Imitation Learning from Observation through Optimal Transport

Wei-Di Chang
wchang@cim.mcgill.ca
McGill University

Scott Fujimoto
McGill University

David Meger
McGill University

Gregory Dudek
McGill University

Abstract

Imitation Learning from Observation (ILfO) is a setting in which a learner tries to imitate the behavior of an expert, using only observational data and without the direct guidance of demonstrated actions. In this paper, we re-examine optimal transport for IL, in which a reward is generated based on the Wasserstein distance between the state trajectories of the learner and expert. We show that existing methods can be simplified to generate a reward function without requiring learned models or adversarial learning. Unlike many other state-of-the-art methods, our approach can be integrated with any RL algorithm and is amenable to ILfO. We demonstrate the effectiveness of this simple approach on a variety of continuous control tasks and find that it surpasses the state of the art in the IlfO setting, achieving expert-level performance across a range of evaluation domains even when observing only a single expert trajectory *without* actions.

1 Introduction

Imitation Learning (IL) is a widely used and effective tool for teaching robots complex behaviors. Although Reinforcement Learning (RL) has demonstrated success in learning motor skills from scratch in real-world systems (Haarnoja et al., 2018b; Kalashnikov et al., 2018), Imitation Learning (IL) remains a proven and practical way to learn behaviors from demonstrations, without the need for a hand-tuned and engineered reward signal required for RL. However, acquiring access to expert actions can be highly impractical. For example, robotic systems that are too challenging to teleoperate smoothly or in applications where the action spaces of the demonstrator and the imitator do not match, such as in Sim-to-Real problems (Desai et al., 2020).

Imitation Learning from Observation (ILfO) eliminates the need for demonstrated actions by learning behaviors from sequences of expert states instead of requiring both expert states and actions. Similar to how humans learn new skills from watching others, ILfO algorithms learn from observational data alone. Consequently, this reduces the cost of data collection, making ILfO algorithms instrumental for deploying IL in complex real-world systems.

Moving to the observation-only space, however, introduces new challenges. While IL algorithms can learn by copying actions, ILfO algorithms require more exploration to succeed (Kidambi et al., 2021), as they can only indirectly imitate the expert through observed outcomes. This emphasis on exploration creates a further challenge in that the states visited by the learner are more likely to be distant or non-overlapping with those of the expert. Distant states are problematic for imitation via distribution matching (Ho & Ermon, 2016; Ghasemipour et al., 2020; Ni et al., 2020), as the widely used KL divergence is ill-defined for non-overlapping distributions. While IL methods can circumvent this problem by accelerating early learning with behavior cloning, ILfO methods must deal with randomly initialized policies, which are unlikely to behave similarly to an expert demonstrator.

The field of optimal transport has garnered much attention in recent years, with theoretical and computational developments allowing it to evaluate distances between distributions defined on high-

dimensional metric spaces (Cuturi, 2013; Bonneel et al., 2015). The Wasserstein distance, in particular, can compare non-overlapping distributions and quantify the spatial shift between the supports of the distributions. These properties make it a natural alternative to KL divergence-based objectives used by existing methods. Moreover, the Wasserstein distance can be computed without requiring separate models or learned components. This makes the Wasserstein distance more computationally efficient and conceptually simpler than other methods that rely on incremental adversarial signals learned via online interaction (Ho & Ermon, 2016; Kostrikov et al., 2019; Papagiannis & Li, 2020).

Prior work (Papagiannis & Li, 2020; Dadashi et al., 2020; Durugkar et al., 2021) based on the Wasserstein distance for IL or ILfO relies on numerous techniques, such as adversarial or learned components, or designed for sample-inefficient on-policy RL algorithms. Building on prior work (Papagiannis & Li, 2020), we introduce a simpler approach that does not require adversarial components or on-policy learning. Our resulting approach, Observational Off-Policy Sinkhorn (OOPS), generates a reward function for *any* RL algorithm, which minimizes the Wasserstein distance between expert and learner state trajectories. We benchmark OOPS against existing methods proposed to optimize the Wasserstein distance (Papagiannis & Li, 2020; Dadashi et al., 2020), as well as current state-of-the-art ILfO algorithms (Ghasemipour et al., 2020; Zhu et al., 2020) on a variety of continuous control tasks. OOPS outperforms state-of-the-art methods for ILfO, achieving near-expert performance in every evaluated task with only a single trajectory without observing any actions. To facilitate reproducibility, all of our code is open-sourced[1].

2 Background

Setting. Our task is formulated by an episodic finite-horizon MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, p_0, T)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, initial state distribution $p_0 : \mathcal{S} \to [0, 1]$, and T the horizon. While the overarching objective is to maximize reward, in the Imitation Learning from Observation (ILfO) setting, the agent never observes the true reward. Instead, ILfO algorithms must use sequences of states (trajectories τ), generated by an unknown expert, to infer a reward signal or objective. We therefore only assume access to a dataset D_E of N state-only trajectories, $D_E = \{\tau_0, \tau_1, ..., \tau_{N-1}\}$.

Optimal Transport. Optimal Transport (OT) seeks to compute a matching between the source and target measures while minimizing the transport cost (Villani, 2009). In our work, we aim to minimize the distance between the distribution of trajectories defined by the learner and the expert.

Writing out trajectories in terms of their transitions $\tau = \{(s_0, s_1), (s_1, s_2), ..., (s_{T-1}, s_T)\}$, and viewing each transition as a datapoint, forms a discrete measure α over the state transition space $\mathcal{S} \times \mathcal{S}$, with weights $\mathbf{a}$ and locations $(s_i, s_{i+1})_E \in \mathcal{S} \times \mathcal{S}$ for the expert: $\alpha = \sum_{i=0}^{T} a_i \sigma_{(s_i, s_{i+1})_E}$ where $\sigma_{(s_i, s_{i+1})}$ is the Dirac delta function at position (s_i, s_{i+1}). Similarly for the learner, with weights $\mathbf{b}$ and locations $(s_i, s_{i+1})_\pi$ for the learner, the trajectory rollout forms the measure $\beta = \sum_{i=0}^{T} b_i \sigma_{(s_i, s_{i+1})_\pi}$ (Peyré et al., 2019). In each trajectory, we consider each timestep as being equally important, and as such restrict the weight vectors $\mathbf{a}$ and $\mathbf{b}$ to the uniform weight vectors: $\sum_{i=0}^{T} a_i = 1, a_i = \frac{1}{T} \, \forall \, 0 < i < T$, and $\sum_{i=0}^{T} b_i = 1, b_i = \frac{1}{T} \, \forall \, 0 < i < T$.

While the Monge formulation of OT enforces a one-to-one matching between measures, the Kantorovich formulation relaxes the OT problem by allowing each source point to split mass: the mass at any source point may be distributed across several locations (Villani, 2009; Peyré et al., 2019). This provides the Wasserstein distance (or Kantorovich metric) over a distance metric d:

$$W_p(\alpha, \beta) := \left(\min_P \left(\sum_i^T \sum_j^T d(\alpha_i, \beta_j)^p P_{i,j} \right) \right)^{\frac{1}{p}}, \tag{1}$$

[1]Link removed for anonymization. Code in supplementary material.

which uses a coupling matrix $P \in \mathbb{R}_+^{n \times m}$, where $P_{i,j}$ is the mass flowing from bin i to bin j:

$$P \in \mathbb{R}^{n \times m} \text{ such that } \sum_j P_{i,j} = \mathbf{a} \text{ and } \sum_i P_{i,j} = \mathbf{b}. \tag{2}$$

The optimal coupling P between α and β gives us the minimal cost transport plan between the measures defined by the trajectories τ_π and τ_E.

Sinkhorn distance. The Sinkhorn distance W_{Sk} is an entropy regularized version of the Wasserstein distance (Cuturi, 2013), for W_1, with $p = 1$ this equals:

$$W_{\text{Sk}}(\tau_\pi, \tau_E) := \min_{\tilde{P}} \sum_{i=0}^{T} \sum_{j=0}^{T} d(\alpha_i, \beta_j) \tilde{P}_{i,j} - \lambda \mathcal{H}(\tilde{P}), \tag{3}$$

where the entropy term $\mathcal{H}(\tilde{P}) := \sum_{i=0}^{T} \sum_{j=0}^{T} \tilde{P}_{ij} \log \tilde{P}_{ij}$. For any given value of $\lambda > 0$, the optimal coupling matrix $\tilde{P}$ for W_{Sk} can be computed efficiently using the iterative Sinkhorn algorithm (Sinkhorn, 1967). At the cost of convergence speed, as λ approaches 0, the Wasserstein distance is recovered, while increasing its value blurs out the transport matrix and spreads the mass between the two measures. This approximation is useful as it provides a computationally efficient method for estimating the optimal coupling matrix for the Wasserstein distance $\tilde{P} \approx P$ for small λ, where W_{Sk} upper bounds W_1.

3 Related Work

Imitation Learning. Learning from Demonstrations (LfD) approaches can be generally classified into two types of approaches: IL methods, which learn directly from expert data, and Inverse Reinforcement Learning (IRL) methods (Ziebart et al., 2008) which infer a reward function that is optimized by RL. GAIL (Ho & Ermon, 2016) and related methods (Kostrikov et al., 2019; Fu et al., 2017) leverage adversarial training. These methods optimize a distribution matching objective between the state-action distribution of the learner and the expert, in terms of various probability divergence metrics (Ho & Ermon, 2016; Ghasemipour et al., 2020; Kostrikov et al., 2018; Ni et al., 2020). Each divergence objective leads to distinct imitative behavior (zero-forcing or mean-seeking or both), which can be exploited in different scenarios (Ke et al., 2019). In contrast, our approach minimizes a Wasserstein distance-based objective, better suited for our ILfO context.

Imitation Learning from Observations. Due to the challenging nature of ILfO, many methods rely on learning a model, via an inverse dynamics model used to infer the missing actions of the expert (Torabi et al., 2018a), use objectives based on the transition dynamics of the expert (Jaegle et al., 2021; Chang et al., 2022), or simply model the entire MDP (Kidambi et al., 2021). Adversarial methods have also been adapted from the IL context (Sun et al., 2019; Torabi et al., 2018b). Another common theme is f-divergence minimization, (Ni et al., 2020) derive an approach based on the analytical gradients of f-divergences and show that different variants (FKL, RKL, JS) can be achieved through their framework. OPOLO (Zhu et al., 2020), leverages off-policy learning on top of an inverse dynamics model and adversarial training. As opposed to existing methods, our approach leverages the Wasserstein distance to compute a non-adversarial and model-free reward for ILfO.

Optimal Transport for Imitation Learning. Minimization of the Wasserstein distance for IL has been previously considered in (Xiao et al., 2019; Zhang et al., 2020) through Wasserstein Generative Adversarial Network (WGAN)-inspired approaches (Arjovsky et al., 2017). In an adversarial policy learning set up similarly to GAIL (Ho & Ermon, 2016) and by restricting the discriminator to be a 1-Lipschitz function, these approaches can minimize the W_1 distance between the policy and the reference trajectory data distribution. However these methods suffer from the drawbacks of adversarial frameworks, which are hard to optimize and tune (Arjovsky & Bottou, 2017), and have been shown to be poor estimators of W_1 (Stanczuk et al., 2021).

More recent works (Papagiannis & Li, 2020; Dadashi et al., 2020; Haldar et al., 2023) use Wasserstein distance solvers, or related approximations, for IL. Our approach is closely based on Sinkhorn

Imitation Learning (SIL) (Papagiannis & Li, 2020), which uses the Sinkhorn distance (Cuturi, 2013) to compute an entropy regularized Wasserstein distance between the state-action occupancy of the learner and expert. However, rather than use an upper bound defined by off-policy samples, they use on-policy RL (Schulman et al., 2015) to optimize the cosine distance over the representation space of an adversarial discriminator trained alongside the imitation agent. In our work, we found that we can vastly improve sample efficiency by using an off-policy agent instead and can consider a more straightforward objective without adversarial or learned representations, an aspect previously thought required for good performance. Another related approach, PWIL (Dadashi et al., 2020), uses a greedy formulation of the Wasserstein distance and matches the current state-action pair (s, a) to its closest counterpart in the expert demonstration dataset at every rollout step. In our experimental analysis (Figure 3), we show that our approximation via the Sinkhorn distance creates a tighter upper bound of the true Wasserstein distance and is crucial for consistent performance. Contrary to SIL and PWIL, we focus on ILfO, giving new results and insights into the capabilities of OT in this context, and show that our approach matches or outperforms existing state-of-the-art methods.

4 Wasserstein Imitation Learning from Observational Demonstrations

In this section, we introduce our approach for minimizing the Wasserstein distance between expert trajectories and learner rollouts. To do so, we derive a reward function based on the distance between state transitions in pairs of trajectories.

Deriving a reward from the Wasserstein distance. With the absence of a true reward signal, the ILfO setting can be framed as a divergence-minimization problem, where the objective is to match the trajectory distributions of the learner and the expert. In our case, we choose the Wasserstein distance as a metric for this task. Unlike the widely used KL divergence, the Wasserstein distance is defined for distributions with non-overlapping support, making it amenable to scenarios where the behavior of the learner and the expert may be particularly distinct. We can define our ILfO task as minimizing the Wasserstein distance W_1 between trajectories τ_π sampled from the learner policy π and example trajectories τ_E provided by an expert E:

$$\min_\pi \mathbb{E}_{\tau_\pi, \tau_E} \left[W_1(\tau_\pi, \tau_E) \right] = \min_\pi \mathbb{E}_{\tau_\pi, \tau_E} \left[\min_P \left(\sum_{i=0}^{T} \sum_{j=0}^{T} d((s_i, s_{i+1})_\pi, (s_j, s_{j+1})_E) P_{i,j} \right) \right]. \tag{4}$$

As the Wasserstein distance between a pair of trajectories can be defined as a sum over each of the transitions in each trajectory, for a given coupling matrix P, we can define a reward function

$$\tilde{r}_t(s_t, s_{t+1} | \tau_\pi, \tau_E, P) := -\sum_{j=0}^{T} d((s_t, s_{t+1})_\pi, (s_j, s_{j+1})_E) P_{t,j}, \tag{5}$$

such that summing the reward $\tilde{r}_t$ over a learner trajectory τ_π is equal to the Wasserstein distance

$$W_1(\tau_\pi, \tau_E) = \min_P \left(-\sum_{i=0}^{T} \tilde{r}_t(s_t, s_{t+1} | \tau_\pi, \tau_E, P) \right). \tag{6}$$

This naturally suggests an objective that involves the sum of rewards $\tilde{r}_t$ over learner trajectories

$$J(\pi | E, P) := \mathbb{E}_{\pi, E} \left[\sum_{t=0}^{T} \tilde{r}_t(s_t, s_{t+1} | \tau_\pi, \tau_E, P) \right], \tag{7}$$

where our original objective (Equation (4)) can be recovered:

$$\max_\pi \min_P J(\pi | E, P) = \min_\pi \mathbb{E}_{\tau_\pi, \tau_E} \left[W_1(\tau_\pi, \tau_E) \right]. \tag{8}$$

As the optimal coupling matrix P can be approximated by the iterative Sinkhorn algorithm (Sinkhorn, 1967), the maximization of the objective J with any RL algorithm, can be used as a replacement to minimizing the Wasserstein distance.

Off-policy minimization of the Wasserstein distance. As the reward $\tilde{r}_t(s_t, s_{t+1}|\tau_{\pi_n}, \tau_E, P)$ is defined as a function of a trajectory τ_{π_n} gathered by the learner π_n, any stale reward determined by trajectories from a previous policy π_{n-m}, $m \geq 1$, will not correspond with the Wasserstein distance of the current learner (as noted in Equation (6)). However, working with the assumption that a policy π_n is better than any previous policy with respect to J, (i.e. $J(\pi_n) \geq J(\pi_{n-m})$ where $m \geq 1$), we remark that stale rewards provide an upper bound on the Wasserstein distance:

$$W_1(\tau_{\pi_n}, \tau_E) = \min_P \left(-\sum_{i=0}^{T} \tilde{r}_t(s_t, s_{t+1}|\tau_{\pi_n}, \tau_E, P) \right) \leq \min_P \left(-\sum_{i=0}^{T} \tilde{r}_t(s_t, s_{t+1}|\tau_{\pi_{n-m}}, \tau_E, P) \right). \quad (9)$$

This means that previously collected off-policy trajectories can be used for learning in a principled manner, at the cost of the tightness of the upper bound of the Wasserstein distance. In our experimental results, we show that reusing prior data dramatically improves the sample efficiency of our algorithm over approaches which rely exclusively on online data (Papagiannis & Li, 2020).

Our final approach, Observational Off-Policy Sinkhorn (OOPS) discovers a reward function in a similar manner to existing approaches (Papagiannis & Li, 2020; Dadashi et al., 2020), but in state transition space rather than state-action space. Unlike these prior approaches, OOPS avoids complexities such as adversarial learning or heuristic-based design of the reward function with multiple hyperparameters. OOPS is summarized in Algorithm 1.

Algorithm 1 OOPS

1: **Input:** Dataset of expert demonstrations D_E.
2: **for** episodes $n = 1, ..., N$ **do**
3: Collect a trajectory from the environment.
4: Compute the coupling matrix P using the Sinkhorn algorithm (Sinkhorn, 1967).
5: Compute the reward $\tilde{r}$ with D_E and P (Equation (5)).
6: Train learner with a RL algorithm, and the collected trajectories and reward $\tilde{r}$.

5 Experiments

5.1 Results

We evaluate our algorithm on five MuJoCo locomotion benchmark environments from the OpenAI Gym suite (Todorov et al., 2012; Brockman et al., 2016), and three robotics tasks (Coumans & Bai, 2016; Tan et al., 2018) in the ILfO setting. For each environment, the dataset of expert trajectories D_E is generated via a pre-trained Soft Actor-Critic agent (Haarnoja et al., 2018a).

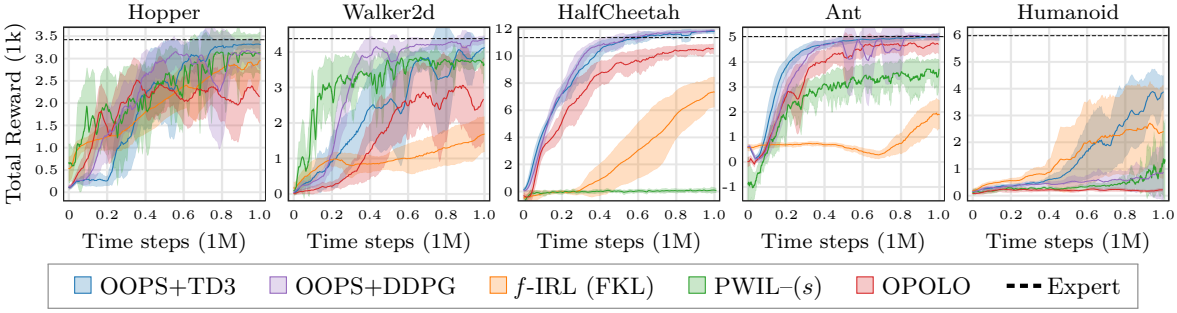

Figure 1: Learning curves for 1 expert demonstrations across 5 random seeds. The shaded area represents a standard deviation. OOPS+TD3 consistently matches or outperforms the baseline approaches.

# Expert Traj.	Algorithm	Hopper 3420 ± 36	Walker2d 4370 ± 124	HalfCheetah 11340 ± 95	Ant 5018 ± 140	Humanoid 5973 ± 17
1	f-IRL (FKL)	0.91 ± 0.03	0.42 ± 0.10	0.63 ± 0.13	0.47 ± 0.10	0.47 ± 0.32
	OPOLO	0.73 ± 0.09	0.80 ± 0.14	0.88 ± 0.02	0.89 ± 0.04	0.04 ± 0.01
	SIL – (s, s')	0.17 ± 0.06	0.07 ± 0.02	-0.17 ± 0.09	-0.41 ± 0.07	0.07 ± 0.00
	PWIL – (s)	0.91 ± 0.14	0.71 ± 0.30	0.01 ± 0.01	0.76 ± 0.05	0.14 ± 0.14
	OOPS+DDPG (Ours)	0.90 ± 0.10	**0.99 ± 0.03**	**1.05 ± 0.01**	**1.00 ± 0.02**	0.16 ± 0.20
	OOPS+TD3 (Ours)	**0.98 ± 0.02**	**0.95 ± 0.09**	**1.05 ± 0.01**	**1.00 ± 0.03**	**0.74 ± 0.04**
4	f-IRL (FKL)	0.92 ± 0.04	0.38 ± 0.12	0.69 ± 0.12	0.38 ± 0.07	0.51 ± 0.28
	OPOLO	0.72 ± 0.15	0.91 ± 0.03	0.90 ± 0.02	**1.02 ± 0.04**	0.20 ± 0.12
	SIL – (s, s')	0.25 ± 0.07	0.09 ± 0.03	-0.22 ± 0.14	-0.61 ± 0.22	0.07 ± 0.01
	PWIL – (s)	**0.98 ± 0.02**	0.88 ± 0.03	0.00 ± 0.02	0.78 ± 0.03	0.23 ± 0.28
	OOPS+DDPG (Ours)	0.75 ± 0.34	**0.96 ± 0.03**	**1.05 ± 0.01**	**0.99 ± 0.01**	0.07 ± 0.01
	OOPS+TD3 (Ours)	**0.94 ± 0.07**	**0.97 ± 0.01**	**1.05 ± 0.01**	**0.99 ± 0.03**	**0.65 ± 0.15**
10	f-IRL (FKL)	0.91 ± 0.05	0.39 ± 0.09	0.65 ± 0.10	0.39 ± 0.17	0.40 ± 0.22
	OPOLO	0.66 ± 0.08	**0.96 ± 0.04**	0.95 ± 0.01	**1.00 ± 0.03**	0.16 ± 0.06
	SIL – (s, s')	0.17 ± 0.09	0.08 ± 0.03	-0.20 ± 0.09	-0.24 ± 0.11	0.07 ± 0.00
	PWIL – (s)	**0.98 ± 0.01**	0.87 ± 0.08	0.01 ± 0.02	0.78 ± 0.04	0.23 ± 0.28
	OOPS+DDPG (Ours)	**0.93 ± 0.03**	0.78 ± 0.39	**1.03 ± 0.04**	0.79 ± 0.38	0.21 ± 0.25
	OOPS+TD3 (Ours)	**0.97 ± 0.01**	**0.95 ± 0.03**	**1.05 ± 0.01**	**1.00 ± 0.02**	**0.64 ± 0.22**

Table 1: Final performance of different ILfO algorithms at 1M timesteps, using 1, 4, 10 expert demonstrations. Values for each task are normalized by the average return of the expert. ± captures the standard deviation. The highest value and any within 0.05 are **bolded**. The average un-normalized return of the expert is listed below each task. All results are averaged across 5 seeds and 10 evaluations.

We use OOPS to generate a reward function for two RL algorithms, TD3 (Fujimoto et al., 2018) and DDPG (Lillicrap et al., 2015). Our baselines include state-of-the-art ILfO methods: f-IRL (Ni et al., 2020) (its best-performing FKL variant in particular) and OPOLO (Zhu et al., 2020), as well as IL methods which also consider the Wasserstein distance: Primal Wasserstein Imitation Learning (PWIL) (Dadashi et al., 2020) and Sinkhorn Imitation Learning (SIL) (Papagiannis & Li, 2020). In order to compare algorithms in the ILfO setting, we use the state-only version of PWIL, PWIL–(s) (Dadashi et al., 2020), and modify SIL (Papagiannis & Li, 2020) by replacing the action a in all pairs (s, a) with the corresponding next state s' in the transition. All algorithms are given a budget of 1M environment interactions (and 1M updates), are evaluated on 5 random seeds, and use the original implementations provided by the authors.

Locomotion. We report the evaluation results of our approach compared against the four baseline algorithms in Table 1, varying the number of expert demonstrations used for imitation. The learning curves for the single demonstration setting are shown in Figure 1.

OOPS+TD3 consistently matches or outperforms all baseline methods regardless of task and number of expert demonstrations. OOPS+DDPG roughly matches the performance of the expert in every environment other than Humanoid. The poor results on Humanoid are unsurprising, as previous results have demonstrated that DDPG tends to fail at the Humanoid task in the standard RL setting (Haarnoja et al., 2018a). Regardless, since DDPG is known to underperform TD3 and SAC, matching the performance of the SAC expert suggests that the OOPS reward function can produce a stronger learning signal than the original task reward. This

# Expert Traj.		BipedalWalker 318.90 ± 9.20	Minitaur 12.36 ± 0.75	MinitaurDuck 10.68 ± 1.20
1	OPOLO	0.96 ± 0.01	0.76 ± 0.08	**1.00 ± 0.04**
	PWIL – (s)	0.89 ± 0.01	0.53 ± 0.19	0.30 ± 0.14
	OOPS+TD3	**0.93 ± 0.01**	**1.01 ± 0.04**	**0.94 ± 0.18**
4	OPOLO	**0.96 ± 0.01**	0.84 ± 0.09	**1.01 ± 0.03**
	PWIL – (s)	0.90 ± 0.01	0.52 ± 0.15	0.21 ± 0.09
	OOPS+TD3	**0.92 ± 0.01**	**0.91 ± 0.09**	**1.02 ± 0.05**
10	OPOLO	**0.98 ± 0.00**	**0.98 ± 0.04**	**1.00 ± 0.02**
	PWIL – (s)	0.88 ± 0.01	0.58 ± 0.09	0.15 ± 0.16
	OOPS+TD3	**0.93 ± 0.01**	**1.03 ± 0.03**	**0.99 ± 0.09**

Table 2: Final performance of ILfO algorithms when using 1, 4, and 10 expert demonstrations. Values for each task are normalized by the average return of the expert. ± captures the standard deviation. The highest value and any within 0.05 are **bolded**. The average un-normalized return of the expert is listed below each task. Results are averaged across 5 seeds and 10 evaluations.

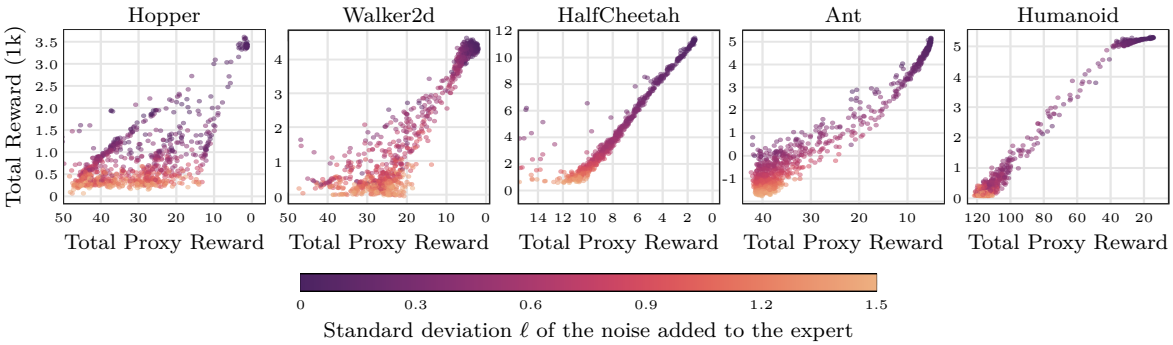

Figure 2: Calibration plot comparing the proxy reward with the original reward function of the benchmark domains. Each point represents the average of the sum of each reward function, over 5 trajectories. Trajectories are generated by adding noise $\mathcal{N}(0, \ell^2)$ to the expert policy. The calibration plots show a strong correlation between the proxy reward and the true task reward.

result indicates that OOPS might not be dependent on the choice of the RL algorithm, assuming the RL algorithm is capable of solving the desired task.

Additional environments. For the top three performing algorithms (OPOLO, PWIL–(s), and OOPS+TD3), we benchmark on three additional robotic-centric tasks in Table 2. While OOPS+TD3 and OPOLO achieve a similar high performance when using all 10 expert demonstrations, OOPS+TD3 surpasses OPOLO when using fewer demonstrations.

5.2 Analysis and Ablations

To better understand the performance of our approach, in this section, we perform additional analysis to test the quality and importance of various components. These results fill the gap in knowledge left by previous work leveraging the Wasserstein distance in IL, examining hyperparameters such as the regularization parameter in the Sinkhorn distance or the effect of using different distance metrics, and provide direct comparison between the various approximations available to use when comparing policy trajectories.

Accuracy of proxy reward. OOPS generates a proxy reward function that minimizes the Wasserstein distance between the learner's trajectories and the demonstrated expert trajectories. We evaluate the correlation between this proxy reward and the true environment reward. To do so, we collect a dataset of varied trajectory quality using the expert policy from the main results, with added Gaussian noise $\mathcal{N}(0, \ell^2)$ with $\ell \in [0, 1.5]$. Figure 2 shows the calibration plots between the proxy reward and the original task reward, showing a strong correlation in every environment.

Next, we compare the quality of trajectories in terms of the Wasserstein distance rather than the true environment reward. In Table 3, we compare the Wasserstein distance between the expert trajectories and the final policy rollouts obtained at the end of training from each of the top-3 performing methods (OOPS, OPOLO, PWIL–(s)). The Wasserstein distance is measured in three spaces: state-only (s), state-transition (s, s'), and state-action (s, a).

Environment	Hopper			Walker2d			HalfCheetah			Ant			Humanoid		
Space	(s)	(s, s')	(s, a)	(s)	(s, s')	(s, a)	(s)	(s, s')	(s, a)	(s)	(s, s')	(s, a)	(s)	(s, s')	(s, a)
OPOLO	5.91	8.40	6.33	3.02	4.32	3.47	**1.60**	**2.39**	**1.91**	4.64	7.24	**5.05**	80.75	114.53	81.90
PWIL – (s)	1.74	2.56	2.38	**2.04**	**2.96**	**2.78**	6.48	9.27	6.93	**3.83**	6.00	5.90	53.52	76.06	54.94
OOPS+TD3	**1.66**	**2.38**	**2.06**	2.28	3.27	3.02	**1.63**	**2.41**	**2.01**	**3.83**	**5.90**	**5.17**	**25.64**	**37.03**	**27.63**

Table 3: Final Wasserstein distance in state occupancy, state transition, and state-action space of the 10 final trained agent rollouts to the expert trajectories for different ILfO algorithms, lower is better. We highlight in blue the best performing agent in state-action space, considered ground truth in this experiment, and **bold** the best performing agent according to each metric. Agents were trained using 10 expert demonstration trajectories, for 1M timesteps. Distances are averaged over 10 reference expert trajectories.

We find that OOPS obtains the lowest state-action Wasserstein distance to the expert trajectories in four of the five studied environments, with Walker2d being the only disagreement with the previous experiment, as even though OOPS+TD3 obtains a better task reward in Table 1, PWIL–(s) obtains a lower state-action Wasserstein distance to the expert.

Finally, to further evaluate the quality of the Wasserstein distance used by PWIL, we take OOPS and replace the Sinkhorn algorithm with the greedy formulation W_{greedy} proposed by PWIL to compute the Wasserstein distance in (s, s') space. The results are reported in Table 4 (under W_{greedy}), and show a loss in performance.

Quality of estimated Wasserstein distance. In Figure 3, we compare the quality of different approximations of the state transition Wasserstein distance: the Sinkhorn distance W_{Sk} with varying λ, the network simplex solver W_{simplex} introduced in (Bonneel et al., 2011), and W_{greedy} proposed for PWIL (Dadashi et al., 2020). Additional results can be found in the Appendix.

To compare each approach, we compute the Wasserstein distance between trajectories generated by the final policy of OOPS+TD3 and the expert trajectories, using each of the various approximations. Each method results in different estimates of the coupling matrix P; they provide an upper bound on the true Wasserstein distance, where lower estimates of the Wasserstein distance are a tighter bound. We find that for very low values of λ, W_{Sk} computes lower cost couplings than W_{simplex}, and up to $\lambda \approx 0.4$ obtains better approximations than W_{greedy}.

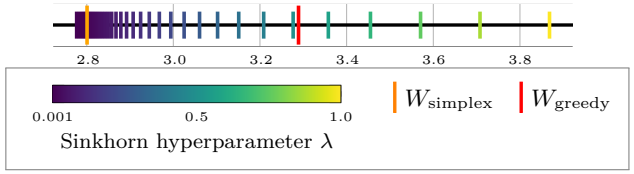

Figure 3: Wasserstein distances between the 10 final rollout trajectories of OOPS+TD3 and the expert on the Hopper environment, using different solvers for the coupling matrix P (W_{greedy} and W_{simplex}) compared against the Sinkhorn distance W_{Sk} when varying the parameter λ. Results are averaged over 10 expert trajectories. The Sinkhorn distance, for low enough values of λ computes a tighter upper bound to the Wasserstein distance estimates than W_{greedy} (Dadashi et al., 2020). Results for the other environments can be found in the Appendix.

Next, we compare these three approaches for computing the Wasserstein distance in terms of performance. The results are shown in Table 4 (Wasserstein Distance Solver). Unsurprisingly, large values of λ, which approximate the Wasserstein distance W_1 poorly, results in lower performance. For sufficiently small values of λ, we find that OOPS+TD3 maintains a consistent performance. This suggests that λ can generally be ignored and left to a default value.

Finally, we attempt different settings for the Wasserstein distance. In Table 4 we display the change in performance from OOPS when using W_1 or W_2 when the distance metric d is the Euclidean distance $||\cdot||_2$, and W_1 when d is the cosine distance. OOPS uses W_1 with the square root of the Euclidean distance, which de-emphasizes large differences in magnitude in a similar fashion to the cosine distance. We find that this choice

	Hopper	Walker2d	HalfCheetah	Ant	Humanoid
Occupancy (Default: (s, s'))					
State only	0.10	-33.93	-0.57	0.30	-0.94
Wasserstein Distance Solver (Default: $\lambda = 0.05$)					
W_{greedy}	-14.99	-7.75	-45.46	-0.79	-19.21
W_{simplex}	-10.91	-6.09	-1.03	-2.40	-33.99
$\lambda = 0.005$	-3.12	-2.65	-0.35	-1.48	-2.34
$\lambda = 0.1$	-1.72	-3.87	-1.39	-3.88	-9.18
$\lambda = 0.5$	-58.64	-25.20	-15.09	-10.95	-24.44
Distance Metric (Default: $W_1, d = \sqrt{\|\|\cdot\|\|_2}$)					
$W_2, d = \|\|\cdot\|\|_2$	-36.52	-21.83	-11.88	-16.10	-47.92
$W_1, d = \|\|\cdot\|\|_2$	-4.61	-1.42	-1.73	-1.95	-22.80
$W_1, d = \cos$	0.13	-10.04	-4.09	-2.84	-34.63
Adversarial Distance (Default: Unused)					
SIL – (s, s')	-82.50	-91.61	-119.10	-124.88	-90.83
OOPS_{adv}	-21.09	-76.58	-101.84	-17.68	-97.87

Table 4: Results for different variations of OOPS in terms of *percent difference*. All results use 10 expert trajectories and are averaged across 5 seeds and 10 evaluations. State only uses W_1 over (s) rather than (s, s'). Wasserstein distance solver modifies the solver used by OOPS to determine the coupling matrix P. Adversarial distance refers to the use of the adversarial distance function from SIL (Papagiannis & Li, 2020) and also includes the full SIL method for comparison.

of d provides significant benefits in high dimensional domains (Humanoid) where magnitudes matter but can vary significantly. We also compare with the learned adversarial distance metric used by SIL (Papagiannis & Li, 2020) (denoted OOPS_{adv}) and find that while this version outperforms vanilla SIL, the adversarial component is harmful.

Transition vs. state occupancy. For OOPS, we define trajectories by their state-next-state transitions (s, s'), rather than individual states s. Matching based on states can potentially admit multiple minimums since trajectories with the same states out of order can still minimize the state occupation distributional distance. Furthermore, if the reward function is based on state *and* action, then it is clear that only matching state occupancy is insufficient. Since expert actions are unavailable in the ILfO setting, we must rely on (s, s'). We posit that enforcing a local ordering of states provides a higher fidelity signal for ILfO. We validate this empirically in our ablations (Table 4 - Occupancy). While using state-only occupancy matches the performance of OOPS+TD3 in most environments, there is a large drop in performance in Walker2d. This aligns with our intuition: matching by state occupancy will often work but can be problematic in certain environments depending on the state representation and transition dynamics.

6 Conclusion

In this paper, we introduce OOPS, an ILfO algorithm that produces a reward function that minimizes the Wasserstein distance between the state transition trajectory of the expert and the imitation agent. We validate our approach through extensive experiments and demonstrate that OOPS surpasses the current state-of-the-art methods in the ILfO setting across benchmark and robotics domains. Combined with off-policy RL, OOPS exhibits exceptional sample efficiency and low variance in performance, key qualities for the practical deployment of IL algorithms on real systems.

References

Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hk4_qw5xe.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. *ACM Trans. Graph.*, 30(6):1–12, dec 2011. ISSN 0730-0301. doi: 10.1145/2070781.2024192. URL https://doi.org/10.1145/2070781.2024192.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Wei-Di Chang, Juan Camilo Gamboa Higuera, Scott Fujimoto, David Meger, and Gregory Dudek. Il-flow: Imitation learning from observation using normalizing flows. *arXiv preprint arXiv:2205.09251*, 2022.

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Robert Dadashi, Leonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. In *International Conference on Learning Representations*, 2020.

Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah Hanna, Peter Stone, and AI Sony. An imitation from observation approach to sim-to-real transfer. 2020.

Ishan Durugkar, Mauricio Tec, Scott Niekum, and Peter Stone. Adversarial intrinsic motivation for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8622–8636, 2021.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, volume 80, pp. 1587–1596. PMLR, 2018.

Scott Fujimoto, David Meger, and Doina Precup. An equivalence between loss functions and non-uniform sampling in experience replay. *Advances in neural information processing systems*, 33: 14219–14230, 2020.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pp. 1259–1277. PMLR, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, volume 80, pp. 1861–1870. PMLR, 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

Siddhant Haldar, Vaibhav Mathur, Denis Yarats, and Lerrel Pinto. Watch and match: Supercharging imitation with regularized optimal transport. In *Conference on Robot Learning*, pp. 32–43. PMLR, 2023.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pp. 4565–4573, 2016.

Andrew Jaegle, Yury Sulsky, Arun Ahuja, Jake Bruce, Rob Fergus, and Greg Wayne. Imitation by predicting observations. In *International Conference on Machine Learning*, pp. 4665–4676. PMLR, 2021.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pp. 651–673, 2018.

Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.

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

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*, 2018.

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 *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Hk4fpoA5Km.

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

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*, 2020.

Georgios Papagiannis and Yunpeng Li. Imitation learning with sinkhorn distances. *arXiv preprint arXiv:2008.09167*, 2020.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

Jan Stanczuk, Christian Etmann, Lisa Maria Kreusser, and Carola-Bibiane Schönlieb. Wasserstein gans work because they fail (to approximate the wasserstein distance). *arXiv preprint arXiv:2103.01678*, 2021.

Wen Sun, Anirudh Vemula, Byron Boots, and Drew Bagnell. Provably efficient imitation learning from observation alone. In *International Conference on Machine Learning*, pp. 6036–6045. PMLR, 2019.

Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.010.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033. IEEE, 2012.

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

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

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

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

M. Zhang, Y. Wang, X. Ma, L. Xia, J. Yang, Z. Li, and X. Li. Wasserstein distance guided adversarial imitation learning with reward shape exploration. In *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 1165–1170, Nov 2020. doi: 10.1109/DDCLS49620.2020.9275169.

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.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438, 2008.

A Additional Results and Experiments

A.1 Comparing Solvers for the State Transition Wasserstein Distance

We show in Figure 4 the full set of results for the comparison of solvers used when computing the Wasserstein distance. See Section 5.2 for the description and discussion of this experiment.

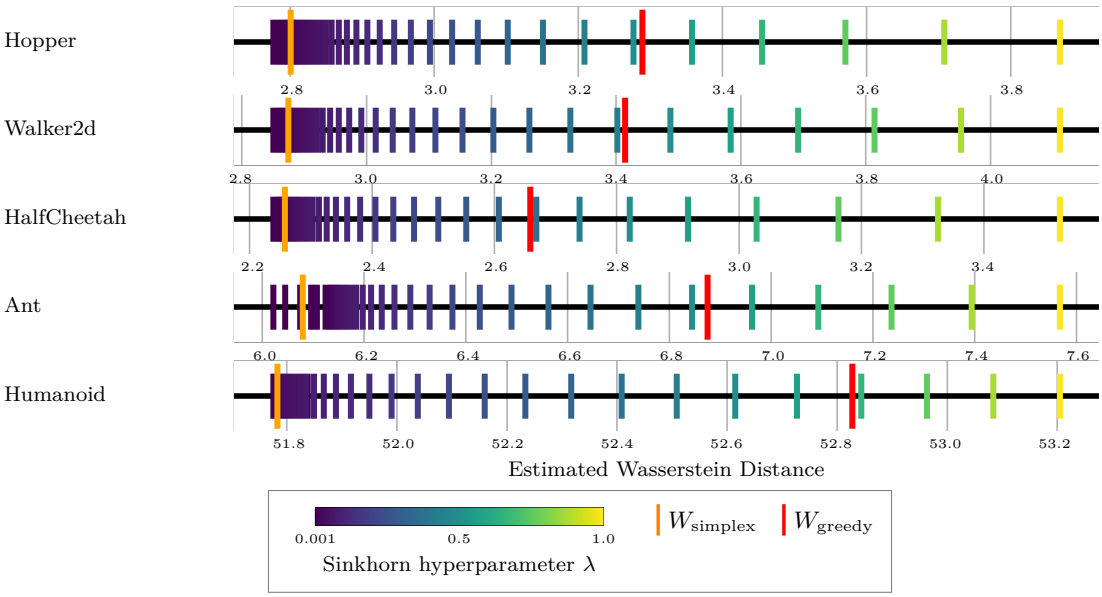

Figure 4: Wasserstein distances between the 10 final rollout trajectories of OOPS+TD3 and the expert, using different solvers for the coupling matrix P (W_{greedy} and W_{simplex}) compared against the Sinkhorn distance W_{Sk} when varying the parameter λ. Results are averaged over 10 expert trajectories. The Sinkhorn distance, for low enough values of λ computes a tighter upper bound to the Wasserstein distance estimates than W_{greedy} (Dadashi et al., 2020).

B Experimental Details

In Table 5, we list the hyperparameters used for TD3 (Fujimoto et al., 2018), our underlying off-policy RL algorithm. On top of these hyperparameters, we use the PAL variant of TD3 for the loss function of the critic (Fujimoto et al., 2020). In Table 6, we list the hyperparameters for the computation of the Sinkhorn distance (Cuturi, 2013) used for OOPS across all experiments, except experiments studying the effect of specific hyperparameters (distance metric and λ).

Parameter	Value
τ	3e-3
Exploration noise	2e-1
Policy noise	1e-1
Actor network architecture (hidden)	[256]
Critic network architecture (hidden)	[1024]
Actor LR	3e-4
Critic LR	3e-4
Optimizer	Adam
Actor non linearity	ReLU
Critic non linearity	ReLU

Table 5: TD3 hyperparameters

Parameter	Value
Maximum number of iterations	20000
λ	0.05
Distance metric	$\sqrt{\|\|\cdot\|\|_2}$

Table 6: Sinkhorn distance computation hyperparameters