OFFLINE REINFORCEMENT LEARNING VIA WEIGHTED f-DIVERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

One of the major challenges of offline reinforcement learning (RL) is dealing with distribution shifts that stem from the mismatch between the trained policy and the data collection policy. Prior offline RL algorithms have addressed this issue by regularizing the policy optimization with f-divergence between the state-action visitation distributions of the data collection policy and the optimized policy. While such regularization provides a theoretical lower bound on performance and has had some practical success, it is not affected by the optimality of state-actions and can be overly pessimistic, especially when valuable state-actions are rare in the dataset. To mitigate the problem, we introduce and analyze a weighted f-divergence regularized RL framework that can less regularize valuable but rare state-actions as long as sampling error allows. This leads to an offline RL algorithm with iterative stationary distribution correction estimation while jointly re-adjusting the regularization for each state-action. We show that the presented algorithm with weighted f-divergence performs competitively with the state-of-the-art methods.

1 INTRODUCTION

Recent advances in reinforcement learning (RL) have enabled agents to make sophisticated decisions under complex environments. However, it assumes free exploration of the agent in the environment to learn from the trial-and-error experiences. Such online interactions can be expensive, unsafe, and even impossible in many real-world applications such as autonomous driving and surgical operation. As a result, the practicality of standard online RL has been questioned due to the high cost of online interaction.

To this end, offline RL has been proposed to train the agent only with the fixed dataset, excluding any online interaction between the environment while training. This is similar to other machine learning schemes from computer vision and natural language processing, where models are trained only with a large-scale offline dataset. However, offline RL suffers from the distributional shift when the trained policy deviates from the policy used to collect the dataset. The shift can cause overestimation during the training step when the agent encounters samples that are not present in the dataset.

Diverse approaches in offline RL were made as a countermeasure for the distribution shift, and one method we focus on in this paper is about regularizing the f-divergence between the state-action visitation distributions of the trained policy and the data collection policy (Wu et al., 2019; Lee et al., 2021). In these algorithms, f-divergence plays a significant role by regulating the distribution shift and enabling safe estimation of the RL objective without sampling from the target policy, by establishing approximate lower bounds on performance. Also known as behavior regularization, they encourage the learned policy to visit states within the data distribution in practice and have proven its effectiveness over a wide range of domains.

However, f-divergence regularization is independent of how valuable state-actions are, and it can cause offline RL policies to be overly pessimistic about the rare but valuable state-actions in the dataset. It is highly likely in practice where the data collection policy is far from being optimal, causing the agent to neglect important and rare state-actions just like other non-important and rare state-actions. The resulting performance loss is inevitable due to the nature of f-divergence regularization.

To address this issue, we present a weighted f-divergence regularized reinforcement learning framework, where we aim to regularize differently based on the optimality of each state-action pair. We apply perspective function to f-divergence to add weights while preserving its convexity, and analyze the possibility of performance gain based on the weighted regularization. Building upon the stationary distribution correction estimation algorithms (Nachum et al., 2019b; Lee et al., 2021), we present an algorithm that solves the optimization with weighted regularization jointly while estimating the weights that can better regularize the policy optimization. We show empirically that the optimized weights correctly adjust the degree of regularization to strengthen rare yet valuable state-actions and suppress common yet deleterious state-actions, and demonstrate the performance of the proposed algorithm in the D4RL benchmark (Fu et al., 2020).

2 Preliminaries

Markov decision process (MDP) We assume the reinforcement learning problem under an infinite-horizon discounted Markov Decision Process (MDP) framework. MDP can be represented as a tuple $\mathcal{M} = \langle S, A, T, R, \mu_0, \gamma \rangle$, where S is a set of states s and A is a set of actions a. $T(s'|s,a) : S \times A \to \Delta(S)$ is a transition probability from the state-action pair (s,a) to the next state s'. $R(s,a) : S \times A \to [0, r_{max}]$ is a reward function of the state-action pair (s,a). $\mu_0 \in \Delta(S)$ is an initial state distribution and $\gamma \in [0,1)$ is the discount factor.

A policy $\pi(a|s) : S \to \Delta(A)$ gives a distribution of actions a of the agent given the state s. For given policy π, the state-action and state value function is defined as $Q^\pi(s,a) := \mathbb{E}_{a_t \sim \pi(s_t), s_{t+1} \sim T(s_{t+1}|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t) | s_0 = s, a_0 = a\right]$ and $V^\pi(s) := \mathbb{E}_{a \sim \pi(s)}[Q^\pi(s,a)]$. $Q^\pi(s,a)$ and $V^\pi(s)$ represent expected sum of discounted rewards when the agent with policy π starts from (s,a) and s. The state-action visitation probability of π is defined as $d^\pi(s,a) := (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s, a_t = a)$. $d^\pi(s,a)$ represents discounted sum of probabilities of the agent with policy π visiting (s,a). Reinforcement learning aims to learn a policy that maximizes expected return $J(\pi)$, an expected sum of discounted rewards:

$$\begin{aligned} \max_\pi J(\pi) &= (1-\gamma)\mathbb{E}_{s_0 \sim \mu_0, a_0 \sim \pi(s_0)}[Q^\pi(s_0,a_0)] \\ &= (1-\gamma)\mathbb{E}_{s_0 \sim \mu_0}[V^\pi(s_0)] = \mathbb{E}_{(s,a) \sim d^\pi(s,a)}[R(s,a)], \end{aligned} \tag{1}$$

Linear Programming form By using linear programming (LP) characterization of V-function (V-LP) (Puterman, 1994; Bertsekas, 1995; Bertsekas & Tsitsiklis, 1996; Nachum & Dai, 2020), evaluation of $J(\pi)$ can be seen as solving the following optimization problem:

$$\begin{aligned} &\min_V \, (1-\gamma)E_{s_0 \sim \mu_0}\left[V(s)\right] \\ &\text{s.t. } V(s) \le R(s,a) + \gamma \mathbb{E}_{s' \sim T(s,a)}[V(s')] \quad \forall s, a. \end{aligned} \tag{2}$$

The dual of V-LP gives equivalent optimization problem with respect to state-action visitation distribution (Nachum & Dai, 2020). Using LP duality, dual of V-LP is defined as

$$\max_d \mathbb{E}_{(s,a) \sim d(s,a)}\left[R(s,a)\right] \tag{3}$$

$$\text{s.t. } \sum_{\tilde{a}} d(s,\tilde{a}) = (1-\gamma)\mu_0(s) + \gamma \sum_{\tilde{s},\tilde{a}} T(s|\tilde{s},\tilde{a}) d(\tilde{s},\tilde{a}) \quad \forall s, \tag{4}$$

$$d(s,a) \ge 0 \quad \forall s, a,$$

where Bellman flow constraints (4) are relaxed to be only on states. Using Lagrangian or Fenchel-Rockafellar duality, the optimization problem (3) can be reformulated to unconstrained optimization problem.

Regularized objective in offline RL In this paper, we focus on offline reinforcement learning (RL), where the interaction between the agent and the environment is not allowed. Instead, the dataset $D = \{(s_i, a_i, r_i, s_i')\}_i^N$ consisting of single-step transitions is given to optimize the policy

π. We denote the empirical distribution of dataset D as d^D. In offline RL, the main purpose is to obtain a policy that can perform better than behavior patterns observed in the dataset D. Such attempt can cause distribution shift, where the offline RL agent is trained under one distribution, while being evaluated on a different distribution to optimize its behavior (Levine et al., 2020).

One way to mitigate the distribution shift is to regularize the f-divergence between state-action visitation distribution d^π and empirical distribution d^D. It corresponds to additionally minimize the f-divergence term with the primal optimization problem (1), resulting in the following regularized policy optimization:

$$\max_{\pi} \mathbb{E}_{(s,a)\sim d^\pi}[R(s,a)] - \alpha D_f(d^\pi \| d^D), \tag{5}$$

where f is continuously differentiable and strictly convex function and α determines the amount of regulation to impose. D_f denotes the f-divergence $D_f(d^\pi \| d^D) = \mathbb{E}_{(s,a)\sim d^D}\left[f\left(\frac{d^\pi(s,a)}{d^D(s,a)}\right)\right]$. In previous studies, it has been relaxed to expected divergence between policies $\mathbb{E}_{d^D}[D_f(\pi \| \pi^D)]$ to derive a practical algorithm, e.g. KL-regularized RL (Fox et al., 2016). On the other hand, Nachum et al. (2019b) and Lee et al. (2021) have made use of the duality to optimize (5) directly. We follow the latter approach.

3 DICE VIA WEIGHTED f-DIVERGENCE

3.1 REINFORCEMENT LEARNING WITH WEIGHTED f-DIVERGENCE REGULARIZATION

Regularized policy optimization objective (5) consists of two terms: reward maximization and distribution shift control. Reward maximization is expressed as inner product of state-action visitation distribution $d^\pi(s,a)$ and the reward $R(s,a)$, while distribution shift control corresponds to f-divergence between $d^\pi(s,a)$ and the dataset distribution $d^D(s,a)$. The resultant policy π after optimizing equation 5 can also be understood as a policy optimized with respect to augmented rewards $R(s,a) - \alpha f'\left(\frac{d^\pi(s,a)}{d^D(s,a)}\right)$ (Nachum et al., 2019b). By choosing appropriate α, this regularization encourages conservative behavior, penalizing rare or nonexistent state-action pairs in the dataset.

In current framework, however, the degree of regularization is only determined by the scarcity of state-action in the dataset. As f-divergence is independent to the value of state-actions, rare yet valuable state-actions are equally penalized. Motivated by this, we hypothesize that alleviating regularization on those special state-actions can lead to better performance. It requires to manipulate the degree of regularization differently for each state-action pairs, and it can not be achieved in standard regularized policy optimization framework (5)—where a single hyperparameter α only controls the overall balance between reward maximization and distribution shift prevention.

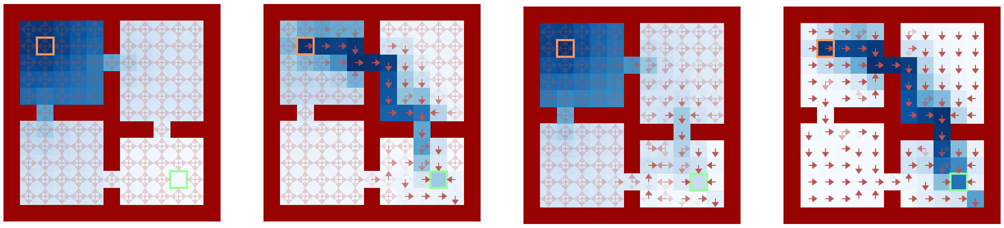
To this end, we propose a weighted f-divergence regularization framework by introducing state-action weighting $k(s,a)$. Note the convexity preservation property of a perspective function:

Lemma 1 (Joint convexity of perspective function)**.** *If f is a convex function and $x > 0$, the perspective function $g(x,y) = xf\left(\frac{y}{x}\right)$ is jointly convex in x and y. (Proof in Appendix A.)*

Above results implies that by using the weighted f-divergence of a form $D_f^k(p\|q) = \mathbb{E}_q\left[k \cdot f\left(\frac{p}{k\cdot q}\right)\right]$, we can still keep the favorable properties of f-divergence regularization based on convexity. Based on the proposed regularization, we restate the policy optimization objective:

$$\begin{aligned}
&\max_{d} \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha \mathbb{E}_{(s,a)\sim d^D}\left[k(s,a) f\left(\frac{d(s,a)}{k(s,a)d^D(s,a)}\right)\right] \\
&\text{s.t.} \sum_{\tilde{a}} d(s,\tilde{a}) = (1-\gamma)\mu_0(s) + \gamma \sum_{\tilde{s},\tilde{a}} T(s|\tilde{s},\tilde{a}) d(\tilde{s},\tilde{a}) \; \forall s, \\
&\quad d(s,a) \geq 0, k(s,a) > 0 \; \forall s,a,
\end{aligned} \tag{6}$$

which extends the optimization problem (3). As proposed in Lee et al. (2021), the Bellman flow constraints only on states are applied so that we can optimize policy implicitly by optimizing the induced state-action distribution d instead. The optimal d^* that solves the problem will be a valid state-action visitation distribution due to the constraints, while maximizing the objective (6). After solving for d^*, the policy π^* that induces d^* can be extracted by various different algorithms.



(a) Dataset dist. d^D (b) Weighted dist. $k^* \cdot d^D$ (c) f-divergence reg. d^* (d) Weighted reg. d^*

Figure 1: **(Four Rooms)** This figure is an illustrative example showing the difference between the f-divergence regularization and weighted f-divergence regularization in Four Rooms domain (Sutton et al., 1998). The red filled squares represent the walls in the environment that cannot be passed, and the orange and green empty squares represent the initial state and the goal state, respectively. The saturation of color in each state represents the state visitation probability $d(s) = \sum_{a \in A} d(s, a)$ based on a state-action distribution denoted below each figure. The policy $\pi(a|s)$ based on a state-action distribution is represented by the red arrows. We collected the dataset using a uniform random policy, which forms the dataset distribution $d^D(s, a)$ shown in (a). (c) shows the resultant state-action visitation distribution of f-divergence regularized RL obtained by running OptiDICE (Lee et al., 2021). (b) shows the multiplication of $d^D(s, a)$ and $k^*(s, a)$ obtained by proposed algorithm, which is the distribution we are regularizing towards with weighted f-divergence. (d) shows the resultant state-action visitation distribution of proposed algorithm.

Note that the proposed weighted regularization can be also seen as penalizing a f-divergence toward the unnormalized density $k \cdot d^D$. In other words, high k will make regularization less penalize the state-actions as if we had more samples of those in the dataset D. Similarly, small k will make regularization severe on the corresponding state-actions as if we had less samples. The usual f-divergence regularization framework (5) can be recovered by setting $k(s, a) = 1\ \forall s, a$.

Illustrative examples In Figure 1, we demonstrate the effectiveness of weighted f-divergence regularization in four rooms environment (Sutton et al., 1998). (a) shows the dataset distribution used in this example. We emphasize that the dataset is collected using a uniform random policy, which has not been considered in previous studies (Lee et al., 2021; Nachum et al., 2019b) on f-divergence regularization. When f-divergence regularization is imposed on a flat dataset distribution d^D, it results in an overly conservative policy, inducing visitations over unnecessary state-actions as shown in (c). On the other hand, (b) shows the result of weighting as a weighted dataset distribution $k^* \cdot d^D$ that weighted f-divergence regularization (6) regularizes towards. k^* used to weight the f-divergence is optimized based on the algorithm that will be introduced later in this paper. (d) shows the resultant state-action visitation distribution based on k^*-weighted f-divergence regularization. It is clearly visible that (d) gives far more focused state-action visitation compared to (c), which directly translates to the performance of the policy. More details can be found in Appendix D.

3.2 DICE VIA WEIGHTED f-DIVERGENCE

In this subsection, we derive an alternative form of the problem (6) that can be practically optimized. We assume the weight k to be fixed and optimize state-action visitation distribution d respect to a given fixed k. We start by obtaining the Lagrangian of (6) where $\nu(s)$ is the Lagrangian multiplier for the equality constraints for valid state-action visitation distribution d.

$$\max_{d \geq 0} \min_{\nu} L_{\alpha,k}(d, \nu) := \mathbb{E}_{(s,a)\sim d}[R(s, a)] - \alpha \mathbb{E}_{(s,a)\sim d^D}\left[k(s, a) f\left(\frac{d(s, a)}{k(s, a) d^D(s, a)}\right)\right] + \sum_s \nu(s)\left((1-\gamma)\mu_0(s) + \gamma \sum_{\tilde{s},\tilde{a}} T(s|\tilde{s}, \tilde{a}) d(\tilde{s}, \tilde{a}) - \sum_{\tilde{a}} d(s, \tilde{a})\right) \quad (7)$$

Interchanging the inner summation over $d(s,a)$ and outer summation over $\nu(s)$, we group the terms that are integrated over $d(s,a)$.

$$L_{\alpha,k}(d,\nu) = \sum_s (1-\gamma)\mu_0(s)\nu(s) - \alpha \mathbb{E}_{(s,a)\sim d^D}\left[k(s,a) f\left(\frac{d(s,a)}{k(s,a)d^D(s,a)}\right)\right] + \sum_{s,a} d(s,a)\left(R(s,a) + \gamma\sum_{\tilde{s}} T(\tilde{s}|s,a)\nu(\tilde{s}) - \nu(s)\right) \tag{8}$$

We apply change of variable $w(s,a) = \frac{d(s,a)}{d^D(s,a)}$ and $e_\nu(s,a) = R(s,a) + \gamma \sum_{\tilde{s}} T(\tilde{s}|s,a)\nu(\tilde{s}) - \nu(s)$ to derive the equivalent objective $L_{\alpha,k}(w,\nu)$ respect to w, ν. Note that $e_\nu(s,a)$ is an advantage function of state-action pair (s,a).

$$\max_{w\geq 0}\min_{\nu} L_{\alpha,k}(w,\nu) := (1-\gamma)\mathbb{E}_{s\sim\mu_0}[\nu(s)] - \alpha\mathbb{E}_{(s,a)\sim d^D}\left[k(s,a) f\left(\frac{w(s,a)}{k(s,a)}\right)\right] + \mathbb{E}_{(s,a)\sim d^D}\left[w(s,a)e_\nu(s,a)\right] \tag{9}$$

Due to the strong duality of (6) with fixed k, it is possible to change the order of optimization. By reordering (9) from maximin to minimax optimization of w and ν, the closed form solution of w with respect to k and ν, $w^*_{\nu,k} := \arg\max_w L_{\alpha,k}(w,\nu)$ can be obtained by solving the inner maximization as shown below. The detailed derivation of $w^*_{\nu,k}$ is deferred to Appendix B.

$$\max_{w\geq 0}\min_{\nu} L_{\alpha,k}(w,\nu) = \min_{\nu}\max_{w\geq 0} L_{\alpha,k}(w,\nu)$$
$$w^*_{\nu,k}(s,a) = \max\left(0, k(s,a)(f')^{-1}\left(\frac{e_\nu(s,a)}{\alpha}\right)\right) \tag{10}$$

We can observe that w^*_ν is directly affected by $k(s,a)$. This derivation is an extension of objective derivation of Lee et al. (2021) to weighted f-divergence regularized problem (6).

Policy extraction After optimizing $\min_\nu L_{\alpha,k}(w^*_{\nu,k},\nu)$, it is possible to compute the optimal state-action visitation distribution $d^* = d^D \cdot w^*_{\nu^*,k}$. While the optimal policy π^* that induces d^* given by above can be easily obtained by performing marginalization $\pi^*(a|s) = \frac{d^D(s,a)w^*_{\nu^*,k}(s,a)}{\sum_{\tilde{a}} d^D(s,\tilde{a})w^*_{\nu^*,k}(s,\tilde{a})}$, it is not straightforward to compute in continuous control domains. For continuous domains, we use the information projection (i.e. I-projection) proposed by Lee et al. (2021):

$$\min_\psi D_{KL}(d^D(s)\pi_\psi(a|s)\|d^D(s)\pi^*(a|s))$$
$$= \min_\psi -\mathbb{E}_{s\sim d^D, a\sim\pi_\psi(s)}[\log w^*_{\nu^*,k}(s,a) - D_{KL}(\pi_\psi(\cdot|s)\|\pi_D(\cdot|s))] + C. \tag{11}$$

3.3 Analysis on the suboptimality of the algorithm on k

So far, we have derived an stationary distribution correction algorithm for policy optimization with a weighted f-divergence regularization. In this subsection, we analyze how weighting f-divergence with k affects the suboptimality of weighted regularized policy optimization framework. We show that the suboptimality of resultant policy can be bounded by two different terms, the regularization loss and the sampling error, that are affected by k in different ways. Based on these results, we show that there is a possibility of improving the optimality of resultant policy with a careful choice of k, which motivates the algorithm to optimize for the right k. Our results are adaptation of the results presented by Zhan et al. (2022).

For further analysis, we denote three policies assuming k fixed: the optimal policy π^* of true MDP $\mathcal{M}$, the regularized optimal policy $\pi^*_{\alpha,k}$ from optimizing the objective (6) under true MDP, and the policy $\hat{\pi}_{\alpha,k}$ obtained from optimizing $L_{\alpha,k}(w,\nu)$ of (9) estimated from samples in the dataset.

Recall that $J(\pi) = \sum_{s,a} d_\pi(s,a)R(s,a)$ corresponds to the expected return we can get by executing policy π. With the policies defined as above, it can be seen that $J(\pi^*)$ is the optimal expected return, and $J(\pi^*) - J(\pi^*_{\alpha,k})$ is the loss we get by adopting regularization. We denote it as a regularization loss. On the other hand, $J(\pi^*_{\alpha,k}) - J(\hat{\pi}_{\alpha,k})$ gives the sampling error we get by using samples from an offline dataset instead of using true MDP. We analyze each contribution to policy suboptimality below. [1]

Regularization loss Regularization loss $J(\pi^*) - J(\pi^*_{\alpha,k})$ is the difference in the expected return between the optimal policy π^* of true MDP $\mathcal{M}$ and the optimal policy $\pi^*_{\alpha,k}$ from (6). The difference is caused by regularizing the unregularized objective (3) with weighted f-divergence. Since $\pi^*_{\alpha,k}$ is the optimal policy in problem (6), we can derive the following inequality.

$$\begin{aligned} J(\pi^*_{\alpha,k}) - \alpha\mathbb{E}_{(s,a)\sim d^D}\left[k(s,a)f\left(\frac{d^{\pi^*_{\alpha,k}}(s,a)}{k(s,a)d^D(s,a)}\right)\right] & \\ \geq J(\pi^*) - \alpha\mathbb{E}_{(s,a)\sim d^D}\left[k(s,a)f\left(\frac{d^{\pi^*}(s,a)}{k(s,a)d^D(s,a)}\right)\right] & \quad (12) \end{aligned}$$

Combined with the optimality of π^* that $J(\pi^*) \geq J(\pi^*_{\alpha,k})$, the regularization loss can be bound by:

$$0 \leq J(\pi^*) - J(\pi^*_{\alpha,k}) \leq \alpha\mathbb{E}_{(s,a)\sim d^D}\left[k(s,a)f\left(\frac{d^{\pi^*}(s,a)}{k(s,a)d^D(s,a)}\right)\right] \quad (13)$$

In other words, the suboptimality of a regularized optimal policy $\pi^*_{\alpha,k}$ is bounded by the weighted f-divergence itself. This implies that k can be optimized to tighten the upper bound of regularization loss by minimizing the f-divergence between $d^{\pi^*}(s,a)$ and $k(s,a)d^D(s,a)$. While this regularization loss (13) will be reduced to 0 when $k(s,a) = \frac{d^{\pi^*}(s,a)}{d^D(s,a)}$ or $\alpha = 0$, we will see in the below that the sampling error can be arbitrarily large in those cases, so they are not what we want to accomplish.

Sampling error Sampling error $J(\pi^*_{\alpha,k}) - J(\hat{\pi}_{\alpha,k})$ is the difference in the expected return between the regularized optimal policy $\pi^*_{\alpha,k}$ and the policy $\hat{\pi}_{\alpha,k}$ extracted from the optimal w of $L_{\alpha,k}(w,\nu)$ of (9). The performance of $\hat{\pi}_{\alpha,k}$ approximately corresponds to the performance of our algorithm, before substituting in the closed form solution of w into the objective (9). We first give $\hat{L}_{\alpha,k}(w,\nu)$ as the sample estimate of $L_{\alpha,k}(w,\nu)$:

$$\hat{L}_{\alpha,k} := (1-\gamma)\frac{1}{n_0}\sum_{j=1}^{n_0}\nu(s_{0,j}) + \frac{1}{n}\sum_{i=1}^{n}\left[-\alpha k(s_i,a_i)f\left(\frac{w(s_i,a_i)}{k(s_i,a_i)}\right) + w(s_i,a_i)e_\nu(s_i,a_i,r_i,s'_i)\right] \quad (14)$$

where n and n_0 is the number of all transitions and the number of initial states in the dataset. The difference between $L_{\alpha,k}(w,\nu)$ and its estimate corresponds to statistical error $|\hat{L}_{\alpha,k}(w,\nu) - L_{\alpha,k}(w,\nu)|$, which we can bound as below:

Theorem 1 (Upper bound of statistical error)**.** *Under some mild assumptions, the statistical error satisfies* $\left|\hat{L}_{\alpha,k}(w,\nu) - L_{\alpha,k}(w,\nu)\right| \leq \mathcal{E}$*:*

$$\mathcal{E} := \mathcal{O}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta_1}}{N}}\right) + \tilde{\mathcal{O}}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta_2}}{N}}\right), \quad (15)$$

where N is the number of samples and δ_1 and δ_2 can be lower-bounded in terms of pseudo-dimension.(Proof in Appendix C.)

[1] The analysis on policy suboptimality is made over the objective (9), and is not exact for the objective (10) where w is maximized in a closed form. This is because e_ν will be biased when computed based on samples, unless we adopt a double-sampling (Baird, 1995; Farahmand & Szepesvári, 2011). The analysis we make here is exact for our algorithm when the environment is deterministic, and is an approximate error bound otherwise.

Now, we bound the sampling error of weighted regularized policy optimization framework in Theorem 2. Theorem 2 is the adaptation of Theorem 3 in Zhan et al. (2022), where f has been replaced with $kf(\frac{x}{k})$.

Theorem 2 (Sampling error)**.** *With at least probability* $1-\delta$*, the sampling error satisfies:*

$$J(\pi^*_{\alpha,k}) - J(\hat{\pi}_{\alpha,k}) \leq \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^*_{\alpha,k}}(s)}[\|\pi^*_{\alpha,k}(\cdot|s) - \hat{\pi}_{\alpha,k}(\cdot|s)\|_1] \leq \frac{4}{1-\gamma}\sqrt{\frac{k_{\max}\mathcal{E}}{\alpha M_f}}, \tag{16}$$

where $d^{\pi^*_{\alpha,k}}(s) = \sum_a d^{\pi^*_{\alpha,k}}(s,a)$ is state-action visitation distribution induced by the policy $\pi^*_{\alpha,k}$, and M_f is the strong convexity coefficient of f. We show that $kf\left(\frac{x}{k}\right)$ is $\frac{M_f}{k}$-strongly convex in Appendix C.1. Combining the result from Theorem 1 and Theorem 2, we derive new upper bound for sampling error that depends on $k_{\max}$. Note that the upper bound increases as k increases, which motivate limiting the maximum of k for suppressing the upper bound of statistical error.

The total suboptimality of applying weighted f-divergence can be approximately given as a sum of the regularization loss and sampling error:

$$J(\pi^*) - J(\hat{\pi}_{\alpha,k}) = J(\pi^*) - J(\pi^*_{\alpha,k}) + J(\pi^*_{\alpha,k}) - J(\hat{\pi}_{\alpha,k}). \tag{17}$$

The analysis motivates us to optimize k according to minimize the upper bound of the regularization loss given by the inequality (13) while bounding $k_{\max}$ to bound the sampling error with high probability. We derive the practical algorithm below based on this motivation.

3.4 Optimization of k

Based on the analysis, we aim to optimize k by minimizing the weighted f-divergence while keeping its maximum value bounded. The corresponding optimization problem can be written as:

$$\begin{aligned} &\min_k \mathbb{E}_{(s,a)\sim d^D}\left[k(s,a)f\left(\frac{d(s,a)}{k(s,a)d^D(s,a)}\right)\right] \\ &\text{s.t. } 0 < k(s,a) < k_{\max}\ \forall s,a \end{aligned} \tag{18}$$

While it is itself a challenging optimization problem that depends on d that is being optimized, fortunately, we have an unconstrained solution for this problem: $w = \frac{d}{d^D}$, and setting $k = w$ does make the objective to be 0, which is the minimum. As we update k, the optimal d will change in a way that it becomes the visitation distribution induced by the optimal policy on a problem regularized by weighted f-divergence. Nevertheless, d is only represented via w and $w = \frac{d}{d^D}$ will mostly hold for implicit d unless the Bellman flow constraint is violated.

Therefore, we simply train k to become a clipped version of w by optimizing:

$$\begin{aligned} &\min_k \frac{1}{2}\mathbb{E}_{(s,a)\sim d^D}\left[\left(k(s,a) - \texttt{stopgrad}[w^*_{\nu,k}(s,a)]\right)^2\right] \\ &\text{s.t. } 0 < k(s,a) < k_{\max},\ \forall s,a \end{aligned} \tag{19}$$

where $w^*_{\nu,k}$ was defined in Equation (10). The `stopgrad` detaches gradient computation, and it is applied to stabilize the learning since k is included in the definition of $w^*_{\nu,k}$. The constrained optimization is done by clipping the $k(s,a)$ itself in the tabular case, and by restricting the output of the neural network itself by clipping. By jointly optimizing k based on the objective (19) and ν based on the objective (10), we look for the optimal regularized policy and improve the regularization at the same time such that we seek for the policy with the best optimality we can get from the given dataset.

4 Related Work

In few offline stationary distribution correction estimation algorithms (DICE) algorithm, Linear Programming (LP) characterization of state-action value function and value function (i.e., Q-LP and V-LP) have been used to reformulate the objective into a more tractable form for both policy evaluation and policy optimization ((Nachum et al., 2019a;b; Lee et al., 2021)). We focus on policy optimization problem with regularized objective using DICE algorithms. As far as we know, AlgaeDICE (Nachum et al. (2019b)) is the first paper that formulates the regularized policy optimization framework by adding f-divergence to the standard RL objective. They derive an unconstrained Lagrangian form objective based on Q-LP. Dual problem of Q-LP has state action dependent Bellman flow constraints and is over-constrained. This works as a big advantage because even if the objective of optimization is changed, it has the same optimal solution (Nachum & Dai (2020)). Through this, AlgaeDICE is possible to reduce the instability of original LP by selecting a tractable object to apply Lagrangian duality. This enabled on-policy policy gradient, without using importance sampling. However AlgaeDICE includes maximin term in derivations because of maximizing under the policy (i.e., maximizing the estimated value of the policy), which is susceptible to unstable. V-LP can address this issue by covering the policy optimization problem easy-to-handle. Unlike Q-LP, dual of V-LP has constraints that only depend to the state and is not over-constrained. While OptiDICE (Lee et al. (2021)) shares the same objective with AlgaeDICE, they apply V-LP and reformulate the objective more tractable form. Therefore, the policy optimization problem is redefined as a single minimization problem which can considerably improve the stability.

5 Experiments

In this section, we compare DICE via weighted f-divergence to past offline reinforcement learning algorithms in tabular and continuous domains. We use random MDPs for tabular domains and a D4RL offline RL dataset (Fu et al. (2020)) for the continuous domain. In D4RL benchmarks, we use 3 tasks of Maze2d and 12 tasks of Gym-MuJoCo. For f-divergence, we follow the choice made by Lee et al. (2021): $f(x) = \frac{1}{2}(x-1)^2$, which is a χ^2-divergence, is used for finite MDPs, and a relaxed χ^2-divergence is used for continuous control, which is defined as:

$$f(x) := \begin{cases} x\log x - x + 1 & \text{if } 0 < x < 1 \\ \frac{1}{2}(x-1)^2 & \text{if } x \geq 1 \end{cases} \tag{20}$$

5.1 Random MDPs

We follow the randomly generated finite MDP experiment previously used to benchmark offline RL algorithms (Laroche et al., 2019; Lee et al., 2020) to compare the performance of DICE via weighted f-divergence and unweighted case, which corresponds to OptiDICE (Lee et al., 2021).

To construct MLE MDP, we generate N trajectories with a uniform random policy. We run finite version of DICE via weighted f-divergence, which is described in Appendix D, and unweighted version of it on MLE MDP to optimize policies. We set $k_{\text{max}} = 10$ for DICE via weighted f-divergence. We test over two types of parameters: the regularization coefficient α and number of trajectories in the dataset. We run the experiment for 30 times and plot the ratio between the average performance of unweighted and weighted cases in Figure 2.

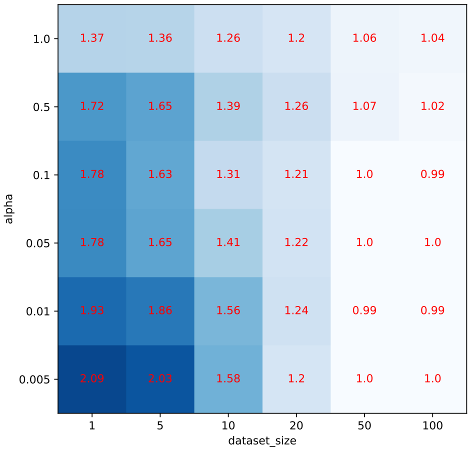


Figure 2: The random MDP experiment showing performance improvement from using weighted f-divergence regularization over different αs and different dataset sizes. The ratios of expected return between DICE with weighted f-divergence and the unweighted version is shown in the figure.

The plot shows that DICE via weighted f-divergence always performs similar to or better than the unweighted case with its robustness to wider range of hyperparameters. Since α that performs well vary by the property of datasets and environments in offline RL, and considering that it is difficult to optimize over hyperparameters in offline RL problems in practice, we argue that the widened coverage of DICE via weighted f-divergence can be quite beneficial.

Table 1: **(D4RL benchmark)** Normalized scores of our algorithm compared with model-free offline reinforcement learning algorithms ((Haarnoja et al., 2018; Kumar et al., 2020; Lee et al., 2021)) on D4RL benchmark (Fu et al. (2020)). DICE via weighted f-divergence attains state-of-the-art performance on 7 tasks and outperforms the unweighted case (OptiDICE) on 13 tasks. For DICE with weighted-f and unweighted f-divergences, we average the score and get the standard error with 95% confidence interval by repeating the experiments 5 times.

Env	**Type**	**BC**	**SAC**	**BRAC-v**	**CQL**	**Unweighted-f**	**Weighted-f**
Maze2d	Umaze	3.8	88.2	-16.0	5.7	97.2 ± 27.4	**135.7** ± **11.5**
Maze2d	Medium	30.3	26.1	33.8	5.0	150.0 ± 16.9	**159.3** ± **12.4**
Maze2d	Large	5.0	-1.9	40.6	12.5	177.8 ± 16.8	**216.2** ± **7.6**
Hopper	Random	9.8	11.3	**12.2**	10.8	11.3 ± 0.1	11.4 ± 0.3
Hopper	Medium	29.0	0.8	31.1	58.0	71.0 ± 24.4	**90.0** ± **14.1**
Hopper	Med-replay	11.8	3.5	0.6	**48.6**	29.3 ± 2.8	28.3 ± 2.2
Hopper	Med-expert	111.9	1.6	0.8	98.7	109.2 ± 3.6	**111.9** ± **0.1**
Walker2d	Random	1.6	4.1	1.9	7.0	7.8 ± 3.2	**11.6** ± **5.4**
Walker2d	Medium	6.6	0.9	**81.1**	79.2	20.9 ± 7.2	26.4 ± 6.6
Walker2d	Med-replay	11.3	1.9	0.9	**26.7**	21.8 ± 4.1	18.2 ± 2.9
Walker2d	Med-expert	6.4	-0.1	51.6	**111.0**	66.7 ± 22.5	72.0 ± 17.6
Halfcheetah	Random	2.1	30.5	31.2	**35.4**	11.6 ± 1.6	13.0 ± 1.1
Halfcheetah	Medium	36.1	-4.3	**46.3**	44.4	38.2 ± 0.1	38.6 ± 0.2
Halfcheetah	Med-replay	38.4	-2.4	**47.7**	46.2	39.6 ± 0.9	40.6 ± 0.2
Halfcheetah	Med-expert	35.8	1.8	41.9	62.4	69.8 ± 8.0	**78.1** ± **7.8**

5.2 D4RL BENCHMARK

We experiment DICE via weighted f-divergence in continuous domain on Datasets for Deep Data-Driven Reinforcement Learning, i.e, D4RL benchmark (Fu et al., 2020). We evaluate the algorithm using Maze2D and Gym-MuJoCo tasks and the normalized scores are presented in Table 1. We compared with BC, offline SAC (Haarnoja et al., 2018), BRAC-v (Wu et al., 2019), CQL (Kumar et al., 2020) and OptiDICE (Lee et al., 2021), which corresponds to the unweighted case of our algorithm. See more experiment details in Appendix E.

In Table 1, it can be found DICE via weighted f-divergence achieve state-of-the-art performance in all of the maze2d domain. When compared to the unweighted case, it can be found that the adoption of $k(s, a)$ which depends on state-action pair turns out to be beneficial in most of the cases. In particular, when the type of environment is random or medium, we can see that our algorithm outperforms the unweighted case overall. This empirically proves that adopting learnable parameters k to f-divergence reduces the impact of suboptimality of dataset and results in a better performance especially when the quality of the given dataset is low.

6 CONCLUSION AND FUTURE WORK

In this paper, we presented DICE via weighted f-divergence, a framework to control the degree of regularization on each state-action by adopting weight k to f-divergence. Based on the analysis on how the total suboptimality of a policy is affected by the adoption of k, we jointly optimized weight k and stationary distribution correction w to extract offline RL policy based on weighted f-divergence regularization. We demonstrated in both finite and continuous domain that the adoption of weighting k does result in a large performance gain over the unweighted algorithm.

While we have only analyzed and demonstrated the weighted f-divergence for the DICE framework in this paper, there is a wide variety of different algorithms that adopts f-divergence regularization differently, e.g. KL-regularized RL. It would be also interesting to adopt a weighted regularization in the other algorithmic frameworks.

REFERENCES

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pp. 30–37. Elsevier, 1995.

Dimitri Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

Dimitri P Bertsekas. Dynamic programming and optimal control, 1995.

Edward G Effros. A matrix convexity approach to some celebrated quantum inequalities. *Proceedings of the National Academy of Sciences*, 106(4):1006–1008, 2009.

Amir-massoud Farahmand and Csaba Szepesvári. Model selection in reinforcement learning. *Machine learning*, 85(3):299–332, 2011.

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 202–211, 2016.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

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

David Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded vapnik-chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.

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

Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, pp. 3652–3661. PMLR, 2019.

Byungjun Lee, Jongmin Lee, Peter Vrancx, Dongho Kim, and Kee-Eung Kim. Batch reinforcement learning with hyperparameter gradients. In *International Conference on Machine Learning*, pp. 5725–5735. PMLR, 2020.

Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pp. 6120–6130. PMLR, 2021.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*, 32, 2019a.

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

David Pollard. *Convergence of stochastic processes*. Springer Science & Business Media, 2012.

Martin L Puterman. Markov decision processes: Discrete stochastic dynamic programming, 1994.

Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998.

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

Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pp. 2730–2775. PMLR, 2022.

A Proof of Lemma 1

Lemma 1 (Joint convexity of perspective function)**.** *If f is a convex function and $x > 0$, the perspective function $g(x, y) = xf\left(\frac{y}{x}\right)$ is jointly convex in x and y. (Proof in Appendix A.)*

Proof. Suppose that f is convex function, the *perspective function* is jointly convex in the sense that if $x = cx_1 + (1-c)x_2$ and $y = cy_1 + (1-c)y_2$ where $[x_i, y_i] = 0$ $(i = 1, 2)$, and $0 \leq c \leq 1$, then $g(x, y) \leq cg(x_1, y_1) + (1-c)g(x_2, y_2)$ (Effros (2009)). Where $\lambda = c\left(\frac{x_1}{cx_1+(1-c)x_2}\right)$ and $1-\lambda = c\left(\frac{x_2}{cx_1+(1-c)x_2}\right)$ satisfy $0 \leq \lambda \leq 1$,

$$\begin{aligned} g(x, y) &= xf\left(\frac{x}{y}\right) \\ &= xf\left(\frac{cy_1}{cx_1+(1-c)x_2} + \frac{(1-c)y_2}{cx_1+(1-c)x_2}\right) \\ &= xf\left(c\frac{y_1}{x_1}\left(\frac{x_1}{cx_1+(1-c)x_2}\right) + (1-c)\frac{y_2}{x_2}\left(\frac{x_2}{cx_1+(1-c)x_2}\right)\right) \\ &= xf\left(\lambda\frac{y_1}{x_1} + (1-\lambda)\frac{y_2}{x_2}\right) \\ &\leq x\lambda f\left(\frac{y_1}{x_1}\right) + x(1-\lambda)f\left(\frac{y_2}{x_2}\right) \\ &= cx_1 f\left(\frac{y_1}{x_1}\right) + (1-c)x_2 f\left(\frac{y_2}{x_2}\right) \\ &= cg(x_1, y_1) + (1-c)g(x_2, y_2). \end{aligned}$$

□

B Proof of optimal w^*

We solve inner maximization problem $\max_{w \geq 0} L_{\alpha,k}(w, \nu)$ by using KKT condition, since the strong duality holds for the optimization problem. Lagrangian of the optimization is given as,

$$\max_{w} \min_{\lambda} L_{\alpha,k,\nu}(w, \lambda) := L_{\alpha,k}(w, \nu) + \sum_{s,a} \lambda(s, a) w(s, a) \tag{21}$$

To satisfy stationarity condition, we obtain the derivative of $L_{k,\nu}(w, \lambda)$ with respect to $w(s, a)$.

$$\frac{\partial}{\partial w(s,a)} L_{k,\nu}(w^*, \lambda^*) = d^D(s,a)[-\alpha f'(\frac{w^*(s,a)}{k(s,a)}) + e_\nu(s,a)] + \lambda^*(s,a) = 0,\ \forall s, a \tag{22}$$

$$w^*(s,a) = k(s,a)(f')^{-1}\left(\frac{e_\nu(s,a)}{\alpha} + \frac{\lambda^*(s,a)}{\alpha d^D(s,a)}\right)\ \forall s, a \tag{23}$$

The combination of primary feasibility($w^*(s,a) \geq 0\ \forall s, a$), dual feasibility ($\lambda^*(s,a) \geq 0\ \forall s, a$) and complementary slackness ($w^*(s,a)\lambda^*(s,a) = 0$) gives two conditions: $w^* > 0, \lambda^* = 0$ and $w^* = 0, \lambda^* \geq 0$. Based on the two conditions, the optimal w^* is given as

$$w^*(s,a) = \max\left(0, k(s,a)(f')^{-1}\left(\frac{e_\nu(s,a)}{\alpha}\right)\right) \tag{24}$$

where $\alpha f'(0) - e_\nu(s,a) \geq 0$ should be satisfied when $w^*(s,a) = 0$. This condition is quite challenging to achieve since most f used in f-divergence has negative gradient at 0. Therefore, we refrain from making $w(s,a) = 0$, just as OptiDICE (Lee et al. (2021)) did.

C Proof of Theorem 1

Theorem 1 (Upper bound of statistical error)**.** *Under some mild assumptions, the statistical error satisfies* $\left|\hat{L}_{\alpha,k}(w,\nu) - L_{\alpha,k}(w,\nu)\right| \leq \mathcal{E}$*:*

$$\mathcal{E} := \mathcal{O}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta_1}}{N}}\right) + \tilde{\mathcal{O}}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta_2}}{N}}\right), \tag{15}$$

where N is the number of samples and δ_1 and δ_2 can be lower-bounded in terms of pseudo-dimension.(Proof in Appendix C.)

Proof. To follow a proof protocol of Nachum et al. (2019a), we introduce some assumptions and lemmas for calculating the upper bound of statistical error using covering number and pseudo-dimension.

Assumption 1. $\nu_0(s) > 0, r(s,a) \in [0,1], \forall s \in S, \forall a \in A.$

Assumption 2 (Boundedness of w)**.** *suppose 0* $\leq w(s,a) \leq B_{\mathcal{W}}$ *for any* $s \in S, a \in A, w \in \mathcal{W}$*, where* $\mathcal{W}$ *is parameterization family of* w*.*

Assumption 3 (Boundedness of k)**.** *suppose* $k_{\min} \leq k(s,a) \leq k_{\max}$ *for any* $s \in S, a \in A, k \in \mathcal{K}$*, where* $\mathcal{K}$ *is parameterization family of* k*.*

Assumption 4 (Properties of f)**.** *Suppose* f *satisfies the following properties:*

1) Boundness:

$$|f'(x)| \leq B_{f'}, \forall 0 \leq x \leq B_w$$
$$|f(x)| \leq B_f, \forall 0 \leq x \leq B_w$$

2) L-Lipschitz continuous: *f is L-Lipschitz continuous function on* $0 \leq x \leq B_w$*.*

Assumption 5 (Boundedness of ν)**.** *Suppose* $\|v\|_\infty \leq B_{\mathcal{V}} := \frac{\alpha B_{f'}+1}{1-\gamma}$ *for any* $\nu \in \mathcal{V}$*, where* $\mathcal{V}$ *is parameterization family of* ν*.*

We recall $L_{\alpha,k}(w,\nu)$ and its empirical version for convenience:

$$\begin{aligned} L_{\alpha,k}(w,\nu) := {} & (1-\gamma)\mathbb{E}_{s\sim\mu_0}[\nu(s)] \\ & - \alpha\mathbb{E}_{(s,a)\sim d^D}\left[k(s,a)f\left(\frac{w(s,a)}{k(s,a)}\right)\right] + \mathbb{E}_{(s,a)\sim d^D}\left[w(s,a)e_\nu(s,a)\right], \end{aligned} \tag{25}$$

and

$$\hat{L}_{\alpha,k} := (1-\gamma)\frac{1}{n_0}\sum_{j=1}^{n_0}\nu(s_{0,j}) + \frac{1}{n}\sum_{i=1}^{n}\left[-\alpha k(s_i,a_i)f\left(\frac{w(s_i,a_i)}{k(s_i,a_i)}\right) + w(s_i,a_i)e_\nu(s_i,a_i,r_i,s_i')\right]. \tag{26}$$

The statistical error can be decomposed to

$$\left|\hat{L}_{\alpha,k} - L_{\alpha,k}\right| \leq \underbrace{\left|\frac{1}{n}\sum_{i=1}^{n} l_i^{w,\nu,k} - \mathbb{E}[l^{w,\nu,k}]\right|}_{\epsilon_1} + \underbrace{(1-\gamma)\left|\frac{1}{n_0}\sum_{j=1}^{n_0}\nu(s_{0,j}) - \mathbb{E}_{s\sim\mu_0}[\nu(s)]\right|}_{\epsilon_2}, \tag{27}$$

where $l^{w,\nu,k} = -\alpha k(s,a)f\left(\frac{w(s,a)}{k(s,a)}\right) + w(s,a)e_\nu(s,a,r,s')$.

We'll find each terms ϵ_1 and ϵ_2 separately, for defining the upper bound of error.

We will need Pollard's tail inequality that relates maximum deviation to the covering number of a function class:

Lemma 2 (Pollard (2012)). *Let $\mathcal{G}$ be a permissible class of $\mathcal{Z} \to [-M, M]$ functions and $\{Z_i\}_{i=1}^N$ are $i.i.d.$ samples from some distribution. Then, for any given $\epsilon > 0$,*

$$\mathbb{P}\left(\sup_{g\in\mathcal{G}}\left|\frac{1}{N}\sum_{i=1}^{N} g(Z_i) - \mathbb{E}[g(Z)]\right| > \epsilon\right) \leq 8\mathbb{E}\left[\mathcal{N}_1\left(\frac{\epsilon}{8}, \mathcal{G}, \{Z_i\}_{i=1}^N\right)\right] \exp\left(\frac{-N\epsilon^2}{512M^2}\right).$$

The covering number can then be bounded in terms of the function class's pseudo-dimension:

Lemma 3 (Corollary 3, Haussler (1995)). *For any set $\mathcal{X}$, any points $x^{1:N} \in \mathcal{X}^N$, any class $\mathcal{F}$ of functions on $\mathcal{X}$ taking values in $[0, M]$ with pseudo-dimension $D_{\mathcal{F}} < \infty$, and any $\epsilon > 0$.*

$$\mathcal{N}_1(\epsilon, \mathcal{F}, x^{1:N}) \geq e(D_{\mathcal{F}} + 1)\left(\frac{2eM}{\epsilon}\right)^{D_{\mathcal{F}}}.$$

With the above technical lemmas, we are ready to bound $\left|\hat{L}_{\alpha,k} - L_{\alpha,k}\right|$.

To bound the term $\left|\hat{L}_{\alpha,k} - L_{\alpha,k}\right|$, we first define the statistical error ϵ_2 and then ϵ_1 for convenience.

Lemma 4 (Statistical error ϵ_1). *Under Assumption 1 to 5, with at least probability $1 - \delta$,*

$$\epsilon_1 = \mathcal{O}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta}}{N}}\right). \tag{28}$$

Proof. Recall $l^{w,\nu,k}(s, a, s') = -\alpha k(s, a) f\left(\frac{w(s,a)}{k(s,a)}\right) + w(s, a)e_\nu(s, a, s')$, we use lemma 2 with $Z = S \times A \times S$, $Z = (s_i, a_i, s_i')$ and $\mathcal{G} = l^{\mathcal{W}\times\mathcal{V}\times\mathcal{K}}$. We first show that $\forall l^{w,\nu,k} \in \mathcal{G}$ is bounded:

$$\begin{aligned}
\|l^{w,\nu,k}\|_\infty &\leq \alpha\|k\|_\infty\|f(\frac{w}{k})\|_\infty + \|w\|_\infty\|e_\nu\|_\infty &(29)\\
&\leq \alpha k_{\max}\left(\|f(\frac{w}{k}) - f(0)\|_\infty + |f(0)|\right) + B_w\left((1+\gamma)B_\nu + 1\right) &(30)\\
&\leq \alpha k_{\max}(L\|\frac{w}{k}\|_\infty + |f(0)|) + B_w\left((1+\gamma)B_\nu + 1\right) &(31)\\
&\leq \alpha\frac{Lk_{\max}}{k_{\min}}\|w\|_\infty + \alpha k_{\max}|f(0)| + B_w\left((1+\gamma)B_\nu + 1\right) &(32)\\
&\leq \alpha\frac{Lk_{\max}B_w}{k_{\min}} + B_w\left((1+\gamma)B_\nu + 1\right) + \alpha k_{\max}|f(0)| = M_1 &(33)
\end{aligned}$$

Applying lemma 2,

$$\begin{aligned}
\mathbb{P}\left(\sup_{g\in\mathcal{G}}\left|\frac{1}{N}\sum_{i=1}^{N} g(Z_i) - \mathbb{E}[g(Z)]\right| > \epsilon\right) &\leq \mathbb{P}\left(\sup_{g\in\mathcal{G}}\left|\frac{1}{N}\sum_{i=1}^{N} l^{w,\nu,k}(Z_i) - \mathbb{E}[l^{w,\nu,k}]\right| > \epsilon\right) &(34)\\
&\leq 8\mathbb{E}\left[\mathcal{N}_1\left(\frac{\epsilon}{8}, \mathcal{G}, \{Z_i\}_{i=1}^N\right)\right] \exp\left(\frac{-N\epsilon^2}{512M^2}\right). &(35)
\end{aligned}$$

We bound the distance in $\mathcal{G}$,

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\left|l^{w_1,\nu_1,k_1}(Z_i)-l^{w_2,\nu_2,k_2}(Z_i)\right| \\
&\leq \frac{1}{N}\frac{\alpha L B_w}{k_{\min}}\sum_{i=1}^{N}|k_1(s_i,a_i)-k_2(s_i,a_i)|+\frac{1}{N}\left(\frac{\alpha L k_{\max}}{k_{\min}}+B_{e_\nu}\right)\sum_{i=1}^{N}|w_1(s_i,a_i)-w_2(s_i,a_i)| \\
&+\frac{B_w}{N}\sum_{i=1}^{N}|\nu_1(s_i)-\nu_2(s_i)|+\frac{\gamma B_w}{N}\sum_{i=1}^{N}|\nu_1(s_i')-\nu_2(s_i')|. \qquad (36)
\end{aligned}$$

This leads to following inequality:

$$\begin{aligned}
&\mathcal{N}_1\Bigg(\Bigg(\Bigg(\frac{\alpha L}{k_{\min}}+1+\gamma)\Bigg)B_w+\frac{\alpha L k_{\max}}{k_{\min}}+B_{e_\nu}\Bigg)\epsilon',\mathcal{G},\{Z_i\}_{i=1}^{N}\Bigg) \\
&\leq \mathcal{N}_1\left(\epsilon',\mathcal{K},\{s_i,a_i\}_{i=1}^{N}\right)\mathcal{N}_1\left(\epsilon',\mathcal{W},\{s_i,a_i\}_{i=1}^{N}\right)\mathcal{N}_1\left(\epsilon',\mathcal{V},\{s_i\}_{i=1}^{N}\right)\mathcal{N}_1\left(\epsilon',\mathcal{V},\{s_i'\}_{i=1}^{N}\right). \qquad (37)
\end{aligned}$$

Using lemma 3, we bound the covering number. Denote pseudo-dimension of $\mathcal{W},\mathcal{V}$ and $\mathcal{K}$ as $D_{\mathcal{W}},D_{\mathcal{V}}$ and $D_{\mathcal{K}}$, we have

$$\begin{aligned}
&\mathcal{N}_1\left(\left(\left(\frac{\alpha L}{k_{\min}}+1+\gamma\right)B_w+\frac{\alpha L k_{\max}}{k_{\min}}+B_{e_\nu}\right)\epsilon',\mathcal{G},\{Z_i\}_{i=1}^{N}\right) \\
&\leq e^4(D_{\mathcal{K}}+1)(D_{\mathcal{W}}+1)(D_{\mathcal{V}}+1)^2\left(\frac{4eM_1}{\epsilon'}\right)^{D_{\mathcal{K}}+D_{\mathcal{W}}+2D_{\mathcal{V}}}. \qquad (38)
\end{aligned}$$

This leads to following inequality:

$$\begin{aligned}
&\mathcal{N}_1\left(\frac{\epsilon}{8},\mathcal{G},\{Z_i\}_{i=1}^{N}\right) \\
&\leq e^4(D_{\mathcal{K}}+1)(D_{\mathcal{W}}+1)(D_{\mathcal{V}}+1)^2\left(\frac{32\left(\left(\frac{\alpha L}{k_{\min}}+1+\gamma\right)B_w+\frac{\alpha L k_{\max}}{k_{\min}}+B_{e_\nu}\right)eM_1}{\epsilon}\right)^{D_{\mathcal{K}}+D_{\mathcal{W}}+2D_{\mathcal{V}}} \qquad (39) \\
&:= C_1\left(\frac{1}{\epsilon}\right)^{D_1}, \qquad (40)
\end{aligned}$$

where $C_1=e^4(D_{\mathcal{K}}+1)(D_{\mathcal{W}}+1)(D_{\mathcal{V}}+1)^2\left(32\left(\left(\frac{\alpha L}{k_{\min}}+1+\gamma\right)B_w+\frac{\alpha L k_{\max}}{k_{\min}}+B_{e_\nu}\right)\right)^{D_1}$ and $D_1=D_{\mathcal{K}}+D_{\mathcal{W}}+2D_{\mathcal{V}}$.

Combine this result with equation 35, we immediately obtain the statistical error,

$$\mathbb{P}\left(\sup_{w\in\mathcal{W},\nu\in\mathcal{V},k\in\mathcal{K}}\left|\hat{l}(w,\nu,k)-l(w,\nu,k)\right|\geq\epsilon\right)\leq 8C_1\left(\frac{1}{\epsilon}\right)^{D_1}\exp\left(\frac{-N\epsilon^2}{512M_1^2}\right). \qquad (41)$$

By setting $\epsilon=\sqrt{\frac{C_2(\log N+\log\frac{1}{\delta})}{N}}$ with $C_2=\max\left((8C_1)^{\frac{2}{D_1}},512M_1D_1,512M_1,1\right)$, we have

$$8C_1\left(\frac{1}{\epsilon}\right)^{D_1}\exp\left(\frac{-N\epsilon^2}{512M_1^2}\right)\leq\delta. \qquad (42)$$

Therefore, we have $\epsilon_1=\mathcal{O}\left(\sqrt{\frac{\log N+\log\frac{1}{\delta}}{N}}\right)$, with $1-\delta$ probability.

□

Lemma 5 (Statistical error ϵ_2). *With at least probability* $1-\delta$,

$$\epsilon_2 = \mathcal{O}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta}}{N}}\right) \tag{43}$$

Proof. We first recall that $\forall \nu \in \mathcal{V}$, ν is bounded by $M_2 = B_\nu$. Then, we apply the lemma 2 with $\mathcal{Z} = S$, $Z_i = (s_{0i})$, and $\mathcal{G} = \nu$,

$$\mathbb{P}\left(\sup_{\nu\in\mathcal{V}}\left|\hat{\mathbb{E}}_{\mathcal{Z}}\left[(1-\gamma)\nu\right] - \mathbb{E}[(1-\gamma)\nu]\right| \geq \epsilon\right) \leq 8\mathbb{E}\left[\mathcal{N}_1\left(\frac{\epsilon}{8}, \mathcal{G}, \{Z_i\}_{i=1}^N\right)\right]\exp\left(\frac{-N\epsilon^2}{512M_2^2}\right). \tag{44}$$

Similarly with derivation of Lemma 4, we have

$$\frac{1-\gamma}{N}\sum_{i=1}^N |\nu_1(\mathcal{Z}_i) - \nu_2(\mathcal{Z}_i)| = \frac{1-\gamma}{N}\sum_{i=1}^N |\nu_1(s_{0i}) - \nu_2(s_{0i})|, \tag{45}$$

leading to

$$\mathcal{N}_1\left((1-\gamma)\epsilon', \mathcal{G}, \{\mathcal{Z}_i\}_{i=1}^N\right) \leq e(D_\mathcal{V}+1)\left(\frac{2eM_2}{\epsilon'}\right)^{D_\mathcal{V}}. \tag{46}$$

This leads to following inequality:

$$\mathcal{N}_1\left(\frac{\epsilon}{8}, \mathcal{G}, \{\mathcal{Z}_i\}_{i=1}^N\right) \leq e(D_\mathcal{V}+1)\left(\frac{16(1-\gamma)eM_2}{\epsilon}\right)^{D_\mathcal{V}} := C_3\left(\frac{1}{\epsilon}\right)^{D_2}, \tag{47}$$

with $C_3 := e(D_\mathcal{V}+1)(16(1-\gamma)eM_2)^{D_2}$ and $D_2 = D_\mathcal{V}$.
We can obtain the statistical error,

$$\mathbb{P}(\epsilon_1 \geq \epsilon) \leq 8C_3\left(\frac{1}{\epsilon}\right)^{D_2}\exp\left(\frac{-N\epsilon^2}{512M_2^2}\right). \tag{48}$$

By setting $\epsilon = \sqrt{\frac{C_4(\log N + \log\frac{1}{\delta})}{N}}$ with $C_4 = \max\left((8C_3)^{\frac{2}{D_2}}, 512M_2D_2, 512M_2, 1\right)$, we have

$$8C_3\left(\frac{1}{\epsilon}\right)^{D_2}\exp\left(\frac{-N\epsilon^2}{512M_2^2}\right) \leq \delta. \tag{49}$$

Therefore, we have $\epsilon_2 = \mathcal{O}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta}}{N}}\right)$, with $1-\delta$ probability.

□

By using Lemma 4 and 5, we can define the total statistical error 27:

$$\left|\hat{L}_{\alpha,k} - L_{\alpha,k}\right| \leq \mathcal{O}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta_1}}{N}}\right) + \tilde{\mathcal{O}}\left(\sqrt{\frac{\log N + \log\frac{1}{\delta_2}}{N}}\right), \tag{50}$$

where $8C_1\left(\frac{1}{\epsilon_1}\right)^{D_1}\exp\left(\frac{-N\epsilon_1^2}{512M_1^2}\right) \leq \delta_1$ and $8C_3\left(\frac{1}{\epsilon_2}\right)^{D_2}\exp\left(\frac{-N\epsilon_2^2}{512M_2^2}\right) \leq \delta_2$.

□

C.1 Strong convexity of perspective function

If f is a M_f-strongly convex function and $k > 0$, the *perspective function* $g(x) = kf\left(\frac{x}{k}\right)$ is a $\frac{M_f}{k}$-strongly convex in x, where k is fixed.

Proof. M_f-strongly convex f satisfy following inequality for all points x, y in its domain:

$$\|f'(x) - f'(y)\| \geq M_f \|x - y\|, \tag{51}$$

where f'_x and f'_y are partial derivative of function f of x and y. Instead of x, y, inserting $\frac{x}{k}, \frac{y}{k}$ is also valid, where k is constant and positive:

$$\left\|f'\left(\frac{x}{k}\right) - f'\left(\frac{y}{k}\right)\right\| \geq M_f \left\|\frac{x}{k} - \frac{y}{k}\right\|. \tag{52}$$

The abbreviation of $1/k$ on the left and right equation is as follows:

$$\left\|f'\left(\frac{x}{k}\right) - f'\left(\frac{y}{k}\right)\right\| \geq \frac{M_f}{k} \left\|x - y\right\|. \tag{53}$$

Note that $g'(x) = f'\left(\frac{x}{k}\right)$. We have

$$\left\|g'(x) - g'(y)\right\| \geq \frac{M_f}{k} \left\|x - y\right\|. \tag{54}$$

□

D WEIGHTED f-DIVERGENCE FOR TABULAR CASE

In tabular settings, MLE MDP $\hat{M} = \langle S, A, T, R, \mu_0, \gamma \rangle$ can be obtained from samples in the dataset as in (Laroche et al., 2019; Lee et al., 2020). From MLE MDP, we obtain the following parameters needed for DICE via f-divergence: initial state distibution $\mu_0(s) \in \mathbb{R}^{|S|}$, transition probability $T(s'|s,a) \in \mathbb{R}^{|S||A| \times |S|}$, reward $R(s,a) \in \mathbb{R}^{|S||A|}$ and dataset distribution $d^D(s,a) \in \mathbb{R}^{|S||A|}$. We also define diagonalized $d^D(s,a)$ as $D = \text{diag}(d^D(s,a)) \in \mathbb{R}^{|S||A| \times |S||A|}$.

Based on given parameters and weighted f-divergence regularization framework 6, we obtain DICE via weighted f-divergence for finite case. We assume $f(x) = \frac{1}{2}(x-1)^2$ for f-divergence to ease vector derivative.

$$\max_{K, w \geq 0} \min_{\nu} L_{\alpha,K}(w, \nu) = (1-\gamma)\mu_0^\top \nu - \frac{\alpha}{2} K (K^{-1}w - 1)^\top D (K^{-1}w - 1) + w^\top D e_\nu \tag{55}$$

where $K = \text{diag}(k(s,a)) \in \mathbb{R}^{|S||A| \times |S||A|}$ is set to represent element-wise multiplication. $\nu, w \in \mathbb{R}^{|S||A|}$ are represented as $|S||A|$-dimensional vectors. We denote $\mathcal{T} \in \mathbb{R}^{|S||A|}$ and $\mathcal{B} \in \mathbb{R}^{|S||A|}$ as a matrix, which satisfy $(\mathcal{T}\nu)((s,a)) = \sum_{s'} T(s'|s,a)\nu(s')$ and $(\mathcal{B}\nu)((s,a)) = \nu(s)$ respectively.

First, we assume K is fixed and optimize over w and ν. We switch $\max_w$ and $\min_\nu$ using strong duality, then obtain optimal $w^*_{\nu,K}$ of inner maximization of w.

$$w^*_{\nu,K} = \max(0, K(\frac{e_\nu}{\alpha} + 1)) \tag{56}$$

Given $w^*_{\nu,K}$ and K, we minimize 55 over ν. The minimization problem is given as:

$$\min_{\nu} L(w^*_{\nu,K}, \nu) = (1-\gamma)\mu_0^\top \nu - \frac{\alpha}{2} K (K^{-1}w^*_{\nu,K} - 1)^\top D (K^{-1}w^*_{\nu,K} - 1) + {w^*_{\nu,k}}^\top D e_\nu \tag{57}$$

However, unlike w, the closed form solution of ν is difficult to find. Therefore, we compute gradient of $L(w^*_{\nu,K}, \nu)$. We also obtain gradient of e_ν and $w^*_{\nu,K}$

$$\frac{\partial e_\nu}{\partial \nu} = \gamma \mathcal{T} - \mathcal{B}, \tag{58}$$

$$\frac{\partial w^*_{\nu,K}}{\partial \nu} = K \left(\frac{1}{\alpha} \gamma \mathcal{T} - \mathcal{B} \right) \odot \mathbb{1}(K(\frac{e_\nu}{\alpha} + 1) \geq 0). \tag{59}$$

By using 58 and 59, we can calculate

$$\frac{\partial L_{\alpha,K}(w,\nu)}{\partial \nu} = (1-\gamma)\mu_0 - \alpha {\frac{\partial w^*_{\nu,K}}{\partial \nu}}^\top D(K^{-1}w^*_{\nu,k} - 1) + {\frac{\partial w^*_{\nu,K}}{\partial \nu}}^\top D e_\nu + (\gamma \mathcal{T} - \mathcal{B})^\top D w^*_{\nu,k}, \tag{60}$$

where $\mathbb{1}$ denotes one vector and $\odot$ denotes row-wise masking. Now we maximize over K, and it is equivalent to minimizing weighted divergence term in 55. The optimal solution of K is given as

$$K^* = \text{diag}(w) \tag{61}$$

One can replace K with diag(w) or gradually update K toward diag(w) with loss function of $(k - w)^2$. However, for both cases, it is important to keep each elements K upper-bounded by $k_{\max}$, to limit upper bound of sampling error. With given update rules for w, ν and K, we iterate DICE via weighted f-divergence in finite domain until convergence.

E Experiment details for D4RL

For DICE via weighted f-divergence in continuous domain, we have two types of parameterized model: value network and policy network. Both networks have fully-connected multi-layer perceptrons (MLP) with two hidden layers which has 256 hidden units on each layers that takes the state-action pairs as input and outputs network value through the ReLU activation functions. We use value networks to estimate value functions $\nu(s_0)$, $e_\nu(s,a)$ and $k(s,a)$ and policy networks to estimate data collection policy π_D and behavior policy π_ψ. For policy network, we use tanh-squashed mixture of normal distributions where their means and standard deviation are obtained from the MLP.

We adopt stochastic gradient descent with Adam optimizer with learning rate {3e-4, 3e-5} to update the networks. For training, we use 1,200,000 total iteration and 200,000 warm-up iterations, where policy network for behavior policy π_ψ will start its training after the warm-up. We choose between the two types of k, which takes state-action or state as input to obtain weight k. $k(s)$ is introduced to prevent over-parameterization where additional $|S||A|$ variables can be excessive for the optimization. We clipped k using $k_{\min}$ and $k_{\max}$. $k_{\max}$ is set to bound the sampling error, while $k_{\min}$ is set to ensure positivity of k. In addition, when learning k, additional regularization to one ($(k-1)^2$) was given to the original loss function of k. With its degree scaled by a hyperparameter k_{reg}, the regularization can provide gradient to clipped k and prevent excessive deviation of $k(s,a)d^D(s,a)$ from $d^D(s,a)$. K denotes the number of mixtures for tanh-squashed mixture of Gaussians policy. For discount factor γ, we use 0.99. Table 2 includes the details of settings and hyperprameters that we used on experiments of D4RL benchmark (Fu et al. (2020)). We follow the settings of K and α proposed by OptiDICE(Lee et al. (2021)).

Table 2: **(D4RL benchmark)** Hyperparampeters and detail settings on the experiments of D4RL benchmark (Fu et al. (2020)).

Env	**Type**	**K**	α	k	$k_{\min}$	$k_{\max}$	k_{reg}	**lr**
Maze2d	Umaze	5	0.001	k(s)	0.5	2	0.1	3e-5
Maze2d	Medium	5	0.0001	k(s)	0.5	2	0.1	3e-5
Maze2d	Large	1	0.01	k(s)	0.5	2	0.1	3e-5
Hopper	Random	5	1	k(s,a)	0.0001	2	100	3e-4
Hopper	Medium	9	0.1	k(s)	0.5	2	0.1	3e-4
Hopper	Med-replay	9	10	k(s)	0.5	2	0.1	3e-4
Hopper	Med-expert	9	1	k(s,a)	0.5	2	1	3e-5
Walker2d	Random	9	0.0001	k(s,a)	0.001	2	10	3e-4
Walker2d	Medium	9	0.01	k(s,a)	0.8	1.2	1	3e-4
Walker2d	Med-replay	9	0.1	k(s,a)	0.5	2	0.1	3e-4
Walker2d	Med-expert	5	0.01	k(s,a)	0.5	2	0.1	3e-5
Halfcheetah	Random	5	0.0001	k(s,a)	0.0001	2	0.1	3e-4
Halfcheetah	Medium	1	0.01	k(s,a)	0.0001	2	1	3e-4
Halfcheetah	Med-replay	9	0.01	k(s,a)	0.0001	2	1.0	3e-5
Halfcheetah	Med-expert	9	0.01	k(s)	0.8	1.2	100	3e-4

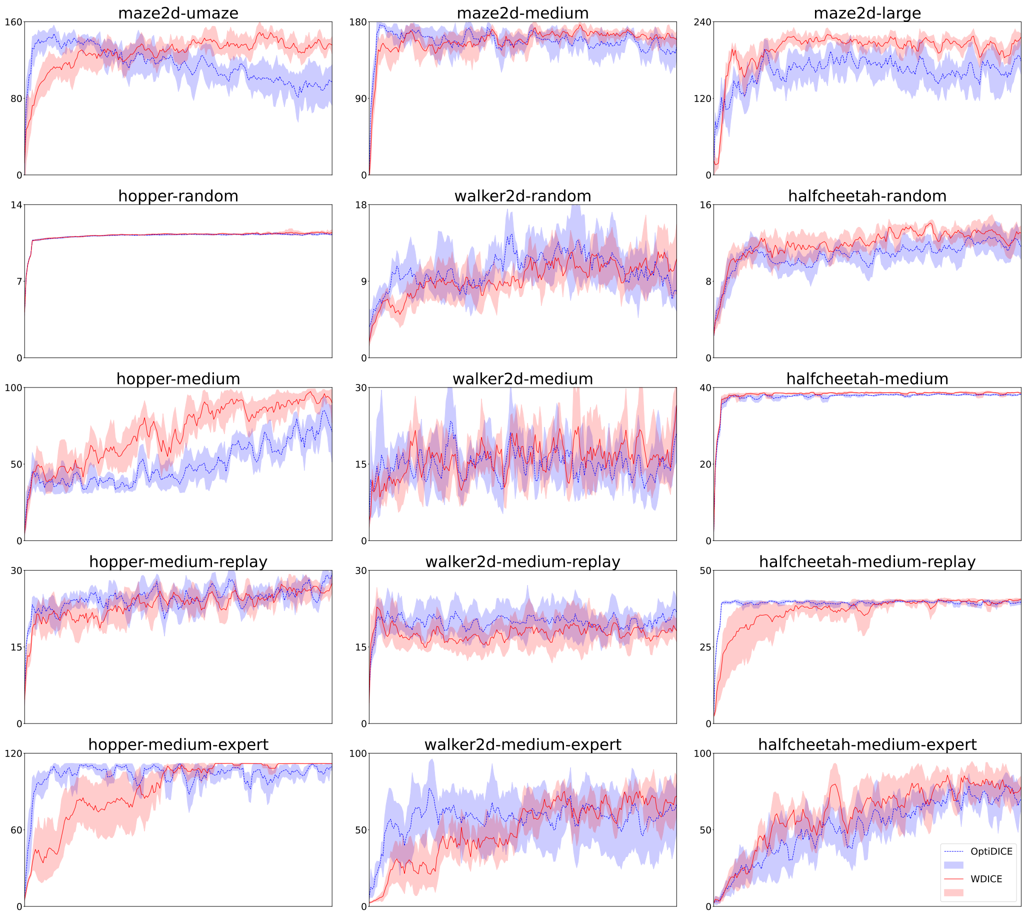


Figure 3: **(D4RL benchmark)** Performance of our algorithm compare to OptiDICE (Lee et al. (2021)) on D4RL benchmark (Fu et al. (2020)). We run 5 times and show means and standard errors with 95 % confidence intervals. x-axis and y-axis represent normalized score and the number of iteration, respectively.

F Ablation Study

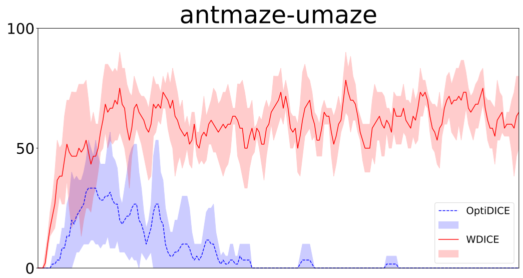


Figure 4: Performance of our algorithm compare to OptiDICE on antmaze-umaze of gym-mujoco with same hyperparameters.

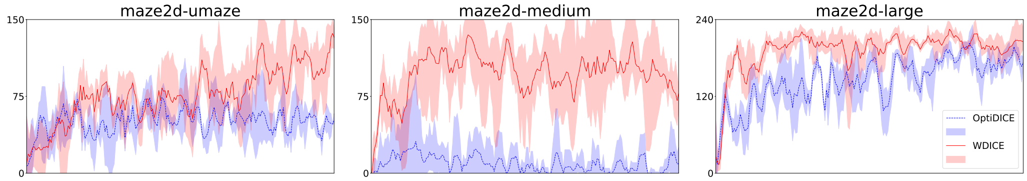


Figure 5: Performance of our algorithm compare to OptiDICE for $f = \frac{1}{2}(x-1)^2$ (i.e, chi-square), on maze2d-umaze, maze2d-medium and maze2d-large of D4RL benchmark.

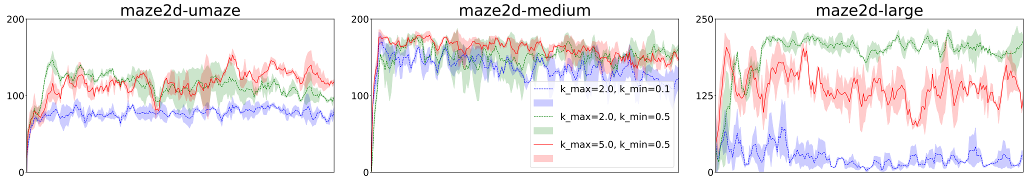


Figure 6: Performance of our algorithm for $(k_{\max}, k_{\min}) \in \{(2.0, 0.1), (2.0, 0.5), (5.0, 0.5)\}$ on maze2d-umaze, maze2d-medium and maze2d-large of D4RL benchmark.

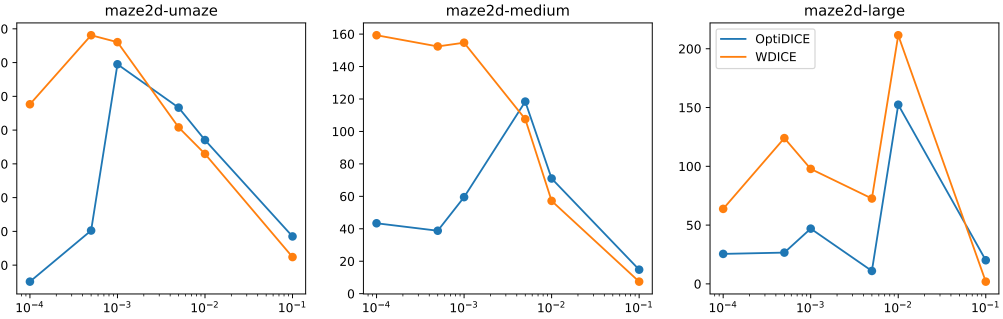


Figure 7: Performance of our algorithm compare to OptiDICE with respect to α with fixed bound on $(k_{\max}, k_{\min}) = (2.0, 0.5)$ on maze2d-umaze, maze2d-medium and maze2d-large of D4RL benchmark. We set $\alpha \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1\}$.