BENCHMARKS FOR DEEP OFF-POLICY EVALUATION

Justin Fu[*1] **Mohammad Norouzi**[*2] **Ofir Nachum**[*2] **George Tucker**[*2]
Ziyu Wang[2] **Alexander Novikov**[3] **Mengjiao Yang**[2] **Michael R. Zhang**[2]
Yutian Chen[3] **Aviral Kumar**[1] **Cosmin Paduraru**[3] **Sergey Levine**[1] **Tom Le Paine**[*3]

[1]UC Berkeley [2]Google Brain [3]DeepMind
justinfu@berkeley.edu,{mnorouzi,ofirnachum,gjt,tpaine}@google.com

ABSTRACT

Off-policy evaluation (OPE) holds the promise of being able to leverage large, offline datasets for both evaluating and selecting complex policies for decision making. The ability to learn offline is particularly important in many real-world domains, such as in healthcare, recommender systems, or robotics, where online data collection is an expensive and potentially dangerous process. Being able to accurately evaluate and select high-performing policies without requiring online interaction could yield significant benefits in safety, time, and cost for these applications. While many OPE methods have been proposed in recent years, comparing results between papers is difficult because currently there is a lack of a comprehensive and unified benchmark, and measuring algorithmic progress has been challenging due to the lack of difficult evaluation tasks. In order to address this gap, we present a collection of policies that in conjunction with existing offline datasets can be used for benchmarking off-policy evaluation. Our tasks include a range of challenging high-dimensional continuous control problems, with wide selections of datasets and policies for performing policy selection. The goal of our benchmark is to provide a standardized measure of progress that is motivated from a set of principles designed to challenge and test the limits of existing OPE methods. We perform an evaluation of state-of-the-art algorithms and provide open-source access to our data and code to foster future research in this area[†].

1 INTRODUCTION

Reinforcement learning algorithms can acquire effective policies for a wide range of problems through active online interaction, such as in robotics (Kober et al., 2013), board games and video games (Tesauro, 1995; Mnih et al., 2013; Vinyals et al., 2019), and recommender systems (Aggarwal et al., 2016). However, this sort of active online interaction is often impractical for real-world problems, where active data collection can be costly (Li et al., 2010), dangerous (Hauskrecht & Fraser, 2000; Kendall et al., 2019), or time consuming (Gu et al., 2017). Batch (or offline) reinforcement learning, has been studied extensively in domains such as healthcare (Thapa et al., 2005; Raghu et al., 2018), recommender systems (Dudík et al., 2014; Theocharous et al., 2015; Swaminathan et al., 2017), education (Mandel et al., 2014), and robotics (Kalashnikov et al., 2018). A major challenge with such methods is the off-policy evaluation (OPE) problem, where one must evaluate the expected performance of policies solely from offline data. This is critical for several reasons, including providing high-confidence guarantees prior to deployment (Thomas et al., 2015), and performing policy improvement and model selection (Bottou et al., 2013; Doroudi et al., 2017).

The goal of this paper is to provide a standardized benchmark for evaluating OPE methods. Although considerable theoretical (Thomas & Brunskill, 2016; Swaminathan & Joachims, 2015; Jiang & Li, 2015; Wang et al., 2017; Yang et al., 2020) and practical progress (Gilotte et al., 2018; Nie et al., 2019; Kalashnikov et al., 2018) on OPE algorithms has been made in a range of different domains, there are few broadly accepted evaluation tasks that combine complex, high-dimensional problems

[*] Equally major contributors.

[†] Policies and evaluation code are available at https://github.com/google-research/deep_ope. See Section 5 for links to modelling code.

commonly explored by modern deep reinforcement learning algorithms (Bellemare et al., 2013; Brockman et al., 2016) with standardized evaluation protocols and metrics. Our goal is to provide a set of tasks with a range of difficulty, excercise a variety of design properties, and provide policies with different behavioral patterns in order to establish a standardized framework for comparing OPE algorithms. We put particular emphasis on large datasets, long-horizon tasks, and task complexity to facilitate the development of scalable algorithms that can solve high-dimensional problems.

Our primary contribution is the **D**eep **O**ff-**P**olicy **E**valuation (DOPE) benchmark. DOPE is designed to measure the performance of OPE methods by **1)** evaluating on challenging control tasks with properties known to be difficult for OPE methods, but which occur in real-world scenarios, **2)** evaluating across a range of policies with different values, to directly measure performance on policy evaluation, ranking and selection, and **3)** evaluating in ideal and adversarial settings in terms of dataset coverage and support. These factors are independent of task difficulty, but are known to have a large impact on OPE performance. To achieve 1, we selected tasks on a set of design principles outlined in Section 3.1. To achieve 2, for each task we include 10 to 96 policies for evaluation and devise an evaluation protocol that measures policy evaluation, ranking, and selection as outlined in Section 3.2. To achieve 3, we provide two domains with differing dataset coverage and support properties described in Section 4. Finally, to enable an easy-to-use research platform, we provide the datasets, target policies, evaluation API, as well as the recorded results of state-of-the-art algorithms (presented in Section 5) as open-source.

2 Background

We briefly review the off-policy evaluation (OPE) problem setting. We consider Markov decision processes (MDPs), defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho_0, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, transition distribution $\mathcal{T}(s'|s,a)$, initial state distribution $\rho_0(s)$, reward function $R(s,a)$ and discount factor $\gamma \in (0,1]$. In reinforcement learning, we are typically concerned with optimizing or estimating the performance of a policy $\pi(a|s)$.

The performance of a policy is commonly measured by the *policy value* V^π, defined as the expected sum of discounted rewards:

$$V^\pi := \mathbb{E}_{s_0 \sim \rho_0, s_{1:\infty}, a_{0:\infty} \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]. \quad (1)$$

If we have access to state and action samples collected from a policy π, then we can use the sample mean of observed returns to estimate the value function above. However, in off-policy evaluation we are typically interested in estimating the value of a policy when the data is collected from a separate *behavior policy* $\pi_B(a|s)$. This setting can arise, for example, when data is being generated online from another process, or in the purely offline case when we have a historical dataset.

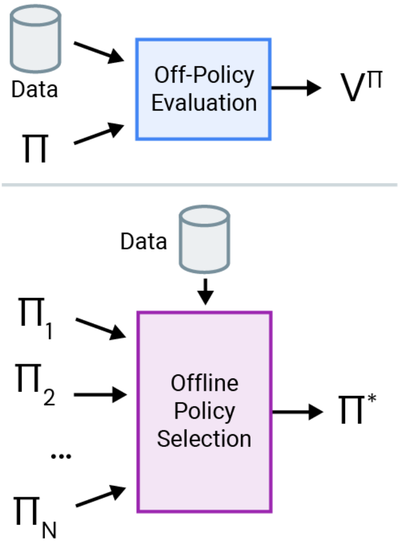


Figure 1: In Off-Policy Evaluation (top) the goal is to estimate the value of a single policy given only data. Offline Policy Selection (bottom) is a closely related problem: given a set of N policies, attempt to pick the best given only data.

In this work we consider the latter, purely offline setting. The typical setup for this problem formulation is that we are provided with a discount γ, a dataset of trajectories collected from a behavior policy $\mathcal{D} = \{(s_0, a_0, r_0, s_1, \ldots)\}$, and optionally the action probabilities for the behavior policy $\pi_B(a_t|s_t)$. In many practical applications, logging action propensities is not possible, for example, when the behavior policy is a mix of ML and hard-coded business logic. For this reason, we focus on the setting without propensities to encourage future work on behavior-agnostic OPE methods. For the methods that require propensities, we estimate the propensities with behavior cloning.

The objective can take multiple flavors, as shown in Fig. 1. A common task in OPE is to estimate the performance, or value, of a policy π (which may not be the same as π_B) so that the estimated value is as close as possible to V^π under a metric such as MSE or absolute error. A second task is to perform policy selection, where the goal is to select the best policy or set of policies out of a group of candidates. This setup corresponds to how OPE is commonly used in practice, which is to find the best performing strategy out of a pool when online evaluation is too expensive to be feasible.

3 DOPE: Deep Off-Policy Evaluation

The goal of the Deep Off-Policy Evaluation (DOPE) benchmark is to provide tasks that are challenging and effective measures of progress for OPE methods, yet is easy to use in order to better facilitate research. Therefore, we design our benchmark around a set of properties which are known to be difficult for existing OPE methods in order to gauge their shortcomings, and keep all tasks amenable to simulation in order for the benchmark to be accessible and easy to evaluate.

3.1 Task Properties

We describe our motivating properties for selecting tasks for the benchmark as follows:

High Dimensional Spaces (H) High-dimensionality is a key-feature in many real-world domains where it is difficult to perform feature engineering, such as in robotics, autonomous driving, and more. In these problems, it becomes challenging to accurately estimate quantities such as the value function without the use of high-capacity models such a neural networks and large datasets with wide state coverage. Our benchmark contains complex continuous-space tasks which exercise these challenges.

Long Time-Horizon (L) Long time horizon tasks are known to present difficult challenges for OPE algorithms. Some algorithms have difficulty doing credit assignment for these tasks. This can be made worse as the state dimension or action dimension increases.

Sparse Rewards (R) Sparse reward tasks increase the difficulty of credit assignment and add exploration challenges, which may interact with data coverage in the offline setting. We include a range robotics and navigation tasks which are difficult to solve due to reward sparsity.

Temporally extended control (T) The ability to make decisions hierarchically is major challenge in many reinforcement learning applications. We include two navigation tasks which require high-level planning in addition to low-level control in order to simulate the difficulty in such problems.

3.2 Evaluation Protocol

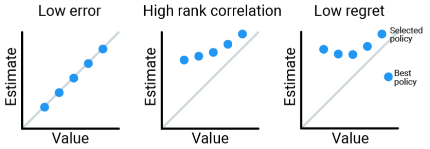


Figure 2: Error is a natural measure for off-policy evaluation. However for policy selection, it is sufficient to (i) rank the policies as measured by rank correlation, or (ii) select a policy with the lowest regret.

The goal of DOPE to provide metrics for policy ranking, evaluation and selection. Many existing OPE methods have only been evaluated on point estimates of value such as MSE, but policy selection is an important, practical use-case of OPE. In order to explicitly measure the quality of using OPE for policy selection, we provide a set of policies with varying value, and devise two metrics that measure how well OPE methods can rank policies.

For each task we include a dataset of logged experiences $\mathcal{D}$, and a set of policies $\{\pi_1, \pi_2, ..., \pi_N\}$ with varying values. For each policy, OPE algorithms must use $\mathcal{D}$ to produce an estimate of the policy's value. For evaluation of these estimates, we provide "ground truth values" $\{V^{\pi_1}, V^{\pi_2}, ..., V^{\pi_N}\}$ that are computed by running the policy for $M \geq 1000$ episodes, where the exact value of M is given by the number of episodes needed to lower the error bar on the ground truth values to 0.666. The estimated values are then compared to these ground truth values using three different metrics encompassing both policy evaluation and selection (illustrated in Figure 2; see Appendix A.1 for mathematical definitions).

Absolute Error This metric measures estimate accuracy instead of its usefulness for ranking. Error is the most commonly used metric to assess performance of OPE algorithms. We opted to use absolute error instead of MSE to be robust to outliers.

Regret@k This metric measures how much worse the best policies identified by the estimates are than the best policy in the entire set. It is computed by identifying the top-k policies according to the estimated returns. Regret@k is the difference between the actual expected return of the best policy in the entire set, and the actual value of the best policy in the top-k set.

Rank correlation This metric directly measures how well estimated values rank policies, by computing the correlation between ordinal rankings according by the OPE estimates and ordinal rankings according to the ground truth values.

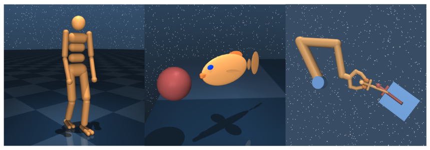
4 Domains

DOPE contains two domains designed to provide a more comprehensive picture of how well OPE methods perform in different settings. These two domains are constructed using two benchmarks previously proposed for offline reinforcement learning: RL Unplugged (Gulcehre et al., 2020) and D4RL (Fu et al., 2020), and reflect the challenges found within them.

The **DOPE RL Unplugged** domain is constrained in two important ways: 1) the data is always generated using online RL training, ensuring there is adequate coverage of the state-action space, and 2) the policies are generated by applying offline RL algorithms to the same dataset we use for evaluation, ensuring that the behavior policy and evaluation policies induce similar state-action distributions. Using it, we hope to understand how OPE methods work as task complexity increases from simple Cartpole tasks to controlling a Humanoid body while controlling for ideal data.

On the other hand, the **DOPE D4RL** domain has: 1) data from various sources (including random exploration, human teleoperation, and RL-trained policies with limited exploration), which results in varying levels of coverage of the state-action space, and 2) policies that are generated using online RL algorithms, making it less likely that the behavior and evaluation policies share similar induced state-action distributions. Both of these result in distribution shift which is known to be challenging for OPE methods, even in simple tasks. So, using it we hope to measure how well OPE methods work in more practical data settings.

4.1 DOPE RL Unplugged



DeepMind Control Suite (Tassa et al., 2018) is a set of control tasks implemented in MuJoCo (Todorov et al., 2012). We consider the subset included in RL Unplugged. This subset includes tasks that cover a range of difficulties. From Cartpole swingup, a simple task with a single degree of freedom, to Humanoid run which involves control of a complex bodies with 21 degrees of freedom. All tasks use the default feature representation of the system state, including proprioceptive information such as joint positions and velocity, and additional sensor information and target position where appropriate. The observation dimension ranges from 5 to 67.

Datasets and policies We train four offline RL algorithms (D4PG (Barth-Maron et al., 2018), ABM (Siegel et al., 2020), CRR (Wang et al., 2020) and behavior cloning), varying their hyperparameters. For each algorithm-task-hyperparameter combination, we train an agent with 3 random seeds on the DM Control Suite dataset from RL Unplugged and record policy snapshots at exponentially increasing intervals (after 25k learner steps, 50k, 100K, 200K, etc). Following Gulcehre et al. (2020), we consider a deterministic policy for D4PG and stochastic policies for BC, ABM and CRR. The datasets are taken from the RL Unplugged benchmark, where they were created by training multiple (online) RL agents and collecting both successful and unsuccessful episodes throughout training. All offline RL algorithms are implemented using the Acme framework (Hoffman et al., 2020).

4.2 DOPE D4RL

Gym-MuJoCo tasks. Gym-MuJoCo consists of several continuous control tasks implemented within the MuJoCo simulator (Todorov et al., 2012) and provided in the OpenAI Gym (Brockman et al., 2016) benchmark for online RL. We include the HalfCheetah, Hopper, Walker2D, and Ant tasks. We include this domain primarily for comparison with past works, as a vast array of popular RL

Statistics	cartpole swingup	cheetah run	finger turn hard	fish swim	humanoid run	walker stand	walker walk	manipulator insert ball	manipulator insert peg
Dataset size	40K	300K	500K	200K	3M	200K	200K	1.5M	1.5M
State dim.	5	17	12	24	67	24	24	44	44
Action dim.	1	6	2	5	21	6	6	5	5
Properties	-	H, L	H, L	H, L	H, L	H, L	H, L	H, L, T	H, L,T

Statistics	maze2d	antmaze	halfcheetah	hopper	walker	ant	hammer	door	relocate	pen
Dataset size	1/2/4M	1M	1M	1M	1M	1M	11K/1M	7K/1M	10K/1M	5K/500K
# datasets	1	1	5	5	5	5	3	3	3	3
State dim.	4	29	17	11	17	111	46	39	39	45
Action dim.	2	8	6	3	6	8	26	28	30	24
Properties	T	T, R	H	H	H	H	H, R	H, R	H, R	H, R

Table 1: Task statistics for RLUnplugged tasks (top) and D4RL tasks (bottom). Dataset size is the number of (s, a, r, s') tuples. For each dataset, we note the properties it possesses: high dimensional spaces (**H**), long time-horizon (**L**), sparse rewards (**R**), temporally extended control (**T**).

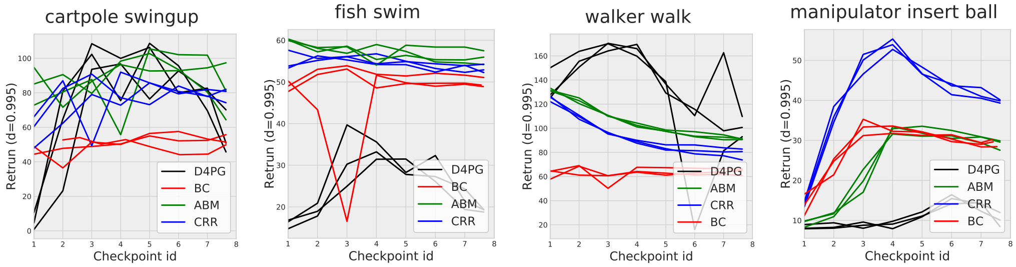
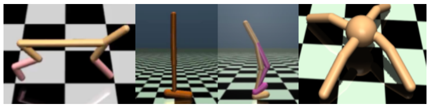
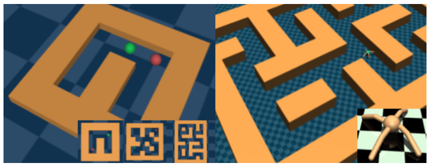


Figure 3: Online evaluation of policy checkpoints for 4 Offline RL algorithms with 3 random seeds. We observe a large degree of variability between the behavior of algorithms on different tasks. Without online evaluation, tuning the hyperparameters (e.g., choice of Offline RL algorithm and policy checkpoint) is challenging. This highlights the practical importance of Offline policy selection when online evaluation is not feasible. See Figure A.7 for additional tasks.

methods have been evaluated and developed on these tasks (Schulman et al., 2015; Lillicrap et al., 2015; Schulman et al., 2017; Fujimoto et al., 2018; Haarnoja et al., 2018).

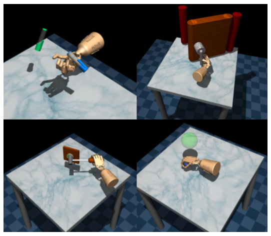


Gym-MuJoCo datasets and policies. For each task, in order to explore the effect of varying distributions, we include 5 datasets originally proposed by Fu et al. (2020). 3 correspond to different performance levels of the agent – "random", "medium", and "expert". We additionally include a mixture of medium and expert dataset, labeled "medium-expert", and data collected from a replay buffer until the policy reaches the medium level of performance, labeled "medium-replay". For policies, we selected 11 policies collected from evenly-spaced snapshots of training a Soft Actor-Critic agent (Haarnoja et al., 2018), which covers a range of performance between random and expert.



Maze2D and AntMaze tasks. Maze2D and AntMaze are two maze navigation tasks originally proposed in D4RL (Fu et al., 2020). The domain consists of 3 mazes ranging from easy to hard ("umaze", "medium", "large"), and two morphologies: a 2D ball in Maze2D and the "Ant" robot of the Gym benchmark in AntMaze. For Maze2D, we provide a less challenging reward computed base on distance to a fixed goal. For the AntMaze environment reward is given only upon reaching the fixed goal.

Maze2D and AntMaze datasets and policies. Datasets for both morphologies consists of undirect data navigating randomly to different goal locations. The datasets for Maze2D are collected by using a high-level planner to command waypoints to a low-level PID controller in order to reach randomly selected goals. The dataset in AntMaze is generated using the same high-level planner, but the low-level planner is replaced with a goal-conditioned policy trained to reach arbitrary waypoints. Both of these datasets are generated from non-Markovian policies, as the high-level controller maintains a history of waypoints reached in order to construct a plan to the goal. We provide policies for all environments except "antmaze-large" by taking training snapshots obtained while running the DAPG algorithm (Rajeswaran et al., 2017). Because obtaining high-performing policies for "antmaze-large" was challenging, we instead used imitation learning on a large amount of expert data to generate evaluation policies. This expert data is obtained by collecting additional trajectories that reach the goal using a high-level waypoint planner in conjunction with a low-level goal-conditioned policy (this is the same method as was used to generate the dataset, Sec. 5 (Fu et al., 2020)).



Adroit tasks. The Adroit domain is a realistic simulation based on the Shadow Hand robot, first proposed by Rajeswaran et al. (2017). There are 4 tasks in this domain: opening a door ("door"), pen twirling ("pen"), moving a ball to a target location ("relocate"), and hitting a nail with a hammer ("hammer"). These tasks all contain sparse rewards and are difficult to learn without demonstrations.

Adroit datasets and policies. We include 3 datasets for each task. The "human" dataset consists of a small amount of human demonstrations performing the task. The "expert" dataset consists of data collected from an expert trained via DAPG (Rajeswaran et al., 2017). Finally, the "cloned" dataset contains a mixture of human demonstrations and data collected from an imitation learning algorithm trained on the demonstrations. For policies, we include 11 policies collected from snapshots while running the DAPG algorithm, which range from random performance to expert performance.

5 Baselines and Results

The goal of our evaluation is two-fold. First, we wish to measure the performance of a variety of existing algorithms to provide baselines and reference numbers for future research. Second, we wish to identify shortcomings in these approaches to reveal promising directions for future research.

5.1 Baselines

We selected six methods to evaluate, which cover a variety of approaches that have been explored for the OPE problem.

Fitted Q-Evaluation (FQE) As in Le et al. (2019), we train a neural network to estimate the value of the evaluation policy π by bootstrapping from $Q(s', \pi(s'))$. We tried two different implementations, one from Kostrikov & Nachum (2020)[3] and another from Paine et al. (2020) labeled FQE-L2 and FQE-D respectively to reflect different choices in loss function and parameterization.

Model-Based (MB) Similar to Paduraru (2007), we train dynamics and reward models on transitions from the offline dataset $\mathcal{D}$. Our models are deep neural networks trained to maximize the log likelihood of the next state and reward given the current state and action, similar to models from successful model-based RL algorithms (Chua et al., 2018; Janner et al., 2019). We follow the setup detailed in Zhang et al. (2021). We include both the feed-forward and auto-regressive models labeled MB-FF and MB-AR respectively. To evaluate a policy, we compute the return using simulated trajectories generated by the policy under the learned dynamics model.

Importance Sampling (IS) We perform importance sampling with a learned behavior policy. We use the implementation from Kostrikov & Nachum (2020)[3], which uses self-normalized (also known as weighted) step-wise importance sampling (Precup, 2000). Since the behavior policy is not known explicitly, we learn an estimate of it via a max-likelihood objective over the dataset $\mathcal{D}$, as advocated by Xie et al. (2018); Hanna et al. (2019). In order to be able to compute log-probabilities when the target policy is deterministic, we add artificial Gaussian noise with standard deviation 0.01 for all deterministic target policies.

[3] Code available at `https://github.com/google-research/google-research/tree/master/policy_eval`.

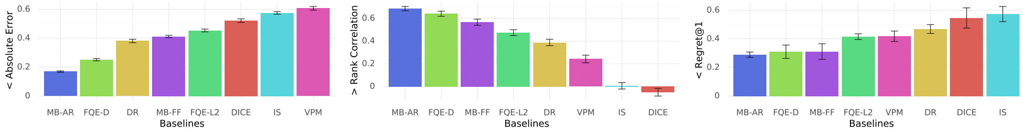


Figure 4: **DOPE RL Unplugged** Mean overall performance of baselines.

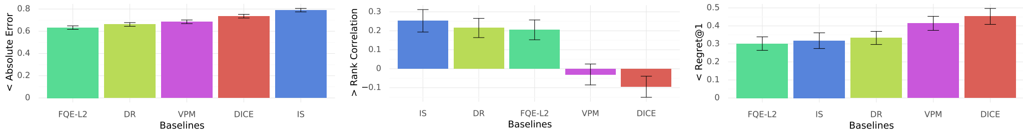


Figure 5: **DOPE D4RL** Mean overall performance of baselines.

Doubly-Robust (DR) We perform weighted doubly-robust policy evaluation Thomas & Brunskill (2016) using the implementation of Kostrikov & Nachum (2020)[3]. Specifically, this method combines the IS technique above with a value estimator for variance reduction. The value estimator is learned using deep FQE with an L2 loss function. More advanced approaches that trade variance for bias exist (e.g., MAGIC (Thomas & Brunskill, 2016)), but we leave implementing them to future work.

DICE This method uses a saddle-point objective to estimate marginalized importance weights $d^{\pi}(s,a)/d^{\pi_B}(s,a)$; these weights are then used to compute a weighted average of reward over the offline dataset, and this serves as an estimate of the policy's value in the MDP. We use the implementation from Yang et al. (2020) corresponding to the algorithm *BestDICE*.[4]

Variational Power Method (VPM) This method runs a variational power iteration algorithm to estimate the importance weights $d^{\pi}(s,a)/d^{\pi_B}(s,a)$ without the knowledge of the behavior policy. It then estimates the target policy value using weighted average of rewards similar to the DICE method. Our implementation is based on the same network and hyperparameters for OPE setting as in Wen et al. (2020). We further tune the hyper-parameters including the regularization parameter λ, learning rates α_{θ} and α_v, and number of iterations on the Cartpole swingup task using ground-truth policy value, and then fix them for all other tasks.

5.2 Results

To facilitate aggregate metrics and comparisons between tasks and between DOPE RL Unplugged and DOPE D4RL, we normalize the returns and estimated returns to range between 0 and 1. For each set of policies we compute the worst value $V_{worst} = min\{V^{\pi_1}, V^{\pi_2}, ..., V^{\pi_N}\}$ and best value $V_{best} = max\{V^{\pi_1}, V^{\pi_2}, ..., V^{\pi_N}\}$ and normalize the returns and estimated returns according to $x' = (x - V_{worst})/(V_{best} - V_{worst})$.

We present results averaged across DOPE RL Unplugged in Fig. 4, and results for DOPE D4RL in Fig. 5. Overall, no evaluated algorithm attains near-oracle performance under any metric (absolute error, regret, or rank correlation). Because the dataset is finite, we do not expect that achieving oracle performance is possible. Nevertheless, based on recent progress on this benchmark (e.g., Zhang et al. (2021)), we hypothesize that the benchmark has room for improvement, making it suitable for driving further improvements on OPE methods and facilitating the development of OPE algorithms that can provide reliable estimates on the types of high-dimensional problems that we consider.

While all algorithms achieve sub-optimal performance, some perform better than others. We find that on the DOPE RL Unplugged tasks model based (MB-AR, MB-FF) and direct value based methods (FQE-D, FQE-L2) significantly outperform importance sampling methods (VPM, DICE, IS) across all metrics. This is somewhat surprising as DICE and VPM have shown promising results in other settings. We hypothesize that this is due to the relationship between the behavior data and evaluation policies, which is different from standard OPE settings. Recall that in DOPE RL Unplugged the behavior data is collected from an online RL algorithm and the evaluation policies are learned via offline RL from the behavior data. In our experience all methods work better when the behavior policy is a noisy/perturbed version of the evaluation policy. Moreover, MB and FQE-based methods may

[4] Code available at `https://github.com/google-research/dice_rl`.

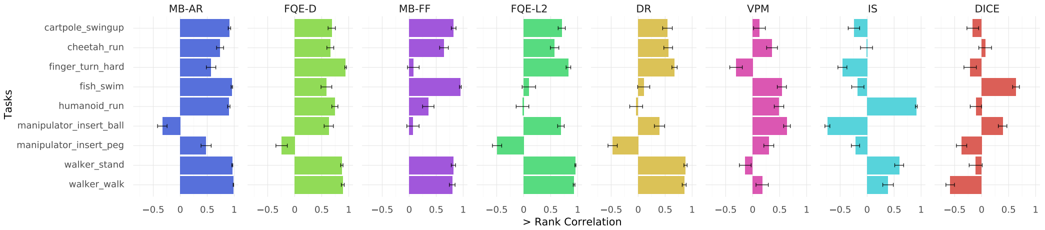


Figure 6: Rank correlation for each baseline algorithm for each RL Unplugged task considered.

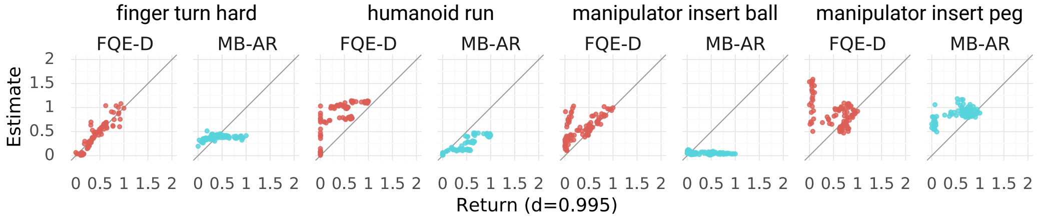


Figure 7: Scatter plots of estimate vs ground truth return for MB-AR and FQE-D on selected tasks.

implicitly benefit from the architectural and optimization advancements made in policy optimization settings, which focus on similar environments and where these methods are more popular than importance sampling approaches. Note that within the MB and FQE methods, design details can create a significant difference in performance. For example model architecture (MB-AR vs MB-FF) and implementation differences (FQE-D vs FQE-L2) show differing performance on certain tasks.

On DOPE D4RL, direct value based methods still do well, with FQE-L2 performing best on the Absolute Error and Regret@1 metrics. However, there are cases where other methods outperform FQE. Notably, IS and DR outperform FQE-L2 under the rank correlation metric. As expected, there is a clear performance gap between DOPE RL Unplugged and DOPE D4RL. While both domains have challenging tasks, algorithms perform better under the more ideal conditions of DOPE RL Unplugged than under the challenging conditions of DOPE D4RL (0.69 vs 0.25 rank correlation respectively).

In Fig. A.2 we show the rank correlation for each task in DOPE RL Unplugged. Most tasks follow the overall trends, but we will highlight a few exceptions. 1) Importance sampling is among the best methods for the *humanoid run* task, significantly outperforming direct value-based methods. 2) while MB-AR and FQE-D are similar overall, there are a few tasks where the difference is large, for example FQE-D outperfroms MB-AR on *finger turn hard*, and *manipulator insert ball*, where as MB-AR outperforms FQE-D on *cartpole swingup*, *fish swim*, *humanoid run*, and *manipulator insert peg*. We show the scatter plots for MB-AR and FQE-D on these tasks in Fig 7 which highlights different failure modes: when MB-AR performs worse, it assigns similar values for all policies; when FQE-D performs worse, it severely over-estimates the values of poor policies.

We present more detailed results, separated by task, in Appendix A.2. Note in particular how in Table A.2.2, which shows the regret@1 metric for different D4RL tasks, the particular choice of dataset for the Gym-MuJoCo, Adroit, and AntMaze domains causes a significant difference in the performance of OPE methods. This indicates the importance of evaluating multiple distinct datasets, with different data distribution properties (e.g., more narrow datasets, such as expert data, vs. broader datasets, such as random data), as no tested method is reliably robust to the effects of dataset variation.

High-dimensional tasks requiring temporally extended control were also challenging, as highlighted by the performance on the AntMaze domain. No algorithm was able to achieve a good absolute error value on such tasks, and importance sampling was the only method able to achieve a correlation consistently above zero, suggesting that these more complex tasks are a particularly important area for future methods to focus on.

6 RELATED WORK

Off-policy evaluation (OPE) has been studied extensively across a range of different domains, from healthcare (Thapa et al., 2005; Raghu et al., 2018; Nie et al., 2019), to recommender systems (Li et al., 2010; Dudík et al., 2014; Theocharous et al., 2015), and robotics (Kalashnikov et al., 2018). While a full survey of OPE methods is outside the scope of this article, broadly speaking we can categories OPE methods into groups based the use of importance sampling (Precup, 2000), value functions (Sutton et al., 2009; Migliavacca et al., 2010; Sutton et al., 2016; Yang et al., 2020), and learned transition models (Paduraru, 2007), though a number of methods combine two or more of these components (Jiang & Li, 2015; Thomas & Brunskill, 2016; Munos et al., 2016). A significant body of work in OPE is also concerned with providing statistical guarantees (Thomas et al., 2015). Our focus instead is on empirical evaluation – while theoretical analysis is likely to be a critical part of future OPE research, combining such analysis with empirical demonstration on broadly accepted and standardized benchmarks is likely to facilitate progress toward practically useful algorithms.

Current evaluation of OPE methods is based around several metrics, including error in predicting the true return of the evaluated policy (Voloshin et al., 2019), correlation between the evaluation output and actual returns (Irpan et al., 2019), and ranking and model selection metrics (Doroudi et al., 2017). As there is no single accepted metric used by the entire community, we provide a set of candidate metrics along with our benchmark, with a detailed justification in Section 5. Our work is closely related to (Paine et al., 2020) which studies OPE in a similar setting, however in our work we present a benchmark for the community and compare a range of OPE methods. Outside of OPE, standardized benchmark suites have led to considerable standardization and progress in RL (Stone & Sutton, 2001; Dutech et al., 2005; Riedmiller et al., 2007). The Arcade Learning Environment (ALE) (Bellemare et al., 2013) and OpenAI Gym (Brockman et al., 2016) have been widely used to compare online RL algorithms to good effect. More recently, Gulcehre et al. (2020); Fu et al. (2020) proposed benchmark tasks for offline RL. Our benchmark is based on the tasks and environments described in these two benchmarks, which we augment with a set of standardized policies for evaluation, results for a number of existing OPE methods, and standardized evaluation metrics and protocols. Voloshin et al. (2019) have recently proposed benchmarking for OPE methods on a variety of tasks ranging from tabular problems to image-based tasks in Atari. Our work differs in several key aspects. Voloshin et al. (2019) is composed entirely of discrete action tasks, whereas out benchmark focuses on continuous action tasks. Voloshin et al. (2019) assumes full support for the evaluation policy under the behavior policy data, whereas we designed our datasets and policies to ensure that different cases of dataset and policy distributions could be studied. Finally, all evaluations in Voloshin et al. (2019) are performed using the MSE metric, and they do not provide standardized datasets. In contrast, we provide a variety of policies for each problem which enables one to evaluate metrics such as ranking for policy selection, and a wide range of standardized datasets for reproducbility.

7 CONCLUSION

We have presented the Deep Off-Policy Evaluation (DOPE) benchmark, which aims to provide a platform for studying policy evaluation and selection across a wide range of challenging tasks and datasets. In contrast to prior benchmarks, DOPE provides multiple datasets and policies, allowing researchers to study how data distributions affect performance and to evaluate a wide variety of metrics, including those that are relevant for offline policy selection. In comparing existing OPE methods, we find that no existing algorithms consistently perform well across all of the tasks, which further reinforces the importance of standardized and challenging OPE benchmarks. Moreover, algorithms that perform poorly under one metric, such as absolute error, may perform better on other metrics, such as correlation, which provides insight into what algorithms to use depending on the use case (e.g., policy evaluation vs. policy selection).

We believe that OPE is an exciting area for future research, as it allows RL agents to learn from large and abundant datasets in domains where online RL methods are otherwise infeasible. We hope that our benchmark will enable further progress in this field, though important evaluation challenges remain. As the key benefit of OPE is the ability to utilize real-world datasets, a promising direction for future evaluation efforts is to devise effective ways to use such data, where a key challenge is to develop evaluation protocols that are both reproducible and accessible. This could help pave the way towards developing intelligent decision making agents that can leverage vast banks of logged information to solve important real-world problems.

References

Charu C Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016.

Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributional policy gradients. In *International Conference on Learning Representations*, 2018.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pp. 4754–4765, 2018.

Shayan Doroudi, Philip S Thomas, and Emma Brunskill. Importance sampling for fair policy selection. *Grantee Submission*, 2017.

Miroslav Dudík, Dumitru Erhan, John Langford, Lihong Li, et al. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Alain Dutech, Timothy Edmunds, Jelle Kok, Michail Lagoudakis, Michael Littman, Martin Riedmiller, Bryan Russell, Bruno Scherrer, Richard Sutton, Stephan Timmer, et al. Reinforcement learning benchmarks and bake-offs ii. *Advances in Neural Information Processing Systems (NIPS)*, 17:6, 2005.

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.

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

Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. Offline a/b testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 198–206, 2018.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3389–3396. IEEE, 2017.

Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Tom Le Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel Mankowitz, Cosmin Paduraru, et al. Rl unplugged: Benchmarks for offline reinforcement learning. *arXiv preprint arXiv:2006.13888*, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Josiah Hanna, Scott Niekum, and Peter Stone. Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, pp. 2605–2613. PMLR, 2019.

Milos Hauskrecht and Hamish Fraser. Planning treatment of ischemic heart disease with partially observable markov decision processes. *Artificial Intelligence in Medicine*, 18(3):221–244, 2000.

Matt Hoffman, Bobak Shahriari, John Aslanides, Gabriel Barth-Maron, Feryal Behbahani, Tamara Norman, Abbas Abdolmaleki, Albin Cassirer, Fan Yang, Kate Baumli, et al. Acme: A research framework for distributed reinforcement learning. *arXiv preprint arXiv:2006.00979*, 2020.

Alexander Irpan, Kanishka Rao, Konstantinos Bousmalis, Chris Harris, Julian Ibarz, and Sergey Levine. Off-policy evaluation via off-policy classification. In *Advances in Neural Information Processing Systems*, pp. 5437–5448, 2019.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pp. 12519–12530, 2019.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8248–8254. IEEE, 2019.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Ilya Kostrikov and Ofir Nachum. Statistical bootstrapping for uncertainty estimation in off-policy evaluation, 2020.

Hoang M Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. *arXiv preprint arXiv:1903.08738*, 2019.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

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.

Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, pp. 1077–1084, 2014.

Martino Migliavacca, Alessio Pecorino, Matteo Pirotta, Marcello Restelli, and Andrea Bonarini. Fitted policy search: Direct policy search using a batch reinforcement learning approach. In *3rd International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems (ERLARS 2010)*, pp. 35. Citeseer, 2010.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. *arXiv preprint arXiv:1606.02647*, 2016.

Xinkun Nie, Emma Brunskill, and Stefan Wager. Learning when-to-treat policies. *arXiv preprint arXiv:1905.09751*, 2019.

Cosmin Paduraru. Planning with approximate and learned models of markov decision processes. 2007.

Tom Le Paine, Cosmin Paduraru, Andrea Michi, Caglar Gulcehre, Konrad Zolna, Alexander Novikov, Ziyu Wang, and Nando de Freitas. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Aniruddh Raghu, Omer Gottesman, Yao Liu, Matthieu Komorowski, Aldo Faisal, Finale Doshi-Velez, and Emma Brunskill. Behaviour policy estimation in off-policy policy evaluation: Calibration matters. *arXiv preprint arXiv:1807.01066*, 2018.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Martin Riedmiller, Jan Peters, and Stefan Schaal. Evaluation of policy gradient methods and variants on the cart-pole benchmark. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 254–261. IEEE, 2007.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. In *International Conference on Learning Representations*, 2020.

Peter Stone and Richard S Sutton. Scaling reinforcement learning toward robocup soccer. In *Icml*, volume 1, pp. 537–544. Citeseer, 2001.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 993–1000, 2009.

Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1): 2603–2631, 2016.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pp. 814–823, 2015.

Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*, pp. 3632–3642, 2017.

Gerald Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3): 58–68, 1995.

Devinder Thapa, In-Sung Jung, and Gi-Nam Wang. Agent based decision support system using reinforcement learning under emergency circumstances. In *International Conference on Natural Computation*, pp. 888–892. Springer, 2005.

Georgios Theocharous, Philip S Thomas, and Mohammad Ghavamzadeh. Personalized ad recommendation systems for life-time value optimization with guarantees. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148, 2016.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

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*, pp. 5026–5033. IEEE, 2012.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Cameron Voloshin, Hoang M Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudık. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pp. 3589–3597. PMLR, 2017.

Ziyu Wang, Alexander Novikov, Konrad Żołna, Jost Tobias Springenberg, Scott Reed, Bobak Shahriari, Noah Siegel, Josh Merel, Caglar Gulcehre, Nicolas Heess, and Nando de Freitas. Critic regularized regression. *arXiv preprint arXiv:2006.15134*, 2020.

Junfeng Wen, Bo Dai, Lihong Li, and Dale Schuurmans. Batch stationary distribution estimation. *arXiv preprint arXiv:2003.00722*, 2020.

Yuan Xie, Boyi Liu, Qiang Liu, Zhaoran Wang, Yuan Zhou, and Jian Peng. Off-policy evaluation and learning from logged bandit feedback: Error reduction via surrogate policy. *arXiv preprint arXiv:1808.00232*, 2018.

Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, and Dale Schuurmans. Off-policy evaluation via the regularized lagrangian. *arXiv preprint arXiv:2007.03438*, 2020.

Michael R Zhang, Thomas Paine, Ofir Nachum, Cosmin Paduraru, George Tucker, ziyu wang, and Mohammad Norouzi. Autoregressive dynamics models for offline policy evaluation and optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=kmqjgSNXby.

A Appendix

A.1 Metrics

The metrics we use in our paper are defined as follows:

Absolute Error We evaluate policies using absolute error in order to be robust to outliers. The absolute error is defined as the difference between the value and estimated value of a policy:

$$\text{AbsErr} = |V^{\pi} - \hat{V}^{\pi}| \tag{2}$$

Where V^{π} is the true value of the policy, and $\hat{V}^{\pi}$ is the estimated value of the policy.

Regret@k Regret@k is the difference between the value of the best policy in the entire set, and the value of the best policy in the top-k set (where the top-k set is chosen by estimated values). It can be defined as:

$$\text{Regret @ k} = \max_{i \in 1:N} V_i^{\pi} - \max_{j \in \text{topk}(1:N)} V_j^{\pi} \tag{3}$$

Where $\text{topk}(1:N)$ denotes the indices of the top K policies as measured by estimated values $\hat{V}^{\pi}$.

Rank correlation Rank correlation (also Spearman's ρ) measures the correlation between the ordinal rankings of the value estimates and the true values. It can be written as:

$$\text{RankCorr} = \frac{\text{Cov}(V_{1:N}^{\pi}, \hat{V}_{1:N}^{\pi})}{\sigma(V_{1:N}^{\pi})\sigma(\hat{V}_{1:N}^{\pi})} \tag{4}$$

A.2 Detailed Results

Detailed results figures and tables are presented here. We show results by task in both tabular and chart form, as well as scatter plots which compare the estimated returns against the ground truth returns for every policy.

A.2.1 Chart Results

First we show the normalized results for each algorithm and task.

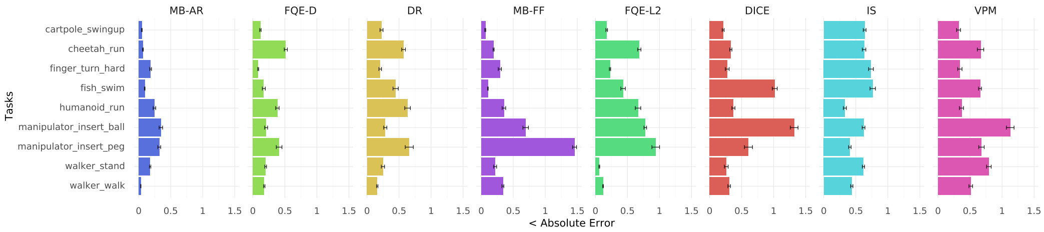


Figure A.1: Absolute error for each baseline algorithm for each RL Unplugged task considered.

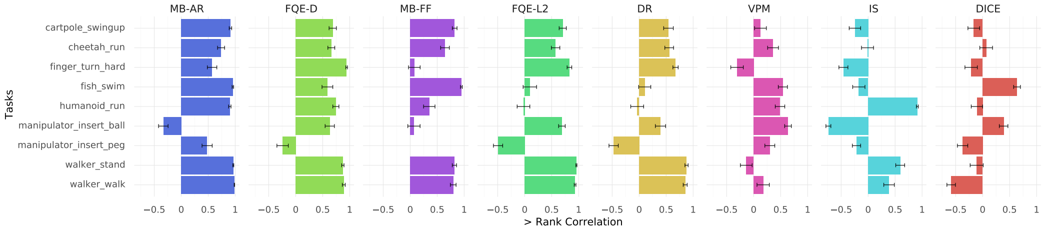


Figure A.2: Rank correlation for each baseline algorithm for each RL Unplugged task considered.

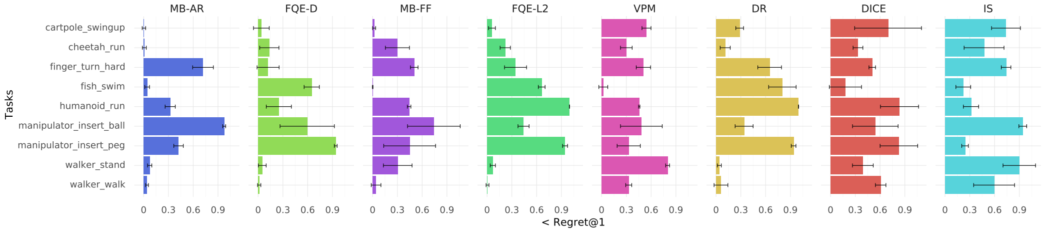


Figure A.3: Regret@1 for each baseline algorithm for each RL Unplugged task considered.

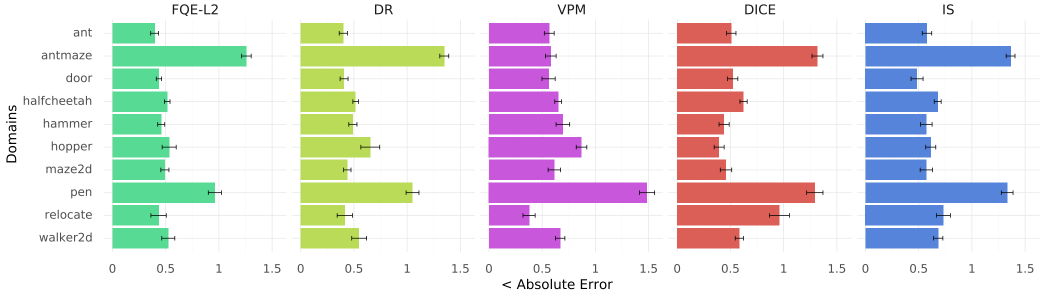


Figure A.4: Absolute error for each baseline algorithm for each D4RL task domain considered.

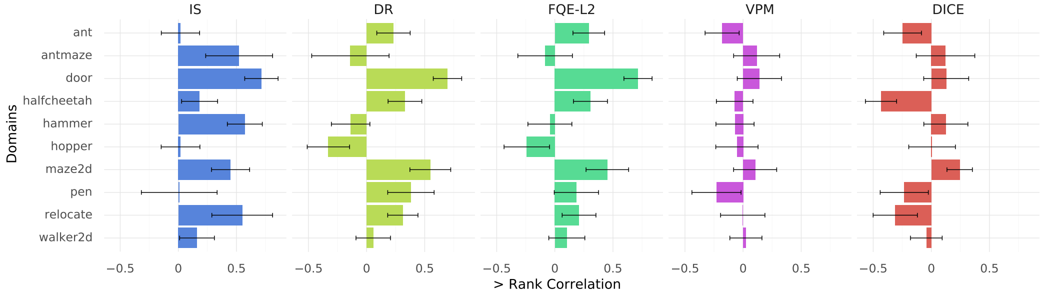


Figure A.5: Rank correlation for each baseline algorithm for each D4RL task domain considered.

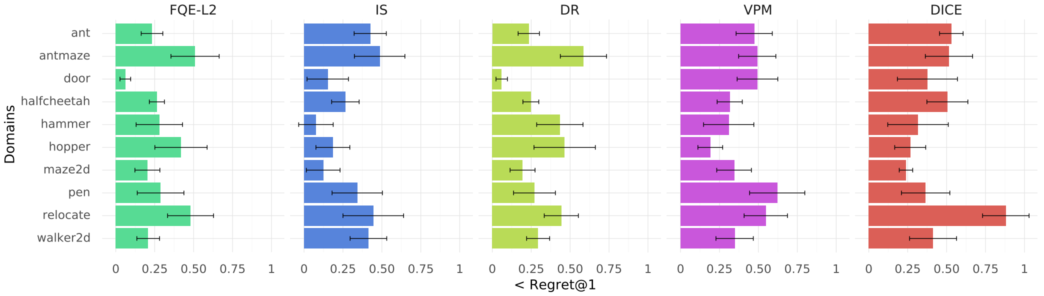


Figure A.6: Regret@1 for each baseline algorithm for each D4RL task domain considered.

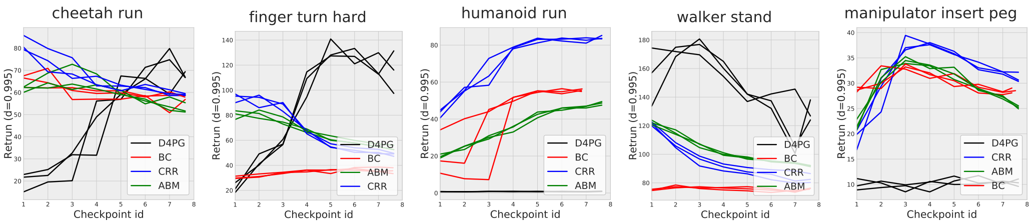


Figure A.7: Online evaluation of policy checkpoints for 4 Offline RL algorithms with 3 random seeds. We observe a large degree of variability between the behavior of algorithms on different tasks.

A.2.2 Tabular Results

Next, we present the results for each task and algorithm in tabular form, with means and standard deviations reported across 3 seeds.

		Cartpole swingup	Cheetah run	Finger turn hard	Fish swim	Humanoid run
Absolute Error btw. OPE and ground truth	Variational power method	37.53 ±3.50	61.89 ±4.25	46.22 ±3.93	31.27 ±0.99	35.29 ±3.03
	Importance Sampling	68.75 ±2.39	44.29 ±1.91	90.10 ±4.68	34.82 ±1.93	27.89 ±1.98
	Best DICE	22.73 ±1.65	23.35 ±1.32	33.52 ±3.48	59.48 ±2.47	31.42 ±2.04
	Model based - FF	**6.80 ±0.85**	13.64 ±0.59	35.99 ±3.00	**4.75 ±0.23**	30.12 ±2.40
	FQE (L2)	19.02 ±1.34	48.26 ±1.78	27.91 ±1.18	19.82 ±1.57	56.28 ±3.52
	Doubly Robust (IS, FQE)	24.38 ±2.51	40.27 ±2.05	25.26 ±2.48	20.28 ±1.90	53.64 ±3.68
	FQE (distributional)	12.63 ±1.21	36.50 ±1.62	**10.23 ±0.93**	7.76 ±0.95	32.36 ±2.27
	Model based - AR	**5.32 ±0.54**	**4.64 ±0.46**	22.93 ±1.72	**4.31 ±0.22**	**20.95 ±1.61**

		Walker stand	Walker walk	Manipulator insert ball	Manipulator insert peg	Median ↓
Absolute Error btw. OPE and ground truth	Variational power method	96.76 ±3.59	87.24 ±4.25	79.25 ±6.19	21.95 ±1.17	46.22
	Importance Sampling	66.50 ±1.90	67.24 ±2.70	29.93 ±1.10	12.78 ±0.66	44.29
	Best DICE	27.58 ±3.01	47.28 ±3.13	103.45 ±5.21	22.75 ±3.00	31.42
	Model based - FF	23.34 ±2.41	52.23 ±2.34	34.30 ±2.55	121.12 ±1.58	30.12
	FQE (L2)	**6.51 ±0.71**	18.34 ±0.95	36.32 ±1.07	31.12 ±2.37	27.91
	Doubly Robust (IS, FQE)	26.82 ±2.66	24.63 ±1.69	13.33 ±1.16	22.28 ±2.34	24.63
	FQE (distributional)	21.49 ±1.41	27.57 ±1.54	**9.75 ±1.10**	12.66 ±1.39	12.66
	Model based - AR	19.12 ±1.23	**5.14 ±0.49**	17.13 ±1.34	**9.71 ±0.70**	9.71

Table A.1: Average absolute error between OPE metrics and ground truth values at a discount factor of 0.995 In each column, absolute error values that are not significantly different from the best ($p > 0.05$) are bold faced. Methods are ordered by median.

		Cartpole swingup	Cheetah run	Finger turn hard	Fish swim	Humanoid run
Rank Correlation btw. OPE and ground truth	Importance Sampling	-0.23 ± 0.11	-0.01 ± 0.12	-0.45 ± 0.08	-0.17 ± 0.11	**0.91 ± 0.02**
	Best DICE	-0.16 ± 0.11	0.07 ± 0.11	-0.22 ± 0.11	0.44 ± 0.09	-0.10 ± 0.10
	Variational power method	0.01 ± 0.11	0.01 ± 0.12	-0.25 ± 0.11	0.56 ± 0.08	0.36 ± 0.09
	Doubly Robust (IS, FQE)	0.55 ± 0.09	0.56 ± 0.08	0.67 ± 0.05	0.11 ± 0.12	-0.03 ± 0.12
	Model based - FF	0.83 ± 0.05	**0.64 ± 0.08**	0.08 ± 0.11	**0.95 ± 0.02**	0.35 ± 0.10
	FQE (distributional)	0.69 ± 0.07	**0.67 ± 0.06**	**0.94 ± 0.01**	0.59 ± 0.10	0.74 ± 0.06
	FQE (L2)	0.70 ± 0.07	0.56 ± 0.08	0.83 ± 0.04	0.10 ± 0.12	-0.02 ± 0.12
	Model based - AR	**0.91 ± 0.02**	**0.74 ± 0.07**	0.57 ± 0.09	**0.96 ± 0.01**	**0.90 ± 0.02**

		Walker stand	Walker walk	Manipulator insert ball	Manipulator insert peg	Median ↑
Rank Correlation btw. OPE and ground truth	Importance Sampling	0.59 ± 0.08	0.38 ± 0.10	-0.72 ± 0.05	-0.25 ± 0.08	-0.17
	Best DICE	-0.11 ± 0.12	-0.58 ± 0.08	0.19 ± 0.11	-0.35 ± 0.10	-0.11
	Variational power method	-0.35 ± 0.10	-0.10 ± 0.11	**0.61 ± 0.08**	**0.41 ± 0.09**	0.01
	Doubly Robust (IS, FQE)	0.88 ± 0.03	0.85 ± 0.04	0.42 ± 0.10	-0.47 ± 0.09	0.55
	Model based - FF	0.82 ± 0.04	0.80 ± 0.05	0.06 ± 0.10	-0.56 ± 0.08	0.64
	FQE (distributional)	0.87 ± 0.02	0.89 ± 0.03	**0.63 ± 0.08**	-0.23 ± 0.10	0.69
	FQE (L2)	**0.96 ± 0.01**	0.94 ± 0.02	**0.70 ± 0.07**	-0.48 ± 0.08	0.70
	Model Based - AR	**0.96 ± 0.01**	**0.98 ± 0.00**	-0.33 ± 0.09	**0.47 ± 0.09**	0.90

Table A.2: Spearman's rank correlation (ρ) coefficient (bootstrap mean ± standard deviation) between different OPE metrics and ground truth values at a discount factor of 0.995. In each column, rank correlation coefficients that are not significantly different from the best ($p > 0.05$) are bold faced. Methods are ordered by median. Also see Table A.3 and Table A.1 for Normalized Regret@5 and Average Absolute Error results.

		Cartpole swingup	Cheetah run	Finger turn hard	Fish swim	Humanoid run
Regret@5 for OPE *vs.* ground truth	Importance Sampling	0.73 ± 0.16	0.40 ± 0.21	0.64 ± 0.05	0.12 ± 0.05	**0.31 ± 0.09**
	Best DICE	0.68 ± 0.41	0.27 ± 0.05	0.44 ± 0.04	**0.35 ± 0.24**	0.84 ± 0.22
	Variational power method	0.50 ± 0.13	0.37 ± 0.04	0.45 ± 0.13	**0.02 ± 0.02**	0.56 ± 0.08
	Doubly Robust (IS, FQE)	0.28 ± 0.05	**0.09 ± 0.05**	0.56 ± 0.12	0.61 ± 0.12	0.99 ± 0.00
	FQE (L2)	0.06 ± 0.04	0.17 ± 0.05	**0.30 ± 0.11**	0.50 ± 0.03	0.99 ± 0.00
	Model based - FF	**0.02 ± 0.02**	0.24 ± 0.12	0.43 ± 0.04	**0.00 ± 0.00**	0.44 ± 0.02
	FQE (distributional)	**0.03 ± 0.09**	**0.11 ± 0.09**	**0.10 ± 0.12**	0.49 ± 0.06	**0.24 ± 0.15**
	Model based - AR	**0.00 ± 0.02**	**0.01 ± 0.02**	0.63 ± 0.11	**0.03 ± 0.02**	**0.32 ± 0.06**

		Walker stand	Walker walk	Manipulator insert ball	Manipulator insert peg	Median ↓
Regret@5 for OPE *vs.* ground truth	Importance Sampling	0.54 ± 0.11	0.54 ± 0.23	0.83 ± 0.05	**0.22 ± 0.03**	0.54
	Best DICE	0.24 ± 0.07	0.55 ± 0.06	**0.44 ± 0.07**	0.75 ± 0.04	0.44
	Variational power method	0.41 ± 0.02	0.39 ± 0.02	**0.52 ± 0.20**	0.32 ± 0.02	0.41
	Doubly Robust (IS, FQE)	**0.02 ± 0.01**	**0.05 ± 0.07**	**0.30 ± 0.10**	0.73 ± 0.01	0.30
	FQE (L2)	**0.04 ± 0.02**	**0.00 ± 0.02**	**0.37 ± 0.07**	0.74 ± 0.01	0.30
	Model based - FF	**0.18 ± 0.10**	**0.03 ± 0.05**	0.83 ± 0.06	0.74 ± 0.01	0.24
	FQE (distributional)	**0.03 ± 0.03**	**0.01 ± 0.02**	**0.50 ± 0.30**	0.73 ± 0.01	0.11
	Model based - AR	**0.04 ± 0.02**	**0.04 ± 0.02**	0.85 ± 0.02	**0.30 ± 0.04**	0.04

Table A.3: Normalized Regret@5 (bootstrap mean ± standard deviation) for OPE methods *vs.* ground truth values at a discount factor of 0.995. In each column, normalized regret values that are not significantly different from the best ($p > 0.05$) are bold faced. Methods are ordered by median.

		Halfcheetah	Halfcheetah	Halfcheetah	Halfcheetah	Halfcheetah
		expert	medium	medium-expert	medium-replay	random
Abs. Error	IS	1404 ±152	**1217 ±123**	1400 ±146	1409 ±154	1405 ±155
	VPM	**945 ±164**	**1374 ±153**	1427 ±111	1384 ±148	1411 ±154
	Best DICE	**944 ±161**	**1382 ±130**	**1078 ±132**	1440 ±158	1446 ±156
	Doubly Robust	**1025 ±95**	**1222 ±134**	**1015 ±103**	**1001 ±129**	**949 ±126**
	FQE (L2)	**1031 ±95**	**1211 ±130**	**1014 ±101**	**1003 ±132**	**938 ±125**

		Antmaze	Antmaze	Antmaze	Antmaze	Antmaze
		large-diverse	large-play	medium-diverse	medium-play	umaze
Abs. Error	IS	0.62 ±0.01	0.85 ±0.00	0.55 ±0.01	0.81 ±0.00	0.62 ±0.04
	VPM	**0.02 ±0.02**	**0.26 ±0.24**	**0.07 ±0.05**	**0.11 ±0.06**	**0.12 ±0.03**
	Best DICE	5.55 ±0.36	19.62 ±1.28	2.42 ±1.56	19.47 ±2.15	14.97 ±1.93
	Doubly Robust	0.99 ±0.01	1.59 ±0.01	0.61 ±0.03	1.47 ±0.01	0.87 ±0.04
	FQE (L2)	0.53 ±0.01	0.78 ±0.00	0.29 ±0.01	0.71 ±0.01	0.39 ±0.03

		Antmaze	Door	Door	Door	Hammer
		umaze-diverse	cloned	expert	human	cloned
Abs. Error	IS	**0.14 ±0.02**	891 ±188	**648 ±122**	870 ±173	7403 ±1126
	VPM	**0.12 ±0.03**	1040 ±188	**879 ±182**	862 ±163	7459 ±1114
	Best DICE	**0.17 ±0.04**	697 ±79	**856 ±134**	1108 ±199	**4169 ±839**
	Doubly Robust	**0.11 ±0.02**	**424 ±73**	1353 ±218	**379 ±65**	6101 ±679
	FQE (L2)	**0.11 ±0.03**	**438 ±81**	1343 ±84	**389 ±60**	**5415 ±558**

		Hammer	Hammer	Maze2d	Maze2d	Maze2d
		expert	human	large	medium	umaze
Abs. Error	IS	**3052 ±608**	**7352 ±1118**	45.61 ±10.43	61.29 ±7.78	50.20 ±9.16
	VPM	7312 ±1117	**7105 ±1107**	44.10 ±10.69	60.30 ±8.37	62.81 ±8.40
	Best DICE	**3963 ±758**	**5677 ±936**	**42.46 ±9.66**	58.97 ±9.57	**21.95 ±4.69**
	Doubly Robust	**3485 ±590**	**5768 ±751**	**22.94 ±6.82**	**23.64 ±4.96**	76.93 ±4.42
	FQE (L2)	**2950 ±728**	**6000 ±612**	**24.31 ±6.56**	**35.11 ±6.33**	79.67 ±4.93

		Pen	Pen	Pen	Relocate	Relocate
		cloned	expert	human	cloned	expert
Abs. Error	IS	1707 ±128	4547 ±222	3926 ±128	**632 ±215**	2731 ±147
	VPM	2324 ±129	2325 ±136	**1569 ±215**	**586 ±135**	**620 ±214**
	Best DICE	**1454 ±219**	2963 ±279	4193 ±244	1347 ±485	**1095 ±221**
	Doubly Robust	**1323 ±98**	**2013 ±564**	2846 ±200	**412 ±124**	**1193 ±350**
	FQE (L2)	**1232 ±105**	**1057 ±281**	2872 ±170	**439 ±125**	1351 ±393

		Relocate	Ant	Ant	Ant	Ant
		human	expert	medium	medium-expert	medium-replay
Abs. Error	IS	**638 ±217**	**605 ±104**	594 ±104	604 ±102	**603 ±101**
	VPM	**806 ±166**	**607 ±108**	570 ±109	604 ±106	**612 ±105**
	Best DICE	4526 ±474	**558 ±108**	**495 ±90**	**471 ±100**	**583 ±110**
	Doubly Robust	**606 ±116**	**584 ±114**	**345 ±66**	**326 ±66**	**421 ±72**
	FQE (L2)	**593 ±113**	**583 ±122**	**345 ±64**	**319 ±67**	**410 ±79**

		Ant	Hopper	Hopper	Hopper	Walker2d
		random	expert	medium	random	expert
Abs. Error	IS	**606 ±103**	**106 ±29**	405 ±48	412 ±45	**405 ±62**
	VPM	**570 ±99**	442 ±43	433 ±44	438 ±44	**367 ±68**
	Best DICE	**530 ±92**	259 ±54	**215 ±41**	**122 ±16**	**437 ±60**
	Doubly Robust	**404 ±106**	426 ±99	**307 ±85**	289 ±50	**519 ±179**
	FQE (L2)	**398 ±111**	282 ±76	**283 ±73**	261 ±42	**453 ±142**

		Walker2d	Walker2d	Walker2d	Walker2d	Median
		medium	medium-expert	medium-replay	random	
Abs. Error	IS	428 ±60	436 ±62	**427 ±60**	**430 ±61**	603.82
	VPM	426 ±60	425 ±61	**424 ±64**	**440 ±58**	585.53
	Best DICE	**273 ±31**	**322 ±60**	**374 ±51**	**419 ±57**	530.43
	Doubly Robust	**368 ±74**	**217 ±46**	**296 ±54**	**347 ±74**	411.99
	FQE (L2)	**350 ±79**	**233 ±42**	**313 ±73**	**354 ±73**	398.37

		Halfcheetah expert	Halfcheetah medium-expert	Halfcheetah medium-replay	Halfcheetah random	Door cloned
Rank Corr.	Best DICE	−0.44 ±0.30	**−0.08 ±0.35**	**−0.15 ±0.41**	−0.70 ±0.22	**0.18 ±0.31**
	VPM	**0.18 ±0.35**	−0.47 ±0.29	**−0.07 ±0.36**	**0.27 ±0.36**	−0.29 ±0.36
	FQE (L2)	**0.78 ±0.15**	**0.62 ±0.27**	**0.26 ±0.37**	**−0.11 ±0.41**	**0.55 ±0.27**
	IS	0.01 ±0.35	**−0.06 ±0.37**	**0.59 ±0.26**	**−0.24 ±0.36**	**0.66 ±0.22**
	Doubly Robust	**0.77 ±0.17**	**0.62 ±0.27**	**0.32 ±0.37**	**−0.02 ±0.38**	**0.60 ±0.28**
		Door expert	Hammer cloned	Hammer expert	Maze2d large	Maze2d medium
Rank Corr.	Best DICE	−0.06 ±0.32	**0.35 ±0.38**	−0.42 ±0.31	**0.56 ±0.21**	−0.64 ±0.23
	VPM	**0.65 ±0.23**	−0.77 ±0.22	**0.39 ±0.31**	−0.26 ±0.33	**−0.05 ±0.39**
	FQE (L2)	**0.89 ±0.09**	**−0.15 ±0.33**	**0.29 ±0.34**	**0.30 ±0.36**	**0.16 ±0.38**
	IS	**0.76 ±0.17**	**0.58 ±0.27**	**0.64 ±0.24**	**0.63 ±0.19**	**0.44 ±0.25**
	Doubly Robust	**0.76 ±0.13**	−0.70 ±0.20	**0.49 ±0.31**	**0.31 ±0.36**	**0.41 ±0.35**
		Pen expert	Relocate expert	Ant expert	Ant medium	Ant medium-expert
Rank Corr.	Best DICE	−0.53 ±0.30	−0.27 ±0.34	**−0.13 ±0.37**	−0.36 ±0.28	**−0.33 ±0.40**
	VPM	**0.08 ±0.33**	**0.39 ±0.31**	**−0.42 ±0.38**	−0.20 ±0.31	**−0.28 ±0.28**
	FQE (L2)	**−0.01 ±0.33**	−0.57 ±0.28	**−0.13 ±0.32**	**0.65 ±0.25**	**0.37 ±0.35**
	IS	−0.45 ±0.31	**0.52 ±0.23**	**0.14 ±0.41**	−0.17 ±0.32	**−0.21 ±0.35**
	Doubly Robust	**0.52 ±0.28**	−0.40 ±0.24	**−0.28 ±0.32**	**0.66 ±0.26**	**0.35 ±0.35**
		Ant medium-replay	Ant random	Hopper expert	Hopper medium	Hopper random
Rank Corr.	Best DICE	−0.24 ±0.39	**−0.21 ±0.35**	**−0.08 ±0.32**	**0.19 ±0.33**	**−0.13 ±0.39**
	VPM	−0.26 ±0.29	**0.24 ±0.31**	**0.21 ±0.32**	**0.13 ±0.37**	−0.46 ±0.20
	FQE (L2)	**0.57 ±0.28**	**0.04 ±0.33**	**−0.33 ±0.30**	**−0.29 ±0.33**	**−0.11 ±0.36**
	IS	**0.07 ±0.39**	**0.26 ±0.34**	**0.37 ±0.27**	−0.55 ±0.26	**0.23 ±0.34**
	Doubly Robust	**0.45 ±0.32**	**0.01 ±0.33**	−0.41 ±0.27	**−0.31 ±0.34**	**−0.19 ±0.36**
		Walker2d expert	Walker2d medium	Walker2d medium-expert	Walker2d medium-replay	Walker2d random
Rank Corr.	Best DICE	**−0.37 ±0.27**	**0.12 ±0.38**	**−0.34 ±0.34**	**0.55 ±0.23**	**−0.19 ±0.36**
	VPM	**0.17 ±0.32**	**0.44 ±0.21**	**0.49 ±0.37**	−0.52 ±0.25	**−0.42 ±0.34**
	FQE (L2)	**0.35 ±0.33**	**−0.09 ±0.36**	**0.25 ±0.32**	−0.19 ±0.36	**0.21 ±0.31**
	IS	**0.22 ±0.37**	**−0.25 ±0.35**	**0.24 ±0.33**	**0.65 ±0.24**	**−0.05 ±0.38**
	Doubly Robust	**0.26 ±0.34**	**0.02 ±0.37**	**0.19 ±0.33**	−0.37 ±0.39	**0.16 ±0.29**
		Median				
Rank Corr.	Best DICE	−0.19				
	VPM	−0.05				
	FQE (L2)	0.21				
	IS	0.23				
	Doubly Robust	0.26				

		Halfcheetah expert	Halfcheetah medium	Halfcheetah medium-expert	Halfcheetah medium-replay	Halfcheetah random
Regret@1	Best DICE	**0.32 ±0.40**	0.82 ±0.29	**0.38 ±0.37**	**0.30 ±0.07**	0.81 ±0.30
	VPM	**0.14 ±0.09**	**0.33 ±0.19**	0.80 ±0.34	**0.25 ±0.09**	**0.12 ±0.07**
	Doubly Robust	**0.11 ±0.08**	0.37 ±0.15	**0.14 ±0.07**	**0.33 ±0.18**	**0.31 ±0.10**
	FQE (L2)	**0.12 ±0.07**	0.38 ±0.13	**0.14 ±0.07**	**0.36 ±0.16**	0.37 ±0.08
	IS	**0.15 ±0.08**	**0.05 ±0.05**	**0.73 ±0.42**	**0.13 ±0.10**	**0.31 ±0.11**

		Antmaze large-diverse	Antmaze large-play	Antmaze medium-diverse	Antmaze medium-play	Antmaze umaze
Regret@1	Best DICE	**0.54 ±0.34**	**0.96 ±0.13**	**0.04 ±0.11**	**0.09 ±0.10**	**0.69 ±0.39**
	VPM	**0.88 ±0.27**	**0.45 ±0.30**	**0.14 ±0.10**	**0.03 ±0.08**	**0.62 ±0.32**
	Doubly Robust	**0.83 ±0.30**	**0.93 ±0.21**	**0.05 ±0.07**	**0.17 ±0.31**	**0.42 ±0.36**
	FQE (L2)	**0.93 ±0.25**	**1.00 ±0.03**	**0.16 ±0.10**	**0.05 ±0.19**	**0.41 ±0.35**
	IS	**0.39 ±0.26**	**0.71 ±0.20**	**0.14 ±0.09**	**0.18 ±0.06**	**0.86 ±0.06**

		Antmaze umaze-diverse	Door cloned	Door expert	Door human	Hammer cloned
Regret@1	Best DICE	**0.42 ±0.28**	**0.65 ±0.45**	0.37 ±0.27	**0.10 ±0.27**	**0.67 ±0.48**
	VPM	**0.63 ±0.32**	**0.81 ±0.33**	**0.03 ±0.03**	0.69 ±0.24	0.72 ±0.39
	Doubly Robust	**0.79 ±0.14**	**0.11 ±0.08**	**0.05 ±0.07**	**0.05 ±0.09**	0.78 ±0.38
	FQE (L2)	**0.64 ±0.37**	**0.11 ±0.06**	**0.03 ±0.03**	**0.05 ±0.08**	**0.36 ±0.39**
	IS	**0.22 ±0.36**	**0.02 ±0.07**	**0.01 ±0.04**	**0.45 ±0.40**	**0.03 ±0.15**

		Hammer expert	Hammer human	Maze2d large	Maze2d medium	Maze2d umaze
Regret@1	Best DICE	**0.24 ±0.34**	**0.04 ±0.08**	**0.15 ±0.08**	**0.44 ±0.05**	**0.03 ±0.07**
	VPM	**0.04 ±0.07**	**0.18 ±0.29**	0.66 ±0.10	**0.24 ±0.24**	**0.06 ±0.12**
	Doubly Robust	**0.09 ±0.09**	0.46 ±0.23	**0.21 ±0.16**	**0.27 ±0.14**	**0.03 ±0.07**
	FQE (L2)	0.05 ±0.04	0.46 ±0.23	**0.20 ±0.14**	**0.31 ±0.14**	**0.03 ±0.07**
	IS	**0.01 ±0.04**	**0.19 ±0.30**	**0.16 ±0.23**	**0.15 ±0.15**	**0.02 ±0.12**

		Pen cloned	Pen expert	Pen human	Relocate cloned	Relocate expert
Regret@1	Best DICE	**0.12 ±0.08**	**0.33 ±0.20**	**0.04 ±0.09**	**0.96 ±0.18**	0.97 ±0.07
	VPM	**0.36 ±0.18**	**0.25 ±0.13**	**0.28 ±0.12**	**0.11 ±0.29**	**0.76 ±0.23**
	Doubly Robust	**0.13 ±0.06**	**0.05 ±0.07**	**0.09 ±0.08**	**0.18 ±0.27**	0.98 ±0.08
	FQE (L2)	**0.12 ±0.07**	**0.11 ±0.14**	**0.07 ±0.05**	**0.29 ±0.42**	1.00 ±0.06
	IS	**0.14 ±0.09**	**0.31 ±0.10**	**0.17 ±0.15**	**0.63 ±0.41**	**0.18 ±0.14**

		Relocate human	Ant expert	Ant medium	Ant medium-expert	Ant medium-replay
Regret@1	Best DICE	0.97 ±0.11	**0.62 ±0.15**	**0.43 ±0.10**	**0.60 ±0.16**	0.64 ±0.13
	VPM	0.77 ±0.18	**0.88 ±0.22**	**0.40 ±0.21**	**0.32 ±0.24**	0.72 ±0.43
	Doubly Robust	**0.17 ±0.15**	**0.43 ±0.22**	**0.12 ±0.18**	**0.37 ±0.13**	**0.05 ±0.09**
	FQE (L2)	**0.17 ±0.14**	**0.43 ±0.22**	**0.12 ±0.18**	**0.36 ±0.14**	**0.05 ±0.09**
	IS	**0.63 ±0.41**	**0.47 ±0.32**	0.61 ±0.18	**0.46 ±0.18**	**0.16 ±0.23**

		Ant random	Hopper expert	Hopper medium	Hopper random	Walker2d expert
Regret@1	Best DICE	**0.50 ±0.29**	0.20 ±0.08	**0.18 ±0.19**	**0.30 ±0.15**	**0.35 ±0.36**
	VPM	**0.15 ±0.24**	**0.13 ±0.10**	**0.10 ±0.14**	0.26 ±0.10	**0.09 ±0.19**
	Doubly Robust	**0.28 ±0.15**	**0.34 ±0.35**	**0.32 ±0.32**	**0.41 ±0.17**	**0.06 ±0.07**
	FQE (L2)	**0.28 ±0.15**	0.41 ±0.20	**0.32 ±0.32**	**0.36 ±0.22**	**0.06 ±0.07**
	IS	**0.56 ±0.22**	**0.06 ±0.03**	**0.38 ±0.28**	**0.05 ±0.05**	**0.43 ±0.26**

		Walker2d medium	Walker2d medium-expert	Walker2d medium-replay	Walker2d random	Median
Regret@1	Best DICE	**0.27 ±0.43**	0.78 ±0.27	**0.18 ±0.12**	**0.39 ±0.33**	0.38
	VPM	**0.08 ±0.06**	**0.24 ±0.42**	0.46 ±0.31	0.88 ±0.20	0.28
	Doubly Robust	**0.25 ±0.09**	**0.30 ±0.12**	0.68 ±0.23	**0.15 ±0.20**	0.25
	FQE (L2)	0.31 ±0.10	**0.22 ±0.14**	**0.24 ±0.20**	**0.15 ±0.21**	0.24
	IS	**0.70 ±0.39**	**0.13 ±0.07**	**0.02 ±0.05**	**0.74 ±0.33**	0.18

A.2.3 Scatter Plots

Finally, we present scatter plots plotting the true returns of each policy against the estimated returns. Each point on the plot represents one evaluated policy.

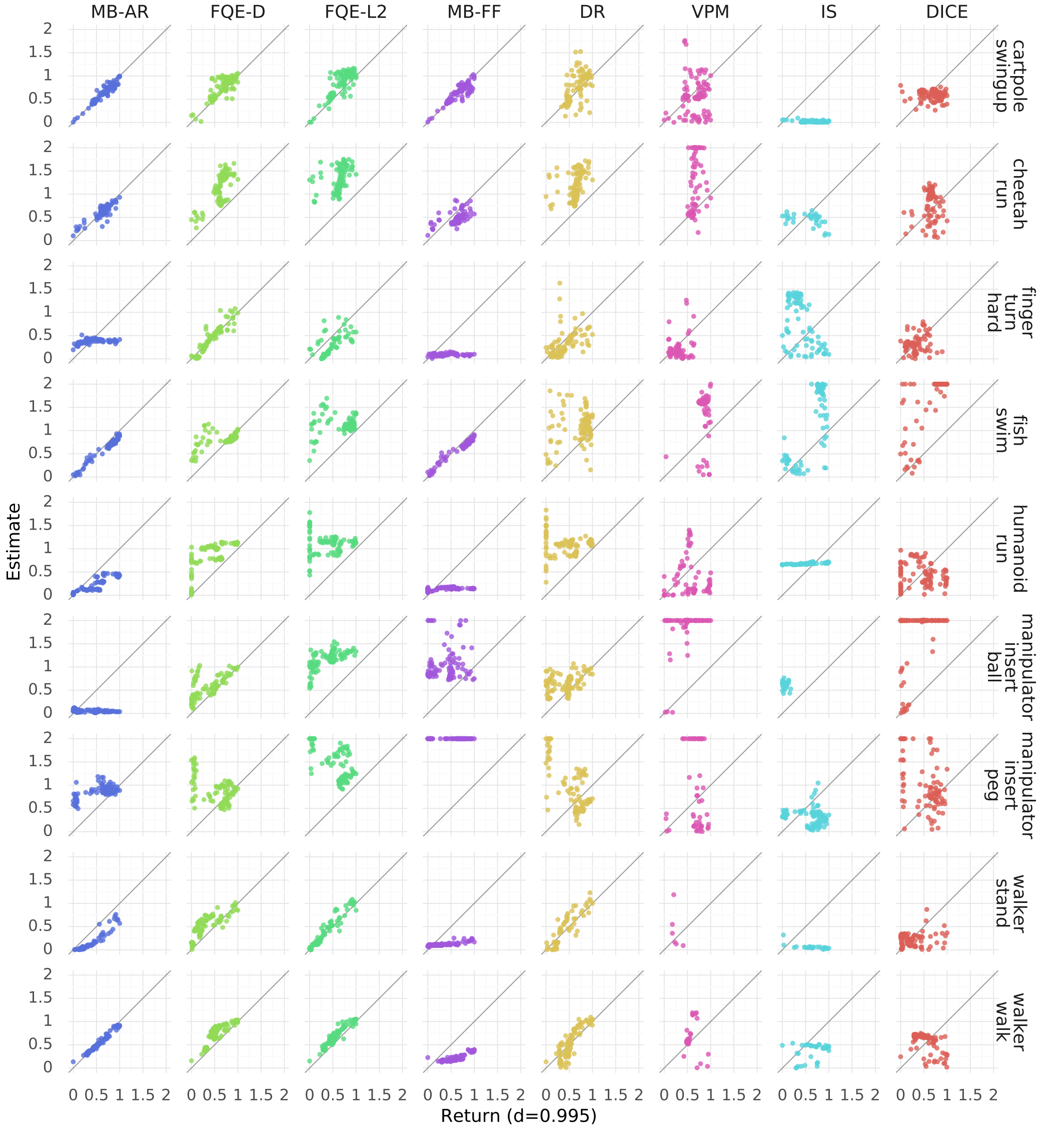


Figure A.8: Scatter plots of estimate vs ground truth return for each baseline on each task in DOPE RL Unplugged.

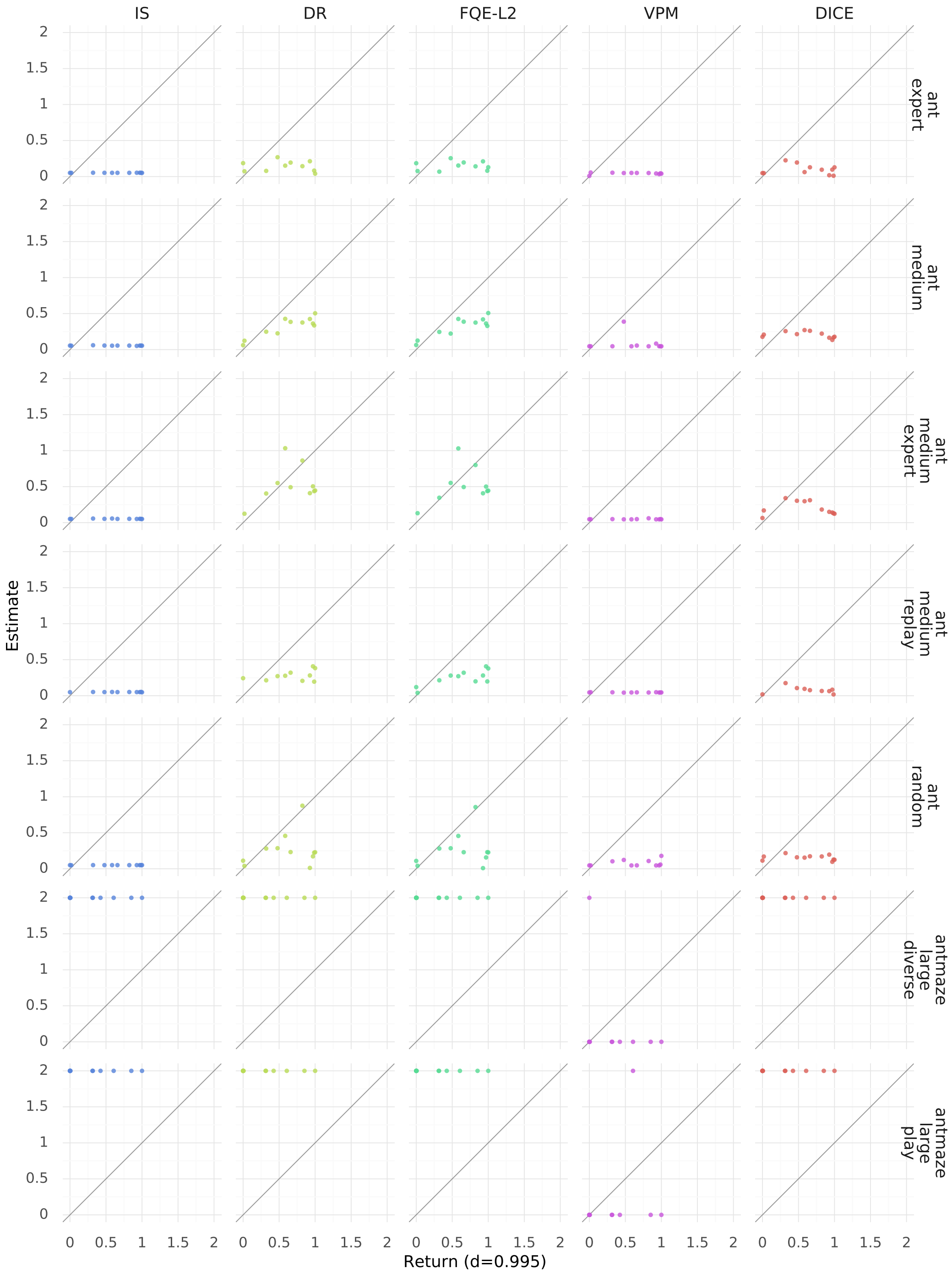


Figure A.9: Scatter plots of estimate vs ground truth return for each baseline on each task in DOPE D4RL (part 1).

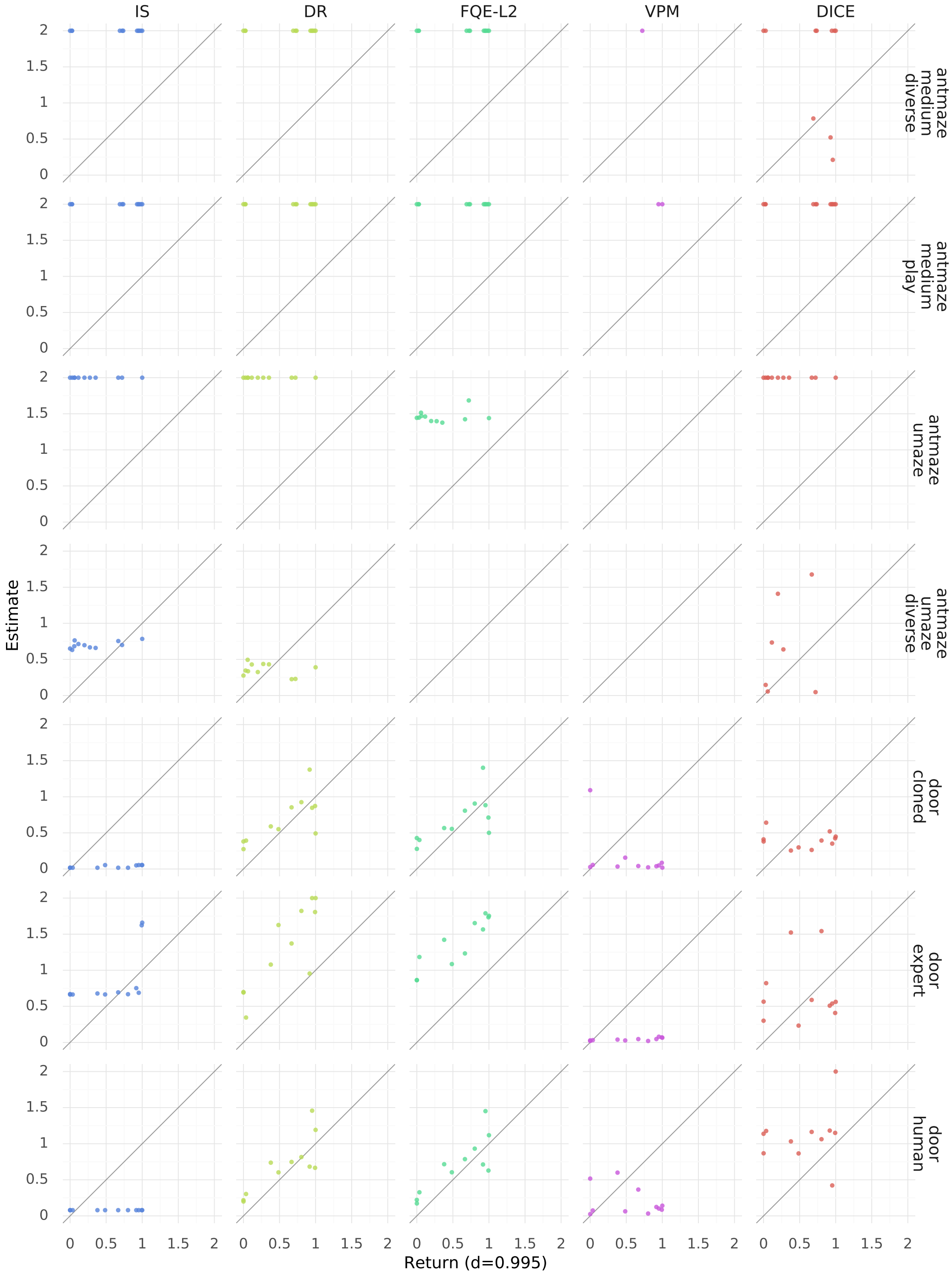


Figure A.10: Scatter plots of estimate vs ground truth return for each baseline on each task in DOPE D4RL (part 2).

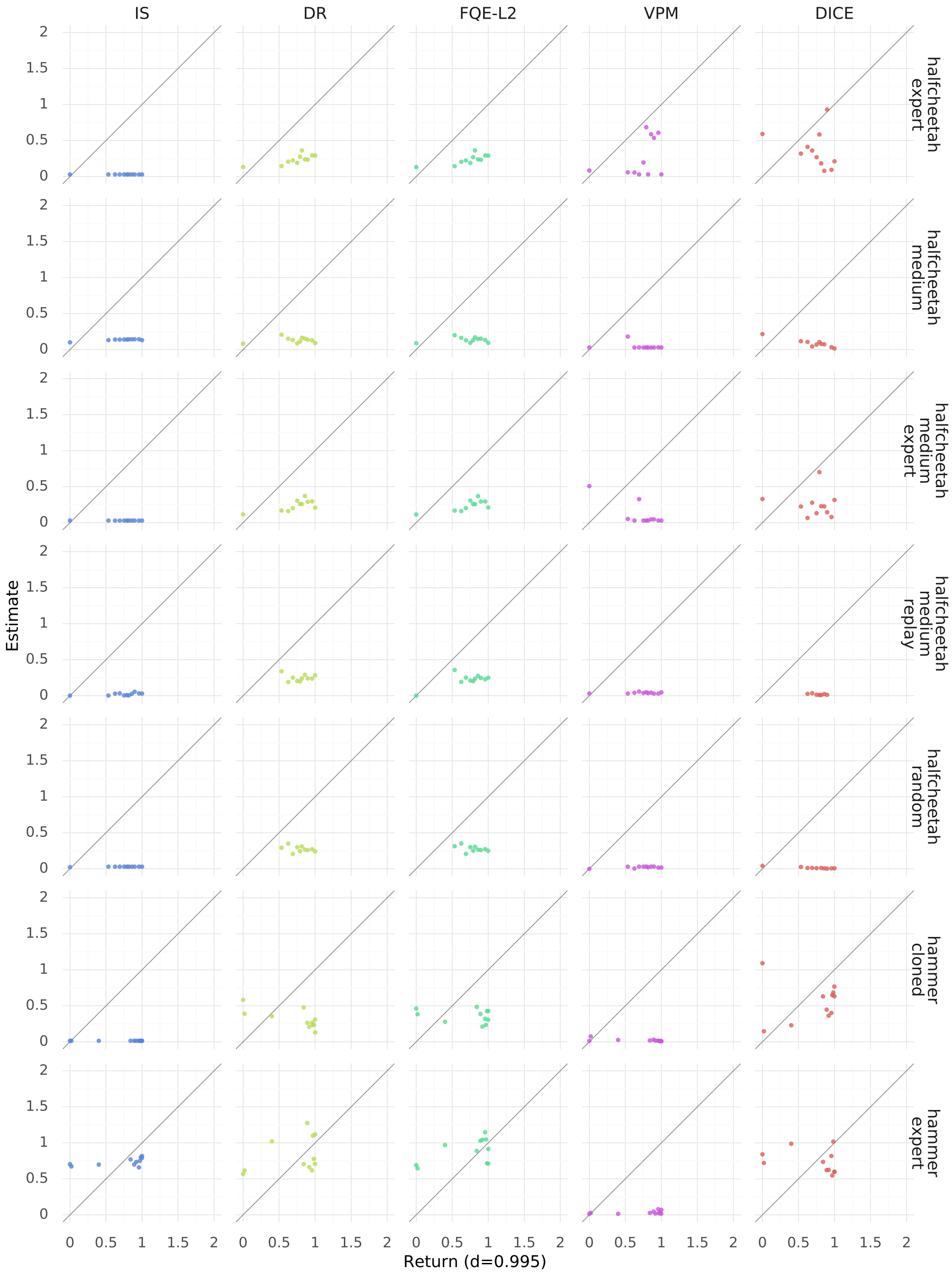


Figure A.11: Scatter plots of estimate vs ground truth return for each baseline on each task in DOPE D4RL (part 3).

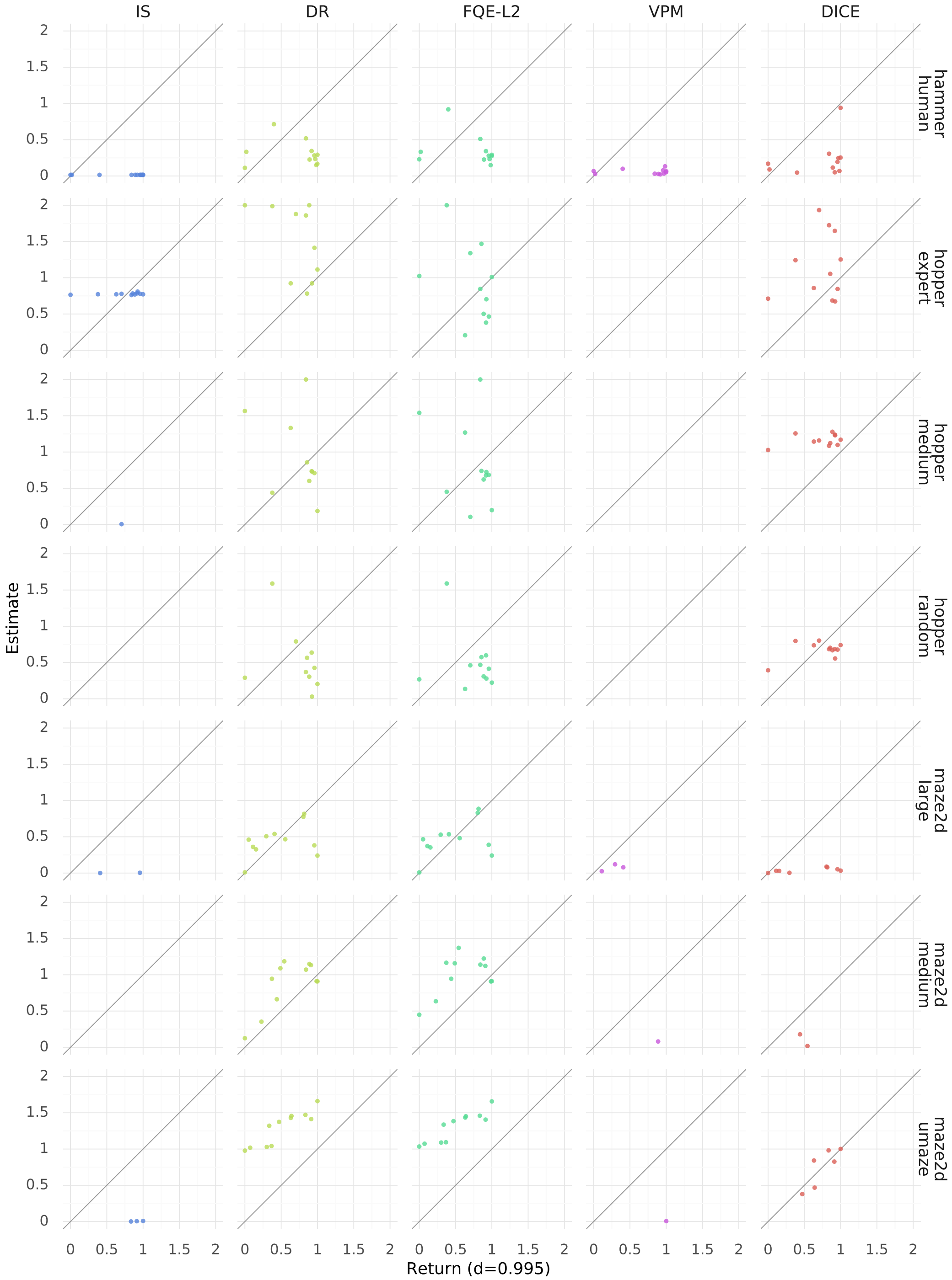


Figure A.12: Scatter plots of estimate vs ground truth return for each baseline on each task in DOPE D4RL (part 4).

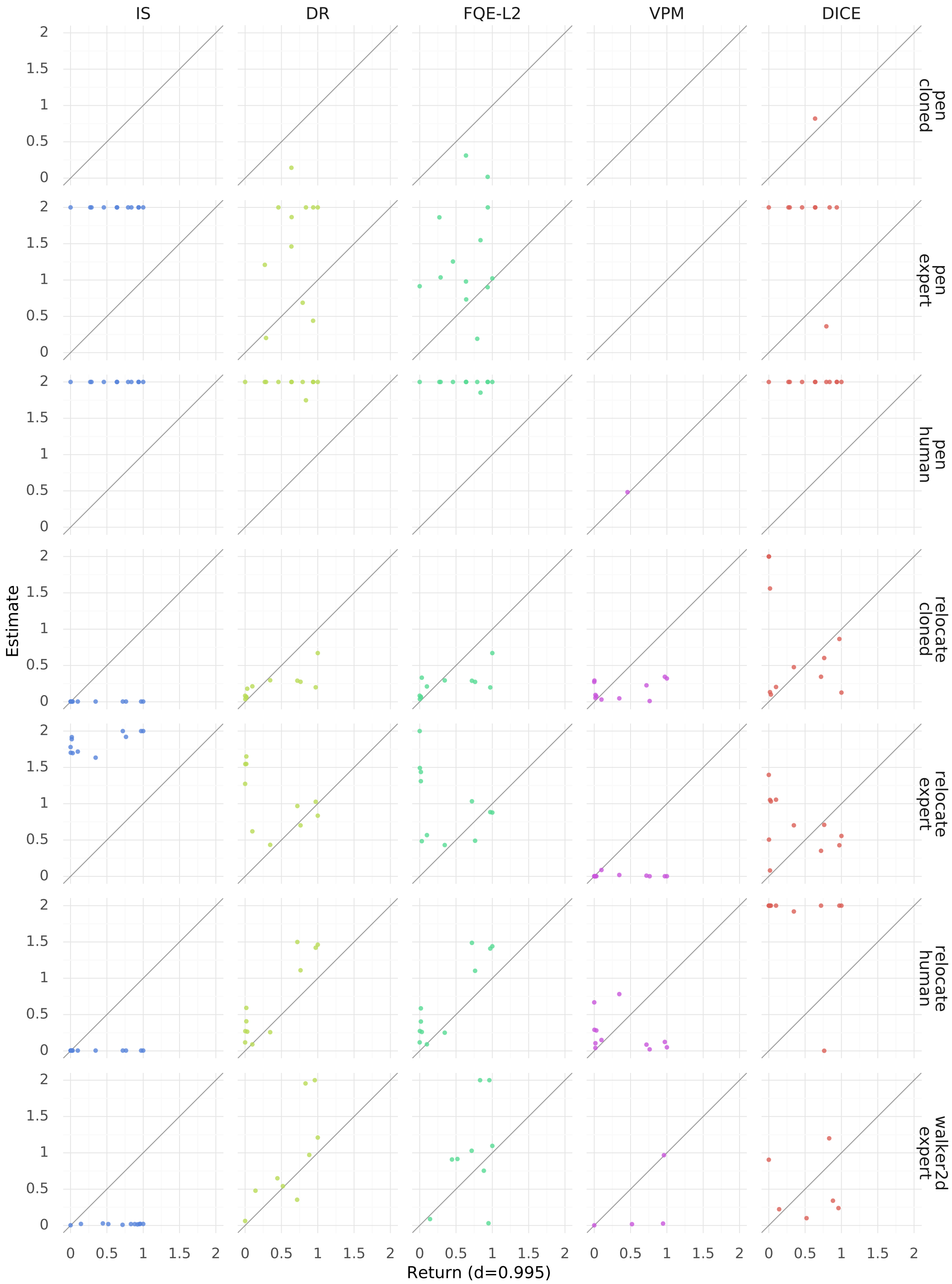


Figure A.13: Scatter plots of estimate vs ground truth return for each baseline on each task in DOPE D4RL (part 5).

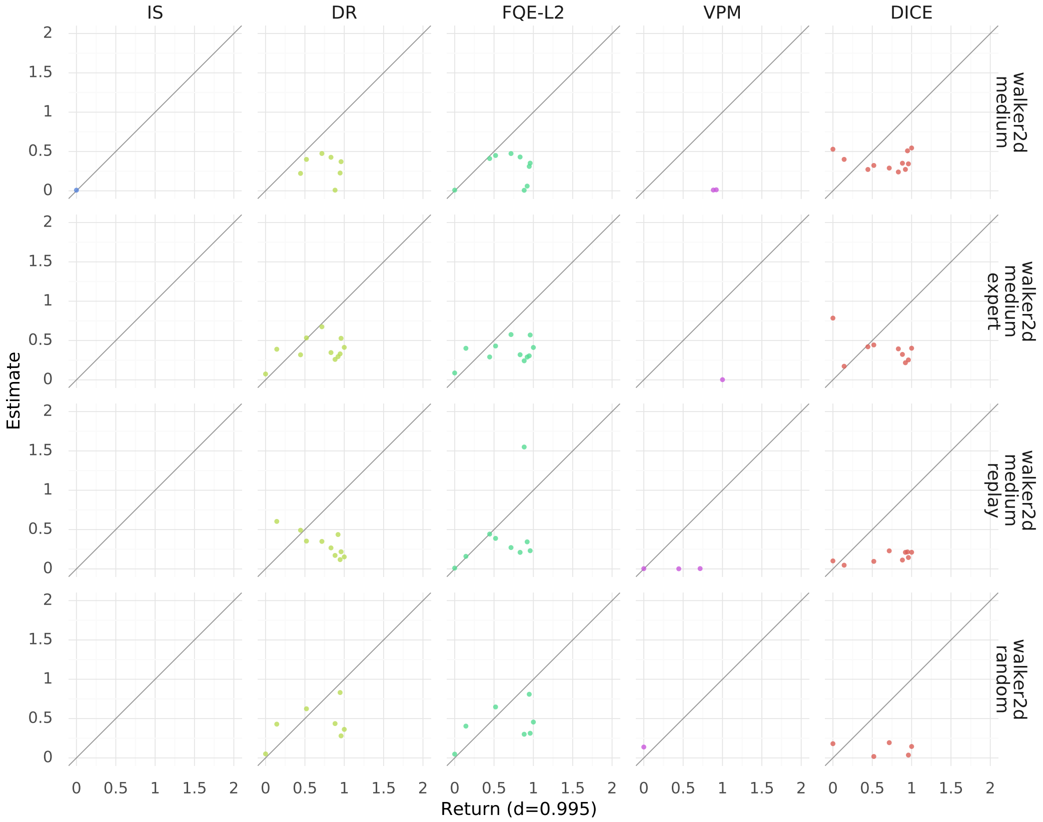


Figure A.14: Scatter plots of estimate vs ground truth return for each baseline on each task in DOPE D4RL (part 6).