

## A IMPLEMENTATION DETAILS

**Designer Policy** To fit the designer policy, we use a gaussian with tanh squashing (Haarnoja et al. (2019)) where we scale the output to be within a predefined interval for each parameter. We train the designer using a similar objective as in Soft Actor-Critic (SAC) (Haarnoja et al. (2019)) with a fixed  $\alpha$ . We use a Normal distribution where the mean is the same as the mean of the designer policy and the variance is fixed at the beginning, as the prior. We search  $\alpha$  from the set  $\{0.1, 0.5, 1.0\}$ , log-standard-deviation of the prior from the set  $\{0.1, 0.25\}$ , and  $\epsilon$  from the set  $\{0.01, 0.05\}$ . We train the designer using the Adam optimizer for  $10^5$  steps with a learning rate of  $3 \cdot 10^{-5}$ .

**Discriminator** We train the discriminator using the Adam optimizer with a learning rate of  $3 \cdot 10^{-5}$ . Instead of using the exact log-ratio as the reward for the designer, we use  $-\log(1 - D(s, a, s'))$  with the next state is passed on the input.

**Online Policy** We use SAC (Haarnoja et al. (2019)) as the online agent. We use default hyperparameters without tuning.

**Offline Policy** We use ValueDICE (Kostrikov et al. (2020)) and Behavioral Cloning (BC) for ValueDICE and D4RL datasets, respectively. We use default parameters for the ValueDICE without tuning. Similar to (Kostrikov et al. (2020)), we use replay regularization where a replay of experience collected in the simulator is used to regularize the policy. However, since we both design and fit an offline policy simultaneously, the replay in our setting consists of experience collected from different simulators with possibly different MDPs. Our experimental results show success under this new setting. For MiniGrid experiments, we use a 3-layer Convolutional Neural Network (CNN) with ReLU activations and BC to learn a behavior policy.

## B GROUND TRUTH DESIGN PARAMETERS FOR MUJOCO

In Table 2, we give a list of ground truth parameters for Mujoco environments that we experimented with. Not that these parameters are shared across different offline datasets such as ValueDICE or D4RL demonstrations.

	HalfCheetah	Hopper	Walker2D	Ant
<i>geom-friction</i>	0.4	2.0	1.9	1.0
<i>actuator-gainprm</i>	1.0	1.0	1.0	1.0
<i>gravity</i>	-9.81	-9.81	-9.81	-9.81

Table 2: Ground truth design parameters for the Mujoco environments.

## C DETAILED COMPARISON

In Figure 7, we present detailed design errors of our method on different expert levels on D4RL dataset. As expected, OTED exhibits higher error and variance on the medium-level data than others. This is not only due to the quality of the data but also the performance of the BC as it performs poorly on all medium tasks.

In Figure 8, we present detailed comparison of OTED-SAC to DR and IS on ValueDDICE demonstrations. OTED-SAC outperforms both baselines on all parameters and environments, illustrating that a misspecified simulator yields a poor performance. We observe that on Hopper and Walker2D environments with *geom-friction* parameter, baselines also achieve relatively high performance, showing the sensitivity of the environments to different parameters.

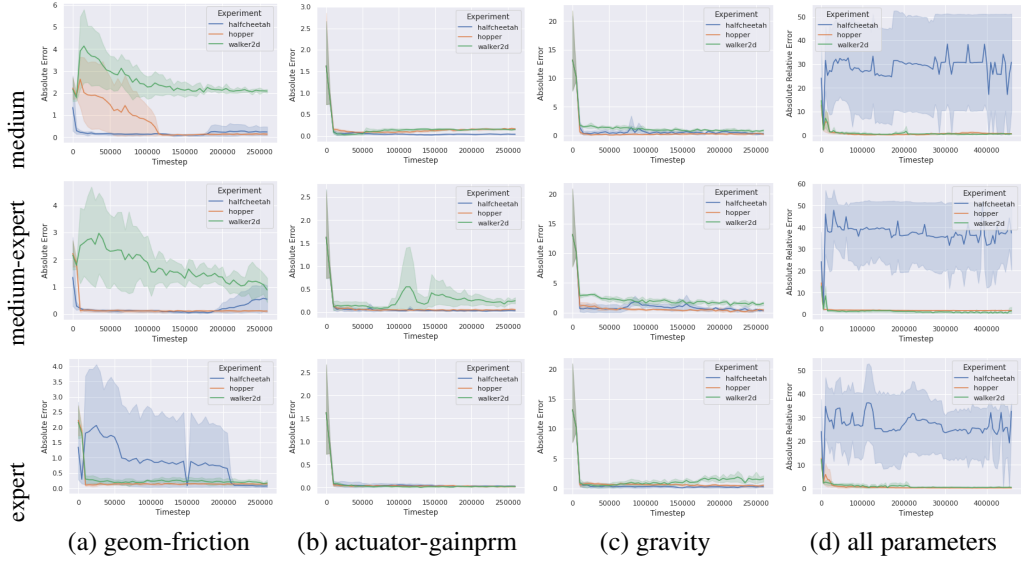


Figure 7: Absolute error of OTED on different simulator parameters using the D4RL dataset with different expert-levels. For *all* parameters, we used absolute relative error as the scale of each parameter is different.

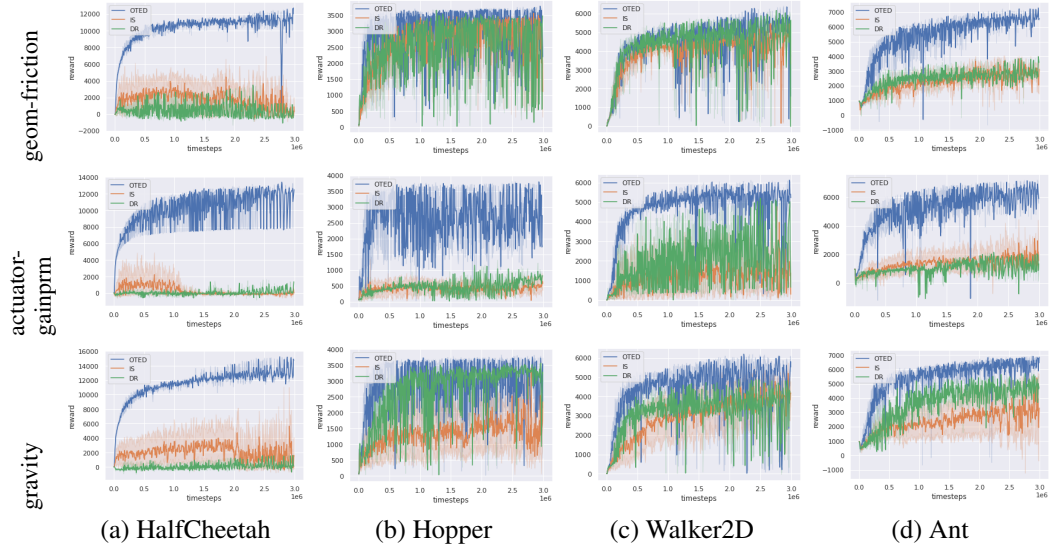


Figure 8: Detailed comparison of OTED-SAC to baseline DR and IS methods on ValueDICE dataset.