IOSTOM: Offline Imitation Learning from Observations Via State Transition Occupancy Matching

Quang Anh Pham, Janaka Chathuranga Brahmanage, Tien Mai, Akshat Kumar
Singapore Management University
{qa.pham.2025, janakat.2022}@phdcs.smu.edu.sg
{atmai, akshatkumar}@smu.edu.sg

Abstract

Offline Learning from Observation (LfO) focuses on enabling agents to imitate expert behavior using datasets that contain only expert state trajectories and separate transition data with suboptimal actions. This setting is both practical and critical in real-world scenarios where direct environment interaction or access to expert action labels is costly, risky, or infeasible. Most existing LfO methods attempt to solve this problem through state or state-action occupancy matching. They typically rely on pretraining a discriminator to differentiate between expert and non-expert states, which could introduce errors and instability—especially when the discriminator is poorly trained. While recent discriminator-free methods have emerged, they generally require substantially more data, limiting their practicality in low-data regimes. In this paper, we propose IOSTOM (*Imitation from Observation via State Transition Occupancy Matching*), a novel offline LfO algorithm designed to overcome these limitations. Our approach formulates a learning objective based on the joint state visitation distribution. A key distinction of IOSTOM is that it first excludes actions entirely from the training objective. Instead, we learn an *implicit policy* that models transition probabilities between states, resulting in a more compact and stable optimization problem. To recover the expert policy, we introduce an efficient action inference mechanism that *avoids training an inverse dynamics model*. Extensive empirical evaluations across diverse offline LfO benchmarks show that IOSTOM substantially outperforms state-of-the-art methods, demonstrating both improved performance and data efficiency.

1 Introduction

Imitation learning is a framework in machine learning where agents learn to perform tasks by mimicking expert demonstrations rather than learning through trial-and-error or explicit reward signals [33, 37, 15]. This approach is particularly useful in environments where designing reward functions is difficult or costly. Its practical relevance spans a wide range of domains, including robotics, healthcare, and autonomous driving, where expert behavior is available but reinforcement learning is either too risky, time-consuming, or expensive to deploy [35, 46, 24]. By leveraging expert demonstrations, imitation learning enables faster deployment of intelligent systems and facilitates safer exploration in complex, real-world environments.

A variant of this framework, known as Imitation from Observations or Learning from Observations (LfO), focuses on learning policies using only state trajectories without access to the expert's actions. This setting presents unique challenges, such as inferring intent and disambiguating optimal behavior from partial information, but also broadens applicability to scenarios where action data is unavailable or hard to record. For example, in video-based learning from human demonstrations in household

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

tasks (e.g., cleaning or cooking), it is often infeasible to capture the precise motor commands or control actions, making observation-only learning a practical and valuable approach.

Recent developments in imitation learning from observations have increasingly focused on scenarios where a limited set of expert state-only trajectories is complemented by sub-optimal state-action demonstrations. While this setup has practical appeal, many existing methods rely on distribution-matching frameworks that operate over complex input tuples such as (s, a, s') or (s, s', s''), where s, s' represents a state and a an action [18, 38]. These formulations appear to be sample-inefficient due to the structural complexity of the inputs. Furthermore, some approaches require estimating a discriminator to support training [27, 48], which can be unreliable in low-data or high-dimensional settings [39]. Other methods rely on learning an inverse dynamics model to recover unobserved expert actions, which introduces approximation errors that may degrade the quality of the learned policy [43, 50]. *To the best of our knowledge, no existing method in the LfO setting addresses all of these limitations simultaneously.*

We aim to address the aforementioned limitations in this work. Our central idea is to ignore sub-optimal actions and instead focus on learning the *transition probabilities between consecutive states*, leading to a simple and compact learning objective that only involves joint state pairs (s, s'). We then develop an efficient method to recover the expert policy without requiring an inverse dynamics model. Specifically, our contributions are as follows:

(i) By first disregarding sub-optimal actions in the demonstration data, we propose to learn *state-to-state transition probabilities*, which can be interpreted as an *implicit policy* that encapsulates the actual state-action policy. We then formulate the learning problem as matching *joint state visitation distributions* and leverage convexity and Lagrangian duality to derive a tractable joint-state Q-learning procedure. This training formulation, in addition to being *discriminator-free*, is *significantly simpler and more compact* than prior approaches that rely on action annotations, as it only involves consecutive state pairs (s, s').

(ii) We further introduce two novel strategies for efficiently extracting a policy from the learned Q-function. First, we propose a Q-weighted behavior cloning (BC) approach, which is theoretically equivalent to the standard advantage-weighted BC but offers a more compact and stable formulation. Second, we propose a single-stage process for recovering the expert policy without estimating an *inverse dynamics model*, thereby avoiding approximation errors that could degrade policy quality.

(iii) We validate our LfO framework using state-of-the-art benchmarks, demonstrating that our algorithm, IOSTOM, significantly outperforms existing methods. The implementation of IOSTOM is publicly available at `https://github.com/quanganh1999/IOSTOM`.

2 Related Work

Learning from Observations Different from Learning from Demonstrations (LfD) [37, 35] using expert state-action dataset, Learning from Observations (LfO) [45] addresses the challenge of imitation learning when expert actions are unavailable, relying instead on state-only expert trajectories. LfO research can be broadly distinguished into online and offline paradigms. In online LfO setting, the agent can actively interact with the environment [44, 49]. Recent advancement in online LfO focuses on improving adversarial imitation learning (AIL) approaches [14, 6]. The core idea of AIL relies on generative adversarial networks (GANs) [11] where a generator policy learns to imitate an expert, while a discriminator differentiates between agent-generated and expert data. In addition to online LfO, its offline setting has also received significant interest due to practical constraints of many real-world scenarios, where continuous interaction is costly or risky. It assumes access to state-only expert demonstrations and an action-labeled background dataset from other interactions [50]. A common approach trains an inverse dynamics model (IDM) on background data to infer expert actions, then applies Behavior Cloning (BC) [43, 4]. However, BC needs extensive, high-quality expert data and can suffer from compounding errors, exacerbated by IDM inaccuracies [34]. Another line adapts the Distribution Correction Estimation (DICE) framework [28]. These methods (e.g., PW-DICE [48], SMODICE [27], LobsDICE [18]) use a discriminator to estimate density ratios as pseudorewards for downstream RL. While avoiding an explicit IDM, their success depends on discriminator quality and RL robustness. Recently, DILO [38] bypass both IDM and discriminator learning by solving the dual

of an occupancy matching objective, directly optimizing a utility function. This function, measuring long-term divergence from expert visitation, is used to extract the imitation policy.

Imitation Learning via Distribution Matching: Distribution Matching objective is a powerful tool in Reinforcement Learning (RL) that has demonstrated its effectiveness in exploration [25], goal-conditioned RL [26, 1], and especially Imitation Learning (IL). Many popular IL methods such as BC, GAIL [13], and DAgger [36] can be formulated as statistical divergence minimization problems [10]. This minimization can be performed over the state, state-action, or trajectory space, resulting in different IL approaches [30]. The well-known DICE-family algorithms [21, 19, 18, 27] also optimizes state or state-action visitation distribution matching problems between the learner and expert via their dual formulations [29]. They often require learning a discriminator to estimate the log-ratio for distribution correction. Recently, [39] introduce ReCOIL, a discriminator-free method that also optimizes the duality of the state-action occupancy matching problem. This work is closely related to our IOSTOM, as both learn a score function that assigns high values to expert samples and low values to non-expert samples. However, IOSTOM focuses on solving the state-transition occupancy matching problem instead of the standard state-action one to address the LfO problem. Our setting is generally considered more challenging than the LfD setting targeted by ReCOIL [17], mainly due to the absence of expert actions in LfO.

3 Background

Markov Decision Process. We consider a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, p_0, \gamma \rangle$, where $\mathcal{S}$ denotes the set of states, $\mathcal{A}$ set of actions, p_0 represents the distribution of initial states, $\mathcal{R} : S \times A \to \mathbb{R}$ defines the reward function for each state-action pair, and $\mathcal{T} : S \times A \to S$ is the transition function, i.e., $\mathcal{T}(s'|s, a)$ is the probability of reaching state $s' \in \mathcal{S}$ when action $a \in \mathcal{A}$ is taken at state $s \in \mathcal{S}$. The parameter $\gamma \in [0, 1)$ is the discount factor. In reinforcement learning (RL), the goal is to find a policy that maximizes the expected long-term accumulated reward, i.e., $\max_\pi \left\{ \mathbb{E}_{(s,a)\sim d^\pi}[\mathcal{R}(s, a)] \right\}$, where $d^\pi(s, a)$ is the occupancy measure (or state-action visitation distribution) of policy π. The definitions of $d^\pi(s, a)$ and other common visitation distributions are include in Table 1.

	State Distribution	State-Action Distribution	Joint Distribution	Transition Distribution
Notation	$d^\pi(s)$	$d^\pi(s, a)$	$d^\pi(s, a, s')$	$d^\pi(s, s')$
Support	$\mathcal{S}$	$\mathcal{S} \times \mathcal{A}$	$\mathcal{S} \times \mathcal{A} \times \mathcal{S}$	$\mathcal{S} \times \mathcal{S}$
Definition	$(1-\gamma)\sum_{t=1}^{\infty} \gamma^t P(s_t = s \mid \pi)$	$d^\pi(s)\pi(a\|s)$	$d^\pi(s, a)\mathcal{T}(s'\|s, a)$	$\sum_{\mathcal{A}} d^\pi(s, a, s')$

Table 1: Overview on different stationary distributions adapted from [50]

Offline Imitation Learning from Observations. Different from standard Imitation Learning, Learning from observations (LfO) relaxes the requirement of action in expert dataset. In offline LfO setting, we assume access to an expert observation-only dataset $D_E = \{s, s'\}$ and a suboptimal interaction dataset $D_I = \{s, a, s'\}$. We also denote the respective visitation distributions of the expert and suboptimal datasets D_E and D_I as d^E and d^I. Several methods have been proposed to handle this challenging scenario. For instance, SMODICE [27], a state-of-the-art approach for learning from observations (LfO), minimizes an upper bound of KL-divergence $\mathbb{D}_{\text{KL}}[d^\pi(s) \,\|\, d^E(s)]$ via the objective:

$$\min \mathbb{E}_{d^\pi(s)} \left[\log \frac{d^I(s)}{d^E(s)} \right] + \mathbb{D}_f \left[d^\pi(s, a) \,\|\, d^I(s, a) \right].$$

where $\mathbb{D}_f$ denotes an f-divergence between two distributions. LobsDICE [18] proposes a similar formulation:

$$\min \mathbb{D}_{\text{KL}} \left(d^\pi(s, s') \,\|\, d^E(s, s') \right) + \alpha \, \mathbb{D}_{\text{KL}} \left(d^\pi(s, a) \,\|\, d^I(s, a) \right),$$

DILO [38] introduces a discriminator-free approach via solving another objective:

$$\min \mathbb{D}_f \big[\beta \, d^\pi(s, s', a') + (1-\beta) \, d^I(s, s', a'), \; \beta \, d^E(s, s', a') + (1-\beta) \, d^I(s, s', a') \big],$$

where $d^\pi(s, s', a') = d^\pi(s, s')\pi(a'|s')$.

Most methods (except DILO) rely on a learned discriminator to predict distribution correction ratios based on s or (s, s'), which can be unreliable in low-data or high-dimensional settings [39]. Although DILO is discriminator-free, it requires costly triplet samples (s, s', s'') and a non-standard visitation structure, which limits its sample efficiency. In contrast, our proposed method **IOSTOM** is discriminator-free and only involves joint state pairs (s, s') during learning. This leads to a more compact representation and improved sample efficiency over prior approaches. Furthermore, IOSTOM is the only approach directly minimizing $\mathbb{D}_f[d^\pi(s, s') \,\|\, d^E(s, s')]$ which is the main objective of LfO [44, 18].

4 IOSTOM - Imitation Learning via State Transition Occupancy Matching

We present a novel framework for imitation learning from observations, which is structured into two sequential stages. The first stage focuses on recovering the state-transition probabilities, denoted as $g(s' \mid s)$, which represents the probability of transitioning to the next state s' given the current state s. In the second stage, we recover a policy based on the learned state transition model $g(s' \mid s)$. The key insight behind our method is to simplify the LfO problem by initially ignoring the action information in the dataset. By doing so, we treat the state transition model $g(s' \mid s)$ as a form of "*implicit policy*" that governs the behavior of the demonstrator. This abstraction allows us to bypass the need for explicit action labels during the early phase of learning.

4.1 Joint State Q-learning Formulation

Our approach centers on recovering the transition probability between states, denoted as $g(s' \mid s)$, which can be viewed as an *implicit policy*. This transition model can be computed based on the underlying policy and environment dynamics as: $g(s' \mid s) = \sum_a \pi(a \mid s)\mathcal{T}(s' \mid s, a)$. To facilitate training, we define the joint visitation distribution over state pairs (s, s') as:

$$d^g(s, s') = d^\pi(s)g(s'|s) = \sum_a d^\pi(s, a)\mathcal{T}(s' \mid s, a),$$

where $d^\pi(s, a)$ is the state-action visitation distribution under policy π, recursively computed via the single-step transpose Bellman equation [29]:

$$d^\pi(s, a) = (1 - \gamma)d_0(s)\pi(a \mid s) + \gamma \sum_{s',a'} d^\pi(s', a')\mathcal{T}(s \mid s', a')\pi(a \mid s),$$

with $d_0(s)$ denoting the initial state distribution. To support learning from observations, our goal is to remove the dependence on actions in the visitation distribution. We introduce the following proposition to this end:

Proposition 1. *The joint visitation distribution $d^g(s, s')$ can be expressed as:*

$$d^g(s, s') = (1 - \gamma)d_0(s)g(s' \mid s) + \gamma g(s' \mid s) \sum_{\overline{s}} d^g(\overline{s}, s) \tag{1}$$

We note that the flow equality in Equation (1) depends solely on the joint state visitation distribution $d^g(s, s')$ and the state transition function $g(s' \mid s)$—the two key quantities we aim to recover.

Using the flow constraints described above and following the approach in [39], our main objective is to minimize the divergence between two joint state visitation distributions: $d^I_{\text{mix}}(s, s')$ and $d^{E,I}_{\text{mix}}(s, s')$, defined as follows:

$$d^I_{\text{mix}}(s, s') = \alpha d(s, s') + (1 - \alpha)d^I(s, s'), \text{ and } d^{E,I}_{\text{mix}}(s, s') \quad = \alpha d^E(s, s') + (1 - \alpha)d^I(s, s'),$$

where $\alpha \in (0, 1)$ is a mixing hyperparameter. Here, $d^I_{\text{mix}}(s, s')$ represents a mixed visitation distribution combining the learned state-transition behavior with that of the suboptimal dataset, while $d^{E,I}_{\text{mix}}(s, s')$ represents a mixed distribution combining expert behavior and suboptimal behavior. Combining this with the flow constraints in (1), we formulate our learning problem as follows:

$$\begin{aligned} \max_{g, d \geq 0} \quad & -\mathbb{D}_f(d^I_{mix}(s, s') \| d^{E,I}_{mix}(s, s')) \\ s.t. \quad & d(s, s') = (1 - \gamma)d_0(s)g(s'|s) + \gamma g(s'|s) \sum_{\overline{s}} d(\overline{s}, s) \end{aligned} \tag{2}$$

The constrained problem described above is convex in d when the transition function g is fixed. Following a similar approach to that in [39], the maximization over d can be equivalently reformulated as an unconstrained optimization problem via Lagrangian duality. We formalize this result in the following proposition, with the full derivation provided in the appendix.

Proposition 2. *The constrained optimization problem in Equation* (2) *is equivalent to the following unconstrained max-min problem:*

$$\max_{g} \min_{Q(s,s')} \Big\{ \alpha(1-\gamma)\, \mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \mathbb{E}_{(s,s')\sim d_{mix}^{E,I}}\left[f^*\left(\gamma \mathbb{E}_{s''\sim g(\cdot|s')}\left[Q(s',s'')\right] - Q(s,s')\right)\right] \\ - (1-\alpha)\, \mathbb{E}_{(s,s')\sim d^I}\left[\gamma \mathbb{E}_{s''\sim g(\cdot|s')}\left[Q(s',s'')\right] - Q(s,s')\right] \Big\}, \tag{3}$$

where $Q(s,s')$ are the Lagrange multipliers and f^* denotes the convex conjugate of a chosen convex function f.

For notational convenience, we define $Q(s,g) = \mathbb{E}_{s'\sim g(\cdot|s)}\left[Q(s,s')\right]$. Using this shorthand, we write the objective function in (3) as (See Appendix B.3 for complete derivation):

$$\max_{g} \min_{Q(s,s')} \Big\{ L(Q,g) = \alpha(1-\gamma)\, \mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \alpha \mathbb{E}_{(s,s')\sim d^E}\left[f^*\left(\gamma Q(s',g) - Q(s,s')\right)\right] \\ + (1-\alpha)\, \mathbb{E}_{(s,s')\sim d^I}\left[\widetilde{f^*}\left(\gamma Q(s',g) - Q(s,s')\right)\right] \Big\}, \tag{4}$$

where $\widetilde{f^*}(t) = f^*(t) - t$. This formulation learns a *joint-state value function* $Q(s,s')$, where actions are entirely ignored. While being more compact and manageable than prior formulations that rely on (s,a,s') or even (s,s',s'') tuples, our approach benefits from ignoring suboptimal actions in the data. This design helps mitigate the imbalance in offline datasets, where expert demonstrations lack action labels, whereas suboptimal trajectories contain fully observed actions.

4.2 Extreme V-learning

Solving the maximin objective in (4) can be done via dual optimization by alternating between optimizing $Q(s,s')$ and g. Specifically, we minimize $L(Q,g)$ over Q, and then maximize it over g with Q fixed. Following [47], the maximization over g can be approximated by computing $\max_g Q(s,g)$ for each state s, which requires sampling from g. In offline RL, this is challenging due to out-of-distribution (OOD) issues when querying Q on unseen state transitions [23]. To address this, we adopt the in-sample soft estimation from [9], replacing the hard maximization with a KL-regularized soft value: $V_Q^g(s) = \mathbb{E}_{s'\sim g(\cdot|s)}[Q(s,s') - \beta \log \frac{g(s'|s)}{\mu(s'|s)}]$, where μ is the behavior policy and β controls the KL strength, keeping g close to μ to avoid OOD samples. This is supported by the following proposition:

Proposition 3. $\max_g\{V_Q^g(s)\}$ *can be approximated via the following Extreme-V objective:*

$$\min_{V} \left\{ J(V \mid Q) = \mathbb{E}_{(s,s')\sim d_{mix}^{E,I}}\left[\exp(\omega(s,s')) + \omega(s,s') - 1\right] \right\}. \tag{5}$$

where $\omega(s,s') = (Q(s,s') - V(s))/\beta$.

Using this estimate, the Q-learning objective becomes:

$$\min_{Q} L(Q,V) = \alpha(1-\gamma)\, \mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \alpha \mathbb{E}_{(s,s')\sim d^E}\left[f^*\left(\gamma V(s') - Q(s,s')\right)\right] \\ + (1-\alpha) \mathbb{E}_{(s,s')\sim d^I}\left[\widetilde{f^*}\left(\gamma V(s') - Q(s,s')\right)\right]. \tag{6}$$

We optimize Q and V jointly: Q by minimizing $L(Q,V)$, and V by minimizing $J(V \mid Q)$. Crucially, $L(Q,V)$ is concave in Q, and $J(V \mid Q)$ is convex in V, forming a bi-concave/convex structure that ensures stable and convergent optimization.

Proposition 4. *Under any convex function f, $L(Q,V)$ is convex in Q, and the Extreme-V loss $J(V \mid Q)$ is convex in V.*

4.3 Policy Extraction

Typically, once the Q and V functions have been learned through Q-learning, a policy can be recovered using advantage-weighted behavior cloning (AW-BC) [31]. In our context, where we operate with an implicit policy $g(s' \mid s)$, the policy can be recovered by solving the following optimization:

$$\max_{g} \ \mathbb{E}_{(s,s')\sim d^I_{\text{mix}}(s,s')} \left[\exp\left(\tau(Q(s,s') - V(s))\right) \log g(s' \mid s)\right], \tag{7}$$

Here, $\tau > 0$ is a parameter controlling the sharpness of advantage weighting. We use $\mathcal{D}^I$ instead of the mixed dataset $\mathcal{D}^{E,I}_{\text{mix}}$ in (7), as the policy π is extracted solely from $\mathcal{D}^I$. This is sufficient under the common coverage assumption that $d^I(s,s') > 0$ whenever $d^E(s,s') > 0$, as adopted in prior works [19, 18, 27].

A limitation of objective (7) is that the value function $V(s)$ in our setting is only an approximation obtained via the extreme-V surrogate, which can introduce noise and bias in the computation of advantages. To address this, we propose an alternative approach based solely on the Q-function. The following proposition shows that this alternative objective can, in theory, recover the same optimal implicit policy as the original advantage-weighted BC formulation.

Proposition 5. *The following Q-weighted behavior cloning objective returns the same optimal implicit policy as the original advantage-weighted BC formulation:*

$$\max_{g} \ \mathbb{E}_{(s,s')\sim d^I} \left[\exp\left(\tau Q(s,s')\right) \log g(s' \mid s)\right]. \tag{8}$$

Given the learned implicit transition model $g(s' \mid s)$, existing approaches often recover the expert policy $\pi(a|s)$ by training an inverse dynamics model (IDM), denoted as $\mathcal{I}(a \mid s, s')$. This is typically done by optimizing the following objective: $\max_{\mathcal{I}} \ \mathbb{E}_{(s,a,s')\sim d^I} \left[\log \mathcal{I}(a \mid s, s')\right]$, and then defining the recovered policy as $\pi(a \mid s) = \sum_{s'} \mathcal{I}(a \mid s, s') g(s' \mid s)$. While intuitive, this two-step approach has several limitations. First, decoupling the learning of $g(s' \mid s)$ and the recovery of $\pi(a \mid s)$ via a separate inverse model introduces additional sources of bias. Second, training the inverse dynamics model $\mathcal{I}(a \mid s, s')$ typically requires a significant amount of high-quality data. When the offline dataset d^I contains a large proportion of low-quality or suboptimal data, the inverse model may be inaccurate—resulting in compounding approximation errors, as also noted in learning-from-observation (LfO) literature [18].

To address limitations of IDM-based recovery, we propose a single-stage policy extraction method that avoids training an inverse dynamics model. Our approach leverages the identity $g(s' \mid s) = \sum_a \mathcal{T}(s' \mid s, a)\pi(a \mid s)$. Using this, we rewrite the Q-weighted BC objective (8) as a direct optimization over π:

$$\max_{\pi} \ F(\pi) = \mathbb{E}_{(s,s')\sim d^I} \left[\exp\left(\tau Q(s,s')\right) \log \left(\sum_a \mathcal{T}(s' \mid s, a)\pi(a \mid s)\right)\right].$$

The objective $F(\pi)$, however, involves a log-sum over actions, making it difficult to optimize directly. We develop a tractable lower bound on this objective, which resembles a weighted behavior cloning loss over $\log \pi(a \mid s)$.

Proposition 6. *The objective $F(\pi)$ is lower-bounded by the following surrogate function $\widetilde{F}(\pi)$, up to an additive constant: $\widetilde{F}(\pi) = \mathbb{E}_{(s,s')\sim \mathcal{D}^I} \left[\exp\left(\tau Q(s,s')\right) \sum_a \mathcal{I}(a \mid s, s') \log \pi(a \mid s)\right]$.*

While $\widetilde{F}(\pi)$ is a lower bound of the original objective $F(\pi)$, maximizing this surrogate function still promotes the maximization of $F(\pi)$ in practice. The primary advantage of the surrogate objective $\widetilde{F}(\pi)$ is that it contains the term $\sum_a \mathcal{I}(a \mid s, s')\pi(a \mid s)$, which can be empirically approximated using offline samples, thus avoiding the need to learn the inverse dynamics. In particular, we can empirically approximate $\widetilde{F}(\pi)$ as:

$$\widetilde{F}(\pi) \approx \mathbb{E}_{(s,a,s')\sim d^I} \left[\exp\left(\tau Q(s,s')\right) \log \pi(a \mid s)\right],$$

where the expectation is taken over offline trajectories (s, a, s'). We note that a similar weighted behavior cloning formulation was used in [38], although without providing theoretical justification. Empirically, their results demonstrate that this single-stage approach can outperform the traditional two-step method involving inverse dynamics modeling.

5 Practical Algorithm

The common choices of f-divergence function in the literature can be KL or Pearson χ^2. In IOSTOM, we choose the χ^2 divergence function with its convex conjugate function $f^*(x) = \frac{x^2}{4} + x$. Our objective (6) becomes (complete derivation can be found in the Appendix B.8):

$$\min_Q L(Q,V) = (1-\gamma)\mathbb{E}_{d_0(s,s')}Q(s,s') + \gamma\mathbb{E}_{s\sim d^E}[V(s)] - \mathbb{E}_{s,s'\sim d^E}[Q(s,s')] + \frac{1}{4\alpha}\mathbb{E}_{s,s'\sim d^{E,I}_{mix}}[(\gamma V(s') - Q(s,s'))^2].$$

The $\min_Q -\alpha\,\mathbb{E}_{(s,s')\sim\mathcal{D}^E}[Q(s,s')]$ term in the above objective which effectively encourages maximizing the Q-values of expert transitions can lead to *unbounded* growth in Q, potentially resulting in learning instability. To address this issue, we adopt a technique from [2] that constrains the expert Q-values, and propose the following practical Q-learning objective (with derivation in Appendix B.9):

$$\widetilde{L}(Q,V) = (1-\gamma)\,\mathbb{E}_{d_0}[Q(s,s')] + \frac{1-\alpha}{4\alpha}\,\mathbb{E}_{d^I}\left[(\gamma V(s') - Q(s,s'))^2\right] + \frac{1}{4}\,\mathbb{E}_{d^E}\left[\left(Q(s,s') - \frac{2}{1-\gamma}\right)^2\right]. \tag{9}$$

Finally, to estimate the term $\mathbb{E}_{(s,s')\sim d_0}[Q(s,s')]$, we sample (s,s') pairs uniformly from the offline dataset rather than from a policy rollout. This empirical estimation, adopted in prior works [8, 38], helps reduce overfitting and improves the robustness of the learned policy by leveraging a diverse range of initial transitions. We present main steps of our IOSTOM algorithm in Algorithm 1.

Algorithm 1 IOSTOM

Input: Expert dataset D^E, suboptimal dataset D^I
Initialize Q, V functions and policy networks Q_ϕ, V_ω, π_θ
Set target network parameters $\phi' \leftarrow \phi$
for $t = 1, 2, \cdots, N$ **do**
  Sample mini-batches from D^E and D^I
  # Update V using $J(V,Q)$ in Equation (5)
  $\omega \leftarrow \omega - \eta\nabla_\omega \widetilde{J}(V_\omega|Q_{\phi'})$
  # Update Q using $\widetilde{L}(Q,V)$ in Equation (9)
  $\phi \leftarrow \phi - \eta\nabla_\phi L(Q_\phi, V_\omega)$
  # Update policy via weighted BC
  $w(s,s') \leftarrow \exp(\tau Q_{\phi'}(s,s'))$
  $\theta \leftarrow \theta + \eta\nabla_\theta \mathbb{E}_{(s,a,s')\sim d^I}[w(s,s')\log\pi_\theta(a\mid s)]$
  # Update target network
  $\phi' \leftarrow \lambda\phi + (1-\lambda)\phi'$
end for
Output: Imitation policy π_θ

6 Experiments

In this section, we compare IOSTOM with previous state-of-the-art approaches on diverse sets of environments and tasks from the D4RL benchmark [7], and real world data. Particularly, we aim to answer the following main questions: **(Q1)** Can IOSTOM outperform other baselines on standard LfO benchmarks? (Section 6.1) **(Q2)** Is our algorithm still robust with limited expert data? (Section 6.2) **(Q3)** How well IOSTOM perform when learning from experts of different dynamics? (Section 6.3) **(Q4)** What is the performance of IOSTOM on real-world instances (Section 6.4)? We also provide implementation details and additional experiments in the Appendix C.

Baselines and experimental setup We choose three SOTA LfO methods in the literature as our main baselines: SMODICE [27], PW-DICE [48], and DILO [38]. Both SMODICE and PW-DICE require learning a discriminator. The main difference between them is that SMODICE aims to minize the KL-divergence distance of state visitation distributions between learner and expert while PW-DICE uses Wasserstein distance [16] instead. DILO is the recent SOTA discriminator-free method for LfO. We train all algorithms for 1 million gradient steps with 5 random seeds and monitor the *normalized score* = $100 * \frac{\text{method score - random score}}{\text{expert score - random score}}$ [7] during training. The average normalized score of last 10 evaluations is used to assess the performance of different methods.

6.1 Offline IL from Observations

To answer the question **(Q1)**, we use the same offline LfO benchmark from DILO [39] with datasets constructed from the D4RL framework [7]. Specifically, we evaluate methods on 8 Mujoco envi-

		LfD approaches			LfO approaches				Expert
Suboptimal Dataset	Env	BC (expert data)	BC (full dataset)	ReCOIL	SMODICE	PW-DICE	DILO	IOSTOM	
random+ expert	hopper	4.52 $\pm$1.42	5.64 $\pm$4.83	108.18 $\pm$3.28	106.56 $\pm$0.53	108.09 $\pm$2.39	86.35 $\pm$38.00	**109.32** $\pm$1.08	111.33
	halfcheetah	2.2 $\pm$0.01	2.25 $\pm$0.00	80.20 $\pm$6.61	85.55 $\pm$1.39	86.11 $\pm$4.39	91.53 $\pm$0.27	**93.02** $\pm$0.40	88.83
	walker2d	0.86 $\pm$0.61	0.91 $\pm$0.5	102.16 $\pm$7.19	107.93 $\pm$1.26	107.48 $\pm$0.53	**108.31** $\pm$0.18	107.98 $\pm$0.20	106.92
	ant	5.17 $\pm$5.43	30.66 $\pm$1.35	126.74 $\pm$4.63	126.08 $\pm$0.73	**126.89** $\pm$1.17	125.39 $\pm$2.37	**128.19** $\pm$1.52	130.75
random+ few-expert	hopper	4.84 $\pm$3.83	3.0 $\pm$0.54	97.85 $\pm$17.89	58.30 $\pm$9.96	75.04 $\pm$14.21	**104.27** $\pm$4.74	**107.28** $\pm$3.92	111.33
	halfcheetah	-0.93 $\pm$0.35	2.24 $\pm$0.01	76.92 $\pm$7.53	3.19 $\pm$1.82	4.02 $\pm$1.74	43.65 $\pm$3.85	**88.77** $\pm$1.26	88.83
	walker2d	0.98 $\pm$0.83	0.74 $\pm$0.20	83.23 $\pm$19.00	3.93 $\pm$0.76	36.11 $\pm$9.19	**108.35** $\pm$0.13	**108.40** $\pm$0.21	106.92
	ant	0.91 $\pm$3.93	35.38 $\pm$2.66	67.14 $\pm$8.30	6.59 $\pm$6.86	99.90 $\pm$2.59	110.79 $\pm$1.33	**120.09** $\pm$5.17	130.75
medium+ expert	hopper	16.09 $\pm$12.80	59.25 $\pm$3.71	88.51 $\pm$16.73	55.74 $\pm$2.10	65.99 $\pm$8.05	108.22 $\pm$1.95	**110.20** $\pm$0.51	111.33
	halfcheetah	-1.79 $\pm$0.22	42.45 $\pm$0.42	81.15 $\pm$2.84	53.80 $\pm$4.18	58.74 $\pm$1.84	88.54 $\pm$3.77	**93.12** $\pm$0.32	88.83
	walker2d	2.43 $\pm$1.82	72.76 $\pm$3.82	108.54 $\pm$1.81	6.91 $\pm$0.71	105.41 $\pm$0.33	86.59 $\pm$12.32	**108.12** $\pm$0.13	106.92
	ant	0.86 $\pm$7.42	95.47 $\pm$10.37	120.36 $\pm$7.67	104.00 $\pm$3.62	108.14 $\pm$1.90	98.46 $\pm$1.44	**124.72** $\pm$3.49	130.75
medium few-expert	hopper	7.37 $\pm$1.13	46.87 $\pm$5.31	50.01 $\pm$10.36	53.50 $\pm$1.55	57.24 $\pm$3.03	96.95 $\pm$7.89	**108.96** $\pm$1.33	111.33
	halfcheetah	-1.15 $\pm$0.06	42.21 $\pm$0.06	75.96 $\pm$4.54	42.88 $\pm$0.63	27.85 $\pm$6.03	59.40 $\pm$6.80	**89.47** $\pm$0.82	88.83
	walker2d	2.02 $\pm$0.72	70.42 $\pm$2.86	91.25 $\pm$17.63	9.08 $\pm$3.67	75.22 $\pm$7.05	74.35 $\pm$0.80	**108.15** $\pm$0.43	106.92
	ant	-10.45 $\pm$1.63	81.63 $\pm$6.67	110.38 $\pm$10.96	88.20 $\pm$1.13	90.34 $\pm$2.56	90.77 $\pm$0.50	**120.36** $\pm$1.25	130.75
cloned+expert	pen	13.95 $\pm$11.04	34.94 $\pm$11.10	95.04 $\pm$4.48	15.71 $\pm$11.36	23.39 $\pm$4.56	26.48 $\pm$3.33	**82.77** $\pm$4.84	106.42
	door	-0.22 $\pm$0.05	0.011 $\pm$0.00	102.75 $\pm$4.05	1.57 $\pm$2.32	0.07 $\pm$0.14	93.29 $\pm$13.65	**102.77** $\pm$0.96	103.94
	hammer	2.41 $\pm$4.48	5.45 $\pm$7.84	95.77 $\pm$17.90	1.07 $\pm$1.30	1.29 $\pm$0.12	**91.80** $\pm$22.17	**94.59** $\pm$9.39	125.71
human+expert	pen	13.83 $\pm$10.76	90.76 $\pm$25.09	103.72 $\pm$2.90	58.62 $\pm$7.52	-2.56 $\pm$1.30	31.95 $\pm$7.43	**95.77** $\pm$8.91	106.42
	door	-0.03 $\pm$0.05	103.71 $\pm$1.22	104.70 $\pm$0.55	29.84 $\pm$12.17	0.15 $\pm$0.02	0.11 $\pm$0.40	**100.77** $\pm$1.68	103.94
	hammer	0.18 $\pm$0.14	122.61 $\pm$4.85	125.19 $\pm$3.29	33.28 $\pm$16.83	2.02 $\pm$0.77	6.93 $\pm$2.45	**93.34** $\pm$7.41	125.71
partial+expert	kitchen	2.5 $\pm$5.0	45.5 $\pm$1.87	60.0 $\pm$5.70	36.67 $\pm$5.77	12.33 $\pm$5.38	23.00 $\pm$25.87	**58.95** $\pm$2.27	75.0
mixed+expert	kitchen	2.2 $\pm$3.8	42.1 $\pm$1.12	52.0 $\pm$1.0	**48.33** $\pm$6.29	7.50 $\pm$4.16	29.17 $\pm$13.97	**46.45** $\pm$0.84	75.0

Table 2: Average normalized return over last 10 evaluations of IOSTOM against baselines on the D4RL suboptimal datasets with 1 expert trajectory. The mean and std are obtained over 5 random seeds. LfO methods with avg. perf within the std-dev of the top performing LfO approach is in **bold**.

.

ronments: 4 locomotion (Hopper, HalfCheetah, Walker2d, Ant) and 4 manipulation (Pen, Door, Hammer, Kitchen) [42]. Each task's expert dataset contains one trajectory. Suboptimal datasets for locomotion mix D4RL 'random' or 'medium' data with 200 ('expert') or 30 ('few-expert') expert trajectories. For manipulation, D4RL non-expert datasets ('mixed' and 'partial' for Kitchen; 'human' and 'cloned' for others) are mixed with up to 30 expert trajectories. This results in 24 diverse tasks for comparing IOSTOM against baselines, with manipulation tasks being more challenging due to larger state spaces. More details on environment and dataset are included in the Appendix.

Table 2 presents results for IOSTOM and baselines. We also include the results of some Learning from Demonstration (LfD) methods such as Behavior Cloning (BC) and ReCOIL [39] to serve as the reference upper bound of LfO methods because they have access to expert actions during learning. We choose these two approaches as ReCOIL is the SOTA offline LfD method while BC is the most popular IL algorithm. Their results are taken directly from ReCOIL's paper which uses a similar setting. As shown in Table 2, IOSTOM leads on 23/24 tasks, only marginally underperforming DILO on 'walker2d random+expert' while still matching expert performance. Discriminator-based methods (SMODICE, PW-DICE) degrade significantly with few expert examples or on high-dimensional manipulation tasks due to discriminator overfitting. While DILO's discriminator-free nature mitigates this, it still struggles in 'few-expert' settings (e.g., 'halfcheetah') and 'human+expert' tasks where training can diverge (see Appendix for further discussion). BC methods with access to expert actions also exhibit poor performance on most tasks. Notably, IOSTOM's performance is comparable to, and sometimes surpasses, ReCOIL on locomotion tasks, showcasing its effectiveness and potential to bridge the gap between LfD and LfO.

6.2 LfO with subsampled expert

This section focuses on benchmarking the sample efficiency of our approach (Question (**Q3**)). We adapt the subsampled expert trajectory setting from LfD literature [13, 20]) to construct a subsampled state-only expert dataset. Specifically, expert trajectories are sub-sampled by keeping a transition every 20 time steps (i.e. subsampling rate is 20) starting with a random offset. This process will create incomplete expert trajectories which makes both BC and DICE method like ValueDICE [21] fail as shown in [51]. This setting may not be valid in case of DILO because it requires the triplet (s, s', s'') which is equivalent to two transitions inside action-labeled expert dataset; we still adapt the 2-transition version of the subsampling procedure only for DILO. LobsDICE also considers the similar setting for LfO like us on locomotion tasks, but they construct D^E using 50 sub-sampled expert trajectories, which means using $\frac{50}{20} = 2.5$ times of total transitions of an expert trajectory. This makes this setting still easy to deal with for both our approach and baselines. Therefore, we

Suboptimal Dataset	Env	SMODICE (full)	SMODICE (sub)	PW-DICE (full)	PW-DICE (sub)	DILO (full)	DILO (sub)	IOSTOM (full)	IOSTOM (sub)
random+ expert	hopper	106.56 ±0.53	108.33 ±0.43	108.09 ±2.39	97.35 ±2.72	86.35 ±38.00	13.25 ±12.95	109.32 ±1.08	**109.94** ±0.46
	halfcheetah	85.55 ±1.39	78.63 ±5.04	86.11 ±4.39	37.95 ±9.94	91.53 ±0.27	92.06 ±0.29	93.02 ±0.40	**93.23** ±0.24
	walker2d	107.93 ±1.26	107.46 ±0.51	107.48 ±0.53	101.59 ±1.11	108.31 ±0.18	41.98 ±35.80	107.98 ±0.20	**108.01** ±0.16
	ant	126.08 ±0.73	124.13 ±3.74	126.89 ±1.17	112.99 ±6.28	125.39 ±2.37	30.27 ±2.47	128.19 ±1.52	**126.23** ±2.87
random+ few-expert	hopper	58.30 ±9.96	58.44 ±10.26	75.04 ±14.21	48.30 ±20.09	104.27 ±4.74	92.52 ±10.81	107.28 ±3.92	**105.20** ±5.90
	halfcheetah	3.19 ±1.82	3.06 ±1.29	4.02 ±1.74	3.91 ±1.17	43.65 ±3.85	44.22 ±4.09	88.77 ±1.26	**86.09** ±3.82
	walker2d	3.93 ±0.76	4.78 ±2.32	36.11 ±9.19	26.29 ±10.97	108.35 ±0.13	33.69 ±5.97	108.40 ±0.21	**104.32** ±8.62
	ant	6.59 ±6.86	6.33 ±3.12	99.90 ±2.59	82.81 ±8.88	110.79 ±1.33	31.91 ±0.65	120.09 ±5.17	**123.83** ±4.29
medium+ expert	hopper	55.74 ±2.10	54.24 ±2.47	65.99 ±8.05	63.03 ±7.24	108.22 ±1.95	54.42 ±0.47	110.20 ±0.51	**109.72** ±0.92
	halfcheetah	53.80 ±4.18	50.06 ±1.87	58.74 ±1.84	62.78 ±2.70	88.54 ±3.77	42.61 ±0.21	93.12 ±0.32	**92.97** ±0.35
	walker2d	6.91 ±0.71	1.77 ±1.63	105.41 ±0.33	82.90 ±18.45	86.59 ±12.32	83.41 ±24.57	108.12 ±0.13	**108.59** ±0.17
	ant	104.00 ±3.62	99.52 ±1.18	108.14 ±1.90	110.68 ±4.20	98.46 ±1.44	105.73 ±5.35	124.72 ±3.49	**124.06** ±1.66
medium few-expert	hopper	53.50 ±1.55	54.26 ±1.09	57.24 ±3.03	50.51 ±4.21	96.95 ±7.89	55.50 ±1.33	108.96 ±1.33	**107.21** ±1.69
	halfcheetah	42.88 ±0.63	42.88 ±0.74	27.85 ±6.03	11.99 ±5.61	59.40 ±6.80	53.53 ±7.52	89.47 ±0.82	**87.45** ±3.67
	walker2d	9.08 ±3.67	3.05 ±2.22	75.22 ±7.05	52.95 ±11.10	74.35 ±0.80	54.55 ±2.89	108.15 ±0.43	**108.45** ±0.30
	ant	88.20 ±1.13	88.80 ±5.18	90.34 ±2.56	89.69 ±2.23	90.77 ±0.50	90.90 ±1.49	120.36 ±1.25	**117.29** ±1.85

Table 3: Comparison of normalized returns obtained by different offline LfO methods on expert dataset with 1 expert trajectory denoted as (full) or 5 subsampled expert trajectories (subsampling rate is 20) denoted as (sub). The mean and std are obtained over 5 random seeds. Methods on subsampled expert dataset with avg. perf within the std-dev of the top performing method is in **bold**. Methods with greater than 5% performance decrease on subsampled expert datasets are highlighted in blue.

construct D^E from 5 subsampled trajectories only (i.e. 0.25x total transitions of an expert trajectory) and evaluate all LfO methods on locomotion tasks with the same suboptimal dataset in Section 6.1.

Table 3 shows the comparison results on the subsampled setting. IOSTOM continues to outperform all baselines on these challenging tasks. Furthermore, its performance does not change much compared to using complete expert trajectory even when the total number of expert samples is reduced by 4 times. SMODICE is also robust on 12/16 tasks but its performance on 'few-expert' setting is still poor. Both DILO and PW-DICE face a large drop (>5 %) on the performance of most tasks in the scenario of less samples and incomplete trajectories.

6.3 LfO with mismatched expert

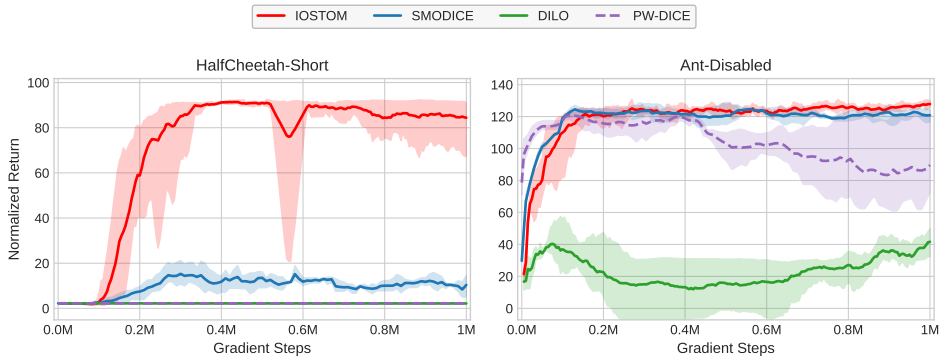

Figure 1: Comparison results for LfO with mismatched experts

To evaluate IOSTOM performance when learning from experts of different dynamics, we adopt SMODICE's mismatched dynamics setting. We test on 'HalfCheetah-Short' (halved torso) and 'Ant-Disabled' (partially amputated front leg) (See Appendix H of SMODIE [27] for illustration), using one expert trajectory from these modified agents. The suboptimal dataset remains the 'random+expert' data (Section 6.1) from the original agents. This setting creates a clear mismatch between expert and interaction datasets. Figure 1 shows IOSTOM outperforming baselines on these challenging tasks, while DILO performs worst. The poor performance of DILO can be due to the use of visitation distribution $d(s, s', a')$ in its objective which matches the wrong a' in the mismatched expert dataset.

6.4 LfO for marine navigation

We next test IOSTOM in a real-world domain, the maritime navigation problem. Our goal is to learn IL policies that can behave like human experts (ship pilots) for navigating vessels (mainly large tankers and cargos). These polices offer significant benefits for operational safety and efficiency.

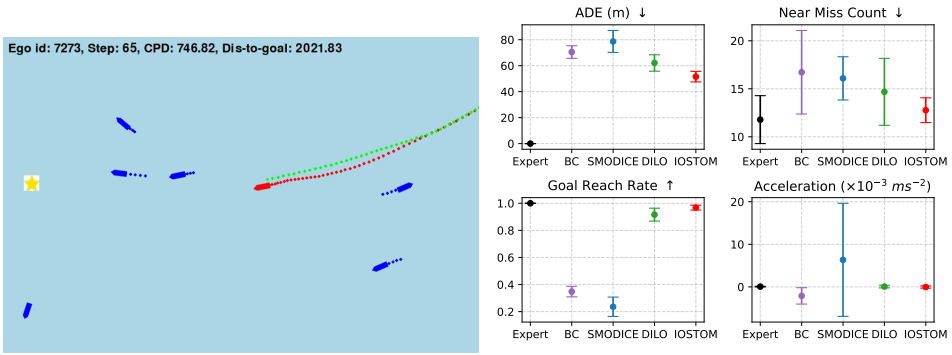

Figure 2: **Left**: Visualization of an episode from our maritime simulator. Blue vessels follows their historical trajectory; red vessel is controlled by IL policy (destination $\star$ marker), green dots denote historical trajectory of ego vessel. **Right**: Comparison results on various performance metrics. Details on metrics are in the Appendix.

For instance, they can be integrated into Vessel Traffic Information Systems to provide port watch operators with accurate short-term predictions of vessel movements, especially in congested ports such as Singapore strait. Learned IL policies also enable to do what-if analysis such as how would safety be affected when there is traffic surge (e.g., by simulating additional vessel arrivals and assigning them IL policies). Prior RL-based approaches to maritime traffic management [40, 41] rely on online learning, requiring costly simulator interactions and accurate simulation of vessel dynamics. In contrast, our IL method directly learns vessel behavior from large offline datasets, offering easier and more accurate modeling as shown in our results (see later metrics such as ADE, goal rate).

We collect a large amount of historical navigation data ($\sim$ 2 years) of vessels operating in a hotspot region in Singpaore strait (among top 5 busiest ports) recorded in the Automatic Identification System (AIS). We use these data with **ShipNaviSim**, a data-driven maritime traffic simulator [32] to construct a realistic environment featuring an IOSTOM-controlled ego agent (red) and log-play agents (blue) that follow their actual trajectories, as illustrated in Figure 2. The ego agent controlled by IL policy tries to reach the goal ('$\star$') while avoiding collisions with other log-play agents. The ego agent can also observe past states (blue and red dots) of its and close surrounding agents (its observation space). Because AIS data does not contain any action information, we use an inverse kinematics model (IVM) to construct the action space and generate action for AIS data. The state-only expert dataset in this setting is easy to obtain due to the action-free nature of AIS data. We generate the suboptimal dataset by adding random noise action-labeled expert trajectories. Further details about environment and dataset generation can be found in the Appendix.

We evaluate our approach in maritime navigation setting against BC, SMODICE, and DILO. Results are shown in Figure 2 using metrics relevant to this domain, introduced in [32], which reflect how well the learned agent imitates expert behavior. **ADE** (Average Displacement Error; lower is better) measures how far, on average, the agent's trajectory deviates from the expert's. **Goal reach rate** (higher is better) indicates how often the ship reaches the goal. **Near-miss count** captures the number of close-quarter situations, defined as scenarios where two ships come close to each other posing a collision risk; lower values indicate reduced collision risk, and **average acceleration** should closely match that of the expert. Mean and standard deviations are over 5 seeds for each method. Our approach outperforms across all three baselines in ADE, near-miss count, and goal reach rate, while maintaining an acceleration profile similar to the expert. DILO is the second-best performer. SMODICE struggles due to high-dimensional observation space—which includes nearby ships and trajectory history; leading to a poorly trained state discriminator and worse performance than BC.

7 Conclusion

We presented IOSTOM, a discriminator-free Q-learning framework for offline imitation learning from observations. By learning an implicit policy in the form of state-to-state transitions and matching joint state visitation distributions, our method avoids reliance on action labels for value function learning (Q-learning) and eliminates the need for inverse dynamics models in policy extraction. Extensive experimental results and ablation studies demonstrate that IOSTOM achieves strong empirical performance and improves sample efficiency compared to prior approaches.

Acknowledgments and Disclosure of Funding

This research/project is supported by the National Research Foundation, Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-017).

References

[1] Siddhant Agarwal, Ishan Durugkar, Peter Stone, and Amy Zhang. f-policy gradients: A general framework for goal-conditioned rl using f-divergences. *Advances in Neural Information Processing Systems*, 36:12100–12123, 2023.

[2] Firas Al-Hafez, Davide Tateo, Oleg Arenz, Guoping Zhao, and Jan Peters. LS-IQ: Implicit reward regularization for inverse reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

[5] Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pages 1458–1467. PMLR, 2017.

[6] Anish Abhijit Diwan, Julen Urain, Jens Kober, and Jan Peters. Noise-conditioned energy-based annealed rewards (NEAR): A generative framework for imitation learning from observation. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

[8] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039, 2021.

[9] Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent RL without entropy. In *The Eleventh International Conference on Learning Representations*, 2023.

[10] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on robot learning*, pages 1259–1277. PMLR, 2020.

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *Advances in Neural Information Processing Systems*, 36:7730–7742, 2023.

[13] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[14] Bo-Ruei Huang, Chun-Kai Yang, Chun-Mao Lai, Dai-Jie Wu, and Shao-Hua Sun. Diffusion imitation from observation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[15] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[16] Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

[17] Rahul Kidambi, Jonathan Chang, and Wen Sun. Mobile: Model-based imitation learning from observation alone. *Advances in Neural Information Processing Systems*, 34:28598–28611, 2021.

[18] Geon-Hyeong Kim, Jongmin Lee, Youngsoo Jang, Hongseok Yang, and Kee-Eung Kim. Lobsdice: Offline learning from observation via stationary distribution correction estimation. *Advances in Neural Information Processing Systems*, 35:8252–8264, 2022.

[19] Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

[20] Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

[21] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2020.

[22] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

[23] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

[24] Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14128–14147, 2022.

[25] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[26] Jason Yecheng Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. Offline goal-conditioned reinforcement learning via f-advantage regression. *Advances in Neural Information Processing Systems*, 35:310–323, 2022.

[27] Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pages 14639–14663. PMLR, 2022.

[28] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

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

[30] Tianwei Ni, Harshit Sikchi, Yufei Wang, Tejus Gupta, Lisa Lee, and Ben Eysenbach. f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*, pages 529–551. PMLR, 2021.

[31] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[32] Quang Anh Pham, Janaka Chathuranga Brahmanage, and Akshat Kumar. Shipnavisim: Data-driven simulation for real-world maritime navigation. In *International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, page 1641–1649, Richland, SC, 2025. International Foundation for Autonomous Agents and Multiagent Systems.

[33] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[34] Nived Rajaraman, Lin Yang, Jiantao Jiao, and Kannan Ramchandran. Toward the fundamental limits of imitation learning. *Advances in Neural Information Processing Systems*, 33:2914–2924, 2020.

[35] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3(1):297–330, 2020.

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

[37] Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

[38] Harshit Sikchi, Caleb Chuck, Amy Zhang, and Scott Niekum. A dual approach to imitation learning from observations with offline datasets. In *8th Annual Conference on Robot Learning*, 2024.

[39] Harshit Sikchi, Qinqing Zheng, Amy Zhang, and Scott Niekum. Dual RL: Unification and new methods for reinforcement and imitation learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[40] Arambam James Singh, Akshat Kumar, and Hoong Chuin Lau. Hierarchical multiagent reinforcement learning for maritime traffic management. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 1278–1286, 2020.

[41] Arambam James Singh, Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Multiagent decision making for maritime traffic management. In *AAAI Conference on Artificial Intelligence*, pages 6171–6178, 2019.

[42] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[43] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4950–4957. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[44] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

[45] Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6325–6331. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[46] Lu Wang, Wenchao Yu, Xiaofeng He, Wei Cheng, Martin Renqiang Ren, Wei Wang, Bo Zong, Haifeng Chen, and Hongyuan Zha. Adversarial cooperative imitation learning for dynamic treatment regimes. In *Proceedings of The Web Conference 2020*, pages 1785–1795, 2020.

[47] Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. Offline RL with no OOD actions: In-sample learning via implicit value regularization. In *The Eleventh International Conference on Learning Representations*, 2023.

[48] Kai Yan, Alex Schwing, and Yu-Xiong Wang. Offline imitation from observation via primal wasserstein state occupancy matching. In *Forty-first International Conference on Machine Learning*, 2024.

[49] Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. *Advances in neural information processing systems*, 32, 2019.

[50] Zhuangdi Zhu, Kaixiang Lin, Bo Dai, and Jiayu Zhou. Off-policy imitation learning from observations. *Advances in neural information processing systems*, 33:12402–12413, 2020.

[51] Li Ziniu, Xu Tian, Yu Yang, and Luo Zhi-Quan. Rethinking valuedice - does it really improve performance? In *ICLR Blog Track*, 2022. https://iclr-blog-track.github.io/2022/03/25/rethinking-valuedice/.

APPENDIX

Contents

A Limitations and Future Work

Despite its strong empirical performance, IOSTOM has some limitations. First, like most LfO methods, it assumes access to high-quality consecutive state pairs (s, s'), which may not always be available in real-world datasets. Second, we assume that actions are fully observable in the sub-optimal dataset, which might not hold in practice. While these limitations are beyond the scope of this work, they highlight important directions for future research.

B Missing Proofs and Derivations

B.1 Proof of Proposition 1

Proposition. *The joint visitation distribution $d^g(s, s')$ can be expressed as:*

$$d^g(s, s') = (1-\gamma)d_0(s)g(s' \mid s) + \gamma g(s' \mid s) \sum_{\overline{s}} d^g(\overline{s}, s) \tag{10}$$

Proof. We recall that $d^\pi(s, a) = (1-\gamma)d_0(s)\pi(a \mid s) + \gamma \sum_{s',a'} d^\pi(s', a')\mathcal{T}(s \mid s', a')\pi(a \mid s)$. Therefore, we have:

$$\begin{aligned}
d^g(s, s') &= \sum_a d^\pi(s, a)\mathcal{T}(s' \mid s, a) \\
&= \sum_a (1-\gamma)d_0(s)\pi(a \mid s)\mathcal{T}(s' \mid s, a) + \sum_a \gamma\mathcal{T}(s' \mid s, a) \sum_{\overline{s},\overline{a}} d^\pi(\overline{s}, \overline{a})\mathcal{T}(s \mid \overline{s}, \overline{a})\pi(a \mid s) \\
&= (1-\gamma)d_0(s)g(s' \mid s) + \gamma g(s' \mid s) \sum_{\overline{s},\overline{a}} d^\pi(\overline{s}, \overline{a})\mathcal{T}(s \mid \overline{s}, \overline{a}) \\
&= (1-\gamma)d_0(s)g(s' \mid s) + \gamma g(s' \mid s) \sum_{\overline{s}} d^g(\overline{s}, s).
\end{aligned}$$

as desired. □

B.2 Proof of Proposition 2

Proposition. *The constrained optimization problem in Equation* (2) *is equivalent to the following unconstrained max-min problem:*

$$\begin{aligned}
\max_g \min_{Q(s,s')} \Big\{ &\alpha(1-\gamma)\,\mathbb{E}_{(s,s')\sim d_0}\left[Q(s, s')\right] + \mathbb{E}_{(s,s')\sim d^{E,I}_{mix}}\left[f^*\left(\gamma\mathbb{E}_{s''\sim g(\cdot|s')}\left[Q(s', s'')\right] - Q(s, s')\right)\right] \\
&- (1-\alpha)\,\mathbb{E}_{(s,s')\sim d^I}\left[\gamma\mathbb{E}_{s''\sim g(\cdot|s')}\left[Q(s', s'')\right] - Q(s, s')\right] \Big\},
\end{aligned}$$

where $Q(s, s')$ are the Lagrange multipliers and f^* denotes the convex conjugate of a chosen convex function f.

Proof. We recall that the primal formulation in Equation (2) is as follows:

$$\begin{aligned}
\max_{g, d\geq 0} \quad & -\mathbb{D}_f(d^I_{mix}(s, s') \| d^{E,I}_{mix}(s, s')) \\
s.t. \quad & d(s, s') = (1-\gamma)d_0(s)g(s'|s) + \gamma g(s'|s) \sum_{\overline{s}} d(\overline{s}, s)
\end{aligned}$$

We first apply duality on the inner maximization problem of the above formulation:

$$\begin{aligned}
&\max_{g,d\geq 0}\min_{Q(s,s')} -\mathbb{D}_f(d^I_{mix}(s,s')\|d^{E,I}_{mix}(s,s')) \\
&+\alpha\sum_{s,s'}Q(s,s')\left((1-\gamma)d_0(s).g(s'|s)+\gamma g(s'|s)\sum_{\overline{s}}d(\overline{s},s)-d(s,s')\right) \qquad (11)\\
&=\max_{\pi,d\geq 0}\min_{Q(s,s')}\alpha(1-\gamma)\mathbb{E}_{d_0(s),g(s'|s)}\left[Q\big(s,s'\big)\right] \\
&+\alpha\mathbb{E}_{s,s'\sim d}\left[\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right]-\mathbb{D}_f(d^I_{mix}(s,s')\|d^{E,I}_{mix}(s,s')) \qquad (12)
\end{aligned}$$

Step (11) to (12) is equivalent to changing the following order of summation:

$$\begin{aligned}
&\sum_{s,s'}Q(s,s')g(s'|s)\sum_{\overline{s}}d(\overline{s},s)\\
&=\sum_{\overline{s},s}d(\overline{s},s)\sum_{s'}Q(s,s')g(s'|s)\\
&=\sum_{s,s'}d(s,s')\sum_{s''}Q(s',s'')g(s''|s')
\end{aligned}$$

By adding and subtracting another term below, (12) becomes:

$$\begin{aligned}
&=\max_{g,d\geq 0}\min_{Q(s,s')}\alpha(1-\gamma)\mathbb{E}_{d_0(s),g(s'|s)}\left[Q\big(s,s'\big)\right]\\
&+\alpha\mathbb{E}_{s,s'\sim d}\left[\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right]\\
&+(1-\alpha)\mathbb{E}_{s,s'\sim d^I}\left[\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right]\\
&-(1-\alpha)\mathbb{E}_{s,s'\sim d^I}\left[\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right]\\
&-\mathbb{D}_f(d^I_{mix}(s,s')\|d^{E,I}_{mix}(s,s')) \qquad (13)
\end{aligned}$$

We can swap $\max_d$ and $\min_Q$ in (13) due to strong duality.

$$\begin{aligned}
(13)=&\max_{g}\min_{Q(s,s')}\max_{d^I_{mix}(s,s')\geq 0}\alpha(1-\gamma)\mathbb{E}_{d_0(s),g(s'|s)}\left[Q\big(s,s'\big)\right]\\
&+\mathbb{E}_{s,s'\sim d^I_{mix}}\left[\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right]-\mathbb{D}_f(d^I_{mix}(s,s')\|d^{E,I}_{mix}(s,s'))\\
&-(1-\alpha)\mathbb{E}_{s,s'\sim d^I}\left[\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right] \qquad (14)\\
=&\max_{g}\min_{Q(s,s')}\alpha(1-\gamma)\mathbb{E}_{d_0(s),g(s'|s)}\left[Q\big(s,s'\big)\right]\\
&+\mathbb{E}_{s,s'\sim d^{E,I}_{mix}}\left[f^*\left(\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right)\right]\\
&-(1-\alpha)\mathbb{E}_{s,s'\sim d^I}\left[\gamma\sum_{s''}g(s''|s')Q(s',s'')-Q(s,s')\right] \qquad (15)
\end{aligned}$$

where f^* is convex conjugate of convex f-divergence function. (14) to (15) can be proved by the following equation using the interchangeability principle [5]:

$$\max_{d^I_{mix}(s,s')\geq 0} \mathbb{E}_{s,s'\sim d^I_{mix}}\left[\gamma\sum_{s''} g(s''|s')Q(s',s'') - Q(s,s')\right] - \mathbb{D}_f(d^I_{mix}(s,s')\|d^{E,I}_{mix}(s,s'))$$
$$= \max_{d^I_{mix}(s,s')\geq 0} \mathbb{E}_{s,s'\sim d^{E,I}_{mix}}\left[\frac{d^I_{mix}(s,s')}{d^{E,I}_{mix}(s,s')}\left(\gamma\sum_{s''} g(s''|s')Q(s',s'') - Q(s,s')\right) - f\left(\frac{d^I_{mix}(s,s')}{d^{E,I}_{mix}(s,s')}\right)\right]$$
$$= \mathbb{E}_{s,s'\sim d^{E,I}_{mix}}\left[f^*\left(\gamma\sum_{s''} g(s''|s')Q(s',s'') - Q(s,s')\right)\right]$$

Finally, the objective (15) is the unconstrained dual problem of Equation (2). □

B.3 Complete derivation of transforming objective function (3) to (4)

$$\alpha(1-\gamma)\mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \mathbb{E}_{(s,s')\sim d^{E,I}{}_{\text{mix}}}\left[f^*\left(\gamma\mathbb{E}_{s''\sim g(\cdot|s')}\left[Q(s',s'')\right] - Q(s,s')\right)\right]$$
$$-(1-\alpha)\,\mathbb{E}_{(s,s')\sim d^I}\left[\gamma\mathbb{E}_{s''\sim g(\cdot|s')}\left[Q(s',s'')\right] - Q(s,s')\right] \tag{16}$$
$$= \alpha(1-\gamma)\mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \mathbb{E}_{(s,s')\sim d^{E,I}_{\text{mix}}}\left[f^*\left(\gamma Q(s',g) - Q(s,s')\right)\right]$$
$$-(1-\alpha)\,\mathbb{E}_{(s,s')\sim d^I}\left[\gamma Q(s',g) - Q(s,s')\right] \tag{17}$$
$$= \alpha(1-\gamma)\mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \alpha\mathbb{E}_{(s,s')\sim d^E}\left[f^*\left(\gamma Q(s',g) - Q(s,s')\right)\right]$$
$$+(1-\alpha)\mathbb{E}_{(s,s')\sim d^I}\left[f^*\left(\gamma Q(s',g) - Q(s,s')\right)\right] - (1-\alpha)\,\mathbb{E}_{(s,s')\sim d^I}\left[\gamma Q(s',g) - Q(s,s')\right] \tag{18}$$
$$= \alpha(1-\gamma)\mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \alpha\mathbb{E}_{(s,s')\sim d^E}\left[f^*\left(\gamma Q(s',g) - Q(s,s')\right)\right]$$
$$+(1-\alpha)\mathbb{E}_{(s,s')\sim d^I}\left[f^*\left(\gamma Q(s',g) - Q(s,s')\right) - \left(\gamma Q(s',g) - Q(s,s')\right)\right] \tag{19}$$
$$= \alpha(1-\gamma)\mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \alpha\mathbb{E}_{(s,s')\sim d^E}\left[f^*\left(\gamma Q(s',g) - Q(s,s')\right)\right]$$
$$+(1-\alpha)\,\mathbb{E}_{(s,s')\sim d^I}\left[\widetilde{f^*}\left(\gamma Q(s',g) - Q(s,s')\right)\right] \tag{20}$$

(16) to (17) by defining $Q(s,g) = \mathbb{E}_{s'\sim g(\cdot|s)}[Q(s,s')]$. (17) to (18) due to $d^{E,I}_{\text{mix}}(s,s') = \alpha d^E(s,s') + (1-\alpha)d^I(s,s')$. (19) to (20) by defining $\widetilde{f^*}(t) = f^*(t) - t$.

B.4 Proof of Proposition 3

Proposition. $\max_g\{V^g_Q(s)\}$ *can be approximated via the following Extreme-V objective:*

$$\min_V\left\{J(V \mid Q) = \mathbb{E}_{(s,s')\sim d^{E,I}_{mix}}\left[\exp(\omega(s,s')) + \omega(s,s') - 1\right]\right\}.$$

where $\omega(s,s') = (Q(s,s') - V(s))/\beta$.

Proof. Recall that:

$$V^g_Q(s) = \mathbb{E}_{s'\sim g(\cdot|s)}\left[Q(s,s') - \beta\log\frac{g(s' \mid s)}{\mu(s' \mid s)}\right],$$

which is the expected reward under transition distribution $g(\cdot \mid s)$, regularized by the KL divergence from a reference distribution $\mu(\cdot \mid s)$. Moreover, the problem $\max_g\left\{V^g_Q(s)\right\}$ is a classic entropy-regularized expected reward maximization problem. The optimal solution has a closed form:

$$\max_g\left\{V^g_Q(s)\right\} = \beta\log\sum_{s'}\mu(s' \mid s)\exp\left(\frac{Q(s,s')}{\beta}\right). \tag{21}$$

We now write the function $J(V \mid Q)$ as:

$$J(V \mid Q) = \sum_{(s,s')}\mu(s' \mid s)\left[\exp\left(\frac{Q(s,s') - V(s)}{\beta}\right) + \frac{Q(s,s') - V(s)}{\beta} - 1\right].$$

For any state s, and fixed Q, the function $J(V \mid Q)$ is convex in $V(s)$ because:

- The exponential function $\exp\left(\frac{Q(s,s')-V(s)}{\beta}\right)$ is convex in $V(s)$,
- The linear term $(Q(s,s')-V(s))/\beta$ is also convex (affine),
- The sum and non-negative weights preserve convexity.

To find the minimum of $J(V \mid Q)$ with respect to V, we take the derivative with respect to $V(s)$ and set it to zero:

$$\frac{\partial J(V \mid Q)}{\partial V(s)} = \sum_{s'} \mu(s' \mid s) \left[-\frac{1}{\beta} \exp\left(\frac{Q(s,s') - V(s)}{\beta} \right) - \frac{1}{\beta} \right] = 0.$$

Rewriting:

$$\sum_{s'} \mu(s' \mid s) \exp\left(\frac{Q(s,s') - V(s)}{\beta} \right) = \sum_{s'} \mu(s' \mid s).$$

We have $\sum_{s'} \mu(s' \mid s) = 1$, this gives:

$$\sum_{s'} \mu(s' \mid s) \exp\left(\frac{Q(s,s') - V(s)}{\beta} \right) = 1.$$

Bringing the constant outside the exponential:

$$\exp\left(-\frac{V(s)}{\beta}\right) \sum_{s'} \mu(s' \mid s) \exp\left(\frac{Q(s,s')}{\beta}\right) = 1,$$

$$\Rightarrow \exp\left(-\frac{V(s)}{\beta}\right) = \frac{1}{\sum_{s'} \mu(s' \mid s) \exp\left(\frac{Q(s,s')}{\beta}\right)},$$

Taking the logarithm of both sides and solving for $V(s)$, we obtain the closed-form solution to $\min_V J(V|Q)$ as:

$$V^*(s) = \beta \log \sum_{s'} \mu(s' \mid s) \exp\left(\frac{Q(s,s')}{\beta} \right). \tag{22}$$

Combined (21) and (22) we get:

$$V^*(s) = \max_g \{V_Q^g(s)\}$$

as desired. □

B.5 Proof of Proposition 4

Proposition. *Under any convex function f, $L(Q,V)$ is concave in Q, and the Extreme-V loss $J(V \mid Q)$ is convex in V.*

Proof. We rewrite the objective $L(Q,V)$ as:

$$\min_Q L(Q,V) = \alpha(1-\gamma)\, \mathbb{E}_{(s,s')\sim\mathcal{D}_0} \left[Q(s,s') \right] + \alpha \mathbb{E}_{(s,s')\sim\mathcal{D}^E} \left[f^* \left(\gamma V(s') - Q(s,s') \right) \right]$$
$$+ (1-\alpha) \mathbb{E}_{(s,s')\sim\mathcal{D}^I} \left[\widetilde{f^*} \left(\gamma V(s') - Q(s,s') \right) \right].$$

We now analyze the convexity of $L(Q,V)$ with respect to Q. Note the following:

- The first term, $\mathbb{E}_{\mathcal{D}_0}[Q(s,s')]$, is linear in Q, and hence convex.
- The functions f^* and $\widetilde{f^*}$ are convex (as they are convex conjugates of proper convex functions).
- The composition of a convex function with an affine function (i.e., $\gamma V(s') - Q(s,s')$) is convex in Q.

- Expectations of convex functions preserve convexity.

Therefore, each term in $L(Q, V)$ is convex in Q, and the entire objective $L(Q, V)$ is convex in Q, as desired.

The convexity of $J(V \mid Q)$ in V follows directly from the discussion in the proof of Proposition (3). □

B.6 Proof of Proposition 5

Proposition. *The following Q-weighted behavior cloning objective returns the same optimal implicit policy as the original advantage-weighted BC formulation:*

$$\max_g \ \mathbb{E}_{(s,s')\sim d^I} \left[\exp\left(\tau Q(s, s')\right) \log g(s' \mid s)\right]$$

Proof. We write the objective function as:

$$F(g) = \sum_{(s,s')} \mu^I(s' \mid s) \exp\left(\tau Q(s, s')\right) \log g(s' \mid s),$$

where $\mu^I(s' \mid s)$ is the state-to-state transition probability (i.e., the implicit behavior policy) for the dataset $\mathcal{D}^I$. For each fixed state s, the expression

$$\sum_{s'} \mu^I(s' \mid s) \exp\left(\tau Q(s, s')\right) \log g(s' \mid s)$$

is a weighted log-likelihood, where the weights $\mu^I(s' \mid s) \exp(\tau Q(s, s'))$ are known. Maximizing this with respect to $g(\cdot \mid s)$ under the constraint that $g(\cdot \mid s)$ is a valid probability distribution (i.e., $\sum_{s'} g(s' \mid s) = 1$) leads to a standard result from maximum likelihood estimation with importance weights. The closed-form solution is:

$$g^*(s' \mid s) = \frac{\mu^I(s' \mid s) \exp\left(\tau Q(s, s')\right)}{\sum_{s''} \mu^I(s'' \mid s) \exp\left(\tau Q(s, s'')\right)}.$$

We now consider the advantage-weighted behavior cloning objective:

$$\max_g \ \mathbb{E}_{(s,s')\sim d^I} \left[\exp\left(\tau(Q(s, s') - V(s))\right) \log g(s' \mid s)\right],$$

In a similar fashion to soft behavior cloning, this yields the following closed-form optimal "implicit policy":

$$g^{**}(s' \mid s) = \frac{\mu^I(s' \mid s) \exp\left(\tau(Q(s, s') - V(s))\right)}{\sum_y \mu^I(y \mid s) \exp\left(\tau(Q(s, y) - V(s))\right)},$$

where $V(s)$ appears in both the numerator and denominator and thus cancels out. This simplifies the expression and leads to:

$$g^*(s' \mid s) = g^{**}(s' \mid s),$$

indicating the equivalence between the advantage-weighted behavior cloning and the Q-weighted behavior cloning formulations.

□

B.7 Proof of Proposition 6

Proposition. *The objective $F(\pi)$ is lower-bounded by the following surrogate function $\widetilde{F}(\pi)$, up to an additive constant: $\widetilde{F}(\pi) = \mathbb{E}_{(s,s')\sim d^I} \left[\exp\left(\tau Q(s, s')\right) \sum_a \mathcal{I}(a \mid s, s') \log \pi(a \mid s)\right]$.*

Proof. We write the objective function as:

$$F(\pi) = \mathbb{E}_{(s,s')\sim \mathcal{D}^I} \left[\exp\left(\tau Q(s, s')\right) \log \left(\sum_a \mathcal{T}(s' \mid s, a) \pi(a \mid s)\right)\right].$$

Given that the logarithm function is concave, we apply Jensen's inequality. Define:

$$\Delta(s, s') = \sum_a \mathcal{T}(s' \mid s, a),$$

Then we have:

$$\log\left(\sum_a \mathcal{T}(s' \mid s, a)\pi(a \mid s)\right) = \log\left(\sum_a \frac{\mathcal{T}(s' \mid s, a)}{\Delta(s, s')}\pi(a \mid s)\right) + \log \Delta(s, s') \tag{23}$$

$$\geq \sum_a \frac{\mathcal{T}(s' \mid s, a)}{\Delta(s, s')} \log \pi(a \mid s) + \log \Delta(s, s') \tag{24}$$

$$= \sum_a \mathcal{I}(a \mid s, s') \log \pi(a \mid s) + \log \Delta(s, s'). \tag{25}$$

Substituting this back into the original objective yields the lower bound:

$$F(\pi) \geq \mathbb{E}_{(s,s')\sim\mathcal{D}^I}\left[\exp\left(\tau Q(s, s')\right) \sum_a \mathcal{I}(a \mid s, s') \log \pi(a \mid s)\right] + \mathbb{E}_{(s,s')\sim\mathcal{D}^I}\left[\exp\left(\tau Q(s, s')\right) \log \Delta(s, s')\right].$$

The second term is independent of π and can be treated as a constant during training. Therefore, we can optimize the surrogate lower bound:

$$\widetilde{F}(\pi) = \mathbb{E}_{(s,s')\sim\mathcal{D}^I}\left[\exp\left(\tau Q(s, s')\right) \sum_a \mathcal{I}(a \mid s, s') \log \pi(a \mid s)\right].$$

□

B.8 Complete Derivation of $L(Q, V)$ using Pearson χ^2 divergence

We recall that the objective function $L(Q, V)$ has the following form:

$$\min_Q L(Q, V) = \alpha(1-\gamma)\,\mathbb{E}_{(s,s')\sim d_0}\left[Q(s, s')\right] + \alpha\mathbb{E}_{(s,s')\sim d^E}\left[f^*\left(\gamma V(s') - Q(s, s')\right)\right] + (1-\alpha)\mathbb{E}_{(s,s')\sim d^I}\left[\widetilde{f^*}\left(\gamma V(s') - Q(s, s')\right)\right] \tag{26}$$

where f^* is the convex conjugate of divergence function f and $\widetilde{f^*}(x) = f^*(x) - x$. Under Pearson χ^2 divergence, its convex conjugate $f^*(x) = \frac{x^2}{4} + x$ and the associated $\widetilde{f^*}(x) = \frac{x^2}{4}$. The objective (26) with Pearson χ^2 divergence becomes:

$$\min_Q L(Q, V) = \alpha(1-\gamma)\,\mathbb{E}_{(s,s')\sim d_0}\left[Q(s, s')\right] + \alpha\mathbb{E}_{(s,s')\sim d^E}\left[\gamma V(s') - Q(s, s')\right] + \frac{\alpha}{4}\mathbb{E}_{(s,s')\sim d^E}\left[\left(\gamma V(s') - Q(s, s')\right)^2\right] + \frac{1-\alpha}{4}\mathbb{E}_{(s,s')\sim d^I}\left[\left(\gamma V(s') - Q(s, s')\right)^2\right] \tag{27}$$

$$\Leftrightarrow \min_Q L(Q, V) = (1-\gamma)\,\mathbb{E}_{(s,s')\sim d_0}\left[Q(s, s')\right] + \mathbb{E}_{(s,s')\sim d^E}\left[\gamma V(s') - Q(s, s')\right] + \frac{1}{4\alpha}\alpha\mathbb{E}_{(s,s')\sim d^E}\left[\left(\gamma V(s') - Q(s, s')\right)^2\right] + \frac{1}{4\alpha}(1-\alpha)\mathbb{E}_{(s,s')\sim d^I}\left[\left(\gamma V(s') - Q(s, s')\right)^2\right] \tag{28}$$

$$\Leftrightarrow \min_Q L(Q, V) = (1-\gamma)\,\mathbb{E}_{(s,s')\sim d_0}\left[Q(s, s')\right] + \mathbb{E}_{s\sim d^E}\left[\gamma V(s)\right] - \mathbb{E}_{(s,s')\sim d^E}\left[Q(s, s')\right] + \frac{1}{4\alpha}\mathbb{E}_{s,s'\sim d^{E,I}_{mix}}\left[\left(\gamma V(s') - Q(s, s')\right)^2\right] \tag{29}$$

B.9 Complete Derivation of $\widetilde{L}(Q, V)$ for bounded Q-learning

The operator $\min_Q -\alpha\,\mathbb{E}_{(s,s')\sim\mathcal{D}^E}\left[Q(s, s')\right]$ in (29) which effectively encourages maximizing the Q-values of expert transitions can lead to *unbounded* growth in Q, potentially resulting in learning instability. To address this issue, we adapt a technique from [2] that bounds the expert Q-values. First,

looking at the objective 28, Let's define $r_Q^E(s,s') = Q(s,s') - \gamma V(s')\ \forall s, s' \sim d^E$. The training objective becomes:

$$\min_Q L(Q,V) = (1-\gamma)\,\mathbb{E}_{(s,s')\sim d_0}\left[Q(s,s')\right] + \mathbb{E}_{(s,s')\sim d^E}\left[-r_Q^E(s,s')\right]$$
$$+ \frac{1}{4\alpha}\alpha\mathbb{E}_{(s,s')\sim d^E}\left[r_Q^E(s,s')^2\right] + \frac{1}{4\alpha}(1-\alpha)\mathbb{E}_{(s,s')\sim d^I}\left[\left(\gamma V(s') - Q(s,s')\right)^2\right]$$
$$\Leftrightarrow \min_Q L(Q,V) = (1-\gamma)\mathbb{E}_{d_0(s,s')}\left[Q\big(s,s'\big)\right] + \frac{1-\alpha}{4\alpha}\mathbb{E}_{s,s'\sim d^I}[(\gamma V(s') - Q(s,s'))^2]$$
$$+ \left[\mathbb{E}_{s,s'\sim d^E}[-r_Q^E(s,s')] + \frac{1}{4}\mathbb{E}_{s,s'\sim d^E}[r_Q^E(s,s')^2]\right]$$
$$\Leftrightarrow \min_Q L(Q,V) = (1-\gamma)\mathbb{E}_{d_0(s,s')}\left[Q\big(s,s'\big)\right] + \frac{1-\alpha}{4\alpha}\mathbb{E}_{s,s'\sim d^I}[(\gamma V(s') - Q(s,s'))^2]$$
$$+ \frac{1}{4}\left[\mathbb{E}_{s,s'\sim d^E}[-4r_Q^E(s,s')] + \mathbb{E}_{s,s'\sim d^E}[r_Q^E(s,s')^2] + 4\right] - 1$$
$$\Leftrightarrow \min_Q L(Q,V) = \min_{Q(s,s')}(1-\gamma)\mathbb{E}_{d_0(s,s')}\left[Q\big(s,s'\big)\right] + \frac{1-\alpha}{4\alpha}\mathbb{E}_{s,s'\sim d^I}[(\gamma V(s') - Q(s,s'))^2]$$
$$+ \frac{1}{4}\mathbb{E}_{s,s'\sim d^E}[(r_Q^E(s,s') - 2)^2] \tag{30}$$
$$\Leftrightarrow \min_Q L(Q,V) = (1-\gamma)\mathbb{E}_{d_0(s,s')}\left[Q\big(s,s'\big)\right] + \frac{1-\alpha}{4\alpha}\mathbb{E}_{s,s'\sim d^I}[(\gamma Q(s',g) - Q(s,s'))^2]$$
$$+ \frac{1}{4}\mathbb{E}_{s,s'\sim d^E}[(Q(s,s') - (\gamma V(s') + 2))^2] \tag{31}$$

Following [2], the minimum of the third term in (30) is reached when $r_Q^E(s,s') = 2$. This will lead to $Q(s,s') = \sum_{t=0}^{\infty}\gamma^t 2 = \frac{2}{1-\gamma}\ \forall s, s' \sim d^E$. Therefore, we can replace the target $\gamma V(s') + 2$ in (31) with fixed target $\frac{2}{1-\gamma}$ to have the following modified objective with bounded Q.

$$\min_Q \widetilde{L}(Q,V) = (1-\gamma)\mathbb{E}_{d_0(s,s')}\left[Q\big(s,s'\big)\right] + \frac{1-\alpha}{4\alpha}\mathbb{E}_{s,s'\sim d^I}\left[\left(\gamma V(s') - Q(s,s')\right)^2\right]$$
$$+ \frac{1}{4}\mathbb{E}_{s,s'\sim d^E}\left[\left(Q(s,s') - \frac{2}{1-\gamma}\right)^2\right]$$

C Experimental and Implementation Details

Our method is implemented in JAX version 0.5.3 (with CUDA 12 capabilities). We conduct our experiments using a computing cluster with 8 NVIDIA RTX 3090 GPUs. For each IOSTOM run, five distinct training seeds are processed simultaneously on a shared hardware set comprising a single GPU, 32 CPU cores, and 128 GB of RAM. This parallel execution on shared resources enables the completion of 1 million training steps for all five seeds in about 60-90 minutes.

C.1 Mujoco tasks

We use the same offline LfO benchmark from DILO [38], which utilizes datasets derived from the D4RL [7] framework, and tests on Mujoco environments. The state-only expert dataset in all tasks includes only one expert trajectory. In terms of locomotion tasks, suboptimal datasets, labeled 'random+expert', 'random+few-expert', 'medium+expert', and 'medium+few-expert', are generated by mixing expert trajectories with lower-quality trajectories from D4RL's 'random-v2' and 'medium-v2' datasets, respectively. The 'random+expert' and 'medium+expert' datasets combine 200 expert trajectories with roughly 1 million transitions from the corresponding 'random-v2' or 'medium-v2' dataset. The 'x+few-expert' variants are similar but incorporate only 30 expert trajectories. In manipulation environments, all suboptimal 'x+expert' datasets are formed using 30 expert trajectories mixed with the complete 'x' D4RL dataset. We also use '-v0' variant of D4RL datasets for all manipulation tasks. Table 4 gives an overview about our LfO Mujoco tasks.

Task	State Dim	Action Dim	Horizion	Suboptimal Dataset	Data Points
Hopper	11	3	1000	random+expert	1e6 random transitions + 200 expert trajectories
				medium+expert	1e6 medium transitions + 200 expert trajectories
				random+few-expert	1e6 random transitions + 30 expert trajectories
				medium+few-expert	1e6 medium transitions + 30 expert trajectories
Walker2d	17	6	1000	random+expert	1e6 random transitions + 200 expert trajectories
				medium+expert	1e6 medium transitions + 200 expert trajectories
				random+few-expert	1e6 random transitions + 30 expert trajectories
				medium+few-expert	1e6 medium transitions + 30 expert trajectories
Halfcheetah	17	6	1000	random+expert	1e6 random transitions + 200 expert trajectories
				medium+expert	1e6 medium transitions + 200 expert trajectories
				random+few-expert	1e6 random transitions + 30 expert trajectories
				medium+few-expert	1e6 medium transitions + 30 expert trajectories
Ant	27	8	1000	random+expert	1e6 random transitions + 200 expert trajectories
				medium+expert	1e6 medium transitions + 200 expert trajectories
				random+few-expert	1e6 random transitions + 30 expert trajectories
				medium+few-expert	1e6 medium transitions + 30 expert trajectories
Pen	45	24	100	cloned+expert	5e6 cloned transitions + 30 expert trajectories
				human+expert	5000 human transitions + 30 expert trajectories
Door	39	28	200	cloned+expert	1e6 cloned transitions + 30 expert trajectories
				human+expert	6729 human transitions + 30 expert trajectories
Hammer	46	26	200	cloned+expert	1e6 cloned transitions + 30 expert trajectories
				human+expert	11310 human transitions + 30 expert trajectories
Kitchen	59	9	280	partial+expert	136950 partial transitions + 1 expert trajectories
				mixed+expert	136950 mixed transitions + 1 expert trajectories

Table 4: Overview of D4RL tasks and their repsective suboptimal dataset we use in LfO setting

C.2 Maritime Navigation task

The Maritime Navigation task was created using historical data from a hotspot in the Singapore Strait, following the AIS-driven simulation paradigm adopted in recent maritime traffic simulator **ShipNaviSim** [32]. We selected the area with the highest traffic density and collision risk—where numerous ships cross paths, as shown in Figure 3—as our planning region. We collect large amount of historical navigation data ($\sim$ 2 years) of vessels operating in this hotspot region recorded in the Automatic Identification System (AIS) from MarineTraffic[1]. The AIS data of each vessel contains two types of information: static and dynamic. The static data contains some information like width, length, type, and ID of vessel. Other vessel movement information like latitude, longitude, speed, heading and course-over-ground are included in dynamic data. To generate trajectory data, we selected tankers and cargo vessels as they represent the riskiest class due to their larger size (200-300 meters) and lower navigational agility. All trajectories were then interpolated at 10-second intervals. The final dataset comprises approximately 125,000 trajectories, totaling around 14 million environment transitions. The average trajectory length in dataset is around 100-150. We used 80% of the data for training and reserved the remaining 20% for evaluation.

The **observation space** is defined from the perspective of the ego agent (the vessel being controlled). At any given time, the agent observes a historical sequence of its own trajectory and those of the 10 closest nearby ships (each over a configurable number of past steps). For the ego agent and nearby ships, and for each historical point, the available features include the x and y coordinates, speed v, and heading angle h. Additionally, the agent observes its goal location. Observing past states and nearby ship information helps capture multi-ship interactions and provides context for decision-making. For simplicity, all algorithms used the same neural network architecture to process the observation space. We did not use any complex structures; instead, we flattened the observation space and provided it as input to the neural network.

The **action space** is modeled as a straightforward, 3-dimensional continuous space. An action is defined as $\langle d_x, d_y, d_h \rangle$, representing the changes in the x and y coordinates and the change in heading h, respectively. The vessel's speed at the next time step is derived from the distance traveled (calculated from $v_{t+1} = \sqrt{d_x^2 + d_y^2}/\delta_T$) divided by the time interval δ_T, which is set to 10 seconds.

[1] `https://www.marinetraffic.com/`

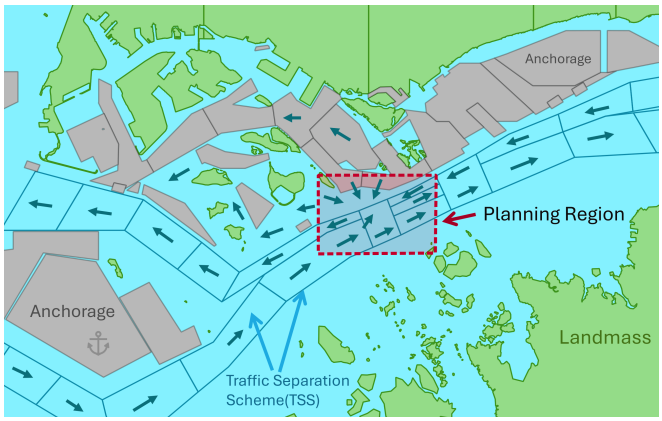

Figure 3: The red region is used as the environment area. The gray areas indicate anchorage zones, the green areas represent landmasses, and the arrows and regions with dark blue borders represent the Maritime Traffic Separation Scheme (TSS). The high density of crossing points in the red area makes it a more challenging region for navigation, providing a suitable setting for testing advanced planning techniques.

This is also known as a delta action space [12] and can be used for any moving object. Because action in our environment represents the difference between some state features of current and next timestep, we can have a simple Inverse Kinematics Model computing this difference to infer action between two consecutive states of state-only trajectories in datasets.

Following **vessel-specific metrics** introduced in **ShipNaviSim** [32] are used to evaluate navigation policies, comparing learned agent behavior to human expert data.

Goal-Conditioned ADE (GC-ADE) measures the average displacement between the learned policy's trajectory and the original historical trajectory in the 2D plane. Given τ_m of length T_m and τ_p of length T_p, GC-ADE computes the error over the minimum of the two lengths.

$$\text{GC-ADE} = \frac{1}{\min(T_m, T_p)} \sqrt{\sum_{t=1}^{\min(T_m,T_p)} (x_t^m - x_t^p)^2 + (y_t^m - y_t^p)^2}$$

Goal Rate is the percentage of times the ego agent successfully reaches its designated goal location. Success is defined as coming within a radius of 200 meters of the goal.

Near Miss Count represents the average number of time-steps per episode during which the ego agent approaches another vessel within a distance of 3 cable lengths (555 meters), which is considered a near-miss by domain experts. The 'near-miss' metric is interpreted broadly as a proxy for high traffic density and increased potential for navigation risk; it does not always imply that vessels in 'near-miss' situation were about to collide.

C.3 Architecture and Hyperparameters

Our implementation builds upon the official implementations of ReCOIL [39] and XQL [9]. We keep most of their parameters and network settings as shown in Table 5. We also add Layer Normalization [3] in V-function network to improve training stability as suggested in XQL. The regularization β was tuned by searching over [3, 5, 7, 10, 15, 20]. For locomotion tasks, we set $\beta = 20$ for standard LfO setting, and $\beta = 15$ for subsampled setting. In terms of manipulation tasks, $\beta = 10$ works best in most cases except 'pen-cloned' setting where β is set to 3. The policy temperature τ is often set to 3 in previous works [22, 39]. However, we find that this value results in very bad performance for IOSTOM because we do not use advantage for updating policy. We tune τ via via hyper-parameter sweeps over [0.01, 0.04, 0.08, 0.1, 0.2]. $\tau = 0.04$ is the best-performing hyperparameter in most tasks except for the 'human' Adroit and 'mixed' Franka Kitchen manipulation tasks, where $\tau = 0.01$ was used. For maritime navigation task, we set $\beta = 20$ and $\tau = 0.04$ which is similar to LfO setting of locomotion tasks.

Type	Hyperparameter	Value
Actor	Network Size	[256, 256]
	Activation Function	ReLU
	Learning Rate	3e-4
	Weight Decay	1e-3
	Training Length	1M steps
	Batch Size	512
	Optimizer	Adam
	Dropout Rate	0.1
	LR decay schedule	cosine
Critic	Network Size	[256, 256]
	Activation Function	ReLU
	Learning Rate	3e-4
	Training Length	1M steps
	Batch Size	512
	Optimizer	Adam
	Mixture Ratio α	0.5
	Polyak Update Rate λ	0.005
	Discount Factor γ	0.99

Table 5: Hyperparameters of IOSTOM

C.4 Baseslines

To evaluate the performance of our approach, we conduct comparative evaluations against three established state-of-the-art (SOTA) techniques: SMODICE [27], PW-DICE [48], and DILO [38]. The SMODICE and PW-DICE algorithms both operate by training a discriminator to guide the learned policy. Their fundamental difference lies in the divergence measure employed: SMODICE seeks to minimize the KL-divergence between the state occupancies of the learner and the expert, while PW-DICE alternatively uses the Wasserstein distance for this alignment. DILO offers a distinct, more recent SOTA paradigm for LfO, notable for its discriminator-free learning process. For all comparative methods, we utilize the publicly accessible codebases provided by their authors. To ensure fair comparisons, we use the hyperparameter settings recommended in their original publications or the default configurations within their code. The only exception is DILO where we can not reproduce consistent results as reported in the paper using their default parameters. After some tuning effort, we find that using Layer Normalization [3] can help to improve DILO performance. However, the training still diverges in some tasks as shown in Figures 4 and 5

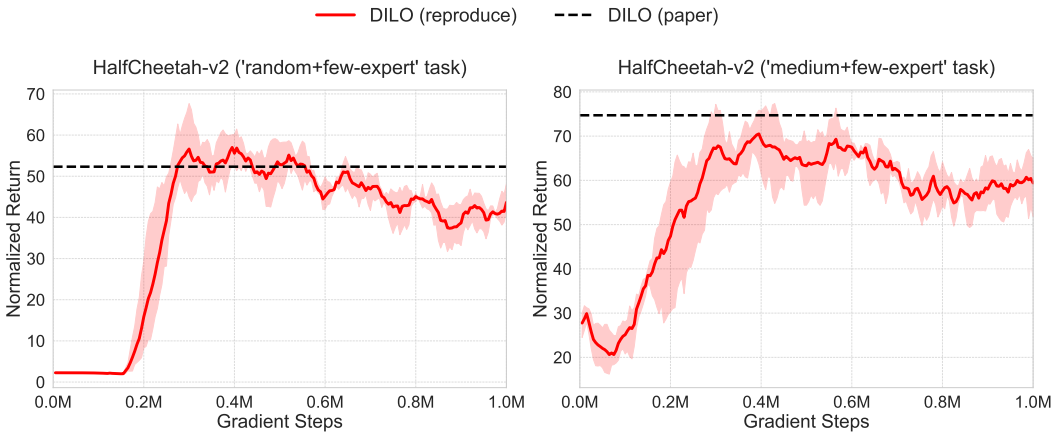

Figure 4: Training divergence of DILO on locomotion tasks

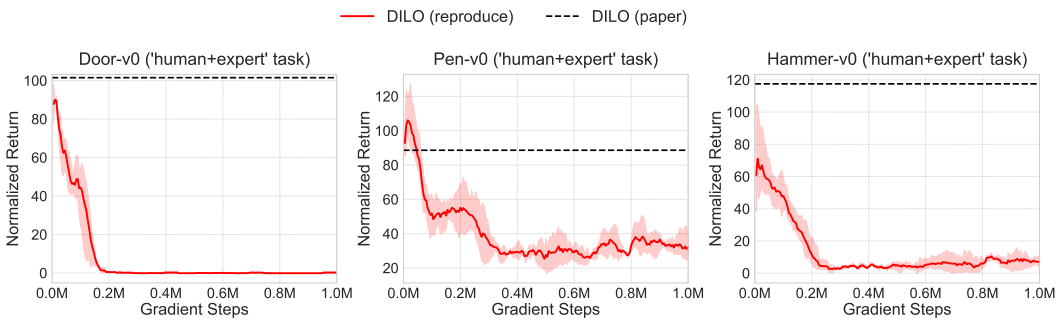

Figure 5: Training divergence of DILO on manipulation tasks

D Additional Experiments

D.1 Comparison with LobsDICE

Suboptimal Dataset	Env	PW-DICE	LobsDICE	IOSTOM
random+expert	hopper	108.09 $\pm$2.39	99.64 $\pm$5.01	**109.32** $\pm$1.08
	halfcheetah	86.11 $\pm$4.39	80.76 $\pm$3.17	**93.02** $\pm$0.40
	walker2d	107.48 $\pm$0.53	107.54 $\pm$0.13	**107.98** $\pm$0.20
	ant	126.89 $\pm$1.17	122.65 $\pm$0.72	**128.19** $\pm$1.52
random+few-expert	hopper	75.04 $\pm$14.21	74.25 $\pm$11.23	**107.28** $\pm$3.92
	halfcheetah	4.02 $\pm$1.74	11.61 $\pm$5.66	**88.77** $\pm$1.26
	walker2d	36.11 $\pm$9.19	101.29 $\pm$5.53	**108.40** $\pm$0.21
	ant	99.90 $\pm$2.59	93.84 $\pm$7.51	**120.09** $\pm$5.17
medium+expert	hopper	65.99 $\pm$8.05	74.43 $\pm$5.28	**110.20** $\pm$0.51
	halfcheetah	58.74 $\pm$1.84	71.09 $\pm$3.76	**93.12** $\pm$0.32
	walker2d	105.41 $\pm$0.33	103.34 $\pm$2.11	**108.12** $\pm$0.13
	ant	108.14 $\pm$1.90	114.96 $\pm$4.05	**124.72** $\pm$3.49
medium+few-expert	hopper	57.24 $\pm$3.03	67.00 $\pm$6.43	**108.96** $\pm$1.33
	halfcheetah	27.85 $\pm$6.03	44.76 $\pm$3.94	**89.47** $\pm$0.82
	walker2d	75.22 $\pm$7.05	95.39 $\pm$5.22	**108.15** $\pm$0.43
	ant	90.34 $\pm$2.56	95.33 $\pm$1.08	**120.36** $\pm$1.25
cloned+expert	pen	23.39 $\pm$4.56	29.83 $\pm$4.59	**82.77** $\pm$4.84
	door	0.07 $\pm$0.14	0.02 $\pm$0.00	**102.77** $\pm$0.96
	hammer	1.29 $\pm$0.12	0.55 $\pm$0.19	**94.59** $\pm$9.39
human+expert	pen	-2.56 $\pm$1.30	42.09 $\pm$5.08	**95.77** $\pm$8.91
	door	0.15 $\pm$0.02	10.98 $\pm$9.71	**100.77** $\pm$1.68
	hammer	2.02 $\pm$0.77	17.06 $\pm$13.44	**93.34** $\pm$7.41
partial+expert	kitchen	12.33 $\pm$5.38	40.33 $\pm$2.24	**58.95** $\pm$2.27
mixed+expert	kitchen	7.50 $\pm$4.16	45.67 $\pm$2.75	**46.45** $\pm$0.84

Table 6: Average normalized return over last 10 evaluations of IOSTOM against LobsDICE and PW-DICE on the D4RL suboptimal datasets with 1 expert trajectory. The mean and std are obtained over 5 random seeds. LfO methods with avg. perf within the std-dev of the top performing LfO approach is in **bold**.

To further strengthen our empirical study, we additionally include a comparison against LobsDICE [18], alongside PW-DICE and our method (IOSTOM), under the same experimental settings. Although previous work [48] has suggested that PW-DICE generally outperforms LobsDICE on the

D4RL "random+expert" benchmark, we perform a direct comparison here for completeness and to ensure a fair evaluation across representative DICE-based baselines.

The corresponding results are reported in Table 6. We observe that IOSTOM consistently and substantially outperforms LobsDICE across all tasks and dataset regimes. As is common among DICE-based methods, LobsDICE exhibits degraded performance in the "few-expert" and manipulation tasks, whereas IOSTOM remains robust. Our findings also confirm that PW-DICE surpasses LobsDICE in the "random+expert" setting, but PW-DICE becomes unstable and underperforms in several other configurations, illustrating the general sensitivity of discriminator-based approaches. In contrast, IOSTOM achieves both stronger performance and greater robustness, reinforcing the benefits of its stable, discriminator-free formulation.

D.2 Sensitivity to Subsampling of Low-Quality Data

Suboptimal Dataset	Env	IOSTOM(full)	IOSTOM(sub5)	IOSTOM(sub20)
random+expert	hopper	109.32 ± 1.08	110.24 ± 0.63	110.66 ± 0.16
	halfcheetah	93.02 ± 0.40	90.81 ± 2.10	**85.59** ± 5.09
	walker2d	107.98 ± 0.20	107.91 ± 0.16	107.53 ± 0.11
	ant	128.19 ± 1.52	127.26 ± 2.48	127.30 ± 1.37
random+few-expert	hopper	107.28 ± 3.92	108.99 ± 0.76	109.34 ± 1.62
	halfcheetah	88.77 ± 1.26	86.62 ± 1.98	**80.49** ± 6.41
	walker2d	108.40 ± 0.21	108.13 ± 0.29	107.78 ± 0.22
	ant	120.09 ± 5.17	119.63 ± 2.40	**105.98** ± 7.03
medium+expert	hopper	110.20 ± 0.51	109.89 ± 0.22	109.58 ± 1.18
	halfcheetah	93.12 ± 0.32	91.07 ± 1.72	**76.30** ± 13.56
	walker2d	108.12 ± 0.13	108.31 ± 0.21	108.03 ± 0.15
	ant	124.72 ± 3.49	127.71 ± 2.72	128.61 ± 0.71
medium+few-expert	hopper	108.96 ± 1.33	109.68 ± 0.56	108.20 ± 0.83
	halfcheetah	89.47 ± 0.82	90.20 ± 1.94	**77.65** ± 11.48
	walker2d	108.15 ± 0.43	107.88 ± 0.83	107.39 ± 0.92
	ant	120.36 ± 1.25	118.23 ± 5.04	**102.14** ± 7.26

Table 7: Average normalized return over the last 10 evaluations of IOSTOM under different subsampling rates of low-quality data (sub5 and sub20) on the D4RL suboptimal datasets with 1 expert trajectory. Performance drops exceeding 5% relative to IOSTOM(full) are shown in **bold**.

To assess the sensitivity of our method to the amount of available low-quality data, we conducted additional experiments where the random (or medium-quality) portion of the dataset was subsampled to only 20% and 5% of its original size. These two variants are denoted as *IOSTOM(sub5)* and *IOSTOM(sub20)*, while the original version is referred to as *IOSTOM(full)*. The corresponding results are summarized in Table 7. Performance drops exceeding 5% relative to IOSTOM(full) are highlighted in **bold**.

When reducing the low-quality data to 20% (IOSTOM(sub5)), our method exhibits strong robustness: across all tasks and datasets, performance remains very close to IOSTOM(full), with no significant degradation observed. However, when the low-quality portion is aggressively reduced to just 5% (IOSTOM(sub20)), we observe a more noticeable performance decline — up to 18% on some tasks, particularly in halfcheetah and ant. Nonetheless, IOSTOM still achieves robust performance on 10 out of 16 tasks, even under this extremely limited data regime.

D.3 Comparison with other variants of IOSTOM

To validate our algorithmic designs, we compare IOSTOM with other variants: IOSTOM-IDM (Using Inverse Dynamics Model), IOSTOM-Adv (Using advantage instead of Q to update policy), and IOSTOM-IQL (Using Implicit Q Learning [22] objective to train V-function network). Table 8 shows these comparison results. Overall, IOSTOM has the best performance on 17/24 tasks and consistently produces high-quality results compared to other variants. IOSTOM-IQL is the second

Suboptimal Dataset	Env	IOSTOM-ADV	IOSTOM-IDM	IOSTOM-IQL	IOSTOM	Expert
random+ expert	hopper	55.93 ±6.67	97.06 ±4.57	**109.74** ±0.26	**109.32** ±1.08	111.33
	halfcheetah	6.43 ±2.51	80.53 ±3.40	**92.82** ±0.71	**93.02** ±0.40	88.83
	walker2d	104.20 ±4.07	79.70 ±29.28	**108.43** ±0.14	107.98 ±0.20	106.92
	ant	120.13 ±3.65	**128.99** ±3.11	**128.71** ±3.73	**128.19** ±1.52	130.75
random+ few-expert	hopper	20.47 ±10.26	50.63 ±32.03	**106.59** ±3.05	**107.28** ±3.92	111.33
	halfcheetah	2.13 ±0.11	68.79 ±4.26	86.44 ±1.55	**88.77** ±1.26	88.83
	walker2d	8.38 ±2.84	69.11 ±23.28	**108.37** ±0.05	**108.40** ±0.21	106.92
	ant	41.18 ±12.90	**123.86** ±1.97	**125.15** ±4.50	120.09 ±5.17	130.75
medium+ expert	hopper	53.46 ±13.16	97.62 ±7.42	**110.71** ±0.35	110.20 ±0.51	111.33
	halfcheetah	49.00 ±3.44	85.21 ±1.86	91.45 ±1.49	**93.12** ±0.32	88.83
	walker2d	85.96 ±21.33	94.90 ±29.34	**108.45** ±0.17	108.12 ±0.13	106.92
	ant	118.74 ±4.55	**128.32** ±0.64	124.66 ±4.62	124.72 ±3.49	130.75
medium few-expert	hopper	42.79 ±2.36	68.32 ±17.89	106.02 ±3.31	**108.96** ±1.33	111.33
	halfcheetah	42.31 ±0.60	76.48 ±5.57	78.18 ±2.24	**89.47** ±0.82	88.83
	walker2d	74.16 ±2.14	107.89 ±0.28	**108.38** ±0.16	108.15 ±0.43	106.92
	ant	99.89 ±1.92	121.80 ±1.78	**121.64** ±2.35	**120.36** ±1.25	130.75
cloned+expert	pen	41.65 ±5.42	63.39 ±11.46	10.54 ±1.51	**82.77** ±4.84	106.42
	door	13.87 ±8.26	18.68 ±12.44	32.25 ±13.03	**102.77** ±0.96	103.94
	hammer	11.77 ±16.66	47.83 ±8.43	57.04 ±6.34	**94.59** ±9.39	125.71
human+expert	pen	**92.73** ±3.73	81.72 ±5.13	**95.26** ±10.16	**95.77** ±8.91	106.42
	door	95.08 ±1.90	78.50 ±20.55	**99.47** ±3.67	**100.77** ±1.68	103.94
	hammer	**88.23** ±5.72	82.12 ±18.51	68.32 ±13.66	**93.34** ±7.41	125.71
partial+expert	kitchen	56.08 ±0.29	66.30 ±5.70	**57.75** ±2.00	**58.95** ±2.27	75.0
mixed+expert	kitchen	**49.42** ±0.58	28.95 ±10.62	47.92 ±1.23	46.45 ±0.84	75.0

Table 8: Average normalized return over last 10 evaluations of IOSTOM against other variants on the D4RL suboptimal datasets with 1 expert trajectory. The mean and std are obtained over 5 random seeds. LfO methods with avg. perf within the std-dev of the top performing LfO approach is in **bold**.

.

best method, but its performance is still significantly worse than IOSTOM on 'cloned' tasks. The results in 'few-expert' setting of IOSTOM-IDM is very bad compared to 'expert' setting which clearly shows the weakness of training Inverse Dynamics Model with low-quality data. IOSTOM-ADV has the worst performance in most tasks.

D.4 α Ablation

Suboptimal Dataset	Env	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$ (our)	$\alpha = 0.7$	$\alpha = 0.9$	$gap_{worst}(\%)$	$gap_{default}(\%)$
random+expert	hopper	109.72 ±0.84	110.21 ±0.68	109.32 ±1.08	109.19 ±0.21	**110.29** ±0.64	1.00	0.88
	halfcheetah	**93.09** ±0.22	92.90 ±0.31	93.02 ±0.40	92.99 ±0.08	92.89 ±0.18	0.21	0.08
	walker2d	107.82 ±0.23	107.97 ±0.08	107.98 ±0.20	108.16 ±0.17	**108.38** ±0.26	0.52	0.37
	ant	**128.37** ±1.89	126.67 ±2.89	128.19 ±1.52	125.78 ±3.00	127.30 ±1.85	2.02	0.14
random+few-expert	hopper	107.41 ±1.27	**108.59** ±2.04	107.28 ±3.92	104.75 ±3.36	107.25 ±4.11	3.54	1.21
	halfcheetah	87.29 ±0.73	87.52 ±2.11	**88.77** ±1.26	88.42 ±1.00	86.61 ±1.84	2.43	0.00
	walker2d	108.09 ±0.24	108.21 ±0.06	108.40 ±0.21	**108.45** ±0.12	108.24 ±0.14	0.33	0.05
	ant	**125.43** ±2.77	121.19 ±2.09	120.09 ±5.17	122.18 ±1.90	123.46 ±1.17	4.26	4.26
medium+expert	hopper	109.79 ±0.95	110.44 ±0.51	110.20 ±0.51	110.30 ±0.54	**110.61** ±0.07	0.74	0.37
	halfcheetah	**93.16** ±0.19	92.82 ±0.32	93.12 ±0.32	93.00 ±0.23	92.68 ±0.17	0.52	0.04
	walker2d	107.54 ±0.45	107.96 ±0.12	108.12 ±0.13	107.28 ±1.44	**108.22** ±0.27	0.87	0.09
	ant	**129.00** ±0.59	124.91 ±2.56	124.72 ±3.49	125.52 ±2.26	127.86 ±2.20	3.32	3.32
medium+few-expert	hopper	107.95 ±1.72	**110.08** ±1.12	108.96 ±1.33	106.99 ±4.23	104.34 ±5.88	5.21	1.02
	halfcheetah	87.98 ±1.17	87.79 ±1.77	**89.47** ±0.82	88.66 ±1.31	89.17 ±0.60	1.88	0.00
	walker2d	108.15 ±0.31	108.24 ±0.33	108.15 ±0.43	108.16 ±0.22	**108.47** ±0.22	0.30	0.30
	ant	120.44 ±1.57	119.36 ±1.84	120.36 ±1.25	119.12 ±0.58	**122.32** ±0.73	2.62	1.60
cloned+expert	pen	50.08 ±17.30	72.62 ±7.45	**82.77** ±4.84	73.96 ±5.58	76.37 ±6.52	39.49	0.00
	door	102.68 ±0.63	**103.79** ±0.63	102.77 ±0.96	102.65 ±1.37	103.64 ±0.92	1.10	0.98
	hammer	101.98 ±4.88	**110.07** ±7.29	94.59 ±9.39	105.58 ±12.35	106.81 ±2.94	14.06	14.06
human+expert	pen	95.75 ±7.18	96.68 ±4.83	95.77 ±8.91	**98.29** ±5.11	97.74 ±5.11	2.58	2.56
	door	100.41 ±3.85	100.94 ±3.65	100.77 ±1.68	99.99 ±2.31	**101.13** ±1.07	1.13	0.36
	hammer	95.24 ±5.79	101.71 ±4.97	93.34 ±7.41	101.04 ±6.36	**103.79** ±8.80	10.07	10.07
partial+expert	kitchen	**64.00** ±7.66	61.83 ±1.38	58.95 ±2.27	61.33 ±4.13	60.08 ±3.21	7.89	7.89
mixed+expert	kitchen	45.25 ±2.38	45.42 ±4.05	**46.45** ±0.84	45.08 ±1.13	45.33 ±2.01	2.95	0.00
Average							4.54	2.07

Table 9: Average normalized return over last 10 evaluations of IOSTOM with different α values on the D4RL suboptimal datasets with 1 expert trajectory. Method with the best avg. perf is in **bold**.

This section presents an ablation study to evaluate the impact of the hyperparameter α on IOSTOM's performance. Table D.4 reports performance for each setting, with the best-performing α for each task highlighted in bold. We also include two additional metrics: gap_{worst}, representing the percentage gap between the best and worst α, and $gap_{default}$, indicating the gap between the best-performing α and our default setting of $\alpha = 0.5$.

According to the table, the average gap_{worst} value across all tasks is just 4.54%, which is relatively small. This indicates that IOSTOM's performance is not highly sensitive to the choice of α. Furthermore, the average performance gap between the task-specific optimal α and our default choice of $\alpha = 0.5$ is even smaller at 2.07%. The $gap_{default}$ value is also under 5% in all but 3 of 24 tasks. This confirms that $\alpha = 0.5$ is a robust and effective hyperparameter choice, consistently providing near-optimal performance.

D.5 β Ablation

Suboptimal Dataset	Env	β=3	β=5	β=10	β=15	β=20	Expert
random+ expert	hopper	13.72 ±3.59	5.56 ±1.16	41.39 ±39.18	**110.36** ±0.46	109.32 ±1.08	111.33
	halfcheetah	52.40 ±11.29	92.64 ±0.90	93.10 ±0.25	**93.18** ±0.35	93.02 ±0.40	88.83
	walker2d	1.18 ±0.29	60.97 ±39.06	6.75 ±14.23	107.67 ±0.14	**107.98** ±0.20	106.92
	ant	118.94 ±7.69	126.17 ±2.26	128.02 ±3.01	**128.20** ±3.52	128.19 ±1.52	130.75
random+ few-expert	hopper	10.68 ±3.56	6.10 ±0.76	35.94 ±15.77	102.30 ±5.12	**107.28** ±3.92	111.33
	halfcheetah	2.24 ±0.01	2.20 ±0.05	83.37 ±2.08	85.20 ±1.61	**88.77** ±1.26	88.83
	walker2d	1.20 ±0.24	11.65 ±3.33	29.87 ±31.90	108.21 ±0.41	**108.40** ±0.21	106.92
	ant	45.72 ±15.95	97.42 ±14.16	**125.28** ±3.05	122.76 ±3.64	120.09 ±5.17	130.75
medium+ expert	hopper	31.99 ±24.17	61.48 ±19.37	109.97 ±0.42	110.02 ±1.00	**110.20** ±0.51	111.33
	halfcheetah	43.20 ±0.47	54.63 ±3.08	92.63 ±0.21	92.96 ±0.29	**93.12** ±0.32	88.83
	walker2d	71.70 ±4.56	107.83 ±0.75	**108.31** ±0.27	108.28 ±0.12	108.12 ±0.13	106.92
	ant	98.70 ±1.81	102.01 ±2.96	121.86 ±2.49	124.12 ±2.93	**124.72** ±3.49	130.75
medium few-expert	hopper	46.86 ±3.74	61.88 ±22.27	105.83 ±4.07	106.80 ±2.29	**108.96** ±1.33	111.33
	halfcheetah	42.85 ±0.27	43.17 ±0.12	49.01 ±1.15	83.52 ±1.72	**89.47** ±0.82	88.83
	walker2d	66.58 ±1.99	69.35 ±6.58	108.33 ±0.28	**108.46** ±0.13	108.15 ±0.43	106.92
	ant	92.15 ±1.80	94.62 ±3.44	98.59 ±1.67	111.45 ±3.09	**120.36** ±1.25	130.75
cloned+expert	pen	**82.77** ±4.84	56.05 ±7.20	11.30 ±2.64	10.33 ±2.27	10.13 ±2.90	106.42
	door	40.58 ±55.52	80.99 ±45.33	**102.77** ±0.96	100.08 ±2.59	86.06 ±6.14	103.94
	hammer	88.63 ±33.94	94.88 ±16.51	94.59 ±9.39	**100.59** ±10.60	90.27 ±18.70	125.71
human+expert	pen	99.27 ±5.22	**101.27** ±6.45	95.77 ±8.91	96.14 ±7.88	99.96 ±13.06	106.42
	door	99.90 ±1.90	**102.22** ±1.42	100.77 ±1.68	99.43 ±2.36	101.22 ±2.95	103.94
	hammer	87.11 ±16.28	**102.10** ±15.65	93.34 ±7.41	97.37 ±15.52	93.39 ±7.70	125.71
partial+expert	kitchen	49.80 ±14.81	**61.10** ±3.24	57.75 ±2.00	63.00 ±3.33	59.70 ±5.28	75.0
mixed+expert	kitchen	46.75 ±1.63	45.80 ±2.65	**47.92** ±1.23	46.85 ±1.80	45.40 ±3.84	75.0

Table 10: Average normalized return over last 10 evaluations of IOSTOM with different β values on the D4RL suboptimal datasets with 1 expert trajectory. Method with the best avg. perf is in **bold**.

This section presents an ablation study to evaluate the impact of the hyperparameter β on IOSTOM's performance. Table 10 summarizes these results. For locomotion tasks (e.g., Hopper, HalfCheetah, Walker2d, Ant), higher β values, typically 15 or 20, generally yield superior scores compared to lower values such as 3 or 5. Conversely, for manipulation tasks (e.g., Pen, Door, Hammer, Kitchen), optimal performance is often achieved with β values of 5 or 10. However, the performance differences across various β settings for these tasks are less pronounced. The only exception is the 'pen' environment within the 'cloned+expert' dataset, where decreasing β leads to improved results, with $\beta = 3$ achieving the highest score.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:[Yes]

   Justification: *Our main claims made in the abstract and introduction precisely reflect our paper's contributions and scope.*

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: *The limitations of our work are included in the Appendix.*

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: *The proofs for all propositions are included in the Appendix. All assumptions are clearly stated in both the main text and the proofs.*

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: *The source-code of our algorithm, along with environment details and exact commands to run are included in the supplementary material.*

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: *The D4RL datasets is publicly available. We have submitted source code with detailed instructions. We can not publish the AIS data we used in Section 6.4 due to a confidential agreement with a third-party data provider.*

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: *All experimental details are described in the main paper and Appendix.*

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: *We have reported the mean and std of results obtained by running with 5 different random sets in the all experimental sections of main paper. In the code, we provide scripts for generating all training curves constructed from mean scores and shaded by standard error.*

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: *We have shown all these information in our Appendix.*

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: *I confirm that my paper conforms to the Code of Ethics.*

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: *The paper provides a general offline imitation learning algorithm that only tests on the simulated environments. As such, we do not foresee any direct societal impact.*

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: *Our work has no risk for misuse because we do not release any dataset or pretrained model.*

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: *Our source code is submitted alongside the paper, accompanied by sufficient instructions.*

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: *We do not conduct any crowdsourcing experiment and research with human subjects.*

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: *We do not conduct any experiment related to human subjects.*

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: *The core method development in this research does not involve LLMs as any important, original, or non-standard components.*

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.