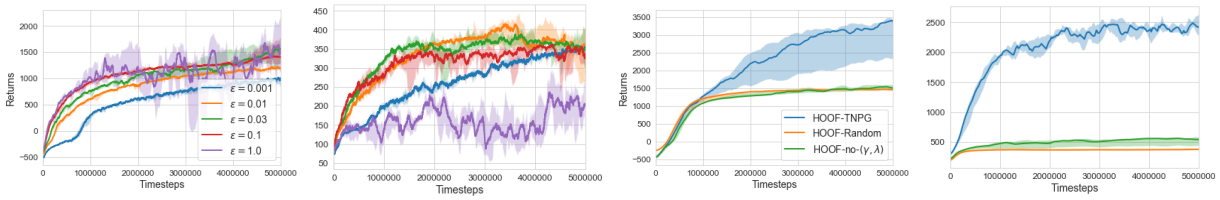


(a) HalfCheetah (b) Hopper (c) HalfCheetah (d) Hopper

Figure 1: (a) and (b) compares the performance of HOOF-A2C with different settings of the KL constraint (ϵ). Clearly $\epsilon = 0.001$ is quite conservative with slow learning while $\epsilon = 1.0$ is too aggressive. In line with existing methods that rely on KL constraints (like TRPO/NPG), we believe that [0.01, 0.1] is a reasonable range and as we have demonstrated, HOOF is robust to settings within this range. In (c) and (d) we compare performance of HOOF-TNPG with two ablations: HOOF-Random where (γ, λ) is chosen randomly (instead of argmax in Eq 4), and HOOF-no-(γ, λ) where the value function does not condition on (γ, λ). Clearly both of these are key to good performance. The performance of the latter is similar to that of HOOF-Random since the value function predictions are quite meaningless, leading to updates that are essentially random. We could not present results for Ant and Walker due to space constraints.

Reviewer 1: The ultimate goal of any hyperparameter optimisation method is to remove the need for expensive manual tuning of the hyperparameters. Our experiments demonstrate that HOOF achieves this goal and we believe this makes it a hyperparameter optimisation algorithm. The fact that it uses a zeroth order optimiser to perform a fresh hyperparameter search at each iteration of the policy gradient algorithm does not detract from its usefulness in achieving this goal.

There a couple of issues with using gradient based methods to solve Eq 4: 1) This requires that $J(\pi_{n+1})$ be differentiable wrt the hyperparameters, which might be difficult to compute or impossible, e.g. with the TRPO update, and 2) it introduces a learning rate and initialisation hyperparameter, which will require tuning thereby sacrificing sample efficiency. Thus we are restricted to zero order optimisers. We used random search to show that the simplest methods can still work well, however we could use any zero order optimiser like Bayesian Optimisation/CMA-ES.

For natural gradients like TNPG, HOOF does not add any new hyperparameters beyond those used by grid search - i.e. the range of the search space, and the number of points in the grid, and these simply express a tradeoff between compute and performance. Larger ranges and finer grids require more compute, but are likely to result in better performance, and the same applies to HOOF. Other methods like PBT introduce more hyperparameters than these.

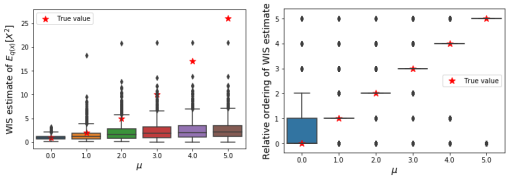
For first order methods, and only if learning the learning rate, HOOF additionally adds the KL constraint hyperparameter epsilon. We disagree that a log scale evaluation of epsilon is warranted – unlike learning rate, a search over the KL constraint (for methods like NPG/TRPO) is usually done on a linear scale. That said, we have presented the results for $\epsilon = \{0.001, 0.01, 0.03, 0.1, 1.0\}$ in Figs 1a and 1b, together with comments in the caption.

Fig 1 caption is correct, and is to show that even after taking 36x samples meta-gradients can't do better than HOOF.

It is possible that highly tuned TRPO might outperform HOOF, but at a cost of an order of magnitude more samples. If we had a budget for that many samples, instead of using WIS to estimate $J(\pi_{n+1})$ in Eq 4, we could do an on-policy evaluation and then there is no a priori reason to believe that HOOF would underperform tuned TRPO, since the noise in solution of Eq 4 due to WIS estimates goes away.

Reviewer 2: Refer to Figs 1c and 1d for comparison to the suggested random baseline. We agree that random search does not scale well with dimensionality of ψ – we could use CMA-ES or other gradient-free optimisers that scale better instead.

We demonstrate the point about relative ordering of WIS estimates empirically. Let $p(x) = N(0, 1)$ be our behaviour distribution. We are interested in $E_{q_i(x)}[X^2]$ where $q_i(x) = N(\mu_i, 1)$, $\mu_i = \{0, 1, 2, 3, 4, 5\}$. We can compute the true value analytically as $1 + \mu_i^2$. Now we compare this to a WIS estimate: we sample 10 points from $p(x)$ and use them to estimate $E_{q_i(x)}[X^2]$. We repeat this 1000 times. The boxplot of the WIS estimates in Fig 2a shows that we cannot rely on them directly as they becomes worse as $q_i(x)$ diverges from $p(x)$. However, in Fig 2b we see that the relative ordering is reliable.



(a) (b)

Figure 2: In (a) the WIS estimates of $E_{q_i(x)}[X^2]$ diverges from the true values as $q_i(x)$ diverges from $p(x)$. However (b) shows that the relative ordering based on the WIS estimates is reliable.

Reviewer 3: Please refer to Figs 1c and 1d for the effect of not learning a value function conditioned on (γ, λ) ('HOOF-no-(γ, λ)'), and Figs 1a and 1b for the effect of the KL-constraint, together with some comments.