ES-MAML: Hessian Free Meta Learning

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

Google Brain, Columbia University, UC Berkeley

Background: MAML

Model Agnostic Meta-Learning (MAML) [Finn17] seeks to find a *meta-policy* θ_{meta} to the following optimization problem: $\theta_{meta} = \min_{\theta} \mathbb{E}_{\mathcal{T}} [\mathcal{L}_{\theta}(\theta - \alpha \nabla \mathcal{L}_{\mathcal{T}}(\theta))]$

For RL, becomes a maximization problem: $\max_{\theta} J(\theta) := \mathbb{E}_{T} \mathbb{E}_{\tau' \sim \mathcal{P}_{T}(\tau' | \theta')} [R_{T}(\tau')]$.

Policy Gradient: $\nabla_{\theta} J(\theta) = \mathbb{E}_{T} \mathbb{E}_{\tau' \sim \mathcal{P}_{T}(\tau' | \theta')} [\nabla_{\theta'} \log \mathcal{P}_{T}(\tau' | \theta') R_{T}(\tau') \nabla_{\theta} U_{T}(\theta)]$

Adaptation Operator: $\theta' = U_{T}(\theta) = \theta + \alpha \nabla_{\theta} \mathbb{E}_{\tau \sim \mathcal{P}_{T}(\tau | \theta)} [R_{T}(\tau)]$

Difficulties in RL setting

- Policy Gradient (PG-MAML) already requires challenging second order estimation:

$$\nabla_\theta U = I + \alpha \int \mathcal{P}_T(\tau|\theta) \nabla_\theta^2 \log \pi_\theta(\tau) R_T(\tau) d\tau$$
$$+ \alpha \int \mathcal{P}_T(\tau|\theta) \nabla_\theta \log \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau)^T R_T(\tau) d\tau.$$

- Not used in original paper [Finn17]
- ProMP [Rothfuss19], T-MAML [Liu19], Other methods [Antoniou19]
- Multiple Hyperparameters involved
  - e.g. TRPO-MAML: batchsize, learning rate, entropy, value-function LR, lambda ...

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

Yes! Through ES methods which perform gradients on Gaussian smoothing of a function: $\widetilde{f}(x) := \mathbb{E}_g[f(x+\sigma g)]$ where $g \sim N(0, I)$

Gradient: $\nabla \widetilde{f}(x) = \frac{1}{\sigma}\mathbb{E}_g[f(x + \sigma g)g]$

ES: Estimate gradient and apply stochastic first-order method.

Very little hyperparameter Tuning (Learning Rate, Sigma)

Also has numerous ways to reduce variance (Orthogonal sampling, DPP sampling, etc.)

ES-MAML

ES Meta-Objective: $\max_{\theta} J(\theta) := \mathbb{E}_{T}[R_{T}(U_{T}(\theta))]$

Zero-Order ES-MAML: $\nabla \widetilde{J}(\theta) = \frac{1}{\sigma}\mathbb{E}_{T,g}[R_{T}(U_{T}(\theta + \sigma g))g]$

- No need for second order estimation! (Hessian-Free)
- Can use *non-smooth* adaptation operators, such as Hill-Climbing.

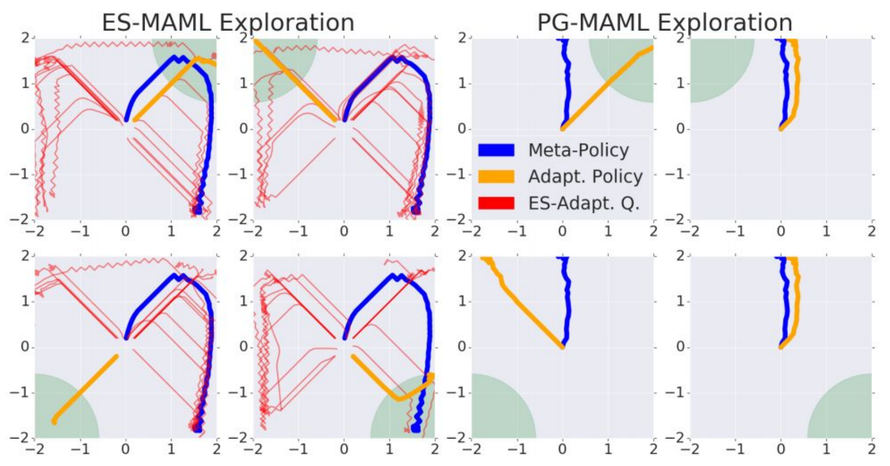
PG-MAML vs ES-MAML (Exploration)

- 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”

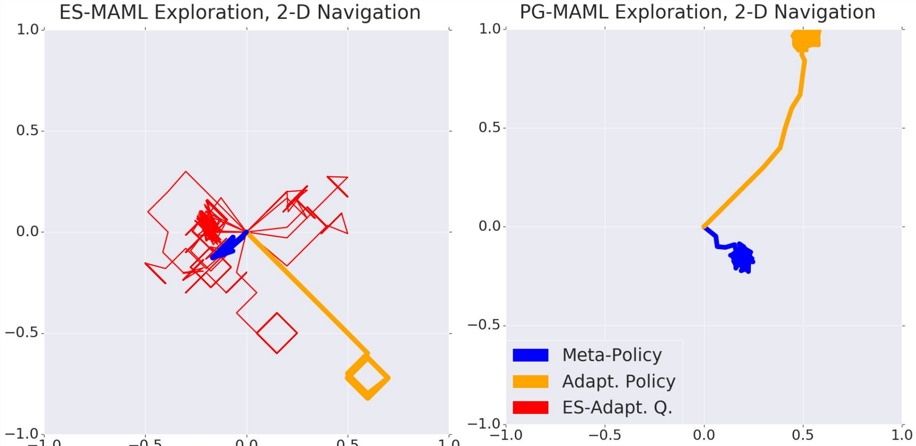
Exploration Differences

- **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



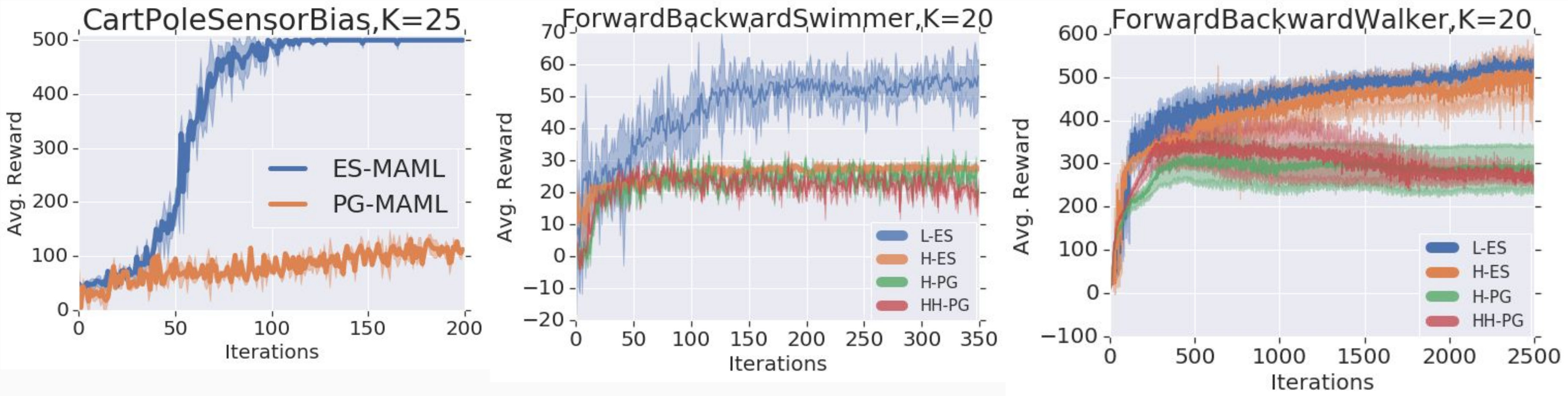
PG-MAML vs ES-MAML (Stability)

- 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

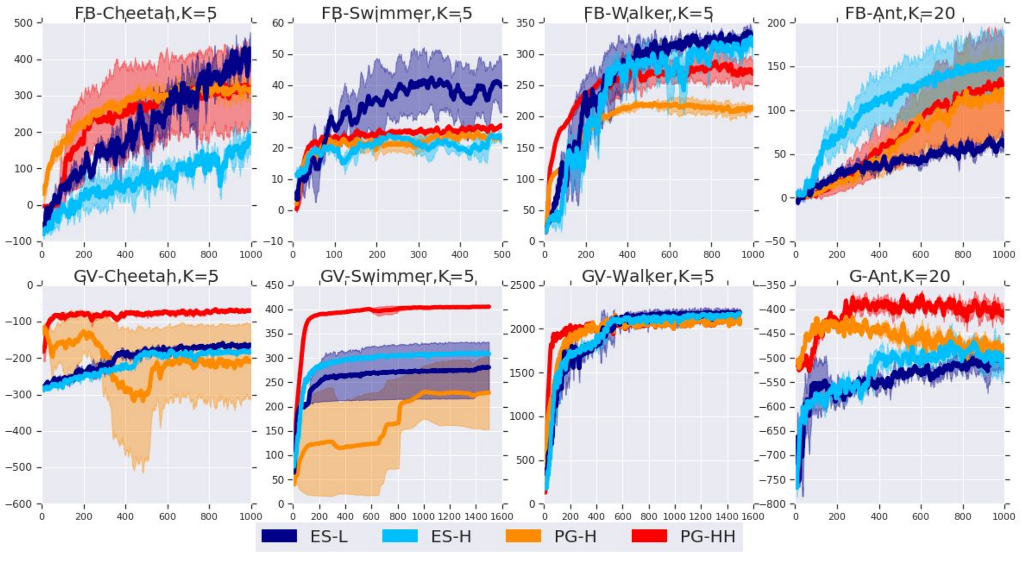
Stability Differences

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



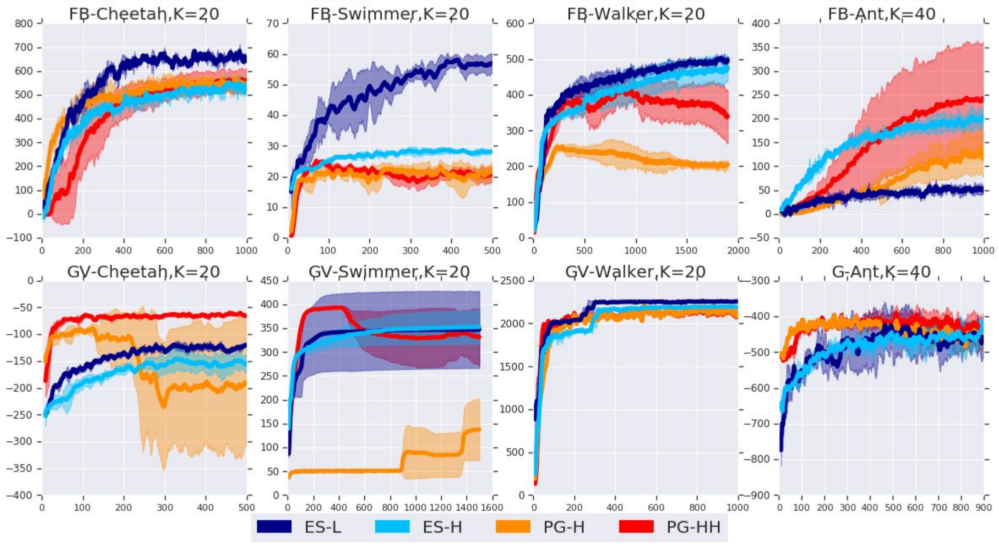
Stability Differences

- 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)



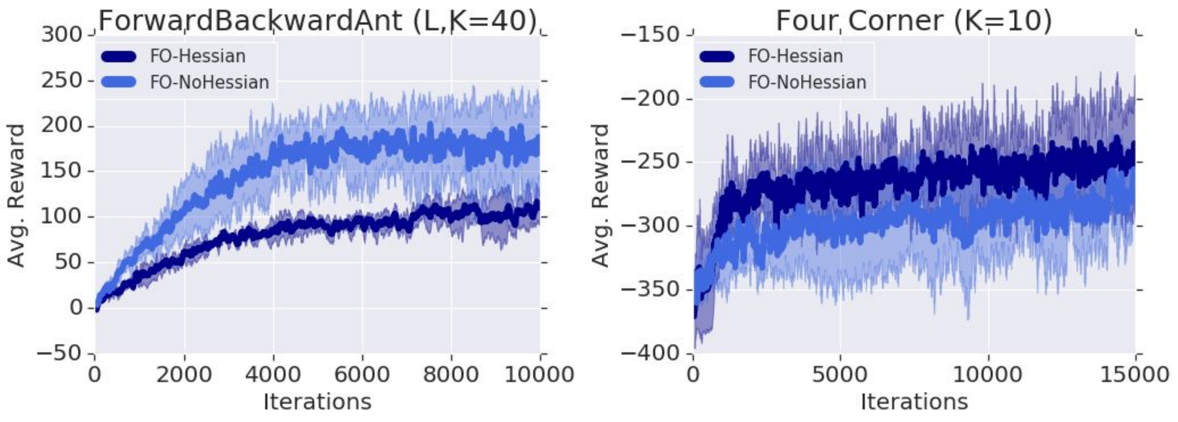
Stability Differences

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



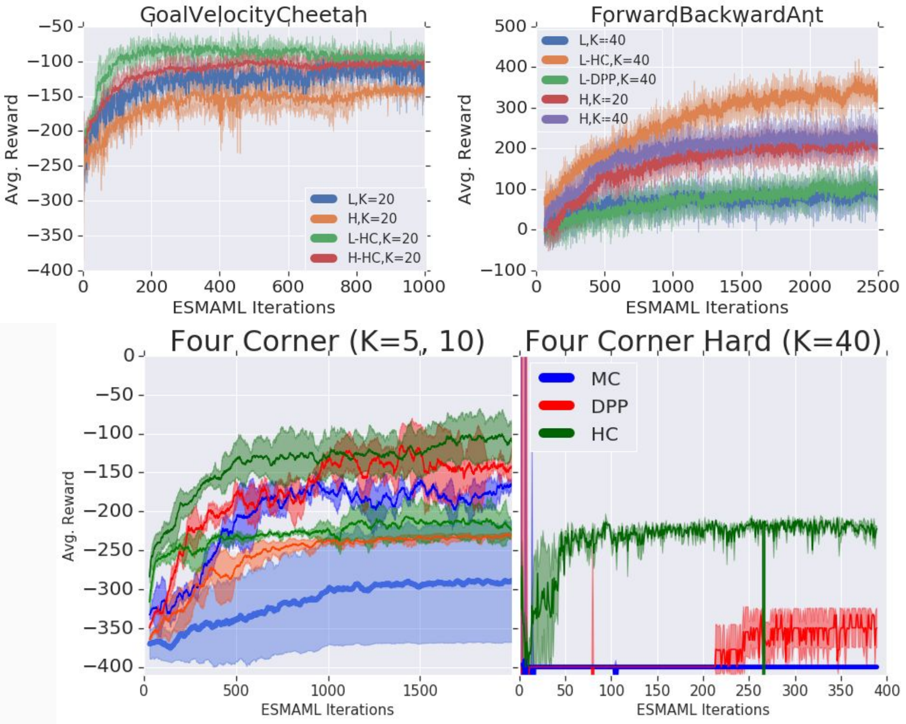
Algorithmic Differences

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



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!

Bibliography

- [Finn17] - Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML 2017
- [Finn18] - Chelsea Finn, Sergey Levine. Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm. ICLR 2018.
- [Rothfuss19] - Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour,and Pieter Abbeel. ProMP: Proximal Meta-Policy Search. ICLR 2019
- [Liu19] - Taming MAML: Efficient Unbiased Meta-Reinforcement Learning. ICML 2019
- [Antoniou19] - Antreas Antoniou, Harrison Edwards, and Amos J. Storkey How to Train your MAML. ICLR 2019.