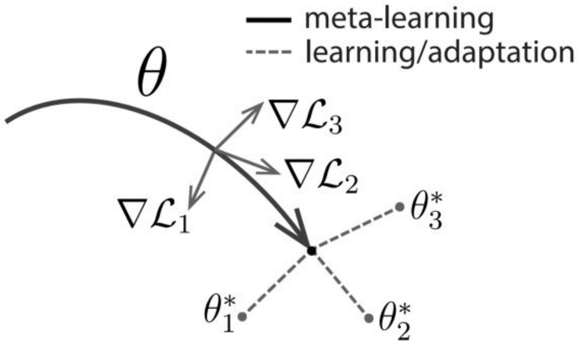
DeepMind

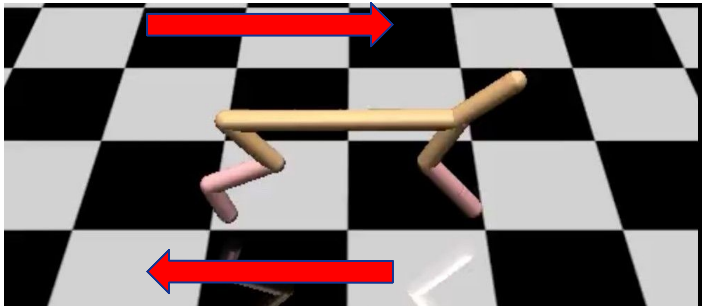
# Biased Gradient Estimate With Drastic Variance Reduction For Meta Reinforcement Learning

Yunhao Tang

# Summary of results

- Meta-RL carries out fast adaptation when agent faces a new environment



What we found in a nutshell

- Unbiased meta-gradient estimates have huge variance – O(N)
  - N – number of inner loop samples
- Biased gradient estimate has better trade-off – bias O(1/N) and variance O(1/N)
  - Sit between score-function estimate and “golden-rule” path-wise estimate

# Meta-RL optimization problem with one-step adaptation

- Basic notations
  - Policy parameter $\theta \in \Theta$
  - Value function $V(\theta)$
  - Trajectory $(\tau_i)_{i=1}^N \sim p_\theta$
  - Trajectory return $R(\tau_i)$

$$\underbrace{R(\tau_i)\nabla_\theta \log p_\theta(\tau_i)}$$

One-sample estimate to PG

$$\max_\theta \mathbb{E}_{(\tau_i)_{i=1}^N \sim p_\theta}\left[V\left(\theta + \underbrace{\eta \frac{1}{N}\sum_{i=1}^N R(\tau_i)\nabla_\theta \log p_\theta(\tau_i)}\right)\right]$$

Outer loop evaluation:

Post-adaptation performance

Inner loop update:

One-step N-sample PG estimate

# N-sample Monte-Carlo objective

- Classic Monte-Carlo objective

$$L(\theta) = \mathbb{E}_{X \sim p_\theta}\left[f(X)\right]$$

- N-sample Monte-Carlo objective

$$L_N(\theta) = \mathbb{E}_{(X_i)_{i=1}^N \sim p_\theta}\left[f\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right)\right]$$

$$\neq \frac{1}{N}\sum_{i=1}^{N} f(X_i)$$

- One-step meta-RL is a special instance
  - Slight modifications – see paper

# Unbiased stochastic gradient estimate

- Unbiased gradient estimate → score-function (SF) estimate has high variance

$$f\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right)\underbrace{\sum_{i=1}^{N} \nabla_\theta \log p_\theta(X_i),}_{\text{"Sum" not "average"}} \quad (X_i)_{i=1}^{N} \sim p_\theta$$

"Sum" not "average"

- Can construct examples where the variance is O(N)
    - In practice (toy example and deep RL), SF exhibits high variance too

# Deriving biased gradient estimate

- Limiting behavior of the estimate

$$f\Big(\underbrace{\frac{1}{N}\sum_{i=1}^{N} X_i}_{\to\ \mu_\theta}\Big)\sum_{i=1}^{N}\nabla_\theta \log p_\theta(X_i), \quad (X_i)_{i=1}^{N} \sim p_\theta$$

- Introduce an "unknown" control variate

$$\left(f\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right) - f\left(\mu_\theta\right)\right)\sum_{i=1}^{N}\nabla_\theta \log p_\theta(X_i)$$

- To bypass the "unknown" control variate
  - First-order Taylor expansion → make use of gradient information → create bias reduces variance
  - Linearized score-function (LSF)

# Biased estimate with drastic variance reduction

- Final LSF estimate

"Average" not "sum"

$$\frac{1}{N}\sum_{i=1}^{N}\left[\nabla f\left(\frac{1}{N}\sum_{i=1}^{N}X_i\right)\right]^T X_i \cdot \nabla_\theta \log p_\theta(X_i)$$

Make use of gradient information

- Theoretical properties: bias O(1/N) and variance O(1/N)
  - See paper for formal statements
- An interpolation of "SF estimate" and "path-wise estimate (PW estimate)"
  - "Blackbox" – compatible with RL and meta-RL, similar to SF estimate
  - "Low variance" – make use of gradient information – similar to PW estimate

# Back to meta-RL

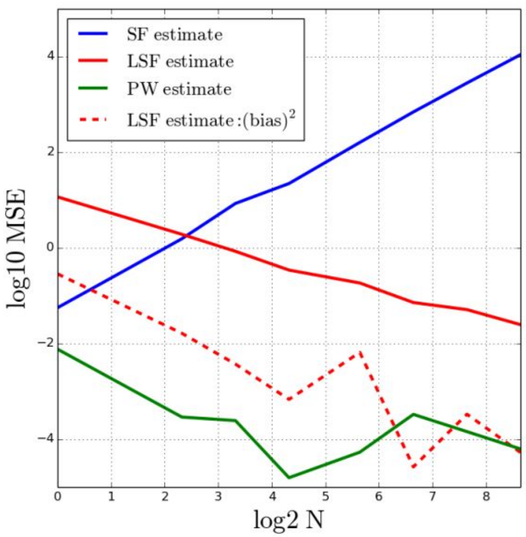
- N-sample MC objective recovers one-step meta-RL as a special case
  - Derive the corresponding LSF:

$$\underbrace{\left(I + \eta \hat{H}_\theta\right)}_{\text{Inner loop Hessian}} \cdot \underbrace{\nabla \hat{V}\left(\theta + \eta \frac{1}{N} \sum_{i=1}^{N} R(\tau_i) \nabla_\theta \log p_\theta(\tau_i)\right)}_{\text{Outer loop gradient}}$$

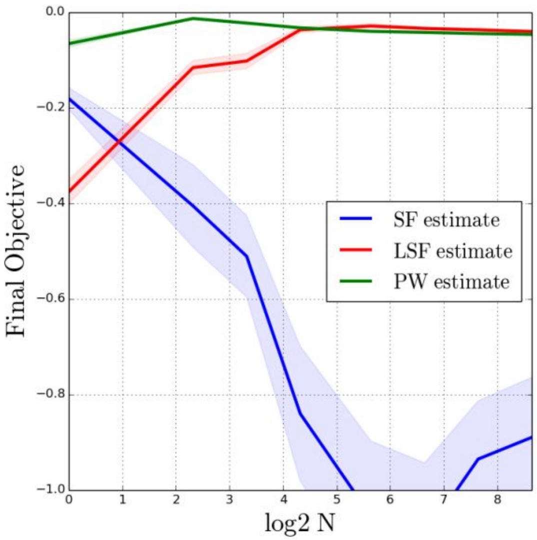
- Convergence guarantees have nicer dependencies on N, compared to SF (Fallah et al,  2020)

# Experiments: 1-D toy example

$$\max_{\theta} L_N(\theta) = \mathbb{E}_{(X_i)_{i=1}^N \sim p_\theta}\left[-(\bar{X}_N - 1)^2\right]$$

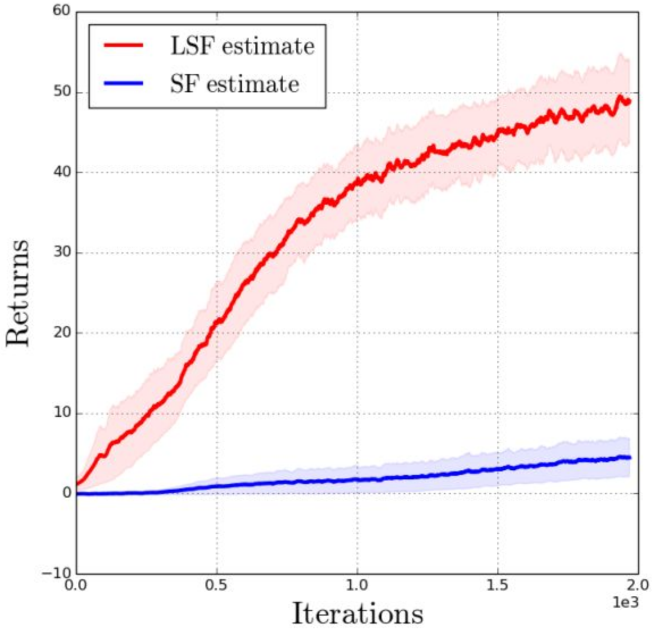



(a) Bias-variance trade-off

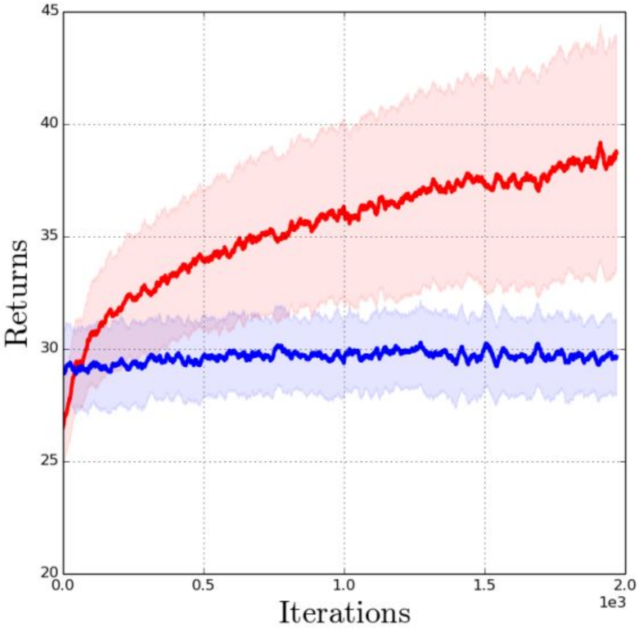



(b) 1-D Optimization

# Experiments: deep RL




(c) Meta-RL HalfCheetah




(d) Meta-RL Walker2D

# Summary

- Most prior work on “unbiased meta-gradient estimates for RL” are in fact biased.
- Some meta-gradient estimates are biased for a very good reason – drastic variance reduction.

DeepMind

# Thank you!