A Background on Optimal Transport

Optimal transport (Villani, 2009; Peyré & Cuturi, 2019) tools allow us to compare probability measures while incorporating the geometry of the space. The entropic 2-Wasserstein distance between two discrete measures $\mu_{\boldsymbol{x}} = \frac{1}{T}\sum_{t=1}^{T} \delta_{\boldsymbol{x}_t}$ and $\mu_{\boldsymbol{y}} = \frac{1}{T}\sum_{t=1}^{T} \delta_{\boldsymbol{y}_t}$ is

$$\mathcal{W}_\epsilon^2(\mu_{\boldsymbol{x}}, \mu_{\boldsymbol{y}}) = \min_{\gamma \in \Gamma} \sum_{t,t'=1}^{T} d^2(\boldsymbol{x}_t, \boldsymbol{y}_{t'})\gamma_{t,t'} - H(\pi), \qquad H(\gamma) = -\sum_{t,t'=1}^{T} \gamma_{t,t'} \log \gamma_{t,t'}, \tag{9}$$

where $\Gamma = \{\gamma \in \mathbb{R}^{T\times T} : \gamma\mathbf{1} = \gamma^T\mathbf{1} = \frac{1}{T}\mathbf{1}\}$ is the set of coupling matrices, and d is a metric. Adding the entropic term has two main benefits: i) the induced problem can be solved in quadratic time via Sinkhorn's algorithm (Sinkhorn & Knopp, 1967), and ii) the resulting distance is smooth in the measures' samples. Intuitively, the optimal coupling matrix γ provides an alignment of the samples of μ_x, μ_y which minimizes the cost of transport between these. The distance then consists of the weighted sum of distances between aligned samples, along with an entropic penalty.

B Algorithms

Algorithm 1 Inverse reinforcement learning core. Different methods can be instantiated by changing the rewarder function.

for $t \in T_{total}$ **do**
  if done **then**
    $r_{1:T} = \text{rewarder}(\text{episode})$
    Update episode with $r_{1:T}$ and add all quadruples $[\boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{o}_{t+1}, r_t]$ to $\mathcal{D}$.
    $\boldsymbol{o}_t = \text{env.reset}()$, done $=$ False, episode $= [\,]$
  end if
  $\boldsymbol{a}_t \sim \pi(\cdot|\boldsymbol{o}_t) \rightarrow \boldsymbol{o}_{t+1}, \text{done} = \text{env.step}(\boldsymbol{a}_t), \text{episode.append}([\boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{o}_{t+1}])$
  Update DrQ-v2's actor and critic, and rewarder-specific functions using $\mathcal{D}$.
end for

C HYPERPARAMETERS

We provide a list of all hyperparameters in table 1. The only environment-specific variation is that the number of n-steps is set to 1 for walker tasks.

Agent	Parameter	Value 1
Common	Replay buffer size	150000
	Learning rate	$1e^{-4}$
	Exploration Schedule	$\text{linear}(1, 0.1, 500000)$
	Discount	0.99
	n-step returns	3
	Action repeat	2
	Frame stack	3
	Seed frames	2000
	Exploration steps	4000
	Mini-batch size	256
	Agent update frequency	2
	Critic soft-update rate	0.01
	Features dim	50
	Hidden dim	1024
	Optimizer	Adam
P-SIL	Target update frequency	10000
	Reward scale factor	10
P-DAC	Gradient penalty coefficient λ	10

Table 1: List of hyperparameters.

D EXTRA ABLATION PLOTS

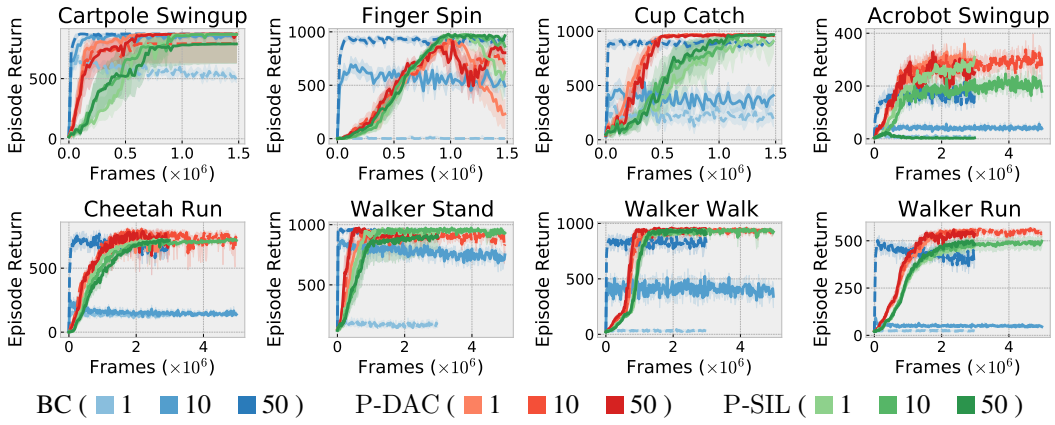

Figure 9: Ablation study on the number of demonstration. We train all P-DAC, P-SIL and BC on $1, 10$ and 50 demonstrations. We observe P-DAC and P-SIL are robust to the number of demonstrations, while BC is not, and fails on most tasks for the 1-demonstration setting.

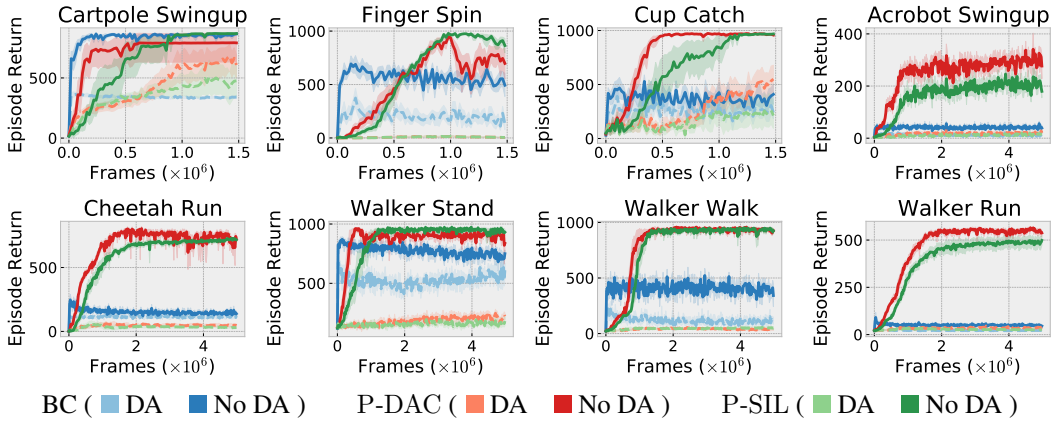

Figure 10: Ablation study on data augmentation (DA). We compare P-DAC, P-SIL and BC with and without data augmentation. We observe the gap in performance to be significantly larger for P-DAC and P-SIL, showing that DA is a crucial component of our methods.

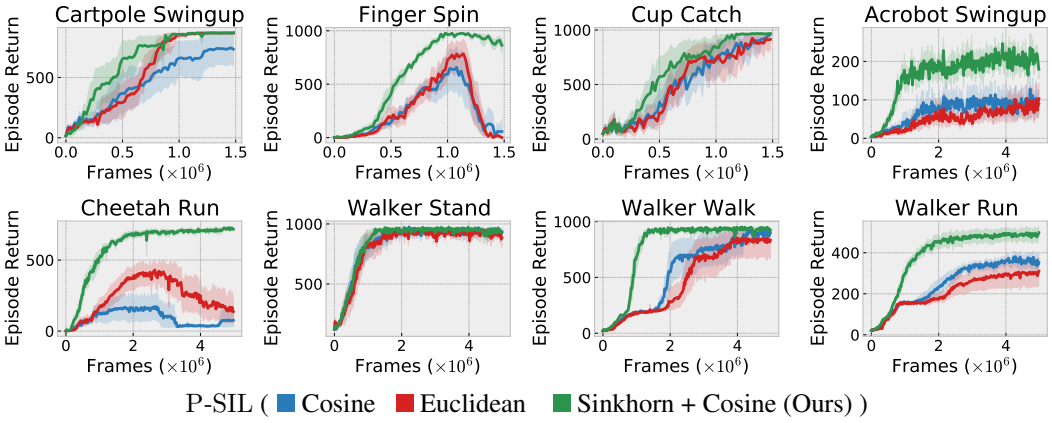

Figure 11: Ablation study on the distance between trajectories for P-SIL. We compare our approach (OT alignment with cosine cost), to the cosine distance and the Euclidean distance between trajectories (without OT). We observe that OT significantly outperform non-OT approaches. Hence, OT alignment is also an essential component of P-SIL.

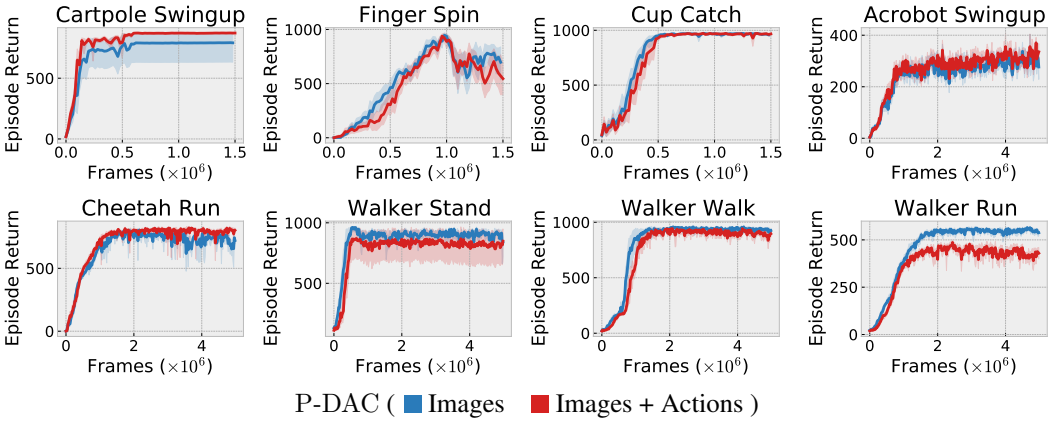

Figure 12: Ablation study on the need for expert actions. We train P-DAC with and without actions concatenated to observations (the former is privileged). We observe the performance gap to be marginal, hence actions are not necessary in the considered environments, contrarily to popular beliefs on GAIL-like algorithms.