

457 Appendix

458 6.1 Code Release

459 Our code is attached as part of the supplementary material to ensure reproducibility.

460 6.2 Theory

461 6.2.1 Derivation for Action-free distribution matching

462 **Theorem 6.1.** *The dual problem to the primal occupancy matching objective (Equation 4) is given*
 463 *by the DILO objective in Equation 5. Moreover, as strong duality holds from Slater’s conditions the*
 464 *primal and dual share the same optimal solution d^* for any offline transition distribution ρ and any*
 465 *choice of mixture distribution ratio β .*

466 We start with the primal objective that matches distributions between the agent’s visitation $d(s, s', a')$
 467 and expert’s visitation $d^E(s, s', a')$. As before ρ denotes the visitation distribution of offline data.

$$\min_{\pi} \mathcal{D}_f(\text{Mix}_{\beta}(d^{\pi}(s, s', a'), \rho) \| \text{Mix}_{\beta}(d^E(s, s', a'), \rho)), \quad (7)$$

468 where for any two distributions μ_1 and μ_2 , $\text{Mix}_{\beta}(\mu_1, \mu_2)$ denotes the mixture distribution with
 469 coefficient $\beta \in (0, 1]$ defined as $\text{Mix}_{\beta}(\mu_1, \mu_2) = \beta\mu_1 + (1 - \beta)\mu_2$.

470 Formulating the objective as a constrained objective in agent’s visitation distribution d allows us to
 471 create a primal objective that is a convex program. This is crucial in subsequently creating a dual
 472 objective that is unconstrained and easy to optimize.

$$\begin{aligned} & \max_{d \geq 0} -\mathcal{D}_f(\text{Mix}_{\beta}(d, \rho) \| \text{Mix}_{\beta}(d^E, \rho)) \\ \text{s.t. } & \sum_{a''} d(s', s'', a'') = (1 - \gamma)d_0(s', s'') + \gamma \sum_{s, a' \in \mathcal{S} \times \mathcal{A}} d(s, s', a') p(s'' | s', a'), \quad \forall s', s'' \in \mathcal{S} \times \mathcal{S}. \end{aligned} \quad (8)$$

473 where the constraints above dictate the conditions that any valid visitation distribution $d(s', s'')$ needs
 474 to satisfy and are our proposed modifications to the commonly known *bellman flow constraints*.

475 Below we outline the derivation of how these specific constraints with the mixture distribution
 476 matching objective allows us to create a dual objective that is independent of expert’s actions.
 477 Applying Lagrangian duality to the above constrained distribution matching objective, we can convert
 478 it to an unconstrained problem with dual variables $V(s, s')$ defined for all $s, s' \in \mathcal{S} \times \mathcal{S}$:

$$\begin{aligned} & \max_{d \geq 0} \min_{V(s', s'')} -D_f(\text{Mix}_{\beta}(d, \rho)(s, s', a') \| \text{Mix}_{\beta}(d^E, \rho)(s, s', a')) \\ & + \sum_{s', s''} V(s', s'') \left((1 - \gamma)d_0(s', s'') + \gamma \sum_{s, a'} d(s, s', a') p(s'' | s', a') - \sum_a d(s', s'', a'') \right) \end{aligned} \quad (9)$$

$$\begin{aligned} & = \max_{d \geq 0} \min_{V(s, s')} (1 - \gamma) \mathbb{E}_{d_0(s, s')} [V(s, s')] + \mathbb{E}_{s, s', a' \sim d} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\ & - D_f(\text{Mix}_{\beta}(d, \rho)(s, s', a') \| \text{Mix}_{\beta}(d^E, \rho)(s, s', a')) \end{aligned} \quad (10)$$

479 where the last equation uses a change of variable from s', s'' to s, s' without loss of generality. Using
 480 a simple algebraic manipulation below, we can get rid of the inner maximization. We add and subtract

the terms shown below:

$$\begin{aligned}
&= \max_{d \geq 0} \min_{V(s, s')} \beta(1 - \gamma) \mathbb{E}_{d_0(s, s')} [V(s, s')] \\
&\quad + \beta \mathbb{E}_{s, s', a' \sim d} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\
&\quad + (1 - \beta) \mathbb{E}_{s, s', a' \sim \rho} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\
&\quad - (1 - \beta) \mathbb{E}_{s, a, g \sim \rho} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\
&\quad - D_f(\text{Mix}_\beta(d, \rho)(s, s', a') \parallel \text{Mix}_\beta(d^E, \rho)(s, s', a'))
\end{aligned} \tag{11}$$

As strong duality holds using Slater's conditions [52] (see [43] for a detailed account of strong duality in RL under visitation distributions). Using the fact that strong duality holds in this problem we can swap the inner max and min and rewrite an equivalent maximization under the mixture distribution:

$$\begin{aligned}
&= \min_{V(s, s')} \max_{\text{Mix}_\beta(d, \rho)(s, s', a') \geq 0} \beta(1 - \gamma) \mathbb{E}_{d_0(s, s')} [V(s, s')] \\
&\quad + \beta \mathbb{E}_{s, s', a' \sim d} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\
&\quad + (1 - \beta) \mathbb{E}_{s, s', a' \sim \rho} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\
&\quad - (1 - \beta) \mathbb{E}_{s, s', a' \sim \rho} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\
&\quad - D_f(\text{Mix}_\beta(d, \rho)(s, s', a') \parallel \text{Mix}_\beta(d^E, \rho)(s, s', a'))
\end{aligned} \tag{12}$$

In the following derivation, we will show that the inner maximization in Eq 12 has a closed form solution even when adhering to the non-negativity constraints. Let $y(s, s', a') = \mathbb{E}_{s'' \sim p(s', a')} [V(s', s'')] - V(s, s')$.

$$\begin{aligned}
&\max_{\text{Mix}_\beta(d, \rho)(s, s', a') \geq 0} \mathbb{E}_{s, s', a' \sim \text{Mix}_\beta(d, \rho)(s, s', a')} \left[\gamma \sum_{s''} p(s'' | s', a') V(s', s'') - V(s, s') \right] \\
&\quad - D_f(\text{Mix}_\beta(d, \rho)(s, s', a') \parallel \text{Mix}_\beta(d^E, \rho)(s, s', a'))
\end{aligned}$$

Now to solve this constrained optimization problem we create the Lagrangian dual and study the KKT (Karush–Kuhn–Tucker) conditions. Let $w(s, s', a') \triangleq \frac{\text{Mix}_\beta(d, \rho)(s, s', a')}{\text{Mix}_\beta(d^E, \rho)(s, s', a')}$, then the constraint $\text{Mix}_\beta(d, \rho)(s, s', a') \geq 0$ holds if and only if $w(s, s', a') \geq 0 \quad \forall s, s', a'$.

$$\begin{aligned}
&\max_{w(s, s', a')} \max_{\lambda \geq 0} \mathbb{E}_{s, s', a' \sim \text{Mix}_\beta(d^E, \rho)(s, s', a')} [w(s, s', a') y(s, s', a')] - \mathbb{E}_{\text{Mix}_\beta(d^E, \rho)(s, s', a')} [f(w(s, s', a'))] \\
&\quad + \sum_{s, s', a'} \lambda (w(s, s', a') - 0)
\end{aligned} \tag{13}$$

Since strong duality holds, we can use the KKT constraints to find the solutions $w^*(s, s', a')$ and $\lambda^*(s, s', a')$.

- **Primal feasibility:** $w^*(s, s', a') \geq 0 \quad \forall s, s', a'$
- **Dual feasibility:** $\lambda^* \geq 0 \quad \forall s, s', a'$

- 495 • **Stationarity:** $\text{Mix}_\beta(d^E, \rho)(s, s', a')(-f'(w^*(s, s', a')) + y(s, s', a') + \lambda^*(s, s', a')) =$
 496 $0 \quad \forall s, s', a'$
- 497 • **Complementary Slackness:** $(w^*(s, s', a') - 0)\lambda^*(s, s', a') = 0 \quad \forall s, s', a'$

498 Using stationarity we have the following:

$$f'(w^*(s, s', a')) = y(s, s', a') + \lambda^*(s, s', a') \quad \forall s, s', a' \quad (14)$$

499 Now using complementary slackness, only two cases are possible $w^*(s, s', a') \geq 0$ or $\lambda^*(s, s', a') \geq$
 500 0 .

501 Combining both cases we arrive at the following solution for this constrained optimization:

$$w^*(s, s', a') = \max\left(0, f'^{-1}(y(s, s', a'))\right) \quad (15)$$

502 Using the optimal closed-form solution (w^*) for the inner optimization in Eq. (12) we obtain

$$\begin{aligned} & \min_{V(s, s')} \beta(1 - \gamma)\mathbb{E}_{d_0(s, s')}[V(s, s')] \\ & + \mathbb{E}_{s, s', a' \sim \text{Mix}_\beta(d^E, \rho)(s, s', a')} [\max(0, (f')^{-1}(y(s, s', a')) y(s, s', a') - \alpha f(\max(0, (f')^{-1}(y(s, s', a'))))] \\ & - (1 - \beta)\mathbb{E}_{s, a \sim \rho} \left[\gamma \sum_{s'} p(s'|s, a) V(s', s'') - V(s, s') \right] \end{aligned} \quad (16)$$

503 For deterministic dynamics, this reduces to the following simplified objective:

$$\begin{aligned} & \min_{V(s, s')} \beta(1 - \gamma)\mathbb{E}_{d_0(s, s')}[V(s, s')] \\ & + \mathbb{E}_{s, s', a' \sim \text{Mix}_\beta(d^E, \rho)(s, s', a')} [\max(0, (f')^{-1}(y(s, s', a')) y(s, s', a') - f(\max(0, (f')^{-1}(y(s, s', a'))))] \\ & - (1 - \beta)\mathbb{E}_{s, a \sim \rho} [\gamma V(s', s'') - V(s, s')] \end{aligned} \quad (17)$$

504 where $y(s, a, g) = \gamma V(s', s'') - V(s, s')$.

505 6.2.2 What does the utility function $V^*(s, s')$ represent?

506 Prior work [43] shows that for the regularized RL problem

$$\begin{aligned} & \max_{d \geq 0} \mathbb{E}_{d(s, a)}[r(s, a)] - \alpha D_f(d(s, a) || d^O(s, a)) \\ & \text{s.t. } \sum_{a \in \mathcal{A}} d(s, a) = (1 - \gamma)d_0(s) + \gamma \sum_{(s', a') \in \mathcal{S} \times \mathcal{A}} d(s', a') p(s|s', a'), \quad \forall s \in \mathcal{S}. \end{aligned} \quad (18)$$

507 the dual optimizes for a Lagrangian variable V that represents a regularized optimal value function.
 508 This insight directly extends to our work with reward function set to zero, our Lagrangian variable
 509 learns only the regularized visitation probabilities under optimal policy.

510 It is easy to see why this is the case using the previous derivation. Following the derivation from the
 511 previous section, note that we had rewritten the inner maximization w.r.t the visitation distribution
 512 d , thus effectively getting rid of manipulating visitation distributions in the final objective. Our
 513 derivation above uses the following substitution shown in Eq 15 that holds as part of the closed form
 514 solution w.r.t inner maximization:

$$\frac{\text{Mix}_\beta(d, \rho)(s, s', a')}{\text{Mix}_\beta(d^E, \rho)(s, s', a')} = \max\left(0, f'^{-1}(y(s, s', a'))\right) \quad (19)$$

515 where $y = \gamma V(s', s'') - V(s, s')$. For deterministic dynamics, at convergence, the following holds
 516 for all s, s', a' where $d^*(s, s', a') > 0$:

$$f'^{-1}(\gamma V^*(s', s'') - V^*(s, s')) = \frac{\text{Mix}_\beta(d^*, \rho)(s, s', a')}{\text{Mix}_\beta(d^E, \rho)(s, s', a')} \quad (20)$$

517 implying:

$$(\gamma V^*(s', s'') - V^*(s, s')) = f' \left(\frac{\text{Mix}_\beta(d^*, \rho)(s, s', a')}{\text{Mix}_\beta(d^E, \rho)(s, s', a')} \right) = -r_i(s, s', a') \quad (21)$$

518 The above relation makes the the interpretation of $V^*(s, s')$ clear. $(V^*(s, s') - \gamma V^*(s', s''))$ de-
 519 notes the implied reward function $r_i(s, s', a')$ under which V^* computes the maximum cumulative
 520 expected return, where a' is the action that leads to s'' . As shown above the the implied reward
 521 function $r_i(s, s', a') = -f' \left(\frac{\text{Mix}_\beta(d^*, \rho)(s, s', a')}{\text{Mix}_\beta(d^E, \rho)(s, s', a')} \right)$ is the divergence between expert stationary visita-
 522 tion distribution and agent's stationary visitation that is obtained after taking the action a' from s'
 523 and then acting optimally to match the expert visitation distribution. Note that the function f' is
 524 non-decreasing as the function f is convex from definition of f -divergences.

525 6.2.3 Analytical form of f_p^* for χ^2 divergence

526 For χ^2 divergence, the generator function $f(x) = (x - 1)^2$. $f'(x) = 2(x - 1)$ and correspondingly
 527 $f'^{-1}(x) = \frac{x}{2} + 1$. Substituting $f'^{-1}(x)$ in definition of f_p^* :

$$f_p^*(x) = \max(0, f'^{-1}(x))(x) - f(\max(0, f'^{-1}(x))) \quad (22)$$

528 Since x we substitute takes the form of residual $\text{residual} = \gamma \mathbb{E}_{s'' \sim p(\cdot | s', a')} [V(s', s'')] - V(s, s')$,
 529 the below pseudocode shows the implementation of f_p^* for DIL0.

```
530 1 def f_star_p(self, residual, type='chi_square'):
531 2     if type=='chi_square':
532 3         omega_star = torch.max(residual / 2 + 1, torch.zeros_like(
533         residual))
534 4         return residual * omega_star - (omega_star - 1)**2
```

535 6.2.4 Intuitive understanding of DIL0

536 To better understand this objective's behavior we consider the last two terms from Eq 5 in its expanded
 537 form below. We ignore the first term as it is simply pushing down Q -values at initial distribution of
 538 states, to prevent overestimation when learning from offline datasets.

$$\begin{aligned} \beta \mathbb{E}_{s, s', s'' \sim \tilde{d}^E} [f_p^*(\gamma V(s', s'') - V(s, s'))] + (1 - \beta) * \mathbb{E}_{s, s', s'' \sim \rho} [f_p^*(\gamma V(s', s'') - V(s, s'))] \\ - (1 - \beta) \mathbb{E}_{s, s', a' \sim \rho} [\gamma \mathbb{E}_{s'' \sim p(\cdot | s', a')} [V(s', s'')] - V(s, s')], \end{aligned} \quad (23)$$

539 Denote $r(s, s', a^E) = V(s, s') - \gamma V(s', s'')$ as the implicit expert reward of under a learned Q -
 540 function. The objective presents a clear intuition when we study the objective's behavior in different
 541 situations individually: (a) For samples from ρ , the objective pushes down the implicit reward to 0 as
 542 shown below:

$$\min_r \mathcal{L}(r) = \begin{cases} (1 - \beta) \frac{r^2}{4}, & \text{if } r < 2, \\ (1 - \beta)r & \text{otherwise.} \end{cases} \quad (24)$$

543 (b) For samples from the expert distribution \tilde{d}^E , the objective ensures that reward is greater than
 544 equal to 2

$$\min_r \mathcal{L}(r) = \begin{cases} \beta(\frac{r^2}{4} - r), & \text{if } r < 2, \\ 0 & \text{otherwise.} \end{cases} \quad (25)$$

545 It becomes clear now that DIL0 is implicitly learning a valid reward function that ensures higher
 546 discounted return for the expert compared to the suboptimal dataset by shaping Q -values directly.

547 6.3 Implementation

548 The algorithm for DIL0 can be found in Algorithm 1. We base the DIL0 implementation on the
 549 official implementation of pytorch-IQL <https://github.com/gwthomas/IQL-PyTorch/tree/main> that is
 550 based on IQL [44]. We keep the same network architecture as the original code and do not vary it
 551 across environments.

6.3.1 Imitation Learning with Proprioceptive Observations

Our experiment design is based on the benchmark from [13, 28] but we explain the setup here for completeness.

Environments: For the offline imitation learning experiments we focus on 9 locomotion and manipulation environments from the MuJoCo physics engine [18] comprising of Hopper, Walker2d, HalfCheetah, Ant, Kitchen, Pen, Door and Hammer to make a total of 24 datasets. The MuJoCo environments used in this work are [licensed under CC BY 4.0](#) and the datasets used from D4RL are also [licensed under Apache 2.0](#).

Suboptimal Datasets: We use the offline imitation learning benchmark from [28] that utilizes offline datasets consisting of environment interactions from the D4RL framework [19]. Specifically, suboptimal datasets are constructed following the composition protocol introduced in SMODICE [13]. The suboptimal datasets, denoted as 'random+expert', 'random+few-expert', 'medium+expert', and 'medium+few-expert' combine expert trajectories with low-quality trajectories obtained from the "random-v2" and "medium-v2" datasets, respectively. For locomotion tasks, the 'random/medium+expert' dataset contains a mixture of some number of expert trajectories (≤ 200) and ≈ 1 million transitions from the "x" dataset. The 'x+few-expert' dataset is similar to 'x+expert', but with only 30 expert trajectories included. For manipulation environments we consider only 30 expert trajectories mixed with the complete 'x' dataset of transitions obtained from D4RL.

Expert Observation Dataset: To enable imitation learning from observation, we use 1 expert observation trajectory obtained from the "expert-v2" dataset for each respective environment.

Baselines: To benchmark and analyze the performance of our proposed methods for offline imitation learning with suboptimal data, we consider different representative baselines in this work: BC [49], SMODICE [13], RCE [53], ORIL [48], IQLearn [50], ReCOIL [28]. SMODICE has been shown to be competitive [13] to DEMODICE [34] and hence we exclude it from comparison. SMODICE is an imitation learning method based on the dual framework, that optimizes an upper bound to the true imitation objective. ORIL adapts generative adversarial imitation learning (GAIL) [9] algorithm to the offline setting, employing an offline RL algorithm for policy optimization. The RCE baseline combines RCE, an online example-based RL method proposed by Eysenbach et al. [53]. RCE also uses a recursive discriminator to test the proximity of the policy visitations to successful examples. [53], with TD3-BC [54]. Both ORIL and RCE utilize a state-based discriminator similar to SMODICE, and TD3-BC serves as the offline RL algorithm. All the compared approaches only have access to the expert state-action trajectory.

The open-source implementations of the baselines SMODICE, RCE, and ORIL provided by the authors [13] are employed in our experiments. We use the hyperparameters provided by the authors, which are consistent with those used in the original SMODICE paper [13], for all the MuJoCo locomotion and manipulation environments.

In our set of environments, we keep the same hyper-parameters across tasks - locomotion, adroit manipulation, and kitchen manipulation. We train until convergence for all algorithms including baselines and we found the following timesteps to be sufficient for different set of environments: Kitchen: 1e6, Few-expert-locomotion: 500k, Locomotion: 300k, Manipulation: 500k

We keep a constant batch size of 1024 across all environments. For all tasks, we average mean returns over 10 evaluation trajectories and 7 random seeds. Full hyper-parameters we used for experiments are given in Table 2. For policy update, using Value Weighted Regression, we use the temperature τ to be 3 for all environments.

Hyperparameters for our proposed off-policy imitation learning method DIL0 are shown in Table 2.

6.3.2 LfO with Image Observations

We use robomimic [20] for our imitation with image observations experiments. The following two environments are used here (the description is taken from their paper and written here for conciseness):

Hyperparameter	Value
Policy learning rate	3e-4
Value learning rate	3e-4
f -divergence	χ^2
max-clip (Value clipping for policy learning)	100
MLP layers	(256,256)
β (mixture ratio)	0.5
η (orthogonal gradient descent)	0.5
τ (policy temperature)	3

Table 2: Hyperparameters for DIL0 in imitation from proprioceptive observations.

Lift: Object observations (10-dim) consist of the absolute cube position and cube quaternion (7-dim), and the cube position relative to the robot end effector (3-dim). The cube pose is randomized at the start of each episode with a random z-rotation in a small square region at the center of the table.

Can Object observations (14-dim) consist of the absolute can position and quaternion (7-dim), and the can position and quaternion relative to the robot end effector (7-dim). The can pose is randomized at the start of each episode with a random z-rotation anywhere inside the left bin.

Robomimic provides three datasets and two modalities of observation (Proprioceptive, Images) for both environments above. The datasets are denoted by - MH (Multi-human), MG (Machine Generated), PH(Proficient-human). We use the MG and MH datasets as the suboptimal datasets in our task and PH as the source of expert observations. MH and MG datasets consists of 200 trajectories of usually suboptimal nature and we use 50 observation-only trajectory from PH datasets. This tasks is complex by the fact that expert-level actions are mostly unseen in the suboptimal dataset and the agent needs to learn the best actions that matches expert visitation from the suboptimal dataset. We implement all algorithms in the Robomimic codebase without any change in network architecture, data-preprocessing or learning hyperparameters. We tune algorithm specific hyperparameters in a course grid for BCO, SMODICE, and DIL0 to compare the best performance of methods independent of hyperparameters. For BCO, we tune the inverse dynamics model learning epochs between [1,5,10]. For SMODICE, we tuned discriminator learning epochs between [1,5], and gradient penalty between [1,5,10,20]. To control overestimation due to learning with offline datasets in DIL0 we consider a linear weighting α between the optimism and pessimism terms in Eq 5 inspired by prior work [28] as follows:

$$\min_Q (1 - \lambda)\beta(1 - \gamma)\mathbb{E}_{\tilde{d}_0} [V(s, s')] + \lambda\mathbb{E}_{s, s', a' \sim \text{Mix}_{\beta}(\tilde{d}^E, \rho)} [f_p^*(\gamma\mathbb{E}_{s'' \sim p(\cdot|s', a')} [V(s', s'')] - V(s, s'))] - \lambda(1 - \beta)\mathbb{E}_{s, s', a' \sim \rho} [\gamma\mathbb{E}_{s'' \sim p(\cdot|s', a')} [V(s', s'')] - V(s, s')],$$

The hyperparameters used for DIL0 can be found in Table 3. For the architecture specific hyperparameters we refer the readers to [20].

Hyperparameter	Value
max-clip (Value clipping for policy learning)	100
λ (pessimism parameter)	0.7
β (mixture ratio)	0.5
η (orthogonal gradient descent)	0.5
τ (policy temperature)	3

Table 3: Hyperparameters for DIL0 in imitation from image observations.

6.4 Robot Manipulation Experiments

Our setup for manipulation experiments is inspired by the robot air hockey environment [55] for applying DIL0 to physical robotics settings. Our setup utilizes a Universal Robotics 5 kilogram e-series (UR5e) 6-degree of freedom robotic arm on a fixed mount, a Robotiq parallel jaw gripper, a 1.93m \times 0.76m Wind Chill air hockey table which is tilted at a 5.5 degree angle, and an overhead

628 Sony Playstation Eye, a high framerate camera, which gathers 640×480 frames at 60 FPS, mounted
 629 to the ceiling to have a full view of the table. The paddle that is held by the robot end effector is
 630 9.5cm in diameter and the puck is 6.3cm.

631 In this setup, the negative x
 632 direction is oriented along
 633 the table, and the y is
 634 across the table. The ac-
 635 tion space for the arm uti-
 636 lizes pose control in x, y
 637 through an inverse kine-
 638 matics controller accessi-
 639 ble through the Universal

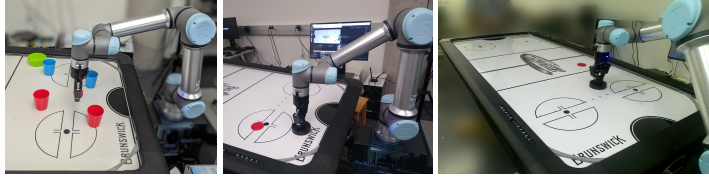


Figure 5: Tasks: Left: Place object and avoid obstacles. Center: Stationary Puck Striking. Right: Dynamic Puck Hitting

640 robotics real-time data exchange interface. Utilizing the serving command, the arm is
 641 controlled using delta positions clipped between 26cm in the x direction and 13cm in
 642 the right direction. These control limits are specified to prevent the robot from trigger-
 643 ing force limits, which results in an emergency stop. Actions are taken at a 20Hz fre-
 644 quency to allow for rapid response to dynamic elements, such as hitting a falling puck.

645 The position and velocity of the end effector
 646 can be recovered through the real-time data ex-
 647 change, but other objects like the puck or the
 648 hand require identification. This work utilizes
 649 an overhead camera running at 60Hz to locate
 650 these objects using a vision pipeline that relies
 651 on hue saturation value segmentation followed
 652 by object identification. This gives an x_c, y_c co-
 653 ordinate in camera space, which we convert via
 654 OpenCV [56] homography to robot coordinates.
 655 This homography is computed by mapping the
 656 end effector positions given by the robot sensors
 657 to visual locations from the camera.

658 We apply imitation learning from observations
 659 on several tasks built on top of the above robot
 660 setup. The following experiments are described
 661 here:

662 **Safe object manipulation:** This task involves
 663 moving a strawberry to a bowl while avoiding
 664 four cups placed in the workspace of the robot.
 665 The bowl is placed in the top right corner of the
 666 workspace, and the cups are placed in fixed loca-
 667 tions. The success metric is the robot stopping
 668 above the bowl while making no contact with any of the cups. The test set involves 10 random
 669 starting locations for the end effector that are fixed between assessments. The observation space is the
 670 2D end effector position, and the strawberry is initialized inside the gripper. Our suboptimal dataset
 671 consists of 50 trajectories of 100 time steps on average where the robot is initialized in a random
 672 location, and the human moves the arm to a random different location, ignoring the positions of the
 673 cups or the bowl. In this setting, we investigated the following expert data, visualized in Figure 5:

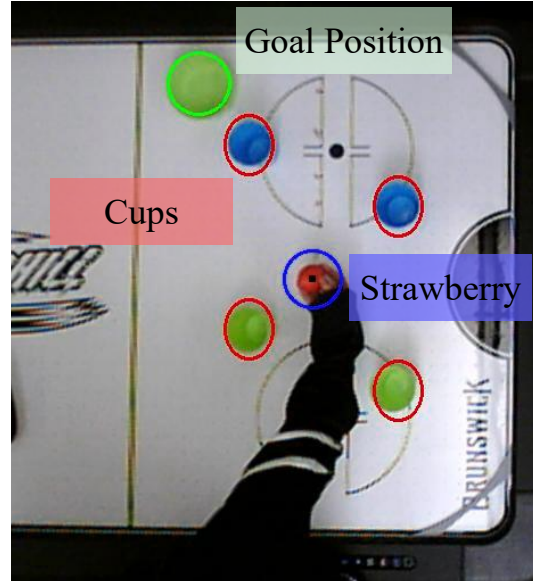


Figure 6: **Cross Embodiment Demonstration:** Tracking of the strawberry and obstacles for learning action-free.

- **Few Trajectories:** The expert data is drawn from a set where the expert is initialized in a random location, sometimes touching an obstacle, and must use the teleoperation system to avoid the obstacle and reach the goal. In this setting we used 15 expert trajectories.

- **Fixed start:** The expert is initialized at the opposite corner of the workspace, and navigates to the goal location following different paths using teleoperation. In this setting we used 15 trajectories.
- **Few Uniform:** Uses the same expert data as the Few Trajectories setting, but the dataset consists of randomly samples 300 transitions, where one trajectory is approximately 60 transitions of data.
- **Cross Embodiment Few/Fixed/Uniform:** The expert is a person holding the strawberry in his/her hand, visualized in Figure 6. They then move the strawberry tracked by the camera to the goal location while avoiding the obstacles, starting from random/fixed locations with 15 expert trajectories respectively or uniform with 300 transitions.

Stationary Striking: This task involves moving the end effector to strike a stationary puck. The success metric is the robot touching the puck. The test set involves 10 initializations of the puck position across the length of the table. To ensure uniformity across evaluations, the set of initialization locations of the puck are fixed across methods. The end effector is initialized at 0.38m from the base in the center of the table, so a success strike does not require backward motion. The observation space is the 2D end effector position and the tracked position of the puck. Our suboptimal dataset consists of 50 trajectories of 75 time steps on average where the robot is initialized at the start position, and the human moves the arm in a random, vaguely striking pattern.

In this setting we used an expert dataset of 400 trajectories where the expert uses mouse teleoperation to strike the puck. The expert efficiently strikes the puck in a single motion. We visualize the expert striking and the puck position in Figure 5. We show the learned action vectors for all algorithms and tasks fixed start (Figure 9), Few Uniform (Figure 8) and Few trajectories (Figure 10).

Dynamic Hitting: This task involves hitting a puck dropped from the top of the table. Because the table is set at an angle, this will cause the puck to fall with increasing acceleration towards the opposite side. The setup is visualized in Figure 5. The test set involves 10 initializations of the puck position dropped from positions across the length of the top of the table. The locations of the 10 puck drops are fixed using indicators across methods to give fair evaluation, and the arm is initialized in the center of the table, 0.68m from the base. The observation space is the 2D end effector position and 2D end effector velocity and the history of the last 5 tracked positions of the puck relative to the position of the end effector. Our suboptimal dataset consists of 50 trajectories of 200 time steps on average where the robot is initialized at the start position, and the human moves the arm around the puck without striking it.

We utilize two success metrics for this task: 1) touching: a trajectory is considered successful if the agent touches the puck. 2) hitting: the puck must have velocity in the opposite direction that it was dropped. This task is especially challenging for existing methods because of the long sequence of actions necessary to position the paddle properly, and the high level of both precision and timing: even a few millimeters of error or a movement at the wrong time will result in a failure, especially for hitting. Previous work has observed that this task is challenging even for humans, who often require several tries of practice, and many dataset trajectories consist of many inaccurate hits. In this domain, Behavior cloning only achieves 30% success at touching the puck, and Implicit Q-learning, a popular offline RL method, can only achieve 60% success, even though it employs a hand-designed reward function.

We used the implementation details from the proprioception task with the difference that in all the real-robot tasks we tune the following parameters across different methods: For BCO, we tune the inverse dynamics model learning epochs between [1,5,10]. For SMODICE, we tuned discriminator learning epochs between [1,5], and gradient penalty between [1,5,10,20]. For DIL0 we tune the conservatism parameter from previous section between [0.5,0.6,0.7,0.8].

In this domain, these challenges appear to be empirically validated in the performance of the baseline methods. We hypothesize that the accumulation of error over long horizons in other learning from observation methods results in poor performance, as visualized in Table 3 and Figure 4. For methods

727 like BCO, learning the necessary inverse dynamics to interpolate the long sequence of actions for a
 728 successful strike is impractical, resulting in behavior that appears listless. On the other hand, while
 729 the SMODICE discriminator rewards are able to occasionally match the visitation distribution of
 730 the expert, there is an exponential explosion of possible combinations of puck history and paddle
 731 positions, resulting in poor generalization: on the left half of the table, SMODICE is unable to hit the
 732 puck.

733 The hitting scenario involves two settings, both using mouse teleoperation to control the puck: one
 734 where the human strikes the puck only once and then the trajectory ends, which utilizes a dataset
 735 of 50 trajectories, and an expert dataset of 850 trajectories where the human keeps hitting the puck
 736 repeatedly for up to 2000 timesteps. The task is challenging for human, so trajectories average only
 737 500 timesteps and cleaned so that human mistakes are removed from the expert dataset. We visualize
 738 the expert hitting and the puck position in Figure 7.

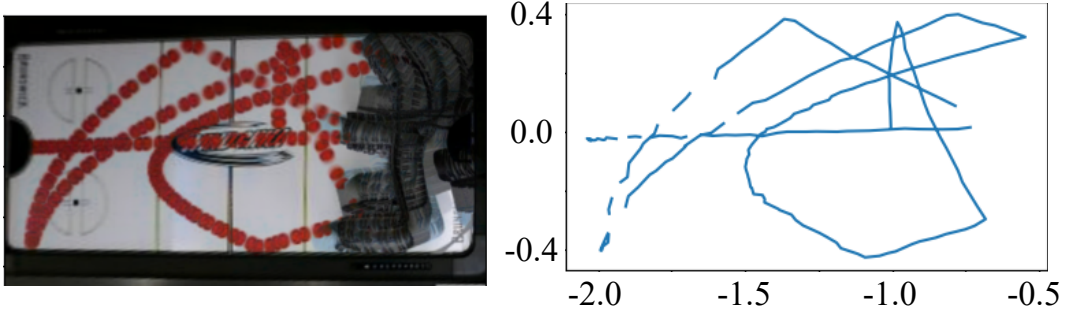


Figure 7: **Expert Hitting:** Visualization of one trajectory of puck tracking and hitting by the expert. **Right:** stacked frames of the environment. **Left:** puck position in robot coordinates

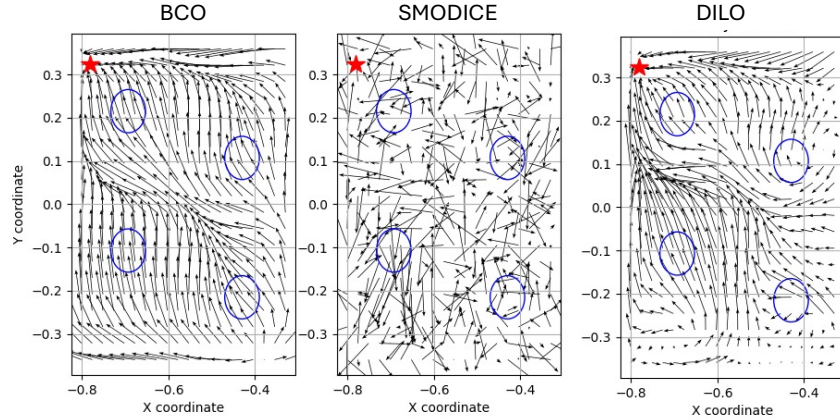


Figure 8: Action Vectors qualitatively showing the next x-y action for the safe manipulation with uniform sampled transitions. BCO generalizes incorrectly at a number of locations producing policies that hit obstacles. DILO learns to mimic expert’s intent better demonstrating signs that it has learned to avoid obstacles by the arrows around

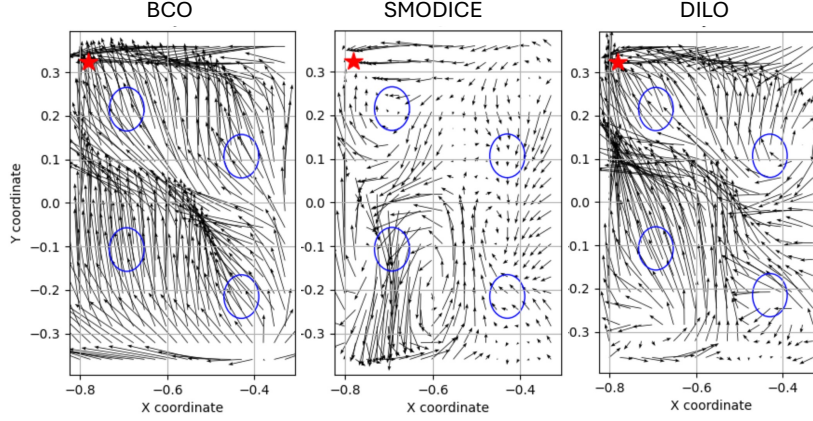


Figure 9: Action Vectors qualitatively showing the next x-y action for the safe manipulation with fixed start state. BCO generalizes incorrectly at a number of locations producing policies that hit obstacles. DILO learns to mimic expert’s intent better demonstrating signs that it has learned to avoid obstacles by the arrows around

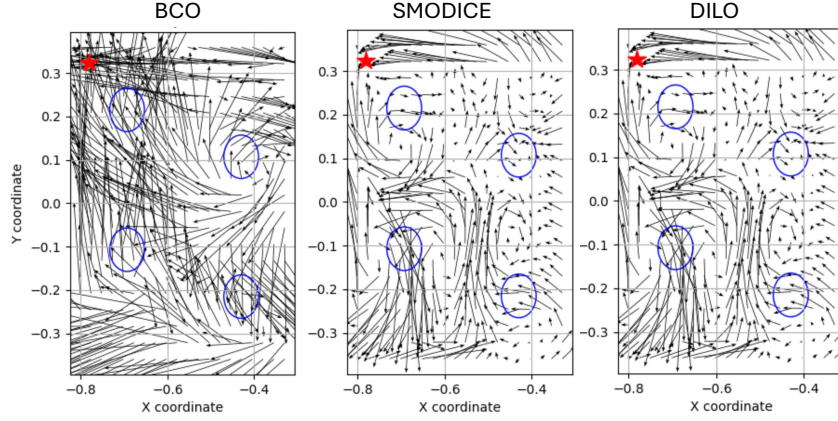


Figure 10: Action Vectors qualitatively showing the next x-y action for the safe manipulation with few trajectories. BCO generalizes incorrectly at a number of locations producing policies that hit obstacles. DILO learns to mimic expert’s intent better demonstrating signs that it has learned to avoid obstacles by the arrows around

739 6.5 Limitations

740 Learning from Observation is a challenging setting, and while DILO makes some key assumptions
 741 in order to achieve good performance. First, matching distributions becomes exponentially more
 742 challenging in the dimensionality of the state space. In this work, while DILO outperforms baselines
 743 in the Expert Image observations, it still shows limited performance. Second, while learning from
 744 observations opens the door for good performance without expert actions, the expert observation
 745 space must match that of the agent. In some video settings, this is not the case ex. the agent might use
 746 a fixed camera when the human is egocentric, or vice versa. Finally, DILO utilizes the conservatism
 747 parameter τ to regulate the degree of extrapolation from the algorithm. In some settings, the values
 748 can diverge, resulting in V^* taking on values that might be too large to be used for learning the
 749 downstream policy. Adaptively selecting τ to maximize extrapolation while avoiding divergence is
 750 an area of action investigation.

751 6.5.1 Failure Cases

752 While DILO outperforms other methods in overall success rate, the failure modes can differ. In
 753 general, DILO tends to be conservative in what actions it takes, learning motions that might be slower
 754 than BCO, or may get stuck before arriving at the goal. In low observation settings, DILO can also

755 exhibit “dead zone” behavior, where the model becomes mostly unresponsive. Below we detail some
756 of the exact error modes in particular tasks:

757 **Safe Manipulation:** While DIL0 and BCO have comparable success rates, the two algorithms fail
758 in different ways. BCO tends to take large actions while ignoring obstacles to reach the goal, while
759 DIL0 takes more conservative actions. Thus, while BCO might fail by knocking over a cup, DIL0
760 will tend to fail to reach the goal. Because this is a low data setting, both algorithms BCO and DIL0
761 can end up coming close to the cups or brushing them gently. As a sidenote, SMODICE fails at even
762 reaching the goal in most cases in this task, possibly because of this low data setting.

763 **Striking:** This domain is challenging because of the narrow data regime, and all methods tend to
764 struggle in similar ways. The most common consequence of low data arises through sensitivity to
765 the x location of the puck (along the table). While intuitively, striking behavior should be relatively
766 invariant for a fixed y (horizontal position on the table), slight variation in x from the dataset can
767 result in a policy that moves the arm in the opposite direction of the puck, probably due to errors
768 in extrapolation. In addition, striking is a dynamic behavior that requires a precise combination of
769 forward and horizontal actions. Even a slight error in the ratio can result in a near miss. Finally,
770 DIL0 tends to learn more conservative policies, and in some locations may not not strike the puck
771 with much force. However, because of the low data coverage, this issue is endemic to all the learned
772 policies.

773 **Hitting:** The primary challenge of achieving a hit in this task is the precise alignment of the paddle
774 to the puck. While DIL0 performs well, it is not perfectly accurate, resulting in touches that bounce
775 off the side of the paddle. This challenge is endemic to all policies. Additionally the conservatism
776 of DIL0 actions appear when it moves under the puck, where it tends to move slowly, and dropping
777 the puck too quickly can result in DIL0 failing to reach the puck. As a result, while DIL0 is likely to
778 succeed at the first hit, it can struggle to generate multiple hits because this can require rapid side to
779 side movement. These issues are largely endemic to all the learned policies, where SMODICE tends
780 to be even less precise and BCO struggle to learn to strike, though it can occasionally position under
781 the puck.

782 Visualizations of the failure modes can be seen in the accompanying video attachment.