Off-Policy Imitation Learning from Observations

Zhuangdi Zhu  
Michigan State University  
zhuzhuan@msu.edu

Kaixiang Lin  
Michigan State University  
linkaixi@msu.edu

Bo Dai  
Google Research  
bodai@google.com

Jiayu Zhou  
Michigan State University  
jiayuz@msu.edu

Abstract

Learning from Observations (LfO) is a practical reinforcement learning scenario from which many applications can benefit through the reuse of incomplete resources. Compared to conventional imitation learning (IL), LfO is more challenging because of the lack of expert action guidance. In both conventional IL and LfO, distribution matching is at the heart of their foundation. Traditional distribution matching approaches are sample-costly which depend on on-policy transitions for policy learning. Towards sample-efficiency, some off-policy solutions have been proposed, which, however, either lack comprehensive theoretical justifications or depend on the guidance of expert actions. In this work, we propose a sample-efficient LfO approach which enables off-policy optimization in a principled manner. To further accelerate the learning procedure, we regulate the policy update with an inverse action model, which assists distribution matching from the perspective of mode-covering. Extensive empirical results on challenging locomotion tasks indicate that our approach is comparable with state-of-the-art in terms of both sample-efficiency and asymptotic performance.

1 Introduction

Imitation Learning (IL) has been widely studied in the reinforcement learning (RL) domain to assist in learning complex tasks by leveraging the experience from expertise [1–5]. Unlike conventional RL that depends on environment reward feedbacks, IL can purely learn from expert guidance, and is therefore crucial for realizing robotic intelligence in practical applications, where demonstrations are usually easier to access than a delicate reward function [6, 7].

Classical IL, or more concretely, Learning from Demonstrations (LfD), assumes that both states and actions are available as expert demonstrations [8, 2, 3]. Although expert actions can benefit IL by providing elaborated guidance, requiring such information for IL may not always accord with the real-world. Actually, collecting demonstrated actions can sometimes be costly or impractical, whereas observations without actions are more accessible resources, such as camera or sensory logs. Consequently, Learning from Observations (LfO) has been proposed to address the scenario without expert actions [9, 11]. On one hand, LfO is more challenging compared with conventional IL, due to missing finer-grained guidance from actions. On the other hand, LfO is a more practical setting for IL, not only because it capitalizes previously unusable resources, but also because it reveals the potential to realize advanced artificial intelligence. In fact, learning without action guidance is an inherent ability for human being. For instance, a novice game player can improve his skill purely by watching video records of an expert, without knowing what actions have been taken [12].

Among popular LfD and LfO approaches, distribution matching has served as a principled solution [2, 8, 9, 10, 13], which works by interactively estimating and minimizing the discrepancy between two
stationary distributions: one generated by the expert, and the other generated by the learning agent. To correctly estimate the distribution discrepancy, traditional approaches require on-policy interactions with the environment whenever the agent policy gets updated. This inefficient sampling strategy impedes wide applications of IL to scenarios where accessing transitions are expensive [14, 15]. The same challenge is aggravated in LfO, as more explorations by the agent are needed to cope with the lack of action guidance.

Towards sample-efficiency, some off-policy IL solutions have been proposed to leverage transitions cached in a replay buffer. Mostly designed for LfD, these methods either lack theoretical guarantee by ignoring a potential distribution drift [4, 16, 17], or hinge on the knowledge of expert actions to enable off-policy distribution matching [3], which makes their approach inapplicable to LfO.

To address the aforementioned limitations, in this work, we propose a LfO approach that improves sample-efficiency in a principled manner. Specifically, we derive an upper-bound of the LfO objective which dispenses with the need of knowing expert actions and can be fully optimized with off-policy learning. To further accelerate the learning procedure, we combine our objective with a regularization term, which is validated to pursue distribution matching between the expert and the agent from a mode-covering perspective. Under a mild assumption of a deterministic environment, we show that the regularization can be enforced by learning an inverse action model. We call our approach OPOLO (Off Policy Learning from Observations). Extensive experiments on popular benchmarks show that OPOLO achieves state-of-the-art in terms of both asymptotic performance and sample-efficiency.

2 Background

We consider learning an agent in an environment of Markov Decision Process (MDP) [18], which can be defined as a tuple: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, p_0)$. Particularly, $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces; $P$ is the state transition probability, with $P(s'|s, a)$ indicating the probability of transitioning from $s$ to $s'$ upon action $a$; $r$ is the reward function, with $r(s, a)$ the immediate reward for taking action $a$ on state $s$; Without ambiguity, we consider an MDP with infinite horizons, with $0 < \gamma < 1$ as a discounted factor; $p_0$ is the initial state distribution. An agent follows its policy $\pi : \mathcal{S} \to \mathcal{A}$ to interact with this MDP with an objective of maximizing its expected return:

$$\max J_{RL}(\pi) := \mathbb{E}_{\pi} \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t),$$

in which $\mu^\pi(s, a)$ is the stationary state-action distribution induced by $\pi$, as defined in Table 1.

**Learning from demonstrations** (LfD) is a problem setting in which an agent is provided with a fixed dataset of expert demonstrations as guidance, without accessing the environment rewards. The demonstrations $\mathcal{R}_E$ contain sequences of both states and actions generated by an expert policy $\pi_E$:

$$\mathcal{R}_E = \{ (s_0, a_0), (s_1, a_1), \ldots | a_i \sim \pi_E(|s_i), s_{i+1} \sim P(|s_i, a_i) \}. \text{ Without ambiguity, we assume that the expert and agent are from the same MDP.}

Among LfD approaches, distribution matching has been a popular choice, which minimizes the discrepancy between two stationary state-action distributions: one is $\mu^E(s, a)$ induced by the expert, and the other is $\mu^\pi(s, a)$ induced by the agent. Without loss of generality, we consider KL-divergence as the discrepancy measure for distribution matching, although any $f$-divergences can serve as a legitimate choice [2, 19, 20]:

$$\min J_{LfD}(\pi) := D_{KL}[\mu^E(s, a)||\mu^\pi(s, a)].$$

**Learning from observations** (LfO) is a more challenging scenario where expert guidance $\mathcal{R}_E$ contains only states. Accordingly, applying distribution matching to solve LfO yields a different objective that involves state-transition distributions [10, 21, 9]:

$$\min J_{LfO}(\pi) := D_{KL}[\mu^E(s', a)||\mu^\pi(s, a)].$$

There exists a close connection between LfO and LfD objectives. In particular, the discrepancy between two objectives can be derived precisely as follows (see Sec 9.2 in the appendix) [10]:

$$D_{KL}[\mu^E(s', a)||\mu^\pi(s, a)] = D_{KL}[\mu^\pi(s, a)||\mu^E(s, a)] - D_{KL}[\mu^\pi(s, a)||\mu^\pi(s, a)].$$

**Remark 1.** In a non-injective MDP, the discrepancy of $D_{KL}[\mu^\pi(a, s')||\mu^E(a, s')]$ cannot be optimized without knowing expert actions. In a deterministic and injective MDP, it satisfies that $\forall \pi : \mathcal{S} \to \mathcal{A}$, $D_{KL}[\mu^\pi(a, s')||\mu^E(a, s')] = 0$. 

2
Definition

Work along this line usually adopts a GAN-style strategy \[25\] to perform distribution matching. Take \( R \) which is induced by a dataset we explicitly introduce an off-policy distribution into the LfO objective, from which we derive a

A common limitation of existing LfO and LfD approaches relies in their inefficient optimization.

The idea of re-using cached transitions to improve sample-efficiency has been adopted by many RL algorithms \[7, 26–28\]. In the same spirit, we start by introducing an off-policy learning \[10, 9\]. In pursuit of sample-efficiency, some off-policy solutions have been proposed. These methods, however, lack theoretical guarantee \[17 \[4\], or rely on the expert actions \[4 \[3\], which makes them inapplicable to LfO. We will provide more explanations in Sec 9.8 in the appendix.

To improve the sample-efficiency of LfO with a principled solution, in the next section we show how we explicitly introduce an off-policy distribution into the LfO objective, from which we derive a feasible upper-bound that enables off-policy optimization without the need of accessing expert actions.

3 **OPolo**: Off-Policy Learning from Observations

3.1 Surrogate Objective

The idea of re-using cached transitions to improve sample-efficiency has been adopted by many RL algorithms \[7, 26, 23\]. In the same spirit, we start by introducing an off-policy distribution \( \mu^R(s, a) \), which is induced by a dataset \( \mathcal{R} \) of historical transitions. Choosing KL-divergence as a discrepancy measure, we obtain an upper-bound of the LfO objective by involving \( \mu^R(s, a) \) (see Sec 9.1 in the appendix for proof):

\[
\text{D}_{\text{KL}} \left[ \mu^\pi(s, s') || \mu^E(s, s') \right] \leq E_{\mu^\pi(s, s')} \left[ \log \frac{\mu^R(s, s')}{\mu^E(s, s')} \right] + \text{D}_{\text{KL}} \left[ \mu^\pi(s, a) || \mu^R(s, a) \right].
\]

As a result, the LfO objective can be optimized by minimizing the RHS of Eq 4. Although widely adopted for its interpretability, KL-divergence can be tricky to estimate due to issues of biased gradients \[29 \[3\]. To avoid the potential difficulty in optimization, we further substitute the term \( \text{D}_{\text{KL}}[\mu^\pi(s, a) || \mu^R(s, a)] \) in Eq 4 by a more aggressive \( f \)-divergence, with \( f(x) = \frac{1}{2}x^2 \), which serves as an upper-bound of the KL-divergence (See Sec 9.4 in the appendix):

\[
\text{D}_{\text{KL}}[P || Q] \leq \text{D}_f[P || Q].
\]

Our choice of \( f \)-divergence can be considered as a variant of Pearson \( \chi^2 \)-divergence with a constant shift, which has also been adopted as a valid measure of distribution discrepancies \[30 \[31\]. Compared with KL-divergence, this \( f \)-divergence enables unbiased estimation without deteriorating the optimality, whose advantages will become increasingly visible in Section 3.2.

Built upon the above transformations, we reach an objective that serves as an effective upper-bound of \( \text{D}_{\text{KL}}[\mu^\pi(s, s') || \mu^E(s, s')] \):

\[
\text{Table 1: Summarization on different stationary distributions, with } \mu^T(s) = p(s_t = s | s_0 \sim p_0( \cdot ), a_t \sim \pi(\cdot | s_i), s_{i+1} \sim P(\cdot | s_i, a_i)), \forall i < t).
\]
We first leverage the dual-form of an $f$ which upon training to optimality, satisfies \( \log(f(x)) \). A similar rationale has also been the key component of 3.3 Adversarial Training with Off-Policy Experience and use this dual transformation to rewrite Eq (6):

$$\min_{\pi} J_{\text{onpolo}}(\pi) := E_{\mu_{\pi}(s,a)} \left[ \log \frac{\mu^E(s,a)}{\mu^R(s,a)} \right] + D_f(\mu^\pi(s,a)||\mu^R(s,a)). \tag{6}$$

### 3.2 Off-Policy Transformation

Optimization Eq (6) is still on-policy and induces additional challenges through the term \( D_f(\mu^\pi(s,a)||\mu^R(s,a)). \) However, we show that it can be readily transformed into off-policy learning. We first leverage the dual-form of an $f$-divergence [32]:

$$-D_f(\mu^\pi(s,a)||\mu^R(s,a)) = \inf_{x:S \times A \rightarrow R} E_{(s,a) \sim \mu^\pi} \left[-x(s,a) + E_{(s,a) \sim \mu^R} [f_*(x(s,a))]\right],$$

and use this dual transformation to rewrite Eq (6):

$$\min_{\pi} J_{\text{onpolo}}(\pi) \equiv \max_{\pi} E_{\mu_{\pi}(s,a)} \left[ -\log \frac{\mu^E(s,a)}{\mu^R(s,a)} \right] - D_f(\mu^\pi(s,a)||\mu^R(s,a))$$

$$\equiv \max_{\pi} \min_{x:S \times A \rightarrow R} J_{\text{onpolo}}(\pi, x) := E_{\mu^\pi(s,a)} \left[ \log \frac{\mu^E(s,a)}{\mu^R(s,a)} - x(s,a) \right] + E_{\mu_R(s,a)} [f_*(x(s,a))]. \tag{7}$$

If we consider a synthetic reward as \( r(s, a, s') = \log \frac{\mu^E(s,a,s')}{\mu^R(s,a,s')} - x(s,a), \) the first term in Eq (7) resembles an RL return function: \( J(\pi) = E_{(s,a,s') \sim \mu^\pi(s,a,s')} [r(s,a,s')] \). Observing this similarity, we turn to learning a $Q$-function by applying a change of variables:

$$Q(s, a) = E_{s' \sim P(\cdot|s,a)} \left[ -x(s,a) + \log \frac{\mu^E(s,a,s')}{\mu^R(s,a,s')} + \gamma Q(s', a') \right].$$

Equivalently, this $Q$ function is a fixed point of a variant Bellman operator \( B^\pi Q: \)

$$Q(s, a) = -x(s,a) + E_{s' \sim P(\cdot|s,a)} \left[ \log \frac{\mu^E(s,a,s')}{\mu^R(s,a,s')} + \gamma Q(s', a') \right] = -x(s,a) + B^\pi Q(s,a).$$

Rewriting $x(s,a) = (B^\pi Q - Q)(s,a)$ and applying it back to Eq (7), we finally remove the on-policy expectation by a series of telescoping (see Sec 9.6 in the appendix for derivation):

$$\max_{\pi} \min_{x:S \times A \rightarrow R} J_{\text{onpolo}}(\pi, x) \equiv \max_{\pi} \min_{Q:S \times A \rightarrow R} J_{\text{onpolo}}(\pi, Q)$$

$$:= E_{(s,a,s') \sim \mu^\pi(s,a,s')} \left[ \log \frac{\mu^E(s,a,s')}{\mu^R(s,a,s')} - (B^\pi Q - Q)(s,a) \right] + E_{(s,a,s') \sim \mu_R(s,a)} [f_*(Q(s,a))]$$

$$= (1 - \gamma)E_{s_0 \sim p_0, a_0 \sim \pi(\cdot|s_0)} [Q(s_0, a_0)] + E_{(s,a,s') \sim \mu_R(s,a)} [f_*(B^\pi Q - Q)(s,a))]. \tag{8}$$

A similar rationale has also been the key component of distribution error correction (DICE) [30][31][33]. Based on the above transformation, we propose our main objective:

$$\max_{\pi} \min_{Q:S \times A \rightarrow R} J_{\text{onpolo}}(\pi, Q) := (1 - \gamma)E_{s_0 \sim p_0, a_0 \sim \pi(\cdot|s_0)} [Q(s_0, a_0)] + E_{(s,a,s') \sim \mu_R(s,a)} [f_*(B^\pi Q - Q)(s,a)).$$

Specifically, when \( f(x) = f^*(x) = \frac{1}{2}x^2 \), the second term \( E_{\mu_R(s,a)} [f_*(B^\pi Q - Q)(s,a)) \) is reminiscent of an Bellman error, for which we can have unbiased estimation by mini-batch gradients. Given access to the off-policy distribution \( \mu^R(s,a) \) and the initial distribution \( p_0 \), optimization (9) can be efficiently realized once we resolve the term \( \log \frac{\mu^E(s,a,s')}{\mu^R(s,a,s')} \) contained in \( B^\pi Q(s,a). \)

### 3.3 Adversarial Training with Off-Policy Experience

We can take the advantage of GAN training [25] to estimate the term \( \log \frac{\mu^E(s,a,s')}{\mu^R(s,a,s')} \) inside \( B^\pi Q(s,a) \), by learning a discriminator \( D \):

$$\max_{D:S \times A \rightarrow R} \mathbb{E}_{(s,s') \sim \mu^E(s,a,s')} \left[ \log(D(s,s')) \right] + \mathbb{E}_{(s,s') \sim \mu_R(s,a,s')} \left[ \log(1 - D(s,s')) \right],$$

which upon training to optimality, satisfies \( \log(\frac{\mu^E(s,a,s')}{\mu^R(s,a,s')}) = \log D^*(s,s') - \log(1 - D^*(s,s')). \)

Unlike prior art [2][9][4] that requires estimating the ratio of \( \log \frac{\mu^E}{\mu^R} \), the discriminator in our case is designed to be off-policy in accordance with our proposed objective. Up to this step, optimization (9) can be achieved by interactively optimizing $Q$, $\pi$, and $D$ with pure off-policy learning.
3.4 Policy Regularization as Forward Distribution Matching

Optimization [9] essentially minimizes an upper-bound of the inverse KL divergence $\mathbb{D}_{\text{KL}}[\mu^E(s, s') || \mu^P(s, s')]$, which is known to encourage a mode-seeking behavior [34]. Although mode-seeking is more robust to covariate-drift than mode-covering (such as behavior cloning), it requires sufficient explorations to find a reasonable state-distribution, especially at early learning stages. On the other hand, a mode-covering strategy has merits in quickly minimizing discrepancies on the expert distribution, by optimizing a forward KL-divergence such as $\mathbb{D}_{\text{KL}}[\pi_E(a|s)||\pi(a|s)]$.

To combine the advantages of both, in this section we show how we further speed up the learning procedure from a mode-covering perspective, without deteriorating the efficacy of our main objective. To achieve this goal, we first derive an optimizable lower-bound from a mode-covering objective:

$$\mathbb{D}_{\text{KL}}[\pi_E(a|s)||\pi(a|s)] = \mathbb{D}_{\text{KL}}[\mu^E(s, s') || \mu^\pi(s', s)] + \mathbb{D}_{\text{KL}}[\mu^E(a|s, s') || \mu^\pi(a|s, s')]$$

(10)

in which we define $\mu^\pi(s'|s) = \int a \pi(a|s) P(s'|s, a) da$ as the conditional state transition distribution induced by $\pi$, likewise for $\mu^E(s'|s)$ (see Sec 9.5.3 in the appendix).

Similar to Remark [1], the discrepancy $\mathbb{D}_{\text{KL}}[\mu^E(a|s, s') || \mu^P(a|s, s')]$ is not optimizable without knowing expert actions. However, under some mild assumptions, we found it feasible to optimize the other term $\mathbb{D}_{\text{KL}}[\mu^E(s'|s) || \mu^\pi(s'|s')]$ by enforcing a policy regularization:

**Remark 2.** In a deterministic MDP, assuming the support of $\mu^E(s, s')$ is covered by $\mu^P(s, s')$, s.t. $\mu^E(s, s') > 0 \implies \mu^P(s, s') > 0$, then regulating policy using $\mu^P(\cdot|s, s')$ minimizes $\mathbb{D}_{\text{KL}}[\mu^E(s'|s) || \mu^\pi(s'|s')]$ (See Sec 9.5.2 in supplementary for a detailed discussion):

$$\exists \tilde{\pi} : \mathcal{S} \to \mathcal{A}, \text{s.t. } \forall (s, s') \sim \mu^E(\cdot|s, s'), \tilde{\pi}(s) \propto \mu^P(\cdot|s, s') \implies \tilde{\pi} = \arg \min_{\pi} \mathbb{D}_{\text{KL}}[\mu^E(s'|s) || \mu^\pi(s'|s')]$$

Intuitively, when expert labels are unavailable, this regularization can be considered as performing states matching, by encouraging the policy to yield actions that lead to desired footprints. Given a transition $s \to s'$ from the expert observations, a conditional distribution $\mu^P(\cdot|s, s')$ only has support on actions that yield this transition $s \to s'$. Therefore, following this regularization avoids the policy from drifting to undesired states.

In practice, we can estimate $\mu^P(\cdot|s, s')$ by learning an inverse action model $P_I$ using off-policy transitions from $\mu^P(s, a, s')$ to optimize the following (See Sec 9.5.3 in the appendix):

$$\max_{P_I: \mathcal{S} \times \mathcal{S} \to \mathcal{A}} I_{\mathcal{S}, \mathcal{S} \to \mathcal{A}} = \max_{P_I: \mathcal{S} \times \mathcal{S} \to \mathcal{A}} \mathbb{E}_{(s, a, s') \sim \mu^P(s, a, s')} \log P_I(a|s, s')$$

(11)

3.5 Algorithm

Based on all the abovementioned building blocks, we now introduce OPOLO in Algorithm 1. OPOLO involves learning a policy $\pi$, a critic $Q$, a discriminator $D$, and an inverse action regularizer $P_I$, all of which can be done through off-policy training.

In particular, $\pi$ and $Q$ is jointly learned to find a saddle-point solution to optimization [9]. The discriminator $D$ assists this process by estimating a density ratio $\log \frac{\mu^E(s, s')}{\mu^P(s, s')}$. For better empirical performance, we adopt $-\log(1 - D(s, s'))$ as the discriminator’s output, which corresponds to a constant shift inside the logarithm term, in that $\log \left( \frac{\mu^E(s, s')}{\mu^P(s, s')} + 1 \right) = -\log(1 - D(s, s'))$. The inverse action model $P_I$ serves as a regularizer to infer proper actions on the expert observation distribution to encourage mode-covering. We defer more implementation details to Sec 9.7 in the appendix.

4 Related Work

Recent development on imitation learning can be divided into two categories:

**Learning from Demonstrations (LfD)** traces back to behavior cloning (BC) [35], in which a policy is pre-trained to minimize the prediction error on expert demonstrations. This approach is inherent with issues such as distribution shift and regret propagations. To address these limitations, [11] proposed a no-regret IL approach called DAGger, which however requires online access to oracle corrections. More recent LfD approaches favor Inverse reinforcement learning (IRL) [8], which work by seeking a reward function that guarantees the superiority of expert demonstrations, based on which regular RL algorithms can be used to learn a policy [36, 37]. A representative instantiation of IRL is Generative Adversarial Imitation Learning (GAIL) [2]. It defines IL as a distribution matching problem and leverages the GAN technique [25] to minimize the Jensen-Shannon divergence between distributions induced by the expert and the learning policy. The success of GAIL has inspired many other related
We compare \textbf{OPOLO} which are locomotion tasks in continuous state-action space. In accordance with our assumption in ValueDICE [3], which inherits the idea of DICE [30] to transform an on-policy LfD objective to an off-policy one. This approach, however, requires the information of expert actions, which otherwise makes off-policy estimation unreachable in a model-free setting. Therefore, their approach is not directly applicable to LfO. We have analyzed this dilemma in Sec 9.8 in the appendix.

Learning from Observations (LfO) tackles a more challenging scenario where expert actions are unavailable. Work along this line falls into model-free and model-based approaches. GAIL [9] is a model-free solution which applies the principle of GAIL to learn a discriminator with state-only inputs. IDDM [10] further analyzed the theoretical gap between the LfD and LfO objectives, and proved that a lower-bound of this gap can be somewhat alleviated by maximizing the mutual-information between \((s, (a, s'))\), given an on-policy distribution \(\mu^\pi(s, a, s')\). Its performance is comparable to GAIL. [24] assumed that the given observation sequences are ranked by superiority, based on which a reward function is designed for policy learning. Similar to GAIL, the sample efficiency of these approaches is suboptimal due to their on-policy strategy.

Model-based LfO can be further organized into learning a forward [23, 39] dynamics model or an inverse action model [17, 21]. Especially, [23] proposed a forward model solution to learn time-dependent policies for finite-horizon tasks, in which the number of policies to be learned equals the number of transition steps. This approach may not be suitable for tasks with long or infinite horizons. Behavior cloning from observations (BCO) [17] learns an inverse model to infer actions missing from the expert dataset, after which behavior cloning is applied to learn a policy. Besides the common issues faced by BC, this strategy does not guarantee that the ground-truth expert actions can be recovered, unless is a deterministic and injective MDP is assumed. Some other recent work focused on different problem settings than ours, in which the expert observations are collected with different transition dynamics [40] or from different viewpoints [21, 41, 42]. Readers are referred to [11] for further discussions of LfO.

5 Experiments

We compare \textbf{OPOLO} against state-of-the-art LfD and LfO approaches on MuJuCo benchmarks, which are locomotion tasks in continuous state-action space. In accordance with our assumption in Sec 3.4 these tasks have deterministic dynamics. Original rewards are removed from all benchmarks to fit into an IL scenario. For each task, we collect 4 trajectories from a pre-trained expert policy. All illustrated results are evaluated across 5 random seeds.
**Baselines:** We compared SAIL against 7 baselines. We first selected 5 representative approaches from prior work: GAIL (on-policy LfD), DAC (off-policy LfD), ValueDICE (off-policy LfD), GAILfO (on-policy LfO), and BCO (off-policy LfO). We further designed two strong off-policy approaches. Specifically, we built DACfO, which is a variation of DAC that learns the discriminator on \((s, s')\) instead of \((s, a)\), and ValueDICEfO, which is built based on ValueDICE. Instead of using ground-truth expert actions, ValueDICEfO learns an inverse model by optimizing Eq (11), and uses the approximated actions generated by the inverse model to fit an LfO problem setting. To the best of our knowledge, DACfO and ValueDICEfO have not been investigated by any prior art. Among these baselines, GAIL, DAC, and ValueDICE are provided with both expert states and actions, while all other approaches only have access to expert states. More experimental details can be found in the supplementary material.

Our experiments focus on answering the following important questions:

1. **Asymptotic performance:** Is OPOLO able to achieve expert-level performance given a limited number of expert observations?
2. **Sample efficiency:** Can OPOLO recover expert policy using less interactions with the environment, compared with the state-of-the-art?
3. **Effects of the inverse action regularization:** Does the inverse action regularization useful in speeding up the imitation learning process?
4. **Sensitivity of the choice of \(f\)-divergence:** Can OPOLO perform well given different \(f\) functions?

### 5.1 Performance Comparison

OPOLO can recover expert performance given a fixed budget of expert observations. As shown in Figure 1, OPOLO reaches (near) optimal performance in all benchmarks. For simpler tasks such as Swimmer and InvertedPendulum, most baselines can successfully recover expertise. For other complex tasks with high state-action space, on-policy baselines, such as GAIL and GAILfO, are struggling to reach their asymptotic performance within a limited number of interactions. As shown in Figure 2, the off-policy baseline BCO is prone to sub-optimality due to its behavior cloning-like strategy. On the other hand, the performance of ValueDICEfO can be deteriorated by potential action-drifts, as the inferred actions are not guaranteed to recover expertise. For fair comparison, performance of all off-policy approaches are summarized in Table 2 given a fixed number of interaction steps.

The asymptotic performance of OPOLO is 1) superior to DACfO and ValueDICEfO, 2) comparable to DAC, and 3) is more robust against overfitting compared with ValueDICE, whereas both DAC and ValueDICE enjoy the advantage of off-policy learning and extra action guidance.

| Env   | HalfCheetah | Hopper | Walker | Swimmer | Ant |
|-------|-------------|--------|--------|---------|-----|
| BCO   | 3881.10±938.81 | 1845.66±628.41 | 421.24±135.18 | 256.88±4.52 | 1529.54±980.86 |
| OPOLO | **7632.80±128.88** | 3851.85±19.08 | 3947.72±97.88 | 246.62±1.56 | 5112.04±327.42 |
| DAC   | 7336.96±117.89 | 3517.39±25.16 | 3803.00±979.85 | 257.38±4.28 | **5783.57±651.98** |
| DACfO | 6900.00±131.24 | 3534.42±10.27 | **4131.05±174.13** | 232.12±2.04 | 5424.28±394.82 |
| ValueDICE | 7035.63±444.14 | 3522.95±93.15 | 3033.02±207.63 | 183.28±2.67 | 4920.76±872.06 |
| ValueDICEfO | 5696.94±2116.94 | **3591.37±8.60** | 1641.38±1230.73 | 262.73±7.76 | 3486.87±1232.25 |
| Expert | 4770.37±644.49 | 3579.51±10.23 | 431.00±140.87 | **265.05±3.45** | 73.08±400.87 |

Table 2: Evaluated performance of off-policy approaches. Results are averaged over 50 trajectories.

### 5.2 Sample Efficiency

OPOLO is comparable with and sometimes superior to DAC in all evaluated tasks, and is much more sample-efficient than on-policy baselines. As shown in Figure 1 the sample-efficiency of OPOLO is emphasized by benchmarks with high state-action dimensions. In particular, for tasks such as Ant or HalfCheetah, the performance curves of on-policy baselines are barely improved at early learning stages. One intuition is that they need more explorations to build the current support of the learning policy, which cannot benefit from cached transitions. For these challenging tasks, OPOLO is even more sample-efficient than DAC that has the guidance of expert actions. We ascribe this improvement to the mode-covering regularization of OPOLO enforced by its inverse action model, whose effect will be further analyzed in Sec 5.3. Meanwhile, other off-policy approaches such as BCO and
ValueDICEfO, are prone to overfitting and performance degradation (as shown in Figure 1), which indicates that the effect of the inverse model alone is not sufficient to recover expertise. On the other hand, the ValueDICE algorithm, although being sample-efficient, is not designed to address LiFO and requires expert actions.

Figure 1: Interaction steps (x-axis) versus learning performance (y-axis). Compared with GAIL, BCO, GAILfO, and DAC, our proposed approach (OPOLO) is the most sample-efficient to reach expert-level performance (Grey horizontal line).

Figure 2: Compared with strong off-policy baselines, OPOLO is the only approach that consistently achieves competitive performance regarding both sample-efficiency and asymptotic performance across all tasks, without accessing expert actions.

5.3 Ablation Study

In this section, we further analyze the effects of the inverse action regularization by a group of ablation studies. Especially, we implement a variant of OPOLO that does not learn an inverse action model to regulate the policy update. We compare this approach, dubbed as OPOLO-x, against our original approach as well as the DAC algorithm.

Effects on Sample efficiency: Performance curves in Figure 2 show that removing the inverse action regularization from OPOLO slightly affects its sample-efficiency, although the degraded version is still comparable to DAC. This impact is more visible in challenging tasks such as HalfCheetah and Ant. From another perspective, the same phenomenon indicates that an inverse action regularization is beneficial for accelerating the IL process, especially for games with high observation-space. An intuitive exploration is that, while our main objective serves as a driving force for mode-seeking, a regularization term assists by encouraging the policy to perform mode-covering. Combing these two motivations leads to a more efficient learning strategy.
Effects on Performance: Given a reasonable number of transition steps, the effects of an inverse-action model are less obvious regarding the asymptotic performance. As shown in Table 2, OPOLO-x is mostly comparable to OPOLO and DAC. This implies that the effect of the state-covering regularization will gradually fade out once the policy learns a reasonable state distribution. From another perspective, it indicates that following our main objective alone is sufficient to recover expert-level performance. Comparing with BCO which uses the inverse model solely for behavior cloning, we find it more effective when serving as a regularization to assist distribution matching from a forward direction.

Figure 3: Removing the inverse action regularization (OPOLO-x) results in slight efficiency drop, although its performance is still comparable to OPOLO and DAC.

5.4 Sensitivity Analysis
To analyze the effects of different f-functions on the performance of the proposed approach, we explored a family of f-divergence where \( f(x) = \frac{1}{p} |x|^p, f^*(y) = \frac{1}{q} |y|^q, \) s.t. \( \frac{1}{p} + \frac{1}{q} = 1, p, q > 1, \) as adopted by DualDICE [30]. Evaluation results show that OPOLO yields reasonable performance across different f-functions, although our choice \( (q = p = 2) \) turns out to be most stable. Results using the Ant task is illustrated in Figure 4.

6 Conclusions
Towards sample-efficient imitation learning from observations (LfO), we proposed a principled approach that performs imitation learning by accessing only a limited number of expert observations. We derived an upper-bound of the original LfO objective to enable efficient off-policy optimization, and augment the objective with an inverse action model regularization to speed up the learning procedure. Extensive empirical studies are done to validate the proposed approach.

7 Acknowledgments
This research was jointly supported by the National Science Foundation IIS-1749940, and the Office of Naval Research N00014-20-1-2382. We would like to thank Dr. Boyang Liu and Dr. Junyuan Hong (Michigan State University) for providing insightful comments. We also appreciate Dr. Mengying Sun (Michigan State University) for her assistance in proofreading the manuscript.
8 Broader Impact

The success of Imitation Learning (IL) is crucial for realizing robotic intelligence. Serving as an effective solution to a practical IL setting, OPOLO has a promising future in various applications, including robotics control [43], game-playing [6], autonomous driving [14], algorithmic trading [44], to name just a few.

On one hand, OPOLO provides an working evidence of sample-efficient IL. OPOLO costs less environment interactions compared with conventional IL approaches. For tasks where taking real actions can be expensive (high-frequency trading) or dangerous (autonomous driving), using less interactions for imitation learning is a crucial requirement for successful applications.

On the other hand, OPOLO validates the feasibility of learning from incomplete guidance, and can enable IL in applications where expert demonstrations are costly to access. Moreover, OPOLO is more resemblant to human intelligence, as it can recover expertise simply by learning from expert observations. In general, OPOLO has a strong impact on the advancement of IL, from the perspective of both theoretical and empirical studies.

References

[1] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics, pages 627–635, 2011.

[2] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In Advances in neural information processing systems, pages 4565–4573, 2016.

[3] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. International Conference on Learning Representations (ICLR), 2019.

[4] Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. ICLR, 2019.

[5] Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. ICLR, 2019.

[6] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. Nature, 550(7676):354–359, 2017.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

[8] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the twenty-first international conference on Machine learning, page 1. ACM, 2004.

[9] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. ICML Workshop on Imitation, Intent, and Interaction (I3), 2019.

[10] Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. In Advances in Neural Information Processing Systems, pages 239–249, 2019.

[11] Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. IJICAI, 2019.

[12] Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In Advances in Neural Information Processing Systems, pages 2930–2941, 2018.
[13] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. ICLR, 2017.

[14] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19):70–76, 2017.

[15] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng. High speed obstacle avoidance using monocular vision and reinforcement learning, In Proceedings of the 22nd international conference on Machine learning, pages 593–600, 2005.

[16] Fumihiro Sasaki, Tetsuya Yohira, and Atsuo Kawaguchi. Sample efficient imitation learning for continuous control. ICLR, 2019.

[17] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. Proceedings of the 27th International Joint Conference on Artificial Intelligence(IJCAI 2018), 2018.

[18] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

[19] Huang Xiao, Michael Herman, Joerg Wagner, Sebastian Ziesche, Jalal Etesami, and Thai Hong Linh. Wasserstein adversarial imitation learning. arXiv preprint arXiv:1906.08113, 2019.

[20] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In Advances in neural information processing systems, pages 271–279, 2016.

[21] Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. ICLR, 2019.

[22] Faraz Torabi, Garrett Warnell, and Peter Stone. Adversarial imitation learning from state-only demonstrations. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pages 2229–2231. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[23] Wen Sun, Anirudh Vemula, Byron Boots, and J Andrew Bagnell. Provably efficient imitation learning from observation alone. arXiv preprint arXiv:1905.10948, 2019.

[24] Daniel S Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. arXiv preprint arXiv:1904.06387, 2019.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[26] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290, 2018.

[27] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. arXiv preprint arXiv:1802.09477, 2018.

[28] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.

[29] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. Communications on Pure and Applied Mathematics, 28(1):1–47, 1975.

[30] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In Advances in Neural Information Processing Systems, pages 2315–2325, 2019.
[31] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. arXiv preprint arXiv:1912.02074, 2019.

[32] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory, 56(11):5847–5861, 2010.

[33] Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. International Conference on Learning Representations (ICLR), 2020.

[34] Seyed Kamyar Seyed Ghasemipour, Shane Gu, and Richard Zemel. Understanding the relation between maximum-entropy inverse reinforcement learning and behaviour cloning. ICLR 2019 Workshop, 2019.

[35] Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. Neural computation, 3(1):88–97, 1991.

[36] Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. In Advances in neural information processing systems, pages 1449–1456, 2008.

[37] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In Aaai, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[38] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.

[39] Ashley D Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L Isbell. Imitating latent policies from observation. ICML, 2019.

[40] Tanmay Gangwani and Jian Peng. State-only imitation with transition dynamics mismatch. ICLR, 2020.

[41] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey 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.

[42] Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. ICLR, 2017.

[43] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research, 32(11):1238–1274, 2013.

[44] Yue Deng, Feng Bao, Youyong Kong, Zhiqian Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. IEEE transactions on neural networks and learning systems, 28(3):653–664, 2016.
9 Appendix

For all the following derivations, we use $D_{KL}[P(X)||Q(X)]$ to denote the KL-divergence between two distributions $P$ and $Q$:

$$D_{KL}[P(X)||Q(X)] = E_{x \sim p(x)} \log \frac{p(x)}{q(x)} = \int_X p(x) \log \frac{p(x)}{q(x)} dx.$$ 

Accordingly, when $P(X|Z)$ and $Q(X|Z)$ are conditional distributions, $D_{KL}[P||Q]$ denotes their conditional KL-divergence:

$$D_{KL}[P(X|Z)||Q(X|Z)] = \int_{Z \times X} p(z|x) \log \frac{p(x|z)}{q(x|z)} dxdz.$$ 

For simplicity, we will equivalently use $E_{x \sim p(x)[\cdot]}$ and $E_{p(x)[\cdot]}$ to denote certain expectation in which $x$ is sampled from the distribution $P(X)$.

9.1 Derivation of Surrogate Objective

We first refer Lemma 1 from [10] for a complete presentation:

**Lemma 1.**

$$D_{KL}[\mu^\pi(s, a, s')||\mu^E(s, a, s')] = D_{KL}[\mu^\pi(s, a)||\mu^E(s, a)].$$

**Proof.**

$$D_{KL}[\mu^\pi(s, a, s')||\mu^E(s, a, s')] = \int_{S \times A \times S} \mu^\pi(s, a, s') \log \frac{\mu^\pi(s, a, s') P(s'|s, a)}{\mu^E(s, a, s') P(s'|s, a)} ds'dads = \int_{S \times A \times S} \mu^\pi(s, a, s') \log \frac{\mu^\pi(s, a)}{\mu^E(s, a)} ds'dads = \int_{S \times A} \mu^\pi(s, a) \log \frac{\mu^\pi(s, a)}{\mu^E(s, a)} dads = D_{KL}[\mu^\pi(s, a)||\mu^E(s, a)].$$

**Lemma 2.**

$$D_{KL}[\mu^\pi(s, s')||\mu^E(s, s')] \leq D_{KL}[\mu^\pi(s, a)||\mu^E(s, a)].$$

**Proof.** As defined in Table 1, $\mu^\pi(a|s, s')$ is the inverse-action transition probability induced by policy $\pi$:

$$\mu^\pi(a|s, s') = \frac{\mu^\pi(s, a, s')}{\mu^\pi(s, s')} = \frac{\mu^\pi(s, \pi(a|s) P(s'|s, a))}{\int_A \mu^\pi(s, \pi(a|s) P(s'|s, a))da} = \frac{\pi(a|s) P(s'|s, a)}{\int_A \pi(a|s) P(s'|s, a)da}$$

Based on this notion, we can derive:

$$D_{KL}[\mu^\pi(s, a)||\mu^E(s, a)] = D_{KL}[\mu^\pi(s, a, s')||\mu^E(s, a, s')]$$

$$= \int_{S \times A \times S} \mu^\pi(s, a, s') \log \frac{\mu^\pi(s, a, s')}{\mu^E(s, a, s')} ds'dads$$

$$= \int_{S \times A \times S} \mu^\pi(s, s') \mu^\pi(a, s') \log \frac{\mu^\pi(s, s') \times \mu^\pi(a|s, s')}{\mu^E(s, s') \times \mu^E(a|s, s')} ds'dads$$

13
whose transition dynamics is shown in Table 3, with Theorem 2.

\begin{align}
&= \int_{S \times A \times S} \mu^e(s, s') \mu^a(s, s') \log \frac{\mu^e(s, s')}{\mu^a(s, s')} ds'ds + \int_{S \times A \times S} \mu^e(s, s') \mu^a(s, s') \log \frac{\mu^e(a, s, s')}{\mu^a(a, s, s')} ds'ds \\
&= \int_{S \times A \times S} \mu^e(s, s') \log \frac{\mu^e(s, s')}{\mu^a(s, s')} ds'ds + D_{KL}[\mu^e(a, s, s')||\mu^a(a, s, s')] \\
&\geq D_{KL}[\mu^e(s, s')||\mu^a(s, s')] \\
&\geq D_{KL}[\mu^e(s, s')||\mu^a(s, s')].
\end{align}

Based on Lemma 2, we can derive the upper-bound of our original objective:

**Theorem 1 (Surrogate Objective as the Divergence Upper-bound).**

\[
D_{KL}[\mu^e(s, s')||\mu^a(s, s')] \leq E_{\mu^a(s, s')}[\log \frac{\mu^R(s, s')}{\mu^R(s, s')} + D_{KL}[\mu^e(s, a)||\mu^R(s, a)].
\]

**Proof.** We can refer Eq (12) from the proof of Lemma 2.

9.2 Connections between LfO and LfD

**Theorem 2.**

\[
D_{KL}[\mu^e(a, s, s')||\mu^a(a, s, s')] = D_{KL}[\mu^e(s, a)||\mu^a(s, a)] - D_{KL}[\mu^e(s, s')||\mu^a(s, s')].
\]

**Proof.** We can refer Eq (12) from the proof of Lemma 2.

\[
D_{KL}[\mu^e(s, a)||\mu^a(s, a)] = D_{KL}[\mu^e(s, s')||\mu^a(s, s')] + D_{KL}[\mu^e(a, s, s')||\mu^a(a, s, s')].
\]

9.3 An Unoptimizable Gap Between LfO and LfD

**Remark 1** In a non-injective MDP, the discrepancy of \(D_{KL}[\mu^e(a, s, s')||\mu^a(a, s, s')]\) cannot be optimized without knowing expert actions.

**Proof.** We provide proof with a counter-example. Consider a non-injective MDP in a tabular case, whose transition dynamics is shown in Table 3 with |S| = 3, and |A| = 4. Especially, there exists two actions which lead to the same deterministic transition, i.e. for \(s_1, s_2 \in S\), \(\exists a_0, a_2 \in A\), s.t. \(P(s_2|s_1, a_2) = P(s_2|s_1, a_0) = 1\), as illustrated in Figure 5.

In this MDP, there is an expert policy \(\pi_E\) as listed in Table 5. Trajectories generated by this expert are illustrated as blue lines in Figure 5. In a LfO scenario, a learning agent only has access to sequences
We provide proof in a finite, discrete state space, although the conclusion is valid to extend to continuous cases.

Based on the given observations $R_E$, a policy $\pi$ can only satisfy the state distribution matching with $D_{KL}[\mu^\pi(s,s')||\mu^E(s,s')] = 0$, but unable to optimize $D_{KL}[\mu^\pi(a|s,\pi)||\mu^E(a|s,\pi)]$, as both $a_0$ and $a_3$ lead to a deterministic transition of $s_1 \rightarrow s_2$. In lack of expert actions, the best guess for a learning policy is to equally distribute action probabilities with $\pi(a_0|s_1) = (a_2|s_1) = 0.5$. which results in $\mu^\pi(a_0|s_1, s_2) = \mu^\pi(a_2|s_1, s_2) = 0.5$, whereas $\mu^E(a_2|s_1, s_2) = 1$, $\mu^E(a_0|s_0, s_1) = 0$. Consequently, we reach at $D_{KL}[\mu^\pi(a|s, s')||\mu^E(a|s, s')] > 0\). □

**Remark:** In a deterministic and injective MDP, it satisfies that $\forall \pi : S \rightarrow A$, $D_{KL}[\mu^\pi(a|s, s')||\mu^E(a|s, s')] = 0$.

We provide proof in a finite, discrete state-action space, although the conclusion is valid to extend to continuous cases.

**Proof.** In a deterministic and injective MDP, we can interpret the transition dynamics with a deterministic function $g$:

$$\exists g : S \times A \rightarrow S, \text{ s.t. } \forall (s, a, s'), g(s, a) = s' \iff P(s'|s,a) = 1, \text{ and } g(s, a) \neq s' \iff P(s'|s,a) = 0.$$  

since this MDP is also injective, given arbitrary policy $\pi$ and a transition $s \rightarrow s'$, $(s, s') \sim \mu^\pi(s, s')$, there exists one and only action $a$ which satisfies $g(s, a) = s', P(s'|s,a) = 1$.

Accordingly, $\mu^\pi(a|s, s') = \frac{\pi(a|s)P(s'|s,a)}{\sum_{a'\in A}P(s'|s,a')} = 1[g(s, a) = s']$ depends only on the transition dynamics, where $1(x)$ is an indicator function. The same conclusion applies to $\mu^E(a|s, s')$ as well. Therefore, we reach at:

$$\forall \pi : S \rightarrow A, \ D_{KL}[\mu^\pi(a|s, s')||\mu^E(a|s, s')]$$
Theorem 4. For two arbitrary distributions \( P \) and \( Q \), the \( f \)-divergence satisfies that \( D_{KL}[P||Q] \leq D_f[P||Q] \).

Proof. Given two distributions \( P \) and \( Q \), their density ratio is denoted as \( w_{p/q} \), with \( w_{p/q} = \frac{p(x)}{q(x)} \geq 0 \). If we consider a function \( g(w) = w \log(w) - \frac{1}{2}w^2 \), \( g(w) \) is constantly decreasing when \( w \in (0, \infty) \), as \( \frac{d g}{d w} = \log w + 1 - w \leq 0 \forall w \geq 0 \).

Since KL-Divergence is a special case of \( f \)-divergence with \( f_{KL}(x) = x \log x \), it is sufficient to show that:

\[
D_{KL}[P||Q] = \int_X q(x) \left( w_{p/q} \log(w_{p/q}) - \frac{1}{2}(w_{p/q})^2 \right) dx
\]

\[
\leq \int_X q(x) \sup_{w \in (0, +\infty)} (w \log(w) - \frac{1}{2}w^2) dx
\]

\[
= \int_X q(x) \lim_{w \to 0^+} (w \log(w) - \frac{1}{2}w^2) dx
\]

\[
= 0.
\]

\[\square\]

9.4 Upper-bound of the KL-Divergence

Theorem 3. For two arbitrary distributions \( P \) and \( Q \), and an \( f \)-divergence with \( f(x) = \frac{1}{2}x^2 \), it satisfies that \( D_{KL}[P||Q] \leq D_f[P||Q] \).

Proof. Given two distributions \( P \) and \( Q \), their density ratio is denoted as \( w_{p/q} \), with \( w_{p/q} = \frac{p(x)}{q(x)} \geq 0 \). If we consider a function \( g(w) = w \log(w) - \frac{1}{2}w^2 \), \( g(w) \) is constantly decreasing when \( w \in (0, \infty) \), as \( \frac{d g}{d w} = \log w + 1 - w \leq 0 \forall w \geq 0 \).

Since KL-Divergence is a special case of \( f \)-divergence with \( f_{KL}(x) = x \log x \), it is sufficient to show that:

\[
D_{KL}[P||Q] = \int_X q(x) \left( w_{p/q} \log(w_{p/q}) - \frac{1}{2}(w_{p/q})^2 \right) dx
\]

\[
\leq \int_X q(x) \sup_{w \in (0, +\infty)} (w \log(w) - \frac{1}{2}w^2) dx
\]

\[
= \int_X q(x) \lim_{w \to 0^+} (w \log(w) - \frac{1}{2}w^2) dx
\]

\[
= 0.
\]

\[\square\]

9.5 Forward Distribution Matching

9.5.1 Lower-bound of the BC Objective

Theorem 4.

\[
D_{KL}[\pi_E(a||s) || \pi(a||s)] = D_{KL}[\mu^E(s'||s) || \mu^E(s'|s)] + D_{KL}[\mu^E(a||s,s') || \mu^E(a||s,s')].
\]

Proof. Based on the definition of \( \mu^E(a||s,s') \) in Table[1],

\[
\mu^E(a||s,s') = \frac{\pi(a||s)P(s'||s,a)}{\int_A \pi(a||s)P(s'||s,a) d\bar{a}} = \frac{\pi(a||s)P(s'||s,a)}{\mu^E(s'|s)},
\]

and similar for \( \mu^E(a||s,s') \), we can derive at the following:

\[
D_{KL}[\pi_E(a||s) || \pi(a||s)]
\]

\[
= \int_{S \times A} \mu^E(s,a) \log \frac{\pi_E(a||s)}{\pi(a||s)} d\bar{s} d\bar{a}
\]

\[
= \int_{S \times A} \mu^E(s,a) \log \frac{\pi_E(a||s)}{\pi(a||s)} d\bar{s} d\bar{a}
\]

\[
= \int_{S \times A \times S} \mu^E(s,a)P(s'|s,a) \log \frac{\pi_E(a||s)P(s'|s,a)}{\pi(a||s)P(s'|s,a)} ds' d\bar{s} d\bar{a}
\]

\[
= \int_{S \times A \times S} \mu^E(s,a,s') \log \frac{\pi_E(a||s,s')P(s'|s,a)}{\pi(a||s)P(s'|s,a)} ds' d\bar{s} d\bar{a}
\]

\[
= \int_{S \times A \times S} \mu^E(s,a,s') \log \frac{\mu^E(a||s,s')\mu^E(s'|s)}{\mu^E(a||s,s')\mu^E(s'|s)} ds' d\bar{s} d\bar{a}
\]

\[\text{Eq. [13]}\]
Without loss of generality, in this section we provide proof based on a finite, discrete state-action space.

**Assumption 1** (Deterministic MDP). \( \exists g : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \) a deterministic function, s.t. \( \forall (s, a, s') \), \( g(s, a) \neq s' \) \( \iff P(s'|s, a) = 0 \), and \( g(s, a) = s' \) \( \iff P(s'|s, a) = 1 \).

Based on Assumption 1 we have the following:

**Corollary 1.** In a deterministic MDP, \( \forall \pi : \mathcal{S} \to \mathcal{A}, \mu^\pi(a|s, s') > 0 \implies P(a|s, s') = 1 \).

**Proof.** \( \mu^\pi(a|s, s') \propto \pi(a|s)P(s'|s, a) > 0 \implies P(s'|s, a) > 0 \). Based on Assumption 1 it holds that \( g(s, a) = s' \), therefore \( P(s'|s, a) = 1 \).

**Assumption 2** (Support Coverage). The support of expert transition distribution \( \mu^E(s, s') \) is covered by \( \mu^R(s, s') \):

\[ \mu^E(s, s') > 0 \implies \mu^R(s, s') > 0. \]

Combining Corollary 1 and Assumption 2 we can reach the following:

**Corollary 2.** \( \forall (s, s') \sim \mu^E(s, s'), \mu^R(a|s, s') > 0 \implies P(a|s, s') = 1 \).

**Lemma 3.** Given a policy \( \hat{\pi} \), s.t. \( \forall (s', s) \sim \mu^E(s, s'), \hat{\pi}(a|s) \propto \mu^R(a|s, s') \), then it satisfies that:

\[ \forall \pi : \mathcal{S} \to \mathcal{A}, \KL[\mu^E(s'|s)||\mu^\pi(s'|s)] \geq \KL[\mu^E(s'|s)||\mu^\hat{\pi}(s'|s)]. \]

**Proof.** In a discrete state-action space, \( \mu^\pi(s'|s) \) can be denoted as \( \mu^\pi(s'|s) = E_{a \sim \pi(\cdot|s)}[P(s'|s, a)] \), and the similar for \( \mu^\hat{\pi}(s'|s) \):

\[ \KL[\mu^E(s'|s)||\mu^\pi(s'|s)] - \KL[\mu^E(s'|s)||\mu^\hat{\pi}(s'|s)] = E_{\mu^E(s, s')} \left[ \log \frac{\mu^E(s'|s)}{\mu^\pi(s'|s)} - \log \frac{\mu^E(s'|s)}{\mu^\hat{\pi}(s'|s)} \right] = E_{\mu^E(s, s')} \left[ \log \mu^\pi(s'|s) - \log \mu^\hat{\pi}(s'|s) \right] = E_{\mu^E(s, s')} \left[ \log E_{a \sim \pi(\cdot|s)}[P(s'|s, a)] - E_{\mu^E(s, s')} \left[ \log E_{a \sim \hat{\pi}(\cdot|s)}[P(s'|s, a)] \right] \right] = E_{\mu^E(s, s')} \left[ \log E_{a \sim \pi(\cdot|s)}[P(s'|s, a)] - E_{\mu^E(s, s')} \left[ \log E_{a \sim \hat{\pi}(\cdot|s, s')}[P(s'|s, a)] \right] \right] \]

\[ = E_{\mu^E(s, s')} \left[ \log E_{a \sim \pi(\cdot|s)}[P(s'|s, a)] \right] \leq E_{\mu^E(s, s')} \left[ \log E_{a \sim \pi(\cdot|s)}[1] \right] = 0. \]

**Remark 2.** In a deterministic MDP, assuming the support of \( \mu^E(s, s') \) is covered by \( \mu^R(s, s') \), s.t. \( \mu^E(s, s') > 0 \implies \mu^R(s, s') > 0 \), then regulating policy using \( \mu^R(\cdot|s, s') \) can minimize \( \KL[\mu^E(s'|s)||\mu^\pi(s'|s)] \):

\[ \exists \hat{\pi} : \mathcal{S} \to \mathcal{A}, \forall (s, s') \sim \mu^E(s, s'), \hat{\pi}(\cdot|s) \propto \mu^R(\cdot|s, s') \implies \hat{\pi} = \arg \min_{\pi} \KL[\mu^E(s'|s)||\mu^\pi(s'|s)]. \]
Proof. Based on Lemma 5, we have that:
\[ \forall \pi : \mathcal{S} \rightarrow \mathcal{A}, \quad \mathbf{D}_{\text{KL}}[\mu^E(s'|s)] || \mu^\pi(s'|s) \geq \mathbf{D}_{\text{KL}}[\mu^E(s'|s)] || \mu^\pi(s'|s)]. \]

Therefore, \( \tilde{\pi} = \arg \min_{\pi} \mathbf{D}_{\text{KL}}[\mu^E(s'|s)] || \mu^\pi(s'|s)]. \)

\[ \square \]

9.5.3 Estimating the Inverse Action Distribution

Theorem 5.
\[
\max_{P_l : \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{A}} -\mathbf{D}_{\text{KL}}[\mu^R(a|s, s') || P_l(a|s, s')] \equiv \max_{P_l : \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{A}} E_{(s,a,s') \sim \mu^R(s,a,s')} [\log P_l(a|s, s')].
\]

Proof.
\[
-\mathbf{D}_{\text{KL}}[\mu^R(a|s, s') || P_l(a|s, s')] \\
= -\int_{\mathcal{S} \times \mathcal{S} \times \mathcal{A}} \mu^R(s, s') \mu^R(a|s, s') \log \frac{\mu^R(a|s, s')}{P_l(a|s, s')} da ds d's' \\
= -\int_{\mathcal{S} \times \mathcal{S} \times \mathcal{A}} \mu^R(s, s') \mu^R(a|s, s') \left( \log \mu^R(a|s, s') - \log P_l(a|s, s') \right) da ds d's' \\
= H[\mu^R(a|s, s')] + \int_{\mathcal{S} \times \mathcal{S} \times \mathcal{A}} \mu^R(s, s') \mu^R(a|s, s') \log P_l(a|s, s') da ds d's'.
\]

Note that we use \( H[\mu^R(a|s, s')] \) to denote the conditional entropy of \( \mu^R(a|s, s') \), with \( H[\mu^R(a|s, s')] = E_{\mu^R(s,a,s')} [-\log \mu^R(a|s, s')] \).

9.6 Derivation of Eq (8):
\[
J_{\text{global}}(\pi, Q) = E_{(s,a,s') \sim \mu_\pi(s,a,s')} [r(s, s') - (B^\pi Q - Q)(s, a)] + E_{(s,a) \sim \mu_R(s,a)} [f_\pi((B^\pi Q - Q)(s, a))]
\]

where \( B^\pi Q(s, a) = E_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')} [r(s, s') + \gamma Q(s', a')] \), and \( r(s, s') = \log \frac{E_{s,a}(s,a,s')}{\mu^R(s,a,s')} \).

Proof. The first term in the RHS of the above equation can be reduced to the following:
\[
E_{(s,a,s') \sim \mu_\pi(s,a,s')} [r(s, s') - (B^\pi Q - Q)(s, a)] \\
E_{(s,a) \sim \mu_R(s,a)} [E_{s' \sim P(\cdot|s,a)} [r(s, s') - ((B^\pi Q - Q)(s, a))]] \\
E_{(s,a) \sim \mu_R(s,a)} [E_{s' \sim P(\cdot|s,a)} [r(s, s') + Q(s, a) - E_{s' \sim P(\cdot|s,a)} (B^\pi Q(s, a))]] \\
E_{(s,a) \sim \mu_R(s,a)} [E_{s' \sim P(\cdot|s,a)} [r(s, s') + Q(s, a) - E_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')} [\gamma Q(s', a')]]] \\
E_{(s,a) \sim \mu_R(s,a)} [Q(s, a) - \gamma E_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')} [Q(s', a')]] \\
= (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t E_{s \sim \mu_\pi(s, a \sim \pi(s))} [Q(s, a)] - (1 - \gamma) \sum_{t=0}^{\infty} \gamma^{t+1} E_{s \sim \mu_\pi(s, a \sim \pi(s), s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s'))} [Q(s', a')]] \\
= (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t E_{s \sim \mu_\pi(s, a \sim \pi(s))} [Q(s, a)] - (1 - \gamma) \sum_{t=0}^{\infty} \gamma^{t+1} E_{s \sim \mu_\pi(s, a \sim \pi(s))} [Q(s, a)] \\
= (1 - \gamma) E_{s \sim P(s, a \sim \pi(s))} [Q(s, a)].
\]

Therefore:
\[
J_{\text{global}}(\pi, Q) = (1 - \gamma) E_{s \sim P(s, a \sim \pi(s))} [Q(s, a)] + E_{(s,a) \sim \mu_R(s,a)} [f_\pi((B^\pi Q - Q)(s, a))].
\]

\[ \square \]
9.7 Implementation Details

9.7.1 Practical Considerations for Algorithm Implementation

We provide some practical considerations to effectively implement our algorithm:

Initial state sampling: To increase the diversity of initial samples, we use state samples from an off-policy buffer and treat them as virtual initial states. A similar strategy is adopted by [3].

Constant shift on synthetic rewards: In practice, we adopt the same strategy of prior art [10] to use \( r(s, s') = -\log(1 - D(s, s')) \), instead of \( \log(D) - \log(1 - D) \) as the discriminator output. A fully optimized discriminator \( D^* \) satisfies \( -\log(1 - D^*(s, s')) = \log(1 + \frac{\mu^E(s, s')}{\mu^R(s, s')} \) \), which corresponds to a constant shift on \( \frac{\mu^E(s, s')}{\mu^R(s, s')} \) before the log term.

\( Q \) and \( \pi \) network update: We follow the advice of AlgeaDICE [31] by using a target Q network and policy gradient clipping. Especially, when taking the gradients of \( J_{\text{opolo}}(\pi, Q, \alpha) \) w.r.t. \( Q \), we use the value from a target Q network to calculate \( B^\pi Q(s, a) \) in order to stabilize training; on the other hand, since an optimal \( x^*(s, a) = (B^\pi Q^* - Q^*)(s, a) = \frac{\mu^E(s, a)}{\mu^R(s, a)} \) represents a density ratio and should always be non-negative, we clip \( (B^\pi Q - Q)(s, a) \) to above 0 when taking gradients w.r.t. \( \pi \).

9.7.2 Hyper-parameters

Table 6 lists the hyper-parameters for GAIL [2], GAfO [9], BCO [17], DAC [4], and our proposed approach \( \text{OPOLO} \). Specifically, for off-policy approaches, each self-generated interaction will be stored the replay buffer in a FIFO manner, and update frequency is the number of interactions sampled from the MDP after which the module is updated. Moreover, considering the different scales for the gradients of \( J(\theta, Q_\phi) \) and \( J_{\text{Reg}}(\theta) \) in Algorithm 1, we apply a coefficient \( \lambda \) for \( \text{OPOLO} \) to adjust the regularization strength when calculating the total policy loss:

\[
\theta \leftarrow \theta + \alpha (J_{\theta \phi}(\pi_\theta, Q_\phi) + \lambda J_{\theta \phi} J_{\text{Reg}}(\pi_\theta)).
\]

| Hyper-parameters                                    | Value                           |
|-----------------------------------------------------|---------------------------------|
| **Shared Parameters for Off-Policy Approaches**     |                                 |
| Buffer size                                         | \( 10^7 \)                      |
| Batch size                                          | \( 100 \)                       |
| Learning rate                                       | \( 3 \times 10^{-4} \)          |
| Discount factor \( \gamma \)                       | \( 0.99 \)                      |
| Network architecture                                | MLP [400, 300]                  |
| \( Q, \pi \) update frequency / gradient steps     | \( 10^3/10^3 \)                 |
| \( D \) update frequency / gradient steps           | \( 500/10 \)                    |
| **Shared Parameters for On-Policy Approaches**      |                                 |
| Batch size                                          | \( 2048 \)                      |
| mini-Batch size                                     | \( 256 \)                       |
| Learning rate                                       | \( 3 \times 10^{-4} \)          |
| Discount factor \( \gamma \)                       | \( 0.99 \)                      |
| Network architecture                                | MLP [400, 300]                  |
| **BCO**                                             |                                 |
| \( P_I \) pre-train gradient steps                 | \( 10^4 \)                      |
| \( P_I \) update frequency / gradient steps        | \( 10^3/100 \)                  |
| **DAC**                                             |                                 |
| Number of extra absorbing states                    | \( 1 \)                         |
| **OPOLO**                                           |                                 |
| \( P_I \) update frequency / gradient steps        | \( 500/50 \)                    |
| \( P_I \) regularization coefficient \( \lambda \)  | \( 0.1 \)                       |

Table 6: Hyper-parameters for Different Algorithms
9.8 Challenges of DICE without Expert Actions

In this section, we analyze the principle of offline imitation learning using DICE [30, 33, 31] and the reason that impedes its direct application to a LfO setting.

In a LfO setting where expert actions are unavailable, the learning objective is to minimize the discrepancy of state-only distributions induced by the agent and the expert. Without loss of generality, we consider an arbitrary f-divergence $D_f$ as the discrepancy measure:

$$
\max_{\pi} -D_f[\mu^\pi(s, s') | | \mu^E(s, s')] = \max_{\pi} \min_{x: \mathcal{S} \times \mathcal{S} \to \mathbb{R}} E_{\mu^\pi(s, s')}[-x(s, s')] + E_{\mu^E(s, s')}[f^*(x(s, s'))],
$$

in which $f^*(x)$ is the conjugate of $f(x)$ for the $f$-divergence. To remove the on-policy dependence of $\mu^\pi(s, s')$, we follow the rationale of DICE and use a similar change-of-variable trick mentioned in Sec 3.2 to learn a value function $v(s, s')$:

$$
v(s, s') := -x(s, s') + \gamma E_{a' \sim \pi(.|s'), s'' \sim P(.|s', a')}[v(s', s'')] = -x(s, s') + B^\pi v(s, s').
$$

This value function is a fixed point solution to a variant Bellman operator $B^\pi$, which, however, is problematic in a model-free setting. To see this, we substitute $x(s, s')$ by $(B^\pi v - v)(s, s')$ to transform Eq (14) into the following:

$$
\max_{\pi} \min_{x: \mathcal{S} \times \mathcal{S} \to \mathbb{R}} E_{\mu^\pi(s, s')}[-x(s, s')] + E_{\mu^E(s, s')}[f^*(x(s, s'))] = \max_{\pi} \min_{v: \mathcal{S} \times \mathcal{S} \to \mathbb{R}} \left( 1 - \gamma \right) E_{s_0 \sim P_0, s_1 \sim P(.|s_0, \pi(s_0))}[v(s_0, s_1)] + E_{\mu^E(s, s')}[f^*(B^\pi v - v)(s, s')].
$$

where $B^\pi v(s, s') = \gamma E_{a' \sim \pi(.|s'), s'' \sim P(.|s', a')}[v(s', s'')]$. Optimizing this objective is troublesome, in that the $B^\pi v(s, s')$ in term 2 requires knowledge of $P(\cdot|s, \pi(s))$, $\forall s \sim \mu^E(s)$. In another word, for any state sampled from the expert distribution, we need to know what would be the next state if following policy $\pi$ from this state. A similar issue is echoed in term 1, where $s_1$ is sampled from $P(\cdot|s_0, \pi(s_0))$. Consequently, directly applying DICE loses its advantage in a LfO setting, as it incurs a dependence on a forward transition model, which is costly to estimate and may counteract the efficiency brought by off-policy learning.