A Meta-Training Details

A.1 LPO-Zero

Our LPO-zero implementation was implemented on top of the learned_optimization library [22]. The drift function is parameterised by a one layer fully connected network with 1 hidden layer and 256 hidden units. Meta-training is done in a distributed fashion using batched, async meta-updates across 350 workers each of which with one TPUv4i accelerator. On a centralized learner process we accumulate gradients from these workers until 350 gradients are computed (using a single perturbation from an antithetic ES based gradient estimator). Once this number is reached, we perform one outer-iteration with Adam using a learning rate of 0.006.

In this experiment, we meta-train over a uniform mixture of the ant, walker2d, halfcheetah and fetch environments. We take the default hparams for PPO from Brax for each implementation except for the number of epochs trained which we set to 183 to match what was done with the ant environment. In each worker, for a particular environment, we perform a full PPO training for both a positive, and negative perturbation of the underlying meta-parameters. At the end of each training, we evaluate 10240 rollouts on the environment with the resulting policy and use these as our fitness function.

Meta-training was done over the course of 2 days and performed approximately 400 outer-updates. We find though performance still increases with increased meta-training time.

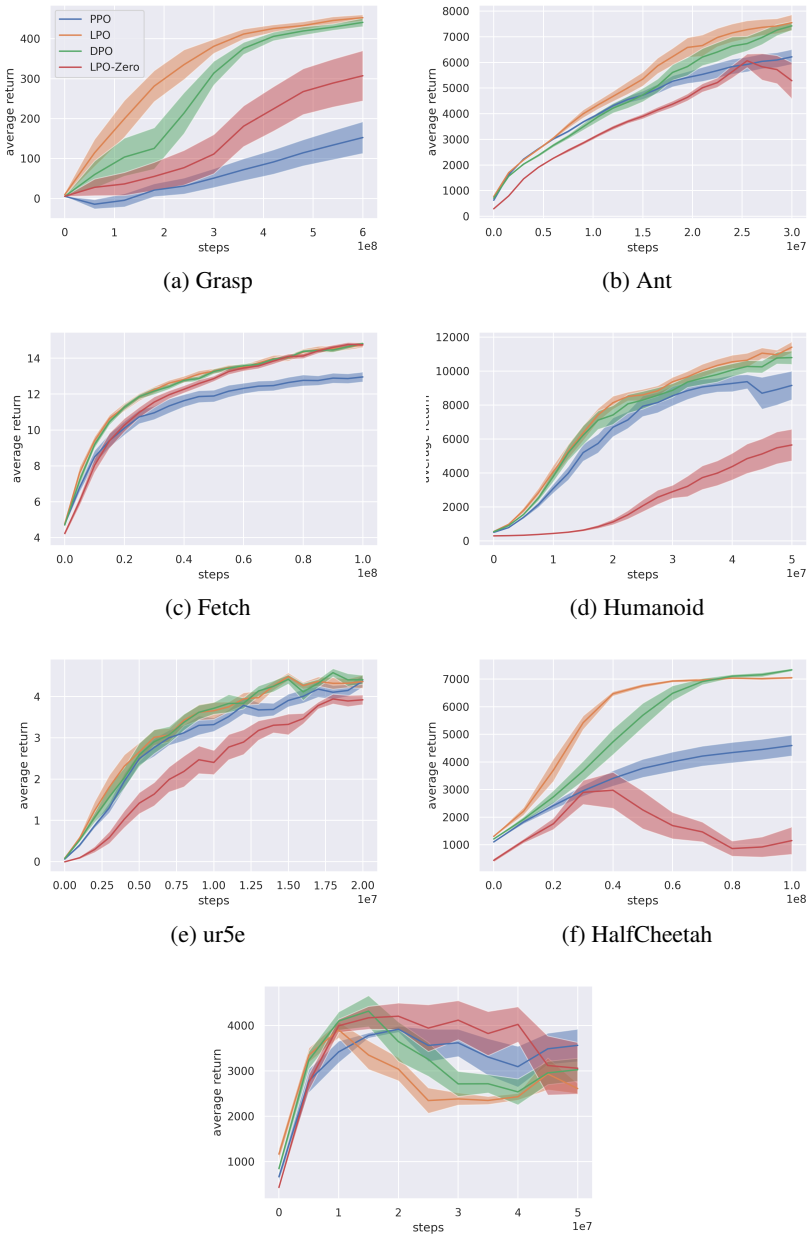
A.2 LPO

Our LPO implementation was implemented on top of the evosax library [19]. The drift function is parameterised by a one layer fully-connected network with 1 hidden layer and 128 hidden units. Meta-training was only done on a single machine with 4 V100 GPU's with synchronous updates. Meta-training was done over the course of 2 days and performed approximately 700 outer-updates. We find though performance still increases with increased meta-training time.

Table 1: Important parameters for Training LPO

Parameter	Value
Population Size	32
Number of Hidden Layers	1
Size of Hidden Layer	128
Number of Generations	672
ES Sigma Init	0.04
ES Sigma Decay	0.999
ES Sigma Limit	0.01
Number of Timesteps	30000000
Unroll Length	5
Number of Minibatches	32
Number of Update Epochs	4
Learning Rate	0.0003
Number of Environments	2048
Batch Size	1024

B LPO-Zero Results



(a) Grasp

(b) Ant

(c) Fetch

(d) Humanoid

(e) ur5e

(f) HalfCheetah

(g) Walker2d

Figure 8: Performance comparison between PPO (blue), LPO (orange), DPO (green), and LPO-Zero (red) in Brax environments. The curves represent mean evaluation return across 10 random seeds, with error bars showing standard error.

C Visualisations of LPO

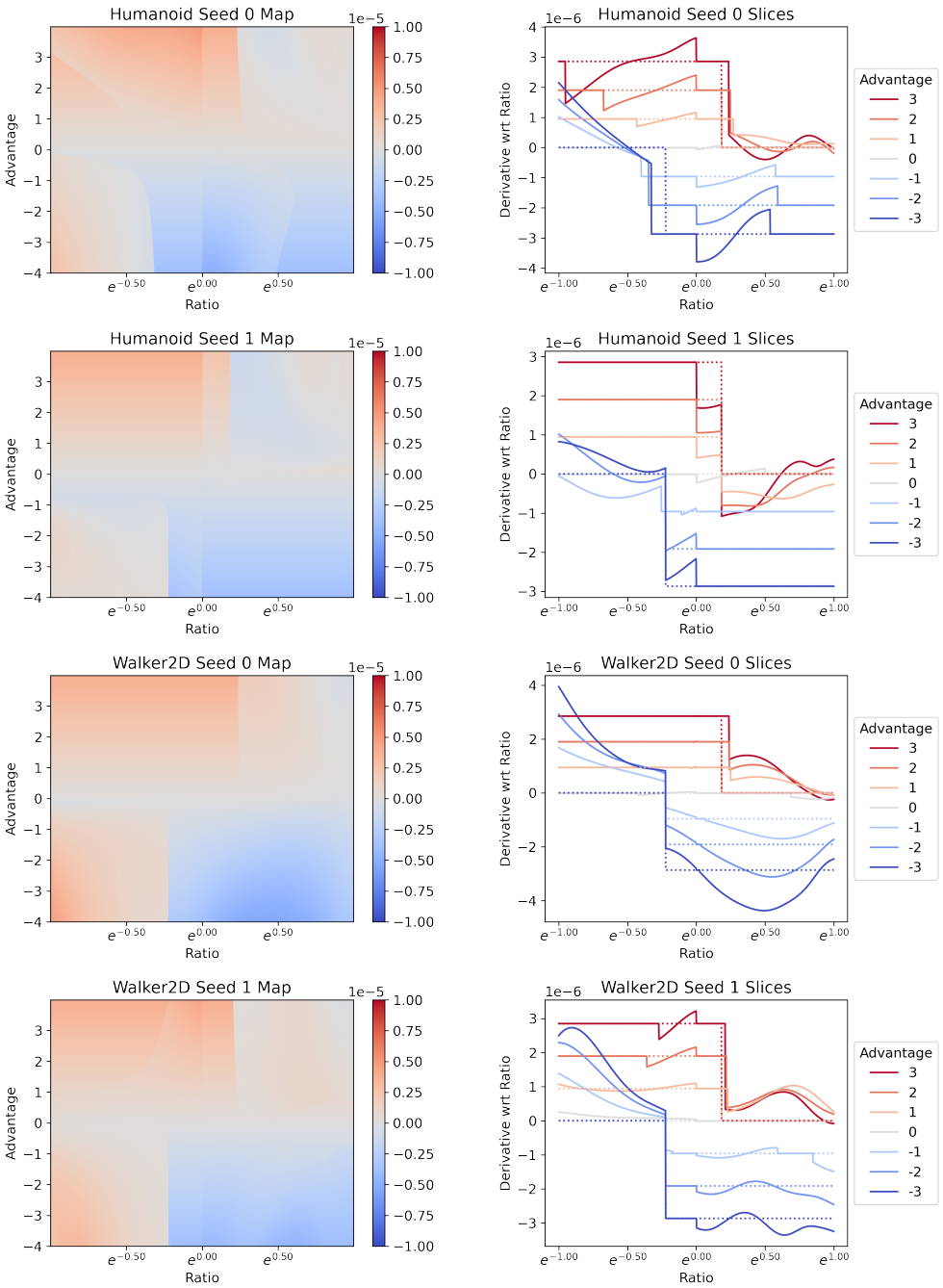


Figure 9: Visualisation of the learned objectives trained on different environments: **(a)** is the heat is the heat map of the ratio derivative of the LPO objective, and **(b)** shows its slices for fixed advantage values. Positive values of the derivative encourage updates towards the action.

D DPO Drift Verification

The DPO drift function is given by

$$f(r, A) = \begin{cases} \text{ReLU}\big((r-1)A - \alpha \tanh((r-1)A/\alpha)\big) & A \geq 0 \\ \text{ReLU}\big(\log(r)A - \beta \tanh(\log(r)A/\beta)\big) & A < 0\,. \end{cases}$$

The first condition for a valid drift is that f be non-negative everywhere, which trivially holds since $\text{ReLU}(x) \geq 0$ for all $x \in \mathbb{R}$.

The second condition is that f be zero at $\pi = \pi_{\text{old}}$. Now $r = \pi/\pi_{\text{old}} = 1$ implies $r - 1 = 0$ and $\log r = 0$, which combined with $\tanh(0) = 0$ imply that $f = 0$ as required.

The final condition is that the gradient of f with respect to π be zero at $\pi = \pi_{\text{old}}$. This is equivalent to having zero gradient with respect to $r = \pi/\pi_{\text{old}}$ at $r = 1$ since the gradients are equal up to a constant. Now writing

$$f^+ = (r-1)A - \alpha \tanh((r-1)A/\alpha) \qquad \text{and} \qquad f^- = \log(r)A - \beta \tanh(\log(r)A/\beta)$$

for $A \geq 0$ and $A < 0$ respectively, we have

$$\frac{\partial f^+}{\partial r} = A - A \cosh^{-2}((r-1)A/\alpha) \qquad \text{and} \qquad \frac{\partial f^-}{\partial r} = \frac{A}{r} - \frac{A}{r} \cosh^{-2}(\log(r)A/\beta)$$

which both evaluate to 0 at $r = 1$, since $\cosh(0) = 1$. This implies for $A \geq 0$ that

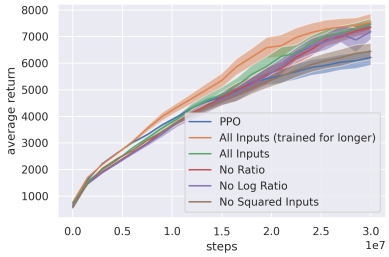
$$\frac{\partial f}{\partial r} = \frac{\partial \text{ReLU}(f^+)}{\partial r} = \begin{cases} \frac{\partial f^+}{\partial r} & \text{if } f^+ \geq 0 \\ 0 & \text{if } f^+ < 0 \end{cases} \quad = \quad 0$$

at $r = 1$ and for $A < 0$ that

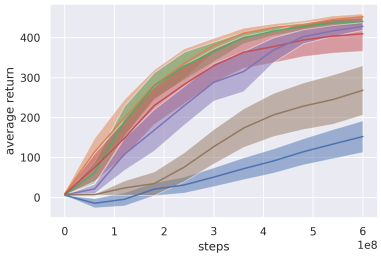
$$\frac{\partial f}{\partial r} = \frac{\partial \text{ReLU}(f^-)}{\partial r} = \begin{cases} \frac{\partial f^-}{\partial r} & \text{if } f^- \geq 0 \\ 0 & \text{if } f^- < 0 \end{cases} \quad = \quad 0$$

at $r = 1$. Taken together we conclude, for all A, that f has zero gradient at $r = 1$.

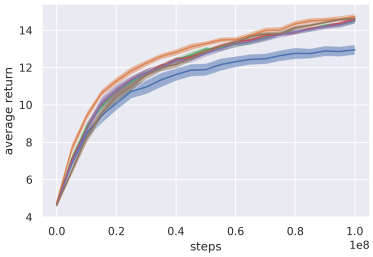
E Ablations on Drift Inputs



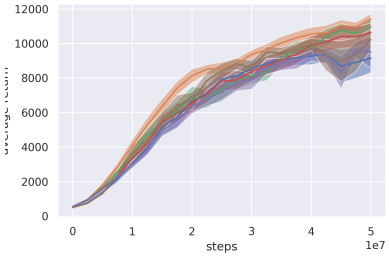
(a) Ant



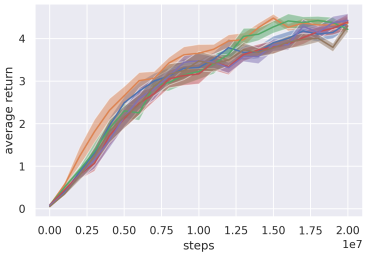
(b) Grasp



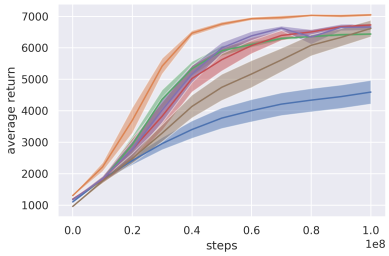
(c) Fetch



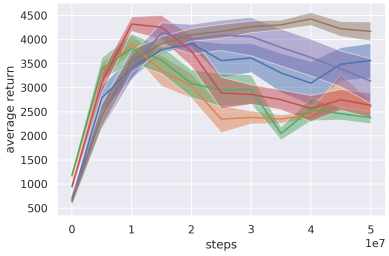
(d) Humanoid



(e) ur5e



(f) HalfCheetah



(g) Walker2d

Figure 10: Performance comparison between different inputs to the meta-training of the drift function. Note that due to computational constraints, the meta-training was only trained for 208 generations instead of 672. The results show that the meta-trained drift function performs well with respect to multiple possible input parameterisations.