligned}$$

respectively. We remark that such a regularization term in the definition of V_λ^π functions as a behavior cloning term. Throughout the learning process, we anneal the regularization parameter λ so that the impact of the behavior cloning term gradually decreases. Formally, for a collection of regularized MDPs $\{\mathcal{M}_{\lambda_k}\}_{k=0}^K$, we aim to minimize the suboptimality gap defined as follows,

$$\text{SubOptGap}(K) = \min_{k \in \{0, 1, \dots, K-1\}} (V_k^*(s_0) - V_k^{\pi_k}(s_0)). \quad (8)$$

Here we denote by $V_k^* = V_{\lambda_k}^{\pi_k^*}$ and $V_k^{\pi_k} = V_{\lambda_k}^{\pi_k}$ for notational convenience, where $\pi_k^* \in \arg \max_\pi \mathbb{E}_{s_0 \sim d_0} [V_{\lambda_k}^\pi(s_0)]$ is an optimal policy for \mathcal{M}_{λ_k} . In other words, equation 8 measures the suboptimality gap between the best policy π_{k^*} and the corresponding optimal policy $\pi_{k^*}^*$ under the regularized MDP $\mathcal{M}_{\lambda_{k^*}}$, where $k^* = \arg \min_{k \in \{0, 1, \dots, K-1\}} (V_k^*(s_0) - V_k^{\pi_k}(s_0))$.

Theo-SCORE. For the simplicity of discussion, we introduce a theoretical counterpart of Algorithm 1 named Theo-SCORE. At the k -th iteration of Theo-SCORE, with the estimated pessimistic Q-function Q_k , the Theo-SCORE objective for the regularized MDP \mathcal{M}_{λ_k} is defined as follows,

$$\begin{aligned} \mathcal{L}_{\text{SCORE}}^k(\pi) &= \mathbb{E}_{s \sim \mathcal{D}} [\langle Q_k(s, \cdot), \pi(\cdot | s) \rangle \\ &\quad - \lambda_k \cdot \text{KL}(\pi(\cdot | s) \| \pi_0(\cdot | s))], \end{aligned} \quad (9)$$

where \mathcal{D} is the static dataset, and the KL divergence corresponds to the behavior cloning term. In the policy improvement step of Theo-SCORE, we employ deterministic policy gradient [18] to maximize equation 9. We remark that the objective function in equation 9 is equivalent to equation 7 under Gaussian policies with the same covariance. While in the policy evaluation step of Theo-SCORE, we assume there exists an oracle that uses the ξ -uncertainty quantifier $U(s, a)$ defined in Definition 2.1 to construct a pessimistic estimator of the Q-function, which is practically achieved by equation 6, as shown in Section 3.2.

To better study the convergence of Theo-SCORE, we further introduce a helper algorithm termed offline proximal optimization (OPO). Formally, we consider the linear function parameterization in the k -th iteration as follows,

$$\begin{aligned} \pi_{\phi_k} &\propto \exp(f_{\phi_k}(s, a)), \\ f_{\phi_k}(s, a) &= \psi(s, a)^\top \phi_k, \\ Q_k(s, a) &= \theta_k(s)^\top a, \end{aligned} \quad (10)$$

where ψ and θ_k are feature vectors, and f_{ϕ_k} is the energy function. We denote by $\pi_k = \pi_{\phi_k}$ and $f_k = f_{\phi_k}$ for notational convenience. With pessimistic Q-function Q_k and current policy π_k in the k -th iteration, OPO's objective function for the regularized MDP \mathcal{M}_{λ_k} is defined as follows,

$$\begin{aligned} \mathcal{L}_{\text{OPO}}^k(\phi) &= \mathbb{E}_{s \sim \mathcal{D}} \left[\left\langle Q_k(s, \cdot) - \lambda_k \cdot \log \frac{\pi_\phi(\cdot | s)}{\pi_0(\cdot | s)}, \pi_\phi(\cdot | s) \right\rangle \right. \\ &\quad \left. - \eta_k \cdot \text{KL}(\pi_\phi(\cdot | s) \| \pi_k(\cdot | s)) \right], \end{aligned} \quad (11)$$

where π_0 is the behavior policy and η_k is the regularization parameter.

Equivalence between Theo-SCORE and OPO. Under the linear function parameterization in equation 10, we now relate the objective functions in equation 11 and equation 9 in the following lemma. For the sake of simplicity, we define $I_\phi = \mathbb{E}_{s \sim \mathcal{D}} [I_\phi(s)]$, where $I_\phi(s) = \text{Var}_{a \sim \pi_\phi(\cdot | s)} [\psi(s, a)]$.

Lemma 3.1 (Equivalence between Theo-SCORE and OPO). The stationary point ϕ_{k+1} of $\mathcal{L}_{\text{OPO}}^k(\phi)$ satisfies

$$\begin{aligned} \phi_{k+1} &= \frac{\eta_k \phi_k + \lambda_k \phi_0}{\eta_k + \lambda_k} + (\eta_k + \lambda_k)^{-1} \cdot I_{\phi_{k+1}}^{-1} \\ &\quad \cdot \mathbb{E}_{s \sim \mathcal{D}} [\nabla_a Q_k(s, \Pi_{\phi_{k+1}}(s)) \nabla_\phi \Pi_{\phi_{k+1}}(s)], \end{aligned}$$

where $\Pi_\phi(s) = \mathbb{E}_{a \sim \pi_\phi(\cdot | s)} [a]$ is the deterministic policy associated with π_ϕ .

Lemma 3.1 states that maximizing the offline proximal objective is equivalent to an implicit natural policy gradient step corresponding to the maximization of the Theo-SCORE objective. As a result, to study the convergence of Theo-SCORE, it suffices to analyze pessimistic OPO.

Convergence Analysis. For simplicity of presentation, we take the regularization parameter $\lambda_k = \alpha^k$, where $0 < \alpha < 1$ quantifies

the speed of annealing. Recall that we employ pessimism to construct estimated Q-functions Q_k at each iteration k , which ensures that there exists a ξ -uncertainty quantifier $U(s, a)$ defined in Definition 2.1. Formally, we apply the following assumption on the estimated Q-functions, which can be achieved by a bootstrapped ensemble method as shown in Section 3.2.

Assumption 3.2 (Pessimistic Q-Functions). For any $k \in [K]$, $U: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a ξ -uncertainty quantifier for the estimated Q-function Q_k , i.e., the event

$$\mathcal{E}_K = \{|\hat{\mathcal{B}}Q_k(s, a) - \mathcal{B}Q_k(s, a)| \leq U(s, a), \\ \forall (s, a, k) \in \mathcal{S} \times \mathcal{A} \times [K]\}$$

holds with probability at least $1 - \xi$.

We further define the pessimistic error as follows,

$$\varepsilon_{\text{Pess}} = \sum_{t=0}^{\infty} 2\gamma^t \cdot \mathbb{E}_{\pi^*}[U(s_t, a_t) | s_0]. \quad (12)$$

Such a pessimistic error in equation 12 quantifies the irremovable intrinsic uncertainty [5]. Now, we introduce our main theoretical result as follows.

Theorem 3.3. Suppose $\lambda_k = \alpha^k$ and $\eta_k + \lambda_k = \sqrt{\zeta/K}$ for any $k \geq 0$, where

$$\zeta = (1 + \alpha^4(1-\alpha)^{-4})^2 \\ \cdot \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}_{\pi^*}[\text{KL}(\pi^*(\cdot | s_t) \| \pi_0(\cdot | s_t)) | s_0],$$

then for OPO with estimated Q-functions satisfying Assumption 3.2, it holds with probability at least $1 - \xi$ that

$$\text{SubOptGap}(K) = O((1 - \gamma)^{-3} \sqrt{\zeta/K}) + \varepsilon_{\text{Pess}},$$

where $\varepsilon_{\text{Pess}}$ is defined in equation 12.

Theorem 3.3 states that the sequence of policies generated by pessimistic OPO converges sublinearly to an optimal policy in the regularized MDP with an additional pessimistic error term $\varepsilon_{\text{Pess}}$. We remark that such an error term $\varepsilon_{\text{Pess}}$ is irremovable, as it arises from the information-theoretic lower bound [5]. Moreover, given the equivalence between offline OPO and Theo-SCORE in Lemma 3.1, we know that Theo-SCORE also converges to an optimal policy under a sublinear rate. For the detailed proofs of the lemma and the theorem, please see Appendix A.

4 RELATED WORK

The techniques used in most existing works in offline RL algorithms fall into two categories, namely, policy-constrained methods and value-penalized methods. Policy-constrained methods defend against OOD actions by restricting the hypothesis space of the policy. For example, BCQ [1], [19] and EMaQ [20] exclusively consider actions proposed by the estimated behavioral policy. Alternatively, some methods [3], [4], [21], [22], [23] reformulate the policy optimization problem as a constrained optimization problem to keep the learned policy sufficiently close to the behavioral policy. More recently, Fujimoto et al. [24] proposes a simple yet effective solution by directly using behavioral cloning in policy optimization. In brief, policy-constraint approaches have an intuitive motivation but often struggle to outperform the behavioral

policy. While SCORE’s regularization term also adopts the policy-constraint technique, it is improved by a decaying factor that avoids an explicit reliance on the behavioral policy.

Alternatively, the value-penalized methods steer the policy towards training distribution by penalizing the value of OOD actions. For example, CQL [9] explicitly minimizes the action value of OOD data via a value regularization term. MOPO [10] and MORL [25] learn pessimistic dynamic models and use model uncertainty as the penalty. However, in a subsequent work [16], the authors claim that the model uncertainty for complex dynamics tends to be unreliable and revert to the CQL-type penalty method. Instead of explicitly learning the dynamics and then inferring its uncertainty, SCORE utilizes bootstrapped q-networks to estimate uncertainty, providing reliable estimations.

Some model-free methods also use uncertainty to construct the penalty term. For example, UWAC [23] proposes to use Monte Carlo (MC) dropout to quantify uncertainty and perform uncertainty-weighted updates. This method relies on a strong policy-constrained method [4]; more importantly, the dropout method does not converge with increasing data [13]. EDAC [26] proposes to use a large number of networks to guarantee diversification so that OOD actions are sufficiently penalized via the traditional min-Q objective. PBRL [8] introduces an OOD sampling procedure similar to CQL and a pseudo target punished by the uncertainty for OOD actions. Theoretically, PBRL analyzes the connection between bootstrapped uncertainty and the LCB penalty, but it does not explicitly address the algorithm’s convergence rate. In contrast, SCORE is supported by a thorough theoretical analysis showing that it achieves a sublinear convergence rate. Since SCORE does not need to sample OOD data, its convergence result remains valid even without access to the exact value of OOD samples.

In addition to algorithmic advances, some research works study the detrimental effects of false correlation from the theoretical perspective. To deal with this problem, most existing works impose various assumptions on the sufficient coverage of the dataset, e.g., the ratio between the visitation measure of the target policy and that of the behavior policy to be upper bounded uniformly over the state-action space [27], [28], [29], [30], [31], [32], [33], [34], [35], or the concentrability coefficient to be upper bounded [36], [37], [38], [39], [40], [41]. Until recently, without assuming sufficient coverage of the dataset, Jin et al. [5] incorporate pessimism into value iteration to establish a data-dependent upper bound on the suboptimality in episodic MDPs. We take a step forward from the theoretical result in [5], extending it to address infinite-horizon regularized MDPs and incorporating the policy optimization process, which bridges the gap between theory and practice. Our algorithm converges to an optimal policy with a sublinear rate, relying on a minimal assumption regarding the compliance of the dataset. Specifically, we assume the data collection process aligns with the MDP of interest.

5 EXPERIMENTS

In this section, we conduct extensive experiments to verify the effectiveness of the propose algorithm. We first present the settings in Section 5.1, followed by the comparison results in Section 5.2. Then we visualize and analyze the uncertainty learned by our method in Section 5.3. Lastly, we discuss the results of ablation studies and hyperparameter analyzes in Section 5.4 and Section 5.5 respectively.

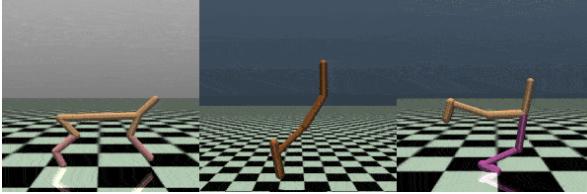


Figure 2. An illustration of the robotic tasks: halfcheetah (left), hopper (middle), walker2d(right).

5.1 Experimental Settings

We conduct extensive experiments on the widely adopted D4RL [2] benchmark. D4RL provides three robotic tasks (halfcheetah, hopper, and walker2d) shown in Figure 2. Each task has five datasets of different qualities: “random” is collected using a random initialized policy, “medium” is collected using a partially trained policy, and “expert” is collected using a well-trained policy. “medium-replay” and “medium-expert” are derived from a mixture of policies¹, with the former containing all the data in the replay buffer of the “medium-level” policy and the latter being the product of mixing the “medium” and “expert” datasets in equal proportions. In brief, unlike online RL experiments that only test algorithms on different tasks, the offline RL benchmark evaluates algorithms on datasets of different qualities. In this way, we can comprehensively assess the ability of an algorithm to learn effective policies on various training data distributions. There are 15 datasets in total.

We evaluate 8 different baselines of diversified types on the D4RL benchmark, as summarized in Table 1:

- **BCQ** [1] is a simple yet strong baseline, providing stable performance. It involves training a Variational Auto-Encoder (VAE) and an agent with the actor-critic architecture. In particular, the actor perturbs the actions proposed by the VAE instead of directly output actions, and the critic determines the final action to be performed from a set of candidate actions.
- **BEAR** [4] employs a similar architecture to BCQ but directly uses the actor to output actions. The authors propose to use the Maximum Mean Discrepancy (MMD) distance to constrain the learning process in policy optimization to avoid large differences with the behavioral policy.
- **CQL** [9] is the state-of-the-art offline RL algorithm. Unlike policy-constrained methods, it incorporates a strong value regularizer into the critic’s loss to suppress the value of OOD actions, thus indirectly leading the agent to resist them. Specifically, the OOD data is sampled using the learned policy in practice, which requires additional computational costs.
- **MOPO** [10] is a model-based offline RL algorithm which modifies the observed reward by incorporating the maximum standard deviation of the learned models (model uncertainty) as a penalty. The agent is then trained with the penalized rewards.
- **MOReL** [25] is also a model-based offline RL algorithm. Instead of the using the maximum standard deviation,

1. In practice, the behavioral policy is usually unknown and comes in the form of a mixture of multiple policies of different quality.

Table 1
Characteristics of recent offline RL algorithms. The notations ‘P’, ‘V’, ‘U’, ‘O’, and ‘M’ represent ‘Policy-constrained’, ‘Value-penalized’, ‘Uncertainty-aware’, ‘OOD sampling’ and ‘Model-based’ respectively.

	P	V	U	O	M
BCQ [1]	✓	✗	✗	✗	✗
BEAR [4]	✓	✗	✗	✗	✗
MOPO [10]	✗	✓	✓	✗	✓
MOReL [25]	✗	✓	✓	✗	✓
CQL [9]	✗	✓	✗	✓	✗
UWAC [23]	✓	✗	✓	✗	✗
TD3-BC [24]	✓	✗	✗	✗	✗
PBRL [8]	✗	✓	✓	✓	✗
SCORE (Ours)	✓	✓	✓	✗	✗

MOReL proposes to use the maximum disagreement of the learned models as the penalty.

- **UWAC** [23] improves BEAR [4] by incorporating MC dropout to estimate the uncertainty of the input sample and weight the loss accordingly.
- **TD3-BC** [24] is a simple offline reinforcement learning algorithm with minimal modifications to the TD3 [42] algorithm. The authors suggest to use a weighted sum of the critic loss and the behavior cloning loss to update the actor. Despite its simplicity, the method also offers stable and good performance.
- **PBRL** [8] quantifies uncertainty through the disagreements of the bootstrapped Q-networks and performs pessimistic updates. Similar to CQL, PBRL utilizes an OOD sampler to obtain pseudo-OOD samples. In the policy evaluation step, PBRL proposes to introduce an additional pseudo target for these samples and imposes a large uncertainty penalty on them. Inspired by [13], PBRL-prior constructs a set of random prior networks of the same size as the Q-networks on top of PBRL. Empirically, the performance is remarkably improved by incorporating these prior networks. Therefore, we report the performance of the PBRL-prior version in the experiments.

We remark that the results reported in the D4RL white paper [2] are based on the “v0” version of D4RL, and most previous work reuse those results for comparison. However, the “v0” version has some errors that may lead to wrong conclusions and the authors release a “v2” version later to correct the errors. To this end, we rerun all the baseline methods on the “v2” version and report the new results.

All the experiments are run on a single NVIDIA Quadro RTX 6000 GPU. To guarantee a fair comparison, we conducted the experiments under the same experimental protocol. The whole training process has a total of 1 million gradient steps, which are divided into 1000 epochs. At the end of each epoch, we run 10 episodes for evaluation. To reduce the effect of randomness on the experimental results, all experiments are run with 5 independent random seeds while keeping other factors unchanged.

We specify both the actor network and the critic network to be a three-layer neural network with 256 neurons per layer. The first two layers of both networks use the ReLU activation function. In particular, the last layer of the actor network uses the tanh activation function for outputting actions. Both networks have a separated Adam optimizer and the learning rate is 3e-4. Some baselines differ in settings and have additional hyperparameters. We use the official code and the suggested settings. When the

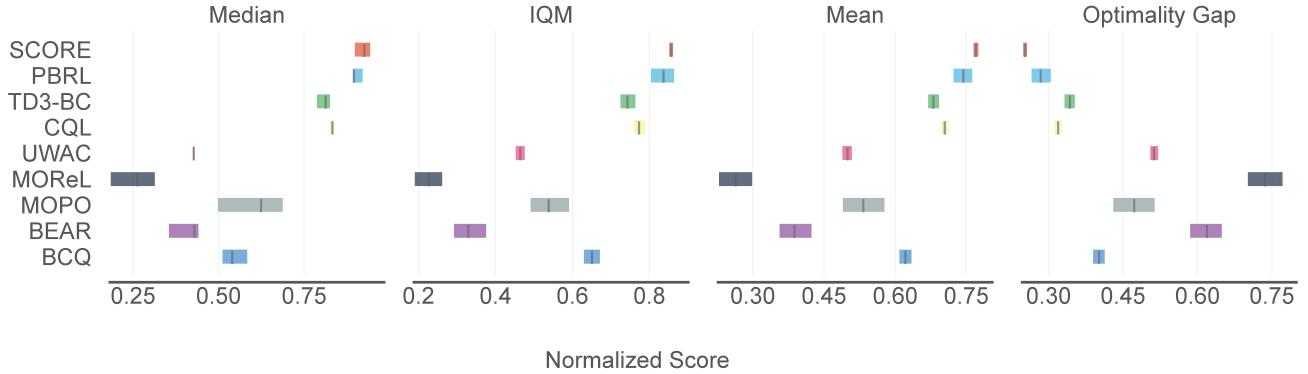


Figure 3. Aggregate metrics on D4RL-MuJoCo with 95% CIs based on 15 tasks. Higher mean, median, and interquartile mean (IQM) scores and lower optimality gap are better. The CIs are estimated using the percentile bootstrap with stratified sampling. IQM typically results in smaller CIs than median scores. All results are based on 5 runs (with different random seeds) per task.

(BCQ) (MOPO) (UWAC) (TD3-BC) (SCORE)
(BEAR) (MOReL) (CQL) (PBRL)

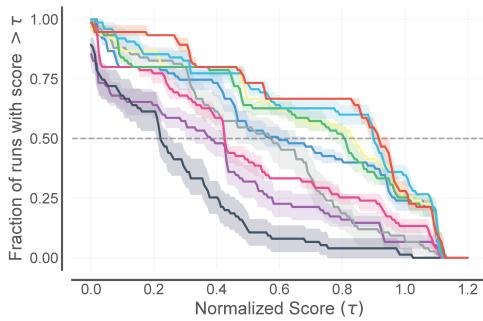


Figure 4. Performance profiles on D4RL-MuJoCo. The distributions are computed using normalized scores obtained after training for 1M steps. Shaded regions show 95% percentile stratified bootstrap CIs. The larger the area under the curve, the better.

default parameters do not perform well, we use grid search to find the optimal hyperparameters for different datasets.

5.2 Comparison Experiments

As suggested by the NeurIPS 2021 outstanding paper [43], we analyze the experimental results via the reliable evaluation suite², which reports the interval estimations of the overall performance instead of the point estimations. As a result, it can effectively prevent the researcher from making unreliable conclusions. We present all four aggregate metrics with the corresponding confidence intervals. Specifically, the IQM score is regarded as a more robust aggregate metric than the commonly used mean score because it is robust to outlier scores and is more statistically efficient. The detailed numerical results are available in the Appendix B. According to Figure 3 and Figure 4, the two best-performing baselines are CQL and PBRL, while SCORE performs the best consistently in all four metrics. Unlike CQL, SCORE and PBRL differentiate the action-value using uncertainty, allowing refined value penalty and better performance. While PBRL is motivated by defending against OOD actions and reducing extrapolation errors. The uncertainty penalty does more than that, as shown in Section 2. PBRL heavily relies on the OOD data sampling procedure during the learning process, which is quite time-consuming. On the other hand, SCORE identifies the core issue as the inability to obtain high-quality uncertainty estimations. We propose using

Table 2
Comparison of the computational costs of the three best-performing methods.

	CQL	PBRL	SCORE
Num of parameters	0.69M	5.66M	0.87M
Run time per epoch	32~35s	52~55s	16~18s

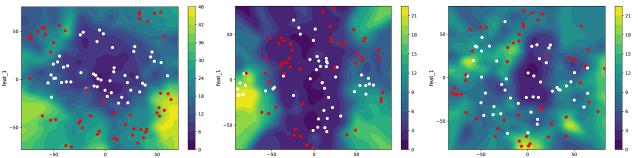


Figure 5. Uncertainty estimation of in-distribution (white dots) and OOD (red dots) samples.

a simple regularizer to effectively address this issue, enabling a reliable implementation of the pessimism principle. According to Table 2, SCORE has a much lower computational cost than CQL and PBRL due to its simplicity. By removing the OOD samplers from CQL and PBRL, SCORE significantly accelerates the training process without sacrificing the performance.

Empirically, SCORE is particularly effective on low- and medium-quality datasets. While the datasets do not satisfy the sufficient coverage assumption since the behavioral policies in these settings are highly suboptimal, they still include some positive behaviors, so it is possible to learn better policies than the behavioral ones. The key to mastering these good behaviors is to avoid suboptimal actions that lead to high returns by chance. While many existing methods focus on the extrapolation errors caused by OOD actions, SCORE strives to address false correlations, a border issue, offering further improvements.

5.3 Visualization and Analysis of Uncertainty

To gain a better insight into SCORE, we visualize the learned uncertainty. Specifically, we apply the Q functions trained on the medium-replay dataset to quantify uncertainty. The in-distribution samples are drawn from the medium-replay dataset, and the OOD samples are from the expert dataset. For visualization purposes, we reduce the features of these samples to two dimensions using t-distributed stochastic neighbor embedding (t-SNE). Figure 5 shows the contour plot of the uncertainty on the two-dimensional

2. <https://github.com/google-research/rliable>

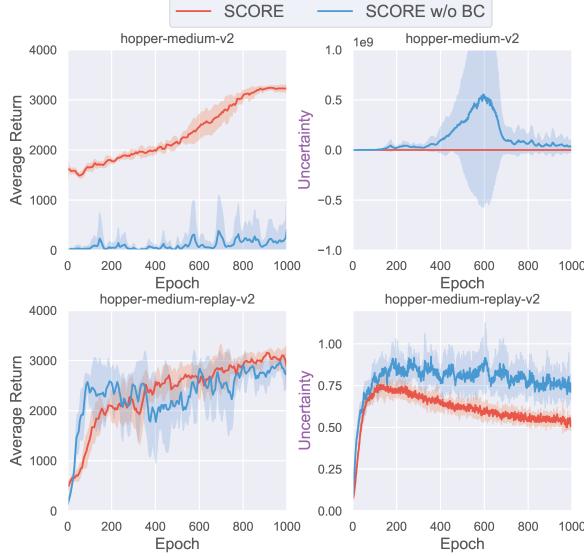


Figure 6. The average return and the uncertainty of SCORE with or without the proposed regularizer.

feature space, in which the white dots denote in-distribution samples and the red dots correspond to OOD samples.

At first glance, the in-distribution samples (white) are more concentrated in regions with low uncertainty (the dark regions), while the OOD samples (red) loosely distribute in regions with higher uncertainty (the bright regions). We remark that the in-distribution and OOD samples are more easily distinguishable on halfcheetah, while the opposite holds for hopper and walker2d. This phenomenon surprisingly matches the performance observed in the experiments. On halfcheetah, the performance on the medium-replay dataset is substantially lower than on the expert dataset, while it is much closer on hopper and walker2d. We suggest that this phenomenon reflects the property of the dataset, i.e., the medium-replay datasets of hopper and walker2d may have better coverage of the state-action pairs induced by the expert policy. Thus, algorithms are more likely to learn high-level policies from these two medium-quality datasets.

5.4 Ablation Studies

To better understand the effects of the proposed regularizer and the uncertainty penalty, we conduct experiments that remove or replace them with other alternatives in this section.

Regularization. As pointed out in previous studies [1], [7], uncertainty-based methods empirically fail in offline RL. Moreover, estimating calibrated uncertainty for neural networks is a challenging task. Figure 6 shows the ablation of the proposed BC regularizer. We first note that it plays a crucial role on narrow dataset distributions, e.g., the hopper-medium dataset collected by a single policy. Without this regularizer, the estimated uncertainty climbs rapidly at the beginning of the training process and then keeps fluctuating dramatically between enormous values. Meanwhile, we observe a remarkably different pattern when using the regularizer, with uncertainty tending to stabilize and slowly decrease after an initial rising phase. Even when the data are sufficiently diverse (less likely to suffer from OOD queries), e.g., the hopper-medium-replay dataset, regularization still helps stabilize uncertainty. These results reveal that the proposed regularizer contributes to reliable uncertainty estimations. Further, such

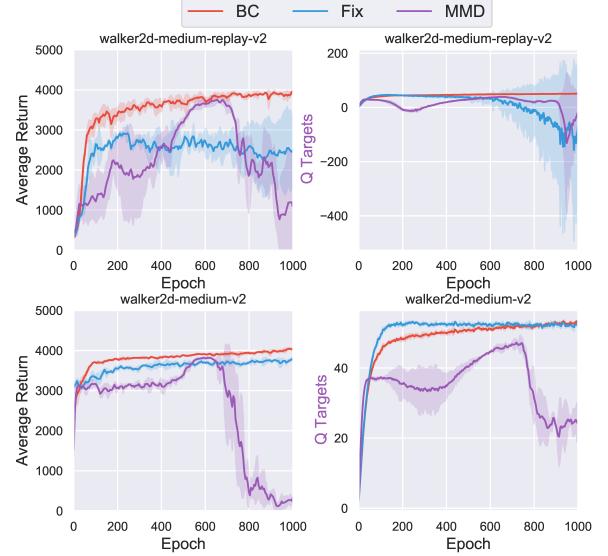


Figure 7. The average return and the Q target of SCORE using different types of policy constraints. "BC" stands for our method, "Fix" uses a VAE to fit the behavioral policy and then use it to propose actions (similar to BCQ [1]), and "MMD" constrains the learned policy to stay close to the estimated behavior policy under MMD (similar to BEAR [4]).

high-quality uncertainty estimations drive SCORE to reduce false correlations and achieve superior performance.

Next, we investigate the effects of different policy constraint methods. In addition to using behavioral cloning, we study 1) using a VAE to fit the behavioral policy and propose actions, and 2) using a VAE to fit the behavioral policy and then constrain the learned policy to stay close to it during the policy optimization process. Figure 7 reports the results. We can see that fitting a behavioral policy is usually a bad idea when the dataset is collected by a mixture of policies, e.g., the medium-replay datasets, since these policies sometimes conflict. As a result, proposing actions or constructing constraints based on the fitted behavioral policy may lead to suboptimal behavior. Even on datasets collected by a single policy, e.g., the medium datasets, using the fitted behavior policy is undesirable as it causes the agent to prematurely converge on suboptimal behavior. Overall, we find the proposed BC regularizer is simple yet effective.

Pessimism. In theory, false correlation is a critical factor that induces suboptimality in offline reinforcement learning, and pessimism in the face of uncertainty is a provably efficient solution to eliminate false correlations. Figure 8 shows the difference between SCORE with and without pessimism. On the hopper-medium-replay dataset, removing the uncertainty penalty affects the stability of the training process and the convergence level. On the walker2d-medium-replay dataset, this even causes severe degradation. Through the curves of Q-values, we find that the performance drop when there is a jitter or explosion in Q-values. The agent without pessimism underperforms on both datasets and shows high Q-values. This is consistent with the intuition we established in Section 3.1, i.e., an agent without knowledge of uncertainty is easily misled by false correlations, which makes the agent believe suboptimal actions have higher value and therefore acquire suboptimal policies. Overall, these results suggest that the regularizer alone cannot effectively solve the offline RL problem and that the pessimism principle is indispensable.

We also test other methods for quantifying uncertainty.

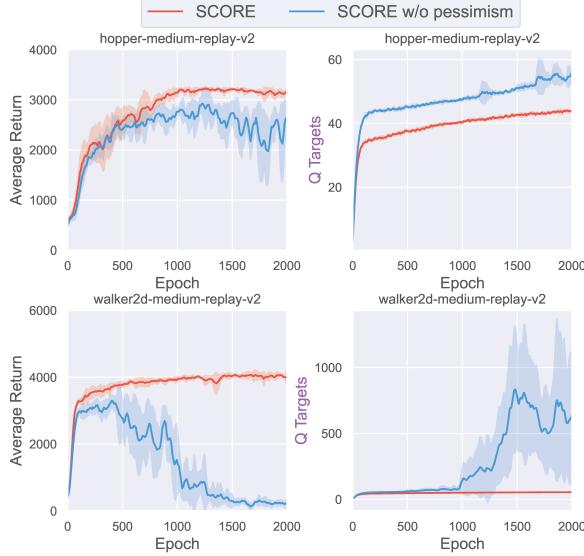


Figure 8. The average return and the Q target of SCORE with and without the uncertainty penalty.

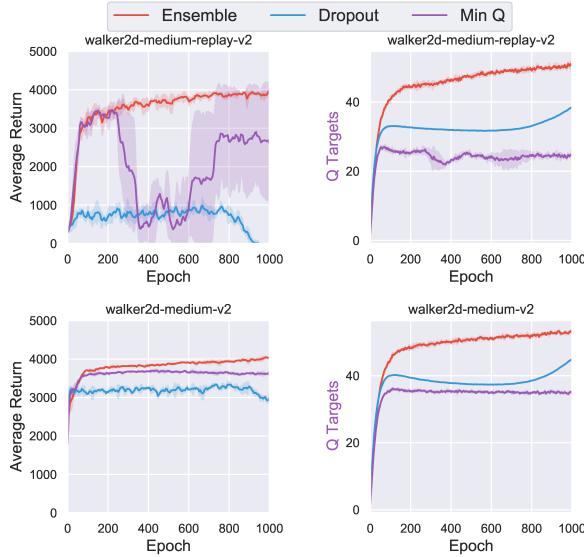


Figure 9. The average return and the Q target of SCORE using different types of uncertainty qualification techniques. "Ensemble" stands for our method, "Dropout" means MC dropout, and "Min Q" ensures pessimism by using the smallest Q estimation.

SCORE quantifies uncertainty based on the bootstrapped ensemble method and uses it to construct the penalty term. Alternatively, one can use the MC dropout method to quantify uncertainty. In addition, one of the simplest ways to achieve pessimism is to use the minimum Q-value of multiple critics as the target. Figure 9 illustrates the experimental results. We can see that both alternatives are inferior to the ensemble method, and they perform poorly on the medium-replay dataset generated by a mixture of multiple policies. As discussed in [13], the dropout distribution does not concentrate with more observed data. Therefore, MC dropout fails to provide the epistemic uncertainty needed to eliminate false correlations in offline RL. As for the Min Q method, it generally needs more networks (accompanied by higher computation cost) and additional techniques to guarantee the diversity of the critics [26]; otherwise, the multiple critics trained

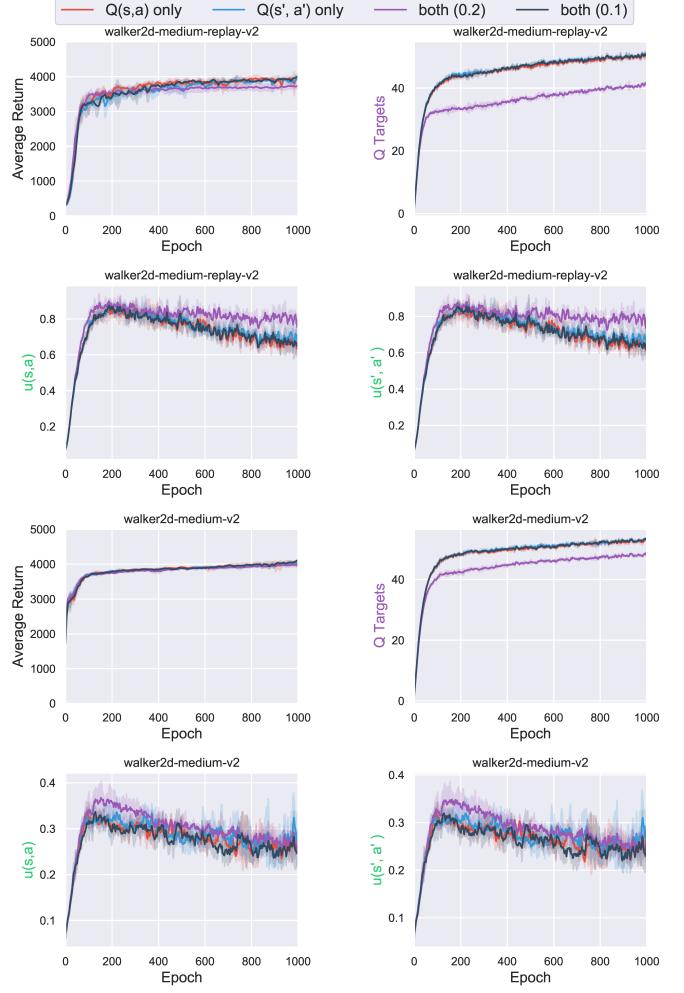


Figure 10. The average return and the Q target of SCORE with different types of penalties. " $Q(s, a)$ only" and " $Q(s', a')$ only" mean the uncertainty penalty is computed only using $Q(s, a)$ or $Q(s', a')$, respectively ($\beta = 0.2$). "both" (β) means the two penalty terms are used simultaneously. "both (0.1)" controls the overall strength penalty to be the same as " $Q(s, a)$ only" and " $Q(s', a')$ only", while "both (0.2)" doubles the strength.

using the same target values may degenerate and fail to perceive uncertainty. Overall, the bootstrapped ensemble method performs well in measuring epistemic uncertainty and ensures SCORE's strong performance.

Penalty. Remark that Equation 6 uses $u(s, a)$ for penalization. We note that researchers tried different types of penalties in previous work [8], including $u(s', a')$ or the combination of $u(s, a)$ and $u(s', a')$. Empirically, PBRL [8] works best with $u(s', a')$. We conduct a similar experiment on SCORE to further study the penalty term. Figure 10 demonstrates that SCORE is robust to different types of penalties, showing similar performance. This result motivates us to further analyze the difference between $u(s, a)$ and $u(s', a')$. $u(s, a)$ is computed based on $Q(s, a)$, which can further decompose as $r(s, a) + \mathbb{E}_{s'} \max_{a'} Q(s', a')$. We note that the Mujoco robotic tasks use a deterministic environment, meaning there is no missing information in $r(s, a)$ and s' . Therefore, if the agent is well-trained, the uncertainty $u(s, a)$ should depend only on $Q(s', a')$, which relates it to $u(s', a')$. This explains why " $Q(s, a)$ only", " $Q(s', a')$ only", and "both (0.1)" show similar performance, Q values, and even uncertainty. We suggest

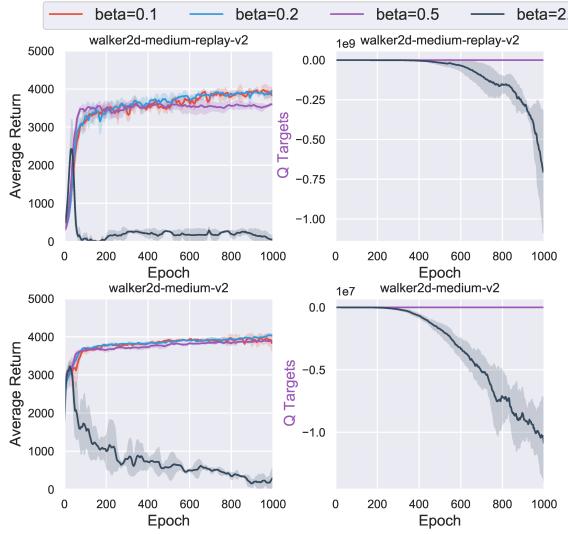


Figure 11. The average return and the Q target of SCORE with different penalty coefficient β .

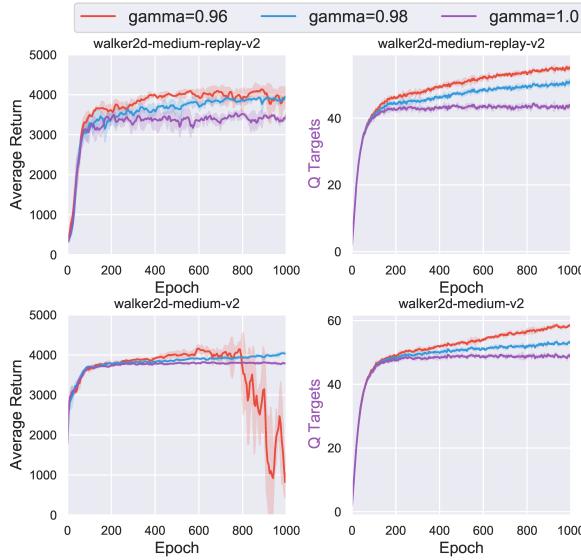


Figure 12. The average return and the Q target of SCORE with different decaying factor γ_{bc} .

that $u(s', a')$ works the best in [8] because it directly fixes the uncertainty of $r(s, a)$ and s' to be zero, avoiding the noise that may affect the training process. Overall, these experimental results support that SCORE obtains reliable uncertainty. The estimated uncertainty captures the long-term effects of the decisions; otherwise, it would equal zero in a deterministic environment.

5.5 Hyperparameter Analyses

The penalty coefficient β . β controls the degree of pessimism. In Figure 8, we have seen that the agent fails to resolve false correlations without pessimism ($\beta = 0$). Here we conduct experiments by choosing β from $\{0.1, 0.2, 0.5, 2.0\}$. The experimental results are shown in Figure 11. We can see that $\beta = 0.1$, $\beta = 0.2$ and $\beta = 0.5$ work similarly on both datasets, with rapid improvement and stable convergence. In particular, when the penalty gets too large ($\beta = 2.0$), the agent becomes over-pessimistic (the Q value rapidly becomes an extremely small negative number). In this

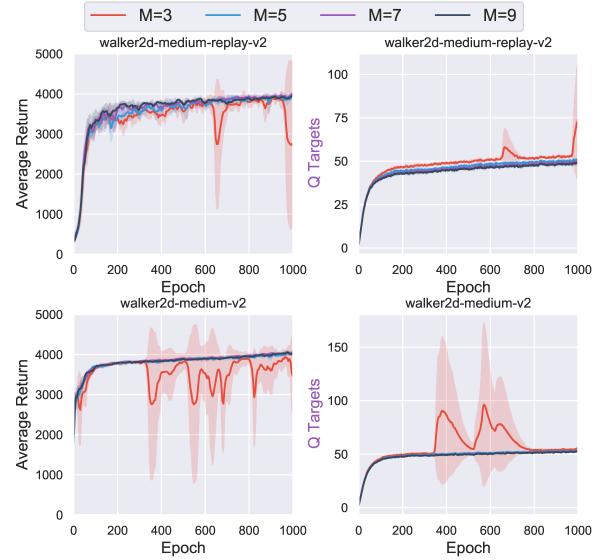


Figure 13. The average return and the Q target of SCORE with different number of ensemble networks.

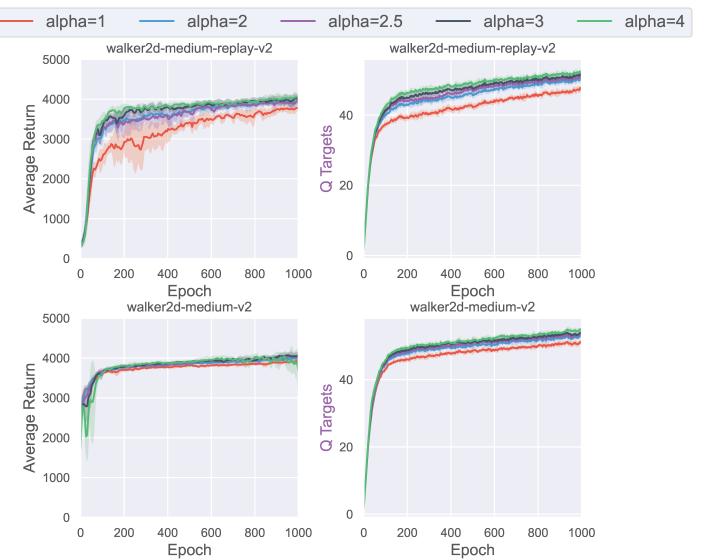


Figure 14. The average return and the Q target of SCORE with different weighting factor α of the Q-loss.

case, the agent tends to act conservatively and cannot effectively exploit the information in the dataset to learn informed decisions, resulting in poor performance.

The discount rate of the behavior cloning coefficient γ_{bc} . We use a decaying factor γ_{bc} to control the weight of the behavior cloning loss in policy's objective function (equation 7). We choose discount rate γ_{bc} from $\{0.96, 0.98, 1.0\}$ to validate the sensitivity with respect to behavior cloning. We remark that $\gamma_{bc} = 1$ corresponds to a constant weight. From Figure 12 we can see that $\gamma_{bc} = 0.96$ and $\gamma_{bc} = 0.98$ work similarly on the medium-replay dataset, with $\gamma_{bc} = 0.96$ converges faster. In contrast, $\gamma_{bc} = 1.0$ results in a sub-optimal policy that converges prematurely. This is because the medium-replay dataset is generated by a mixture of policies of different quality. A strong behavior cloning regularizer hinders the agent to take the essence and discard the dross. On the medium dataset collected by a single behavioral policy, $\gamma_{bc} = 0.96$, $\gamma_{bc} = 0.98$ and $\gamma_{bc} = 1.0$ perform similarly at

the initial training stage, with $\gamma_{bc} = 0.98$ displaying the highest level of convergence. Although $\gamma_{bc} = 0.96$ shows the same rapid improvement, it collapses in the later stages of the strategy. In general, a small γ_{bc} is preferable when the dataset is diverse; conversely, a strong constraint is required.

The number of Q-networks. Empirically, using more ensemble networks provides better uncertainty quantification. We examine the case of $M \in \{3, 5, 7, 9\}$. Figure 13 shows that the bootstrapped ensemble method fails to provide reliable uncertainty estimations when when $M = 3$, leading to unstable performance. Interestingly, SCORE works very well with merely five networks. Using more networks improves stability but has little impact on performance improvements. Compared to [26] and [8], the computational overhead is much lower.

The weighting factor of the Q-loss. Following Fujimoto et al. [24], we employ the Q-value normalization trick to scale the Q-loss during policy optimization. [24] found that their algorithm is robust to the weighting factor α and the recommended setting is $\alpha = 2.5$. Figure 14 shows a similar observation. SCORE is robust to a wide range of α . We use the default value $\alpha = 2.5$ in all the other experiments.

6 CONCLUSION

In this work, we propose a novel offline RL algorithm named SCORE (falSe CORrelation REduction), which achieves the SoTA performance with 3.1x acceleration. We identify the false correlation between epistemic uncertainty and decision-making as a core issue in offline RL, which is a broader and more rigorous mathematical concept than the widely studied OOD action problem. To address this issue, SCORE employs the bootstrapped ensemble method to quantify uncertainty and treats it as a penalty. We point out that the failure of previous work is due to the inability to obtain high-quality uncertainty estimations, and propose introducing a simple annealing BC regularizer to solve the problem. Theoretically, we take a step forward from existing work by analyzing policy optimization. We show the proposed algorithm converges to the optimal policy with a sublinear rate under mild assumptions. Moreover, according to the extensive empirical results, SCORE is both provably efficient and practically effective. In the future, we plan to extend the theory to include safety, robustness, and other desiderata and design practical algorithms consistent with the theory.

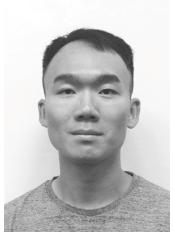
REFERENCES

- [1] S. Fujimoto, D. Meger, and D. Precup, “Off-Policy Deep Reinforcement Learning without Exploration,” in *Proceedings of the 36th International Conference on Machine Learning*, May 2019, pp. 2052–2062.
- [2] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, “D4RL: Datasets for Deep Data-Driven Reinforcement Learning,” *arXiv preprint arXiv:2004.07219*, Feb. 2020.
- [3] Y. Wu, G. Tucker, and O. Nachum, “Behavior Regularized Offline Reinforcement Learning,” *arXiv preprint arXiv:1911.11361*, Nov. 2019.
- [4] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, “Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [5] Y. Jin, Z. Yang, and Z. Wang, “Is Pessimism Provably Efficient for Offline RL?” in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 5084–5096.
- [6] T. Xie, C.-A. Cheng, N. Jiang, P. Mineiro, and A. Agarwal, “Bellman-consistent Pessimism for Offline Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 6683–6694.
- [7] S. Levine, A. Kumar, G. Tucker, and J. Fu, “Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems,” *arXiv preprint arXiv:2005.01643*, Nov. 2020.
- [8] 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 *International Conference on Learning Representations*, Mar. 2022.
- [9] A. Kumar, A. Zhou, G. Tucker, and S. Levine, “Conservative Q-Learning for Offline Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1179–1191.
- [10] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma, “MOPO: Model-based Offline Policy Optimization,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 14129–14142.
- [11] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, “Deep Exploration via Bootstrapped DQN,” in *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [12] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [13] I. Osband, J. Aslanides, and A. Cassirer, “Randomized Prior Functions for Deep Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [14] K. Ciosek, Q. Vuong, R. Loftin, and K. Hofmann, “Better Exploration with Optimistic Actor Critic,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [15] K. Lee, M. Laskin, A. Srinivas, and P. Abbeel, “SUNRISE: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning,” in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 6131–6141.
- [16] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, “COMBO: Conservative Offline Model-Based Policy Optimization,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 28954–28967.
- [17] Y. Guo, S. Feng, N. L. Roux, E. Chi, H. Lee, and M. Chen, “Batch Reinforcement Learning Through Continuation Method,” in *International Conference on Learning Representations*, Feb. 2022.
- [18] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic Policy Gradient Algorithms,” in *Proceedings of the 31st International Conference on Machine Learning*, Jan. 2014, pp. 387–395.
- [19] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau, “Benchmarking Batch Deep Reinforcement Learning Algorithms,” *arXiv preprint arXiv:1910.01708*, Oct. 2019.
- [20] S. K. S. Ghasemipour, D. Schuurmans, and S. S. Gu, “EMaQ: Expected-Max Q-Learning Operator for Simple Yet Effective Offline and Online RL,” in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 3682–3691.
- [21] A. Nair, A. Gupta, M. Dalal, and S. Levine, “AWAC: Accelerating Online Reinforcement Learning with Offline Datasets,” *arXiv preprint arXiv:2006.09359*, Apr. 2021.
- [22] I. Kostrikov, R. Fergus, J. Tompson, and O. Nachum, “Offline Reinforcement Learning with Fisher Divergence Critic Regularization,” in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 5774–5783.
- [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 *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 11319–11328.
- [24] S. Fujimoto and S. S. Gu, “A Minimalist Approach to Offline Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 20132–20145.
- [25] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims, “MOREL: Model-Based Offline Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21810–21823.
- [26] G. An, S. Moon, J.-H. Kim, and H. O. Song, “Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 7436–7447.
- [27] M. Farajtabar, Y. Chow, and M. Ghavamzadeh, “More Robust Doubly Robust Off-policy Evaluation,” in *Proceedings of the 35th International Conference on Machine Learning*, Jul. 2018, pp. 1447–1456.
- [28] Q. Liu, L. Li, Z. Tang, and D. Zhou, “Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [29] T. Xie, Y. Ma, and Y.-X. Wang, “Towards Optimal Off-Policy Evaluation for Reinforcement Learning with Marginalized Importance Sampling,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [30] O. Nachum, Y. Chow, B. Dai, and L. Li, “DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

- [31] N. Jiang and J. Huang, "Minimax Value Interval for Off-Policy Evaluation and Policy Optimization," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 2747–2758.
- [32] Y. Duan, Z. Jia, and M. Wang, "Minimax-Optimal Off-Policy Evaluation with Linear Function Approximation," in *Proceedings of the 37th International Conference on Machine Learning*, Nov. 2020, pp. 2701–2709.
- [33] M. Yang, O. Nachum, B. Dai, L. Li, and D. Schuurmans, "Off-Policy Evaluation via the Regularized Lagrangian," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6551–6561.
- [34] M. Yin, Y. Bai, and Y.-X. Wang, "Near-Optimal Provable Uniform Convergence in Offline Policy Evaluation for Reinforcement Learning," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, Mar. 2021, pp. 1567–1575.
- [35] R. Zhang, B. Dai, L. Li, and D. Schuurmans, "GenDICE: Generalized Offline Estimation of Stationary Values," in *International Conference on Learning Representations*, Feb. 2022.
- [36] B. Scherrer, M. Ghavamzadeh, V. Gabilon, B. Lesner, and M. Geist, "Approximate Modified Policy Iteration and its Application to the Game of Tetris," *Journal of Machine Learning Research*, vol. 16, no. 49, pp. 1629–1676, 2015.
- [37] J. Chen and N. Jiang, "Information-Theoretic Considerations in Batch Reinforcement Learning," in *Proceedings of the 36th International Conference on Machine Learning*, May 2019, pp. 1042–1051.
- [38] T. Xie and N. Jiang, "Q* Approximation Schemes for Batch Reinforcement Learning: A Theoretical Comparison," in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, Aug. 2020, pp. 550–559.
- [39] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A Theoretical Analysis of Deep Q-Learning," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, Jul. 2020, pp. 486–489.
- [40] T. Xie and N. Jiang, "Batch Value-function Approximation with Only Realizability," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, Jul. 2021, pp. 11404–11413.
- [41] P. Liao, Z. Qi, R. Wan, P. Klasnja, and S. Murphy, "Batch Policy Learning in Average Reward Markov Decision Processes," *The Annals of Statistics*, vol. 50, no. 6, pp. 3364–3387, Sep. 2022.
- [42] S. Fujimoto, H. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *Proceedings of the 35th International Conference on Machine Learning*, Jul. 2018, pp. 1587–1596.
- [43] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep Reinforcement Learning at the Edge of the Statistical Precipice," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 29304–29320.
- [44] Q. Cai, Z. Yang, C. Jin, and Z. Wang, "Provably Efficient Exploration in Policy Optimization," in *Proceedings of the 37th International Conference on Machine Learning*, Nov. 2020, pp. 1283–1294.



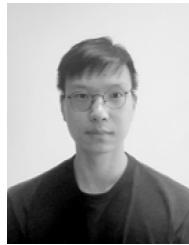
Zihong Deng received his undergraduate and master degree in computer science in 2017 and 2020 respectively from Sun Yat-sen University. He is pursuing a PhD degree at University of Technology Sydney. His research interests span machine learning and data mining, with a special focus on reinforcement learning. He has published papers in multiple international conferences and journals, such as AAAI, ICLR, IEEE TCYB and IEEE TNNLS.



Zuyue Fu received his Ph.D. degree from Northwestern University in 2022. His research interest lies in the intersection of optimization and machine learning, with a special focus on reinforcement learning. He is now a research scientist at Meta.



Lingxiao Wang received his undergraduate degree in mathematics in 2017 from Nanyang Technological University. He received his master's and Ph.D. degrees from Northwestern University in 2020 and 2022, respectively. His research interests lie in reinforcement learning, with a special focus on exploration and sample efficiency analysis.



Zhuoran Yang is an Assistant Professor of Statistics and Data Science at Yale University, starting in July 2022. His research interests lie in the interface between machine learning, statistics, and optimization. He is particularly interested in the foundations of reinforcement learning, representation learning, and deep learning. Before joining Yale, Zhuoran worked as a postdoctoral researcher at the University of California, Berkeley, advised by Michael I. Jordan. Prior to that, he obtained his Ph.D. from the Department of Operations Research and Financial Engineering at Princeton University, co-advised by Jianqing Fan and Han Liu. He received his bachelor's degree in Mathematics from Tsinghua University in 2015.



Chenjia Bai received the B.S., M.S., and Ph.D. degrees in Computer Science and Technology from the Harbin Institute of Technology, Harbin, China, in 2015, 2017, and 2022, respectively. He is currently a Researcher with Shanghai Artificial Intelligence Laboratory. His main research interests include reinforcement learning, deep learning networks, and robotics.



Tianyi Zhou is a tenure-track assistant professor of computer science at the University of Maryland, College Park. He received his Ph.D. from the school of computer science & engineering at the University of Washington, Seattle. His research interests are in machine learning, optimization, and natural language processing (NLP). He published over 70 papers and is a recipient of the Best Student Paper Award at ICDM 2013 and the 2020 IEEE Computer Society TCSC Most Influential Paper Award.



Zhaoran Wang is an assistant professor at Northwestern University, working at the interface of machine learning, statistics, and optimization. He is the recipient of the AISTATS (Artificial Intelligence and Statistics Conference) notable paper award, ASA (American Statistical Association) best student paper in statistical learning and data mining, INFORMS (Institute for Operations Research and the Management Sciences) best student paper finalist in data mining, Microsoft Ph.D. Fellowship, Simons-Berkeley/J.P. Morgan AI Research Fellowship, Amazon Machine Learning Research Award, and NSF CAREER Award.



Jing Jiang is a Senior Lecturer in the School of Computer Science, a core member of Australian Artificial Intelligence Institute (AAII), at the University of Technology Sydney (UTS) in Australia. Her research interests focus on machine learning and its applications. She is the recipient of the DECRA (Discovery Early Career Researcher Award) fellowship funded by ARC (Australian Research Council). She has published over 70 papers in the related areas of AI in the top-tier conferences and journals, such as NeurIPS, ICML, ICLR, AAAI, IJCAI, KDD, TNNLS and TKDE.

APPENDIX A

THEORETICAL ANALYZES AND PROOFS

A.1 Proof of Theorem 3.3

Proof. We denote by

$$\text{AveSubOptGap}(K) = \frac{1}{K} \cdot \sum_{k=0}^{K-1} (V_k^*(s_0) - V_k^{\pi_k}(s_0)). \quad (13)$$

By the definition of $\text{SubOptGap}(K)$ in equation 8, we know that $\text{SubOptGap}(K) \leq \text{AveSubOptGap}(K)$.

Before we prove the theorem, we first introduce the following useful lemmas.

Lemma A.1 (Suboptimality Decomposition). For $\text{AveSubOptGap}(K)$ defined in equation 13, we have

$$\begin{aligned} \text{AveSubOptGap}(K) &= \frac{1}{K} \cdot \sum_{k=0}^{K-1} \sum_{t=0}^{\infty} \gamma^t \cdot \left(\mathbb{E}_{\pi^*} \left[\left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_k(\cdot | s_t) \right\rangle \middle| s_0 \right] \right. \\ &\quad \left. + \mathbb{E}_{\pi^*} [\iota_k(s_t, a_t) | s_0] - \mathbb{E}_{\pi_k} [\iota_k(s_t, a_t) | s_0] \right), \end{aligned}$$

where $\iota_k(s, a) = r(s, a) + \gamma \cdot \mathbb{E}_{s' \sim \mathcal{P}(\cdot | s, a)} [V_k(s')] - Q_k(s, a)$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$.

Proof. See proof of Lemma 4.2 in [44] for a detailed proof. \square

Lemma A.2 (Policy Improvement). It holds for any k that

$$\begin{aligned} &(\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_k(\cdot | s_t) \right\rangle \\ &\leq \text{KL}(\pi^*(\cdot | s_t) \| \pi_k(\cdot | s_t)) - \text{KL}(\pi^*(\cdot | s_t) \| \pi_{k+1}(\cdot | s_t)) \\ &\quad + (\eta_k + \lambda_k)^{-2} \cdot (1 + \lambda_k \cdot \alpha^4 (1 - \alpha)^{-4})^2 \cdot (1 - \gamma)^{-2}. \end{aligned}$$

Proof. See Section A.2.2 for a detailed proof. \square

Lemma A.3 (Pessimism). Under Assumption 3.2, with probability at least $1 - \xi$, it holds for any $(s, a, k) \in \mathcal{S} \times \mathcal{A} \times [K]$ that

$$0 \leq \iota_k(s, a) \leq 2U(s, a),$$

where $\iota_k = \mathcal{B}Q_k - \hat{\mathcal{B}}Q_k$ is the epistemic error defined in equation 1.

Proof. See proof of Lemma 5.1 in [5] for a detailed proof. \square

Now we prove the theorem. By Lemma A.1, we have

$$\begin{aligned} \text{AveSubOptGap}(K) &= \frac{1}{K} \cdot \sum_{k=0}^{K-1} \sum_{t=0}^{\infty} \gamma^t \cdot \left(\mathbb{E}_{\pi^*} \left[\left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_k(\cdot | s_t) \right\rangle \middle| s_0 \right] \right. \\ &\quad \left. + \mathbb{E}_{\pi^*} [\iota_k(s_t, a_t) | s_0] - \mathbb{E}_{\pi_k} [\iota_k(s_t, a_t) | s_0] \right) \\ &\leq \frac{1}{K} \cdot \sum_{k=0}^{K-1} \sum_{t=0}^{\infty} \gamma^t \cdot \left(\mathbb{E}_{\pi^*} [\eta \cdot \text{KL}(\pi^*(\cdot | s_t) \| \pi_k(\cdot | s_t)) - \eta \cdot \text{KL}(\pi^*(\cdot | s_t) \| \pi_{k+1}(\cdot | s_t))] \right. \\ &\quad \left. + \eta^{-1} \cdot (1 + \lambda_k \cdot \alpha^4 (1 - \alpha)^{-4})^2 \cdot (1 - \gamma)^{-2} \right. \\ &\quad \left. + \mathbb{E}_{\pi^*} [\iota_k(s_t, a_t) | s_0] - \mathbb{E}_{\pi_k} [\iota_k(s_t, a_t) | s_0] \right), \end{aligned} \quad (14)$$

where we denote by $\eta = \eta_k + \lambda_k$, and the last inequality comes from Lemma A.2. Further, by telescoping the sum of k on the right-hand side of equation 14 and the non-negativity of the KL divergence, it holds with probability at least $1 - \xi$ that

$$\begin{aligned} \text{AveSubOptGap}(K) &\leq \frac{\eta}{K} \cdot \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}_{\pi^*} [\text{KL}(\pi^*(\cdot | s_t) \| \pi_0(\cdot | s_t))] + \eta^{-1} \cdot (1 + \alpha^4 (1 - \alpha)^{-4})^2 \cdot (1 - \gamma)^{-3} \\ &\quad + \frac{1}{K} \cdot \sum_{k=0}^{K-1} \sum_{t=0}^{\infty} \gamma^t \cdot \left(\mathbb{E}_{\pi^*} [\iota_k(s_t, a_t) | s_0] - \mathbb{E}_{\pi_k} [\iota_k(s_t, a_t) | s_0] \right) \\ &\leq \frac{\eta}{K} \cdot \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}_{\pi^*} [\text{KL}(\pi^*(\cdot | s_t) \| \pi_0(\cdot | s_t))] + \eta^{-1} \cdot (1 + \alpha^4 (1 - \alpha)^{-4})^2 \cdot (1 - \gamma)^{-3} \\ &\quad + \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}_{\pi^*} [2U(s_t, a_t) | s_0], \end{aligned} \quad (15)$$

where the last inequality comes from Lemma A.3. Now, by taking $\eta = \sqrt{\zeta/K}$, where

$$\zeta = (1 + \alpha^4(1 - \alpha)^{-4})^2 \cdot \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}_{\pi^*} [\text{KL}(\pi^*(\cdot | s_t) \| \pi_0(\cdot | s_t))],$$

combining equation 15, with probability at least $1 - \xi$, we have

$$\text{AveSubOptGap}(K) = O((1 - \gamma)^{-3} \sqrt{\zeta/K}) + \varepsilon_{\text{Pess}}.$$

Here $\varepsilon_{\text{Pess}}$ is the intrinsic uncertainty defined in equation 12. By the fact that $\text{SubOptGap}(K) \leq \text{AveSubOptGap}(K)$, we conclude the proof. \square

A.2 Proof of Lemmas

A.2.1 Proof of Lemma 3.1

Proof. By plugging the definition of $\Pi_\phi(s) = \mathbb{E}_{a \sim \pi_\phi(\cdot | s)}[a]$ and the linear parameterization $Q_k(s, a) = \theta_k(s)^\top a$ into equation 11, we have

$$\mathcal{L}_{\text{OPO}}^k(\phi) = \mathbb{E}_{s \sim \mathcal{D}} [Q_k(s, \Pi_\phi(s)) - \lambda_k \cdot \text{KL}(\pi_\phi(\cdot | s) \| \pi_0(\cdot | s)) - \eta_k \cdot \text{KL}(\pi_\phi(\cdot | s) \| \pi_k(\cdot | s))]. \quad (16)$$

It holds for any $s \in \mathcal{S}$ that

$$\begin{aligned} \nabla_\phi \text{KL}(\pi_\phi(\cdot | s) \| \pi_0(\cdot | s)) &= \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(\cdot | s)} [\log(\pi_\phi(a | s) / \pi_0(a | s))] \\ &= \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(\cdot | s)} [(\phi - \phi_0)^\top \psi(s, a) + Z_\phi(s) - Z_{\phi_0}(s)] \\ &= \nabla_\phi \mathbb{E}_{a \sim \pi_\phi(\cdot | s)} [\psi(s, a)] (\phi - \phi_0) + \mathbb{E}_{a \sim \pi_\phi(\cdot | s)} [\psi(s, a)] - \nabla_\phi Z_\phi(s) \\ &= \nabla_\phi^2 Z_\phi(s) (\phi - \phi_0) = \text{Var}_{a \sim \pi_\phi(\cdot | s)} [\psi(s, a)] (\phi - \phi_0) \\ &= I_\phi(s) (\phi - \phi_0). \end{aligned} \quad (17)$$

Similarly, we have

$$\nabla_\phi \text{KL}(\pi_\phi(\cdot | s) \| \pi_k(\cdot | s)) = I_\phi(s) (\phi - \phi_k). \quad (18)$$

for any $s \in \mathcal{S}$. Thus, by combining equation 16, equation 17, and equation 18, the stationary point ϕ_{k+1} of $\mathcal{L}_{\text{OPO}}^k(\phi)$ satisfies

$$\begin{aligned} \mathbb{E}_{s \sim \mathcal{D}} [\nabla_a Q_k(s, \Pi_{\phi_{k+1}}(s)) \nabla_\phi \Pi_{\phi_{k+1}}(s) - \lambda_k \cdot I_{\phi_{k+1}}(s) (\phi_{k+1} - \phi_0) \\ - \eta_k \cdot I_{\phi_{k+1}}(s) (\phi_{k+1} - \phi_k)] = 0. \end{aligned} \quad (19)$$

Now, by equation 19, we have

$$\phi_{k+1} = \frac{\eta_k \phi_k + \lambda_k \phi_0}{\eta_k + \lambda_k} + (\eta_k + \lambda_k)^{-1} \cdot I_{\phi_{k+1}}^{-1} \mathbb{E}_{s \sim \mathcal{D}} [\nabla_a Q_k(s, \Pi_{\phi_{k+1}}(s)) \nabla_\phi \Pi_{\phi_{k+1}}(s)],$$

which concludes the proof. \square

A.2.2 Proof of Lemma A.2

Proof. First, by maximizing equation 11, we have

$$\pi_{k+1}(a | s) \propto \exp\{(\eta_k + \lambda_k)^{-1} \cdot (Q_k(s, a) + \eta_k f_k(s, a) + \lambda_k f_0(s, a))\}.$$

Thus, for any policy π' and π'' , it holds for any $s \in \mathcal{S}$ that

$$\begin{aligned} \left\langle \log \frac{\pi_{k+1}(\cdot | s)}{\pi_k(\cdot | s)}, \pi'(\cdot | s) - \pi''(\cdot | s) \right\rangle \\ = (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s)}{\pi_0(\cdot | s)}, \pi'(\cdot | s) - \pi''(\cdot | s) \right\rangle. \end{aligned} \quad (20)$$

We will use equation 20 in the following proof.

Note that

$$\begin{aligned} &\text{KL}(\pi^*(\cdot | s_t) \| \pi_k(\cdot | s_t)) - \text{KL}(\pi^*(\cdot | s_t) \| \pi_{k+1}(\cdot | s_t)) \\ &= \left\langle \log \frac{\pi_{k+1}(\cdot | s_t)}{\pi_k(\cdot | s_t)}, \pi^*(\cdot | s_t) \right\rangle \\ &= \left\langle \log \frac{\pi_{k+1}(\cdot | s_t)}{\pi_k(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle + \text{KL}(\pi_{k+1}(\cdot | s_t) \| \pi_k(\cdot | s_t)). \end{aligned} \quad (21)$$

In the meanwhile, we have

$$\begin{aligned}
& \left\langle \log \frac{\pi_{k+1}(\cdot | s_t)}{\pi_k(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle \\
&= \left\langle \log \frac{\pi_{k+1}(\cdot | s_t)}{\pi_k(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_k(\cdot | s_t) \right\rangle + \left\langle \log \frac{\pi_{k+1}(\cdot | s_t)}{\pi_k(\cdot | s_t)}, \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle \\
&= (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_k(\cdot | s_t) \right\rangle \\
&\quad + (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle,
\end{aligned} \tag{22}$$

where the last equality comes from equation 20. Combining equation 21 and equation 22, we have

$$\begin{aligned}
& (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_k(\cdot | s_t) \right\rangle \\
&= \text{KL}(\pi^*(\cdot | s_t) \| \pi_k(\cdot | s_t)) - \text{KL}(\pi^*(\cdot | s_t) \| \pi_{k+1}(\cdot | s_t)) - \text{KL}(\pi_{k+1}(\cdot | s_t) \| \pi_k(\cdot | s_t)) \\
&\quad - (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle \\
&\leq \text{KL}(\pi^*(\cdot | s_t) \| \pi_k(\cdot | s_t)) - \text{KL}(\pi^*(\cdot | s_t) \| \pi_{k+1}(\cdot | s_t)) - \|\pi_{k+1}(\cdot | s_t) - \pi_k(\cdot | s_t)\|_1^2 / 2 \\
&\quad - (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle,
\end{aligned} \tag{23}$$

where the last inequality comes from Pinsker's inequality. To upper bound the last term on the right-hand side of equation 23, we characterize $\log(\pi_k(a | s) / \pi_0(a | s))$ as follows.

Characterization of $\log(\pi_k / \pi_0)$. For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\log \frac{\pi_k}{\pi_0} = \log \left(\frac{\pi_k}{\pi_{k-1}} \cdot \frac{\pi_{k-1}}{\pi_{k-2}} \cdots \frac{\pi_1}{\pi_0} \right) = \sum_{i=0}^{k-1} \log \frac{\pi_{i+1}}{\pi_i}.$$

Then, we have

$$\log \frac{\pi_k}{\pi_0} = \sum_{i=0}^{k-1} \log \frac{\pi_{i+1}}{\pi_i} = \sum_{i=0}^{k-1} \left(Q_i + \lambda_i \cdot \log \frac{\pi_i}{\pi_0} \right) + Z_1, \tag{24}$$

where Z_1 is a function independent of a . Now, by recursively applying equation 24, we have

$$\log \frac{\pi_k}{\pi_0} = \sum_{i=0}^{k-1} Q_i \cdot \sum_{j=i+1}^k \lambda_j \left(1 + \sum_{\ell=0}^{k-j-1} \varepsilon_\ell \prod_{p=0}^\ell \lambda_{k-p} \right) + Z_2, \tag{25}$$

where ε_ℓ is either 1 or -1 , and Z_2 is a function independent of a .

Now, by equation 25, the last term on the right-hand side of equation 23 can be upper bounded as follows,

$$\begin{aligned}
& -(\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle \\
&= -(\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \sum_{i=0}^{k-1} Q_i(s_t, \cdot) \sum_{j=i+1}^k \lambda_j \left(1 + \sum_{\ell=0}^{k-j-1} \varepsilon_\ell \prod_{p=0}^\ell \lambda_{k-p} \right) - \lambda_k \cdot Z_2(s_t), \right. \\
&\quad \left. \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle \\
&= -(\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \sum_{i=0}^{k-1} Q_i(s_t, \cdot) \sum_{j=i+1}^k \lambda_j \left(1 + \sum_{\ell=0}^{k-j-1} \varepsilon_\ell \prod_{p=0}^\ell \lambda_{k-p} \right), \right. \\
&\quad \left. \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle \\
&\leq (\eta_k + \lambda_k)^{-1} \cdot \left\| Q_k(s_t, \cdot) - \lambda_k \sum_{i=0}^{k-1} Q_i(s_t, \cdot) \sum_{j=i+1}^k \lambda_j \left(1 + \sum_{\ell=0}^{k-j-1} \varepsilon_\ell \prod_{p=0}^\ell \lambda_{k-p} \right) \right\|_\infty \\
&\quad \cdot \|\pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t)\|_1,
\end{aligned} \tag{26}$$

where the last line comes from Hölder's inequality. In the meanwhile, it holds that

$$\|Q_k\|_\infty \leq (1 - \gamma)^{-1}, \tag{27}$$

and

$$\left\| \sum_{i=0}^{k-1} Q_i(s_t, \cdot) \sum_{j=i+1}^k \lambda_j \left(1 + \sum_{\ell=0}^{k-j-1} \varepsilon_\ell \prod_{p=0}^\ell \lambda_{k-p} \right) \right\|_\infty \leq \alpha^4 (1-\alpha)^{-4} (1-\gamma)^{-1}. \quad (28)$$

Now, by plugging equation 27 and equation 28 into equation 26, we have

$$\begin{aligned} & (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t) \right\rangle \\ & \leq (\eta_k + \lambda_k)^{-1} \cdot (1 + \lambda_k \alpha^4 (1-\alpha)^{-4}) (1-\gamma)^{-1} \cdot \|\pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t)\|_1. \end{aligned} \quad (29)$$

Now, combining equation 23 and equation 29, it holds that

$$\begin{aligned} & (\eta_k + \lambda_k)^{-1} \cdot \left\langle Q_k(s_t, \cdot) - \lambda_k \cdot \log \frac{\pi_k(\cdot | s_t)}{\pi_0(\cdot | s_t)}, \pi^*(\cdot | s_t) - \pi_k(\cdot | s_t) \right\rangle \\ & \leq \text{KL}(\pi^*(\cdot | s_t) \| \pi_k(\cdot | s_t)) - \text{KL}(\pi^*(\cdot | s_t) \| \pi_{k+1}(\cdot | s_t)) - \|\pi_{k+1}(\cdot | s_t) - \pi_k(\cdot | s_t)\|_1^2 / 2 \\ & \quad + (\eta_k + \lambda_k)^{-1} \cdot (1 + \lambda_k \alpha^4 (1-\alpha)^{-4}) (1-\gamma)^{-1} \cdot \|\pi_k(\cdot | s_t) - \pi_{k+1}(\cdot | s_t)\|_1 \\ & \leq \text{KL}(\pi^*(\cdot | s_t) \| \pi_k(\cdot | s_t)) - \text{KL}(\pi^*(\cdot | s_t) \| \pi_{k+1}(\cdot | s_t)) \\ & \quad + (\eta_k + \lambda_k)^{-2} \cdot (1 + \lambda_k \cdot \alpha^4 (1-\alpha)^{-4})^2 \cdot (1-\gamma)^{-2}, \end{aligned}$$

which concludes the proof. \square

APPENDIX B SUPPLEMENTARY TABLES

Table 3
Hyperparameters of SCORE

Basic hyperparameters from TD3 [42]		Value	SCORE hyperparameters		Value
Notation	Description		Notation	Description	
σ	The std of the Gaussian exploration noise	0.2	M	The number of critic networks	5
c	The max noise	0.5	d_{bc}	The update frequency of the behavior cloning coefficient	10000
d	The update frequency of the actor network and the target networks	2	γ_{bc}	The discount rate of the behavior cloning coefficient	{0.96, 0.98, 1.0}
τ	The target network update rate	0.005	β	The uncertainty penalty coefficient	{0.1, 0.2, 0.5}

Table 4

Average normalized scores over 5 random seeds on the D4RL datasets. We compare SCORE with both model-based methods (MOPO, MOReL) and model-free methods (BCQ, BEAR, UWAC, CQL, TD3-BC, PBRL, PBRL w/o prior). The standard deviation is reported in the parentheses. A score of zero corresponds to the performance of the random policy and a score of 100 corresponds to the performance of the expert policy.

	Task	SCORE	MOPO	MOReL	BCQ	BEAR	UWAC	CQL	TD3-BC	PBRL	PBRL w/o prior
Random	halfcheetah	29.1±2.6	35.9±2.9	30.3±5.9	2.2±0.0	2.3±0.0	2.3±0.0	21.7±0.6	10.6±1.7	13.1±1.2	11.0±5.8
	hopper	31.3±0.3	16.7±12.2	44.8±4.8	8.1±0.5	3.9±2.3	2.6±0.3	8.1±1.4	8.6±0.4	31.6±0.3	26.8±9.3
	walker2d	3.7±7.0	4.2±5.7	17.3±8.2	4.6±0.7	12.8±10.2	1.8±0.4	0.5±1.3	1.5±1.4	8.8±6.3	8.1±4.4
Medium Replay	halfcheetah	48.0±0.7	69.2±1.1	31.9±6.0	40.9±1.1	36.3±3.1	36.4±3.3	47.2±0.4	44.8±0.5	49.5±9.8	45.1±8.0
	hopper	94.0±1.8	32.7±9.4	54.2±32.0	40.9±16.7	52.2±19.3	23.7±2.6	95.6±2.4	57.8±17.3	100.7±0.4	100.6±1.0
	walker2d	84.8±1.1	73.7±2.4	13.7±8.0	42.5±13.7	7.0±7.8	24.3±5.4	85.3±2.7	81.9±2.7	86.2±3.4	77.7±14.5
Medium	halfcheetah	55.2±0.4	73.1±2.4	20.4±13.8	45.4±1.7	43.0±0.2	42.3±0.3	49.2±0.3	47.8±0.4	58.2±1.5	57.9±1.5
	hopper	99.6±2.8	38.3±34.9	53.2±32.1	54.0±3.7	51.8±4.0	50.2±5.2	62.7±3.7	69.1±4.5	81.6±14.5	75.3±31.2
	walker2d	89.2±1.2	41.2±30.8	10.3±8.9	74.5±3.7	-0.2±0.1	72.8±4.1	83.3±0.8	81.3±3.0	90.3±1.2	89.6±0.7
Medium Expert	halfcheetah	92.6±3.5	70.3±21.9	35.9±19.2	94.0±1.2	46.0±4.7	42.8±0.3	70.6±13.6	88.9±5.3	93.6±2.3	92.3±1.1
	hopper	100.3±6.9	60.6±32.5	52.1±27.7	108.6±6.0	50.6±25.3	48.6±7.8	111.0±1.2	102.0±10.1	111.2±0.7	110.8±0.8
	walker2d	109.3±0.5	77.4±27.9	3.9±2.8	109.7±0.6	22.1±44.5	96.9±7.1	109.7±0.3	110.5±0.3	109.8±0.2	110.1±0.3
Expert	halfcheetah	96.4±0.6	81.3±21.8	2.2±5.4	92.7±2.5	92.7±0.6	93.5±0.9	97.5±1.8	96.3±0.9	96.2±2.3	92.4±1.7
	hopper	112.0±0.3	62.5±29.0	26.2±14.0	105.3±8.1	54.6±21.0	103.9±13.6	105.4±5.9	109.5±4.1	110.4±0.3	110.5±0.4
	walker2d	109.4±0.6	62.4±3.2	-0.3±0.3	109.0±0.4	106.8±6.8	108.2±0.5	109.0±0.4	110.3±0.4	108.8±0.2	108.3±0.3
Overall		77.0±2.0	53.3±16.3	26.4±12.6	62.6±4.0	38.8±10.0	50.0±3.5	70.5±2.5	68.1±3.5	76.6±2.4	74.4±5.3