MAHALO: Unifying Offline Reinforcement Learning and Imitation Learning from Observations

Anqi Li [1] **Byron Boots** [1] **Ching-An Cheng** [2]

Abstract

We study a new paradigm for sequential decision making, called offline policy learning from observations (PLfO). Offline PLfO aims to learn policies using datasets with substandard qualities: *1)* only a subset of trajectories is labeled with rewards, *2)* labeled trajectories may not contain actions, *3)* labeled trajectories may not be of high quality, and *4)* the data may not have full coverage. Such imperfection is common in real-world learning scenarios, and offline PLfO encompasses many existing offline learning setups, including offline imitation learning (IL), offline IL from observations (ILfO), and offline reinforcement learning (RL). In this work, we present a generic approach to offline PLfO, called Modality-agnostic Adversarial Hypothesis Adaptation for Learning from Observations (MAHALO). Built upon the pessimism concept in offline RL, MAHALO optimizes the policy using a performance lower bound that accounts for uncertainty due to the dataset's insufficient coverage. We implement this idea by adversarially training data-consistent critic and reward functions, which forces the learned policy to be robust to data deficiency. We show that MAHALO consistently outperforms or matches specialized algorithms across a variety of offline PLfO tasks in theory and experiments. Our code is available at https://github.com/AnqiLi/mahalo.

1. Introduction

Online reinforcement learning (RL) has shown great promise in solving simulated tasks (Silver et al., 2016; Mnih et al., 2015). However, exploratory interactions with the environment, which are central to online RL, often can not be afforded in risk-sensitive applications, such as robotics (Ibarz et al., 2021) and healthcare (Gottesman et al., 2018). In these domains, it is more practical to consider an offline setting (Levine et al., 2020), where data is collected by behavioral policies satisfying certain criteria.

There are two main approaches to solving decision making problems offline: offline imitation learning (IL) (Chang et al., 2021; Kidambi et al., 2021; Kim et al., 2021) and offline RL (Fujimoto et al., 2019; Levine et al., 2020). Offline IL generally does not assume access to the reward. Theses approaches learn with a small set of expert demonstrations and potentially a separate dynamics dataset with unknown quality. Offline IL seeks to mimic expert behavior while avoiding distribution shift caused by using offline datasets. Offline imitation learning from observations (ILfO) (Kidambi et al., 2021; Ma et al., 2022) further relaxes the requirements of expert actions. ILfO allows learning from experts with different action spaces (Edwards et al., 2020), or when the expert has a different action modality or a embodiment (Cao & Sadigh, 2021; Radosavovic et al., 2021).

Offline RL, on the other hand, does not require expert-level demonstrations. It instead assumes that each transition in the offline dataset is labeled with reward. The goal of offline RL is to learn a policy which *1)* always improves upon the behavioral policy (Fujimoto et al., 2019; Laroche et al., 2019), and *2)* can outperform any other policies whose state-action distribution is covered by data (Xie et al., 2021).

However, in real-world applications, it is expensive to either acquire expert-level demonstrations (even if they are observation-only), or label every transition with reward. In this paper, we propose a more general and realistic formulation called offline *Policy Learning from Observations* (PLfO). Our goal is to learn from datasets where *1)* a subset of trajectories is labeled with rewards, *2)* labeled trajectories may not contain actions, *3)* labeled trajectories may not be of high quality, and *4)* the overall data may not have full coverage. The flexibility of this formulation allows us to directly take advantage of more data sources, such as dynamics data collected for other tasks and reward data collected by a non-expert agent with a different action space.

Offline PLfO considers two offline datasets: the reward dataset $\mathcal{D}_R = \{(s, r, s')\}$ and dynamics dataset $\mathcal{D}_\mathcal{A} = \{(s, a, s')\}$, where the dynamics dataset is consistent with

[1]University of Washington [2]Microsoft Research. Correspondence to: Anqi Li <anqil4@cs.washington.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Table 1. Different problem formulations for sequential decision making on offline datasets. Our PLfO formulation is the most general and can leverage the broadest range of data, which makes it the most realistic. The other formulations can be reduced to PLfO with additional restrictions on data. * denotes that data can only be used partially, with either action or reward removed.

	(s, a, r, s')	(s, a, s')	Expert (s, a, s')	Expert (s, s')	Non-expert (s, r, s')
Offline IL	✗*	✓	✓	✗	✗
Offline ILfO	✗*	✓	✗*	✓	✗
Offline RL	✓	✗	✗	✗	✗
Offline RL w/ Unlabeled Data	✓	✓	✗	✗	✗
Offline PLfO (Proposed)	✓	✓	✓	✓	✓

the Markovian dynamics that the learner aims to solve (i.e. it is collected by agents that have the same embodiment as the learner). In offline RL setting, the reward and dynamics datasets are aligned, since they are from the same underlying dataset $\mathcal{D} = \{(s, a, r, s')\}$. Recent work (Yu et al., 2022) relaxes this requirement by assuming that only a subset of transitions are labeled with rewards, i.e., the set of state transitions contained in $\mathcal{D}_R$ is a subset of $\mathcal{D}_\mathcal{A}$. On the contrary, in offline PLfO, we make no assumption on how these two datasets are related to each other.

Offline ILfO can also be viewed as a special case of offline PLfO. Although ILfO does not assume knowledge of reward, it makes an implicit assumption that expert trajectories attain high returns, while making no assumptions on reward information elsewhere (e.g., on the dynamics data $\mathcal{D}_A$). In other words, from the perspective of offline PLfO, expert demonstrations essentially act as the reward-labeled dataset $\mathcal{D}_R$ for ILfO. Practically, we can simply label the expert demonstrations with the maximum reward. This observation is in line with existing work (Fu et al., 2018a; Eysenbach et al., 2021; Smith et al., 2023). We refer readers to Table 1 for a summary of comparison between offline PLfO and existing formulations. In Appendix A we provide a more comprehensive literature review.

The key challenge to offline PLfO is the mismatch among the reward dataset, the dynamics dataset, and the test-time distribution. We present a generic approach to offline PLfO, called Modality-agnoistic Adversarial Hypothesis Adaptation for Learning from Observations (MAHALO). Built upon the concept of pessimism from offline RL literature (Jin et al., 2021; Liu et al., 2020; Kumar et al., 2020; Xie et al., 2021; Cheng et al., 2022), MAHALO optimizes for a performance lower bound accounting for insufficient data coverage on reward and dynamics. It can be realized by modifying existing offline RL algorithms based on adversarial training, such as (Xie et al., 2021; Cheng et al., 2022; Uehara & Sun, 2022; Rigter et al., 2022; Xie et al., 2022). In particular, we present a model-free instantiation of MAHALO built upon ATAC (Cheng et al., 2022), an offline RL algorithm based on a Stackelberg game of relative pessimism. In MAHALO, we consider the actor policy as the leader in the Stackelberg game, and adversarially train critic and reward functions so that they are data-consistent and can detect potential deficiency of the actor policy. As a result, the policy can be robust to the missing data coverage.

The contribution of this paper is two-fold. First, we propose offline PLfO, a novel formulation which relaxes data assumption for policy learning with offline data. This general formulation encompasses most existing offline formulations, including, but not limited to, offline IL, ILfO, RL, and RL with unlabeled data. Second, we present MAHALO, a solution to offline PLfO based on pessimism. We further present a model-free realization of MAHALO. In theory and experiments, we show that MAHALO consistently outperforms or matches performance with more specialized algorithms across various offline PLfO scenarios and tasks.

2. Preliminaries

Markov Decision Process We consider RL in a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $\gamma \in [0, 1)$ is the discount factor, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability, where $\Delta(\cdot)$ denotes the space of probability distributions. We assume that the reward function R is defined on state transitions, i.e., $R : \mathcal{S} \times \mathcal{S} \to [0, R_{\max}]$, as we consider learning from observations. This state-transition reward function R induces an effective state-action reward function $\bar{R}(s, a) := \mathbb{E}_{s' \sim P(\cdot|s,a)}[R(s, s')]$, which is the expected state-transition reward under the transition probability P. We denote a Markovian policy as $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. The goal of RL is to find a policy which maximizes the expected discounted return $J(\pi) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where $r_t = R(s_t, s_{t+1})$ and the expectation is over the randomness of running policy π with transition probability P starting from an initial state distribution $d_0(s)$. For a policy π and any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we define the transition operator $\mathcal{P}^\pi$ as $(\mathcal{P}^\pi f)(s, a) := \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[f(s', \pi)]$, where $f(s', \pi) = \sum_{a'} \pi(a'|s') f(s', a')$. For a policy π, we define the average state-action occupancy measure $d^\pi(s, a) := (1 - \gamma)\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a)]$. We recall that $J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s,a \sim d^\pi} \mathbb{E}_{s' \sim P(s'|s,a)}[R(s, s')]$.

Offline RL Offline RL studies the problem of policy learning from a reward-labeled transition dataset $\mathcal{D} = \{(s, a, r, s')\}$. The goal of offline RL is to learn the best policy that can be explained by data, while not making assumptions on the data coverage quality. An offline RL algorithm

ideally is able to learn the optimal policy of the MDP $\mathcal{M}$, as long as the dataset covers the states and actions that the optimal policy would visit. Such robustness of offline RL to data coverage quality is commonly realized by pessimism, which reasons about the worst case for states and actions not covered by the offline data. Being pessimistic in the face of uncertainty naturally forces the agent to search for good policies within the data support. Typically, the pessimism is implemented via behavior regularization (Fujimoto & Gu, 2021; Wu et al., 2019), value penalty (Jin et al., 2021; Yu et al., 2020b; Kumar et al., 2020), or adversarial training via a two-player game (Cheng et al., 2022; Xie et al., 2021; Rigter et al., 2022; Uehara & Sun, 2022).

Offline IL Offline IL (such as behavior cloning) studies the problem of policy learning using only the transition dataset $\mathcal{D} = \{(s, a, s')\}$ without reward labels. The transition data is a union of near-optimal expert data $\mathcal{D}_E$ and (optionally) a separately collected data of unknown quality $\mathcal{D}_X$. Like offline RL, the data in offline IL does not have full coverage, and the principle of mimicking the expert data in IL also effectively encourages the learner to stay within the the data distribution. In fact, Cheng et al. (2022) show that offline IL can be viewed as an offline RL problem with the largest reward uncertainty: By running an offline RL algorithm that optimizes for the relative performance between the learner and the behavioral policy under data uncertainty, an IL mimicking behavior would naturally occur.

Stackelberg game A Stackelberg game is a sequential two-player game (Von Stackelberg, 2010) between a leader x and a follower y. In this game, the leader plays first and then the follower plays after seeing the leader's decision. The game can be written as a bilevel optimization problem: $\max_x f(x, y_x)$ s.t. $y_x \in \max_y g(x, y)$, where f and g are the objectives of the leader and the follower, respectively.

3. Offline PLfO: A Unified Formulation for Offline RL and IL from Observations

In this section, we first introduce the generic setup of offline policy learning from observations (PLfO). Then we discuss practical scenarios where the data cannot be fully leveraged in offline RL and IL setups but is within the PLfO setup.

3.1. Problem Formulation

In offline PLfO, we assume access to pre-collected offline data consisting of transitions. In contrast to typical offline RL, we allow our data to include transitions which contain *either* reward or action. In other words, in offline PLfO, we consider two datasets, a reward dataset $\mathcal{D}_R = \{(s, r, s')\}$ and a dynamics dataset $\mathcal{D}_\mathcal{A} = \{(s, a, s')\}$. We note that these two datasets may not necessarily have an intersection.

We assume that both datasets are compliant with the underlying MDP. For the dynamics dataset, we follow standard compliance assumption in offline RL literature (Jin et al., 2021): *1)* for any $(s, a, s') \in \mathcal{D}_R$, we have $s' \sim P(\cdot|s, a)$; and *2)* the state-action pairs (s, a) in $\mathcal{D}_\mathcal{A}$ are sampled from the discounted state-action occupancy d^μ of a behavioral policy $\mu : \mathcal{S} \to \Delta(\mathcal{A})$. We slightly abuse notation to use μ to also denote the discounted state-action occupancy d^μ, i.e., $\mu = d^\mu$. For the reward dataset, for any $(s, r, s') \in \mathcal{D}_R$, we assume that the reward function R is defined on the state transition (s, s') and $r = R(s, s')$. We do not make assumption on the underlying distribution of state transitions $(s, s') \sim \nu$, e.g., s' in (s, s') may not be sampled from $P(\cdot|s, a)$ for some action a. With a slight abuse of notation, we will also use ν to denote the underlying distribution of transition tuple containing reward (s, r, s'). Like offline RL, we do not assume the coverage of these datasets on states and actions. The goal of offline PLfO is to learn a policy π which obtains high expected discounted return $J(\pi)$ in MDP $\mathcal{M}$ while using reward and dynamics datasets of limited coverage.

3.2. Relation to Existing Formulations

Offline PLfO is a general formulation encompassing many existing problem setups. This means that an offline PLfO algorithm can solve any of the following problems, or the combination of them, via simple reductions.

Offline RL Offline RL can be reduced to offline PLfO where the reward dataset and dynamics dataset are generated from the same underlying offline dataset $\mathcal{D} = \{(s, a, r, s')\}$. The alignment assumption makes offline RL in general an easier problem than offline PLfO since there is no mismatch between reward and dynamics data.

Offline RL with unlabeled data Yu et al. (2022); Singh et al. (2020); Hu et al. (2023) consider a formulation where a subset of dynamics data is labeled with reward. In this scenario, the reward data is well-covered by dynamics data. The main challenge here is to leverage unlabeled dynamics data while not suffering from insufficient reward coverage.

Offline IL Offline IL uses a dataset of expert demonstrations $\mathcal{D}_E$. Although IL generally assumes no reward information, it makes an implicit assumption on expert performance. In other words, IL is a learning problem where only positive (i.e., high return) examples are given. This observation is also in line with existing work which takes a density matching perspective on IL (Ho & Ermon, 2016; Kim et al., 2021) and reward learning from demonstrations (Fu et al., 2018a; Eysenbach et al., 2021). Recent work such as (Kim et al., 2021; Chang et al., 2021; Smith et al., 2023) has considered offline IL with a separately-collected dynamics dataset $\mathcal{D}_X$ of unknown quality.

Offline IL can be reduced to offline PLfO: The reward dataset $\mathcal{D}_R = \{(s, R_{\max}, s')\}$ comes from expert demonstrations, where $R_{\max}$ is the maximum reward. The dynamics dataset contains both the expert demonstrations $\mathcal{D}_E$ and,

if given, the separately-collected dynamics data $\mathcal{D}_X$.

Offline ILfO Offline ILfO is similar to offline IL, except that the expert demonstrations only contain state transitions, i.e., $\mathcal{D}_E = \{(s_E, s'_E)\}$. As such, offline ILfO can be viewed as offline PLfO with reward dataset $\mathcal{D}_R = \{(s_E, R_{\max}, s'_E)\}$ and dynamics dataset $\mathcal{D}_\mathcal{A} = \mathcal{D}_X$. Compared to the previous setups, offline ILfO faces an additional challenge of insufficient action coverage, as the expert state transitions (s_E, s'_E) may not be in the dynamics dataset $\mathcal{D}_\mathcal{A}$.

3.3. Practical Scenarios

We now consider a few practical examples of reward and dynamics data sources. As we will see, it is likely that the coverage of reward data mismatches with that of dynamics data. In such scenario, offline PLfO formulation is well-suited as it can leverage *all* available data.

Sources of dynamics data Since dynamics data is task-agnostic, dynamics data can be obtained from running data-collection policies on the MDP, as well as any other MDPs with the same dynamics. For example, in a multi-task setting (Yu et al., 2020a), dynamics data can be acquired when solving different task rewards. However, it is expensive to label dynamics data with rewards for *every* task. Realistically, data often only has reward information of the particular task that the data collection is for, which means that some state transitions only have actions, but not rewards.

Sources of reward data One practical strategy to reward labeling is to label randomly sampled trajectories from a pre-collected dynamics dataset (Yu et al., 2022). This means that a large subset of the dynamics data remains unlabeled. Another setting is to re-use reward data collected by another agent (with the same state space). It is possible that the action information is unavailable or unusable in the underlying MDP. For example, the other agent can have different embodiment or use a different control modality. Additionally, reward information can be implicitly provided through expert demonstrations, as is discussed in Section 3.2.

4. MAHALO

We propose a generic approach to offline PLfO, called Modality-agnostic Adversarial Hypothesis Adaptation for Learning from Observations (MAHALO). It is inspired by the idea of pessimism in offline RL (Jin et al., 2021).

4.1. Solution Concept to Offline PLfO

In order to tackle the heterogeneous uncertainty in offline PLfO, we leverage the concept of version space in the offline RL literature (Cheng et al., 2022; Xie et al., 2021; Rigter et al., 2022; Uehara & Sun, 2022; Xie et al., 2022) to construct a performance lower bound for optimizing policies in MAHALO. Here, a version space, denoted as $\mathcal{J} = \{\hat{J} : \Pi \to \mathbb{R}\}$, is the space of policy performance hypotheses that remain feasible after observing data in $\mathcal{D}_\mathcal{A}$ and $\mathcal{D}_R$. For example, if a bipedal robot has experienced that falling down receives zero rewards, then any hypothesis in the version space would give a zero reward to any policy that makes the robot fall, but the hypotheses in the version may disagree on which reward to give for other behaviors.

In MAHALO we use the version space $\mathcal{J}$ to encapsulate uncertainty due to heterogeneous missing coverage. Because the version space $\mathcal{J}$ by definition includes the true performance function J, we can use $\mathcal{J}$ to construct a policy performance lower bound. For example, for a policy π we can compute its absolute performance lower bound naturally as $\min_{\hat{J}\in\mathcal{J}} \hat{J}(\pi)$. Thus we can optimize policies through solving a saddle-point problem: $\max_{\pi\in\Pi} \min_{\hat{J}\in\mathcal{J}} \hat{J}(\pi)$ which provides a way to systematically optimize policy performance accounting for missing information in data.

For the case where the data are fully labeled with rewards and actions, offline RL literature has proposed several designs of the version space $\mathcal{J}$: with MDP models or value functions, in conjunction with absolute or relative pessimism (Cheng et al., 2022; Xie et al., 2022; 2021; Rigter et al., 2022; Uehara & Sun, 2022). Here we generalize this technique to offline PLfO. In particular, we will design model-free versions of MAHALO, as model-free methods are simpler to implement, use less hyperparameters, and have demonstrated superior empirical performance in offline RL (Yu et al., 2021). In principle, MAHALO can be realized by *any* version-space offline RL algorithm, including those based on models.

4.2. Model-Free Realization of MAHALO

We now present a model-free realization of MAHALO based on the concept of relative pessimism in offline RL (Cheng et al., 2022). For clarity, we first present the formulation at the population level. We will later provide theoretical analysis for the finite-sample scenario in Section 4.2.1. We introduce and analyze another realization of MAHALO based on absolute pessimism (Xie et al., 2021) in Appendix D.

We formulate offline PLfO as a Stackelberg game, with the actor policy $\pi \in \Pi$ as the leader. The followers consist of critic $f \in \mathcal{F}$ and reward function $g \in \mathcal{G}$.

$$\hat{\pi} \in \arg\max_{\pi\in\Pi} \mathcal{L}_\mu(\pi, f^\pi) \tag{1}$$

$$\text{s.t.} \quad f^\pi \in \arg\min_{f\in\mathcal{F}, g\in\mathcal{G}} \mathcal{L}_\mu(\pi, f) + \alpha\mathcal{E}_\nu(g) + \beta\mathcal{E}_\mu(\pi, f, g),$$

with $\alpha \geq 0, \beta \geq 0$ being hyperparameters, and

$$\mathcal{L}_\mu(\pi, f) := \mathbb{E}_\mu\big[f(s,\pi) - f(s,a)\big], \tag{2}$$

$$\mathcal{E}_\nu(g) := \mathbb{E}_\nu\big[\big(g(s,s') - r\big)^2\big], \tag{3}$$

$$\mathcal{E}_\mu(\pi, f, g) := \mathbb{E}_\mu\big[\big((f - \bar{g} - \mathcal{P}^\pi f)(s,a)\big)^2\big]. \tag{4}$$

where $f(s,\pi) := \mathbb{E}_{a\sim\pi(\cdot|s)}[f(s,a)]$ and $\bar{g}(s,a) := \mathbb{E}_{s'\sim P(\cdot|s,a)}[g(s,s')]$. This optimization problem can be viewed as a regularized version of the constrained problem with a version space $\mathcal{J} = \{\hat{J} : \hat{J}(\pi) = J(\mu) + \frac{1}{1-\gamma}\mathcal{L}_\mu(\pi,f), \mathcal{E}_\nu(g) \le \epsilon_\alpha, \mathcal{E}_\mu(\pi,f,g) \le \epsilon_\beta\}$ for some $\epsilon_\alpha, \epsilon_\beta \ge 0$ related to α, β.[1] We adopt the regularized version, as it is easier to implement numerically.

To understand the above formulation, let us start by considering the last two terms in the followers' objective function. Recall that ν is the underlying distribution of reward data, so $\mathcal{E}_\nu(g)$ measures whether the candidate reward function g is data consistent. $\mathcal{E}_\mu(\pi,f,g)$ quantifies whether the candidate critic f is Bellman-consistent on the dynamics data for policy π and reward function g. Therefore, with sufficiently large α and β, the reward g^π and critic f^π are both (approximately) consistent with the reward and dynamics data. On the other hand, $\mathcal{L}_\mu(\pi,f)$ is the relative performance of between the candidate policy π and behavioral policy μ, with value estimated by critic f. The followers minimizes this quantity, meaning that f^π provides a *pessimistic* relative evaluation of policy π (which will be formally shown in Proposition 4.4). The learned actor policy $\hat{\pi}$ maximizes this lower bound to improve over the the behavioral policy.

We would like to stress on the importance of making the reward function additionally minimize for the Bellman error in the followers' objective (1). By minimizing (4), the reward and critic functions work *together* to provide a pessimistic evaluation of the policy. In other words, the Bellman error (4) connects the learned reward function to the pessimistic loss (2), since the reward function can change in a way to make the critic more pessimistic.

The Stackelberg game for MAHALO in (1) is similar to ATAC (Cheng et al., 2022). The main difference is that ATAC uses the observed reward in the Bellman error term as it assumes the reward is available for every transition. MAHALO, on the other hand, trains the reward function to be consistent with the reward data (3) and Bellman-consistent with a pessimistic critic function (4) on the dynamics data.

Below we discuss three desirable properties of MAHALO. First, given large dynamics and reward datasets, MAHALO can outperform any policy whose state-action distribution is well-covered by both datasets. Second, MAHALO ensures safe policy improvement (Fujimoto et al., 2019; Laroche et al., 2019), i.e., the learned policy $\hat{\pi}$ is no worse than the behavioral policy μ given sufficient data. Third, MAHALO can automatically adapt to the structure within data. When applied to more restrictive formulations such as offline RL, offline IL, and offline ILfO, MAHALO shows similar behavior as specialized algorithms.

[1]This analogy between the constrained and the regularized versions can be derived following the principle in (Xie et al., 2021).

4.2.1. Theoretical Properties

We analyze the solution to a finite-sample version of (1) based on dynamics dataset $\mathcal{D}_\mathcal{A}$ and reward dataset $\mathcal{D}_R$:

$$\hat{\pi} \in \arg\max_{\pi\in\Pi} \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi) \tag{5}$$
$$\text{s.t.} \quad f^\pi \in \arg\min_{f\in\mathcal{F},g\in\mathcal{G}} \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi,f) + \alpha\mathcal{E}_{\mathcal{D}_R}(g) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi,f,g),$$

where $\mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi,f)$ and $\mathcal{E}_{\mathcal{D}_R}(g)$ are the empirical estimates of $\mathcal{L}_\mu(\pi,f)$ and $\mathcal{E}_\nu(g)$, respectively, and

$$\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi,f,g) := \mathbb{E}_{\mathcal{D}_\mathcal{A}}\left[\left(f(s,a) - g(s,s') - \gamma f(s',\pi)\right)^2\right] - \min_{f'\in\mathcal{F}} \mathbb{E}_{\mathcal{D}_\mathcal{A}}\left[\left(f'(s,a) - g(s,s') - \gamma f(s',\pi)\right)^2\right]. \tag{6}$$

The quantity $\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi,f,g)$ is the estimated Bellman error (Antos et al., 2008). We show in Appendix C (Lemma C.2) that $\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi,f,g)$ can be used to approximate the Bellman error $\mathcal{E}_\mu(\pi,f,g)$ defined in (4).

For clarity purposes, we make a perfect realizability and completeness assumption below. Our analysis can be easily modified to consider an approximate version.

Assumption 4.1 (Realizability & Completeness)**.** We assume $\mu \in \Pi$, $R \in \mathcal{G}$, and for all $\pi \in \Pi$, $Q^\pi \in \mathcal{F}$. In addition, $\inf_{f'\in\mathcal{F}} \|f' - \bar{g} - \mathcal{P}^\pi f\|_\mu = 0, \forall g \in \mathcal{G}, f \in \mathcal{F}$.

Due to the nature of offline learning, we will compare the performance of the learned policy $\hat{\pi}$ with a policy π whose induced state-action occupancy d^π is "well-covered" by data distributions. Similar to (Xie et al., 2021; Cheng et al., 2022; Uehara & Sun, 2022), we define error transfer coefficients to measure distribution shift from the data distributions based on critic class $\mathcal{F}$, and reward class $\mathcal{G}$. This is a weaker notion than, e.g., density ratio (Munos & Szepesvári, 2008).

Definition 4.2 (Error transfer coefficients)**.** The Bellman error transfer coefficient between $\rho \in \Delta(\mathcal{S}\times\mathcal{A})$ and $\mu \in \Delta(\mathcal{S}\times\mathcal{A})$ under policy π, critic class $\mathcal{F}$ and reward class $\mathcal{G}$ is defined as

$$\mathcal{C}(\rho;\mu,\mathcal{F},\mathcal{G},\pi) := \sup_{f\in\mathcal{F},g\in\mathcal{G}} \frac{\|f - \bar{g} - \mathcal{P}^\pi f\|_{2,\rho}^2}{\|f - \bar{g} - \mathcal{P}^\pi f\|_{2,\mu}^2}. \tag{7}$$

Similarly, the reward error transfer coefficient between ρ and $\nu \in \Delta(\mathcal{S}\times\mathcal{S})$ under reward class $\mathcal{G}$ is defined as

$$\mathcal{C}(\rho;\nu,\mathcal{G}) := \sup_{g\in\mathcal{G}} \frac{\|\bar{g} - \bar{R}\|_{2,\rho}^2}{\|g - R\|_{2,\nu}^2}. \tag{8}$$

We use $d_{\mathcal{F},\mathcal{G},\Pi}$ to denote the joint statistical complexity of critic class $\mathcal{F}$, reward class $\mathcal{G}$ and policy class Π, and use $d_\mathcal{G}$ to denote the statistical complexity of reward class $\mathcal{G}$ (e.g., when $\mathcal{F}$, $\mathcal{G}$ and Π are all finite, we have $d_{\mathcal{F},\mathcal{G},\Pi} = \mathcal{O}(\log {}^{|\mathcal{F}||\mathcal{G}||\Pi|}/_\delta)$ where δ is the failure probability). In Appendix C, we establish statistical complexity using covering number. We now state the main theoretical property of MAHALO of relative pessimism in (5).

Theorem 4.3. *Under Assumption 4.1, let $\hat{\pi}$ be the solution to (5) and let $\pi \in \Pi$ be any comparator policy. Let $C_1 \geq 1, C_2 \geq 1$ be constants, $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$ be a distribution that satisfy $\mathcal{C}(\rho; \mu, \mathcal{F}, \mathcal{G}, \pi) \leq C_1$ and $\mathcal{C}(\rho; \nu, \mathcal{G}) \leq C_2$. Define $\epsilon_\mu := V_{max}^2 d_{\mathcal{F},\mathcal{G},\Pi}/|\mathcal{D}_\mathcal{A}|$, $\epsilon_\nu := R_{max}^2 d_\mathcal{G}/|\mathcal{D}_R|$ and $\epsilon := (\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu})^2$. Choosing $\alpha = \Theta\left(V_{max}^{1/3}\epsilon^{1/3}/\epsilon_\nu\right)$ and $\beta = \Theta\left(V_{max}^{1/3}\epsilon^{1/3}/\epsilon_\mu\right)$, with high probability,*

$$
\begin{aligned}
& J(\pi) - J(\hat{\pi}) \qquad (9) \\
\leq & \mathcal{O}\left(\frac{1}{1-\gamma}\left(\frac{C_1^{1/3} V_{max}(d_{\mathcal{F},\mathcal{G},\Pi})^{1/3}}{|\mathcal{D}_\mathcal{A}|^{1/3}} + \frac{C_2^{1/3} R_{max}(d_\mathcal{G})^{1/3}}{|\mathcal{D}_R|^{1/3}}\right)\right) \\
& + \underbrace{\frac{\langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi \rangle}{1-\gamma}}_{\text{off-support error (dynamics)}} + \underbrace{\frac{\langle (d^\pi \ominus \mu) \setminus \rho, |\bar{R} - \bar{g}^\pi| \rangle}{1-\gamma}}_{\text{off-support error (reward)}},
\end{aligned}
$$

where $(d^\pi \ominus \mu) := d^\pi \setminus \mu + \mu \setminus d^\pi$ with $(d_1 \setminus d_2)(s,a) := \max(d_1(s,a) - d_2(s,a), 0)$ and $\langle \iota, f \rangle := \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \iota(s,a) f(s,a)$.

The first term in (9) is the statistical error, which vanishes as $|\mathcal{D}_\mathcal{A}|, |\mathcal{D}_R| \to \infty$. The second and third terms measure how much the comparator policy π is outside of the support of dynamics and reward data distributions. In other words, our learned policy $\hat{\pi}$ can compete with any policy π that is well-supported by both dynamics and reward data. Compared with Theorem 5 of ATAC in (Cheng et al., 2022), we have an extra term about the statistical error of the estimated reward, and an off-support reward error, because we do not assume access to rewards on all transitions.

Despite using partial reward labels, MAHALO has a robust policy improvement property, similar to ATAC (Cheng et al., 2022), which guarantees that, for a known range of hyperparameters, the learned policy $\hat{\pi}$ is no worse than the behavioral policy by more than statistical errors.

Proposition 4.4 (Robust Policy Improvement)**.** *Under Assumption 4.1, let $\hat{\pi}$ be the solution to (5). Let ϵ_μ and ϵ_ν be as defined in Theorem 4.3. We have, for any fixed $\alpha \geq 0$ and $\beta \geq 0$, with high probability,*

$$
\begin{aligned}
& J(\mu) - J(\hat{\pi}) \qquad (10) \\
\leq & \mathcal{O}\left(\frac{V_{max}}{1-\gamma}\sqrt{\frac{d_{\mathcal{F},\mathcal{G},\Pi}}{|\mathcal{D}_\mathcal{A}|}} + \frac{\alpha R_{max}^2 d_\mathcal{G}}{(1-\gamma)|\mathcal{D}_R|} + \frac{\beta V_{max}^2 d_{\mathcal{F},\mathcal{G},\Pi}}{(1-\gamma)|\mathcal{D}_\mathcal{A}|}\right).
\end{aligned}
$$

As $|\mathcal{D}_\mathcal{A}|$ and $|\mathcal{D}_R| \to \infty$, we can see that the solution actor policy $\hat{\pi}$ is guaranteed to be no worse than the behavioral policy μ with *any* choices of fixed $\alpha, \beta \geq 0$.

4.2.2. Adaption to Structure within Data

Since MAHALO can be used to solve offline PLfO, MAHALO can be used to solve more restrictive problems via simple reductions. Then, a natural question is: Can MAHALO achieve similar performance as specialized algorithms? The answer is yes. Below we show that this is because MAHALO can adapt to the hidden structure within data despite being agnostic to the relationship between dynamics and reward datasets.

Offline RL In offline RL, since the reward and dynamics datasets are aligned, we have $\mathcal{E}_\nu(g) = \mathbb{E}_\mu[\mathbb{E}_{s'\sim P(\cdot|s,a)}[(g(s,s') - R(s,s'))^2]]$. The expected Bellman error of critic f (with the true reward R), $\mathcal{E}_\mu(\pi, f, R)$, can be upper bounded by $2\mathcal{E}_\nu(g) + 2\mathcal{E}_\mu(\pi, f, g)$. With sufficiently large α and β, for any actor policy π, its corresponding critic f^π is an approximately Bellman-consistent critic function. Therefore, MAHALO behaves similarly as ATAC (Cheng et al., 2022). This can also be seen from Theorem 4.3. Since $|\mathcal{D}_\mathcal{A}| = |\mathcal{D}_R|$, the statistical error is dominated by the first term, which is the same as the statistical error as ATAC. In offline RL, good dynamics coverage implies good data coverage, i.e., we have $(d^\pi \ominus \mu) \setminus \rho \leq d^\pi \setminus \rho$. This gives us a similar off-support error term.

Offline RL with unlabeled data For the sake of simplicity, we consider a tabular setting. In this case, regardless of the policy $\pi \in \Pi$, with sufficiently large α, the reward function g^π such that $g^\pi(s,s') \approx R(s,s')$ when (s,s') is within coverage of reward data ν; the value of $g^\pi(s,s')$ on other states would adapt pessimistically according to the learner policy. The critic f^π is effectively conducting a pessimistic policy evaluation for π in such a reward function. This means that MAHALO in the tabular setting has a similar behavior as UDS (Yu et al., 2022), a strategy where zero reward is given to *all* unlabeled data. The difference, though, is that MAHALO still assigns an accurate reward to unlabeled dynamics transitions (s,a,s') when (s,s') is within coverage of ν. This implies that MAHALO induces less bias than UDS, even though UDS knows more about the underlying data generation process.

Offline IL and ILfO In the simplest offline IL setting where only the set of expert demonstrations $\mathcal{D}_E$ is given, Proposition 4.4 (with $\pi_E = \mu$) shows that the learned policy $\hat{\pi}$ is no worse than the expert policy π_E up to statistical errors. Now consider the scenario where we have access the expert demonstrations $\mathcal{D}_E$ (which may or may not have actions)[2] and a separately collected dynamics dataset $\mathcal{D}_X$ with unknown quality. Theorem 4.3 (with $r(s,s') = R_{\max}\mathbb{1}[(s,s') \in \text{supp}(\nu)]$) shows that the learned policy would stay within the support of the expert distribution, similar to (Wang et al., 2019; Smith et al., 2023).

Summary MAHALO is an general, data-agnostic algorithm for solving offline PLfO problems. When applied to more restrictive settings when data presents additional structure, MAHALO can automatically adapt to such structure, and achieves behavior on par with existing specialized algorithms. This means that MAHALO can leverage broad

[2]When $\mathcal{D}_E$ contains action, we can alternatively replace $\mathcal{L}_{\mathcal{D}_A}(\pi, f)$ with $\mathcal{L}_{\mathcal{D}_E}(\pi, f)$, which would ensure robust policy improvement to the expert policy.

Algorithm 1 MAHALO (realized by ATAC)

1: **Input:** Batch datasets $\mathcal{D}_R$, $\mathcal{D}_\mathcal{A}$; policy π, critics f_1, f_2; coefficients α, $\beta \geq 0$ and $\tau, w \in [0, 1]$.
2: Initialize target networks $\bar{f}_1 \leftarrow f_1$, $\bar{f}_2 \leftarrow f_2$.
3: **for** $k = 1, 2, \ldots, K$ **do**
4: Sample minibatches $\mathcal{D}_R^{\text{mini}}$ and $\mathcal{D}_\mathcal{A}^{\text{mini}}$ from $\mathcal{D}_R$ and $\mathcal{D}_\mathcal{A}$.
5: Compute critic loss $l_{\text{critic}}(f_i) \leftarrow \mathcal{L}_{D_\mathcal{A}^{\text{mini}}}(\pi, f_i) + \beta \mathcal{E}^w_{D_\mathcal{A}^{\text{mini}}}(\pi, f_i, g)$, for $i \in \{1, 2\}$
6: Compute reward loss $l_{\text{reward}}(g) \leftarrow \alpha \mathcal{E}_{\mathcal{D}_R^{\text{mini}}}(g) + \beta \sum_{i=\{1,2\}} \mathcal{E}^w_{D_\mathcal{A}^{\text{mini}}}(\pi, f_i, g)$.
7: Update critic network $f_i \leftarrow \text{Proj}_\mathcal{F}(f_i - \eta_{\text{fast}} \nabla l_{\text{critic}})$ for $i \in \{1, 2\}$.
8: Update reward network $g \leftarrow \text{Proj}_\mathcal{G}(g - \eta_{\text{fast}} \nabla l_{\text{reward}})$.
9: Compute actor loss $l_{\text{actor}} \leftarrow -\mathcal{L}_{\mathcal{D}_\mathcal{A}^{\text{mini}}}(\pi, f_1)$.
10: Update actor network $\pi \leftarrow \text{Proj}_\Pi(\pi - \eta_{\text{slow}} \nabla l_{\text{actor}})$
11: Update target $\bar{f} \leftarrow (1 - \tau)\bar{f} + \tau f$ for $(f, \bar{f}) \in \{(f_i, \bar{f}_i)\}_{i=1,2}$.
12: **end for**

sources of data and no special care, e.g. data alignment or management, needs to be taken during data collection.

4.3. Implementation

The MAHALO realization above can be implemented by making a few simple modifications to ATAC (Cheng et al., 2022). The resulting algorithm is presented in Algorithm 1, with the modifications marked in magenta. This implementation of MAHALO is based on a reduction of two-player game to no-regret policy optimization (Cheng et al., 2022). We use ADAM (Kingma & Ba, 2015) optimizer with a faster learning rate η_{fast} for the critic and reward functions, and a smaller learning rate η_{slow} for the actor.

4.3.1. Actor and Critic Update

We use a strategy for updating the critic similar to ATAC. ATAC uses a surrogate to the Bellman error term in Equation (4) called double Q residual algorithm (DQRA) loss, which combines double Q heuristics (Fujimoto et al., 2018), the residual algorithm (Baird, 1995), and target networks (Mnih et al., 2015). The critic is parameterized by two networks $\{f_1, f_2\}$, each with a delayed target $\{\bar{f}_1, \bar{f}_2\}$. The target value is computed by taking the minimum of the two targets $\bar{f}_{\min}(s, a) = \min_{i \in \{1,2\}} \bar{f}_i(s, a)$. DQRA uses a convex combination of the temporal difference (TD) losses of the critic network and target networks to stabilize learning. For $i \in \{1, 2\}$, the DQRA loss is defined as

$$\mathcal{E}^w_{\mathcal{D}_\mathcal{A}^{\text{mini}}}(\pi, f_i, g) \coloneqq (1 - w)\mathcal{E}^{\text{td}}_{\mathcal{D}_\mathcal{A}^{\text{mini}}}(\pi, f_i, f_i, g) + w\mathcal{E}^{\text{td}}_{\mathcal{D}_\mathcal{A}^{\text{mini}}}(\pi, f_i, \bar{f}_{\min}, g), \tag{11}$$

where $w \in [0, 1]$ is the weight and the TD loss is given by

$$\mathcal{E}^{\text{td}}_{\mathcal{D}_\mathcal{A}^{\text{mini}}}(\pi, f, f', g) \coloneqq \mathbb{E}_{\mathcal{D}_\mathcal{A}^{\text{mini}}}[(f(s, a) - g(s, s') - \gamma f'(s', \pi))^2].$$

Note that here we use predicted reward $g(s, s')$ since reward r is not observed in $\mathcal{D}_\mathcal{A}$. After each gradient update, we apply an ℓ_2 projection on critic network weights (not on bias terms) (line 7) as is done in ATAC (Cheng et al., 2022).

We update the actor using the same way as ATAC. The actor policy optimizes for a *single* critic f_1 and uses a Lagrangian relaxation of minimum entropy constraint similar to SAC (Haarnoja et al., 2018).

4.3.2. Reward Update

The major difference between Algorithm 1 and ATAC is the reward function update. We estimate the reward prediction loss empirically from minibatches sampled from the reward dataset $\mathcal{D}_R$: $\mathcal{E}_{\mathcal{D}_R^{\text{mini}}}(g) = \mathbb{E}_{\mathcal{D}_R^{\text{mini}}}[(g(s, s') - r)^2]$. The reward loss is the weighted sum of reward prediction loss $\mathcal{E}_{\mathcal{D}_R^{\text{mini}}}(g)$ and DQRA losses $\sum_{i=\{1,2\}} \mathcal{E}^w_{D_\mathcal{A}^{\text{mini}}}(\pi, f_i, g)$ (line 6). The DQRA losses connect the reward to the critic which minimizes also the performance difference; as a result, the learned reward function is also pessimistically estimated. In other words, the critic and the reward functions jointly form a hypothesis that adversarially adapts to the learner's policy.

5. Experiments

We aim to answer the following questions: (a) Is MAHALO effective in solving different instances of offline PLfO problems? (b) Can MAHALO achieve similar performance as other specialized algorithms? (c) Whether MAHALO can obtain comparable performance to oracle algorithms with full reward and dynamics information? (d) In what situation is the pessimistic reward function of MAHALO critical in achieving good performance?

Scenarios To answer question (a), we design five instances of offline PLfO inspired by practical scenarios. **ILfO:** The learner is presented with a relatively small set of expert-level state-only trajectories, and a dynamics dataset of mixed quality. The dynamics data contains trajectories collected by policies with different performance-level. **IL:** Similar to ILfO, but we additionally provide expert actions to the learner. **RLfO:** Similar to ILfO, but the learner is additionally given the reward along the expert trajectories. This simulates the scenario where we provide manual la-

Table 2. Results on D4RL benchmark (Fu et al., 2020). We show the average normalized score over 50 evaluation trials across 10 random seeds. (The standard errors are reported in Table 6). Algorithms with scores greater than 90% of the best score (excluding Oracle) are in bold. † ATAC only uses data with both dynamics and reward information. + Oracle has access to reward for all dynamics data.

Scenario	Dataset	MAHALO	RP	AP	UDS	ATAC†	BCO	BC	SMODICE	Oracle+
ILfO	hopper	**104.66**	**97.48**	45.97	-	-	46.80	-	67.72	-
	walker	**88.60**	77.15	61.11	-	-	63.02	-	1.52	-
	halfcheetah	**61.24**	36.00	4.87	-	-	5.16	-	**59.64**	-
IL	hopper	**104.06**	**97.88**	53.35	32.21	63.56	-	32.12	74.75	-
	walker	**89.03**	77.71	63.53	8.45	78.35	-	18.78	0.94	-
	halfcheetah	**54.99**	23.20	3.74	25.82	3.64	-	22.36	**58.40**	-
RLfO	hopper	**106.47**	**105.65**	47.01	-	-	-	-	-	103.39
	walker	**96.65**	**97.26**	63.30	-	-	-	-	-	98.52
	halfcheetah	50.38	**68.66**	3.35	-	-	-	-	-	63.57
RL-expert	hopper	87.73	**105.56**	51.54	**98.63**	65.45	-	-	-	103.39
	walker	**103.18**	**98.31**	56.27	72.97	66.40	-	-	-	98.52
	halfcheetah	48.43	**64.37**	3.47	13.68	3.41	-	-	-	63.57
RL-sample	hopper	**103.08**	**101.66**	71.92	0.95	71.50	-	-	-	103.34
	walker	**95.00**	**94.59**	5.94	0.00	0.48	-	-	-	95.73
	halfcheetah	**68.30**	**68.71**	13.26	20.36	19.38	-	-	-	69.91

Table 3. Five scenarios of PLfO considered in our experiments.

Scenarios	Mixed Quality Data	Expert
ILfO	state + action	state
IL	state + action	state + action
RLfO	state + action	state + reward
RL-expert	state + action	state + action + reward
RL-sample	state + action + (sampled trajs) reward	-

bels to the expert demonstrations. **RL-expert:** The learner is presented with the expert action in addition to what is given in RLfO. **RL-sample:** The learner is provided with the mixed quality dynamics data, with a subset of trajectories labeled with reward. The information available in each scenario is summarized in Table 3.

Baselines To address question (b), we consider a few specialized baseline algorithms for each scenario. We consider behavior cloning from observation (**BCO**) (Torabi et al., 2018) as a baseline for ILfO, and behavior cloning (**BC**) for IL. We also include **SMODICE** (Ma et al., 2022), a state-of-the art offline ILfO algorithm as a baseline algorithm for ILfO, and a variation of SMODICE which uses a state-action discriminator as a baseline for offline IL. Since IL, RL-expert and RL-sample can be viewed as RL with unlabeled data, we present two baselines for these settings: running an offline RL algorithm, we use **ATAC** (Cheng et al., 2022) since it is the closest to MAHALO, only on labeled data and **UDS** (Yu et al., 2022). We note that UDS requires knowing the common transitions between reward and dynamics datasets. We implement UDS with ATAC, which is slightly different than (Yu et al., 2022),where CQL (Kumar et al., 2020) is used. Since learning inverse dynamics is a common approach to learning with observations (Torabi et al., 2018; Edwards et al., 2020), we implement a baseline called action prediction (**AP**). It pretrains an inverse dynamics model on the dynamics dataset, predicts the missing actions in the reward dataset, and runs ATAC (Cheng et al., 2022) on the reward dataset. For question (d), we implement a baseline algorithm called reward prediction (**RP**). It pretrains a reward function on the reward dataset. This reward function is then fixed for offline RL training using ATAC (Cheng et al., 2022). Comparing MAHALO with RP can give us information on whether the adversarial training of reward function in MAHALO is effective. Finally, to answer question (c), we train ATAC on a fully-labeled dynamics dataset (**Oracle**) to evaluate if MAHALO can achieve comparable performance with a privileged offline RL algorithm (which has access to more information).

We evaluate MAHALO and above-mentioned algorithms on two sets of environments: locomotion tasks from the D4RL benchmark (Fu et al., 2020) and robot manipulation tasks from the Meta-World domain (Yu et al., 2020a).

5.1. Evaluation on D4RL

We consider three environments from D4RL (Fu et al., 2020): hopper-v2, walker2d-v2, and halfcheetah-v2. The rewards in the three environments promote moving forward. In hopper and walker2d, the agent is required to stay within a health height range, otherwise the episode terminates. We construct a mixed quality dynamics dataset with 3.4 M transitions through concatenating the random, medium, medium-replay, and full-replay datasets. The expert data is consisted of 10k transitions ($\sim$ 10 trajectories) generated by randomly sampling trajectories from the expert dataset. For RL-sample, we sample 34k transitions (1% of overall data).

The normalized return for five scenarios for each dataset is listed in Table 2. MAHALO achieves top performance in almost every task except halfcheetah. MAHALO also

Table 4. Success rate of the final policy (with the exception of SMODICE*) on Meta-World (Yu et al., 2020a). The success rate is computed over 50 evaluation episodes. We report the average success rate across 10 random seeds. (The standard errors across random seeds are reported in Table 7). We consider an episode success if it is able to reach the goal within 128 steps. * SMODICE often diverges during training; we therefore take the success rate of its best performing policy during training instead of the final one.

Scenario	Dataset	MAHALO	RP	AP	UDS	ATAC†	BCO	BC	SMODICE*	Oracle+
ILfO	reach	**65.0**	**62.6**	13.0	-	-	11.6	-	10.6	-
	push	**62.4**	11.6	12.4	-	-	14.6	-	0.4	-
	plate-slide	**100.0**	22.0	**94.0**	-	-	75.4	-	0.0	-
	handle-press	75.4	32.4	**96.6**	-	-	**87.8**	-	16.6	-
	button-press	**100.0**	**100.0**	**93.6**	-	-	**93.8**	-	0.4	-
IL	reach	24.6	23.6	38.0	21.6	**98.0**	-	62.2	19.8	-
	push	**92.6**	11.8	35.2	79.2	50.0	-	**91.4**	0.2	-
	plate-slide	80.4	34.4	**89.8**	76.2	**89.4**	-	**85.3**	0.0	-
	handle-press	71.4	34.8	**100.0**	35.2	**97.0**	-	75.2	20.2	-
	button-press	**100.0**	**99.8**	**96.2**	**100.0**	**99.6**	-	**100.0**	0.2	-
RLfO	reach	**86.4**	**88.0**	15.0	-	-	-	-	-	51.6
	push	**58.2**	32.0	20.2	-	-	-	-	-	91.8
	plate-slide	**100.0**	**100.0**	83.2	-	-	-	-	-	100.0
	handle-press	77.8	84.4	**96.0**	-	-	-	-	-	78.6
	button-press	**100.0**	**100.0**	**92.6**	-	-	-	-	-	100.0
RL-expert	reach	39.2	54.0	42.0	57.8	**98.4**	-	-	-	51.6
	push	**95.6**	88.6	40.4	**90.4**	**99.6**	-	-	-	91.8
	plate-slide	**100.0**	**100.0**	89.6	**99.4**	85.4	-	-	-	100.0
	handle-press	72.6	82.0	**97.6**	81.2	**99.0**	-	-	-	78.6
	button-press	**100.0**	**100.0**	**97.0**	**100.0**	**100.0**	-	-	-	100.0
RL-sample	reach	**86.6**	**87.4**	63.0	**87.4**	**85.4**	-	-	-	88.8
	push	40.6	**47.8**	29.2	35.8	35.0	-	-	-	46.0
	plate-slide	**100.0**	**100.0**	**97.6**	**100.0**	**99.6**	-	-	-	100.0
	handle-press	78.4	81.0	**100.0**	76.8	82.8	-	-	-	83.4
	button-press	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	-	-	-	100.0

shows comparable performance with the privileged oracle. Reward prediction (RP) also performs well in all RL tasks. It, however, does not perform as well in IL tasks, since the reward function of RP can not generalize beyond the constant training reward. UDS does not perform well in these scenarios potentially due to being overly pessimistic. SMODICE uniformly performs worse than MAHALO in hopper and walker tasks. We hypothesize that the fixed discriminator reward function of SMODICE becomes overly pessimistic and discourages policy learning when the expert and dynamics distribution mismatches with one another.

5.2. Evaluation on Meta-World

We additionally evaluate MAHALO and other baseline algorithms on five robot manipulation tasks from Meta-World (Yu et al., 2020a). For Meta-World tasks, we use a non-positive reward function that promotes agent to reach the goal, which is a terminal state, as fast as possible. We generate a mixed quality dataset by adding different level of noises to a scripted policy provided by Meta-World for each task. We collect 100 trajectories each for zero-mean Gaussian noise with standard deviations $[0.1, 0.5, 1.0]$, and use these 300 trajectories as the mixed quality dataset. We separately collect 100 expert trajectories. For, RL-sample, we randomly sample trajectories to label 50% of the dataset. Note that we use more reward data for Meta-World since the state space (which includes the space of goals) is larger.

We observe that MAHALO is one of the overall best-performing algorithms. ATAC also achieves strong results in IL and RL-expert. This is because ATAC, in these scenarios, is only presented with expert data (we do not see similar effect in D4RL tasks since the expert dataset is much smaller there). UDS shows similar performance as MAHALO in RL scenarios. They, however, perform worse than MAHALO in IL. Reward prediction (RP) performs worse than MAHALO in most cases, especially in IL and ILfO scenarios. This shows that the pessimistic reward function in MAHALO is critical in achieving robust performance across different tasks and scenarios. SMODICE is not able to train reliably, and often diverges during training. We therefore take the success rate of the best performing policy during training instead of the final one. We find that the best policy of SMODICE underperforms the final policy of most algorithms, including MAHALO, potentially due to this instability in training.

Acknowledgements

We thank Andrey Kolobov for sharing data collected on the Meta-World tasks. We thank Mohak Bhardwaj for providing the script for processing results and Sinong Geng for helpful discussions on Meta-World experiments.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Antos, A., Szepesvári, C., and Munos, R. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

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

Cao, Z. and Sadigh, D. Learning from imperfect demonstrations from agents with varying dynamics. *IEEE Robotics and Automation Letters*, 6(3):5231–5238, 2021.

Cao, Z., Hao, Y., Li, M., and Sadigh, D. Learning feasibility to imitate demonstrators with different dynamics. In *Conference on Robot Learning*, pp. 363–372. PMLR, 2021.

Chang, J., Uehara, M., Sreenivas, D., Kidambi, R., and Sun, W. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34:965–979, 2021.

Cheng, C.-A., Xie, T., Jiang, N., and Agarwal, A. Adversarially trained actor critic for offline reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2022.

Desai, S., Durugkar, I., Karnan, H., Warnell, G., Hanna, J., and Stone, P. An imitation from observation approach to transfer learning with dynamics mismatch. *Advances in Neural Information Processing Systems*, 33:3917–3929, 2020.

Edwards, A., Sahni, H., Liu, R., Hung, J., Jain, A., Wang, R., Ecoffet, A., Miconi, T., Isbell, C., and Yosinski, J. Estimating Q(s, s') with deep deterministic dynamics gradients. In *International Conference on Machine Learning*, pp. 2825–2835. PMLR, 2020.

Eysenbach, B., Levine, S., and Salakhutdinov, R. R. Replacing rewards with examples: Example-based policy search via recursive classification. *Advances in Neural Information Processing Systems*, 34:11541–11552, 2021.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018a.

Fu, J., Singh, A., Ghosh, D., Yang, L., and Levine, S. Variational inverse control with events: A general framework for data-driven reward definition. *Advances in neural information processing systems*, 31, 2018b.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Gottesman, O., Johansson, F., Meier, J., Dent, J., Lee, D., Srinivasan, S., Zhang, L., Ding, Y., Wihl, D., Peng, X., et al. Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint arXiv:1805.12298*, 2018.

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

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Hu, H., Yang, Y., Zhao, Q., and Zhang, C. The provable benefits of unsupervised data sharing for offline reinforcement learning. In *International Conference on Learning Representations*, 2023.

Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., and Levine, S. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Kidambi, R., Chang, J., and Sun, W. Mobile: Model-based imitation learning from observation alone. *Advances in Neural Information Processing Systems*, 34:28598–28611, 2021.

Kim, G.-H., Seo, S., Lee, J., Jeon, W., Hwang, H., Yang, H., and Kim, K.-E. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

Kim, K., Gu, Y., Song, J., Zhao, S., and Ermon, S. Domain adaptive imitation learning. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Konyushkova, K., Zolna, K., Aytar, Y., Novikov, A., Reed, S., Cabi, S., and de Freitas, N. Semi-supervised reward learning for offline reinforcement learning. *arXiv preprint arXiv:2012.06899*, 2020.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2021.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Laroche, R., Trichelair, P., and Des Combes, R. T. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, pp. 3652–3661. PMLR, 2019.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, J., Hu, X., Xu, H., Liu, J., Zhan, X., Jia, Q.-S., and Zhang, Y.-Q. Mind the gap: Offline policy optimization for imperfect rewards. In *International Conference on Learning Representations*, 2023.

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

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Provably good batch off-policy reinforcement learning without great exploration. *Advances in neural information processing systems*, 33:1264–1274, 2020.

Ma, Y., Shen, A., Jayaraman, D., and Bastani, O. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9 (5), 2008.

Ng, A. Y., Russell, S., et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, volume 1, pp. 2, 2000.

Radosavovic, I., Wang, X., Pinto, L., and Malik, J. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7865–7871. IEEE, 2021.

Rigter, M., Lacerda, B., and Hawes, N. Rambo-RL: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 2022.

Schmeckpeper, K., Rybkin, O., Daniilidis, K., Levine, S., and Finn, C. Reinforcement learning with videos: Combining offline observations with interaction. In *Conference on Robot Learning*, pp. 339–354. PMLR, 2021.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Singh, A., Yang, L., Finn, C., and Levine, S. End-to-end robotic reinforcement learning without reward engineering. In *Robotics: Science and Systems*, 2019.

Singh, A., Yu, A., Yang, J., Zhang, J., Kumar, A., and Levine, S. Cog: Connecting new skills to past experience with offline reinforcement learning. In *Conference on Robot Learning*. PMLR, 2020.

Smith, M., Maystre, L., Dai, Z., and Ciosek, K. A strong baseline for batch imitation learning. *arXiv preprint arXiv:2302.02788*, 2023.

Torabi, F., Warnell, G., and Stone, P. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4950–4957, 2018.

Torabi, F., Warnell, G., and Stone, P. Adversarial imitation learning from state-only demonstrations. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2229–2231, 2019a.

Torabi, F., Warnell, G., and Stone, P. Recent advances in imitation learning from observation. In *International Joint Conference on Artificial Intelligence*, 2019b.

Uehara, M. and Sun, W. Pessimistic model-based offline reinforcement learning under partial coverage. In *International Conference on Learning Representations*, 2022.

Von Stackelberg, H. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

Wang, R., Ciliberto, C., Amadori, P. V., and Demiris, Y. Random expert distillation: Imitation learning via expert policy support estimation. In *International Conference on Machine Learning*, pp. 6536–6544. PMLR, 2019.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

Xie, T., Bhardwaj, M., Jiang, N., and Cheng, C.-A. Armor: A model-based framework for improving arbitrary baseline policies with offline data. *arXiv preprint arXiv:2211.04538*, 2022.

Xu, H., Zhan, X., Yin, H., and Qin, H. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *International Conference on Machine Learning*, pp. 24725–24742. PMLR, 2022.

Yang, C., Ma, X., Huang, W., Sun, F., Liu, H., Huang, J., and Gan, C. Imitation learning from observations by minimizing inverse dynamics disagreement. *Advances in neural information processing systems*, 32, 2019.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020a.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020b.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Yu, T., Kumar, A., Chebotar, Y., Hausman, K., Finn, C., and Levine, S. How to leverage unlabeled data in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 25611–25635. PMLR, 2022.

Yue, S., Wang, G., Shao, W., Zhang, Z., Lin, S., Ren, J., and Zhang, J. CLARE: Conservative model-based reward learning for offline inverse reinforcement learning. In *International Conference on Learning Representations*, 2023.

Zhu, Z., Lin, K., Dai, B., and Zhou, J. Off-policy imitation learning from observations. *Advances in Neural Information Processing Systems*, 33:12402–12413, 2020.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

Zolna, K., Novikov, A., Konyushkova, K., Gulcehre, C., Wang, Z., Aytar, Y., Denil, M., de Freitas, N., and Reed, S. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.

A. Related Work

Offline RL Existing work on offline RL can be broadly classified into model-based approaches (Yu et al., 2020b; Kidambi et al., 2020; Yu et al., 2021; Uehara & Sun, 2022; Rigter et al., 2022) and model-free approaches (Jin et al., 2021; Fujimoto & Gu, 2021; Xie et al., 2021; Cheng et al., 2022; Kumar et al., 2020; Kostrikov et al., 2021). The main challenge to offline RL is the mismatch between the offline dataset and test-time distribution. Two common strategies to offline RL are behavior regularization (Fujimoto & Gu, 2021; Wu et al., 2019; Kostrikov et al., 2021) (restricting policy to be close to the behavioral policy) and pessimism (Jin et al., 2021; Liu et al., 2020; Xie et al., 2021; Cheng et al., 2022; Kumar et al., 2020). Our paper follows more closely with model-free approaches built upon the concept of pessimism. In particular, we draw inspirations from approaches which conduct pessimistic policy evaluation on a version space of data-consistent hypotheses (Xie et al., 2021; Cheng et al., 2022; Uehara & Sun, 2022; Rigter et al., 2022). However, we consider a more general formulation than offline RL, where reward or action can be missing from a subset of data. Edwards et al. (2020) approach RL from observation by combining offline RL with a separately learned inverse dynamics model. It however makes implicit assumption that the dynamics data is abundant.

Offline RL with unlabeled data Yu et al. (2022) and Singh et al. (2020) consider a setting that reward can be missing from a subset of data, and uses a strategy of giving zero reward to these transitions. We show that MAHALO has a similar behavior when applied to this setting, while incurring less bias by correctly labeling rewards to transitions which are in support of the reward data. Recently, a strategy called PDS (Hu et al., 2023) is proposed to construct a pessimistic reward function using a ensemble of neural networks. MAHALO differs from PDS as the pessimistic reward function is constructed together with a pessimistic critic function. We also provide theoretical analysis of MAHALO in a general function approximator setting while PDS only has theoretical guarantees for linear MDPs. Li et al. (2023) consider a related setting where rewards can be mislabeled or imperfect. This is slightly different to our problem formation, where the reward (and action) for each transition is either correct or missing.

Offline IL Our work is also related to offline IL, where the expert dataset and the optional, often separately-collected, dynamics dataset can have different coverage. Zolna et al. (2020) adapt GAIL (Ho & Ermon, 2016) to an offline setting. Kim et al. (2021) seek to match the state-action occupancy using distribution correction estimation. Chang et al. (2021) minimize the state-action occupancy divergence using a pessimistic model. More recently, DWBC (Xu et al., 2022) proposes to approach offline IL through weighted behavior cloning, where the weights are given by a discriminator. CLARE (Yue et al., 2023) extends maximum entropy inverse reinforcement learning to an offline setting with an unknown quality dynamics dataset. Smith et al. (2023) propose to run offline RL with a binary reward indicating whether the transition is generated by the expert. These approaches require knowing the actions in expert demonstrations, which is a more restrictive assumption than our formulation.

IL from observations ILfO further relaxes the assumption of observing expert actions. Earlier approaches (Torabi et al., 2018; 2019a;b; Yang et al., 2019; Schmeckpeper et al., 2021) focus more on the online setting, where additional data collection is allowed when training the policy. There are a few recent papers that consider the offline ILfO setting. For example, Zhu et al. (2020); Ma et al. (2022) take an off-policy distribution matching approach, and Kidambi et al. (2021) use a model-based approach to minimax IL. These work, however, assumes a near-optimal expert. MAHALO, however, can work with both near-optimal expert trajectories and non-expert state-only trajectories labeled with reward.

Learning with dynamics mismatch Learning from observation is often closely related to learning with dynamics mismatch (Liu et al., 2019; Desai et al., 2020; Cao & Sadigh, 2021; Radosavovic et al., 2021; Cao et al., 2021; Ma et al., 2022), as it is hard to directly use action information from a different MDP. Although MAHALO can potentially handle this setting, we consider it beyond the scope of this paper and defer it to future work.

Learning reward functions When applied to offline IL/ILfO settings, our approach also has connection with work on reward learning from demonstrations (Ng et al., 2000; Abbeel & Ng, 2004; Ziebart et al., 2008; Fu et al., 2018a;b; Singh et al., 2019; Eysenbach et al., 2021; Konyushkova et al., 2020; Kim et al., 2020), as MAHALO produces a reward function. The main difference between MAHALO and this line of work is that we do not treat reward learning and policy learning as two separate processes. We view the learned reward function rather as a by-product of the algorithm.

B. Implementation Details

B.1. Neural Network Architectures

For MAHALO and all baseline algorithms, we parameterize the policy and (if applicable) critic networks with fully connected neural networks with 3 hidden layers of size 256. The policy is implemented as a tanh Gaussian distribution, where the mean and standard deviations are predicted by the two heads of the policy network. For MAHALO, we use the same network structure for the reward function. Our code is adapted from https://github.com/chinganc/lightATAC.

B.2. Hyperparameters

Since both MAHALO and most baseline algorithms are implemented via modifying ATAC (Cheng et al., 2022), we use similar choices of hyperparameters across different algorithms. We use a fixed learning rate across all algorithms and experiments. Same as Cheng et al. (2022), we use $\eta_{\text{slow}} = 5 \times 10^{-7}$ for policy updates and $\eta_{\text{fast}} = 5 \times 10^{-4}$ for updating the critic (and reward for MAHALO). We use a common discount factor of $\gamma = 0.99$ for all experiments. For target update, we use $\tau = 0.005$, same as Cheng et al. (2022); Haarnoja et al. (2018). We use a fixed batch size of 256. Our ATAC implementation is from https://github.com/chinganc/lightATAC.

To tune hyperparameter β, we run all algorithms with $\beta \in [0.1, 1.0, 10.0, 100.0, 1000.0]$ and report results using the best β for each scenario and task. For MAHALO, we tune hyperparameter α by experimenting with $(\alpha/\beta) \in [10.0, 100.0, 1000.0, 10000.0, 100000.0]$. To make it a fair comparison with other baseline algorithms, we report results generated by a *fixed* ratio (α/β) for each set of results. We use a fixed ratio of $(\alpha/\beta) \equiv 100000.0$ for D4RL tasks and $(\alpha/\beta) \equiv 100.0$ for Meta-World.

For SMODICE (Ma et al., 2022), we adapt the implementation from https://github.com/JasonMa2016/SMODICE and use the default parameters therein. We additionally experiment with SMODICE using χ^2 divergence (SMODICE uses KL divergence by default) and report its performance in Table 6 and Table 7 in Appendix E.

B.3. Training

For all algorithms based on ATAC, we warm-start training with 1) behavior cloning the behavioral policy and 2) learning a critic function to match the value of the behavioral policy. For MAHALO, we additionally train the reward function in this phase. For D4RL, we ran 1k steps of pretraining for D4RL tasks and 1M steps of pretraining for Meta-World. We then run ATAC and MAHALO updates for 1M steps and report the results.

C. Theoretical Analysis

C.1. Concentration Analysis

In this section, we state a few concentration results useful in finite-sample regime. These results can be obtained by straightforward modifications of the analysis in (Cheng et al., 2022). As such, we will not provide detailed proofs. We will instead point out the the correspondence of each lemma in (Cheng et al., 2022) and required modifications. We first provide the definition of covering number, which will be used to establish concentration results.

Definition C.1 (ϵ-covering number)**.** An ϵ-cover of a set Φ with respect to a metric $d(\cdot,\cdot)$ is a set $\{\tilde{\phi}_1, \dots, \tilde{\phi}_n\} \subseteq \Phi$ such that for each $\phi \in \Phi$, there exists some $\tilde{\phi}_i \in \{\tilde{\phi}_1, \dots, \tilde{\phi}_n\}$ such that $d(\phi, \tilde{\phi}_i) \le \epsilon$. We define the ϵ-covering number of a set Φ under a metric d to be the cardinality of the smallest ϵ-cover, denoted $\mathcal{N}_d(\Phi, \epsilon)$.

In particular, we use $\mathcal{N}_\infty(\mathcal{F}, \epsilon)$ to denote the ϵ-covering number under ℓ_∞ norm on set $\mathcal{F} \subseteq (\mathcal{S} \times \mathcal{A} \to [0, V_{\max}])$:

$$d_{\mathcal{F}}(f_1, f_2) := \|f_1 - f_2\|_\infty = \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} |f_1(s,a) - f_2(s,a)|. \tag{12}$$

Similarly, we use $\mathcal{N}_\infty(\mathcal{G}, \epsilon)$ to denote the ϵ-covering number under ℓ_∞ norm on set $\mathcal{G} \subseteq (\mathcal{S} \times \mathcal{S} \to [0, R_{\max}])$. Finally, we define metric for the policy class as

$$d_\Pi(\pi_1, \pi_2) := \|\pi_1 - \pi_2\|_{\infty,1} = \sup_{s\in\mathcal{S}} \|\pi_1(\cdot|s) - \pi_2(\cdot|s)\|_1, \tag{13}$$

and denote the corresponding ϵ-covering number as $\mathcal{N}_{\infty,1}(\Pi, \epsilon)$.

We first establish the concentration results for $\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f, g)$. The following lemma can obtained by modifying Theorem 9 from (Cheng et al., 2022) by additionally considering an $\frac{R_{\max}}{|\mathcal{D}_\mathcal{A}|}$-cover of $\mathcal{G}$.

Lemma C.2. *Suppose Assumption 4.1 holds. With probability at least $1-\delta$, for any $\pi \in \Pi$ and $f \in \mathcal{F}$, $g \in \mathcal{G}$,*

$$\sqrt{\mathcal{E}_\mu(\pi, f, g)} - \sqrt{\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f, g)} \leq \mathcal{O}\left(V_{max}\sqrt{\frac{\log(|\mathcal{N}_\infty(\mathcal{F}, V_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_\infty(\mathcal{G}, R_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_{\infty,1}(\Pi, 1/|\mathcal{D}_\mathcal{A}|)|/\delta)}{|\mathcal{D}_\mathcal{A}|}}\right) =: \mathcal{O}(\sqrt{\epsilon_\mu}), \tag{14}$$

We then consider $\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, Q^\pi, R)$. The following Lemma is a direct result of Theorem 8 from (Cheng et al., 2022) when Assumption 4.1 holds.

Lemma C.3. *Suppose Π, $\mathcal{F}$ and $\mathcal{G}$ satisfies Assumption 4.1. With probability at least $1-\delta$, for any $\pi \in \Pi$,*

$$\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, Q^\pi, R) \leq \mathcal{O}\left(\frac{V_{max}^2 \log(|\mathcal{N}_\infty(\mathcal{F}, V_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_{\infty,1}(\Pi, 1/|\mathcal{D}_\mathcal{A}|)|/\delta)}{|\mathcal{D}_\mathcal{A}|}\right) \leq \mathcal{O}(\epsilon_\mu). \tag{15}$$

We can also modify Theorem 8 and 9 from (Cheng et al., 2022) to provide concentration results on $\mathcal{E}_{\mathcal{D}_R}(g)$ and $\mathcal{E}_{\mathcal{D}_R}(R)$. This can be done by considering reward class $\mathcal{G}$ instead $\mathcal{F}$ (which also means that we need $\frac{R_{\max}}{|\mathcal{D}_R|}$-cover of $\mathcal{G}$).

Lemma C.4. *With probability at least $1-\delta$, for any $g \in \mathcal{G}$,*

$$\sqrt{\mathcal{E}_\nu(g)} - \sqrt{\mathcal{E}_{\mathcal{D}_R}(g)} = \mathcal{O}\left(R_{max}\sqrt{\frac{\log(\mathcal{N}_\infty(\mathcal{G}, R_{max}/|\mathcal{D}_R|)/\delta)}{|\mathcal{D}_R|}}\right) =: \mathcal{O}(\sqrt{\epsilon_\nu}), \tag{16}$$

Lemma C.5. *With probability at least $1-\delta$,*

$$\mathcal{E}_{\mathcal{D}_R}(R) \leq \mathcal{O}\left(\frac{R_{max}^2 \log(\mathcal{N}_\infty(\mathcal{G}, R_{max}/|\mathcal{D}_R|)/\delta)}{|\mathcal{D}_R|}\right) = \mathcal{O}(\epsilon_\nu). \tag{17}$$

Finally, the concentration result for $\mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, f)$ can be obtained by Hoeffding's inequality.

Lemma C.6. *With probability at least $1-\delta$, for any $\pi \in \Pi$ and $f \in \mathcal{F}$,*

$$\left|\mathcal{L}_\mu(\pi, f) - \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, f)\right| \leq \mathcal{O}\left(V_{max}\sqrt{\frac{\log(|\mathcal{N}_\infty(\mathcal{F}, V_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_{\infty,1}(\Pi, 1/|\mathcal{D}_\mathcal{A}|)|/\delta)}{|\mathcal{D}_\mathcal{A}|}}\right) \leq \mathcal{O}(\sqrt{\epsilon_\mu}). \tag{18}$$

C.2. Auxiliary Lemmas

In this section, we provide two auxiliary lemmas for showing the main results. These lemmas are stated in a general manner. For example, we use μ to denote a probability distribution on $\mathcal{S} \times \mathcal{A}$ rather than the dynamics data distribution. Similarly, we use f to denote an arbitrary function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

The first lemma bounds the expectation of a function f on an arbitrary distribution μ by the execution of its absolute value on another distribution ρ and an inner product of difference of the two distributions and f. In later analysis, we will use this lemma to decompose the expectation on an arbitrary distribution into an "in-support" expectation and an "off-support" term.

Lemma C.7. *For any $\mu, \rho \in \Delta(\mathcal{S} \times \mathcal{A})$, and any $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,*

$$\mathbb{E}_\mu[f(s,a)] \leq 2\mathbb{E}_\rho[|f(s,a)|] + \langle \mu \setminus \rho, f\rangle \tag{19}$$

where $(\mu \setminus \rho)(s,a) := \max(\mu(s,a) - \rho(s,a), 0)$ and $\langle \iota, f\rangle := \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \iota(s,a) \cdot f(s,a)$ for any $\iota : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

Proof. This lemma can be shown by decomposing probability measure μ into probability measure ρ and a signed measure

$\mu - \rho$, and then splitting the signed measure $\mu - \rho$ into its positive and negative parts.

$$\begin{aligned}
\mathbb{E}_\mu[f(s,a)] &= \sum_{(s,a)} \mu(s,a) \cdot f(s,a) \\
&= \sum_{(s,a)} \rho(s,a) \cdot f(s,a) + \sum_{(s,a)} (\mu(s,a) - \rho(s,a)) \cdot f(s,a) \\
&= \mathbb{E}_\rho[f(s,a)] + \sum_{(s,a)} (\mu(s,a) - \rho(s,a)) \cdot f(s,a) \\
&= \mathbb{E}_\rho[f(s,a)] + \sum_{(s,a)} \mathbb{1}(\mu(s,a) > \rho(s,a))(\mu(s,a) - \rho(s,a)) \cdot f(s,a) \\
&\qquad + \sum_{(s,a)} \mathbb{1}(\mu(s,a) \leq \rho(s,a))(\mu(s,a) - \rho(s,a)) \cdot f(s,a) \\
&= \mathbb{E}_\rho[f(s,a)] + \langle \mu \setminus \rho, f \rangle + \sum_{(s,a)} \mathbb{1}(\rho(s,a) \geq \mu(s,a))(\rho(s,a) - \mu(s,a)) \cdot (-f(s,a)) \\
&\leq \mathbb{E}_\rho[|f(s,a)|] + \langle \mu \setminus \rho, f \rangle + \sum_{(s,a)} \mathbb{1}(\rho(s,a) \geq \mu(s,a))(\rho(s,a) - \mu(s,a)) \cdot |f(s,a)| \\
&\leq \mathbb{E}_\rho[|f(s,a)|] + \langle \mu \setminus \rho, f \rangle + \sum_{(s,a)} \rho(s,a) \cdot |f(s,a)| \\
&= 2\mathbb{E}_\rho[|f(s,a)|] + \langle \mu \setminus \rho, f \rangle
\end{aligned}$$

where the first inequality follows from $|f(s,a)| \geq f(s,a)$ and $|f(s,a)| \geq -f(s,a)$, and the second inequality follows from $\rho(s,a) \geq \mathbb{1}(\rho(s,a) \geq \mu(s,a))(\rho(s,a) - \mu(s,a)) \geq 0$ and $|f(s,a)| \geq 0$. ■

The second lemma bounds the ℓ_1 norm of a function f under a probability measure μ by its ℓ_2 norm under the same measure.

Lemma C.8. *For any $\mu \in \Delta(\mathcal{S} \times \mathcal{A})$, and $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,*

$$\mathbb{E}_\mu\big[|f(s,a)|\big] \leq \sqrt{\mathbb{E}_\mu\big[\big(f(s,a)\big)^2\big]} = \|f\|_{2,\mu}. \tag{20}$$

Proof. By Jensen's inequality,

$$\mathbb{E}_\mu\big[|f(s,a)|\big] = \mathbb{E}_\mu\left[\sqrt{\big(f(s,a)\big)^2}\right] \leq \sqrt{\mathbb{E}_\mu[\big(f(s,a)\big)^2]} = \sqrt{\|f\|_{2,\mu}^2} = \|f\|_{2,\mu}. \tag{21}$$

■

The next lemma is a simple fact about $f(d_0, \pi)$. The proof is omitted as it can be shown by a standard telescoping argument.

Lemma C.9. *Let d_0 be the initial state distribution of the MDP. For any policies $\pi, \tilde{\pi}$ and $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,*

$$(1-\gamma) f(d_0, \pi) = \mathbb{E}_{\tilde{\pi}}[f(s,\pi) - \mathcal{P}^\pi f(s,a)]. \tag{22}$$

In particular, $\mathbb{E}_{\tilde{\pi}}[f(s,\pi) - \mathcal{P}^\pi f(s,a)] = (1-\gamma) f(d_0,\pi) = \mathbb{E}_\pi[f(s,\pi) - \mathcal{P}^\pi f(s,a)]$.

C.3. Main Results

We first show the approximate robust policy improvement property in the next lemma. This property provides a performance lower bound for the learned policy $\hat{\pi}$ relative to the behavioral policy μ. Note that, in the infinite sample regime, $\epsilon_\nu \to 0$ and $\epsilon_\mu \to 0$, we have robust policy improvement property for *any* $\alpha \geq 0$ and $\beta \geq 0$.

Proposition C.10 (Robust Policy Improvement. General version of Proposition 4.4)**.** *Suppose that the policy class Π, critic class $\mathcal{F}$ and reward class $\mathcal{G}$ satisfy Assumption 4.1. Let $\hat{\pi}$ be the solution to (5). Denote $\epsilon_\mu := \frac{V_{max}^2 \log(|\mathcal{N}_\infty(\mathcal{F}, V_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_\infty(\mathcal{G}, R_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_{\infty,1}(\Pi, 1/|\mathcal{D}_\mathcal{A}|)|/\delta)}{|\mathcal{D}_\mathcal{A}|}$ and $\epsilon_\nu := \frac{R_{max}^2 \log(\mathcal{N}_\infty(\mathcal{G}, R_{max}/|\mathcal{D}_R|)/\delta)}{|\mathcal{D}_R|}$. Define*

$$\epsilon_J := \sqrt{\epsilon_\mu} + \alpha\epsilon_\nu + \beta\epsilon_\mu. \tag{23}$$

For any policy $\pi \in \Pi$,

$$J(\mu) - J(\hat{\pi}) \leq \frac{1}{1-\gamma}\Big(-\mathcal{L}_\mu(\pi, f^\pi) + \mathcal{O}(\epsilon_J)\Big), \tag{24}$$

where f^π is the solution to the followers' objective defined in (5). In particular, with the choice of $\pi = \mu$, we have robust policy improvement:

$$J(\mu) - J(\hat{\pi}) \leq \mathcal{O}\left(\frac{\epsilon_J}{1-\gamma}\right). \tag{25}$$

Proof. Since $(\hat{\pi}, f^{\hat{\pi}}, g^{\hat{\pi}})$ is the solution to the Stackelberg game (5),

$$\begin{aligned}
(1-\gamma)(J(\hat{\pi}) - J(\mu)) &= \mathbb{E}_\mu[Q^{\hat{\pi}}(s,\hat{\pi}) - Q^{\hat{\pi}}(s,a)] && \text{(Performance Difference Lemma)}\\
&= \mathcal{L}_\mu(\hat{\pi}, Q^{\hat{\pi}}) && \text{(Definition of } \mathcal{L}_\mu)\\
&\geq \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, Q^{\hat{\pi}}) + \alpha\mathcal{E}_{\mathcal{D}_R}(R) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, Q^{\hat{\pi}}, R)\\
&\quad - \mathcal{O}(\sqrt{\epsilon_\mu}) - \mathcal{O}(\alpha\epsilon_\nu) - \mathcal{O}(\beta\epsilon_\mu) && \text{(Lemma C.3, Lemma C.5, Lemma C.6)}\\
&\geq \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, f^{\hat{\pi}}) + \alpha\mathcal{E}_{\mathcal{D}_R}(g^{\hat{\pi}}) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, f^{\hat{\pi}}, g^{\hat{\pi}})\\
&\quad - \mathcal{O}(\sqrt{\epsilon_\mu} + \alpha\epsilon_\nu + \beta\epsilon_\mu) && \text{(Optimality of } (f^{\hat{\pi}}, g^{\hat{\pi}}), Q^{\hat{\pi}} \in \mathcal{F}, R \in \mathcal{G})\\
&\geq \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, f^{\hat{\pi}}) - \mathcal{O}(\sqrt{\epsilon_\mu} + \alpha\epsilon_\nu + \beta\epsilon_\mu) && (\mathcal{E}_{\mathcal{D}_R}(g^{\hat{\pi}}) \geq 0 \text{ and } \mathcal{E}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, f^{\hat{\pi}}, g^{\hat{\pi}}) \geq 0)\\
&\geq \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, f^{\pi}) - \mathcal{O}(\sqrt{\epsilon_\mu} + \alpha\epsilon_\nu + \beta\epsilon_\mu) && \text{(Optimality of } \hat{\pi})\\
&\geq \mathcal{L}_{\mu}(\pi, f^{\pi}) - \mathcal{O}(\sqrt{\epsilon_\mu}) - \mathcal{O}(\sqrt{\epsilon_\mu} + \alpha\epsilon_\nu + \beta\epsilon_\mu) && \text{(Lemma C.6)}\\
&= \mathcal{L}_{\mu}(\pi, f^{\pi}) - \mathcal{O}(\sqrt{\epsilon_\mu} + \alpha\epsilon_\nu + \beta\epsilon_\mu).
\end{aligned}$$

■

The following lemma establishes an upper bound on the reward error $\mathcal{E}_{\mathcal{D}_R}$ and Bellman error $\mathcal{E}_{\mathcal{D}_\mathcal{A}}$ for f^π and g^π, the minimizer of the followers' objective in (5). Intuitively, with sufficiently large α and β, f^π and g^π should induce small reward error $\mathcal{E}_{\mathcal{D}_R}(g^\pi)$ and Bellman error $\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi)$.

Lemma C.11. *Assume Assumption 4.1 holds. We have, for any $\pi \in \Pi$,*

$$\alpha\mathcal{E}_{\mathcal{D}_R}(g^\pi) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi) \leq 2V_{max} + \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu), \tag{26}$$

where ϵ_μ and ϵ_ν are defined in Lemma C.2 and Lemma C.4, respectively. This implies that $\mathcal{E}_{\mathcal{D}_R}(g^\pi) \leq \mathcal{O}\left(\frac{1}{\alpha}V_{max} + \epsilon_\nu + \frac{\beta}{\alpha}\epsilon_\mu\right)$ and $\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi) \leq \mathcal{O}\left(\frac{1}{\beta}V_{max} + \frac{\alpha}{\beta}\epsilon_\nu + \epsilon_\mu\right)$.

Proof. By Assumption 4.1, $Q^\pi \in \mathcal{F}$ and $R \in \mathcal{G}$. By optimality of f^π and g^π,

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi) + \alpha\mathcal{E}_{\mathcal{D}_R}(g^\pi) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi) &\leq \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, Q^\pi) + \alpha\mathcal{E}_{\mathcal{D}_R}(R) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, Q^\pi, R)\\
&\leq \mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, Q^\pi) + \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu).
\end{aligned} \tag{27}$$

The last inequality follows from Lemma C.2 and Lemma C.4.

By definition of $\mathcal{L}_{\mathcal{D}_\mathcal{A}}$, we have $\mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi) = \mathbb{E}_{\mathcal{D}_\mathcal{A}}[f^\pi(s,\pi) - f^\pi(s,a)] \geq \mathbb{E}_\mu[0 - V_{\max}] = -V_{\max}$. Similarly, $\mathcal{L}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, Q^{\hat{\pi}}) = \mathbb{E}_{\mathcal{D}_\mathcal{A}}[Q^{\hat{\pi}}(s,\hat{\pi}) - Q^{\hat{\pi}}(s,a)] \leq \mathbb{E}_\mu[V_{\max} - 0] = V_{\max}$. By combining these with (27), we have

$$\alpha\mathcal{E}_\nu(g^\pi) + \beta\mathcal{E}_\mu(\pi, f^\pi, g^\pi) \leq 2V_{\max} + \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu). \tag{28}$$

■

We are now ready to prove the main theorem. It provides a performance lower bound for the learned policy $\hat{\pi}$ with respect to any comparator policy $\pi \in \Pi$.

Theorem C.12 (General version of Theorem 4.3)**.** *Let $\hat{\pi}$ be the solution to the Stackelberg game (5) and let $\pi \in \Pi$ be any comparator policy. Let $C_1 \geq 1, C_2 \geq 1$ be any constants, $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$ be an arbitrary distribution that satisfies $\mathcal{C}(\rho; \mu, \mathcal{F}, \mathcal{G}, \pi) \leq C_1$ and $\mathcal{C}(\rho; \nu, \mathcal{G}) \leq C_2$. Let ϵ_μ and ϵ_ν be as defined in Lemma C.2 and Lemma C.4, respectively. Choosing $\alpha = \Theta\left(\frac{V_{max}^{1/3}(\sqrt{C_1\epsilon_\nu}+\sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\nu}\right)$ and $\beta = \Theta\left(\frac{V_{max}^{1/3}(\sqrt{C_1\epsilon_\nu}+\sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\mu}\right)$, with high probability:*

$$
\begin{aligned}
J(\pi) - J(\hat{\pi}) \leq \mathcal{O}\left(\frac{(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu}) + V_{max}^{1/3}(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu})^{2/3}}{1-\gamma}\right) \\
+ \underbrace{\frac{\langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\rangle}{1-\gamma}}_{\textit{off-support error (dynamics)}} + \underbrace{\frac{\langle (d^\pi \ominus \mu) \setminus \rho, |\bar{R} - \bar{g}^\pi|\rangle}{1-\gamma}}_{\textit{off-support error (reward)}}
\end{aligned}
\tag{29}
$$

where $(d^\pi \ominus \mu) \coloneqq d^\pi \setminus \mu + \mu \setminus d^\pi$. See Lemma C.7 for definitions of $\cdot \setminus \cdot$ and $\langle \cdot, \cdot \rangle$.

Proof. We have,

$$
\begin{aligned}
(1-\gamma)(J(\pi) - J(\hat{\pi})) &= (1-\gamma)(J(\pi) - J(\mu)) - (1-\gamma)(J(\hat{\pi}) - J(\mu)) \\
&\leq (1-\gamma)(J(\pi) - J(\mu)) - \mathcal{L}_\mu(\pi, f^\pi) + \mathcal{O}(\epsilon_J) && \text{(Proposition C.10)} \\
&= \mathbb{E}_\pi[\bar{R}(s,a)] - \mathbb{E}_\mu[\bar{R}(s,a)] \\
&\quad - \mathbb{E}_\mu[f^\pi(s,\pi) - f^\pi(s,a)] + \mathcal{O}(\epsilon_J) && \text{(Definition of } J \text{ and } \mathcal{L}_\mu) \\
&= \mathbb{E}_\pi[\bar{R}(s,a)] - \mathbb{E}_\mu[\bar{R}(s,a)] \\
&\quad - \mathbb{E}_\mu[f^\pi(s,\pi) - \mathcal{P}^\pi f(s,a) + \mathcal{P}^\pi f^\pi(s,a) - f^\pi(s,a)] + \mathcal{O}(\epsilon_J) \\
&= \mathbb{E}_\pi[\bar{R}(s,a)] - \mathbb{E}_\mu[f^\pi(s,\pi) - \mathcal{P}^\pi f^\pi(s,a)] \\
&\quad + \mathbb{E}_\mu[f^\pi(s,a) - \bar{R}(s,a) - \mathcal{P}^\pi f^\pi(s,a)] + \mathcal{O}(\epsilon_J) \\
&= \mathbb{E}_\pi[\bar{R}(s,a)] - \mathbb{E}_\pi[f^\pi(s,\pi) - \mathcal{P}^\pi f^\pi(s,a)] \\
&\quad + \mathbb{E}_\mu[(f^\pi - \bar{R} - \mathcal{P}^\pi f^\pi)(s,a)] + \mathcal{O}(\epsilon_J) && \text{(Lemma C.9)} \\
&= \mathbb{E}_\pi[(\bar{R} + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)] \\
&\quad + \mathbb{E}_\mu[(f^\pi - \bar{R} - \mathcal{P}^\pi f^\pi)(s,a)] + \mathcal{O}(\epsilon_J) \\
&= \underbrace{\mathbb{E}_\pi[(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)]}_{\text{(I)}} + \underbrace{\mathbb{E}_\mu[(f^\pi - \bar{g}^\pi - \mathcal{P}^\pi f^\pi)(s,a)]}_{\text{(II)}} \\
&\quad + \underbrace{\mathbb{E}_\pi[(\bar{R} - \bar{g}^\pi)(s,a)] + \mathbb{E}_\mu[(\bar{g}^\pi - \bar{R})(s,a)]}_{\text{(III)}} + \mathcal{O}(\epsilon_J)
\end{aligned}
$$

We first establish an upper bound for term (II). We will use this later to bound for (I). By Lemma C.8,

$$
\begin{aligned}
\text{(II)} &\leq \|(f^\pi - \bar{g}^\pi - \mathcal{P}^\pi f^\pi)(s,a)\|_{2,\mu} = \sqrt{\mathcal{E}_\mu(\pi, f^\pi, g^\pi)} \\
&\leq \sqrt{\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi)} + \mathcal{O}(\sqrt{\epsilon_\mu}) && \text{(Lemma C.2)} \\
&\leq \mathcal{O}\left(\sqrt{\frac{1}{\beta}V_{\max} + \frac{\alpha}{\beta}\epsilon_\nu + \epsilon_\mu}\right) + \mathcal{O}(\sqrt{\epsilon_\mu}) && \text{(Lemma C.11)} \\
&= \mathcal{O}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right).
\end{aligned}
$$

We now bound term (I). By Lemma C.7,

$$
\text{(I)} = \mathbb{E}_\pi[(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)] \leq 2\mathbb{E}_\rho[|(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)|] + \langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\rangle \tag{30}
$$

where, by Lemma C.8,

$$\begin{aligned}\mathbb{E}_\rho[|(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)|] &\le \|\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\|_{2,\rho} \\ &\le \sqrt{C_2 \|\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\|_{2,\mu}^2} && \text{(Definition of } \mathcal{C}(\rho;\mu,\mathcal{F},\mathcal{G},\pi)) \\ &\le \mathcal{O}\left(\sqrt{C_2}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right)\right). && \text{(Lemma C.11)}\end{aligned}$$

Hence, we have

$$\text{(I)} \le \mathcal{O}\left(\sqrt{C_2}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right)\right) + \langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\rangle \tag{31}$$

Finally, we establish a bound for term (III) in a similar way.

$$\begin{aligned}\text{(III)} &= \mathbb{E}_\pi[(\bar{R} - \bar{g}^\pi)(s,a)] + \mathbb{E}_\mu[(\bar{g}^\pi - \bar{R})(s,a)] \\ &= \langle d^\pi \setminus \mu, \bar{R} - \bar{g}^\pi\rangle + \langle \mu \setminus d^\pi, \bar{g}^\pi - \bar{R}\rangle \\ &\le \langle d^\pi \ominus \mu, |\bar{R} - \bar{g}^\pi|\rangle && (|\bar{R} - \bar{g}^\pi| \ge \bar{R} - \bar{g}^\pi, |\bar{R} - \bar{g}^\pi| \ge \bar{g}^\pi - \bar{R}) \\ &\le 2\mathbb{E}_\rho[|\bar{R} - \bar{g}^\pi|] + \langle (d^\pi \ominus \mu) \setminus \rho, |\bar{R} - \bar{g}^\pi|\rangle && \text{(Lemma C.7)}\end{aligned}$$

where $(d^\pi \ominus \mu) \coloneqq d^\pi \setminus \mu + \mu \setminus d^\pi$.

By Lemma C.8, we have,

$$\begin{aligned}\mathbb{E}_\rho[|\bar{R} - \bar{g}^\pi|] &\le \|\bar{R} - \bar{g}^\pi\|_{2,\rho} \\ &\le \sqrt{C_1 \|R - g^\pi\|_{2,\nu}^2} = \sqrt{C_1}\sqrt{\mathcal{E}_\nu(g^\pi)} && \text{(Definition of } \mathcal{C}(\rho;\nu,\mathcal{G})) \\ &\le \sqrt{C_1}\left(\sqrt{\mathcal{E}_{\mathcal{D}_R}(g^\pi)} + \mathcal{O}(\sqrt{\epsilon_\nu})\right) && \text{(Lemma C.4)} \\ &\le \mathcal{O}\left(\sqrt{C_1}\left(\sqrt{\frac{1}{\alpha}V_{\max}} + \sqrt{\epsilon_\nu} + \sqrt{\frac{\beta}{\alpha}\epsilon_\mu}\right)\right) && \text{(Lemma C.11)}\end{aligned}$$

Finally, we have

$$\begin{aligned}(1-\gamma)(J(\pi) - J(\hat{\pi})) \le \mathcal{O}&\left(\sqrt{C_1}\left(\sqrt{\frac{1}{\alpha}V_{\max}} + \sqrt{\epsilon_\nu} + \sqrt{\frac{\beta}{\alpha}\epsilon_\mu}\right) + \sqrt{C_2}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right) + \alpha\epsilon_\nu + \beta\epsilon_\mu\right) \\ &+ \langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\rangle + \langle (d^\pi \ominus \mu) \setminus \rho, |\bar{R} - \bar{g}^\pi|\rangle\end{aligned} \tag{32}$$

By choosing $\alpha = \Theta\left(\frac{V_{\max}^{1/3}(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\nu}\right)$ and $\beta = \Theta\left(\frac{V_{\max}^{1/3}(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\mu}\right)$, we have

$$\begin{aligned}(1-\gamma)(J(\pi) - J(\hat{\pi})) \le \mathcal{O}&\left((\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu}) + V_{\max}^{1/3}(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu})^{2/3}\right) \\ &+ \langle d^\pi \setminus \rho, \bar{g} + \mathcal{P}^\pi f^\pi - f^\pi\rangle + \langle (d^\pi \ominus \mu) \setminus \rho, |\bar{R} - \bar{g}^\pi|\rangle.\end{aligned} \tag{33}$$

■

D. Absolute Pessimism

We can implement MAHALO based on absolute pessimism to optimize for $J(\pi)$, instead of $J(\pi) - J(\mu)$. This results in a realization of MAHALO based on PSPI (Xie et al., 2021). For simplicity of illustration, we assume that d_0 is known. A model-free version can be realized the solving the two-player game below:

$$\begin{aligned}\hat{\pi} &\in \arg\max_{\pi\in\Pi}(1-\gamma)f^\pi(d_0,\pi) \\ \text{s.t.} \quad f^\pi &\in \arg\min_{f\in\mathcal{F},g\in\mathcal{G}}(1-\gamma)f^\pi(d_0,\pi) + \alpha\mathcal{E}_{\mathcal{D}_R}(g) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi,f,g),\end{aligned} \tag{34}$$

with $\alpha \geq 0, \beta \geq 0$ being hyperparameters, and

$$\mathcal{E}_{\mathcal{D}_R}(g) \coloneqq \mathbb{E}_{\mathcal{D}_R}\big[\big(g(s,s') - r\big)^2\big],$$
$$\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f, g) \coloneqq \mathbb{E}_{\mathcal{D}_\mathcal{A}}\big[\big(f(s,a) - g(s,s') - \gamma f(s',\pi)\big)^2\big] - \min_{f' \in \mathcal{F}} \mathbb{E}_{\mathcal{D}_\mathcal{A}}\big[\big(f'(s,a) - g(s,s') - \gamma f(s',\pi)\big)^2\big].$$

Compared with (5), the formulation in (34) replaces $\mathcal{L}_{\mathcal{D}_\mathcal{A}}(\pi, f)$ with $(1-\gamma)f^\pi(s_0, \pi)$ which is a surrogate of $(1-\gamma)J(\pi)$. Below we adapt the proof of Theorem 4.3 to give a guarantee on the absolute pessimism case.

Theorem D.1 (Absolute Pessimism Version of MAHALO)**.** *Let $\hat{\pi}$ be the solution to the Stackelberg game (34) and let $\pi \in \Pi$ be any comparator policy. Let $C_1 \geq 1, C_2 \geq 1$ be any constants, $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$ be an arbitrary distribution that satisfies $\mathcal{C}(\rho; \mu, \mathcal{F}, \mathcal{G}, \pi) \leq C_1$ and $\mathcal{C}(\rho; \nu, \mathcal{G}) \leq C_2$. Let ϵ_μ and ϵ_ν be as defined in Lemma C.2 and Lemma C.4, respectively. Choosing $\alpha = \Theta\left(\frac{V_{max}^{1/3}(\sqrt{C_1\epsilon_\nu}+\sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\nu}\right)$ and $\beta = \Theta\left(\frac{V_{max}^{1/3}(\sqrt{C_1\epsilon_\nu}+\sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\mu}\right)$, with high probability:*

$$J(\pi) - J(\hat{\pi}) \leq \mathcal{O}\left(\frac{(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu}) + V_{max}^{1/3}(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu})^{2/3}}{1-\gamma}\right) + \underbrace{\frac{\langle d^\pi \setminus \rho, \bar{R} + \mathcal{P}^\pi f^\pi - f^\pi\rangle}{1-\gamma}}_{\textit{off-support error}} \tag{35}$$

See Lemma C.7 for definitions of $\cdot \setminus \cdot$ and $\langle \cdot, \cdot \rangle$.

If we compare Theorem D.1 of absolute pessimism and Theorem C.12 of relative pessimism, we see that the main difference is in how the off-support error is measured. Theorem D.1 measures the off-support error in $d^\pi \setminus \rho$ for both Bellman error and rewards (because $\frac{\langle d^\pi \setminus \rho, \bar{R} + \mathcal{P}^\pi f^\pi - f^\pi\rangle}{1-\gamma} = \frac{\langle d^\pi \setminus \rho, \bar{g} + \mathcal{P}^\pi f^\pi - f^\pi\rangle}{1-\gamma} + \frac{\langle d^\pi \setminus \rho, \bar{R} - \bar{g}\rangle}{1-\gamma}$), whereas Theorem C.12 uses $d^\pi \setminus \rho$ for Bellman error and $(d^\pi \ominus \mu) \setminus \rho$ for reward. As a result, the upper bound in Theorem C.12 considers how reward generalizes (off-support) on both both d^π and μ, but the one in Theorem D.1 only concerns (one-sided) generalization on d^π, which is preferable in some sense. However, the policy learned by the absolute pessimism version does not have robust policy improvement guarantee. In experiments, we found this absolute pessimism version is more sensitive to hyperparameter choices than the MAHALO based on relative pessimism.

D.1. Technical Lemmas

Proposition D.2 (Absolute Pessimism Version of Proposition C.10)**.** *Suppose that the policy class Π, critic class $\mathcal{F}$ and reward class $\mathcal{G}$ satisfy Assumption 4.1. Let $\hat{\pi}$ be the solution to (5). Denote $\epsilon_\mu \coloneqq \frac{V_{max}^2 \log(|\mathcal{N}_\infty(\mathcal{F}, V_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_\infty(\mathcal{G}, R_{max}/|\mathcal{D}_\mathcal{A}|)||\mathcal{N}_{\infty,1}(\Pi, 1/|\mathcal{D}_\mathcal{A}|)|/\delta)}{|\mathcal{D}_\mathcal{A}|}$ and $\epsilon_\nu \coloneqq \frac{R_{max}^2 \log(\mathcal{N}_\infty(\mathcal{G}, R_{max}/|\mathcal{D}_R|)/\delta)}{|\mathcal{D}_R|}$. Define*

$$\epsilon_J \coloneqq \alpha\epsilon_\nu + \beta\epsilon_\mu. \tag{36}$$

For any policy $\pi \in \Pi$,

$$f^\pi(d_0, \pi) \leq J(\hat{\pi}) + \mathcal{O}(\epsilon_J), \tag{37}$$

where f^π is the solution to the followers' objective defined in (34).

Proof. Since $(\hat{\pi}, f^{\hat{\pi}}, g^{\hat{\pi}})$ is the solution to the Stackelberg game (5),

$$\begin{aligned}
J(\hat{\pi}) &\geq Q^{\hat{\pi}}(d_0, \hat{\pi}) + \alpha\mathcal{E}_{\mathcal{D}_R}(R) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, Q^{\hat{\pi}}, R) \\
&\quad - \mathcal{O}(\alpha\epsilon_\nu) - \mathcal{O}(\beta\epsilon_\mu) && \text{(Lemma C.3, Lemma C.5)} \\
&\geq f^{\hat{\pi}}(d_0, \hat{\pi}) + \alpha\mathcal{E}_{\mathcal{D}_R}(g^{\hat{\pi}}) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, f^{\hat{\pi}}, g^{\hat{\pi}}) \\
&\quad - \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu) && \text{(Optimality of } (f^{\hat{\pi}}, g^{\hat{\pi}}), Q^{\hat{\pi}} \in \mathcal{F}, R \in \mathcal{G}) \\
&\geq f^{\hat{\pi}}(d_0, \hat{\pi}) - \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu) && (\mathcal{E}_{\mathcal{D}_R}(g^{\hat{\pi}}) \geq 0 \text{ and } \mathcal{E}_{\mathcal{D}_\mathcal{A}}(\hat{\pi}, f^{\hat{\pi}}, g^{\hat{\pi}}) \geq 0) \\
&\geq f^\pi(d_0, \pi) - \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu) && \text{(Optimality of } \hat{\pi})
\end{aligned}$$

■

Lemma D.3 (Absolute Pessimism Version of Lemma C.11)**.** *Assume Assumption 4.1 holds. We have, for any $\pi \in \Pi$,*

$$\alpha\mathcal{E}_{\mathcal{D}_R}(g^\pi) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi) \leq V_{max} + \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu), \tag{38}$$

where ϵ_μ and ϵ_ν are defined in Lemma C.2 and Lemma C.4, respectively. This implies that $\mathcal{E}_{\mathcal{D}_R}(g^\pi) \leq \mathcal{O}\left(\frac{1}{\alpha}V_{max} + \epsilon_\nu + \frac{\beta}{\alpha}\epsilon_\mu\right)$ and $\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi) \leq \mathcal{O}\left(\frac{1}{\beta}V_{max} + \frac{\alpha}{\beta}\epsilon_\nu + \epsilon_\mu\right)$.

Proof. By Assumption 4.1, $Q^\pi \in \mathcal{F}$ and $R \in \mathcal{G}$. By optimality of f^π and g^π,

$$\begin{aligned} f^\pi(d_0, \pi) + \alpha\mathcal{E}_{\mathcal{D}_R}(g^\pi) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, f^\pi, g^\pi) &\leq Q^\pi(d_0, \pi) + \alpha\mathcal{E}_{\mathcal{D}_R}(R) + \beta\mathcal{E}_{\mathcal{D}_\mathcal{A}}(\pi, Q^\pi, R) \\ &\leq Q^\pi(d_0, \pi) + \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu). \end{aligned} \tag{39}$$

The last inequality follows from Lemma C.2 and Lemma C.4. Since $Q^\pi(d_0, \pi) \leq V_{\max}$ and $f^\pi(d_0, \pi) \geq 0$, we have

$$\alpha\mathcal{E}_\nu(g^\pi) + \beta\mathcal{E}_\mu(\pi, f^\pi, g^\pi) \leq V_{\max} + \mathcal{O}(\alpha\epsilon_\nu + \beta\epsilon_\mu). \tag{40}$$

■

D.2. Proof of Theorem D.1

Proof. We have,

$$\begin{aligned} (1-\gamma)(J(\pi) - J(\hat{\pi})) &\leq (1-\gamma)J(\pi) - (1-\gamma)f^\pi(d_0, \pi) + \mathcal{O}(\epsilon_J) && \text{(Proposition D.2)} \\ &= \mathbb{E}_\pi[\bar{R}(s,a)] - (1-\gamma)f^\pi(d_0, \pi) + \mathcal{O}(\epsilon_J) && \text{(Definition of } J \text{ and } \mathcal{L}_\mu) \\ &= \mathbb{E}_\pi[\bar{R}(s,a)] - \mathbb{E}_\pi[f^\pi(s,\pi) - \mathcal{P}^\pi f(s,a)] + \mathcal{O}(\epsilon_J) && \text{(Lemma C.9)} \\ &= \mathbb{E}_\pi[(\bar{R} + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)] + \mathcal{O}(\epsilon_J) \\ &= \underbrace{\mathbb{E}_\pi[(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)]}_{\text{(I)}} + \underbrace{\mathbb{E}_\pi[(\bar{R} - \bar{g}^\pi)(s,a)]}_{\text{(II)}} + \mathcal{O}(\epsilon_J) \end{aligned}$$

We bound term (I). By Lemma C.7,

$$\text{(I)} = \mathbb{E}_\pi[(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)] \leq 2\mathbb{E}_\rho[|(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)|] + \langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\rangle \tag{41}$$

where, by Lemma C.8,

$$\begin{aligned} \mathbb{E}_\rho[|(\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi)(s,a)|] &\leq \|\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\|_{2,\rho} \\ &\leq \sqrt{C_2\|\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\|_{2,\mu}^2} && \text{(Definition of } \mathcal{C}(\rho; \mu, \mathcal{F}, \mathcal{G}, \pi)) \\ &\leq \sqrt{C_2\|\bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\|_{2,\mathcal{D}_A}^2 + C_2\epsilon_\mu} && \text{(Definition of } \mathcal{C}(\rho; \mu, \mathcal{F}, \mathcal{G}, \pi)) \\ &\leq \mathcal{O}\left(\sqrt{C_2}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right)\right). && \text{(Lemma D.3)} \end{aligned}$$

Hence, we have

$$\text{(I)} \leq \mathcal{O}\left(\sqrt{C_2}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right)\right) + \langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\rangle \tag{42}$$

Finally, we establish a bound for term (II) in a similar way.

$$\begin{aligned} \text{(II)} &= \mathbb{E}_\pi[(\bar{R} - \bar{g}^\pi)(s,a)] \\ &\leq 2\mathbb{E}_\rho[|\bar{R} - \bar{g}^\pi|] + \langle d^\pi \setminus \rho, \bar{R} - \bar{g}^\pi\rangle && \text{(Lemma C.7)} \end{aligned}$$

By Lemma C.8, we have,

$$\begin{aligned}
\mathbb{E}_\rho[|\bar{R} - \bar{g}^\pi|] &\leq \|\bar{R} - \bar{g}^\pi\|_{2,\rho} \\
&\leq \sqrt{C_1 \|R - g^\pi\|_{2,\nu}^2} = \sqrt{C_1}\sqrt{\mathcal{E}_\nu(g^\pi)} && \text{(Definition of } \mathcal{C}(\rho;\nu,\mathcal{G})) \\
&\leq \sqrt{C_1}\big(\sqrt{\mathcal{E}_{\mathcal{D}_R}(g^\pi)} + \mathcal{O}(\sqrt{\epsilon_\nu})\big) && \text{(Lemma C.4)} \\
&\leq \mathcal{O}\left(\sqrt{C_1}\left(\sqrt{\frac{1}{\alpha}V_{\max}} + \sqrt{\epsilon_\nu} + \sqrt{\frac{\beta}{\alpha}\epsilon_\mu}\right)\right) && \text{(Lemma D.3)}
\end{aligned}$$

Finally, we have

$$\begin{aligned}
(1-\gamma)(J(\pi) - J(\hat{\pi})) &\leq \mathcal{O}\left(\sqrt{C_1}\left(\sqrt{\frac{1}{\alpha}V_{\max}} + \sqrt{\epsilon_\nu} + \sqrt{\frac{\beta}{\alpha}\epsilon_\mu}\right) + \sqrt{C_2}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right) + \alpha\epsilon_\nu + \beta\epsilon_\mu\right) \\
&\quad + \langle d^\pi \setminus \rho, \bar{g}^\pi + \mathcal{P}^\pi f^\pi - f^\pi\rangle + \langle d^\pi \setminus \rho, \bar{R} - \bar{g}^\pi\rangle \\
&= \mathcal{O}\left(\sqrt{C_1}\left(\sqrt{\frac{1}{\alpha}V_{\max}} + \sqrt{\epsilon_\nu} + \sqrt{\frac{\beta}{\alpha}\epsilon_\mu}\right) + \sqrt{C_2}\left(\sqrt{\frac{1}{\beta}V_{\max}} + \sqrt{\frac{\alpha}{\beta}\epsilon_\nu} + \sqrt{\epsilon_\mu}\right) + \alpha\epsilon_\nu + \beta\epsilon_\mu\right) \\
&\quad + \langle d^\pi \setminus \rho, \bar{R} + \mathcal{P}^\pi f^\pi - f^\pi\rangle
\end{aligned} \tag{43}$$

By choosing $\alpha = \Theta\left(\frac{V_{\max}^{1/3}(\sqrt{C_1\epsilon_\nu}+\sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\nu}\right)$ and $\beta = \Theta\left(\frac{V_{\max}^{1/3}(\sqrt{C_1\epsilon_\nu}+\sqrt{C_2\epsilon_\mu})^{2/3}}{\epsilon_\mu}\right)$, we have

$$\begin{aligned}
(1-\gamma)(J(\pi) - J(\hat{\pi})) &\leq \mathcal{O}\left((\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu}) + V_{\max}^{1/3}(\sqrt{C_1\epsilon_\nu} + \sqrt{C_2\epsilon_\mu})^{2/3}\right) \\
&\quad + \langle d^\pi \setminus \rho, \bar{R} + \mathcal{P}^\pi f^\pi - f^\pi\rangle
\end{aligned} \tag{44}$$

■

D.3. Experimental Results on D4RL Benchmark

We implement MAHALO realized by PSPI (Xie et al., 2021) which we refer to as MAHALO-PSPI, and test it on the D4RL benchmark. We observe that MAHALO-PSPI has similar performance as MAHALO-ATAC. MAHALO-PSPI achieves top performance in all hopper and walker tasks. It does slighly worse than MAHALO-ATAC in halfcheetah tasks. We hypothize that this is a limitation of the base algorithm PSPI, since it achieves lower score than ATAC when doing offline RL on halfcheetah datasets (Cheng et al., 2022). For MAHALO-PSPI, we use a *fixed* ratio of $(\alpha/\beta) \equiv 10000.0$ and tune $\beta \in [10.0, 100.0, 1000.0]$. For other hyperparameters, we use what is described in Appendix B.

E. Experimental Results with Standard Error

We report the average scores and standard errors across random seeds for D4RL and Meta-World in Table 6 and Table 7, respectively. MAHALO is the overall best-performing algorithm across the board, and its standard errors are in general small. Here we additionally consider two baseline algorithms. First, we implement a variant of UDS (Yu et al., 2022) which we call **UDS-A**. UDS-A has two copies of labeled transitions, one labeled with true reward, while the other (falsely) labeled with minimum reward. UDS requires more information than MAHALO: UDS needs to know the common transitions between reward and dynamics datasets. UDS-A, hence, would be a more fair to compare with MAHALO as they make the same assumption on datasets. Second, we consider a version of SMODICE (Ma et al., 2022) using χ^2 divergence, which we refer to as **SMODICE-χ^2**. We observe that UDS-A performs similarly with UDS, and hence underperforms MAHALO. SMODICE-χ^2 achieves top performance (slighly better, but comparable to MAHALO) on Meta-World IL and ILfO tasks, potentially thanks to the training stability from χ^2 divergence. But it fails miserably on all D4RL tasks.

Table 5. Results of MAHALO-PSPI on D4RL benchmark (Fu et al., 2020) We show the average normalized score over 50 evaluation trials across 10 random seeds. We paste the scores of MAHALO-ATAC (which is presented in the main text) from Table 2 for a comparison with MAHALO-PSPI. MAHALO-PSPI shows comparable performance as MAHALO-ATAC. MAHALO-PSPI results are generated using a fixed α/β ratio of 10000.0.

Scenario	Dataset	MAHALO-PSPI	MAHALO-ATAC
ILfO	hopper	108.71	104.66
	walker	92.86	88.60
	halfcheetah	29.89	61.24
IL	hopper	106.36	104.06
	walker	96.02	89.03
	halfcheetah	51.81	54.99
RLfO	hopper	101.81	106.47
	walker	98.70	96.65
	halfcheetah	30.93	50.38
RL-expert	hopper	98.09	87.73
	walker	91.07	103.18
	halfcheetah	27.15	48.43
RL-sample	hopper	104.04	103.08
	walker	95.66	95.00
	halfcheetah	28.58	68.30

Table 6. Results on D4RL benchmark (Fu et al., 2020) We show the average normalized scores over 50 evaluation trials and the standard errors across 10 random seeds. Algorithms with scores greater than 0.9 of the best score (excluding Oracle) is marked in bold. MAHALO achieves top performance in almost every scenario and task except halfcheetah. MAHALO is also able to match Oracle performance, despite having access to less reward data. † ATAC only uses data with both dynamics and reward information. $^{+}$ Oracle has access to reward for all dynamics data.

Scenario	Dataset	MAHALO	RP	AP	UDS	UDS-A	ATAC†	BCO	BC	SMODICE	SMODICE-χ^2	Oracle^{+}
ILfO	hopper	**104.66±2.38**	**97.48±0.29**	45.97±4.62	-	-	-	46.98±6.04	-	67.72 ± 4.19	4.12±3.00	-
	walker	**88.60±0.89**	77.15±1.11	61.11±5.55	-	-	-	60.48±4.30	-	1.52±0.73	21.89±6.35	-
	halfcheetah	**61.24±1.14**	36.00±6.01	4.87±1.02	-	-	-	5.53±0.95	-	**59.64±1.17**	37.96±1.46	-
IL	hopper	**104.06±2.37**	**97.88±0.16**	53.35±6.30	32.21±2.17	28.53±1.90	63.56±6.37	-	30.83±4.34	74.75±4.19	0.97±0.01	-
	walker	**89.03±1.04**	77.71±0.81	63.53±3.43	8.45±4.35	14.56±6.52	78.35±4.08	-	14.96±5.63	0.94 ± 0.36	71.66±2.50	-
	halfcheetah	**54.99±5.63**	23.20±4.53	3.74±1.06	25.82±2.87	28.46±2.03	3.64±0.95	-	26.87±2.73	**58.40±1.00**	39.63±2.55	-
RLfO	hopper	**106.47±1.06**	**105.65±0.18**	47.01±5.73	-	-	-	-	-	-	-	103.39±1.66
	walker	**96.65±2.56**	**97.26±0.30**	63.30±5.52	-	-	-	-	-	-	-	98.52±0.47
	halfcheetah	50.38±1.51	**68.66±0.70**	3.35±0.42	-	-	-	-	-	-	-	63.57±0.98
RL-expert	hopper	87.73±7.30	**105.56±0.28**	51.54±4.30	**98.63±0.15**	**98.08±0.27**	65.45±3.05	-	-	-	-	103.39±1.66
	walker	**103.18±3.06**	**98.31±0.38**	56.27±4.80	72.97±0.65	74.06±0.85	66.40±5.68	-	-	-	-	98.52±0.47
	halfcheetah	48.43±2.02	**64.37±1.50**	3.47±0.83	13.68±2.58	9.99±2.69	3.41±0.83	-	-	-	-	63.57±0.98
RL-sample	hopper	**103.08±1.80**	**101.66±3.24**	71.92±10.30	0.95±0.01	0.94±0.00	71.50±7.80	-	-	-	-	103.34±2.55
	walker	**95.00±0.46**	**94.59±0.61**	5.94±3.11	-0.00±0.01	0.00±0.01	0.48±0.30	-	-	-	-	95.73±0.33
	halfcheetah	**68.30±0.76**	**68.71±0.92**	13.26±4.07	20.36±3.95	24.58±4.49	19.38±4.18	-	-	-	-	69.91±0.43

Table 7. Success rate of the final policy (with the exception of SMODICE*) on Meta-World (Yu et al., 2020a). The success rate is computed over 50 evaluation episodes. We consider an episode success if it is able to reach the goal within 128 steps. We report the average success rate across 10 random seeds. MAHALO is one of the overall best-performing algorithms. ATAC also achieves strong results in IL and RL-expert because it is presented with expert-only data. * SMODICE is not able to train reliably, and often diverges during training. We therefore take the success rate of the best performing policy during training instead of the final one.

Scenario	Dataset	MAHALO	RP	AP	UDS	UDS-A	ATAC†	BCO	BC	SMODICE*	SMODICE-χ^2	Oracle$^+$
ILfO	reach	**65.0±3.1**	**62.6±1.8**	13.0±2.2	-	-	-	11.6±1.7	-	10.6 ± 4.3	58.4±3.0	-
	push	**62.4±1.9**	11.6±3.7	12.4±3.7	-	-	-	14.6±2.2	-	0.4±0.3	**62.2±3.7**	-
	plate-slide	**100.0±0.0**	22.0±11.0	**94.0±3.9**	-	-	-	75.4±6.8	-	0.0±0.0	**99.8±0.2**	-
	handle-press	75.4±4.2	32.4±11.2	**96.6±2.5**	-	-	-	87.8±7.7	-	16.6±5.1	**98.0±1.0**	-
	button-press	**100.0±0.0**	**100.0±0.0**	**93.6±1.8**	-	-	-	**93.8±2.2**	-	0.4±0.4	**99.2±0.8**	-
IL	reach	24.6±9.1	23.6±5.0	38.0±2.9	21.6±8.2	13.6±5.3	**98.0±0.6**	-	62.2±7.3	19.8±6.9	61.4±1.6	-
	push	**92.6±2.2**	11.8±6.5	35.2±6.2	79.2±4.9	72.4±3.9	50.0±15.4	-	**91.4±2.7**	0.2±0.2	65.0±4.0	-
	plate-slide	80.4±9.9	34.4±11.2	89.8±5.0	76.2±8.8	62.0±11.8	89.4±6.7	-	85.3±8.5	0.0±0.0	**100.0±0.0**	-
	handle-press	71.4±5.7	34.8±9.6	**100.0±0.0**	35.2±10.0	57.4±11.2	**97.0±1.8**	-	75.2±5.7	20.2±5.7	**98.4±0.6**	-
	button-press	**100.0±0.0**	**99.8±0.2**	**96.2±2.8**	**100.0±0.0**	**100.0±0.0**	**99.6±0.4**	-	**100.0±0.0**	0.2±0.2	**100.0±0.0**	-
RLfO	reach	**86.4±1.8**	**88.0±2.4**	15.0±2.0	-	-	-	-	-	-	-	51.6±13.2
	push	**58.2±3.4**	32.0±2.2	20.2±3.1	-	-	-	-	-	-	-	91.8±1.6
	plate-slide	**100.0±0.0**	**100.0±0.0**	83.2±5.4	-	-	-	-	-	-	-	100.0±0.0
	handle-press	77.8±3.1	84.4±3.2	**96.0±3.2**	-	-	-	-	-	-	-	78.6±2.5
	button-press	**100.0±0.0**	**100.0±0.0**	**92.6±2.5**	-	-	-	-	-	-	-	100.0±0.0
RL-expert	reach	39.2±12.1	54.0±13.0	42.0±2.3	57.8±12.3	33.6±10.8	**98.4±0.8**	-	-	-	-	51.6±13.2
	push	**95.6±1.7**	88.6±1.4	40.4±5.5	**90.4±1.9**	**90.0±2.2**	**99.6±0.4**	-	-	-	-	91.8±1.6
	plate-slide	**100.0±0.0**	**100.0±0.0**	89.6±3.8	**99.4±0.6**	**99.0±0.6**	85.4±7.3	-	-	-	-	100.0±0.0
	handle-press	72.6±9.1	82.0±3.5	**97.6±2.1**	81.2±2.6	80.8±2.9	**99.0±0.4**	-	-	-	-	78.6±2.5
	button-press	**100.0±0.0**	**100.0±0.0**	**97.0±1.7**	**100.0±0.0**	**100.0±0.0**	**96.4±3.4**	-	-	-	-	100.0±0.0
RL-sample	reach	**86.6±1.4**	**87.4±2.4**	63.0±3.0	**87.4±1.6**	**89.8±1.8**	**85.4±3.0**	-	-	-	-	88.8±1.8
	push	40.6±3.8	**47.8±4.0**	29.2±4.9	35.8±3.3	37.4±4.4	35.0±4.8	-	-	-	-	46.0±2.3
	plate-slide	**100.0±0.0**	**100.0±0.0**	**97.6±1.2**	**100.0±0.0**	**100.0±0.0**	**99.6±0.4**	-	-	-	-	100.0±0.0
	handle-press	78.4±3.9	81.0±2.6	**100.0±0.0**	76.8±6.6	77.8±6.2	82.8±4.4	-	-	-	-	83.4±3.1
	button-press	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	-	-	-	-	100.0±0.0