ES-MAML: Hessian Free Meta Learning

Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, Yunhao Tang

Google Brain, Columbia University, UC Berkeley

Key Question: Can we perform meta-learning in blackbox case?

Yes! Through ES methods which perform gradients on Gaussian smoothing of the function.

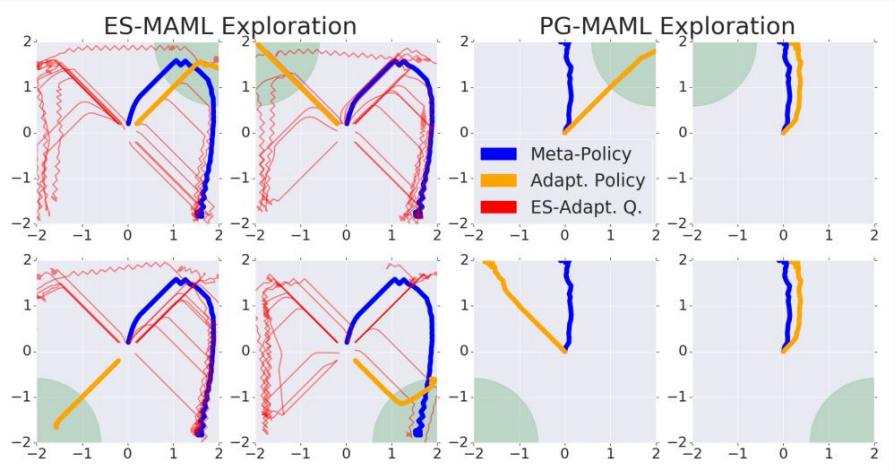
PG-MAML vs ES-MAML (Exploration)

- Single Meta-Policy generates K trajectories
- Reliance on entropy, which can be unstable - “Exploration in Action Space”

- K different policies generate rewards
- Deterministic policies allow stable exploration -“Exploration in Parameter Space”

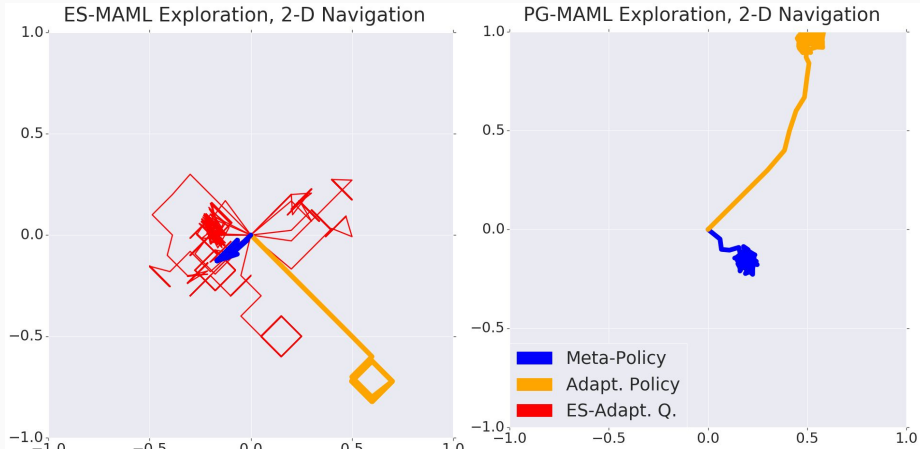
Exploration Differences

- **Four Corner Task** - agent only gets reward signal if within green radius
- ES-MAML adaptation targets only 1 or 2 Corners
- PG-MAML must “circle around” all 4 Corners



Exploration Differences

- 2D Goal Task - Agent receives distance penalty to goal point
- ES-MAML broadly explores around
- PG-MAML “Triangulates” Goal using small steps



PG-MAML vs ES-MAML (Stability)

- Policies necessarily stochastic
  - Instability/lower rewards on e.g. vanilla Swimmer/Walker (see ARS [Mania18])
- More Layers improves performance
  - See [Finn18]
- Can be unstable in low-K settings

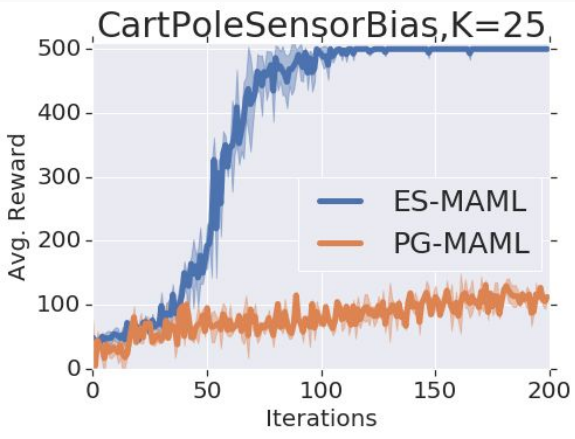
- Deterministic Policies allowed
  - Swimmer/Walker have significantly higher performance automatically
- Fewer Layers improves performance
  - Linear policies are allowed!
- Surprisingly stable in the low K = 5,10 regime
  - More realistic number of rollouts in real world robotics

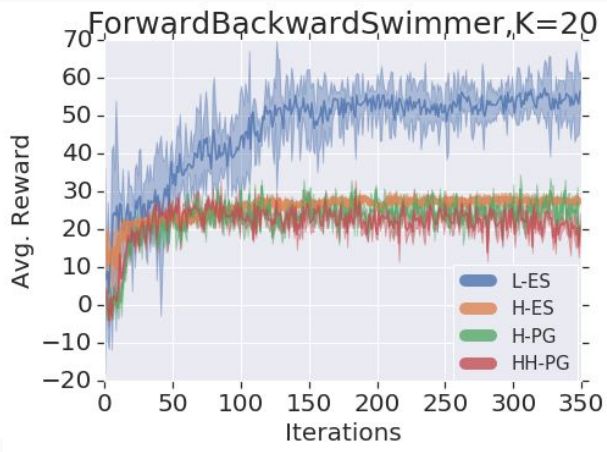
[Mania18]: Simple random search provides a competitive approach to reinforcement learning, NeurIPS 2018.

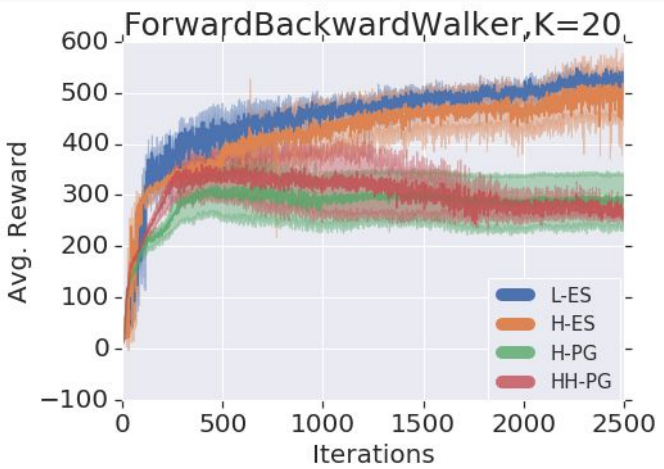
[Finn 18]: Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm, ICLR 2018.

Stability Differences

- ForwardBackwardSwimmer, ForwardBackwardWalker: high gaps
- BiasedSensorCartPole: PG-stochasticity bad for unstable environment

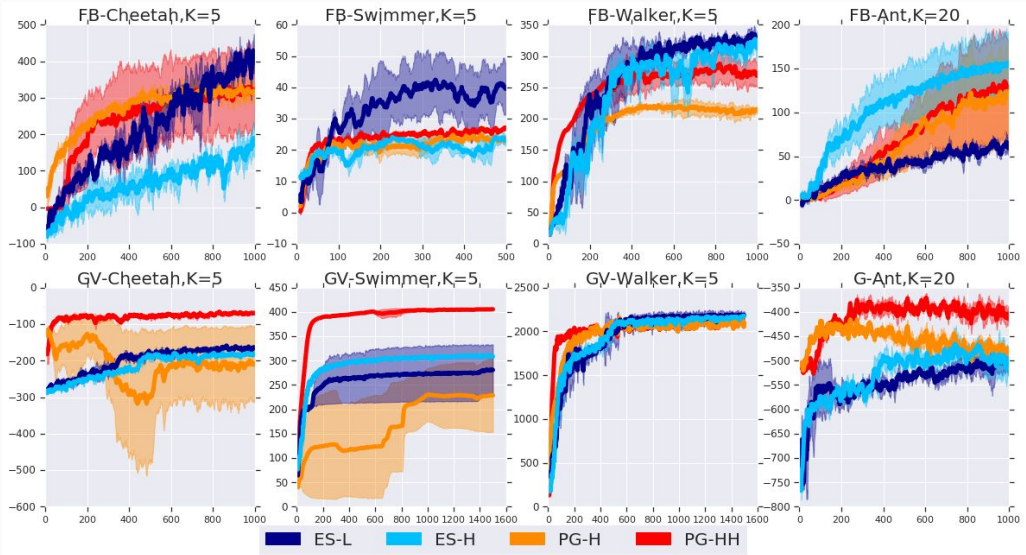






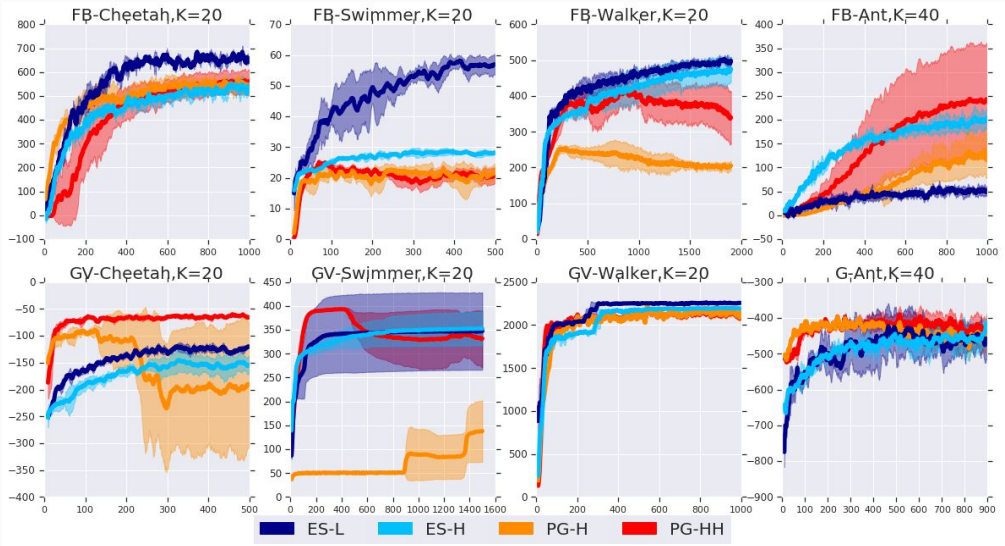
Stability Differences

- Low K benchmarking
- ES-MAML only has K scalar rewards,
  - All runs were relatively stable
- PG-MAML still has K*H state-action pairs
  - Potentially catastrophic runs (High variance across trajectories)



Stability Differences

- Normal K benchmarking
- In general, Linear policies perform better than Hidden Layers for ES-MAML



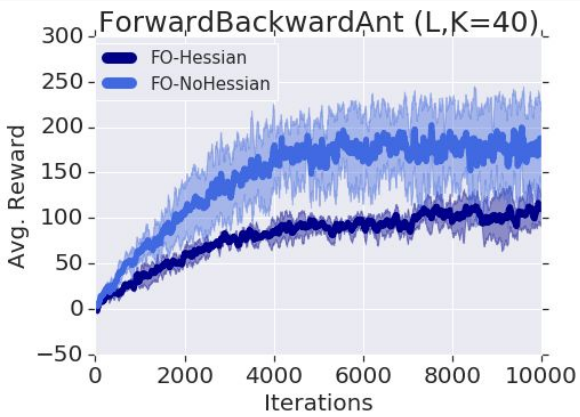
PG-MAML vs ES-MAML (Algorithmic)

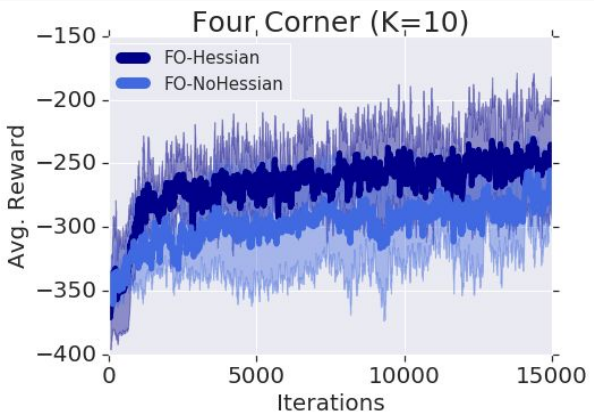
- Hessian Estimation
  - Quite complicated, high variance, estimator bias (LVC)
- Multiple Hyperparameters involved
  - e.g. TRPO-MAML: batchsize, learning rate, entropy, value-function LR, lambda ...
- Variance Reduction mainly relies on Hessian

- Hessian Estimation in ES actually does not improve performance very much
  - Very Flexible in Adaptation Operators
    - Ex: HillClimbing
- Very little hyperparameter Tuning
  - Learning Rate, Sigma
- Various ES Variance Reduction (Orthogonal sampling, Antithetic sampling, etc.)

Algorithmic Differences

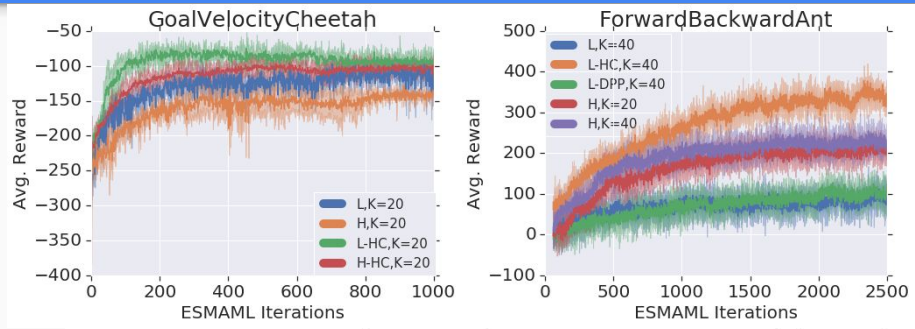
- Hessian does *not* improve ES-MAML much.
  - Slightly improves Exploration
  - Poor for FBAnt

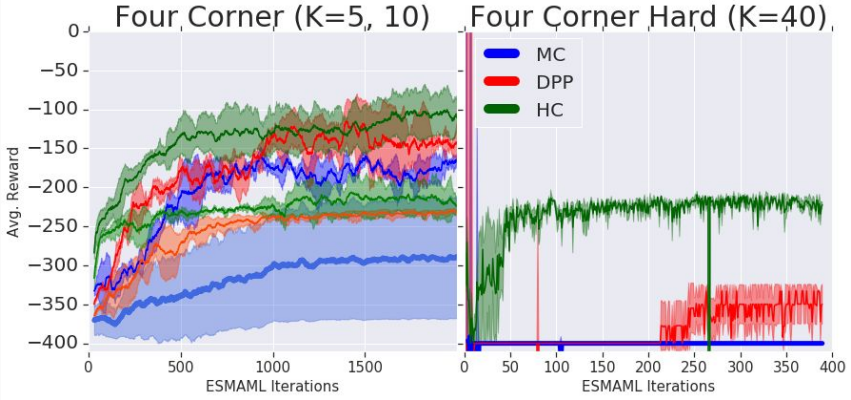




Algorithmic Differences

- Alternative to Hessian: Different Adaptation Operators!
  - HillClimbing was best
    - Enforces Monotonic improvement
    - Non-differentiable, can't easily be implemented in PG
    - Improves exploration and overall performance
  - Others: DPP





Conclusion

- ES-MAML:
  - Does not require second derivatives
  - Conceptually simpler than PG.
  - Flexible with different adaptation operators.
  - Deterministic and linear policies allows safer adaptation

Thank you!