LobsDICE: Offline Learning from Observation via Stationary Distribution Correction Estimation

Geon-Hyeong Kim^{1,3,*,†}**, Jongmin Lee**^{4,*}**, Youngsoo Jang**^{1,3,†}**,**
Hongseok Yang^{1,2,5} **Kee-Eung Kim**^{1,2}
[1] School of Computing, KAIST
[2] Kim Jaechul Graduate School of AI, KAIST
[3] LG AI Research
[4] University of California, Berkeley
[5] Discrete Mathematics Group, Institute for Basic Science (IBS)

Abstract

We consider the problem of learning from observation (LfO), in which the agent aims to mimic the expert's behavior from the state-only demonstrations by experts. We additionally assume that the agent cannot interact with the environment but has access to the action-labeled transition data collected by some agents with unknown qualities. This offline setting for LfO is appealing in many real-world scenarios where the ground-truth expert actions are inaccessible and the arbitrary environment interactions are costly or risky. In this paper, we present LobsDICE, an offline LfO algorithm that learns to imitate the expert policy via optimization in the space of stationary distributions. Our algorithm solves a single convex minimization problem, which minimizes the divergence between the two state-transition distributions induced by the expert and the agent policy. Through an extensive set of offline LfO tasks, we show that LobsDICE outperforms strong baseline methods.

1 Introduction

The ability to learn from experience is one of the core aspects of an intelligent agent. Reinforcement learning (RL) [36] provides a framework to acquire such an intelligent behavior autonomously through interactions with the environment while receiving reward feedback. However, the practical applicability of RL to real-world domains has been limited for two reasons. First, designing a suitable reward function for complex tasks can be extremely difficult. The RL agent learns behaviors incentivized by the reward function rather than the ones intended, which can be nontrivial to specify in terms of reward. Second, the need for online interaction with the environment during the RL training loop has hindered its adoption in many real-world domains, where environment interactions are costly or risky.

Imitation learning (IL) [1, 33, 34] circumvents the difficulty of reward design in RL by leveraging demonstrations given by experts, where the goal is to mimic the expert's behavior. However, the standard IL requires the expert demonstrations to contain not only the state information (e.g. robot joint angles) but also the precise action (e.g. robot joint torques) executed by the expert at each time step. This demand for explicit action labels is in contrast to the way human imitates (e.g. learning by watching videos) and precludes leveraging a massive amount of data in which the action label is missing. Therefore, developing an imitation learning algorithm that can learn from observing the

[*] Equal contribution.

[†] Work done while the authors were students at KAIST.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

experts' state-only trajectories is a promising direction for creating a practical, autonomous intelligent agent. Learning from observation (LfO) [2, 14, 26, 39, 45] concerns this particular learning scenario and has been gaining interest in recent years.

In this paper, we are particularly interested in solving the LfO problem in an *offline* setting: Given the state-only demonstrations by experts and abundant state-action demonstrations by imperfect agents of arbitrary levels of optimality, we aim to find a policy that follows the sequence of expert's states without further interaction with the environment. This problem setting is appealing in many practical situations where environment interactions are costly or risky. Yet, it cannot be straightforwardly tackled by existing approaches. Most of the existing LfO methods are on-policy algorithms [25, 35, 39, 40], which is not directly applicable to the offline setting. One of the few exceptions is BCO [38], which performs behavior cloning on the inferred action via an inverse dynamics model. This method could be suboptimal unless the inverse dynamics disagreement is always zero, i.e. the underlying environment dynamics is deterministic and injective. Another one is OPOLO [45], an off-policy LfO algorithm, but it relies on nested min-max optimization as well as out-of-distribution (OOD) action values, which can be unstable especially in the offline setting. IQ-Learn [8] solves an Inverse RL and can be applied to the offline LfO by learning a state-only reward function. However, it suffers from numerical instability due to using OOD action values in the offline setting. Lastly, RCE [5], an example-based control algorithm, can in principle be applied to the offline LfO setting by providing the expert trajectories as successful example states; however, its empirical performance is known to be limited without online data collection.

We present an offline LfO algorithm that minimizes the divergence between state-transition distributions induced by the expert and the learned policy, without requiring an inverse dynamics model. Our algorithm, *offline Learning from OBServation via stationary DIstribution Correction Estimation* (LobsDICE), essentially optimizes in the space of state-action stationary distributions and state-transition stationary distributions, rather than in the space of policies. We show that our formulation can be reduced to a single convex minimization problem that can be solved efficiently in practice, unlike existing imitation learning (from observation) algorithms that rely on nested min-max optimization. In the experiments, we demonstrate that LobsDICE can successfully recover the state visitations by the expert, outperforming strong baseline methods.

2 Preliminaries

2.1 Markov decision process

We consider an environment modeled as a Markov Decision Process (MDP), defined by $M = \langle S, A, T, R, p_0, \gamma \rangle$ [36], where S is the set of states, A is the set of actions, $T : S \times A \to \Delta(S)$ is the transition probability, $R : S \times A \to \mathbb{R}$ is the reward function, $p_0 \in \Delta(S)$ is the distribution of the initial state, and $\gamma \in (0, 1)$ is the discount factor. The policy $\pi : S \to \Delta(A)$ is a mapping from states to distribution over actions. For the given policy π, its *state-action* stationary distribution $d^\pi(s, a)$ and *state-transition* stationary $\bar{d}^\pi(s, s')$ are defined as:

$$d^\pi(s,a) := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s, a_t = a), \quad \bar{d}^\pi(s,s') := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s, s_{t+1} = s'),$$

where $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t)$, and $s_{t+1} \sim T(\cdot|s_t, a_t)$ for all timesteps $t \geq 0$. For brevity, the bar notation $\bar{(\cdot)}$ will be used to denote the distributions for (s, s'), e.g. $\bar{d}^\pi(s, s')$.

We assume offline LfO setting, where direct, online interactions with the environment are not allowed, and the policy should be optimized solely from a pre-collected dataset. We denote the dataset of state-only demonstrations collected by experts as $D^E = \{(s, s')_i\}_{i=1}^{N_E}$ and the dataset of state-action demonstrations by some imperfect agents as $D^I = \{(s, a, s')_i\}_{i=1}^{N_I}$. That is, we do not have information about the actions taken by the expert, but instead, we have additional action-labeled transition data collected by some other agents with unknown levels of optimality. We denote the corresponding distributions of the datasets D^E and D^I by $\bar{d}^E$ and d^I, respectively. For brevity, we will abuse notation d^I to represent $(s, a) \sim d^I$, $(s, a, s') \sim d^I$, and $(s, s') \sim \bar{d}^I$ unless ambiguous.

2.2 Imitation learning and learning from observation

Imitation learning (IL) aims to mimic the expert policy from its state-action demonstrations. IL can be naturally formulated as a distribution matching problem that minimizes the divergence between *state-action* stationary distributions induced by the expert and the target policy [11, 13]. For example, one can consider minimizing KL-divergence [17]:

$$\min_{\pi} D_{\mathrm{KL}}(d^{\pi}(s,a) \| d^{E}(s,a)) = \mathbb{E}_{(s,a)\sim d^{\pi}}\left[\log \tfrac{d^{\pi}(s,a)}{d^{E}(s,a)}\right]. \tag{1}$$

However, the standard IL requires action labels in the expert demonstrations, which may be a too strong requirement for various practical situations. *Learning from observation* (LfO) relaxes the requirement on action labels, and aims to imitate the expert's behavior only from the state observations. Since the expert's action information is missing in the demonstrations, the distribution matching for state-action stationary distribution is no longer readily applicable. Therefore, LfO is reformulated as another distribution matching problem that minimizes the divergence between *state-transition* stationary distributions induced by the expert and the target policy [39, 40, 45][1]:

$$\min_{\pi} D_{\mathrm{KL}}(\bar{d}^{\pi}(s,s') \| \bar{d}^{E}(s,s')) = \mathbb{E}_{(s,s')\sim \bar{d}^{\pi}}\left[\log \tfrac{\bar{d}^{\pi}(s,s')}{\bar{d}^{E}(s,s')}\right]. \tag{2}$$

Still, optimizing Eq. (2) in a purely offline manner is challenging since naively estimating the expectation would require the knowledge of $T(\cdot|s,a)$ for the OOD action $a \sim \pi(s)$ by the target policy, which is inaccessible in the offline LfO setting; see Section 9.8 in [45] for more discussions. OPOLO [45], an off-policy LfO algorithm, mitigates this challenge by minimizing the following *upper bound* of the divergence $D_{\mathrm{KL}}(\bar{d}^{\pi}(s,s') \| \bar{d}^{E}(s,s'))$:

$$(2) \le \mathbb{E}_{\bar{d}^{\pi}(s,s')}\left[\log \tfrac{\bar{d}^{I}(s,s')}{\bar{d}^{E}(s,s')}\right] + D_{\mathrm{KL}}(d^{\pi}(s,a) \| d^{I}(s,a)) \tag{3}$$

and applying DualDICE trick [29] to the RHS of (3). The upper bound gap is given by the inverse dynamics disagreement between the target policy and the imperfect demonstrator:

$$(3) - (2) = D_{\mathrm{KL}}(d^{\pi}(a|s,s') \| d^{I}(a|s,s')), \tag{4}$$

which usually gets larger as the stochasticity of the environment transition increases.

3 LobsDICE

We present *offline Learning from OBServation via stationary DIstribution Correction Estimation* (LobsDICE), a principled offline LfO algorithm that further extends the recent progress made by the DIstribution Correction Estimation (DICE) methods for offline RL. LobsDICE essentially optimizes the stationary distributions of the target policy to match the expert's state visitations.

In the context of offline RL, the policy constraint principle (i.e. prevent deviating too much from the data support) is one of the common approaches to avoid severe performance degradation [7, 12, 18, 22, 30]. In the same manner, we use KL divergence minimization between $\bar{d}^{\pi}(s,s')$ and $\bar{d}^{E}(s,s')$ with additional KL regularization on the deviation from d^{I}:

$$\min_{\pi} D_{\mathrm{KL}}(\bar{d}^{\pi}(s,s') \| \bar{d}^{E}(s,s')) + \alpha D_{\mathrm{KL}}(d^{\pi}(s,a) \| d^{I}(s,a)). \tag{5}$$

Here the hyperparameter $\alpha > 0$ balances between encouraging state-transition matching and preventing distribution shift from the distribution of imperfect demonstrations. All the proofs can be found in Appendix F.

3.1 Lagrange dual formulation

The derivation of our algorithm starts by rewriting the (regularized) distribution matching problem (5) in terms of directly optimizing stationary distribution, rather than policy:

$$\max_{d,\bar{d}\ge 0} \; -D_{\mathrm{KL}}(\bar{d}(s,s') \| \bar{d}^{E}(s,s')) - \alpha D_{\mathrm{KL}}(d(s,a) \| d^{I}(s,a)) \tag{6}$$

$$\text{s.t.} \; \sum_{a'} d(s',a') = (1-\gamma)p_0(s') + \gamma \sum_{s,a} d(s,a)T(s'|s,a) \quad \forall s', \tag{7}$$

$$\sum_{a} d(s,a)T(s'|s,a) = \bar{d}(s,s') \quad \forall s,s', \tag{8}$$

[1] See Appendix A for a discussion of why $\bar{d}^{\pi}(s,s')$-matching is preferable to $\bar{d}^{\pi}(s)$-matching.

The Bellman flow constraint (7) ensures $d(s,a)$ to be a valid state-action stationary distribution of some policy, where $d(s,a)$ can be interpreted as a normalized occupancy measure of (s,a). The marginalization constraint (8) enforces $\bar{d}(s,s')$ to be the state-transition stationary distribution that is directly induced by $d(s,a)$. In essence, the constrained optimization problem (6-8) seeks the stationary distributions of an optimal policy, which best matches the state-transition trajectories of the expert. Once we have computed the optimal solution $(d^*, \bar{d}^*)$, its corresponding optimal policy can also be obtained by normalizing d^* for each state [32]: $\pi^*(a|s) = \frac{d^*(s,a)}{\sum_a d^*(s,a)}$.

Note that DemoDICE [15] considers a similar optimization problem to ours, but it deals with the offline IL (i.e. *state-action* stationary distribution matching), whereas we consider the offline LfO (e.g *state-transition* stationary distribution matching). Accordingly, the optimization variable $\bar{d}(s,s')$ and the marginalization constraint (8) are newly added in our formulation.

We then consider the Lagrangian for the constrained optimization (6-8):

$$\min_{\mu,\nu} \max_{d,\bar{d}\geq 0} -\mathbb{E}_{\bar{d}}\Big[\log\frac{\bar{d}(s,s')}{d^E(s,s')}\Big] - \alpha\mathbb{E}_{d}\Big[\log\frac{d(s,a)}{d^I(s,a)}\Big] + \sum_{s,s'}\mu(s,s')\big(\bar{d}(s,s') - \sum_a d(s,a)T(s'|s,a)\big)$$
$$+\sum_{s'}\nu(s')\big((1-\gamma)p_0(s') + \gamma\sum_{s,a} d(s,a)T(s'|s,a) - \sum_{a'} d(s',a')\big), \tag{9}$$

where $\nu(s)\in\mathbb{R}$ are the Lagrange multipliers for the Bellman flow constraints (7), and $\mu(s,s')\in\mathbb{R}$ are the the Lagrange multipliers for the marginalization constraint (8). Note that the Lagrangian (9) cannot be naively optimized in an offline manner since it requires evaluation of $T(s'|s,a)$ for $(s,a)\sim d$, which is not accessible in the offline LfO setting. Therefore, we rearrange the terms in (9) to eliminate the direct dependence on d and $\bar{d}$, introducing new optimization variables w and $\bar{w}$ that denote stationary distribution correction ratios for (s,a) and (s,s'), respectively:

$$(9) = \min_{\mu,\nu}\max_{d,\bar{d}\geq 0}(1-\gamma)\mathbb{E}_{s_0\sim p_0}[\nu(s_0)] + \mathbb{E}_{(s,s')\sim\bar{d}}\Big[\mu(s,s')\overbrace{-\log\underbrace{\tfrac{\bar{d}(s,s')}{\bar{d}^I(s,s')}}_{=:\bar{w}(s,s')} + \log\underbrace{\tfrac{\bar{d}^E(s,s')}{\bar{d}^I(s,s')}}_{=:r(s,s')}}^{=-\log\frac{\bar{d}(s,s')}{d^E(s,s')}}\Big]$$
$$+\mathbb{E}_{(s,a)\sim d}\Big[\underbrace{\mathbb{E}_{s'}[-\mu(s,s')+\gamma\nu(s')]-\nu(s)}_{=:e_{\mu,\nu}(s,a)} - \alpha\log\underbrace{\tfrac{d(s,a)}{d^I(s,a)}}_{=:w(s,a)}\Big] \tag{10}$$

$$=\min_{\mu,\nu}\max_{w,\bar{w}\geq 0}(1-\gamma)\mathbb{E}_{s_0\sim p_0}[\nu(s_0)] + \mathbb{E}_{(s,s')\sim\bar{d}^I}\big[\bar{w}(s,s')\big(r(s,s')+\mu(s,s')-\log\bar{w}(s,s')\big)\big]$$
$$+\mathbb{E}_{(s,a)\sim d^I}\big[w(s,a)\big(e_{\mu,\nu}(s,a)-\alpha\log w(s,a)\big)\big] =: \mathcal{L}(w,\bar{w},\mu,\nu). \tag{11}$$

Similar to the assumption of full coverage which is fairly standard across a broad set of recent offline RL approaches [19, 22, 27], we make a milder assumption that $\bar{d}^I(s,s')>0$ whenever $\bar{d}^E(s,s')>0$. This assumption is necessary to recover the expert's behavior successfully. We introduce the log ratio $r(s,s')=\log\frac{\bar{d}^E(s,s')}{\bar{d}^I(s,s')}$ in (10) to take an expectation under $\bar{d}^I$ (instead of $\bar{d}^E$), which is assumed to have a broader support than $\bar{d}^E$. This log ratio $r(s,s')$ can be easily estimated using a pretrained discriminator for two datasets D^E and D^I, which will be explained in detail in the following section.

In summary, LobsDICE aims to solve the minimax optimization,

$$\min_{\mu,\nu}\max_{w,\bar{w}\geq 0}\mathcal{L}(w,\bar{w},\mu,\nu). \tag{12}$$

The optimal solution $(w^*,\bar{w}^*)$ of (12) represents stationary distribution corrections of an optimal policy π^*: $w^*(s,a)=\frac{d^{\pi^*}(s,a)}{d^I(s,a)}$ and $\bar{w}^*(s,s')=\frac{\bar{d}^{\pi^*}(s,s')}{\bar{d}^I(s,s')}$.

3.2 Log ratio estimation via a pretrained discriminator

To optimize (11), an estimate of the log ratio $r(s,s')=\log\frac{\bar{d}^E(s,s')}{\bar{d}^I(s,s')}$ is required. The log ratio estimation is straightforward for tabular MDPs since we can use empirical distributions from the datasets to estimate $\bar{d}^E(s,s')$ and $\bar{d}^I(s,s')$. For continuous MDPs, we train a discriminator $c: S\times S\to[0,1]$ by solving the following maximization problem [9, 45]:

$$c^* = \operatorname*{arg\,max}_{c:S\times S\to[0,1]}\mathbb{E}_{(s,s')\sim\bar{d}^E}[\log c(s,s')] + \mathbb{E}_{(s,s')\sim\bar{d}^I}[\log(1-c(s,s'))]. \tag{13}$$

It is easy to show that the optimal discriminator satisfies $c^*(s,s') = \frac{\bar{d}^E(s,s')}{d^E(s,s')+d^I(s,s')}$. Thus, $r(s,s')$ can be derived from the optimal discriminator c^* as,

$$r(s,s') = -\log\big(\tfrac{1}{c^*(s,s')} - 1\big). \tag{14}$$

3.3 Minimax to min: a closed-form solution

Exploiting the strict convexity of $x \log x$, we can derive a closed-form solution to the inner maximization for $(w, \bar{w})$ in (11).

Proposition 3.1. *For any* (μ, ν)*, the closed-form solution to the inner maximization of* (11)*, i.e.* $(w_{\mu,\nu}, \bar{w}_\mu) = \arg\max_{w,\bar{w}\geq 0} \mathcal{L}(w, \bar{w}, \mu, \nu)$*, is given by:*

$$w_{\mu,\nu}(s,a) = \exp\big(\tfrac{1}{\alpha} e_{\mu,\nu}(s,a) - 1\big) \text{ and } \bar{w}_\mu(s,s') = \exp(r(s,s') + \mu(s,s') - 1). \tag{15}$$

Based on the above result, we reduce the nested min-max optimization of (11) to a single minimization by plugging the closed-form solution $(w_{\mu,\nu}, \bar{w}_\mu)$ into $\mathcal{L}(w, \bar{w}, \mu, \nu)$:

$$\begin{aligned}\min_{\mu,\nu} \mathcal{L}(w_{\mu,\nu}, \bar{w}_\mu, \mu, \nu) &= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,s')\sim \bar{d}^I}\big[\exp\big(r(s,s') + \mu(s,s') - 1\big)\big] \\ &\quad + \alpha \mathbb{E}_{(s,a)\sim d^I}\big[\exp\big(\tfrac{1}{\alpha} e_{\mu,\nu}(s,a) - 1\big)\big]. \end{aligned} \tag{16}$$

We can even show that $\mathcal{L}(w_{\mu,\nu}, \bar{w}_\mu, \mu, \nu)$ is a convex function of μ and ν.

Proposition 3.2. $\mathcal{L}(w_{\mu,\nu}, \bar{w}_\mu, \mu, \nu)$ *is convex with respect to* μ *and* ν*.*

In short, by operating in the space of stationary distributions, offline LfO can, in principle, be resolved by solving a *convex minimization* problem. This is in contrast to the existing LfO algorithms, which typically involve either an adversarial training that optimizes the policy and the discriminator [14, 39, 40] or a nested min-max optimization for the policy and the critic [45].

3.4 Policy extraction

So far, we have derived an algorithm that essentially solves the state-transition distribution matching problem via convex minimization. However, we obtain (μ^*, ν^*) as the solution of (16), instead of the optimal policy π^*. The remaining problem is to extract the optimal policy from (μ^*, ν^*). The first step is to see that we can obtain the state-action stationary distribution correction $w_{\mu^*,\nu^*}(s,a) = \frac{d^{\pi^*}(s,a)}{d^I(s,a)}$ of the optimal policy from the closed-form solution (15). Among many possibilities to extract the policy from the state-action distribution correction, we adopt weighted behavior cloning (WBC):

$$\max_\pi \mathbb{E}_{(s,a)\sim d^{\pi^*}}[\log \pi(a|s)] = \mathbb{E}_{(s,a)\sim d^I}[w_{\mu^*,\nu^*}(s,a) \log \pi(a|s)] \tag{17}$$

which aims to maximize the predicted probability of actions chosen by optimal policy π^*. This is done by BC on the offline dataset D^I where each sample (s,a) is weighted by $w_{\mu^*,\nu^*}(s,a)$. For tabular MDPs, we can formally show that WBC extracts an optimal policy π^* (Appendix C).

3.5 Practical algorithm with sample-based approximation

In practice, we estimate $\mathcal{L}(w, \bar{w}, \mu, \nu)$ in (11) using samples from distribution d^I. Let $\hat{\mathbf{E}}_{x\in D}[f(x)] := \frac{1}{|D|}\sum_{x\in D} f(x)$ be a Monte-Carlo estimate of $\mathbb{E}_{x\sim p}[f(x)]$ where $D = \{x_i\}_{i=1}^{|D|} \sim p$. We denote each sample (s,a,s') in D^I as x for brevity.

$$\begin{aligned}\min_{\mu,\nu} \max_{w,\bar{w}\geq 0} \widehat{\mathcal{L}}(w, \bar{w}, \mu, \nu) &:= (1-\gamma)\hat{\mathbf{E}}_{s_0\in D_0}[\nu(s_0)] \\ &+ \hat{\mathbf{E}}_{x\in D^I}\big[\bar{w}(s,s')\big(r(s,s') + \mu(s,s') - \log \bar{w}(s,s')\big) + w(s,a)\big(\hat{e}_{\mu,\nu}(s,a,s') - \alpha \log w(s,a)\big)\big]\end{aligned} \tag{18}$$

where $\hat{e}_{\mu,\nu}(s,a,s') := -\mu(s,s') + \gamma\nu(s') - \nu(s)$ is a single-sample estimate of $e_{\mu,\nu}(s,a)$. Note that this sample-based objective function $\widehat{\mathcal{L}}(w, \bar{w}, \mu, \nu)$ can be estimated only from samples in the offline dataset D^I and is an unbiased estimator of $\mathcal{L}(w, \bar{w}, \mu, \nu)$ as long as every sample in D^I was collected by interacting with the underlying MDP. To the best of our knowledge, this is the first result to directly solve the state-transition distribution matching problem in a fully offline manner. In contrast, OPOLO [45] relies on the (potentially loose) *upper bound* of the objective in (3).

To reduce the minimax optimization to a single minimization, we apply the non-parametric closed-form solution for each sample $x = (s, a, s')$ in D^I:

$$\widehat{w}_{\mu,\nu}(x) = \exp\left(\tfrac{1}{\alpha}\hat{e}_{\mu,\nu}(s,a,s') - 1\right) \text{ and } \widehat{\bar{w}}_{\mu}(x) = \exp(r(s,s') + \mu(s,s') - 1). \tag{19}$$

which is analogous to (15). Plugging this result into (18) yields a sample-based objective function for minimization[2]:

$$\begin{aligned}\min_{\mu,\nu} \widehat{\mathcal{L}}(\mu,\nu) = (1-\gamma)\hat{\mathbf{E}}_{s_0 \in D_0}[\nu(s_0)] + \hat{\mathbf{E}}_{x \in D^I}\Big[&\exp\big(r(s,s') + \mu(s,s') - 1\big) \\ &+ \alpha \exp\big(\tfrac{1}{\alpha}\hat{e}_{\mu,\nu}(s,a,s') - 1\big)\Big].\end{aligned} \tag{20}$$

Still, the variable $\mu \in \mathbb{R}^{S \times S}$ is much higher dimensional than $\nu \in \mathbb{R}^S$. So, it works as the main bottleneck for the overall optimization. Fortunately, we can further simplify (20) by eliminating its dependence on μ via exploiting an additional closed-form solution.

Proposition 3.3. *For any ν, the closed-form solution to the minimization* (20) *with respect to μ, i.e. $\mu_\nu = \arg\min_\mu \widehat{\mathcal{L}}(\mu,\nu)$, is*

$$\mu_\nu(s,s') = \tfrac{1}{1+\alpha}\big(-\alpha r(s,s') + \gamma\nu(s') - \nu(s)\big). \tag{21}$$

Using the above result in (20), we obtain the following minimization problem:

$$\min_{\widehat{\nu}} \widehat{\mathcal{L}}(\widehat{\nu}) = (1-\gamma)\hat{\mathbf{E}}_{s_0 \in D_0}[\widehat{\nu}(s_0)] + (1+\alpha)\hat{\mathbf{E}}_{x \in D^I}\left[\exp\left(\tfrac{1}{1+\alpha}\widehat{A}_{\widehat{\nu}}(s,a,s') - 1\right)\right], \tag{22}$$

where $\widehat{A}_\nu(s,a,s') := r(s,s') + \gamma\nu(s') - \nu(s)$. The remaining issue is that optimizing (22) is not practical because $\exp(\cdot)$ often causes numerical instability and gradient explosion. To address this, we use a numerically-stable alternative of (22):

Proposition 3.4. *Let $\widetilde{\mathcal{L}}(\widetilde{\nu})$ be the function:*

$$\widetilde{\mathcal{L}}(\widetilde{\nu}) = (1-\gamma)\hat{\mathbf{E}}_{s_0 \in D_0}[\widetilde{\nu}(s_0)] + (1+\alpha)\log\hat{\mathbf{E}}_{x \in D^I}\left[\exp\left(\tfrac{1}{1+\alpha}\widehat{A}_{\widetilde{\nu}}(s,a,s')\right)\right]. \tag{23}$$

Then, $\min_{\widehat{\nu}} \widehat{\mathcal{L}}(\widehat{\nu}) = \min_{\widetilde{\nu}} \widetilde{\mathcal{L}}(\widetilde{\nu})$ holds. Also, $\widetilde{\mathcal{L}}(\widetilde{\nu})$ is convex with respect to $\widetilde{\nu}$.

In order see why minimizing $\widetilde{\mathcal{L}}(\widetilde{\nu})$ no longer suffers from numerical instability, note that the gradient $\nabla_x \log \mathbb{E}_{x\sim p}[\exp(h(x))] = \mathbb{E}_{x\sim p}[\frac{\exp(h(x))}{\mathbb{E}_{\bar{x}\sim p}[\exp(h(\bar{x}))]}\nabla_x h(x)]$ normalizes $\exp(\cdot)$ by softmax and thus tames large numerical values. Finally, we can show the following connection between $\widehat{\nu}^*$ and $\widetilde{\nu}^*$:

Proposition 3.5. *Let $\widehat{V}$ and $\widetilde{V}$ be the sets $\arg\min_{\widehat{\nu}} \widehat{\mathcal{L}}(\widehat{\nu})$ and $\arg\min_{\widetilde{\nu}} \widetilde{\mathcal{L}}(\widetilde{\nu})$, respectively. Then, $\widetilde{V} = \{\widehat{\nu}^* + C \mid \widehat{\nu}^* \in \widehat{V}, C \in \mathbb{R}\}$ holds.*

Proposition 3.5 implies that an unnormalized stationary distribution corrections of an optimal policy would be obtained from $\widetilde{\nu}^* \in \widetilde{V}$.

$$\begin{aligned}\widehat{w}_{\mu_{\widehat{\nu}^*},\widehat{\nu}^*}(x) &= \exp\left(\tfrac{1}{\alpha}\big(-\mu_{\widehat{\nu}^*}(s,s') + \gamma\widehat{\nu}^*(s') - \widehat{\nu}^*(s)\big) - 1\right) \\ &= \exp\left(\tfrac{1}{1+\alpha}\widehat{A}_{\widehat{\nu}^*}(s,a,s') - 1\right) && \text{(by (21))} \\ &\propto \exp\left(\tfrac{1}{1+\alpha}\widehat{A}_{\widetilde{\nu}^*}(s,a,s')\right) =: \widetilde{w}_{\widetilde{\nu}^*}(x) && \text{(by Proposition 3.5)}\end{aligned}$$

[2] In contrast to the results in the previous sections where (16) and (11) are identical, $\widehat{\mathcal{L}}(\mu,\nu)$ in (20) is an upper bound of $\max_{w,\bar{w}} \widehat{\mathcal{L}}(w,\bar{w},\mu,\nu)$ in (18), due to applying *per-sample* closed-form solutions. While unbiasedness has been sacrificed at the cost of eliminating the nested optimization, it enables much stable optimization in practice. The upper bound gap vanishes when the transition dynamics are deterministic.

Policy extraction We must take caution when using $\widetilde{w}_{\widetilde{\nu}^*}(x)$ since it is an unnormalized density ratio, i.e. $\mathbb{E}_{x\sim d^I}[\widetilde{w}_{\widetilde{\nu}^*}(x)] \neq 1$. Therefore, to extract a policy, we perform weighted BC using self-normalized importance sampling [31]:

$$\max_{\pi} \frac{\sum_{x\in D^I} \widetilde{w}_{\widetilde{\nu}^*}(x) \log \pi(a|s)}{\sum_{x\in D^I} \widetilde{w}_{\widetilde{\nu}^*}(x)}, \tag{24}$$

which completes the derivation of the practical version of LobsDICE. To sum up, LobsDICE solves $\widetilde{\nu}^* = \arg\min_{\widetilde{\nu}} \widetilde{\mathcal{L}}(\widetilde{\nu})$ of (23) via gradient descent, and extracts a policy via self-normalized weighted BC of (24). Pseudocode for LobsDICE can be found in Appendix G.2.

4 Related Work

Learning from observation (LfO) Recent approaches for LfO are mostly *on-policy* [25, 26, 35, 39, 40] algorithms and are not directly applicable to the offline LfO setting considered in this work. MobILE [14] is a model-based LfO algorithm, but it encourages uncertainty for online exploration, which is not suitable for the offline setting. BCO [38] uses an inverse dynamics model (IDM) to infer the missing expert actions and performs BC on the generated expert's state-action dataset. In addition to common issues by vanilla BC, BCO is not guaranteed to recover the expert's behavior in general. OPOLO [45] is a principled off-policy LfO algorithm, but it solves a nested optimization and requires evaluation on OOD action values during training, which suffers from numerical instability in the offline setting. IQ-Learn [8] solves online and offline IL problems while avoiding adversarial training by learning a single Q-function. However, it also suffers from numerical instability in the offline setting by overestimating Q due to using OOD action values. RCE [5] aims to solve example-based control tasks in which a collection of example success states is assumed to be provided instead of entire expert trajectories. While RCE can be applied to LfO in principle, it tends to stay in a few expert states that are easy to reach in the offline setting, as discussed in Section C.2 of [5].

Stationary DIstribution Correction Estimation (DICE) DICE-family algorithms perform stationary distribution estimation, and many of them have been proposed for off-policy evaluation [4, 29, 42–44]. Other lines of works consider reinforcement learning [22, 23, 30], offline policy selection [41]. ValueDICE [17] and OPOLO [45] derive off-policy IL and LfO objectives using DICE. However, they suffer from numerical instability in the offline setting due to the nested min-max optimization and OOD action evaluation. DemoDICE [15] is an offline IL algorithm that directly optimizes stationary distribution as ours and reduces to solving a convex minimization. Yet, it requires expert action labels and is not directly applicable to LfO. Concurrent to our work, SMODICE [28] is an offline LfO method, aiming to match stationary *state* distributions by minimizing an objective similar to OPOLO [45] in (3). Thus it also relies on potentially loose upper bound of the divergence. In addition, matching the state distributions may not be sufficient to recover the expert's behavior, on which we provide detailed discussion in Appendix A and Remark B.1.

5 Experiments

In this section, we evaluate LobsDICE on both tabular and continuous MDPs. We use four baseline methods in tabular MDPs: BC on imperfect demonstrations, BCO [38], and OPOLO [45]. Additionally, we designed a strong baseline DemoDICEfO, which extends the state-of-the-art offline IL algorithm, DemoDICE [15]. DemoDICEfO trains an inverse dynamics model, uses it to fill the missing actions in the expert demonstrations, and then runs DemoDICE using the approximate expert demonstrations and the imperfect demonstrations. For continuous control tasks, we use two additional baselines: IQ-Learn [8] and RCE [5].

5.1 Random MDPs

We first evaluate LobsDICE and baseline algorithms on randomly generated finite MDPs using a varying number of expert/imperfect trajectories and different degrees of stochasticity in the environment . We follow the experimental protocol in previous offline RL works [20–22] but with additional control on the stochasticity of transition probabilities. We conduct repeated experiments for 1K runs. For each run, (1) a random MDP is generated, (2) expert trajectories and imperfect trajectories are collected, and (3) each offline LfO algorithm is tested on the collected offline dataset. We evaluate

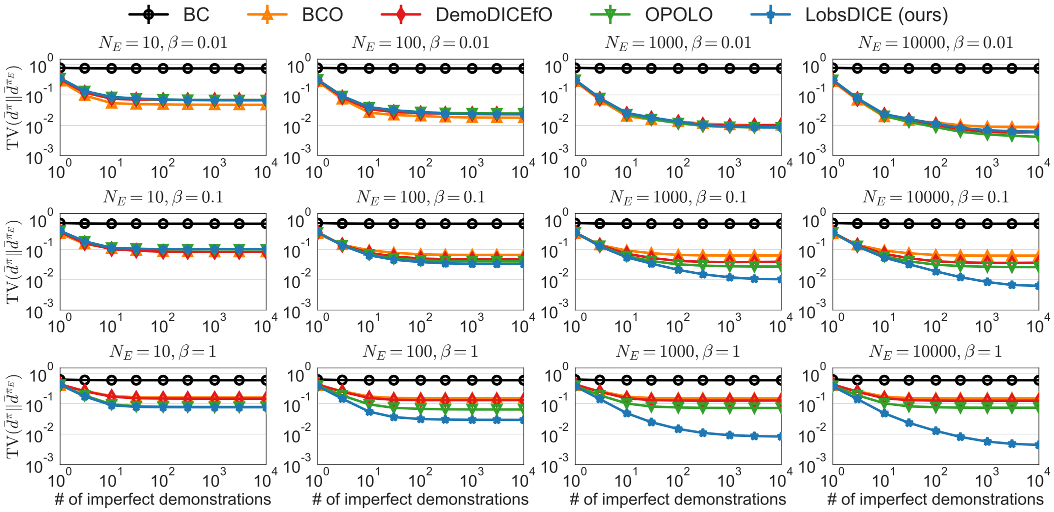

Figure 1: Performance of tabular LobsDICE and baselines in randomly generated MDPs. The first row indicates near-deterministic dynamics and the last row indicates highly stochastic dynamics. As the level of stochasticity increases, baselines fall into suboptimal, even the number of state-only expert demonstrations and imperfect demonstrations increases, while LobsDICE goes to optimal. For each algorithm, we measure the performance using total variation between state-transition stationary distributions of expert and learned policy. We plot the mean and standard error of total variations $\mathrm{TV}(\bar{d}^{\pi}(s, s'), \bar{d}^{\pi_E}(s, s'))$ over 1000 random seeds.

the performance of each algorithm by measuring the total variation distance between state-transition stationary distributions by the expert policy and the learned policy, i.e. $D_{\mathrm{TV}}(\bar{d}^{\pi}(s, s') \| \bar{d}^{\pi_E}(s, s'))$. For the tabular MDP experiments, we adopt tabular methods but not function approximation. Our tabular LobsDICE optimizes (16) on the empirical MDP model constructed from the action-labeled dataset D^I while extracting the policy through (17). The pseudocode for tabular LobsDICE can be found in Appendix G.1. The detailed experimental setup such as random MDP generation and offline dataset generation can be found in Appendix I.1.

Results Figure 1 presents the results in random MDP experiments, where β is the hyperparameter that controls the stochasticity of the underlying MDP. The first row corresponds to the case when $\beta = 0.01$ (nearly deterministic MDP). In this situation, the inverse dynamics disagreement will be close to zero, i.e. $D_{\mathrm{KL}}(d^{\pi_1}(a|s, s') || d^{\pi_2}(a|s, s')) \approx 0$ for any two policies π_1 and π_2. Thus, the algorithms whose performance directly relies on the IDM's accuracy (i.e. BCO and DemoDICEfO) even perform very well since it is very easy to learn a perfect inverse dynamics model in this scenario. OPOLO's upper bound gap (4) will also be close to zero, thus OPOLO directly minimizes the divergence of state-transition distributions. As a result, there is no performance gap among different algorithms, except for BC whose performance is determined by the quality of imperfect demonstrations. Also, the performance of all algorithms (except for BC) improves as more data is given, which is natural.

The second row and the third row in Figure 1 presents the result when $\beta = 0.1$ (weakly stochastic MDP) and $\beta = 1.0$ (highly stochastic MDP) respectively. In the stochastic MDPs, the IDM trained by the *imperfect* demonstrations faces a challenge in predicting the expert's actions accurately (more challenging as β gets larger). As a consequence, BCO gets suboptimal and its suboptimality cannot be improved even if more data is given. DemoDICEfO performs better than BCO since it additionally considers the distributional shift by considering state distribution matching. However, it is still suboptimal due to its nature that directly depends on the quality of inferred action by the learned IDM. OPOLO does not rely on the learned IDM and outperforms both BCO and DemoDICEfO. Still, OPOLO can be inherently suboptimal due to its nature of optimizing the *upper bound* unless the underlying transition dynamics are deterministic and injective. This upper bound gap (4) is not controllable by the algorithm and implies that OPOLO can be suboptimal even given an infinite

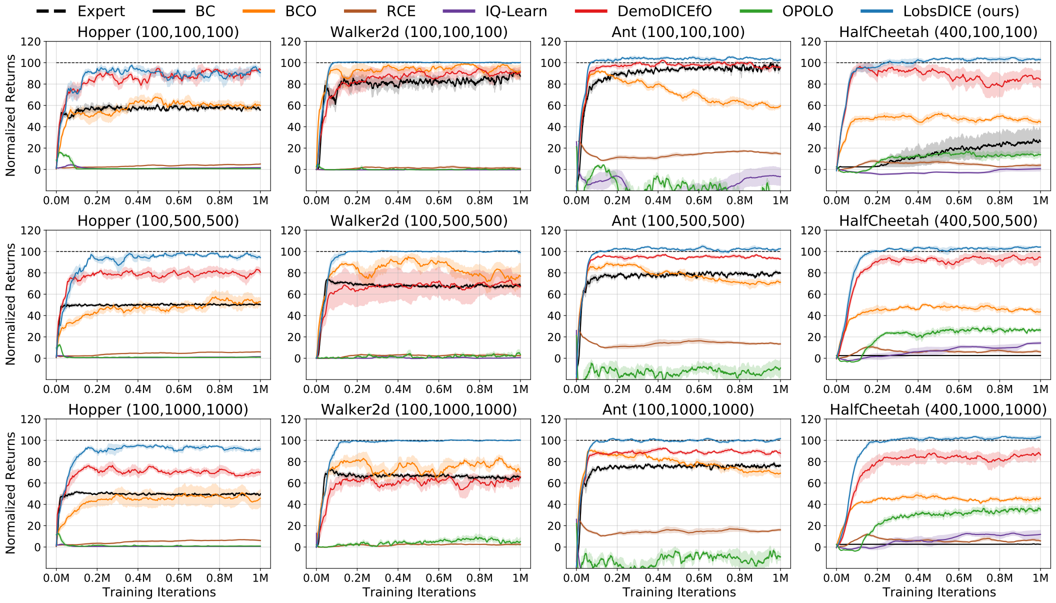

Figure 2: Performance of LobsDICE and baseline algorithms on various MuJoCo control tasks. We build state-only expert demonstrations using 5 trajectories from `expert-v2`. For each task (X, Y, Z) we construct imperfect demonstrations using X, Y, and Z trajectories from `expert-v2`, `medium-v2`, and `random-v2`, respectively. We plot the mean and the standard errors (shaded area) of the normalized scores over five random seeds.

amount of data with sufficient dataset coverage, which can be seen in the rightmost figures. Finally, our tabular LobsDICE using (16) essentially solves the exact state-transition distribution matching problem (as $\alpha \to 0$). LobsDICE is the only offline LfO algorithm that can asymptotically recovers the expert's state demonstrations even though the underlying MDP is stochastic.

5.2 Continuous control tasks (Gym-MuJoCo)

We present the empirical performance of LobsDICE and baselines on MuJoCo [37] continuous control tasks using the OpenAI Gym [3] framework. We utilize the D4RL dataset [6] for offline LfO tasks in four MuJoCo environments: Hopper, Walker2d, HalfCheetah, and Ant. Implementation details for LobsDICE and baseline algorithms such as hyperparameters and evaluation metric are provided in Appendix I.2.

Task setup For each MuJoCo environments, we employ `expert-v2`, `medium-v2`, and `random-v2` from D4RL datasets [6]. Across all environments, we consider three tasks, each of which uses different imperfect demonstrations while sharing the same expert observations. First, we construct the state-only expert demonstration $D^E = \{(s, s')_i\}_{i=1}^{N_E}$ using the first 5 trajectories in `expert-v2`. Then, we use trajectories in `expert-v2`, `medium-v2`, and `random-v2` to construct imperfect demonstrations with different ratios. We denote the composition of imperfect demonstrations as (X, Y, Z) in the title of each subplot in Figure 2, which means that the imperfect dataset consists of X trajectories from `expert-v2`, Y trajectories from `medium-v2`, and Z trajectories from `random-v2`.

Results Figure 2 summarizes that the empirical results of LobsDICE and baselines on continuous control tasks. We first remark that LobsDICE (blue) significantly outperforms OPOLO (green) in all tasks across all domains, although both LobsDICE and OPOLO are DICE-based algorithms. The failure of OPOLO comes from its numerical instability due to its dependence on nested optimization and using OOD action values during training. In contrast, LobsDICE solves a single minimization (23) while it does not involve any evaluation on OOD actions, thus it is optimized stably. IQ-Learn (purple) also suffers from numerical instability due to the usage of OOD action values during training, showing poor performance similar to OPOLO. RCE (brown) tends to stay in a few states that are easy to reach in the offline setting, rather than following the entire expert trajectories. Naive BC on imperfect demonstrations (black) is inherently suboptimal since it does not consider distribution-matching with the expert's observation at all. While BCO (orange) exploits the expert's demonstrations with the inferred actions by the IDM, its policy learning is done only on the very scarce expert dataset (i.e. 5 trajectories), which makes the algorithm perform not well. DemoDICEfO (red) exploits both expert demonstrations (where the missing actions are filled with the IDM) and the abundant imperfect demonstrations, but its performance is affected by the quality of the learned IDM. We empirically observe that the IDM error (on the true expert data) increases as the proportion of the non-expert data (i.e. `medium-v2` and `random-v2`) increases, resulting in performance degradation of both BCO and DemoDICEfO. Finally, LobsDICE is the only algorithm that was able to fully recover the expert's performance regardless of the increase of non-expert data in the imperfect demonstrations, significantly outperforming baseline algorithms. This result highlights the effectiveness of our method that solves a state-transition stationary matching problem in a principled manner. We provide additional experiments in Appendix H.

6 Conclusion

We presented LobsDICE, an algorithm for offline learning from observations (LfO), which successfully achieves state-of-the-art performance on various tabular and continuous tasks. We formulated the offline LfO as a state-transition stationary distribution matching problem, where the stationary distribution is optimized via convex minimization. Experimental results demonstrated that LobsDICE achieves promising performance in both tabular and continuous offline LfO tasks.

Acknowledgments and Disclosure of Funding

This work was supported by National Research Foundation (NRF) of Korea (NRF-2019R1A2C1087634), Field-oriented Technology Development Project for Customs Administration through National Research Foundation (NRF) of Korea funded by the Ministry of Science & ICT and Korea Customs Service (NRF-2021M3I1A1097938), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00940, Foundations of Safe Reinforcement Learning and Its Applications to Natural Language Processing; No.2022-0-00311, Development of Goal-Oriented Reinforcement Learning Techniques for Contact-Rich Robotic Manipulation of Everyday Objects; No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST); No.2021-0-02068, Artificial Intelligence Innovation Hub), and Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (22ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System). Hongseok Yang was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921) and also by the Institute for Basic Science (IBS-R029-C1). Kee-Eung Kim was supported by KAIST-NAVER Hypercreative AI Center.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[2] D. C. Bentivegna, C. G. Atkeson, and G. Cheng. Learning tasks from observation and practice. *Robotics and Autonomous Systems*, 47(2):163–169, 2004. Robot Learning from Demonstration.

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

[4] B. Dai, O. Nachum, Y. Chow, L. Li, C. Szepesvari, and D. Schuurmans. CoinDICE: Off-policy confidence interval estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9398–9411, 2020.

[5] B. Eysenbach, S. Levine, and R. R. Salakhutdinov. Replacing rewards with examples: Example-based policy search via recursive classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[6] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4RL: Datasets for deep data-driven reinforcement learning, 2020.

[7] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning (ICML)*, pages 2052–2062. PMLR, 2019.

[8] D. Garg, S. Chakraborty, C. Cundy, J. Song, and S. Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017.

[11] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[12] Y. Jin, Z. Yang, and Z. Wang. Is pessimism provably efficient for offline rl? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5084–5096, 18–24 Jul 2021.

[13] L. Ke, S. Choudhury, M. Barnes, W. Sun, G. Lee, and S. Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 313–329. Springer, 2020.

[14] R. Kidambi, J. D. Chang, and W. Sun. MobILE: Model-based imitation learning from observation alone. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=_Rtm4rYnIIL`.

[15] G.-H. Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations (ICLR)*, 2022.

[16] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[17] I. Kostrikov, O. Nachum, and J. Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations (ICLR)*, 2020.

[18] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[19] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[20] R. Laroche, P. Trichelair, and R. T. Des Combes. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning (ICML)*, pages 3652–3661. PMLR, 2019.

[21] B. Lee, J. Lee, P. Vrancx, D. Kim, and K.-E. Kim. Batch reinforcement learning with hyperparameter gradients. In *International Conference on Machine Learning (ICML)*, pages 5725–5735. PMLR, 2020.

[22] J. Lee, W. Jeon, B.-J. Lee, J. Pineau, and K.-E. Kim. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning (ICML)*, 2021.

[23] J. Lee, C. Paduraru, D. J. Mankowitz, N. Heess, D. Precup, K.-E. Kim, and A. Guez. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=FLA55mBee6Q`.

[24] Z. Li, T. Xu, Y. Yu, and Z.-Q. Luo. Rethinking valuedice: Does it really improve performance? *arXiv preprint arXiv:2202.02468*, 2022.

[25] F. Liu, Z. Ling, T. Mu, and H. Su. State alignment-based imitation learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[26] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018.

[27] Y. Ma, D. Jayaraman, and O. Bastani. Conservative offline distributional reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[28] Y. J. Ma, A. Shen, D. Jayaraman, and O. Bastani. Smodice: Versatile offline imitation learning via state occupancy matching. *arXiv preprint arXiv:2202.02433*, 2022.

[29] O. Nachum, Y. Chow, B. Dai, and L. Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[30] O. Nachum, B. Dai, I. Kostrikov, Y. Chow, L. Li, and D. Schuurmans. AlgaeDICE: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

[31] A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[32] M. L. Puterman. Markov decision processes: Discrete stochastic dynamic programming, 1994.

[33] S. Ross, G. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011.

[34] S. Schaal et al. Learning from demonstration. *Advances in neural information processing systems (NeurIPS)*, pages 1040–1046, 1997.

[35] W. Sun, A. Vemula, B. Boots, and D. Bagnell. Provably efficient imitation learning from observation alone. In *International conference on machine learning*, pages 6036–6045. PMLR, 2019.

[36] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.

[37] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[38] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[39] F. Torabi, G. Warnell, and P. Stone. Generative adversarial imitation from observation. *ICML Workshop on Imitation, Intent, and Interaction (I3)*, 2019.

[40] C. Yang, X. Ma, W. Huang, F. Sun, H. Liu, J. Huang, and C. Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[41] M. Yang, B. Dai, O. Nachum, G. Tucker, and D. Schuurmans. Offline policy selection under uncertainty, 2020.

[42] M. Yang, O. Nachum, B. Dai, L. Li, and D. Schuurmans. Off-policy evaluation via the regularized lagrangian. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[43] R. Zhang, B. Dai, L. Li, and D. Schuurmans. GenDICE: Generalized offline estimation of stationary values. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[44] S. Zhang, B. Liu, and S. Whiteson. GradientDICE: Rethinking generalized offline estimation of stationary values. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2020.

[45] Z. Zhu, K. Lin, B. Dai, and J. Zhou. Off-policy imitation learning from observations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See abstract and introduction (Section 1)
   (b) Did you describe the limitations of your work? [Yes] Due to space limit, we discuss it in the supplementary material (Section L)
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Due to space limit, we discuss it in the supplementary material (Section N).
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In appropriate places.
   (b) Did you include complete proofs of all theoretical results? [Yes] Due to space limit, we provide proofs in the supplementary material. See Section F
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We submit the code as supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specify them in the supplementary material. See Section I
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report them in each figure. See Figure 1 and 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We specify them in the supplementary material. See Section K
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite Random MDP protocols and D4RL dataset. See Section 5.
   (b) Did you mention the license of the assets? [Yes] D4RL dataset is licensed under the Apache 2.0. See Section M.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All the datasets we used are public.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] None of datasets contain identifiable contents.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Not applicable.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Not applicable.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Not applicable.

A Why State-transition Occupancy Matching Instead of State Occupancy Matching?

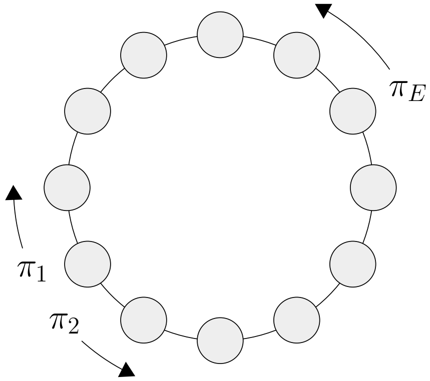

Figure 3: An example MDP with $|S| = 12$, $|A| = 2$ and $p_0(s) = \text{Unif}(s)$. In this example MDP, $\{\pi_1, \pi_2\} \subseteq \arg\min_\pi D_{\text{KL}}(d^\pi(s)||d^{\pi_E}(s))$, while $\{\pi_2\} = \arg\min_\pi D_{\text{KL}}(d^\pi(s,s')||d^{\pi_E}(s,s'))$.

For learning from observation, we adopt the objective function for $d(s,s')$-matching (i.e., $\min_\pi D_{\text{KL}}(d^\pi(s,s')||d^E(s,s'))$) instead of $d(s)$-matching (i.e. $\min_\pi D_{\text{KL}}(d^\pi(s)||d^E(s))$) since $d(s)$-matching may ignore the *directionality* of the expert trajectories. To see this, consider an MDP with $|S|$ states and 2 actions, where states are denoted as circles in Figure 3. Each action moves the agent from the current state to the neighboring state deterministically either clockwise or counterclockwise. Initial state distribution p_0 is defined as uniform distribution over the entire states. Finally, let an expert π_E be a policy that moves in a counterclockwise direction, and we want to mimic this expert's behavior. In this example, the desired imitation policy should be π_2 in Figure 3. However, if we perform $d(s)$-matching, even π_1 can be obtained as a resulting policy, which is the complete opposite of what we wanted to obtain. This is due to the fact that π_1 and π_2 share the same state stationary distribution of a uniform distribution: $d^\pi(s) = \frac{1}{|S|}\ \forall s$. Thus, both π_1 and π_2 must be an optimal solution of $\min_\pi D_{\text{KL}}(d^\pi(s)||d^E(s))$. In contrast, the $d(s,s')$-matching has a unique solution of π_2, exploiting the directionality information (i.e. moving counterclockwise) of the expert's trajectories.

B LobsDICE for State Occupancy Matching

While we have provided a counter-example where $d(s)$-matching can fail to recover the expert's behavior in Appendix A, we still can derive an offline LfO algorithm for $d(s)$-matching. For the given policy π, its state stationary distribution is defined as:

$$\bar{d}^\pi(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s).$$

Then, similar to the state-transition occupancy matching objective (5), we formulate a state occupancy matching problem as follows:

$$\min_\pi D_{\text{KL}}(\bar{d}^\pi(s)\|d^E(s)) + \alpha D_{\text{KL}}(d^\pi(s,a)\|d^I(s,a)). \tag{25}$$

where the hyperparameter $\alpha > 0$ balances between encouraging state matching and preventing distribution shift from the distribution of imperfect demonstrations. This objective can be reformulated in terms of optimizing stationary distribution:

$$\max_{d,\bar{d}\geq 0} \ -D_{\text{KL}}(\bar{d}(s)\|d^E(s)) - \alpha D_{\text{KL}}(d(s,a)\|d^I(s,a)) \tag{26}$$

$$\text{s.t.} \ \sum_{a'} d(s',a') = (1-\gamma)p_0(s') + \gamma\sum_{s,a} d(s,a)T(s'|s,a) \quad \forall s', \tag{27}$$

$$\sum_a d(s,a) = \bar{d}(s) \quad \forall s, \tag{28}$$

Then, we consider the Lagrangian for the constrained optimization (26-28):

$$\begin{aligned}\min_{\mu,\nu}\max_{d,\bar{d}\geq 0} -\mathbb{E}_{\bar{d}}\Big[\log\tfrac{\bar{d}(s)}{d^E(s)}\Big] - \mathbb{E}_{d}\Bigg[\log\tfrac{d(s,a)}{d^I(s,a)}\Bigg] + \sum_s \mu(s)\big(\bar{d}(s) - \sum_a d(s,a)\big) \\ + \sum_{s'}\nu(s')\big((1-\gamma)p_0(s') + \gamma\sum_{s,a} d(s,a)T(s'|s,a) - \textstyle\sum_{a'} d(s',a')\big)\end{aligned} \tag{29}$$

where $\nu(s') \in \mathbb{R}$ are the Lagrange multipliers for the Bellman flow constraints (27), and $\mu(s) \in \mathbb{R}$ are the Lagrange multipliers for the marginalization constraints (28). To make the optimization tractable in the offline setting, we rearrange the terms in (29), introducing new optimization variables w and $\bar{w}$ that denote stationary distribution correction ratios for (s,a) and s respectively.

$$\begin{aligned}(29) &= \min_{\mu,\nu}\max_{d,\bar{d}\geq 0}(1-\gamma)\mathbb{E}_{s_0\sim p_0}[\nu(s_0)] + \mathbb{E}_{s\sim\bar{d}}\Big[\mu(s) - \log\underbrace{\tfrac{\bar{d}(s)}{d^I(s)}}_{=:\bar{w}(s)} + \underbrace{\log\tfrac{d^E(s)}{d^I(s)}}_{=:r(s)}\Big] \\ &\quad + \mathbb{E}_{(s,a)\sim d}\Big[\underbrace{-\mu(s) + \mathbb{E}_{s'}[\gamma\nu(s')] - \nu(s)}_{=:e_{\mu,\nu}(s,a)} - \alpha\log\underbrace{\tfrac{d(s,a)}{d^I(s,a)}}_{=:w(s,a)}\Big] \\ &= \min_{\mu,\nu}\max_{w,\bar{w}\geq 0}(1-\gamma)\mathbb{E}_{s_0\sim p_0}[\nu(s_0)] + \mathbb{E}_{s\sim d^I}\left[\bar{w}(s)\big(r(s) + \mu(s) - \log\bar{w}(s)\big)\right] \\ &\quad + \mathbb{E}_{(s,a)\sim d^I}\left[w(s,a)\big(e_{\mu,\nu}(s,a) - \alpha\log w(s,a)\big)\right] =: \mathcal{L}(w,\bar{w},\mu,\nu).\end{aligned} \tag{30}$$

We introduce the log ratio $r(s) = \log\frac{d^E(s)}{d^I(s)}$ to make expectation for d^I instead of d^E. The log ratio $r(s)$ can be estimated by using a pretrained discriminator $c : S \to [0,1]$, where c is trained by:

$$c^* = \operatorname*{arg\,max}_{c:S\to[0,1]}\mathbb{E}_{s\sim d^E}[\log c(s)] + \mathbb{E}_{s\sim d^I}[\log c(s)].$$

Then, it is proven that the optimal discriminator c^* satisfies $c^*(s) = \frac{d^E(s)}{d^E(s)+d^I(s)}$. Therefore, $r(s)$ can be estimated by:

$$r(s) = -\log\left(\frac{1}{c^*(s)} - 1\right).$$

Similar to Proposition 3.1, we can easily derive the closed-form solution to the inner maximization in (30):

$$w_{\mu,\nu}(s,a) = \exp\left(\tfrac{1}{\alpha}e_{\mu,\nu}(s,a) - 1\right),$$

$$\bar{w}_{\mu}(s) = \exp(r(s) + \mu(s) - 1).$$

Finally, by plugging these closed-form solutions into (30), the nested min-max optimization of (30) is reduced to a single minimization:

$$\begin{aligned}\min_{\mu,\nu}\mathcal{L}(w_{\mu,\nu},\bar{w}_{\mu},\mu,\nu) &= (1-\gamma)\mathbb{E}_{s_0\sim p_0}[\nu(s_0)] + \mathbb{E}_{s\sim d^I}\left[\exp\big(r(s) + \mu(s) - 1\big)\right] \\ &\quad + \alpha\mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}e_{\mu,\nu}(s,a) - 1\right)\right],\end{aligned} \tag{31}$$

which is a convex function of μ and ν. Finally, once we obtain the optimal solution (μ^*,ν^*) of (31), we have $w_{\mu^*,\nu^*}(s,a) = \frac{d^{\pi^*}(s,a)}{d^I(s,a)}$. Then, we can extract an optimal policy via weighted BC:

$$\max_{\pi}\mathbb{E}_{(s,a)\sim d^{\pi^*}}[\log\pi(a|s)] = \mathbb{E}_{(s,a)\sim d^I}[w_{\mu^*,\nu^*}(s,a)\log\pi(a|s)] \tag{32}$$

Remark B.1. To the best of our knowledge, (31) is the first result to directly solve the state distribution matching problem in an offline setting. Although SMODICE [28], a concurrent work with ours, also aims to solve offline LfO in terms of optimizing stationary distribution, it essentially minimizes the (potentially loose) *upper bound*, i.e.,

$$\min_{\pi}\mathbb{E}_{s\sim d^\pi}\left[\log\frac{d^I(s)}{d^E(s)}\right] + D_{\mathrm{KL}}(d^\pi(s,a)||d^I(s,a)) \tag{33}$$

$$\geq\mathbb{E}_{s\sim d^\pi}\left[\log\frac{d^I(s)}{d^E(s)}\right] + D_{\mathrm{KL}}(d^\pi(s)||d^I(s)) \tag{34}$$

$$= D_{\mathrm{KL}}(d^\pi(s)||d^E(s)) \tag{35}$$

The upper bound gap $(33) - (35)$ is given by $D_{\mathrm{KL}}(\pi(a|s)||\pi^I(a|s))$. Unlike OPOLO of (4), this upper bound gap does not vanish even when the transition is deterministic. Consequently, SMODICE may not be able to precisely recover the expert's state visitations even given an infinite amount of data due to the upper bound gap unless d^I is collected purely by expert. In contrast, we are directly minimizing the divergence: solving (31) is exactly equivalent to solving (25). As can be seen in Figure 4, SMODICE optimizing (33) fails to recover the expert's behavior due to the upper bound gap that is irreducible even for deterministic MDPs.

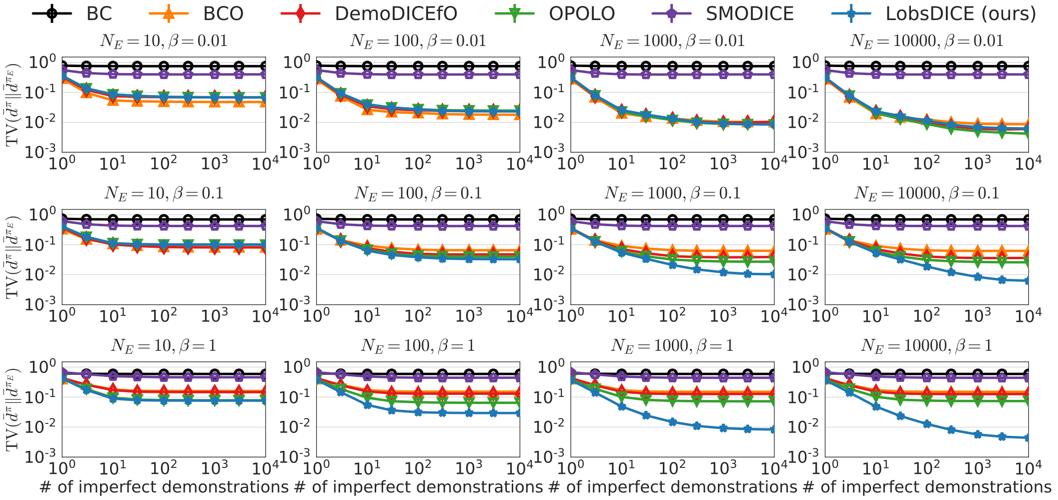

Figure 4: Performance of tabular LobsDICE and baselines in randomly generated MDPs. The first row indicates near-deterministic dynamics and the last row indicates highly stochastic dynamics. As the level of stochasticity increases, baselines fall into suboptimal, even the number of state-only expert demonstrations and imperfect demonstrations increases, while LobsDICE goes to optimal. For each algorithm, we measure the performance using total variation between state-transition stationary distributions of expert and learned policy. We plot the mean and standard error of total variations $\mathrm{TV}(\bar{d}^{\pi}(s,s'), \bar{d}^{\pi_E}(s,s'))$ over 1000 random seeds.

C Validity of Weighted Behavior Cloning for Policy Extraction

For tabular MDPs, we can show that the weighted behavior cloning of (17)

$$\max_{\pi} \mathbb{E}_{(s,a)\sim d^I}[w_{\mu^*,\nu^*}(s,a) \log \pi(a|s)] \tag{17}$$

exactly yields an optimal policy π^* of the original optimization problem (5). First, the optimization problem (5) and the constrained optimization problem (6-8) are equivalent with the following relationship between π^* and d^*:

$$\pi^*(a|s) = \frac{d^*(s,a)}{\sum_{a'} d^*(s,a')} \tag{36}$$

where d^* is the optimal solution of both (6-8) and (10). Then, $w_{\mu^*,\nu^*}(s,a)$ is computed by (15) with (μ^*,ν^*), the optimal solution of both (16) and (11). Consequently, $w_{\mu^*,\nu^*}(s,a) = \frac{d^*(s,a)}{d^I(s,a)}$ holds, and thus the weighted behavior cloning of (17) essentially performs the following:

$$\max_{x} \sum_{s,a} d^*(s,a) \log x(a|s) \tag{37}$$

$$\text{s.t. } \sum_{a} x(a|s) = 1 \quad \forall s \tag{38}$$

Now, consider the Lagrangian for the constrained optimization (37-38):

$$L := \sum_{s,a} d^*(s,a) \log x(a|s) - \sum_{s} \lambda(s) \big(\sum_{a} x(a|s) - 1 \big) \tag{39}$$

where $\lambda(s)$ is the Lagrange multiplier for the constraint (38). Then,

$$\frac{\partial L}{\partial x(a|s)} = \frac{d^*(s,a)}{x(a|s)} - \lambda(s) = 0 \quad \Rightarrow x^*(a|s) = \frac{d^*(s,a)}{\lambda(s)} \; \forall s,a \tag{40}$$

Finally, considering the constraint $\sum_a x^*(a|s) = 1 \; \forall s$, we obtain:

$$x^*(a|s) = \frac{d^*(s,a)}{\sum_{a'} d^*(s,a')} \tag{41}$$

which is identical to (36). In summary, the weighted behavior cloning performs the process of extracting the optimal policy π^* from w_{μ^*,ν^*}.

D Challenges of Extending DemoDICE to Offline Learning from Observation

DemoDICE [15] is an algorithm that optimizes the state-action stationary distribution $d(s,a)$, where its naive application to (s,s')-distribution matching problem yields:

$$\min_{d \geq 0} \overbrace{\mathbb{E}_{\substack{(s,a)\sim d \\ s'\sim T(s,a)}} \Big[\log \tfrac{\sum_a d(s,a)T(s'|s,a)}{d^E(s,s')} \Big]}^{=D_{\mathrm{KL}}(\bar{d}(s,s')||\bar{d}^E(s,s'))} + \alpha \overbrace{\mathbb{E}_{(s,a)\sim d}\Big[\log \tfrac{d(s,a)}{d^I(s,a)} \Big]}^{=D_{\mathrm{KL}}(d(s,a)||d^I(s,a))} \tag{42}$$

$$\text{s.t. } \sum_{a'} d(s',a') = (1-\gamma)p_0(s') + \gamma \sum_{s,a} d(s,a)T(s'|s,a) \quad \forall s', \tag{43}$$

However, estimating $\mathbb{E}_{\substack{(s,a)\sim d \\ s'\sim T(s,a)}} \Big[\log \tfrac{\sum_a d(s,a)T(s'|s,a)}{d^E(s,s')} \Big]$ is intractable in the offline setting due to its inclusion of marginalization over a inside the $\log(\cdot)$. This challenge can be addressed by using the *upper bound* proposed by OPOLO [45]:

$$D_{\mathrm{KL}}(\bar{d}^\pi(s,s')||d^E(s,s') \leq \mathbb{E}_{(s,s')\sim \bar{d}^\pi(s,s')}\Big[\log \tfrac{\bar{d}^I(s,s')}{\bar{d}^E(s,s')} \Big] + D_{\mathrm{KL}}(d^\pi(s,a) \| d^I(s,a)), \tag{3}$$

Replacing $D_{\mathrm{KL}}(\bar{d}^\pi(s,s')||d^E(s,s'))$ with the (potentially loose) *upper bound* in the objective function yields the following tractable (but biased) optimization problem:

$$\min_{d\geq 0} \mathbb{E}_{\substack{(s,a)\sim d \\ s'\sim T(s,a)}} \Big[\log \tfrac{\bar{d}^I(s,s')}{\bar{d}^E(s,s')} \Big] + (1+\alpha)\mathbb{E}_{(s,a)\sim d}\Big[\log \tfrac{d(s,a)}{d^I(s,a)} \Big] \tag{44}$$

$$\text{s.t. } \sum_{a'} d(s',a') = (1-\gamma)p_0(s') + \gamma \sum_{s,a} d(s,a)T(s'|s,a) \quad \forall s', \tag{45}$$

In contrast, we address the intractability challenge by introducing an additional optimization variable $\bar{d}(s,s')$ along with the marginalization constraint (8). This allows for tractable offline optimization by (11,16) without introducing bias, which is our novel contribution.

E More Discussions on LobsDICE with Finite Samples

In this section, we provide additional discussions on LobsDICE with finite samples.

E.1 LobsDICE with sample-based approximation: distribution matching on the MLE MDP

We first show that LobsDICE that optimizes (18) is equivalent to performing optimization of (6-8) on the MLE MDP constructed by D^I. Let $\hat{M} \setminus R = \langle S, A, \hat{T}, \widehat{p}_0, \gamma \rangle$ be the MLE MDP constructed using D^I. Then, constrained optimization (6-8) for the MLE MDP $\hat{M} \setminus R$ can be formulated as:

$$\max_{d,\bar{d}\geq 0} \; - D_{\mathrm{KL}}(\bar{d}(s,s') \| \widehat{\bar{d}}^E(s,s')) - \alpha D_{\mathrm{KL}}(d(s,a) \| \widehat{d}^I(s,a)) \tag{46}$$

$$\text{s.t. } \sum_{a'} d(s',a') = (1-\gamma)\widehat{p}_0(s') + \gamma \sum_{s,a} d(s,a)\hat{T}(s'|s,a) \quad \forall s', \tag{47}$$

$$\sum_a d(s,a)\hat{T}(s'|s,a) = \bar{d}(s,s') \quad \forall s,s', \tag{48}$$

where $\widehat{p}_0$, $\widehat{d}^E$, and $\widehat{d}^I$ are empirical distributions of D_0, D^E, and D^I, respectively. We consider Lagrangian for the above constrained optimization:

$$
\begin{aligned}
&\min_{\mu,\nu} \max_{d,\bar{d}\geq 0} -\mathbb{E}_{\bar{d}}\Big[\log \tfrac{\bar{d}(s,s')}{\widehat{d}^E(s,s')}\Big] - \alpha\mathbb{E}_{d}\Big[\log \tfrac{d(s,a)}{\widehat{d}^I(s,a)}\Big] + \sum_{s,s'} \mu(s,s')\big(\bar{d}(s,s') - \sum_a d(s,a)\hat{T}(s'|s,a)\big) \\
&+ \sum_{s'} \nu(s')\big((1-\gamma)\widehat{p}_0(s') + \gamma \sum_{s,a} d(s,a)\hat{T}(s'|s,a) - \sum_{a'} d(s',a')\big) \\
=& \min_{\mu,\nu} \max_{d,\bar{d}\geq 0} (1-\gamma)\mathbb{E}_{s_0\sim\widehat{p}_0}[\nu(s_0)] + \mathbb{E}_{(s,s')\sim\bar{d}}\Big[\mu(s,s') - \log \underbrace{\tfrac{\bar{d}(s,s')}{\widehat{d}^I(s,s')}}_{=:\bar{w}(s,s')} + \log \underbrace{\tfrac{\widehat{d}^E(s,s')}{\widehat{d}^I(s,s')}}_{=:\widehat{r}(s,s')}\Big] \\
&\qquad + \mathbb{E}_{(s,a)\sim d}\Big[\underbrace{\mathbb{E}_{s'\sim\hat{T}(\cdot|s,a)}[-\mu(s,s') + \gamma\nu(s')] - \nu(s)}_{=:\widehat{e}^I_{\mu,\nu}(s,a)} - \alpha \log \underbrace{\tfrac{d(s,a)}{\widehat{d}^I(s,a)}}_{=:w(s,a)}\Big] \\
=& \min_{\mu,\nu} \max_{w,\bar{w}\geq 0} (1-\gamma)\mathbb{E}_{s_0\sim\widehat{p}_0}[\nu(s_0)] + \mathbb{E}_{(s,s')\sim\widehat{d}^I}\big[\bar{w}(s,s')\big(\widehat{r}(s,s') + \mu(s,s') - \log \bar{w}(s,s')\big)\big] \\
&\qquad + \mathbb{E}_{(s,a)\sim\widehat{d}^I}\big[w(s,a)\big(\widehat{e}^I_{\mu,\nu}(s,a) - \alpha\log w(s,a)\big)\big] =: \mathcal{L}^I(w,\bar{w},\mu,\nu),
\end{aligned}
$$

Note that this is identical to (18). We can also derive the closed-form solution $(w^I_{\mu,\nu}, \bar{w}^I_{\mu})$ to the $\max_{w,\bar{w}\geq 0} \mathcal{L}^I(w,\bar{w},\mu,\nu)$:

$$
w^I_{\mu,\nu}(s,a) = \exp\Big(\tfrac{1}{\alpha}\widehat{e}^I_{\mu,\nu}(s,a) - 1\Big) \;\; \textit{and} \;\; \bar{w}^I_{\mu}(s,s') = \exp(\widehat{r}(s,s') + \mu(s,s') - 1). \tag{49}
$$

Note that this is different from (19) in that (49) is the closed-form solution for each (s,a) and (s,s'), while (19) is the *non-parametric* closed-form solution for each sample $x \in D^I$. Exploiting (49) does *not* introduce an additional bias, but requires evaluation of the expectation within the $\exp(\cdot)$, i.e. transition model is needed.

The last step is to plug the solution $(w^I_{\mu,\nu}, \bar{w}^I_{\mu})$ in (49) into $\mathcal{L}^I(w,\bar{w},\mu,\nu)$:

$$
\begin{aligned}
&\min_{\mu,\nu} \mathcal{L}^I(w^I_{\mu,\nu}, \bar{w}^I_{\mu}, \mu, \nu) = (1-\gamma)\mathbb{E}_{s\sim\widehat{p}_0}[\nu(s)] + \mathbb{E}_{(s,s')\sim\widehat{d}^I}\Big[\exp\big(\widehat{r}(s,s') + \mu(s,s') - 1\big)\Big] \\
&\qquad + \alpha\mathbb{E}_{(s,a)\sim\widehat{d}^I}\Big[\exp\big(\tfrac{1}{\alpha}\widehat{e}^I_{\mu,\nu}(s,a) - 1\big)\Big].
\end{aligned}
$$

Finally, this can be rewritten as:

$$
\begin{aligned}
&\min_{\mu,\nu} J(\mu,\nu) = (1-\gamma)\sum_s \widehat{p}_0(s)\nu(s) + \sum_{s,s'} \widehat{d}^I(s,s')\Big[\exp\big(\widehat{r}(s,s') + \mu(s,s') - 1\big)\Big] \\
&\qquad + \alpha\sum_{s,a} \widehat{d}^I(s,a)\Big[\exp\Big(\tfrac{1}{\alpha}\sum_{s'} \widehat{T}^I(s'|s,a)\big(-\mu(s,s') + \gamma\nu(s') - \nu(s)\big) - 1\Big)\Big],
\end{aligned}
$$

which is the objective function for our tabular LobsDICE (52).

E.2 When will LobsDICE reduce to BCO?

LobsDICE reduces to BCO when D^I is collected by an *expert* policy and D^E is a dataset identical to D^I except that action is missing, i.e., $D^E = \{(s,s')|(s,a,s') \in D^I\}$.

In this situation, note that $\widehat{d}^I(s,s')$ and $\widehat{d}^I(s,a)$ are valid stationary distributions on the MLE MDP constructed by D^I, since they satisfy both the Bellman flow constraint (47) and the marginalization constraint (48) on the MLE MDP. The $\widehat{d}^I(s,s')$ and $\widehat{d}^I(s,a)$ are also the optimal solution of (46-48), making the KL divergences to 0:

$$
D_{\mathrm{KL}}(\widehat{d}^I(s,s')\|\widehat{d}^E(s,s')) = 0 \text{ and } D_{\mathrm{KL}}(\widehat{d}^I(s,a)\|\widehat{d}^I(s,a)) = 0.
$$

Finally, the policy obtained by LobsDICE will simply reduce to BC on D^I by noting that:

$$
\pi^*(a|s) = \frac{\widehat{d}^I(s,a)}{\sum_{a'} \widehat{d}^I(s,a')}
$$

Furthermore, one can easily show that BC on D^I is identical to BCO (i.e. BC on D^E where the missing actions are inferred by the IDM trained by D^I), which concludes that LobsDICE is equivalent to BCO in this special case.

However, D^I will include demonstrations collected by *non-expert* policies in general, which makes the LobsDICE's solution different from (and usually better than) the BCO's solution in general cases.

F Theoretical Analysis

F.1 Closed-form solutions

Proposition 3.1. *For any* (μ, ν)*, the closed-form solution to the inner maximization of* (11)*, i.e.* $(w_{\mu,\nu}, \bar{w}_\mu) = \arg\max_{w,\bar{w}\geq 0} \mathcal{L}(w, \bar{w}, \mu, \nu)$*, is given by:*

$$w_{\mu,\nu}(s,a) = \exp\left(\tfrac{1}{\alpha} e_{\mu,\nu}(s,a) - 1\right) \text{ and } \bar{w}_\mu(s,s') = \exp(r(s,s') + \mu(s,s') - 1). \tag{15}$$

Proof. Let $e_{\mu,\nu}(s,a,s') = -\mu(s,s') + \gamma\nu(s') - \nu(s)$. Then, $e_{\mu,\nu}(s,a) = \mathbb{E}_{s'\sim T(\cdot|s,a)}[e_{\mu,\nu}(s,a,s')]$. For (s,a) with $d^I(s,a) > 0$, we can compute the derivative $\frac{\partial \mathcal{L}}{\partial w(s,a)}$ of $\mathcal{L}$ w.r.t. $w(s,a)$ as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w(s,a)} &= \sum_{s'} d^I(s,a,s')(e_{\mu,\nu}(s,a,s') - \alpha \log w(s,a) - \alpha) = 0 \\
&\Leftrightarrow \sum_{s'} T(s'|s,a)(e_{\mu,\nu}(s,a,s') - \alpha \log w(s,a) - \alpha) = 0 \\
&\Leftrightarrow \sum_{s'} T(s'|s,a)(e_{\mu,\nu}(s,a,s') - \alpha) = \alpha \log w(s,a) \\
&\Leftrightarrow w(s,a) = \exp\left(\tfrac{1}{\alpha}\mathbb{E}_{s'\sim T(\cdot|s,a)}[e_{\mu,\nu}(s,a,s')] - 1\right) = \exp\left(\tfrac{1}{\alpha} e_{\mu,\nu}(s,a) - 1\right).
\end{aligned}$$

Similar to the aforementioned derivation, when $\bar{d}^I(s,s') > 0$, we can derive the derivative of $\mathcal{L}$ w.r.t. $\bar{w}(s,s')$:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \bar{w}(s,s')} &= \bar{d}^I(s,s')\big(\mu(s,s') - \log \bar{w}(s,s') + r(s,s') - 1\big) = 0 \\
&\Leftrightarrow \big(\mu(s,s') - \log \bar{w}(s,s') + r(s,s') - 1\big) = 0 \\
&\Leftrightarrow \log \bar{w}(s,s') = \mu(s,s') + r(s,s') - 1 \\
&\Leftrightarrow \bar{w}(s,s') = \exp\big(\mu(s,s') + r(s,s') - 1\big).
\end{aligned}$$

□

Proposition 3.2. $\mathcal{L}(w_{\mu,\nu}, \bar{w}_\mu, \mu, \nu)$ *is convex with respect to* μ *and* ν*.*

Proof. Using the fact that $\exp(\cdot)$ is a convex function, we can easily prove the convexity of $\mathcal{L}(w_{\mu,\nu}, \bar{w}_{\mu,\nu}, \mu, \nu)$. For brevity, let $\mathcal{L}(\mu,\nu) := \mathcal{L}(w_{\mu,\nu}, \bar{w}_{\mu,\nu}, \mu, \nu)$. Then, for given (μ_1, ν_1), (μ_2, ν_2) and $t \in [0,1]$,

$$\begin{aligned}
&\mathcal{L}\left(t\mu_1 + (1-t)\mu_2, t\nu_1 + (1-t)\nu_2\right) \\
&= (1-\gamma)\mathbb{E}_{p_0}[t\nu_1(s) + (1-t)\nu_2(s)] + \mathbb{E}_{d^I}[\exp(r(s,s') + t\mu_1(s,s') + (1-t)\mu_2(s,s') - 1)] \\
&\quad + \alpha\mathbb{E}_{d^I}\Big[\exp\Big(\tfrac{1}{\alpha}\mathbb{E}_{s'}\big[-t\mu_1(s,s') - (1-t)\mu_2(s,s') + \gamma t\nu_1(s') + \gamma(1-t)\nu_2(s') \\
&\qquad\qquad - t\nu_1(s) - (1-t)\nu_2(s)\big] - 1\Big)\Big] \\
&= t(1-\gamma)\mathbb{E}_{p_0}[\nu_1(s)] + (1-t)(1-\gamma)\mathbb{E}_{p_0}[\nu_2(s)] + \mathbb{E}_{d^I}[\exp(r(s,s') + t\mu_1 + (1-t)\mu_2 - 1)] \\
&\quad + \alpha\mathbb{E}_{d^I}\Big[\exp\Big(\tfrac{1}{\alpha}\mathbb{E}_{s'}\big[-t\mu_1(s,s') - (1-t)\mu_2(s,s') + \gamma t\nu_1(s') + \gamma(1-t)\nu_2(s') \\
&\qquad\qquad - t\nu_1(s) - (1-t)\nu_2(s)\big] - 1\Big)\Big]
\end{aligned}$$

$$\begin{aligned}
(Cont.) \leq\ & t(1-\gamma)\mathbb{E}_{p_0}[\nu_1(s)] + (1-t)(1-\gamma)\mathbb{E}_{p_0}[\nu_2(s)] \\
& + t\mathbb{E}_{d^I}[\exp(r(s,s') + \mu_1(s,s') - 1)] + (1-t)\mathbb{E}_{d^I}[\exp(r(s,s') + \mu_2(s,s') - 1)] \\
& + \alpha\mathbb{E}_{d^I}\Big[\exp\Big(\tfrac{1}{\alpha}\mathbb{E}_{s'}\big[-t\mu_1(s,s') - (1-t)\mu_2(s,s') + \gamma t\nu_1(s') + \gamma(1-t)\nu_2(s') \\
& \qquad\qquad\qquad\qquad - t\nu_1(s) - (1-t)\nu_2(s)\big] - 1\Big)\Big] \\
\leq\ & t(1-\gamma)\mathbb{E}_{p_0}[\nu_1(s)] + (1-t)(1-\gamma)\mathbb{E}_{p_0}[\nu_2(s)] \\
& + t\mathbb{E}_{d^I}[\exp(r(s,s') + \mu_1(s,s') - 1)] + (1-t)\mathbb{E}_{d^I}[\exp(r(s,s') + \mu_2(s,s') - 1)] \\
& + t\alpha\mathbb{E}_{d^I}\Big[\exp\Big(\tfrac{1}{\alpha}\mathbb{E}_{s'}\big[\mu_1(s,s') + \gamma\nu_1(s') - \nu_1(s)\big] - 1\Big)\Big] \\
& + (1-t)\alpha\mathbb{E}_{d^I}\Big[\exp\Big(\tfrac{1}{\alpha}\mathbb{E}_{s'}\big[-\mu_2(s,s') + \gamma\nu_2(s') - \nu_2(s)\big] - 1\Big)\Big] \\
=\ & t\mathcal{L}(\mu_1,\nu_1) + (1-t)\mathcal{L}(\mu_2,\nu_2).
\end{aligned}$$

For the inequalities in the above formulation, we use the fact that $\mathbb{E}_{d^I}[\exp(\cdot)]$ is a instance of convex functions. □

Proposition 3.3. *For any ν, the closed-form solution to the minimization* (20) *with respect to μ, i.e. $\mu_\nu = \arg\min_\mu \widehat{\mathcal{L}}(\mu,\nu)$, is*

$$\mu_\nu(s,s') = \tfrac{1}{1+\alpha}\big(-\alpha r(s,s') + \gamma\nu(s') - \nu(s)\big). \tag{21}$$

Proof. For simplicity, let $\widehat{\mathcal{L}}(\mu,\nu) := \widehat{\mathcal{L}}(w_{\mu,\nu}, \bar{w}_{\mu,\nu}, \mu, \nu)$ and $x := (s,a,s')$. Then,

$$\begin{aligned}
\tfrac{\partial\widehat{\mathcal{L}}}{\partial\mu(x)} &= d^I(x)\Big[\exp(r(x) + \mu(x) - 1) - \exp\Big(\tfrac{1}{\alpha}(-\mu(x) + \gamma\nu(s') - \nu(s)) - 1\Big)\Big] = 0 \\
&\Leftrightarrow \exp(r(x) + \mu(x) - 1) = \exp\Big(\tfrac{1}{\alpha}(-\mu(x) + \gamma\nu(s') - \nu(s)) - 1\Big) \\
&\Leftrightarrow \alpha(r(x) + \mu(x)) = -\mu(x) + \gamma\nu(s') - \nu(s) \\
&\Leftrightarrow (1+\alpha)\mu(x) = -\alpha r(x) + \gamma\nu(s') - \nu(s) \\
&\Leftrightarrow \mu(x) = \tfrac{1}{1+\alpha}\big(-\alpha r(x) + \gamma\nu(s') - \nu(s)\big)
\end{aligned}$$

□

F.2 Surrogate Objective

We first show a property of $\tilde{\mathcal{L}}$ that will be used to prove propositions:

Lemma F.1. *For given function $\nu : S \to \mathbb{R}$,*

$$\tilde{\mathcal{L}}(\nu) = \tilde{\mathcal{L}}(\nu + C) \quad \forall C \in \mathbb{R}.$$

Proof. For any $C \in \mathbb{R}$,

$$\begin{aligned}
&\tilde{\mathcal{L}}(\nu + C) \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu+C}(s,a,s')\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\big(r(s,s') + \gamma(\nu(s') + C) - (\nu(s) + C)\big)\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\big(r(s,s') + \gamma\nu(s') - \nu(s)\big)\Big)\exp\Big(\tfrac{(\gamma-1)C}{1+\alpha}\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\big(r(s,s') + \gamma\nu(s') - \nu(s)\big)\Big)\Big] + (\gamma-1)C \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\big(r(s,s') + \gamma\nu(s') - \nu(s)\big)\Big)\Big] \\
&= \tilde{\mathcal{L}}(\nu).
\end{aligned}$$

□

Now, we prove Proposition 3.4 and Proposition 3.5:

Proposition 3.4. *Let $\widetilde{\mathcal{L}}(\widetilde{\nu})$ be the function:*

$$\widetilde{\mathcal{L}}(\widetilde{\nu}) = (1-\gamma)\hat{\mathbf{E}}_{s_0 \in D_0}[\widetilde{\nu}(s_0)] + (1+\alpha)\log \hat{\mathbf{E}}_{x \in D^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\widehat{A}_{\widetilde{\nu}}(s,a,s')\Big)\Big]. \qquad (23)$$

Then, $\min_{\widehat{\nu}} \widehat{\mathcal{L}}(\widehat{\nu}) = \min_{\widetilde{\nu}} \widetilde{\mathcal{L}}(\widetilde{\nu})$ *holds. Also,* $\widetilde{\mathcal{L}}(\widetilde{\nu})$ *is convex with respect to* $\widetilde{\nu}$*.*

Proof. First of all, from the fact that $\log(x+1) \leq x$ for all $x > -1$, we can easily conclude that $\min_\nu \tilde{\mathcal{L}}(\nu) \leq \min_\nu \hat{\mathcal{L}}(\nu)$. Now, we will show that $\min_\nu \hat{\mathcal{L}}(\nu) \leq \min_\nu \tilde{\mathcal{L}}(\nu)$. For given $\nu^* \in \arg\min_\nu \tilde{\mathcal{L}}(\nu)$, we define a constant C as follows:

$$C = \tfrac{1+\alpha}{1-\gamma} \log \mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s') - 1\Big)\Big],$$

which implies

$$\exp\Big(\tfrac{(1-\gamma)C}{1+\alpha}\Big) = \mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s') - 1\Big)\Big].$$

Then, from the following two equations:

$$\begin{aligned}
&\hat{\mathcal{L}}(\nu^* + C) \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*+C}(s,a,s') - 1\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s') - 1\Big)\exp\Big(\tfrac{(\gamma-1)C}{1+\alpha}\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s') - 1\Big)\Big]\exp\Big(-\tfrac{(1-\gamma)C}{1+\alpha}\Big) \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)
\end{aligned}$$

and

$$\begin{aligned}
&\tilde{\mathcal{L}}(\nu^* + C) \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*+C}(s,a,s')\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s')\Big)\Big] + (\gamma-1)C \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s')\Big)\Big] \\
&\quad - (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s') - 1\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + (1+\alpha)\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s')\Big)\Big] \\
&\quad - (1+\alpha)\Big(\log\mathbb{E}_{(s,a,s')\sim d^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s')\Big)\Big] - 1\Big) \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s) + C] + 1 + \alpha,
\end{aligned}$$

we can conclude that

$$\hat{\mathcal{L}}(\nu^* + C) = \tilde{\mathcal{L}}(\nu^* + C).$$

From the Lemma F.1, we obtain

$$\min_\nu \hat{\mathcal{L}}(\nu) \leq \hat{\mathcal{L}}(\nu^* + C) = \tilde{\mathcal{L}}(\nu^* + C) = \min_\nu \tilde{\mathcal{L}}(\nu).$$

We show that $\min_\nu \tilde{\mathcal{L}}(\nu) \leq \min_\nu \hat{\mathcal{L}}(\nu)$ and $\min_\nu \hat{\mathcal{L}}(\nu) \leq \min_\nu \tilde{\mathcal{L}}(\nu)$, so $\min_\nu \hat{\mathcal{L}}(\nu) = \min_\nu \tilde{\mathcal{L}}(\nu)$.

Finally, similar to the proof steps for Proposition 3.2, we can easily show that $\hat{\mathcal{L}}_{\text{FD}}(\nu)$ is convex w.r.t. ν (Remark that log-sum-exp is a convex function). □

Proposition 3.5. *Let $\widehat{V}$ and $\widetilde{V}$ be the sets* $\arg\min_{\widehat{\nu}} \widehat{\mathcal{L}}(\widehat{\nu})$ *and* $\arg\min_{\widetilde{\nu}} \widetilde{\mathcal{L}}(\widetilde{\nu})$*, respectively. Then,* $\widetilde{V} = \{\widehat{\nu}^* + C \mid \widehat{\nu}^* \in \widehat{V}, C \in \mathbb{R}\}$ *holds.*

Proof. We will prove this proposition by showing $\tilde{V} \subseteq \{\nu^* + C' | \nu^* \in \hat{V}, C \in \mathbb{R}\}$ and $\tilde{V} \supseteq \{\nu^* + C' | \nu^* \in \hat{V}, C \in \mathbb{R}\}$.
($\subseteq$) For given $\nu^* \in \arg\min_\nu \tilde{\mathcal{L}}(\nu)$, we define a constant C as follows:

$$C = \tfrac{1+\alpha}{1-\gamma} \log \mathbb{E}_{(s,a,s')\sim d^I}\left[\exp\left(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s') - 1\right)\right],$$

which implies

$$\exp\left(\tfrac{(1-\gamma)C}{1+\alpha}\right) = \mathbb{E}_{(s,a,s')\sim d^I}\left[\exp\left(\tfrac{1}{1+\alpha}\hat{A}_{\nu^*}(s,a,s') - 1\right)\right].$$

Then, by the proof steps of Proposition 3.4, we obtain

$$\hat{\mathcal{L}}(\nu^* + C) = \tilde{\mathcal{L}}(\nu^* + C) = \min_\nu \tilde{\mathcal{L}}(\nu) = \min_\nu \hat{\mathcal{L}}(\nu).$$

It means $(\nu^* + C) \in \arg\min_\nu \hat{\mathcal{L}}(\nu)$ and thus,

$$\nu^* \in \{\hat{\nu}^* + C | \hat{\nu}^* \in \arg\min_\nu \hat{\mathcal{L}}(\nu), C \in \mathbb{R}\},$$

i.e.,

$$\arg\min_\nu \tilde{\mathcal{L}}(\nu) \subseteq \{\hat{\nu}^* + C | \hat{\nu}^* \in \arg\min_\nu \hat{\mathcal{L}}(\nu), C \in \mathbb{R}\}$$

($\supseteq$) Let $\hat{\nu}^* \in \arg\min_\nu \hat{\mathcal{L}}(\nu)$. Then,

$$\begin{aligned}
\frac{d^*(s,a)}{d^I(s,a)} &= w^*_{\mu^*,\nu^*}(s,a) = \exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'\sim T(\cdot|s,a)}[\gamma\nu(s') - \mu(s,s')] - \nu(s)\right) - 1\right),\\
\frac{d^*(s,s')}{d^I(s,s')} &= \bar{w}^*_{\mu^*,\nu^*}(s,s') = \exp(r(s,s') + \mu^*(s,s') - 1).
\end{aligned}$$

Let $C \in \mathbb{R}$. Then, we derive the following equation:

$$\begin{aligned}
&\tilde{\mathcal{L}}(\hat{\nu}^* + C) = \tilde{\mathcal{L}}(\hat{\nu}^*)\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s)] + (1+\alpha)\log \mathbb{E}_{(s,a,s')\sim d^I}\left[\exp\left(\tfrac{1}{1+\alpha}\hat{A}_{\hat{\nu}^*}(s,a,s')\right)\right]\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s)] + (1+\alpha)\log \mathbb{E}_{(s,a,s')\sim d^I}\left[\exp\left(\tfrac{1}{1+\alpha}\hat{A}_{\hat{\nu}^*}(s,a,s') - 1 + 1\right)\right]\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s)] + 1 + \alpha.
\end{aligned}$$

Here, we apply Lemma F.1 to derive the first equality. Because

$$\begin{aligned}
\min_\nu \tilde{\mathcal{L}}(\nu) &= \min_\nu \hat{\mathcal{L}}(\nu)\\
&= \hat{\mathcal{L}}(\hat{\nu}^*)\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s)] + (1+\alpha)\mathbb{E}_{(s,a,s')\sim d^I}\left[\exp\left(\tfrac{1}{1+\alpha}\hat{A}_{\hat{\nu}^*}(s,a,s') - 1\right)\right]\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu^*(s)] + 1 + \alpha,
\end{aligned}$$

we can conclude that $\mathcal{L}_{\text{FD}}(\hat{\nu}^* + C) = \min_\nu \mathcal{L}_{\text{FD}}(\nu)$, i.e., $(\hat{\nu}^* + C) \in \arg\min_\nu \tilde{\mathcal{L}}(\nu)$. Consequently,

$$\arg\min_\nu \tilde{\mathcal{L}}(\nu) \supseteq \{\hat{\nu}^* + C | \hat{\nu}^* \in \arg\min_\nu \hat{\mathcal{L}}(\nu), C \in \mathbb{R}\}.$$

□

F.3 Fenchel dual formulation

Let

$$\delta_C(x) := \begin{cases} 0 & x \in C \\ \infty & \text{otherwise} \end{cases}.$$

Then we can provide following proposition:

Proposition F.2. *We can rewrite the optimization problem (6-8) as*

$$\max_{d\geq 0} -\delta_{(1-\gamma)p_0}(-(\gamma\mathcal{T}-\mathcal{B})_*d) - D_{\mathrm{KL}}\big((\bar{\mathcal{T}}_*d)\|\bar{d^E}\big) - \alpha D_{\mathrm{KL}}(d\|d^I). \tag{50}$$

Then, the dual problem of (50) *is given by*

$$\min_{\mu,\nu} \mathcal{L}_{FD}(\mu,\nu) \tag{51}$$
$$:= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \log \mathbb{E}_{\bar{d}^I}\left[\exp\left(r(s,s') + \mu(s,s')\right)\right] + \alpha \log \mathbb{E}_{d^I}\left[\exp\left(\tfrac{1}{\alpha}e_{\mu,\nu}(s,a)\right)\right].$$

Proof. We first define following three functions

$$\begin{aligned} f(\cdot) &:= \delta_{\{(1-\gamma)p_0\}}(\cdot), \\ g(\cdot;r) &:= \langle\cdot, -r\rangle + D_{\mathrm{KL}}(\cdot\|\bar{\mathcal{T}}_*d^I), \\ h(\cdot;\mu) &:= \langle\cdot, \bar{\mathcal{T}}\mu\rangle + \alpha D_{\mathrm{KL}}(\cdot\|d^I), \end{aligned}$$

and conjugate functions,

$$\begin{aligned} f_*(\cdot) &:= (1-\gamma)\mathbb{E}_{s\sim p_0}[\cdot], \\ g_*(\cdot;r) &:= \log \mathbb{E}_{(s,s')\sim\bar{\mathcal{T}}_*d^I}[\exp(\cdot + r(s,s'))], \\ h_*(\cdot;\mu) &:= \alpha\log\mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\frac{\cdot - (\bar{\mathcal{T}}\mu)(s,a)}{\alpha}\right)\right]. \end{aligned}$$

Then, the dual formulation of the primal (50) can be derived as follows:

$$\begin{aligned} &\max_{d\geq 0} -\delta_{(1-\gamma)p_0}\big(-(\gamma\mathcal{T}-\mathcal{B})_*d\big) - \mathbb{E}_{\substack{(s,a)\sim d,\\ s'\sim T(\cdot|s,a)}}\left[\log\frac{(\bar{\mathcal{T}}_*d)(s,s')}{d^E(s,s')}\right] - \alpha D_{\mathrm{KL}}(d\|d^I) \\ &= \max_{d\geq 0} -f\big(-(\gamma\mathcal{T}-\mathcal{B})_*d\big) + \mathbb{E}_{\substack{(s,a)\sim d,\\ s'\sim T(\cdot|s,a)}}\left[-\log\frac{(\bar{\mathcal{T}}_*d)(s,s')}{(\bar{\mathcal{T}}_*d^I)(s,s')} + \underbrace{\log\frac{d^E(s,s')}{(\bar{\mathcal{T}}_*d^I)(s,s')}}_{:=r(s,s')}\right] - \alpha D_{\mathrm{KL}}(d\|d^I) \\ &= \max_{d\geq 0} -f\big(-(\gamma\mathcal{T}-\mathcal{B})_*d\big) - g(\bar{\mathcal{T}}_*d;r) - \alpha D_{\mathrm{KL}}(d\|d^I) \\ &= \max_{d\geq 0} -f\big(-(\gamma\mathcal{T}-\mathcal{B})_*d\big) - \left\{\max_{\mu} -g_*(\mu;r) + \langle\bar{\mathcal{T}}_*d,\mu\rangle\right\} - \alpha D_{\mathrm{KL}}(d\|d^I) \\ &= \max_{d\geq 0}\min_{\mu} -f\big(-(\gamma\mathcal{T}-\mathcal{B})_*d\big) + g_*(\mu;r) - \langle\bar{\mathcal{T}}_*d,\mu\rangle - \alpha D_{\mathrm{KL}}(d\|d^I) \\ &= \max_{d\geq 0}\min_{\mu} -f\big(-(\gamma\mathcal{T}-\mathcal{B})_*d\big) + g_*(\mu;r) - \langle d,\bar{\mathcal{T}}\mu\rangle - \alpha D_{\mathrm{KL}}(d\|d^I) \\ &= \max_{d\geq 0}\min_{\mu} -f\big(-(\gamma\mathcal{T}-\mathcal{B})_*d\big) + g_*(\mu;r) - h(d;\mu) \\ &= \max_{d\geq 0}\min_{\mu} -\left\{\max_{\nu}\langle-(\gamma\mathcal{T}-\mathcal{B})_*d,\nu\rangle - f_*(\nu)\right\} + g_*(\mu;r) - h(d;\mu) \\ &= \max_{d\geq 0}\min_{\nu,\mu}\langle(\gamma\mathcal{T}-\mathcal{B})_*d,\nu\rangle + f_*(\nu) + g_*(\mu;r) - h(d;\mu) \\ &= \max_{d\geq 0}\min_{\nu,\mu}\langle d,(\gamma\mathcal{T}-\mathcal{B})\nu\rangle + f_*(\nu) + g_*(\mu;r) - h(d;\mu) := \mathcal{L}_{\mathrm{FD}}(d;\mu,\nu) \end{aligned}$$

Here, we can reorder the maximin to minimax and therefore, we can derive the

$$\begin{aligned} &\min_{\nu,\mu}\max_{d\geq 0}\mathcal{L}_{\mathrm{FD}}(d;\mu,\nu) \\ &= \min_{\nu,\mu}\max_{d\geq 0}\langle d,(\gamma\mathcal{T}-\mathcal{B})\nu\rangle + f_*(\nu) + g_*(\mu;r) - h(d;\mu) \\ &= \min_{\nu,\mu}\left\{\max_{d\geq 0}\langle d,(\gamma\mathcal{T}-\mathcal{B})\nu\rangle - h(d;\mu)\right\} + f_*(\nu) + g_*(\mu;r) \\ &= \min_{\nu,\mu} h_*((\gamma\mathcal{T}-\mathcal{B})\nu) + f_*(\nu) + g_*(\mu;r). \end{aligned}$$

We can rewrite the last term as

$$\min_{\nu,\mu}(1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \log \mathbb{E}_{\substack{(s,a)\sim d^I,\\ s'\sim T(\cdot|s,a)}}[\exp(\mu(s,s') + r(s,s'))] + \alpha \log \mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha} e_{\mu,\nu}(s,a)\right)\right].$$

□

Lemma F.3. *For given function $\mu : S \times S \to \mathbb{R}$ and $\nu : S \to \mathbb{R}$,*

$$\mathcal{L}_{FD}(\mu,\nu) = \mathcal{L}_{FD}(\mu + C, \nu + C') \quad \forall C, C' \in \mathbb{R}.$$

Proof.

$$\begin{aligned}
&\mathcal{L}_{\text{FD}}(\mu + C, \nu + C')\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s) + C'] + \log \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\mu(s,s') + C + r(s,s'))]\\
&\quad + \alpha \log \mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'\sim T(\cdot|s,a)}[\gamma\nu(s') + \gamma C' - \mu(s,s') - C] - \nu(s) - C'\right)\right)\right]\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + (1-\gamma)C' + \log \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\mu(s,s') + r(s,s'))] + C\\
&\quad + \alpha \log \mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'\sim T(\cdot|s,a)}[\gamma\nu(s') - \mu(s,s')] - \nu(s)\right)\right)\right] + (\gamma C' - C - C')\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \log \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\mu(s,s') + r(s,s'))]\\
&\quad + \alpha \log \mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'\sim T(\cdot|s,a)}[\gamma\nu(s') - \mu(s,s')] - \nu(s)\right)\right)\right]\\
&= \mathcal{L}_{\text{FD}}(\mu,\nu).
\end{aligned}$$

□

Finally, we can show that the relation between $\arg\min_{\mu,\nu}\mathcal{L}$ and $\arg\min_{\mu,\nu}\mathcal{L}_{\text{FD}}$:

Proposition F.4. *Let V and V_{FD} be the set of optimal solutions (μ^*,ν^*) of* $\arg\min_{\mu,\nu}\mathcal{L}(w_{\nu,\mu}, \bar{w}_{\nu,\mu}, \mu, \nu)$ *and* $\arg\min_{\mu,\nu}\mathcal{L}_{\text{FD}}(\mu,\nu)$*, respectively. Then,*

$$V_{FD} = \{(\mu^* + C, \nu^* + C') | (\mu^*,\nu^*) \in V, C \in \mathbb{R}, C' \in \mathbb{R}\}$$

holds.

Proof. For brevity, we will denote $\mathcal{L}(w_{\nu,\mu}, \bar{w}_{\nu,\mu}, \mu, \nu)$ as $\mathcal{L}(\mu,\nu)$.

($\subseteq$) For given $(\hat{\mu}^*, \hat{\nu}^*) \in \arg\min_{\mu,\nu}\mathcal{L}_{\text{FD}}(\mu,\nu)$, we define two constants as follows:

$$\begin{aligned}
C &= -\log \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\hat{\mu}^*(s,s') + r(s,s') - 1)],\\
C' &= \frac{\alpha}{1-\gamma}\log \mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\frac{\mathbb{E}_{s'}[\gamma\hat{\nu}^*(s') - \hat{\mu}^*(s,s')] - \hat{\nu}^*(s')}{\alpha} - 1\right)\right] - \frac{C}{1-\gamma}.
\end{aligned}$$

Then, from the following two equations:

$$\begin{aligned}
&\mathcal{L}(\hat{\mu}^* + C, \hat{\nu}^* + C')\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s) + C'] + \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\hat{\mu}^*(s,s') + C + r(s,s') - 1)]\\
&\quad + \alpha\mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'}[\gamma\hat{\nu}^*(s') + \gamma C' - \hat{\mu}^*(s,s') - C] - \hat{\nu}^*(s') - C'\right) - 1\right)\right]\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s)] + (1-\gamma)C' + \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\hat{\mu}^*(s,s') + r(s,s') - 1)]\exp(C)\\
&\quad + \alpha\mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'}[\gamma\hat{\nu}^*(s') - \hat{\mu}^*(s,s')] - \hat{\nu}^*(s')\right) - 1\right)\right]\exp\left(\tfrac{1}{\alpha}\left(-(1-\gamma)C' - C\right)\right)\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s)] + (1-\gamma)C' + 1 + \alpha,
\end{aligned}$$

and

$$\begin{aligned}
&\mathcal{L}_{\text{FD}}(\hat{\mu}^* + C, \hat{\nu}^* + C')\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s) + C'] + \log \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\hat{\mu}^*(s,s') + C + r(s,s'))]\\
&\quad + \alpha \log \mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'}[\gamma\hat{\nu}^*(s') + \gamma C' - \hat{\mu}^*(s,s') - C] - \hat{\nu}^*(s') - C'\right)\right)\right]\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s)] + (1-\gamma)C' + \log \mathbb{E}_{(s,a,s')\sim d^I}[\exp(\hat{\mu}^*(s,s') + r(s,s'))] + C\\
&\quad + \alpha \log \mathbb{E}_{(s,a)\sim d^I}\left[\exp\left(\tfrac{1}{\alpha}\left(\mathbb{E}_{s'}[\gamma\hat{\nu}^*(s') - \hat{\mu}^*(s,s')] - \hat{\nu}^*(s')\right)\right)\right] + (\gamma C' - C - C')\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\hat{\nu}^*(s) + C'] + 1 + \alpha,
\end{aligned}$$

we can conclude that

$$\mathcal{L}(\hat{\mu}^* + C, \hat{\nu}^* + C') = \mathcal{L}_{\text{FD}}(\hat{\mu}^* + C, \hat{\nu}^* + C') = \mathcal{L}_{\text{FD}}(\hat{\mu}^*, \hat{\nu}^*) = \min_{\mu,\nu} \mathcal{L}_{\text{FD}}(\mu, \nu) = \min_{\mu,\nu} \mathcal{L}(\mu, \nu).$$

It means $(\hat{\mu}^* + C, \hat{\nu}^* + C') \in \arg\min_{\mu,\nu} \mathcal{L}(\mu, \nu)$ and thus,

$$(\hat{\mu}^*, \hat{\nu}^*) \in \{\mu + C, \nu + C' | (\mu, \nu) \in \arg\min_{\mu,\nu} \mathcal{L}(\mu, \nu), C \in \mathbb{R}, C' \in \mathbb{R}\},$$

i.e.,

$$\arg\min_{\mu,\nu} \mathcal{L}_{\text{FD}}(\mu, \nu) \subseteq \{\mu + C, \nu + C' | (\mu, \nu) \in \arg\min_{\mu,\nu} \mathcal{L}(\mu, \nu), C \in \mathbb{R}, C' \in \mathbb{R}\}$$

$(\supseteq)$ Let $(\mu^*, \nu^*) \in \arg\min_{\mu,\nu} \mathcal{L}(\mu, \nu)$. Then,

$$\begin{aligned}
\frac{d^*(s,a)}{d^I(s,a)} &= w^*_{\mu^*,\nu^*}(s,a) = \exp\Big(\tfrac{1}{\alpha}\big(\mathbb{E}_{s' \sim T(\cdot|s,a)}[\gamma\nu(s') - \mu(s,s')] - \nu(s)\big) - 1\Big), \\
\frac{d^*(s,s')}{d^I(s,s')} &= \bar{w}^*_{\mu^*,\nu^*}(s,s') = \exp(r(s,s') + \mu^*(s,s') - 1).
\end{aligned}$$

Let $C \in \mathbb{R}$ and $C' \in \mathbb{R}$. Then, we derive the following equation:

$$\begin{aligned}
&\mathcal{L}_{\text{FD}}(\mu^* + C, \nu^* + C') \\
&= \mathcal{L}_{\text{FD}}(\mu^*, \nu^*) \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + \log \mathbb{E}_{(s,a,s') \sim d^I}[\exp(\mu^*(s,s') + r(s,s'))] \\
&\quad + \alpha \log \mathbb{E}_{(s,a) \sim d^I}\Big[\exp\Big(\tfrac{1}{\alpha}\big(\mathbb{E}_{s' \sim T(\cdot|s,a)}[\gamma\nu^*(s') - \mu^*(s,s')] - \nu^*(s)\big)\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + \log \mathbb{E}_{(s,a,s') \sim d^I}[\bar{w}^*_{\mu^*,\nu^*}(s,s')\exp(1)] \\
&\quad + \alpha \log \mathbb{E}_{(s,a) \sim d^I}[w(s,a)\exp(1)] \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + \log \mathbb{E}_{(s,a,s') \sim d^*}[\exp(1)] + \alpha \log \mathbb{E}_{(s,a) \sim d^*}[\exp(1)] \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + 1 + \alpha \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + 1 + \alpha.
\end{aligned}$$

Here, we apply Lemma F.1 to derive the first equality. Because

$$\begin{aligned}
&\min_{\mu,\nu} \mathcal{L}_{\text{FD}}(\mu, \nu) \\
&= \min_{\mu,\nu} \mathcal{L}(\mu, \nu) \\
&= \mathcal{L}(\mu^*, \nu^*) \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + \mathbb{E}_{(s,a,s') \sim d^I}[\exp(\mu^*(s,s') + r(s,s') - 1)] \\
&\quad + \alpha \mathbb{E}_{(s,a) \sim d^I}\Big[\exp\Big(\tfrac{1}{\alpha}\big(\mathbb{E}_{s' \sim T(\cdot|s,a)}[\gamma\nu^*(s') - \mu^*(s,s')] - \nu^*(s)\big) - 1\Big)\Big] \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + \mathbb{E}_{(s,a,s') \sim d^I}[\bar{w}^*_{\mu^*,\nu^*}(s,s')] + \alpha \mathbb{E}_{(s,a) \sim d^I}[w^*_{\mu^*,\nu^*}(s,a)] \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + \mathbb{E}_{(s,a,s') \sim d^*}[1] + \alpha \mathbb{E}_{(s,a) \sim d^*}[1] \\
&= (1-\gamma)\mathbb{E}_{s \sim p_0}[\nu^*(s)] + 1 + \alpha,
\end{aligned}$$

we can conclude that $\mathcal{L}_{\text{FD}}(\mu^* + C, \nu^* + C') = \min_{\mu,\nu} \mathcal{L}_{\text{FD}}(\mu, \nu)$, i.e., $(\mu^* + C, \nu^* + C') \in \arg\min_{\mu,\nu} \mathcal{L}_{\text{FD}}(\mu, \nu)$. It means,

$$\arg\min_{\mu,\nu} \mathcal{L}_{\text{FD}}(\mu, \nu) \supseteq \{\mu + C, \nu + C' | (\mu, \nu) \in \arg\min_{\mu,\nu} \mathcal{L}(\mu, \nu), C \in \mathbb{R}, C' \in \mathbb{R}\}$$

□

Remark that the proposed objective (51) is stable and the aforementioned proposition allows us to use self-normalized weighted importance sampling to extract policy. However, as we discussed in Section 3.5, using μ is main bottleneck for the overall optimization and optimizing $\tilde{\mathcal{L}}(\nu)$ shows better performance in practice.

G Pseudocode of LobsDICE

G.1 Tabular LobsDICE

For tabular MDPs, we first construct a maximum-likelihood estimation (MLE) MDP $\hat{M} = \langle S, A, \hat{T}, \hat{p}_0, \gamma \rangle$ using an offline dataset D^I. Then, tabular LobsDICE solves (16) on the MLE MDP $\hat{M}$. In tabular case, note that $\widehat{d^I}(s,a)$, $\widehat{d^I}(s,s')$, and $\widehat{d^E}(s,s')$ are explicitly accessible by normalized counts of (s,a) and (s,s') in the datasets. In addition, $\widehat{r}(s,s') = \frac{\widehat{d^E}(s,s')}{\widehat{d^I}(s,s')}$ is the log ratio between two empirical distributions. Also, $\nu \in \mathbb{R}^{|S|}$ is represented as a $|S|$-dimensional vector, and $\mu \in \mathbb{R}^{|S| \times |S|}$ is represented as a $|S|^2$-dimensional vector, and we solve the following convex minimization problem for (μ, ν):

$$\min_{\mu,\nu} J(\mu,\nu) = (1-\gamma) \sum_{s} \hat{p}_0(s)\nu(s) + \sum_{s,s'} \widehat{d^I}(s,s') \Big[\exp\big(\widehat{r}(s,s') + \mu(s,s') - 1 \big) \Big]$$
$$+ \alpha \sum_{s,a} \widehat{d^I}(s,a) \Big[\exp\Big(\tfrac{1}{\alpha} \sum_{s'} \hat{T}(s'|s,a) \big(-\mu(s,s') + \gamma\nu(s') - \nu(s) \big) - 1 \Big) \Big] \tag{52}$$

This process is summarized in Algorithm 1.

Algorithm 1 Tabular LobsDICE

```
Input: state-only demonstrations by experts $D^E = \{(s,s')_i\}_{i=1}^{N_E}$, state-action demonstrations by
    some imperfect agents $D^I = \{(s,a,s')_i\}_{i=1}^{N_I}$, a learning rate $\eta$.
Construct an MLE MDP $\hat{M} = \langle S, A, \hat{T}, \hat{p}_0, \gamma \rangle$ via normalized visitation counts of $D^I$.
Construct $\widehat{d^I}(s,a)$, $\widehat{d^I}(s,s')$, and $\widehat{d^E}(s,s')$ via normalized visitation counts of $D^I$ and $D^E$.
$\widehat{r}(s,s') \leftarrow \log \frac{\widehat{d^E}(s,s')}{\widehat{d^I}(s,s')}$ for all $s, s'$.
Randomly initialize $\nu$ and $\mu$.
while $(\nu, \mu)$ is not converged do
    $(\mu,\nu) = (\mu,\nu) - \eta \nabla_{\mu,\nu} J(\mu,\nu)$     (Eq. (52))
end while
$w^*(s,a) = \exp\Big( \frac{1}{\alpha} \big( \sum_{s'} \hat{T}(s'|s,a)(-\mu(s,s') + \gamma\nu(s') - \nu(s)) \big) - 1 \Big)$ for all $s, a$.   (Eq. (15))
$\pi^*(a|s) \leftarrow \frac{\widehat{d^I}(s,a) w^*(s,a)}{\sum_{a'} \widehat{d^I}(s,a') w^*(s,a')}$ for all $s, a$.
Output: The imitation policy $\pi^*$.
```

G.2 Practical LobsDICE (with function approximation)

To deal with continuous or large MDPs, we represent our optimization variable ν and discriminator c as neural networks, parameterized by θ and ϕ respectively: $\nu_\theta : S \to \mathbb{R}$ is an MLP that takes a state as an input and outputs a scalar value, and $c_\phi : S \times S \to [0,1]$ is defined similarly. For the policy π_ψ, we use a tanh-squashed Gaussian policy, where the parameters of a Gaussian distribution (i.e. mean and covariance) are output by the neural network.

The parameters of c_ϕ are trained by:

$$\max_{\phi} J_c(\phi) := \mathbb{E}_{\substack{\text{batch}(\bar{d}^E) \sim \bar{d}^E \\ \text{batch}(\bar{d}^I) \sim \bar{d}^I}} \Big[\hat{\mathbf{E}}_{(s,s') \sim \text{batch}(\bar{d}^E)} [\log c_\phi(s,s')] + \hat{\mathbf{E}}_{(s,s') \sim \text{batch}(\bar{d}^I)} [\log(1 - c_\phi(s,s'))] \Big], \tag{53}$$

which is analogous to (13). The parameters of ν_θ are trained by:

$$\min_{\theta} J_\nu(\theta) := \mathbb{E}_{\substack{\text{batch}(d^I) \sim d^I \\ \text{batch}(p_0) \sim \hat{p}_0}} \Big[(1-\gamma) \hat{\mathbf{E}}_{s \sim \text{batch}(p_0)} [\nu_\theta(s)] \tag{54}$$
$$+ (1+\alpha) \log \hat{\mathbf{E}}_{(s,a,s') \sim \text{batch}(d^I)} [\exp(\tfrac{1}{1+\alpha} \hat{A}_\theta(s,a,s'))] \Big],$$

where $\hat{A}_\theta(s,a,s') = r_\phi(s,s') + \gamma\nu_\theta(s') - \nu_\theta(s)$ and $r_\phi(s,s') = -\log\Big(\frac{1}{c_\phi(s,s')} - 1\Big)$, and this is analogous to (23). Due to the logarithm outside the expectation in the second term, the mini-batch approximation would introduce additional bias in gradient estimation, but we found that using the biased gradient estimate worked well in practice with moderately large batch size (e.g. 256). Finally, the parameters of π_ψ are trained by:

$$\max_{\psi} J_\pi(\psi) := \mathbb{E}_{\text{batch}(d^I)\sim d^I}\left[\frac{\hat{\mathbf{E}}_{(s,a,s')\sim\text{batch}(d^I)}\Big[\exp\big(\tfrac{1}{1+\alpha}\hat{A}_\theta(s,a,s')\big)\log\pi_\psi(a|s)\Big]}{\hat{\mathbf{E}}_{(s,a,s')\sim\text{batch}(d^I)}\Big[\exp\big(\tfrac{1}{1+\alpha}\hat{A}_\theta(s,a,s')\big)\Big]}\right], \tag{55}$$

which is analogous to (24). Here, the self-normalized importance sampling may introduce an additional bias with mini-batch approximations, but it is well known that the self-normalized importance sampling provides a consistent estimator [31]. Also, it worked well in practice with moderately large batch size (e.g. 256) in our experiments. We optimize the parameters (ϕ, θ, ψ) jointly in practice. We are not using a target network at all. The overall training process is summarized in Algorithm 2.

Algorithm 2 LobsDICE (with function approximation)

```
Input: state-only demonstrations by experts $D^E = \{(s, s')_i\}_{i=1}^{N_E}$, state-action demonstrations by
    some imperfect agents $D^I = \{(s, a, s')_i\}_{i=1}^{N_I}$, a learning rate $\eta$.
Initialize parameter vectors $\phi, \theta, \psi$.
for each gradient step do
    Sample mini-batches from $D^E$ and $D^I$.
    Compute gradients and perform SGD update:
        $\phi \leftarrow \phi + \eta\nabla_\phi J_c(\phi)$            (Eq. (53))
        $\theta \leftarrow \theta - \eta\nabla_\theta J_\nu(\theta)$            (Eq. (54))
        $\psi \leftarrow \psi + \eta\nabla_\psi J_\pi(\psi)$            (Eq. (55))
end for
Output: The imitation policy $\pi_\psi$.
```

H Additional Experiments

H.1 Subsampled expert demonstrations

Existing works on imitation learning [11] conduct experiments where the expert demonstrations are subsampled. The following figure presents the result when given the subsampled expert demonstrations. LobsDICE still significantly outperforms the baseline algorithms even with the subsampled expert demonstrations[3].

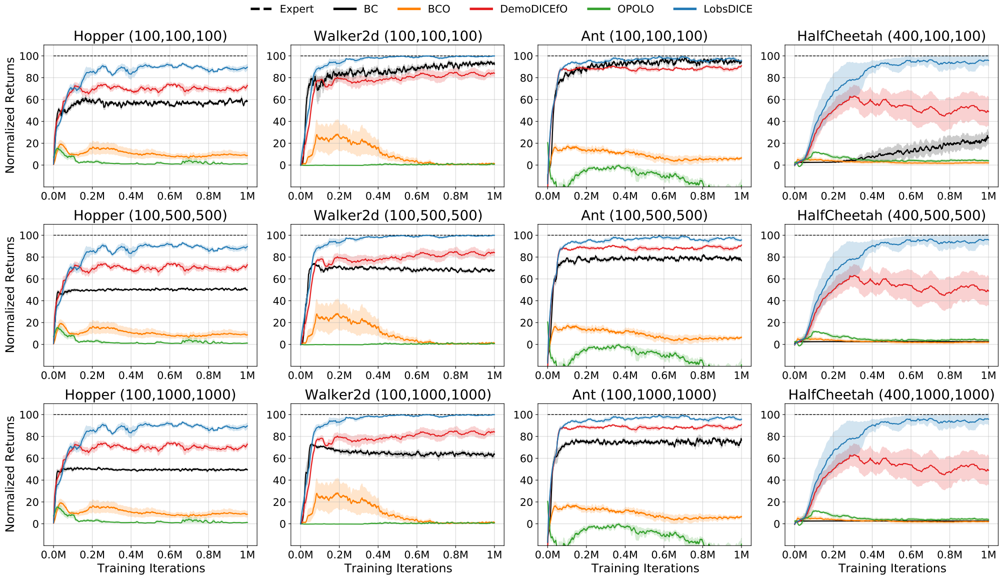

Figure 5: Performance of LobsDICE and baseline algorithms on various MuJoCo control tasks. We build state-only expert demonstrations using 50 subsampled trajectories from `expert-v2` (subsampling rate is 20).

H.2 Error of inverse dynamics model in MuJoCo domains

In principle, IDM trained with arbitrary demonstrations should be able to predict the expert's missing actions accurately in deterministic MDPs. However, in Section 5.2, we observed that the empirical performance of BCO and DemoDICEfO tends to degrade as the number of imperfect demonstrations increases (see Figure 2). This is due to the use of function approximation for IDM: given that the expressive power of a function approximator is limited, the more data unrelated to the expert demonstrations (i.e. `medium-v2` and `random-v2`) is used for training, the more the prediction accuracy for the expert data would be adversely affected. Figure 6 shows that the mean squared error of IDM for expert demonstrations D^E increases as the number of non-expert demonstrations increases.

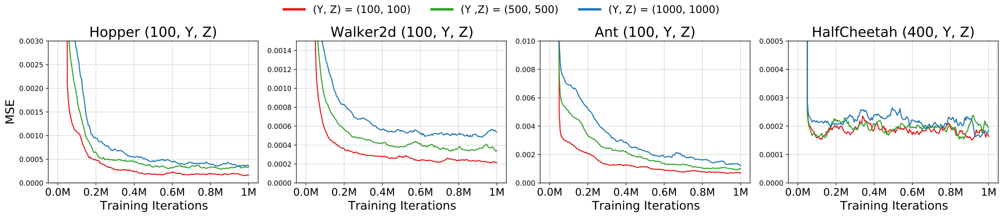

Figure 6: Mean squared error of IDM for expert demonstrations D^E. For each task (Y, Z), we construct imperfect demonstrations using 100 (400 for HalfCheetah), Y, and Z trajectories from `expert-v2`, `medium-v2`, and `random-v2`, respectively.

[3] Recently, Li et al. reported that ValueDICE [17] (without target network) significantly underperforms DAC [16] (with target network) in the subsampled expert trajectories setting due to divergence issue and pointed to the absence of the target network in ValueDICE as a reason for performance degradation. However, we didn't observe any divergence issue due to the absence of the target network, and LobsDICE still performs well even when the trajectories are subsampled.

H.3 Fewer expert demonstrations

In this section, we provide additional experiments with fewer expert demonstrations. Specifically, we tested LobsDICE when the number of state-only expert demonstrations is 5, 3, and 1, respectively. The Figure 7 shows that LobsDICE (#exp=3) and LobsDICE (#exp=1) still perform well.

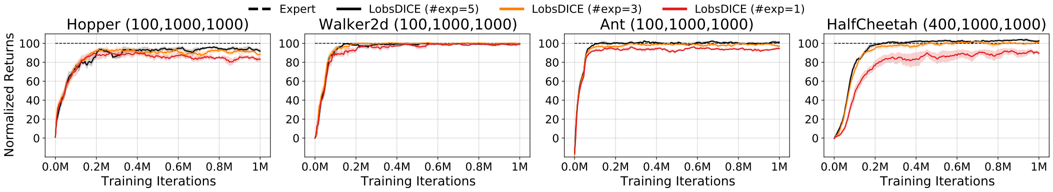

Figure 7: Performance of LobsDICE on various MuJoCo control tasks. We build state-only expert demonstrations using 1, 3, and 5 trajectories from `expert-v2`. For each task (X, Y, Z) we construct imperfect demonstrations using X, Y, and Z trajectories from `expert-v2`, `medium-v2`, and `random-v2`, respectively. We plot the mean and the standard errors (shaded area) of the normalized scores over five random seeds.

I Experimental Details

I.1 Experimental details for random MDPs

Random MDP generation For MDP $M = \langle S, A, T, R, \gamma, p_0 \rangle$, we first set $|S| = 20$, $|A| = 4$, $\gamma = 0.95$, $p_0(s) = 1$ for a fixed $s = s_0$. For each (s, a), we samples four different state $\{s'_1, s'_2, s'_3, s'_4\}$. Then, we set the transition probability $(T(s'_1|s, a), T(s'_2|s, a), T(s'_3|s, a), T(s'_4|s, a)) = (1 - \beta)X + \beta Y$, where $X \sim$ Categorical$(0.25, 0.25, 0.25, 0.25)$ and $Y \sim$ Dir$(1, 1, 1, 1)$. $\beta \in [0, 1]$ is the hyperparameter to control transition stochasticity: when $\beta = 0$, the transition probability becomes one-hot vector, i.e., deterministic MDP. In contrast, when $\beta = 1$, the transition of MDP becomes stochastic. Finally, the reward of 1 is only given to a state that minimizes the optimal value at initial state s_0; other states have zero rewards.

Offline dataset generation For each random MDP M, we generate state-only expert demonstrations by executing a (softmax) expert policy and state-action imperfect demonstrations by a uniform random policy. We perform experiments for a varying number of expert demonstrations $N_E \in \{10, 100, 1000, 1000\}$ and imperfect demonstrations $N_I \in \{1, 3, 10, 30, 100, 300, 1000, 3000, 10000\}$.

Hyperparameters We compare our tabular LobsDICE with BC, BCO, DemoDICEfO, and OPOLO. For KL-regularization hyperparameters of DemoDICEfO and LobsDICE, we use $\alpha = 0.1$.

I.2 Experimental details for MuJoCo control tasks

Hyperparameters	BC	BCO	DemoDICEfO	LobsDICE
γ (discount factor)	0.99	0.99	0.99	0.99
α (regularization coefficient)	-	-	0.1	0.1
learning rate (actor)	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
network size (actor)	[256, 256]	[256, 256]	[256, 256]	[256, 256]
learning rate (critic)	-	-	3×10^{-4}	3×10^{-4}
network size (critic)	-	-	[256,256]	[256, 256]
learning rate (discriminator)	-	-	3×10^{-4}	3×10^{-4}
network size (discriminator)	-	-	[256,256]	[256, 256]
learning rate (inverse dynamics)	-	3×10^{-4}	3×10^{-4}	-
network size (inverse dynamics)	-	[256,256]	[256, 256]	-
gradient L2-norm coefficient (critic)	-	-	1×10^{-4}	1×10^{-4}
gradient penalty coefficient (discriminator)	-	-	0.1	0.1
batch size	512	512	512	512
# of expert trajectories	5	5	5	5
# of training iterations	1,000,000	1,000,000	1,000,000	1,000,000

Table 1: Configurations of hyperparameters used in our experimental results.

For fair comparison, we use the same learning rate to train actors of BC, BCO, DemoDICEfO, and LobsDICE. We implement our network architectures for BC, BCO, DemoDICEfO, and LobsDICE based on the implementation of OptiDICE[4]. For OPOLO, we use its official implmentation[5]. We have tried hyperparameter tuning for OPOLO but never obtained a successful learning curve, showing numerical instability due to using out-of-distribution action values. Vanilla *online* off-policy algorithms commonly fail in the *offline* learning setting. Therefore, we report the results of official implementation without any modification to network architectures or hyperparameters. For stable discriminator learning, we use gradient penalty regularization on the $r(s, a)$ and $r(s, s')$ functions, which was proposed in [10] to enforce 1-Lipschitz constraint. To stabilize critic training, we add gradient L2-norm to the critic loss for the regularization. In addition, we use every state s in D^I

[4] `https://github.com/secury/optidice`

[5] `https://github.com/illidanlab/opolo-code`

to define D_0, i.e., $D_0 = \{s|(s, a, s') \in D^I\}$, following ValueDICE [17]. Detailed hyperparameter configurations used for our main experiments are summarized in Table 1.

Evaluation metric For each environment, the normalized score is measured by $100 \times \frac{\texttt{score}-\texttt{random score}}{\texttt{expert score}-\texttt{random score}}$, where the `expert score` and `random score` are average returns of trajectories in `expert-v2` and `random-v2`, respectively.

J Generalization to $\gamma = 1$

In this section, we generalize LobsDICE for discounted MDPs ($\gamma < 1$) to undiscounted MDPs ($\gamma = 1$). Suppose that the stationary distributions d and $\bar{d}$ satisfy the Bellman flow constraints (7) and marginalization constraints (8), respectively. When $\gamma = 1$, for any constant $c \geq 0$, cd and $c\bar{d}$ also satisfy the corresponding constraints.
It means that the original constrained optimization problem (6-8) for the stationary distributions d and $\bar{d}$ is an ill-posed problem. We tackle this ill-posedness by adding additional normalization constraint $\sum_{s,a} d(s,a) = 1$ to (6-8):

$$\begin{aligned}
\max_{d, \bar{d} \geq 0} \quad & -D_{\mathrm{KL}}(\bar{d}(s,s') \| \bar{d}^E(s,s')) - \alpha D_{\mathrm{KL}}(d(s,a) \| d^I(s,a)) \\
\text{s.t.} \quad & \sum_{a'} d(s',a') = (1-\gamma) p_0(s') + \gamma \sum_{s,a} d(s,a) T(s'|s,a) \quad \forall s', \\
& \sum_{a} d(s,a) T(s'|s,a) = \bar{d}(s,s') \quad \forall s, s', \\
& \sum_{s,a} d(s,a) = 1.
\end{aligned}$$

Using a derivation similar to that from (9) to (11), we obtain the following min-max optimization for $w, \bar{w}, \mu, \nu$, and λ:

$$\begin{aligned}
& \min_{\mu,\nu,\lambda} \max_{w, \bar{w} \geq 0} \mathcal{L}(w, \bar{w}, \mu, \nu, \lambda) \\
& := \mathcal{L}(w, \bar{w}, \mu, \nu) + \lambda (1 - \mathbb{E}_{(s,a) \sim d^I}[w(s,a)]) \\
& = (1-\gamma) \mathbb{E}_{s_0 \sim p_0}[\nu(s_0)] + \mathbb{E}_{(s,s') \sim \bar{d}^I}\big[\bar{w}(s,s')\big(r(s,s') + \mu(s,s') - \log \bar{w}(s,s')\big)\big] \\
& \qquad + \mathbb{E}_{(s,a) \sim d^I}\big[w(s,a)\big(\underbrace{e_{\mu,\nu}(s,a) - \lambda}_{:=e_{\mu,\nu,\lambda}(s,a)} - \alpha \log w(s,a)\big)\big] + \lambda, \qquad (56)
\end{aligned}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier for the normalization constraint $\sum_{s,a} d(s,a) = 1$. Similar to Proposition 3.1, we can derive a closed-form solution to the inner maximization of (56):

$$w_{\mu,\nu,\lambda}(s,a) = \exp\big(\tfrac{1}{\alpha} e_{\mu,\nu,\lambda}(s,a) - 1\big) \;\; \textit{and} \;\; \bar{w}_\mu(s,s') = \exp(r(s,s') + \mu(s,s') - 1).$$

Then, we can reduce the nested min-max optimization of (56) to a single minimization by plugging the closed-form solution $(w_{\mu,\nu,\lambda}, \bar{w}_\mu)$ into $\mathcal{L}(w, \bar{w}, \mu, \nu, \lambda)$:

$$\begin{aligned}
\min_{\mu,\nu,\lambda} & \mathcal{L}(w_{\mu,\nu,\lambda}, \bar{w}_\mu, \mu, \nu, \lambda) = (1-\gamma) \mathbb{E}_{s \sim p_0}[\nu(s)] \\
& + \mathbb{E}_{(s,s') \sim \bar{d}^I}\big[\exp\big(r(s,s') + \mu(s,s') - 1\big)\big] + \alpha \mathbb{E}_{(s,a) \sim d^I}\big[\exp\big(\tfrac{1}{\alpha} e_{\mu,\nu,\lambda}(s,a) - 1\big)\big] + \lambda.
\end{aligned}$$

In practice, we estimate $\mathcal{L}(w_{\mu,\nu,\lambda}, \bar{w}_\mu, \mu, \nu, \lambda))$ using samples from distribution d^I. To derive a sample-based objective, we use analogous derivation in Section 3.5. Sample-based objective can be represented as

$$\begin{aligned}
\min_{\mu,\nu,\lambda} & \widehat{\mathcal{L}}(\mu, \nu, \lambda) = (1-\gamma) \hat{\mathbf{E}}_{s \in D_0}[\nu(s)] \qquad (57) \\
& + \hat{\mathbf{E}}_{x \in D^I}\big[\exp\big(r(s,s') + \mu(s,s') - 1\big) + \alpha \exp\big(\tfrac{1}{\alpha} \hat{e}_{\mu,\nu,\lambda}(s,a,s') - 1\big)\big] + \lambda,
\end{aligned}$$

where $\hat{e}_{\mu,\nu,\lambda}(s,a,s') = -\mu(s,s') + \gamma\nu(s') - \nu(s) - \lambda$. Then, the closed-form solution to the minimization (57) with respect to μ is:

$$\mu_{\nu,\lambda}(s,s') = \frac{1}{1+\alpha}\big(-\alpha r(s,s') + \gamma\nu(s') - \nu(s) - \lambda\big) \qquad (58)$$

Using the above solution, we obtain the following minimization problem:

$$\min_{\widehat{\nu},\widehat{\lambda}} \widehat{\mathcal{L}}(\widehat{\nu},\widehat{\lambda}) = (1-\gamma)\hat{\mathbf{E}}_{s\in D_0}[\nu(s)] + \widehat{\lambda} + (1+\alpha)\hat{\mathbf{E}}_{x\in D^I}\Big[\exp\big(\tfrac{1}{1+\alpha}\widehat{A}_{\widehat{\nu},\widehat{\lambda}}(s,a,s') - 1\big)\Big], \quad (59)$$

where $\widehat{A}_{\nu,\lambda}(s,a,s') := r(s,s') + \gamma\nu(s') - \nu(s) - \lambda$. Finally, similar to Proposition 3.4, we can obtain the following objective with the same minimum as (59):

$$\begin{aligned}\min_{\nu,\lambda} \widetilde{\mathcal{L}}(\widetilde{\nu},\widetilde{\lambda}) &= (1-\gamma)\hat{\mathbf{E}}_{s\in D_0}[\widetilde{\nu}(s)] + \widetilde{\lambda} + (1+\alpha)\log\hat{\mathbf{E}}_{x\in D^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\widehat{A}_{\widetilde{\nu},\widetilde{\lambda}}(s,a,s')\Big)\Big] \\ &= (1-\gamma)\hat{\mathbf{E}}_{s\in D_0}[\widetilde{\nu}(s)] + (1+\alpha)\log\hat{\mathbf{E}}_{x\in D^I}\Big[\exp\Big(\tfrac{1}{1+\alpha}\widehat{A}_{\widetilde{\nu}}(s,a,s')\Big)\Big] = \min_{\nu}\widetilde{\mathcal{L}}(\widetilde{\nu}).\end{aligned}$$

Interestingly, the resulting objective is the same as the original objective $\widetilde{\mathcal{L}}(\widetilde{\nu})$ of (23). Based on this theoretical result, we compare LobsDICE for $\gamma = 1$, denoted by LobsDICE ($\gamma = 1$), with LobsDICE for $\gamma = 0.99$, denoted by LobsDICE ($\gamma = 0.99$). In Figure 8, LobsDICE ($\gamma = 1$) shows good performance, but LobsDICE ($\gamma = 0.99$) is more stable and achieves better performance in practice.

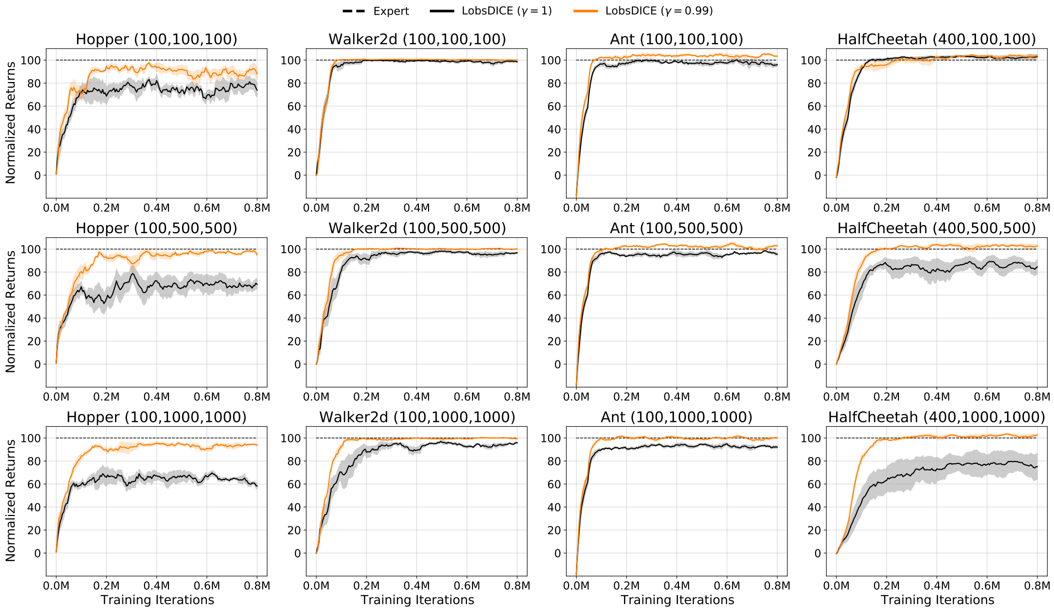

Figure 8: Performance of LobsDICE ($\gamma = 0.99$) and LobsDICE ($\gamma = 1$) on various MuJoCo control tasks. We build state-only expert demonstrations using 5 trajectories from `expert-v2`. For each task (X, Y, Z) we construct imperfect demonstrations using X, Y, and Z trajectories from `expert-v2`, `medium-v2`, and `random-v2`, respectively. We plot the mean and the standard errors (shaded area) of the normalized scores over five random seeds.

K Computation Resources

We used 10 servers equipped with the following specification:

- CPU: Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz.
- Memory: 32 GB.
- GPU: TITAN V.

L Limitation

Although we demonstrated that LobsDICE successfully recovered the expert's behavior in the experiments, it requires the assumption that support of D^I covers D^E. Relaxing this assumption remains as a future work.

M License

D4RL [6] is licensed under the Apache 2.0

N Social Impact

This work contributes to an algorithmic foundation for imitation learning from observation in an offline setting. Given that state-only expert demonstrations are much easier to be collected than the action-labeled expert demonstrations and that the imperfect demonstrations of arbitrary optimality is also easy to be collected, our method has a potential to be widely adopted in many real-world applications. On the other hand, this work could adversely affect employment by contributing to the automation of tasks having done by human experts (e.g. factory automation, autonomous driving).