OptiDICE: Offline Policy Optimization via Stationary Distribution Correction Estimation (Supplementary Material)

A. Proof of Proposition 1

We first show that our original problem (2-4) is an instance of convex programming due to the convexity of f.

Lemma 5. *The constraint optimization (2-4) is a convex optimization.*

Proof.

$$\max_{d} \ \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f(d||d^D) \tag{2}$$

$$\text{s.t. } (\mathcal{B}_* d)(s) = (1-\gamma)p_0(s) + \gamma(\mathcal{T}_* d)(s) \ \ \forall s, \tag{3}$$

$$d(s,a) \geq 0 \ \ \forall s, a, \tag{4}$$

The objective function $\mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f(d||d^D)$ is concave for $d : S \times A \to \mathbb{R}$ (not only for probability distribution $d \in \Delta(S \times A)$) since $D_f(d||d^D)$ is convex in d: for $t \in [0,1]$ and any $d_1 : S \times A \to \mathbb{R}, d_2 : S \times A \to \mathbb{R}$,

$$\begin{aligned} D_f((1-t)d_1 + td_2||d^D) &= \sum_{s,a} d^D(s,a) f\left((1-t)\frac{d_1(s,a)}{d^D(s,a)} + t\frac{d_2(s,a)}{d^D(s,a)}\right) \\ &< \sum_{s,a} d^D(s,a)\left\{(1-t)f\left(\frac{d_1(s,a)}{d^D(s,a)}\right) + tf\left(\frac{d_2(s,a)}{d^D(s,a)}\right)\right\} \\ &= (1-t)D_f(d_1||d^D) + tD_f(d_2||d^D), \end{aligned}$$

where the strict inequality follows from assuming f is strictly convex. In addition, the equality constraints (3) are affine in d, and the inequality constraints (4) are linear and thus convex in d. Therefore, our problem is an instance of a convex programming, as we mentioned in Section 3.1. □

In addition, by using the strong duality and the change-of-variable from d to w, we can rearrange the original maximin optimization to the minimax optimization.

Lemma 6. *We assume that all states $s \in S$ are reachable for a given MDP. Then,*

$$\max_{w\geq 0} \min_{\nu} L(w,\nu) = \min_{\nu} \max_{w\geq 0} L(w,\nu).$$

Proof. Let us define the Lagrangian of the constraint optimization (2-4)

$$\begin{aligned} \mathcal{L}(d,\nu,\mu) := \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f(d||d^D) + \sum_s \nu(s)\left((1-\gamma)p_0(s) + \gamma\sum_{\bar{s},\bar{a}} T(s|\bar{s},\bar{a})d(\bar{s},\bar{a}) - \sum_{\bar{a}} d(s,\bar{a})\right) \\ + \sum_{s,a} \mu(s,a)d(s,a) \end{aligned}$$

with Lagrange multipliers $\nu(s)\ \forall s$ and $\mu(s,a)\ \forall s,a$. With the Lagrangian $\mathcal{L}(d,\nu,\mu)$, the original problem (2-4) can be represented by

$$\max_{d\geq 0} \min_{\nu} \mathcal{L}(d,\nu,0) = \max_{d} \min_{\nu,\mu\geq 0} \mathcal{L}(d,\nu,\mu).$$

For an MDP where every $s \in S$ is reachable, there always exists d such that $d(s,a) > 0 \; \forall s, a$. From Slater's condition for convex problems (the condition that there exists a strictly feasible d (Boyd et al., 2004)), the strong duality holds, i.e., we can change the order of optimizations:

$$\max_{d} \min_{\nu, \mu \geq 0} \mathcal{L}(d, \nu, \mu) = \min_{\nu, \mu \geq 0} \max_{d} \mathcal{L}(d, \nu, \mu) = \min_{\nu} \max_{d \geq 0} \mathcal{L}(d, \nu, 0).$$

Here, the last equality holds since $\max_{d \geq 0} \mathcal{L}(d, \nu, 0) = \max_d \min_{\mu \geq 0} \mathcal{L}(d, \nu, \mu) = \min_{\mu \geq 0} \max_d \mathcal{L}(d, \nu, \mu)$ for fixed ν due to the strong duality. Finally, by applying the change of variable $w = d/d^D$, we have

$$\max_{w \geq 0} \min_{\nu} L(w, \nu) = \min_{\nu} \max_{w \geq 0} L(w, \nu).$$

□

Finally, the solution of the inner maximization $\max_{w \geq 0} L(w, \nu)$ can be derived as follows:

Proposition 1. *The closed-form solution of the inner maximization of* (10)*, i.e.*

$$w^*_\nu := \arg\max_{w \geq 0} (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu(s)] + \mathbb{E}_{(s,a) \sim d^D}\left[-\alpha f\big(w(s,a)\big)\right] + \mathbb{E}_{(s,a) \sim d^D}\left[w(s,a)\big(e_\nu(s,a)\big)\right]$$

is given as

$$w^*_\nu(s,a) = \max\left(0, (f')^{-1}\left(\frac{e_\nu(s,a)}{\alpha}\right)\right) \quad \forall s, a, \tag{32}$$

where $(f')^{-1}$ is the inverse function of the derivative f' of f and is strictly increasing by strict convexity of f.

Proof. For a fixed ν, let the maximization $\max_{w \geq 0} L(w, \nu)$ be the primal problem. Then, its corresponding dual problem is

$$\max_{w} \min_{\mu \geq 0} L(w, \nu) + \sum_{s,a} \mu(s,a) w(s,a).$$

Since the strong duality holds, satisfying KKT condition is both necessary and sufficient conditions for the solutions w^* and μ^* of primal and dual problems (we will use w^* and μ^* instead of w^*_ν and μ^*_ν for notational brevity).

Condition 1 (Primal feasibility). $w^* \geq 0 \; \forall s, a$.

Condition 2 (Dual feasibility). $\mu^* \geq 0 \; \forall s, a$.

Condition 3 (Stationarity). $d^D(s,a)(-\alpha f'(w^*(s,a)) + e_\nu(s,a) + \mu^*(s,a)) = 0 \; \forall s, a$.

Condition 4 (Complementary slackness). $w^*(s,a)\mu^*(s,a) = 0 \; \forall s, a$.

From **Stationarity** and $d^D > 0$, we have

$$f'(w^*(s,a)) = \frac{e_\nu(s,a) + \mu^*(s,a)}{\alpha} \; \forall s, a$$

and since f' is invertible due to the strict convexity of f,

$$w^*(s,a) = (f')^{-1}\left(\frac{e_\nu(s,a) + \mu^*(s,a)}{\alpha}\right) \; \forall s, a.$$

Now for fixed $(s,a) \in S \times A$, let us consider two cases: either $w^*(s,a) > 0$ or $w^*(s,a) = 0$, where **Primal feasibility** is always satisfied in either way:

Case 1 ($w^*(s,a) > 0$). $\mu^*(s,a) = 0$ due to **Complementary slackness**, and thus,

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

Note that **Dual feasibility** holds. Since f' is a strictly increasing function, $e_\nu(s,a) > \alpha f'(0)$ should be satisfied if $f'(0)$ is well-defined.

Case 2 ($w^*(s,a)=0$). $\mu^*(s,a) = \alpha f'(0) - e_\nu(s,a) \geq 0$ due to **Stationarity** and **Dual feasibility**, and thus, $e_\nu(s,a) \leq \alpha f'(0)$ should be satisfied if $f'(0)$ is well-defined.

In summary, we have

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

□

B. Proofs of Proposition 2 and Corollary 3

Proposition 2. *$L(w^*_\nu, \nu)$ is convex with respect to ν.*

Proof by Lagrangian duality. Let us consider Lagrange dual function

$$g(\nu,\mu) := \max_d \mathcal{L}(d,\nu,\mu),$$

which is always convex in Lagrange multipliers ν, μ since $\mathcal{L}(d,\nu,\mu)$ is affine in ν,μ. Also, for any $\mu_1, \mu_2 \geq 0$ and its convex combination $(1-t)\mu_1 + t\mu_2$ for $0 \leq t \leq 1$, we have

$$\min_{\mu\geq 0} g((1-t)\nu_1 + t\nu_2, \mu) \leq g((1-t)\nu_1 + t\nu_2, (1-t)\mu_1 + t\mu_2) \leq (1-t)g(\nu_1,\mu_1) + tg(\nu_2,\mu_2)$$

by using the convexity of $g(\nu,\mu)$. Since the above statement holds for any $\mu_1, \mu_2 \geq 0$, we have

$$\min_{\mu\geq 0} g((1-t)\nu_1 + t\nu_2, \mu) \leq (1-t)\min_{\mu_1\geq 0} g(\nu_1,\mu_1) + t\min_{\mu_2\geq 0} g(\nu_2,\mu_2).$$

Therefore, a function

$$G(\nu) := \min_{\mu\geq 0} g(\nu,\mu) = \min_{\mu\geq 0}\max_d \mathcal{L}(d,\nu,\mu) = \max_{d\geq 0}\mathcal{L}(d,\nu,0)$$

is convex in ν. By following the change-of-variable, we have

$$\max_{d\geq 0}\mathcal{L}(d,\nu,0) = \max_{w\geq 0} L(w,\nu) = L(\arg\max_{w\geq 0} L(w,\nu),\nu) = L(w^*_\nu,\nu)$$

is convex in ν.

□

Proof by exploiting second-order derivative. Suppose $((f')^{-1})'$ is well-defined, where f we consider in this work satisfies the condition. Let us define

$$h(x) := -f\Big(\max\Big(0,(f')^{-1}(x)\Big)\Big) + \max\Big(0,(f')^{-1}(x)\Big)\cdot x. \tag{33}$$

Then, $L(w^*_\nu,\nu)$ can be represented by using h:

$$\begin{aligned}
&L(w^*_\nu,\nu)\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\Big[-\alpha f\Big(\max\Big(0,(f')^{-1}\big(\tfrac{1}{\alpha}e_\nu(s,a)\big)\Big)\Big) + \max\Big(0,(f')^{-1}\big(\tfrac{1}{\alpha}e_\nu(s,a)\big)\Big)e_\nu(s,a)\Big]\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\Big[\alpha h\Big(\tfrac{1}{\alpha}e_\nu(s,a)\Big)\Big]
\end{aligned} \tag{34}$$

We prove that $h(x)$ is convex in x by showing $h''(x) \geq 0\ \forall x$. Recall that f' is a strictly increasing function by the strict convexity of f, which implies that $(f')^{-1}$ is also a strictly increasing function.

Case 1. If $(f')^{-1}(x) > 0 \; \forall x$,

$$\begin{aligned} h(x) &= -f((f')^{-1}(x)) + (f')^{-1}(x) \cdot x, \\ h'(x) &= -\underbrace{f'((f')^{-1}(x))}_{\text{(identity function)}}((f')^{-1})'(x) + ((f')^{-1})'(x) \cdot x + (f')^{-1}(x) \\ &= -x \cdot ((f')^{-1})'(x) + ((f')^{-1})'(x) \cdot x + (f')^{-1}(x) \\ &= (f')^{-1}(x), \\ h''(x) &= ((f')^{-1})'(x) > 0, \end{aligned}$$

where $((f')^{-1})'(x) > 0$ since it is the derivative of the strictly increasing function $(f')^{-1}$.

Case 2. If $(f')^{-1}(x) \leq 0 \; \forall x$,

$$h(x) = -f(0) \;\; \Rightarrow \;\; h'(x) = 0 \;\; \Rightarrow \;\; h''(x) = 0.$$

Therefore, $h''(x) \geq 0$ holds for all x, which implies that $h(x)$ is convex in x. Finally, for $t \in [0, 1]$ and any $\nu_1 : S \to \mathbb{R}$, $\nu_2 : S \to \mathbb{R}$,

$$\begin{aligned} &L(w^*_{t\nu_1+(1-t)\nu_2}, t\nu_1 + (1-t)\nu_2) \\ &= (1-\gamma)\mathbb{E}_{s\sim p_0}[t\nu_1(s) + (1-t)\nu_2(s)] \\ &\quad + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\Big(\tfrac{1}{\alpha}\Big(t\{R(s,a) + \gamma(\mathcal{T}\nu_1)(s,a) - (\mathcal{B}\nu_1)(s,a)\} + (1-t)\{R(s,a) + \gamma(\mathcal{T}\nu_2)(s,a) - (\mathcal{B}\nu_2)(s,a)\}\Big)\Big)\right] \\ &\leq t\left\{(1-\gamma)\mathbb{E}_{s\sim p_0}[\nu_1(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\Big(\tfrac{1}{\alpha}\big(R(s,a) + \gamma(\mathcal{T}\nu_1)(s,a) - (\mathcal{B}\nu_1)(s,a)\big)\Big)\right]\right\} \qquad \text{(by convexity of } h\text{)} \\ &\quad + (1-t)\left\{(1-\gamma)\mathbb{E}_{s\sim p_0}[\nu_2(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\Big(\tfrac{1}{\alpha}\big(R(s,a) + \gamma(\mathcal{T}\nu_2)(s,a) - (\mathcal{B}\nu_2)(s,a)\big)\Big)\right]\right\} \\ &= tL(w^*_{\nu_1}, \nu_1) + (1-t)L(w^*_{\nu_2}, \nu_2) \end{aligned}$$

which concludes the proof. □

Corollary 3. *$\tilde{L}(\nu)$ in (13) is an upper bound of $L(w^*_\nu, \nu)$ in (12), i.e. $L(w^*_\nu, \nu) \leq \tilde{L}(\nu)$ always holds, where equality holds when the MDP is deterministic.*

Proof by Lagrangian duality. Let us consider a function h in (33). From **Proposition 2**, we have $\mathbb{E}_{(s,a)\sim d^D}[h(\frac{1}{\alpha}e_\nu(s,a))]$ is convex in ν, i.e., for $t \in [0, 1]$, $\nu_1 : S \to \mathbb{R}$ and $\nu_2 : S \to \mathbb{R}$,

$$\begin{aligned} \mathbb{E}_{(s,a)\sim d^D}\left[h\left(\tfrac{1}{\alpha}e_{(1-t)\nu_1+t\nu_2}(s,a)\right)\right] &= \mathbb{E}_{(s,a)\sim d^D}\left[h\left((1-t)\cdot\tfrac{1}{\alpha}e_{\nu_1}(s,a) + t\cdot\tfrac{1}{\alpha}e_{\nu_2}(s,a)\right)\right] \\ &\leq (1-t)\mathbb{E}_{(s,a)\sim d^D}\left[h\left(\tfrac{1}{\alpha}e_{\nu_1}(s,a)\right)\right] + t\mathbb{E}_{(s,a)\sim d^D}\left[h\left(\tfrac{1}{\alpha}e_{\nu_2}(s,a)\right)\right] \\ &= \mathbb{E}_{(s,a)\sim d^D}\left[(1-t)\cdot h\left(\tfrac{1}{\alpha}e_{\nu_1}(s,a)\right) + t\cdot h\left(\tfrac{1}{\alpha}e_{\nu_2}(s,a)\right)\right]. \end{aligned}$$

Since **Proposition 2** should be satisfied for any MDP and $d^D > 0$, we have

$$h((1-t)\cdot\tfrac{1}{\alpha}e_{\nu_1}(s,a) + t\cdot\tfrac{1}{\alpha}e_{\nu_2}(s,a)) \leq (1-t)\cdot h\left(\tfrac{1}{\alpha}e_{\nu_1}(s,a)\right) + t\cdot h\left(\tfrac{1}{\alpha}e_{\nu_2}(s,a)\right) \;\forall s, a.$$

To prove this, if

$$h((1-t)\cdot\tfrac{1}{\alpha}e_{\nu_1}(s,a) + t\cdot\tfrac{1}{\alpha}e_{\nu_2}(s,a)) > (1-t)\cdot h\left(\tfrac{1}{\alpha}e_{\nu_1}(s,a)\right) + t\cdot h\left(\tfrac{1}{\alpha}e_{\nu_2}(s,a)\right) \;\exists s, a,$$

we can always find out $d^D > 0$ that contradicts **Proposition 2**. Also, since $\frac{1}{\alpha}e_\nu(s,a)$ can have an arbitrary real value, h should be a convex function. Therefore, it can be shown that

$$h\left(\mathbb{E}_{s'\sim T(s,a)}\left[\tfrac{1}{\alpha}\hat{e}_\nu(s,a,s')\right]\right) \leq \mathbb{E}_{s'\sim T(s,a)}\left[h\left(\tfrac{1}{\alpha}\hat{e}_\nu(s,a,s')\right)\right] \;\forall s, a,$$

due to Jensen's inequality, and thus,

$$\mathbb{E}_{(s,a)\sim d^D}\left[h\left(\tfrac{1}{\alpha}e_\nu(s,a)\right)\right] = \mathbb{E}_{(s,a)\sim d^D}\left[h\left(\mathbb{E}_{s'\sim T(s,a)}\left[\tfrac{1}{\alpha}\hat{e}_\nu(s,a,s')\right]\right)\right] \leq \mathbb{E}_{(s,a,s')\sim d^D}\left[h\left(\tfrac{1}{\alpha}\hat{e}_\nu(s,a,s')\right)\right].$$

Also, the inequality becomes tight when the transition model is deterministic since $h\big(\frac{1}{\alpha}\mathbb{E}_{s'\sim T(s,a)}[\hat{e}(s,a,s')]\big) = \mathbb{E}_{s'\sim T(s,a)}[h\big(\frac{1}{\alpha}\hat{e}(s,a,s')\big)]$ should always hold for the deterministic transition T. □

Proof by exploiting second-order derivative. We start from (34) in the proof of **Proposition 2**.

$$
\begin{aligned}
L(w_\nu^*, \nu) &= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\Big(\tfrac{1}{\alpha}e_\nu(s,a)\Big)\right] && (34)\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\Big(\tfrac{1}{\alpha}\mathbb{E}_{s'\sim T(s,a)}[\hat{e}_\nu(s,a,s')]\Big)\right] \\
&\le (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathop{\mathbb{E}}_{\substack{(s,a)\sim d^D\\ s'\sim T(s,a)}}\left[\alpha h\Big(\tfrac{1}{\alpha}\hat{e}_\nu(s,a,s')\Big)\right] \qquad \text{(by Jensen's inequality with the convexity of } h) \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a,s')\sim d^D}\Bigg[-\alpha f\Big(\max\Big(0,(f')^{-1}\big(\tfrac{1}{\alpha}\hat{e}_\nu(s,a,s')\big)\Big)\Big) \qquad \text{(by definition of } h) \\
&\qquad\qquad + \max\Big(0,(f')^{-1}\big(\tfrac{1}{\alpha}\hat{e}_\nu(s,a,s')\big)\Big)\big(\hat{e}_\nu(s,a,s')\big)\Bigg] = \tilde{L}(\nu) && (13)
\end{aligned}
$$

Also, Jensen's inequality becomes tight when the transition model is deterministic for the same reason we describe in *Proof by Lagrangian duality*. □

C. OptiDICE for Finite MDPs

For tabular MDP experiments, we assume that the data-collection policy is given to OptiDICE for a fair comparison with SPIBB (Laroche et al., 2019) and BOPAH (Lee et al., 2020), which directly exploit the data-collection policy π_D. However, the extension of tabular OptiDICE to not assuming π_D is straightforward.

As a first step, we construct an MLE MDP $\hat{M} = \langle S, A, T, R, p_0, \gamma\rangle$ using the given offline dataset. Then, we compute a stationary distribution of the data-collection policy π_D on the MLE MDP, denoted as d^{π_D}. Finally, we aim to solve the following policy optimization problem on the MLE MDP:

$$\pi^* := \arg\max_\pi \mathbb{E}_{(s,a)\sim d^\pi}[R(s,a)] - \alpha D_f(d^\pi || d^{\pi_D}),$$

which can be reformulated in terms of optimizing the stationary distribution corrections w with Lagrange multipliers ν:

$$\min_\nu \max_{w\ge 0} L(w,\nu) = (1-\gamma)\mathbb{E}_{s\sim p_0(s)}[\nu(s)] + \mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[-\alpha f\big(w(s,a)\big) + w(s,a)\Big(R(s,a) + \gamma(\mathcal{T}\nu)(s,a) - (\mathcal{B}\nu)(s,a)\Big)\Big]. \tag{35}$$

For tabular MDPs, we can describe the problem using vector-matrix notation. Specifically, $\nu \in \mathbb{R}^{|S|}$ is represented as a $|S|$-dimensional vector, $w \in \mathbb{R}^{|S||A|}$ by $|S||A|$-dimensional vector, and $R \in \mathbb{R}^{|S||A|}$ by $|S||A|$-dimensional reward vector. Then, we denote $D = \mathrm{diag}(d^{\pi_D}) \in \mathbb{R}^{|S||A|\times|S||A|}$ as a diagonal matrix, $\mathcal{T} \in \mathbb{R}^{|S||A|\times|S|}$ as a matrix, and $\mathcal{B} \in \mathbb{R}^{|S||A|\times|S|}$ as a matrix that satisfies

$$
\begin{aligned}
\mathcal{T}\nu \in \mathbb{R}^{|S||A|} \quad \text{s.t.} \quad (\mathcal{T}\nu)((s,a)) &= \sum_{s'} T(s'|s,a)\nu(s') \\
\mathcal{B}\nu \in \mathbb{R}^{|S||A|} \quad \text{s.t.} \quad (\mathcal{B}\nu)((s,a)) &= \nu(s)
\end{aligned}
$$

For brevity, we only consider the case where $f(x) = \frac{1}{2}(x-1)^2$ that corresponds to χ^2-divergence-regularized policy optimization, and the problem (35) becomes

$$\min_\nu \max_{w\ge 0} L(w,\nu) = (1-\gamma)p_0^\top v - \frac{\alpha}{2}(w-1)^\top D(w-1) + w^\top D(R + \gamma\mathcal{T}\nu - \mathcal{B}v) \tag{36}$$

From **Proposition 1**, we have the closed-form solution of the inner maximization as $w_\nu^* = \max\big(0, \frac{1}{\alpha}(R+\gamma\mathcal{T}\nu - \mathcal{B}\nu) + 1\big)$ since $(f')^{-1}(x) = x+1$. By plugging w_ν^* into $L(w,\nu)$, we obtain

$$\min_\nu L(w_\nu^*, \nu) = L(\nu) := (1-\gamma)p_0^\top\nu - \frac{\alpha}{2}(w_\nu^* - 1)D(w_\nu^* - 1) + w_\nu^{*\top} D(R + \gamma\mathcal{T}\nu - \mathcal{B}v) \tag{37}$$

Since $L(\nu)$ is convex in ν by **Proposition 2**, we perform a second-order optimization, i.e., Newton's method, to compute an optimal ν^* efficiently. For almost every ν, we can compute the first and second derivatives as follows:

$$
\begin{aligned}
e_\nu :=& R + \gamma \mathcal{T}\nu - \mathcal{B}\nu && \text{(advantage using } \nu\text{)} \\
m :=& \mathbb{1}\left(\tfrac{1}{\alpha} e_\nu + 1 \geq 0\right) && \text{(binary masking vector)} \\
w_\nu^* :=& \left(\tfrac{1}{\alpha} e_\nu + 1\right) \odot m \qquad \text{(where } \odot m \text{ denotes element-wise masking)} && \text{(closed-form solution)} \\
J :=& \frac{\partial w_\nu^*}{\partial \nu} = \tfrac{1}{\alpha}(\gamma\mathcal{T} - \mathcal{B}) \odot m \quad \text{(where } \odot m \text{ denotes row-wise masking)} && \text{(Jacobian matrix)} \\
g :=& \frac{\partial L(\nu)}{\partial \nu} = (1-\gamma)p_0 - \alpha J^\top D(w_\nu^* - 1) + J^\top D e_\nu + (\gamma\mathcal{T} - \mathcal{B})^\top D w_\nu^* && \text{(first-order derivative)} \\
H :=& \frac{\partial^2 L(\nu)}{\partial \nu^2} = -\alpha J^\top D J + J^\top D(\gamma\mathcal{T} - \mathcal{B}) + (\gamma\mathcal{T} - \mathcal{B})^\top D J && \text{(second-order derivative).}
\end{aligned}
$$

We iteratively update ν in the direction of $-H^{-1}g$ until convergence. Finally, $w_{\nu^*}^*$ and the corresponding optimal policy $\pi^*(a|s) \propto w_{\nu^*}^*(s,a) \cdot d^{\pi_D}(s,a)$ are computed. The pseudo-code of these procedures is presented in **Algorithm 2**.

Algorithm 2 Tabular OptiDICE ($f(x) = \frac{1}{2}(x-1)^2$)

Input: MLE MDP $\hat{M} = \langle S, A, \mathcal{T}, r, \gamma, p_0 \rangle$, data-collection policy $\pi_{_D}$, regularization hyperparameter $\alpha > 0$.

$d^{\pi_D} \leftarrow$ COMPUTESTATIONARYDISTRIBUTION$(\hat{M}, \pi_{_D})$
$D \leftarrow \mathrm{diag}(d_{_D}^{\pi})$
$\nu \leftarrow$ (random initialization)
while ν is not converged **do**
  $e \leftarrow r + \gamma \mathcal{T} v - \mathcal{B} v$
  $m \leftarrow \mathbb{1}\left(\frac{1}{\alpha} e_\nu + 1 \geq 0\right)$
  $w \leftarrow \left(\frac{1}{\alpha} e + 1\right) \odot m$
  $J \leftarrow \frac{1}{\alpha}\left(\gamma\mathcal{T} - \mathcal{B}\right) \odot m$
  $g \leftarrow (1-\gamma)p_0 - \alpha J^\top D(w-1) + J^\top D e + (\gamma\mathcal{T} - \mathcal{B})^\top D w$
  $H \leftarrow -\alpha J^\top D J + J^\top D(\gamma\mathcal{T} - \mathcal{B}) + (\gamma\mathcal{T} - \mathcal{B})^\top D J$
  $\nu \leftarrow \nu - \eta H^{-1} g$ (where η is a step-size)
end while

$$\pi^*(a|s) \leftarrow \frac{w(s,a) d^{\pi_D}(s,a)}{\sum_{a'} w(s,a') d^{\pi_D}(s,a')} \quad \forall s, a$$

Output: π^*, w

D. Proof of Proposition 4

Proposition 4. *The closed-form solution of the inner maximization with normalization constraint, i.e.,*

$$w^*_{\nu,\lambda} := \arg\max_{w \geq 0} (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[-\alpha f\big(w(s,a)\big)\right] + \mathbb{E}_{(s,a)\sim d^D}\left[w(s,a)\big(e_\nu(s,a)-\lambda\big)\right] + \lambda$$

is given as

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

Proof. Similar to the proof for **Proposition 1**, we consider the maximization problem

$$\max_{w\geq 0} L(w,\nu,\lambda)$$

for fixed ν and λ, where we consider this maximization as a primal problem. Then, its dual problem is

$$\max_{w}\min_{\mu\geq 0} L(w,\nu,\lambda) + \sum_{s,a}\mu(s,a)w(s,a).$$

Since the strong duality holds, KKT condition is both necessary and sufficient conditions for primal and dual solutions w^* and μ^*, where dependencies on ν, λ are ignored for brevity. While the KKT conditions on **Primal feasibility**, **Dual feasibility** and **Complementary slackness** are the same as those in the proof of **Proposition 1**, the condition on **Stationarity** is slighted different due to the normalization constraint:

Condition 1 (Primal feasibility). $w^* \geq 0\ \forall s,a$.

Condition 2 (Dual feasibility). $\mu^* \geq 0\ \forall s,a$.

Condition 3 (Stationarity). $d^D(s,a)(-\alpha f'(w^*(s,a)) + e_\nu(s,a) + \mu^*(s,a) - \lambda) = 0\ \forall s,a$.

Condition 4 (Complementary slackness). $w^*(s,a)\mu^*(s,a) = 0\ \forall s,a$.

The remainder of the proof is similar to the proof of **Proposition 1**. From **Stationarity** and $d^D > 0$, we have

$$f'(w^*(s,a)) = \frac{e_\nu(s,a)+\mu^*(s,a)-\lambda}{\alpha}\ \forall s,a,$$

and since f' is invertible due to the strict convexity of f,

$$w^*(s,a) = (f')^{-1}\left(\frac{e_\nu(s,a)+\mu^*(s,a)-\lambda}{\alpha}\right)\ \forall s,a.$$

Given s,a, assume either $w^*(s,a) > 0$ or $w^*(s,a) = 0$, satisfying **Primal feasibility**.

Case 1 ($w^*(s,a) > 0$). $\mu^*(s,a) = 0$ due to **Complementary slackness**, and thus,

$$w^*(s,a) = (f')^{-1}\left(\frac{e_\nu(s,a)-\lambda}{\alpha}\right) > 0.$$

Note that **Dual feasibility** holds. Since f' is a strictly increasing function due to the strict convexity of f, $e_\nu(s,a) - \lambda > \alpha f'(0)$ should be satisfied if $f'(0)$ is well-defined.

Case 2 ($w^*(s,a) = 0$). $\mu^*(s,a) = \alpha f'(0) - e_\nu(s,a) + \lambda \geq 0$ due to **Stationarity** and **Dual feasibility**, and thus, $e_\nu(s,a) - \lambda \leq \alpha f'(0)$ should be satisfied if $f'(0)$ is well-defined.

In summary, we have

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

□

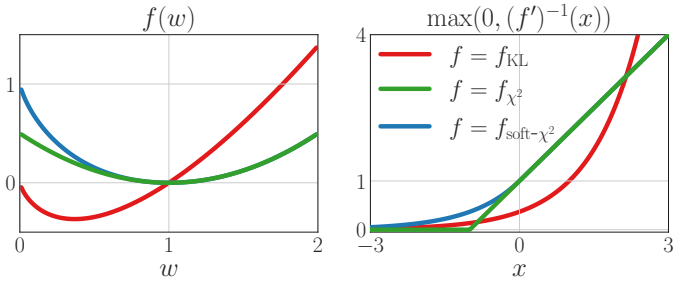

Figure 6. We depict (a) generator functions f of f-divergences and (b) corresponding functions $\max(0, (f')^{-1}(\cdot))$ used to define the closed-form solution in **Proposition 4**. While $f_{\mathrm{KL}}(x)$ has a numerical instability for large x and $f_{\chi^2}(x)$ provides zero gradients for negative x, $f_{\text{soft-}\chi^2}$ does not suffer from both issues.

E. f-divergence

Pertinent to the result of **Proposition 4**, one can observe that the choice of the function f of f-divergence can affect the numerical stability of optimization especially when using the closed-form solution of $w^*_{\nu,\lambda}$:

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

For example, for the choice of $f(x) = f_{\mathrm{KL}}(x) := x\log x$ that corresponds to KL-divergence, we have $(f'_{\mathrm{KL}})^{-1}(x) = \exp(x-1)$. This yields the following closed-form solution of $w^*_{\nu,\lambda}$:

$$w^*_{\nu,\lambda}(s,a) = \exp\left(\frac{e_\nu(s,a)-\lambda}{\alpha} - 1\right).$$

However, the choice of f_{KL} can incur numerical instability due to its inclusion of an $\exp(\cdot)$, i.e. for values of $\frac{1}{\alpha}(e_\nu(s,a)-\lambda)$ in order of tens, the value of $w^*_{\nu,\lambda}(s,a)$ easily explodes and so does the gradient $\nabla_\nu w^*_{\nu,\lambda}(s,a)$.

Alternatively, for the choice of $f(x) = f_{\chi^2}(x) := \frac{1}{2}(x-1)^2$ that corresponds to χ^2-divergence, we have $(f'_{\chi^2})^{-1}(x) = x+1$. This yields the following closed-form solution of $w^*_{\nu,\lambda}$:

$$w^*_{\nu,\lambda}(s,a) = \mathrm{ReLU}\left(\frac{e_\nu(s,a)-\lambda}{\alpha} + 1\right),$$

where $\mathrm{ReLU}(x) := \max(0,x)$. Still, this choice may suffer from dying gradient problem: for values of negative $\frac{1}{\alpha}(e_\nu(s,a)-\lambda)+1$, the gradient $\nabla_\nu w^*_{\nu,\lambda}(s,a)$ becomes zero, which can make training ν slow or even fail.

Consequently, we adopt the function $f = f_{\text{soft-}\chi^2}$ that combines the form of f_{KL} and f_{χ^2}, which can prevent both of the aforementioned issues:

$$f_{\text{soft-}\chi^2}(x) := \begin{cases} x\log x - x + 1 & \text{if } 0 < x < 1 \\ \frac{1}{2}(x-1)^2 & \text{if } x \geq 1. \end{cases} \quad \Rightarrow \quad (f_{\text{soft-}\chi^2}(x)')^{-1}(x) = \begin{cases} \exp(x) & \text{if } x < 0 \\ x+1 & \text{if } x \geq 0 \end{cases}$$

This particular choice of f yields the following closed-form solution of $w^*_{\nu,\lambda}$:

$$w^*_{v,\lambda}(s,a) = \mathrm{ELU}\left(\frac{e_v(s,a)-\lambda}{\alpha}\right) + 1.$$

Here, $\mathrm{ELU}(x) := \exp(x) - 1$ if $x < 0$ and x for $x \geq 0$. Note that the solution for $f = f_{\text{soft-}\chi^2}$ is numerically stable for large $\frac{1}{\alpha}(e_\nu(s,a)-\lambda)$ and always gives non-zero gradients. We use $f = f_{\text{soft-}\chi^2}$ for the D4RL experiments.

F. Experimental Settings

F.1. Random MDPs

We validate tabular OptiDICE's efficiency and robustness using randomly generated MDPs with varying numbers of trajectories and the degree of optimality of the data-collection policy, where we follow the experimental protocol of (Laroche et al., 2019; Lee et al., 2020). We conduct repeated experiments for 10,000 runs. For each run, an MDP is generated randomly, and a data-collection policy is constructed according to the given degree of optimality $\zeta \in \{0.9, 0.5\}$. Then, N trajectories for $N \in \{10, 20, 50, 100, 200, 500, 1000, 2000\}$ are collected using the generated MDP and the data-collection policy π_D. Finally, the constructed data-collection policy and the collected trajectories are given to each offline RL algorithm, and we measure the mean performance and the CVaR 5% performance.

F.1.1. Random MDP generation

We generate random MDPs with $|S| = 50$, $|A| = 4$, $\gamma = 0.95$, and a deterministic initial state distribution, i.e. $p_0(s) = 1$ for a fixed $s = s_0$. The transition model has connectivity 4: for each (s, a), non-zero probabilities of transition to next states are given to four different states (s'_1, s'_2, s'_3, s'_4), where the random transition probabilities are sampled from a Dirichlet distribution $[p(s'_1|s,a), p(s'_2|s,a), p(s'_3|s,a), p(s'_4|s,a)] \sim \mathrm{Dir}(1,1,1,1)$. The reward of 1 is given to one state that minimizes the optimal state value at the initial state; other states have zero rewards. This design of the reward function can be understood as we choose a goal state that is the most difficult to reach from the initial state. Once the agent reaches the rewarding goal state, the episode terminates.

F.1.2. Data-collection policy construction

The notion of ζ-optimality of a policy is defined as a relative performance with respect to a uniform random policy π_{unif} and an optimal policy π^*:

$$\big(\zeta\text{-optimal policy } \pi\text{'s performance } V^{\pi}(s_0)\big) = \zeta V^*(s_0) + (1-\zeta) V^{\pi_{\mathrm{unif}}}(s_0)$$

However, there are infinitely many ways to construct a ζ-optimal policy. In this work, we follow the way introduced in Laroche et al. (2019) to construct a ζ-optimal data-collection policy, and the process proceeds as follows. First, an optimal policy π^* and the optimal value function Q^* are computed. Then, starting from $\pi_{\mathrm{soft}} := \pi^*$, the policy π_{soft} is softened via $\pi_{\mathrm{soft}} \propto \exp(Q^*(s,a)/\tau)$ by increasing the temperature τ until the performance reaches $\frac{\zeta+1}{2}$-optimality. Finally, the softened policy π_{soft} is perturbed by discounting action selection probability of an optimal action at randomly selected state. This perturbation continues until the performance of the perturbed policy reaches ζ-optimality. The pseudo-code for the process of the data-collection policy construction is presented in **Algorithm 3**.

Algorithm 3 Data-collection policy construction

Input: MDP M, Degree of optimality of the data-collection policy ζ
Compute the optimal policy π^* and its value function $Q^*(s,a)$ on the given MDP M.
Initialize $\pi_{\mathrm{soft}} \leftarrow \pi^*$
Initialize a temperature parameter $\tau \leftarrow 10^{-7}$
while $V^{\pi_{\mathrm{soft}}}(s_0) > \frac{1}{2}V^*(s_0) + \frac{1}{2}\Big(\zeta V^*(s_0) + (1-\zeta)V^{\pi_{\mathrm{unif}}}(s_0)\Big)$ **do**
  Set π_{soft} to $\pi_{\mathrm{soft}}(a|s) \propto \exp\left(\frac{Q^*(s,a)}{\tau}\right) \quad \forall s, a$
  $\tau \leftarrow \tau / 0.9$
end while
Initialize $\pi_D \leftarrow \pi_{\mathrm{soft}}$
while $V^{\pi_D}(s_0) > \zeta V^*(s_0) + (1-\zeta)V^{\pi_{\mathrm{unif}}}(s_0)$ **do**
  Sample $s \in S$ uniformly at random.
  $\pi_D(a^*|s) \leftarrow 0.9\pi_D(a^*|s)$ where $a^* = \arg\max_a Q^*(s,a)$.
  Normalize $\pi_D(\cdot|s)$ to ensure $\sum_a \pi_D(a|s) = 1$.
end while
Output: The data-collection policy π_D

F.1.3. HYPERPARAMETERS

We compare our tabular OptiDICE with BasicRL, RaMDP (Petrik et al., 2016), RobustMDP (Nilim & El Ghaoui, 2005; Iyengar, 2005), SPIBB (Laroche et al., 2019), and BOPAH (Lee et al., 2020). For the hyperparameters, we follow the setting in the public code of SPIBB and BOPAH, which are listed as follows:

RaMDP. $\kappa = 0.003$ is used for the reward-adjusting hyperparameter.

RobustMDP. $\delta = 0.001$ is used for the confidence interval hyperparameter to construct an uncertainty set.

SPIBB. $N_\wedge = 5$ is used for the data-collection policy bootstrapping threshold.

BOPAH. The 2-fold cross validation criteria and fully state-dependent KL-regularization is used.

OptiDICE. $\alpha = N^{-1}$ for the number N of trajectories is used for the reward-regularization balancing hyperparameter. We also use $f(x) = \frac{1}{2}(x-1)^2$ which corresponds to χ^2-divergence.

F.2. D4RL benchmark

F.2.1. TASK DESCRIPTIONS

We use Maze2D and Gym-MuJoCo environments of D4RL benchmark (Fu et al., 2021) to evaluate OptiDICE and CQL (Kumar et al., 2020) in continuous control tasks. We summarize the descriptions of tasks in D4RL paper (Fu et al., 2021) as follows:

Maze2D. This is a navigation task in 2D state space, while the agent tries to reach a fixed goal location. By using priorly gathered trajectories, the goal of the agent is to find out a shortest path to reach the goal location. The complexity of the maze increases with the order of "maze2d-umaze", "maze2d-medium" and "maze2d-large".

Gym-MuJoCo. For each task in {hopper, walker2d, halfcheetah} of MuJoCo continuous controls, the dataset is gathered in the following ways.

random. The dataset is generated by a randomly initialized policy in each task.

medium. The dataset is generated by using the policy trained by SAC (Haarnoja et al., 2018) with early stopping.

medium-replay. The "replay" dataset consists of the samples gathered during training the policy for "medium" dataset. The "medium-replay" dataset includes both "medium" and "replay" datasets.

medium-expert. The dataset is given by using the same amount of expert trajectories and suboptimal trajectories, where those suboptimal ones are gathered by using either a randomly uniform policy or a medium-performance policy.

F.2.2. HYPERPARAMETER SETTINGS FOR CQL

We follow the hyperparameters specified by Kumar et al. (2020). For learning both the Q-funtions and the policy, fully-connected multi-layer perceptrons (MLPs) with three hidden layers and ReLU activations are used, where the number of hidden units on each layer is equal to 256. A Q-function learning rate of 0.0003 and a policy learning rate of 0.0001 are used with Adam optimizer for these networks. CQL($\mathcal{H}$) is evaluated, with an approximate max-backup (see Appendix F of (Kumar et al., 2020) for more details) and a static $\alpha = 5.0$, which controls the conservativeness of CQL. The policy of CQL is updated for 2,500,000 iterations, while we use 40,000 warm-up iterations where we update Q-functions as usual, but the policy is updated according to the behavior cloning objective.

F.2.3. HYPERPARAMETER SETTINGS FOR OPTIDICE

For neural networks $\nu_\theta, e_\phi, \pi_\psi$ and π_β in **Algorithm 1**, we use fully-connected MLPs with two hidden layers and ReLU activations, where the number of hidden units on each layer is equal to 256. For π_ψ, we use tanh-squashed normal distribution. We regularize the entropy of π_ψ with learnable entropy regularization coefficients, where target entropies are set to be the same as those in SAC (Haarnoja et al., 2018) ($-\dim(A)$ for each task). For π_β, we use tanh-squashed mixture of normal distributions, where we build means and standard deviations of each mixture component upon shared hidden outputs. No entropy regularization is applied to π_β. For both π_ψ and π_β, means are clipped within $(-7.24, 7.24)$, while log of standard deviations are clipped within $(-5, 2)$. For the optimization of each network, we use stochastic gradient descent with Adam optimizer and its learning rate 0.0003. The batch size is set to be 512. Before training neural networks, we

preprocess the dataset D by standardizing observations and rewards. We additionally scale the rewards by multiplying 0.1. We update the policy π_ψ for 2,500,000 iterations, while we use 500,000 warm-up iterations for other networks, i.e., those networks other than π_ψ are updated for 3,000,000 iterations.

For each task and OptiDICE methods (OptiDICE-minimax, OptiDICE-MSE), we search the number K of mixtures (for π_β) within $\{1, 5, 9\}$ and the coefficient α within $\{0.0001, 0.001, 0.01, 0.1, 1\}$, while we additionally search α over $\{2, 5, 10\}$ for hopper-medium-replay. The hyperparameters K and α showing the best mean performance were chosen, which are described as follows:

Table 2. Hyperaparameters

Task	OptiDICE-MSE		OptiDICE-minimax	
	K	α	K	α
maze2d-umaze	5	0.001	1	0.01
maze2d-medium	5	0.0001	1	0.01
maze2d-large	1	0.01	1	0.01
hopper-random	5	1	5	1
hopper-medium	9	0.1	9	0.1
hopper-medium-replay	9	10	1	2
hopper-medium-expert	9	1	5	1
walker2d-random	9	0.0001	1	0.0001
walker2d-medium	9	0.01	5	0.01
walker2d-medium-replay	9	0.1	9	0.1
walker2d-medium-expert	5	0.01	5	0.01
halfcheetah-random	5	0.0001	9	0.001
halfcheetah-medium	1	0.01	1	0.1
halfcheetah-medium-replay	9	0.01	1	0.1
halfcheetah-medium-expert	9	0.01	9	0.01

G. Experimental results

G.1. Experimental results for $\gamma = 0.99$

Table 3. Normalized performance of OptiDICE compared with baselines. Mean scores for baselines—BEAR (Kumar et al., 2019), BRAC (Wu et al., 2019), AlgaeDICE (Nachum et al., 2019b), and CQL (Kumar et al., 2020)— come from D4RL benchmark. We also report the performance of CQL (Kumar et al., 2020) obtained by running the code released by authors (denoted as CQL (ours) in the table). OptiDICE achieves the best performance on 6 tasks compared to our baselines. Note that 3-run mean scores without confidence intervals were reported on each task by Fu et al. (2021). For CQL (ours) and OptiDICE, we use 5 runs and report means and 95% confidence intervals.

Task	BC	SAC	BEAR	BRAC -v	Algae DICE	CQL	CQL (ours)	OptiDICE minimax	OptiDICE MSE
maze2d-umaze	3.8	88.2	3.4	-16.0	-15.7	5.7	-14.9 $\pm$ 0.7	**111.0 $\pm$ 8.3**	105.8 $\pm$ 17.5
maze2d-medium	30.3	26.1	29.0	33.8	10.0	5.0	17.2 $\pm$ 8.7	109.9 $\pm$ 7.7	**145.2 $\pm$ 17.5**
maze2d-large	5.0	-1.9	4.6	40.6	-0.1	12.5	1.6 $\pm$ 3.8	116.1 $\pm$ 43.1	**155.7 $\pm$ 33.4**
hopper-random	9.8	11.3	11.4	**12.2**	0.9	10.8	10.7 $\pm$ 0.0	11.2 $\pm$ 0.1	10.7 $\pm$ 0.2
hopper-medium	29.0	0.8	52.1	31.1	1.2	58.0	89.8 $\pm$ 7.6	92.9 $\pm$ 2.6	**94.1 $\pm$ 3.7**
hopper-medium-replay	11.8	3.5	33.7	0.6	1.1	**48.6**	33.3 $\pm$ 2.2	36.4 $\pm$ 1.1	30.7 $\pm$ 1.2
hopper-medium-expert	111.9	1.6	96.3	0.8	1.1	98.7	**112.3 $\pm$ 0.2**	111.5 $\pm$ 0.6	106.7 $\pm$ 1.8
walker2d-random	1.6	4.1	7.3	1.9	0.5	7.0	3.0 $\pm$ 1.8	9.4 $\pm$ 2.2	**9.9 $\pm$ 4.3**
walker2d-medium	6.6	0.9	59.1	**81.1**	0.3	79.2	73.7 $\pm$ 2.7	21.8 $\pm$ 7.1	20.8 $\pm$ 3.1
walker2d-medium-replay	11.3	1.9	19.2	0.9	0.6	**26.7**	13.4 $\pm$ 0.8	21.5 $\pm$ 2.9	21.6 $\pm$ 2.1
walker2d-medium-expert	6.4	-0.1	40.1	81.6	0.4	**111.0**	99.7 $\pm$ 7.2	74.7 $\pm$ 7.5	74.8 $\pm$ 9.2
halfcheetah-random	2.1	30.5	25.1	31.2	-0.3	**35.4**	25.5 $\pm$ 0.5	8.3 $\pm$ 0.8	11.6 $\pm$ 1.2
halfcheetah-medium	36.1	-4.3	41.7	**46.3**	-2.2	44.4	42.3 $\pm$ 0.1	37.1 $\pm$ 0.1	38.2 $\pm$ 0.1
halfcheetah-medium-replay	38.4	-2.4	38.6	**47.7**	-2.1	46.2	43.1 $\pm$ 0.7	38.9 $\pm$ 0.5	39.8 $\pm$ 0.3
halfcheetah-medium-expert	35.8	1.8	53.4	41.9	-0.8	62.4	53.5 $\pm$ 13.3	76.2 $\pm$ 7.0	**91.1 $\pm$ 3.7**

G.2. Experimental results with importance-weighted BC

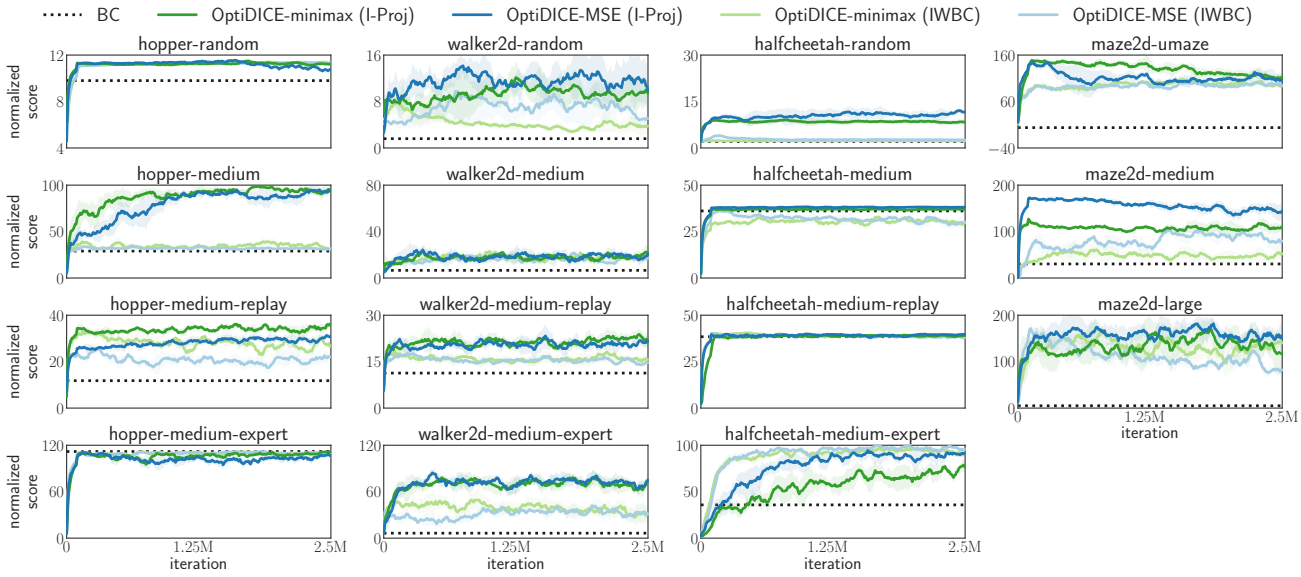

Figure 7. Performance of BC, OptiDICE with importance-weighted BC (IWBC) and information projection (I-Proj) methods on D4RL benchmark. $\gamma = 0.99$ is used.

The empirical results of OptiDICE for different policy extraction methods (information-weighted BC, I-projection methods) are depicted in *Figure 7*. For those results with IWBC, we search α within $\{0.0001, 0.001, 0.01, 0.1, 1\}$ and choose one with the best mean performance, which are summarized in *Table* 4. The hyperparameters other than α are the same as those used for information-projection methods, which is described in Section F.2.3. We empirically observe that policy extraction with information projection method performs better than the extraction with importance-weighted BC, as discussed in Section 3.3.

Table 4. Hyperaparameters for importance-weighted BC

Task	OptiDICE-MSE	OptiDICE-minimax
	α	α
maze2d-umaze	0.001	0.001
maze2d-medium	0.0001	0.001
maze2d-large	0.001	0.001
hopper-random	1	1
hopper-medium	0.01	0.1
hopper-medium-replay	0.1	0.1
hopper-medium-expert	0.1	0.1
walker2d-random	0.0001	0.0001
walker2d-medium	0.1	0.1
walker2d-medium-replay	0.1	0.1
walker2d-medium-expert	0.01	0.01
halfcheetah-random	0.001	0.01
halfcheetah-medium	0.0001	0.1
halfcheetah-medium-replay	0.01	0.01
halfcheetah-medium-expert	0.01	0.01

G.3. Experimental results for $\gamma \in \{0.99, 0.999, 0.9999, 1.0\}$

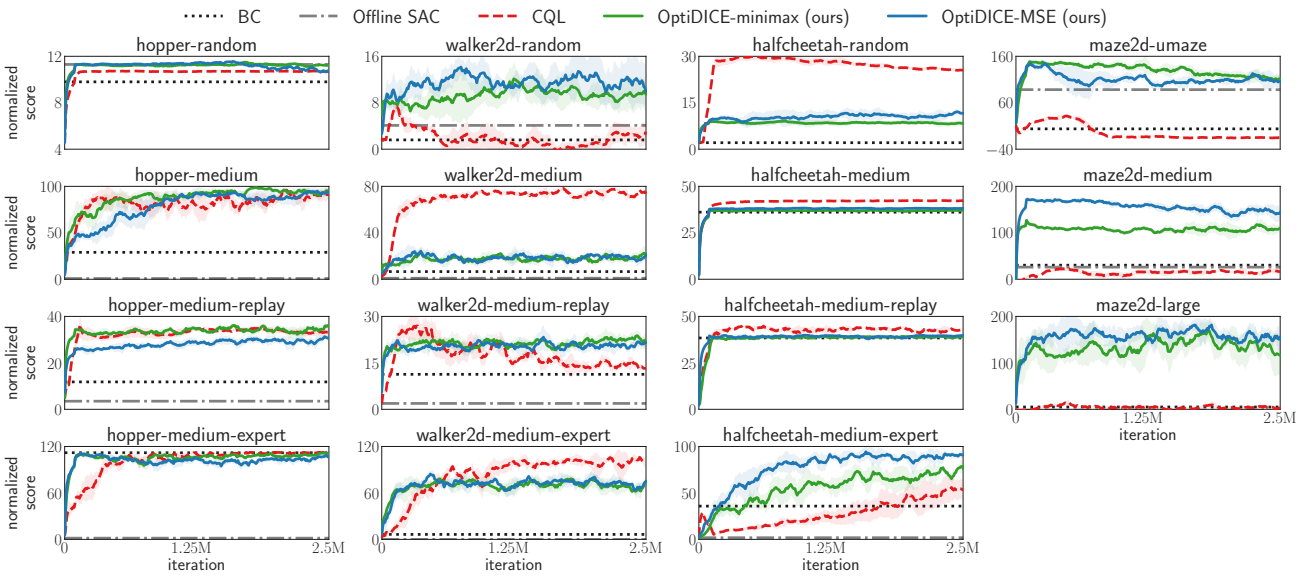

Figure 8. Performance of BC, CQL, OptiDICE-minimax and OptiDICE-MSE on D4RL benchmark for $\gamma = 0.99$.

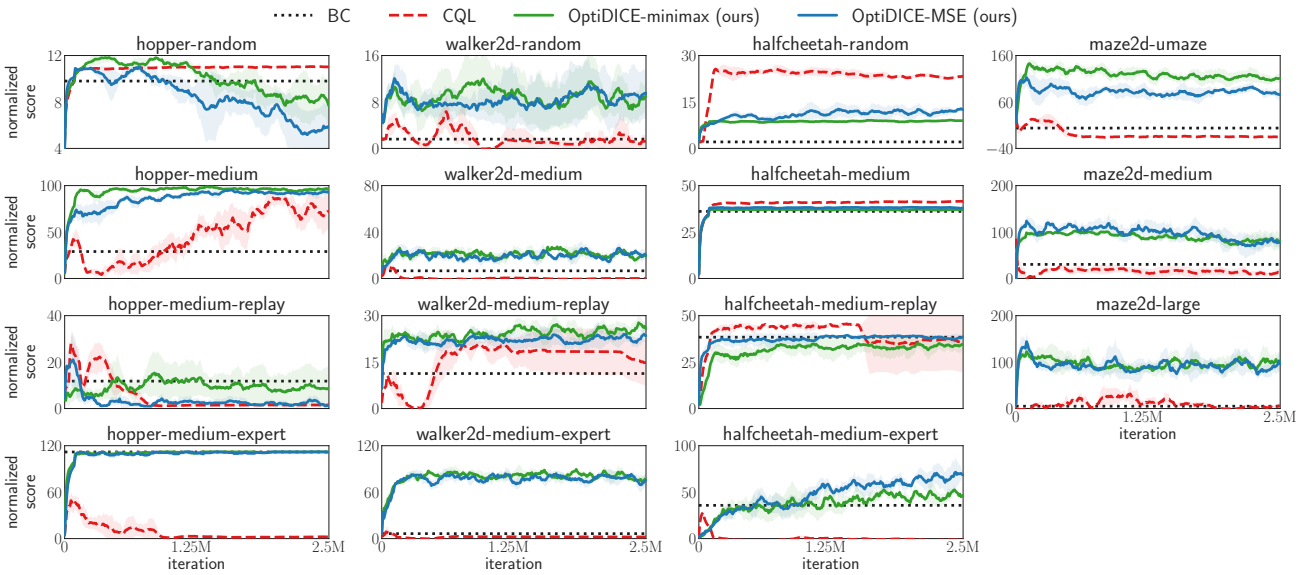

Figure 9. Performance of BC, CQL, OptiDICE-minimax and OptiDICE-MSE on D4RL benchmark for $\gamma = 0.999$.

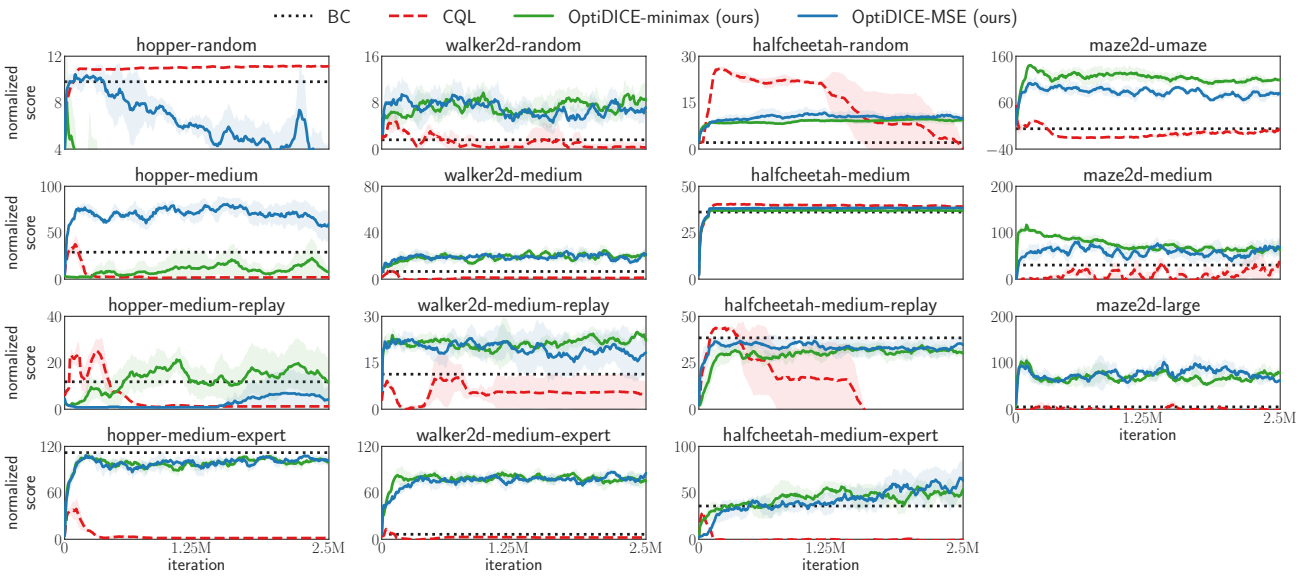

Figure 10. Performance of BC, CQL, OptiDICE-minimax and OptiDICE-MSE on D4RL benchmark for $\gamma = 0.9999$.

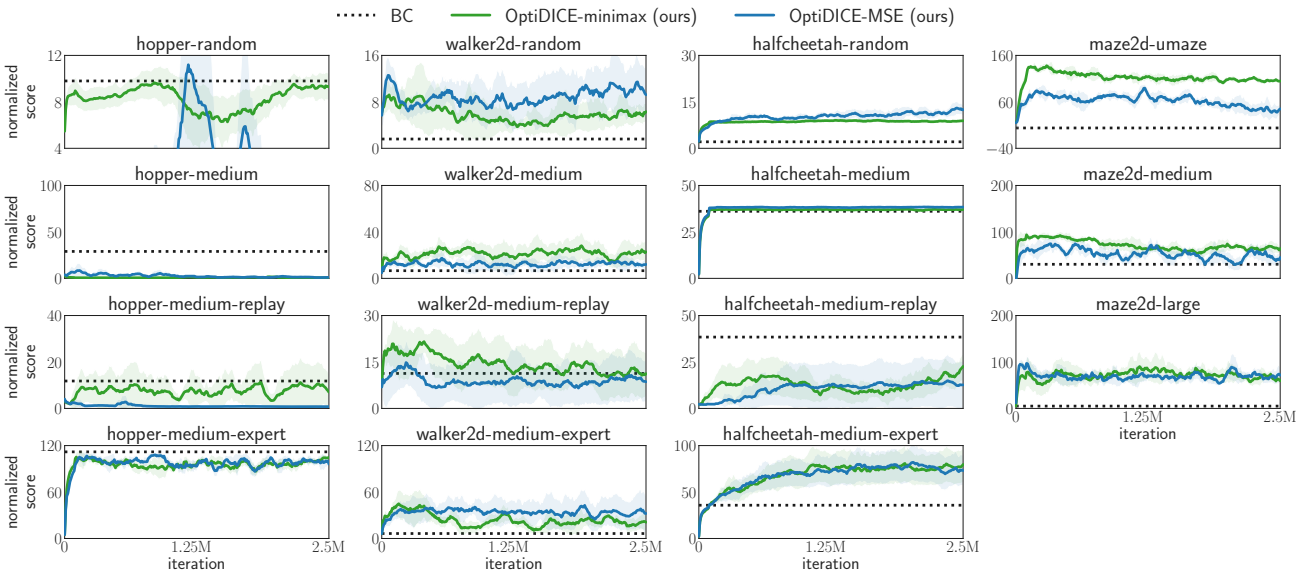

Figure 11. Performance of BC, OptiDICE-minimax and OptiDICE-MSE on D4RL benchmark for $\gamma = 1$. Note that CQL cannot deal with $\gamma = 1$ case. Thus, we only provide the results of OptiDICE and BC.