Fast Efficient Hyperparameter Tuning for Policy Gradients

Supratik Paul, Vitaly Kurin, Shimon Whiteson

Department of Computer Science, University of Oxford
{supratik.paul,vitaly.kurin,shimon.whiteson}@cs.ox.ac.uk

The Curse of Hyperparameters in RL

- Performance is super sensitive to learning rate.
- Optimal learning rate can differ significantly between environments, policy gradient algorithms, choice of the optimiser (RMSProp, ADAM, SGD).
- Well-chosen GAE(γ, λ) can significantly speed up learning.
- New domains require a fresh hyperparameter search.

Hyperparameter Search is Inefficient

- Critical hyperparameters like learning rate are tuned while others like GAE(γ, λ) are fixed to known reasonable, but not necessarily optimal, values.
- Grid search, random search, Bayesian Optimisation, and Population-Based Training [JDO+17], all require samples from multiple training runs.
- Gradient-based methods (e.g. meta-gradients [XvHS18]) introduce their own hyperparameters which require tuning, and thus multiple training runs.
- Not viable for physical applications.

A Good Hyperparameter Search Algorithm ...

- Automatically learns hyperparameters in a single training run.
- Is robust to the setting of its own hyperparameters.
- Computationally cheap and easy to implement.

Hyperparameter Optimisation on the Fly (HOOF)

- Policy gradient update: $\pi' = \pi + f(\psi, \pi) - f$ is a step along the gradient direction for some hyperparameters ψ.
- At each iteration HOOF generates candidate policies $\{\pi'_1, \pi'_2, ..., \pi'_k\}$ for different hyperparameters $\{\psi_1, \psi_2, ..., \psi_k\}$ sampled from a given range, and sets $\pi' = \text{argmax}_i J(\pi'_i)$.
- To preserve sample efficiency $J(\pi'_i)$ is estimated using an off-policy technique like importance sampling (IS).
- More elaborate schemes of hyperparameter sampling might be applied.

But aren't IS estimates high variance?

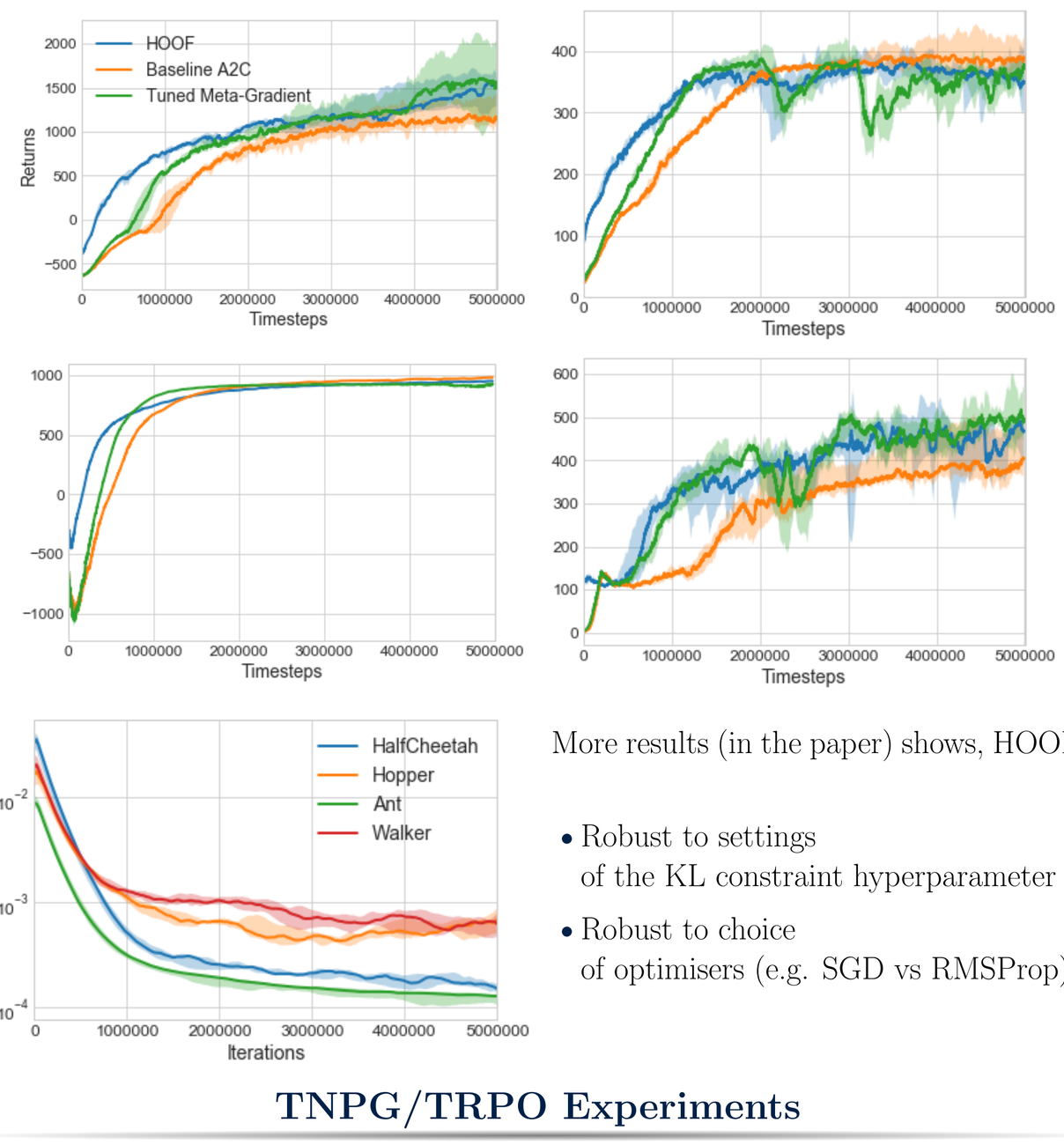
The raw estimates can have high variance, but the relative ordering of candidates (which HOOF solves for) has much lower variance.

- IS can still fail if candidate π'_i is very different from π, e.g. if the learning rate α_i is very high.
- Solution: Enforce a KL constraint between π'_i and π.
- Natural PG methods already bound the KL, so no such additional constraint required.

Acknowledgements: This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement #637713) and Samsung R&D Institute, UK. The experiments were made possible by a generous equipment grant from NVIDIA.

A2C Experiments

Focus on learning the learning rate since it is the most critical hyperparameter, and compare against OpenAI Baselines default settings and meta-gradients.

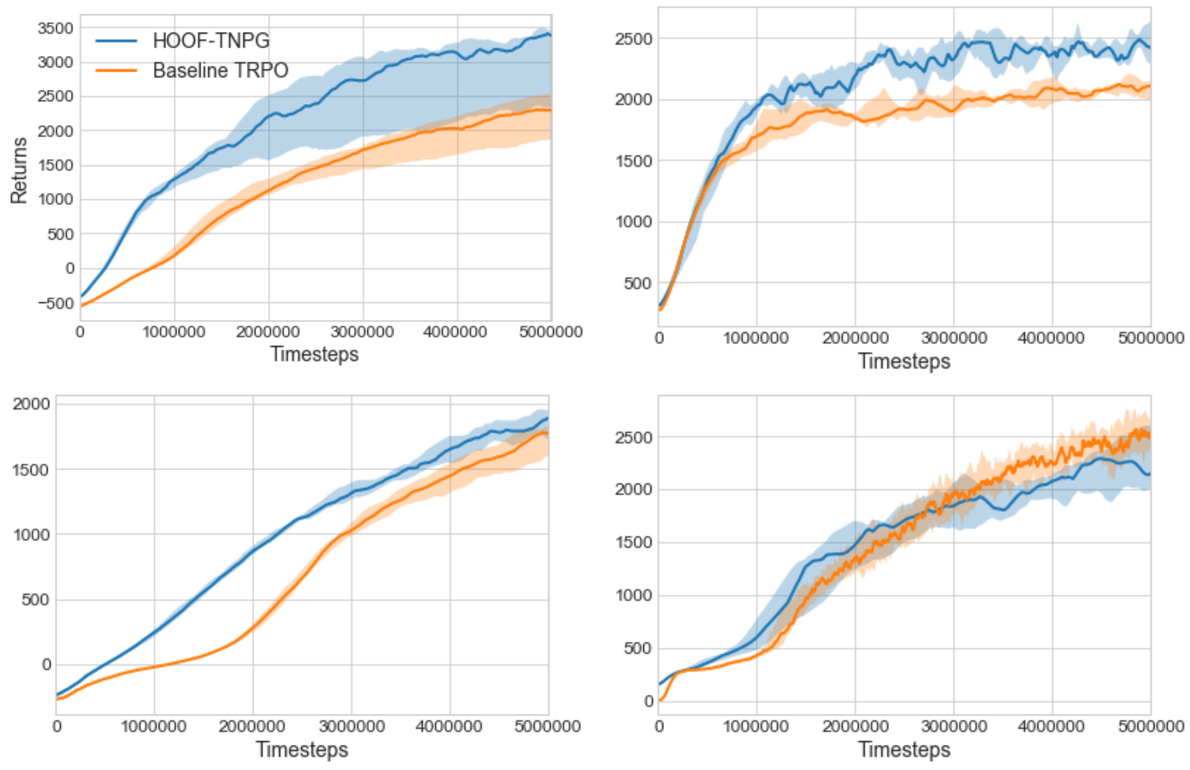


More results (in the paper) shows, HOOF is:

- Robust to settings of the KL constraint hyperparameter
- Robust to choice of optimisers (e.g. SGD vs RMSProp)

TNPG/TRPO Experiments

Use HOOF to learn KL constraint and GAE(γ, λ) of NPG vs TRPO with the commonly used default settings.



References

[JDO+17] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al., *Population based training of neural networks*, arXiv preprint arXiv:1711.09846 (2017).

[XvHS18] Zhongwen Xu, Hado P van Hasselt, and David Silver, *Meta-gradient reinforcement learning*, Advances in neural information processing systems, 2018, pp. 2396–2407.