On Merits of Biased Gradient Estimates for Meta Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite the empirical success of meta reinforcement learning (meta-RL), there are still a number poorly-understood discrepancies between theory and practice. Critically, biased gradient estimates are almost always implemented in practice, whereas prior theory on meta-RL only establishes convergence under unbiased gradient estimates. In this work, (1) We show that unbiased gradient estimates have variance $O(N)$ which linearly depends on the sample size N of the inner loop updates; (2) We propose linearized score function (LSF) gradient estimates, which have bias $O(1/\sqrt{N})$ and variance $O(1/N)$; (3) We show that most empirical prior work in fact implements variants of the LSF estimates; (4) We establish convergence guarantees for the LSF estimates in meta-RL, showing better dependency on N than prior work. Due to time constraints, the proof and appendix are not yet complete. They will be complete for camera-ready.

1 Introduction

By design, many reinforcement learning (RL) algorithms learn from scratch. This entails RL to achieve high profile success in a number of important and challenging applications [1–3]. However, at the same time, RL is highly inefficient compared to how humans learn, usually consuming orders of magnitude more samples to acquire skills at the same level as humans. One potential source of such inefficiencies is that unlike humans, RL algorithms do not exploit prior knowledge on the tasks at hand.

To resolve such an issue, meta-reinforcement learning (meta-RL) formalizes the learning and transfer of prior knowledge in RL [4]. On a high level, an agent interacts with a distribution of tasks at *meta-training* time. The objective is that after meta-training, the agent can learn significantly faster when faced with unseen tasks at *meta-testing* time. If an agent achieves good performance at meta-testing time, it embodies the ability to transfer knowledge from prior experiences during meta-training. There are many concrete formulations of meta-RL (see, e.g. [5–13]), Our focus is meta-RL through gradient-based adaptations [4], where the agent carries out policy gradient (PG) inner loop updates [14] at both meta-training and meta-testing time.

Motivation. Our work is motivated by a number of important discrepancies between meta-RL theory and practice. Recently, there is a growing interest in establishing performance guarantees for meta-RL algorithms with unbiased gradient estimates [15]. However, since the inception of the field, meta-RL practitioners have almost always implemented biased gradient estimates [4, 16–19]. It is natural to ask: why are unbiased gradient estimates potentially undesirable in practice, and what do we gain by introducing bias into gradient estimates?

Our focus. We focus on the *N-sample meta-RL objective* where the inner loop updates are N-sample PG estimates. In prior work, this was called the E-MAML objective [16, 17, 15], as opposed to the MAML objective [4] where the inner loop update is exact PG. This objective is of practical

Submitted to the 5th Workshop on Meta-Learning at NeurIPS 2021, held virtually. Do not distribute.

interest, because at meta-testing time, inner loop updates can only be implemented with N-sample PG estimates. See Sec 2 for details.

Contributions. We make a few contributions that bridge meta-RL theory and practice.

- **High variance of unbiased estimates.** By formulating the meta-RL objective as a generic N-sample additive Monte-Carlo objective, we show that the unbiased gradient estimates have variance on the order of $O(N)$, rendering the estimates useless when N is large (see Sec 3).
- **Novel derivation of biased estimates.** We propose the linearized score function (LSF) gradient estimate for the N-sample additive Monte-Carlo objective, which has variance $O(1/N)$ and bias $O(1/\sqrt{N})$. Its application to meta-RL enjoys better properties at large N (see Sec 4).
- **Prior work implements biased estimates.** We observe that despite their claims of unbiasedness, most prior work in fact implements variants of LSF estimates. This implies they are both biased w.r.t. the MAML and the N-sample meta-RL objective (see Sec 5).
- **Performance guarantee with better dependency at large N.** We provide performance guarantee of meta-RL algorithms with biased gradient estimates. Such guarantee contrasts with results of unbiased estimates, where the guarantee degrades significantly at large N due to high variance [15] (see Sec 6).

2 Background

2.1 Task-based reinforcement learning

Consider a Markov decision process (MDP) with state space $\mathcal{S}$ and action space $\mathcal{A}$. At time $t \geq 0$, the agent takes action $a_t \in \mathcal{A}$ in state $s_t \in \mathcal{S}$, receives a reward r_t and transitions to a next state $x_{t+1} \sim p(\cdot|s_t, a_t)$. Without loss of generality, we assume that the at $t = 0$ the agent starts at the same state. We assume the reward $r_t = r(s_t, a_t, g)$ to be a deterministic function of state-action pair (s_t, a_t) and the task variable $g \in \mathcal{G}$. The task variable $g \sim p_{\mathcal{G}}$ is sampled for every episode. A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ specifies a distribution over actions at each state. We further assume that the MDP terminates within a finite horizon of H almost surely under all policies.

Parameterized policy. In general, the policy is parameterized π_θ with parameter $\theta \in \mathbb{R}^D$,

Value function. Let $\tau \coloneqq (s_t, a_t, r_t)_{t=0}^{H-1}$ be a trajectory. The policy π_θ induces a distribution over trajectories $p_{\theta,g}(\tau) \coloneqq \Pi_{t=0}^{H-1} p(x_{t+1}|s_t, a_t)\pi_\theta(a_t|s_t, g)$. We define $R(\tau, g) \coloneqq \sum_{t=0}^{H-1} \gamma^t r_t$ as the cumulative return along trajectory τ under task g. We also define the value function as the expected returns over trajectories $V_g(\pi_\theta) \coloneqq \mathbb{E}_{\tau \sim p_\theta}[R(\tau, g)]$. We also overload the notations $V_g(\theta) \coloneqq V_g(\pi_\theta)$.

Note that unlike other work in RL, we define the value function as expected cumulative returns starting from the *initial state*. This definition will greatly simplify notations in later sections.

Policy gradient and stochastic estimates. Policy gradient (PG) [14] is the gradient of the value function with respect to policy parameter $\nabla_\theta V_g(\theta) = \mathbb{E}_{\tau \sim p_\theta}[R(\tau, g)\nabla_\theta \log p_{\theta,g}(\tau)]$. In practice, it is not feasible to compute PG exactly and it is of interest to construct stochastic PG estimates given sampled trajectories. Indeed, $\hat{\nabla}_\theta V_g(\theta) = R(\tau, g)\nabla_\theta \log p_{\theta,g}(\tau)$ with $\tau \sim p_\theta$ is an unbiased PG estimate in that $\mathbb{E}[\hat{\nabla}_\theta V_g(\theta)] = \nabla_\theta V_g(\theta)$.

2.2 Meta reinforcement learning

Meta-RL aims to maximize the average value function evaluated at the updated policy parameter $\theta'_N = \theta + \eta \frac{1}{N} \sum_{i=1}^N R(\tau_i, g)\nabla_\theta \log p_{\theta,g}(\tau_i)$ obtained by ascent with N-sample PG estimates. Here, $(\tau_i)_{i=1}^N \sim p_\theta$ i.i.d. and η is a fixed stepsize. Formally, consider the following optimization problem,

$$\max_\theta \; \mathbb{E}_g\left[L_N(\theta, g)\right], L_N(\theta, g) \coloneqq \mathbb{E}_{(\tau_i)_{i=1}^N}\left[V_g\left(\theta + \eta \frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\nabla_\theta \log p_{\theta,g}(\tau_i)\right)\right], \quad (1)$$

The expectations are over the goal distribution $g \sim p_{\mathcal{G}}$ and random trajectories $(\tau_i)_{i=1}^N \sim p_\theta$. The N-sample PG estimate update from θ to θ'_N is called the *inner loop update*. We call L_N the *N-sample meta-RL objective* due to its critical dependency on N. Since the task distribution $p_{\mathcal{G}}$ does not depend on θ, we mostly focus on discussing of properties of L_N as a function of θ in later sections. The N-sample meta-RL objective was initially proposed in [16, 17] under the name E-MAML and analyzed in [15] in more theoretical contexts.

The limit case $N \to \infty$. Under mild conditions, the limit exists when $N \to \infty$ and Eqn 1 converges to the following problem

$$\max_\theta \mathbb{E}_g \left[L_\infty(\theta, g)\right], L_\infty(\theta, g) := V_g\left(\theta + \eta \nabla_\theta V_g(\theta)\right). \tag{2}$$

In other words, the inner loop update becomes exact PG ascent. This objective was proposed in the initial MAML framework [4].

Short notes on prior work. Though prior literature mainly focuses on deriving gradient estimates to the MAML objective, we show that there is a fundamental challenge in obtaining unbiased estimates (see Sec 5). Instead, we start the discussion in Sec 3 on the N-sample meta-RL objective.

2.3 Estimating stochastic gradient of Monte-Carlo objectives

To facilitate discussions in later sections, we provide a brief background on optimizing general Monte-Carlo objectives. Monte-Carlo (MC) objectives are common in RL, generative modeling and various probability inference problems (see, e.g., [20, 21] for related reviews). In its general form, MC objectives are defined as $L(\theta) := \mathbb{E}_{X\sim p_\theta}\left[f(X)\right]$ where random variables X are drawn from a distribution p_θ that depends on learnable parameter θ. For simplicity, we first consider when f depends explicitly on X only, though it can also depend on θ. To optimize $L(\theta)$, it is of direct interest to construct unbiased estimates to $\nabla_\theta L(\theta)$.

Score function gradient estimate. The score function (SF) gradient estimate is well defined under the general assumption that f is bounded.

$$\hat{\nabla}_\theta^{\text{SF}} L(\theta) := f(X)\nabla_\theta \log p_\theta(X), X \sim p_\theta.$$

By construction, The estimate is unbiased. However, due to the gradient of score function $\nabla_\theta \log p_\theta(X)$, the estimate often has high variance in practice.

Path-wise gradient estimate. If there exists an elementary distribution $\zeta \sim p_\zeta$ (e.g. normal distribution $\mathcal{N}(0,1)$) and parameter-dependent transformation function $\mathcal{T}_\theta$ such that $\mathcal{T}_\theta(\zeta)$ is equal in distribution to $X \sim p_\theta$, we call X reparameterizable. When X is reparameterizable and f is differentiable, the path-wise (PW) gradient estimate exists and is unbiased

$$\hat{\nabla}_\theta^{\text{PW}} L(\theta) := \left[\nabla_X f(X)\right]_{X=\mathcal{T}_\theta(\zeta)} \nabla_\theta \mathcal{T}_\theta(\zeta), \zeta \sim p_\zeta.$$

Intuitively, PW gradient estimate makes use of the gradient $\nabla_X f(X)$ and enjoys lower variance compared to the SF gradient estimate in many applications [22]. However, the PW gradient estimate is less generally applicable due to assumptions on X and f. For example, those assumptions are not satisfied for important applications such as RL and meta-RL.

3 Meta-RL as N-sample additive Monte-Carlo objective

We start our discussion by extending the MC objective to N-sample additive MC objective. This general framework encompasses meta-RL as a special case and entails the natural derivative of a new estimate in Sec 4.

3.1 N-sample additive Monte-Carlo objective

Let $(X_i)_{i=1}^N \sim p_\theta$ be i.i.d. samples from a parameterized distribution p_θ on domain $\mathcal{X}$. Define $\phi : \mathcal{X} \mapsto \mathbb{R}^h$ as feature mapping function and let $f : \mathbb{R}^h \mapsto \mathbb{R}$ be a scalar function. We define the N-sample additive MC objective as follows,

$$L(\theta) := \mathbb{E}_{(X_i)_{i=1}^N}\left[f\left(\frac{\sum_{i=1}^N \phi(X_i)}{N}\right)\right]. \tag{3}$$

The N-sample additive MC objective can be recovered as a special case of the MC objective by defining $X := (X_i)_{i=1}^N$. However, we will find it very useful to make clear the dependency on N samples when studying the property of $L(\theta)$. In addition, the objective defines interactions between $\phi(X_i)$ in an additive manner, which seems quite restrictive. We will see later that this restrictive definition generalizes the meta-RL objective $L_N(\theta, g)$ as a special case.

We ground the discussion with a toy example.

Toy N-sample additive MC objective. Consider when p_θ is a parameterized Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ where $\sigma > 0$ is fixed. The feature mapping ϕ and objective f are both identity functions.

3.2 Gradient estimates for N-sample additive MC objective

The SF gradient estimate to the N-sample additive MC objective is

$$\hat{\nabla}_\theta^{\text{SF}} L(\theta) \coloneqq f\left(\frac{\sum_{i=1}^N \phi(X_i)}{N}\right) \sum_{i=1}^N \nabla_\theta \log p_\theta(X_i), (X_i)_{i=1}^N \sim p_\theta. \tag{4}$$

Since the SF estimate changes distributions over N variables at the same time, $\sum_{i=1}^N \nabla_\theta \log p_\theta(X_i)$ sums over N terms. This implies high variance, which we calculate exactly for the toy example.

Lemma 3.1. In the toy MC objective example, $\mathbb{V}\left[\hat{\nabla}_\theta^{\text{SF}} L(\theta)\right] = O(N)$.

The variance depends linearly on N! This makes the estimate very hard to use in applications with large N. Compared to the SF estimate, when the PW estimate $\hat{\nabla}_\theta^{\text{PW}} L(\theta)$ is available, it has much lower variance. In the toy example, it is indeed the case since $X = \sigma \cdot \zeta + \mu, \zeta \sim \mathcal{N}(0, 1)$,

Lemma 3.2. In the toy MC objective example, $\mathbb{V}\left[\hat{\nabla}_\theta^{\text{PW}} L(\theta)\right] = 0$.

The zero variance is specialized to the toy example. Since PW gradient estimates are not applicable in RL and meta-RL, we will not discuss them further. Nevertheless, they serve as an golden standard for low-variance unbiased gradient estimates.

3.3 Gradient estimates when f, ϕ depends on θ

Next, we the discussion to the case where f, ϕ depends on parameter θ. Define the *generalized* N-sample additive MC objective as follows

$$G(\theta) \coloneqq \mathbb{E}_{(X_i)_{i=1}^N}\left[f\left(\frac{\sum_{i=1}^N \phi(X_i, \theta)}{N}, \theta\right)\right]. \tag{5}$$

We start by deriving exact gradient to the objective

Lemma 3.3. Let $\bar{\phi}_N \coloneqq \frac{1}{N}\sum_{i=1}^N \phi(X_i, \theta)$. The generalized N-sample additive MC objective has gradient $\nabla_\theta G(\theta)$ as follows where $(X_i)_{i=1}^N \sim p_\theta$ i.i.d.,

$$\mathbb{E}_{(X_i)_{i=1}^N}\left[\underbrace{f\left(\bar{\phi}_N, \theta\right) \sum_{i=1}^N \nabla_\theta \log p_\theta(X_i)}_{\text{term (i)}} + \underbrace{\nabla_\theta f\left(\theta, \bar{\phi}_N\right) + \left(\frac{1}{N}\sum_{i=1}^N \nabla_\theta \phi(\theta, X_i)\right) \nabla_{\bar{\phi}_N} f(\theta, \bar{\phi}_N)}_{\text{term (ii)}}\right]$$

Generalized SF gradient estimate. With access to samples $(X_i)_{i=1} \sim p_\theta$, we define the generalized SF gradient estimate $\hat{\nabla}_\theta^{\text{SF}} G(\theta)$ as follows

$$\underbrace{f\left(\bar{\phi}_N, \theta\right) \sum_{i=1}^N \nabla_\theta \log p_\theta(X_i)}_{\text{term (i)}} + \underbrace{\nabla_\theta f\left(\theta, \bar{\phi}_N\right) + \left(\frac{1}{N}\sum_{i=1}^N \nabla_\theta \phi(X_i, \theta)\right) \nabla_{\bar{\phi}_N} f(\bar{\phi}_N, \theta)}_{\text{term (ii)}}. \tag{6}$$

The two terms in the estimate echo the two terms in the exact gradient in Lemma 3.3. Term (i) corresponds to the SF estimate in Eqn 4. Term (ii) is a direct result of how f, ϕ depends on θ. We provide a full derivation in Appendix A. Examining term (i) and term (ii), we argue that the variance of the overall estimate mainly comes from term (i). This is because term (ii) **averages** over N terms (e.g., with $\bar{\phi}_N$) whereas term (i) **sums** over N score function gradients $\nabla_\theta \log p_\theta(X_i)$.

3.4 Meta-RL as generalized N-sample additive MC objective

With the conversion: $X_i := \tau_i, \phi(X_i, \theta) := R(\tau_i, g)\nabla_\theta \log p_{\theta,g}(\tau_i)$ and $f(\bar{\phi}_N, \theta) = V_g(\theta + \eta\bar{\phi}_N)$, we can cast meta-RL as a special instance of the generalized N-sample additive MC objective.

We start by computing gradient of the N-sample objective $J_N(\theta, g) := \nabla_\theta L_N(\theta, g)$ as a direct result of Lemma 3.3.

Lemma 3.4. Let $\tau_i \sim p_\theta$ i.i.d., $\nabla V_g(\theta'_N)$ denotes $[\nabla_\theta V_g(\theta)]_{\theta=\theta'_N}$ and $\theta'_N := \theta + \eta \frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\nabla_\theta \log p_{\theta,g}(\tau_i)$ is the (random) updated parameter. Then the gradient $J_N(\theta, g) := \nabla_\theta L_N(\theta, g)$ is

$$\underbrace{\mathbb{E}_{(\tau_i)_{i=1}^N}\left[V_g(\theta'_N)\sum_{i=1}^N \nabla_\theta \log p_{\theta,g}(\tau_i)\right]}_{=:J_N^{(i)}(\theta,g)} + \underbrace{\mathbb{E}_{(\tau_i)_{i=1}^N}\left[\left(I + \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right)\nabla V_g(\theta'_N)\right]}_{=:J_N^{(ii)}(\theta,g)}, \tag{7}$$

We reiterate intuitions about the two gradient terms in the context of meta-RL. The parameter θ influences the objective $L_N(\theta, g)$ in two different ways. The first term arises from the fact that the N random trajectories are sampled from p_θ, which depends on θ. The second term is a result of how θ impacts $L_N(\theta, g)$ explicitly through the inner loop N-sample PG estimate.

Unbiased meta-RL gradient estimate. In the following, we specify an algorithmic procedure to construct unbiased estimates to $J_N(\theta, g)$. This is a direct instantiation of the generalized SF gradient estimate in Eqn 6 in the context of meta-RL.

Corollary 3.5. First, sample $(\tau_i)_{i=1}^N \sim p_\theta$ and computed the updated parameter θ'_N. Then, construct unbiased estimates to $\nabla V_g(\theta'_N)$ and $V_g(\theta'_N)$, e.g. with trajectories sampled under $\pi_{\theta'_N}$. Let these estimates be $\nabla\hat{V}_g(\theta'_N)$ and $\hat{V}_g(\theta'_N)$ respectively[1]. The final estimate is

$$\underbrace{\hat{V}_g(\theta'_N)\sum_{i=1}^N \nabla_\theta \log p_{\theta,g}(\tau_i)}_{=:\hat{J}_{N,\text{SF}}^{(i)}(\theta,g)} + \underbrace{\left(I + \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right)\nabla\hat{V}_g(\theta'_N)}_{=:\hat{J}_{N,\text{SF}}^{(ii)}(\theta,g)}. \tag{8}$$

Both terms are unbiased $\mathbb{E}[\hat{J}_{N,\text{SF}}^{(i)}(\theta, g)] = J_N^{(i)}(\theta, g), \mathbb{E}[\hat{J}_{N,\text{SF}}^{(ii)}(\theta, g)] = J_N^{(ii)}(\theta, g)$ with respect to the two terms in Eqn 7. This implies that the overall estimate is also unbiased.

Variance of the unbiased gradient estimate. As direct implications of the properties of SF gradient estimate and generalized SF gradient estimate, $\hat{J}_N$ has very high variance. In fact, building on the N-sample additive MC objective toy example, we can construct meta-RL examples where $\mathbb{V}[\hat{J}^{(ii)}] = O(N)$. See Appendix A for more details. Our objective now is to develop new estimates which bypass the high variance of the unbiased estimate.

4 Linearized score function gradient estimate

We now introduce a major development in this paper: a new gradient estimate for the N-sample additive MC objective. This estimate is in general biased but has significantly lower variance ($O(1/N)$) compared to the SF estimate ($O(N)$) when N is large, making it attractive in practice.

4.1 Linearized SF gradient estimate for N-sample additive MC objective.

When the PW gradient estimate is applicable, it often has lower variance than the SF gradient estimate. Previously, we argue that this is because PW leverages gradient information in the objective f while SF does not. Building on this intuition, we propose a new gradient estimate called *linearized score function* (LSF) gradient estimate as follows,

$$\hat{\nabla}_\theta^{\text{LSF}} L(\theta) := \left[\nabla f\left(\bar{\phi}_N\right)\right]^T \frac{1}{N}\sum_{i=1}^N \phi(X_i)\nabla_\theta \log p_\theta(X_i). \tag{9}$$

[1]For now, we just require the estimates to be unbiased. In Section 6, we make these estimates concrete for refined convergence analysis.

Recall that $\bar{\phi}_N := \frac{\sum_{i=1}^N \phi(X_i)}{N}$ and $\nabla f(\bar{\phi}_N)$ denotes $[\nabla_x f(x)]_{x=\bar{\phi}_N}$. The LSF gradient estimate makes use of the gradient of f yet does not require reparameterization of the random variables X. In this sense, it is more general than the PW gradient estimate, yet leverages more information than the SF estimate. Indeed, LSF achieves significant variance reduction than SF.

Lemma 4.1. In the toy MC objective example, $\mathbb{V}\left[\hat{\nabla}_\theta^{\text{LSF}} L(\theta)\right] = O(1/N)$.

In the toy example, the PW gradient estimate is the gold standard unbiased estimate with zero variance. Yet, as discussed before, it is not generally applicable. The LSF gradient estimate has variance $O(1/N)$, which decays as N increases. This makes LSF applicable in large N regimes. However, unlike the SF estimate which is by design unbiased, the LSF estimate is in general biased.

Lemma 4.2. In general, when f is twice continuously differentiable and $\left\|\nabla_x^2 f(x)\right\|_2 \leq C$ for all x in domains of f with a constant C. Then $\text{Bias}[\hat{\nabla}_\theta^{\text{LSF}} L(\theta)] = O(1/\sqrt{N})$.

Derivation of the estimate. The naming *linearized* implies how the estimate was derived in the first place, which we show in Appendix A. In a nutshell, LSF is derived using a local linearization of $f(\bar{\phi}_N)$ used in the SF estimate, based on Taylor expansion. This allows LSF to utilize gradient information $\nabla f(\bar{\phi}_N)$ to reduce variance, yet still remain generally applicable.

4.2 Gradient estimate for Generalized N-sample additive MC objective

We extend the LSF gradient estimate to the generalized N-sample additive MC objective in Eqn 5. We do so by replacing the term (i) SF estimate by LSF estimate in Eqn 6. This produces the generalized LSF gradient estimate $\hat{\nabla}_\theta^{\text{LSF}} G(\theta)$ as follows,

$$\underbrace{\left[\nabla_{\bar{\phi}_N} f\left(\bar{\phi}_N, \theta\right)\right]^T \frac{1}{N} \sum_{i=1}^N \phi(X_i, \theta) \nabla_\theta \log p_\theta(X_i, \theta)}_{\text{term (i)}} + \underbrace{\nabla_\theta f\left(\theta, \bar{\phi}_N\right) + \left(\frac{1}{N}\sum_{i=1}^N \nabla_\theta \phi(X_i, \theta)\right) \nabla_{\bar{\phi}_N} f(\bar{\phi}_N, \theta)}_{\text{term (ii)}}. \tag{10}$$

Due to the bias in the LSF gradient estimate, the generalized LSF estimate is also biased. However, the key trade-off is that the new term (i) in Eqn 10 **averages** over N samples and achieves significantly smaller variance than the generalized SF estimate.

4.3 Biased gradient estimate to the meta-RL objective.

We next apply the generalized LSF gradient estimate to the N-sample meta-RL objective.

Corollary 4.3. Let $u_i := \nabla_\theta \log p_{\theta,g}(\tau_i)$. Define $\nabla \hat{V}_g, \hat{V}_g$ in the same way as in Lemma 3.5. Then the LSF gradient estimate $\hat{J}_{N,\text{LSF}}(\theta, g)$ to $L_N(\theta, g)$ is expressed as follows,

$$\underbrace{\left(\frac{1}{N}\sum_{i=1}^N R(\tau_i, g) u_i u_i^T\right) \nabla \hat{V}_g(\theta_N')}_{=:\hat{J}^{(i)}_{N,\text{LSF}}(\theta,g)} + \underbrace{\left(I + \eta \frac{1}{N}\sum_{i=1}^N R(\tau_i, g) \nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right) \nabla \hat{V}_g(\theta_N')}_{=:\hat{J}^{(ii)}_{N,\text{LSF}}(\theta,g)}, \tag{11}$$

While the unbiased SF estimate $\hat{J}^{(iii)}_{N,\text{SF}}$ has high variance when N is large, the LSF estimate $\hat{J}^{(i)}_{N,\text{LSF}}$ achieves a good trade-off between bias and variance. We will show how such trade-off impacts the convergence analysis in Section 6.

Connections to exact gradient for meta-RL objective $L_\infty(\theta, g)$. It is now worthwhile to contrast the generalized LSF estimate to the gradient of $J_\infty(\theta, g) := \nabla_\theta L_\infty(\theta, g)$.

Corollary 4.4. Let $u_i := \nabla_\theta \log p_{\theta,g}(\tau_i)$ and $\theta' = \theta + \eta \mathbb{E}_{\tau \sim p_{\theta,g}}[R(\tau, g)\nabla_\theta \log p_{\theta,g}(\tau)]$ be the updated parameter with exact PG ascent. In the following, let $(\tau_i)_{i=1}^N \sim p_{\theta,g}$ i.i.d., then $J_\infty(\theta, g)$ is

$$\underbrace{\mathbb{E}_{(\tau_i)_{i=1}^N}\left[\frac{1}{N}\sum_{i=1}^N R(\tau_i, g) u_i u_i^T \nabla V_g(\theta')\right]}_{=: J_\infty^{(i)}(\theta,g)} + \underbrace{\mathbb{E}_{(\tau_i)_{i=1}^N}\left[\left(I + \eta \frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right)\nabla V_g(\theta')\right]}_{=: J_\infty^{(ii)}(\theta,g)}, \tag{12}$$

Here, since θ' is the updated parameter resulting from exact PG ascent, it is not easy to construct unbiased estimate to $J_\infty(\theta, g)$. This is because even if we can compute θ'_N as N-sample unbiased estimate to θ', in general we still have $\nabla V_g(\theta') \neq \mathbb{E}[\nabla V_g(\theta'_N)]$. However, note that there are similarities between the parametric forms of $\hat{J}_{N,\text{LSF}}(\theta, g)$ and $J_\infty(\theta, g)$. We can interpret $\hat{J}_{N,\text{LSF}}(\theta, g)$ as also a biased estimate to $\hat{J}_{N,\text{LSF}}(\theta, g)$, obtained by replacing θ' with θ'_N.

5 Discussion on prior work

N-sample meta-RL objective. As noted earlier, the N-sample meta-RL objective was considered in both empirical [16, 17] and theoretical contexts [15]. This objective is of practical interest because of budget on inner loop samples. The limit case $N = \infty$ was considered in the original MAML formulation of meta-RL [4].

Unbiased gradient to the limit case $J_\infty(\theta, g)$. In the author's original implementation of the MAML gradient estimate with auto-differentiation libraries [4], a term equivalent to $J_\infty^{(i)}(\theta, g)$ was unintentionally dropped, resulting in a biased estimate. This fuels the motivation for a number of follow-up work to derive unbiased gradients [23, 18]. However, they are **biased** in general. This is mainly because practical algorithms can only estimate $\nabla_g V_g(\theta'_N)$ instead of $\nabla_g V_g(\theta')$, as required by $J_\infty(\theta, g)$ in Eqn 12. This observation was also hinted at recently in [19].

Prior work in fact constructs the LSF gradient estimate. Since most prior work derive meta-RL gradient estimates based on $J_\infty(\theta, g)$ [23, 17–19], and due to the *accidental* replacement of θ' by θ'_N, we conclude that they in fact construct variants of the LSF gradient estimate (see comments following Corollary 4.4). In particular they construct $\hat{J}$ such that $\mathbb{E}[\hat{J}] = \mathbb{E}[\hat{J}_{N,\text{LSF}}(\theta, g)]$ but with potentially lower variance. All of them focus on reducing variance of estimating the multiplier matrix to $\nabla V_g(\theta'_N)$. Variance reduction methods include control variates [18], as well as introducing further bias to the LSF gradient estimate [17, 19].

Unbiased gradient estimate to N-sample meta-RL objective. The exact gradient and unbiased gradient estimate to N-sample meta-RL objective was derived in [16, 17, 15]. A comprehensive derivation was carried out in [17], where they contrasted $J_\infty(\theta, g)$ with $J_N(\theta, g)$. However, they erroneously claimed that $J_\infty^{(ii)}(\theta, g) = J_N^{(ii)}(\theta, g)$ and only differs in $J_\infty^{(i)}(\theta, g) \neq J_N^{(i)}(\theta, g)$. This is not true. Our derivation shows that $J_\infty^{(ii)}(\theta, g) \neq J_N^{(ii)}(\theta, g)$ in general because $\mathbb{E}[\nabla V_g(\theta'_N)] \neq \nabla V_g(\theta')$.

Convergence analysis of gradient-based meta-learning and meta-RL. Recently, [24] established generic convergence guarantees for gradient-based meta-learning algorithms for supervised learning with one inner loop update. Recently, [25] extended the analysis to multi-step inner loop updates.

For meta-RL, [15] established convergence for the N-sample meta-RL objective. They motivated the objective in a similar manner as [16, 17] and constructed unbiased estimates exactly as the generalized SF estimate $\hat{J}_{N,\text{SF}}(\theta, g)$. However, since the estimate has variance linear in N, the final guarantee becomes less applicable in practice. Contrast to this work, we show how the biased generalized LSF estimate achieves performance guarantee with more desirable dependency on N.

6 Full Algorithm and Convergence theory with biased gradient estimate

We start by presenting the meta-RL full algorithm with generalized LSF estimate. This algorithm closely resembles how practical algorithms are implemented. The same algorithm with generalized SF estimate was analyzed in [15].

6.1 Full algorithm and key assumptions

The full meta-RL algorithm is in Algorithm 1. Two important notes: (1) We instantiate the unbiased gradient estimate $\nabla V_g(\theta'_N)$ by M-sample PG estimates with trajectories collected under the updated parameter θ'_N; (2) So far we have focused on presenting gradient estimate for a single task g. In practice, we sample a batch of B tasks $(g_i)_{i=1}^B$ and compute gradient estimate for each $\hat{J}_{N,\text{LSF}}(\theta, g_i)$. The overall gradient $\hat{J}_{N,\text{LSF}}$ is an average across tasks, which is used for the final update at each iteration $\theta \leftarrow \theta + \alpha \hat{J}$ with learning rate $\alpha > 0$.

Algorithm 1 N-sample meta-RL algorithm with linearized SF gradient estimate

Require: Inputs: Hyper-parameters: batch sizes (B, N, M). Step size η. Initial parameter θ.
for ite $= 1, 2...$ **do**
  Inner loop sampling. Sample B task variables g_i and N trajectories under $(\tau_{i,j})_{j=1}^N \sim p_{\theta,g_i}$.
  Inner update. Compute inner loop update $\theta'_{i,N} = \theta + \eta \frac{1}{N} \sum_{j=1}^N R(\tau_{i,j}, g_i) \nabla_\theta \log p_{\theta,g_i}(\tau_{i,j})$.
  Outer sampling at adapted parameters. Collect M trajectories $(\tau'_{i,k})_{k=1}^M \sim p_{\theta'_{i,N},g_i}$ for the outer loop PG estimate $\nabla_\theta \hat{V}_{g_i}(\theta'_{i,N}) = \frac{1}{M} \sum_{k=1}^M R(\tau'_{i,k}, g_i) \nabla_\theta \log p_{\theta,g_i}(\tau'_{i,k})$.
  Gradient estimate and update. Compute $\hat{J}_{N,\text{LSF}}(\theta, g_i)$ based on Eqn 11. Then compute $\hat{J}_{N,\text{LSF}} = \frac{1}{B} \sum_{i=1}^B \hat{J}_{N,\text{LSF}}(\theta, g_i)$ as the average estimate. Update outer loop $\theta \leftarrow \theta + \alpha \hat{J}_{N,\text{LSF}}$.
end for
Output trained meta-RL policy π_θ.

We also need a few common assumptions [15] for theoretical analysis.

Assumption 6.1. (Smooth parameterization assumptions) For all $s \in \mathcal{S}, a \in \mathcal{A}, g \in \mathcal{G}$, $\|\nabla_\theta \log \pi_\theta(a|s,g)\|_2 \leq G_1$ and $\left\|\nabla_\theta^2 \log \pi_\theta(a|s,g)\right\|_2 \leq G_2$. In addition, for all $\theta_1, \theta_2 \in \mathbb{R}^D$, $\left\|\nabla_\theta^2 \log \pi_{\theta_1}(a|s,g) - \nabla_\theta^2 \log \pi_{\theta_2}(a|s,g)\right\|_2 \leq \rho \|\theta_1 - \theta_2\|_2$.

6.2 Main result

The meta-RL objectives takes an average over the parameter-independent distribution $g \sim p_{\text{G}}$ and hence its overall gradients are $J_N(\theta) \coloneqq \mathbb{E}_g[J_N(\theta, g)]$ and for the limit case $J_\infty(\theta) \coloneqq \mathbb{E}_g[J_\infty(\theta, g)]$. As previously discussed, the generalized LSF estimate is biased in general. We start by characterizing its bias against $J_N(\theta)$. We have the following.

Proposition 6.2. For all $\theta \in \mathbb{R}^D$, $\left\|\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] - J_\infty(\theta)\right\|_2 \leq O(1/\sqrt{N})$ and $\|J_\infty(\theta) - J_N(\theta)\|_2 \leq O(1/\sqrt{N})$.

The above also implies a bound on the bias $\left\|\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] - J_N(\theta)\right\|_2 = O(1/\sqrt{N})$. This is consistent with the result in Sec 4. We next characterize the variance of the generalized LSF estimate.

Proposition 6.3. For all $\theta \in \mathbb{R}^D$, $\mathbb{V}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] \leq O(1/M) + O(1/N) + O(1/B)$.

The three terms on the upper bound above indicate these three sources of randomness that contribute the variance of the generalized LSF estimate $\hat{J}_{N,LSF}(\theta)$: the batch of B tasks, the batch of N inner loop trajectories τ_{ij} per task and the batch of M trajectories τ'_{ik} for estimating outer loop PG. By letting $B \to \infty, M \to \infty$, we see that the variance is of order $O(1/N)$. This is consistent with the variance of the LSF estimate for the N-sample additive MC objective in Sec 4.

The above implies convergence guarantees for the biased gradients, we will complete the result in camera-ready. We will also make explicit comparison to [15] and show that our guarantees have more superior dependency on N.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[2] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[3] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[5] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

[6] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[7] Rein Houthooft, Richard Y Chen, Phillip Isola, Bradly C Stadie, Filip Wolski, Jonathan Ho, and Pieter Abbeel. Evolved policy gradients. *arXiv preprint arXiv:1802.04821*, 2018.

[8] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[9] Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

[10] Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J Smola. Meta-q-learning. *arXiv preprint arXiv:1910.00125*, 2019.

[11] Pedro A Ortega, Jane X Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, et al. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

[12] Junhyuk Oh, Matteo Hessel, Wojciech M Czarnecki, Zhongwen Xu, Hado van Hasselt, Satinder Singh, and David Silver. Discovering reinforcement learning algorithms. *arXiv preprint arXiv:2007.08794*, 2020.

[13] Zhongwen Xu, Hado van Hasselt, Matteo Hessel, Junhyuk Oh, Satinder Singh, and David Silver. Meta-gradient reinforcement learning with an objective discovered online. *arXiv preprint arXiv:2007.08433*, 2020.

[14] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[15] Alireza Fallah, Kristian Georgiev, Aryan Mokhtari, and Asuman Ozdaglar. Provably convergent policy gradient methods for model-agnostic meta-reinforcement learning. *arXiv preprint arXiv:2002.05135*, 2020.

[16] Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*, 2017.

[17] Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. *arXiv preprint arXiv:1810.06784*, 2018.

[18] Hao Liu, Richard Socher, and Caiming Xiong. Taming maml: Efficient unbiased meta-reinforcement learning. In *International Conference on Machine Learning*, pages 4061–4071. PMLR, 2019.

[19] Yunhao Tang, Tadashi Kozuno, Mark Rowland, Rémi Munos, and Michal Valko. Unifying gradient estimators for meta-reinforcement learning via off-policy evaluation. *arXiv preprint arXiv:2106.13125*, 2021.

[20] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[21] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21(132):1–62, 2020.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Jakob Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktäschel, Eric Xing, and Shimon Whiteson. Dice: The infinitely differentiable monte carlo estimator. In *International Conference on Machine Learning*, pages 1529–1538. PMLR, 2018.

[24] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1082–1092. PMLR, 2020.

[25] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Multi-step model-agnostic meta-learning: Convergence and improved algorithms. *arXiv preprint arXiv:2002.07836*, 2020.

A Derivation of the Linearized Score Function Estimate

Since X_is are i.i.d., we expect the average $\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)$ to approach $\bar{\phi} \coloneqq \mathbb{E}\left[\phi(X_i)\right]$ as $N \to \infty$. Consider the Taylor expansion of $f(\bar{\phi})$ with $\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)$ as its refernece point,

$$f\left(\bar{\phi}\right) = f\left(\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right) + \left[\nabla f\left(\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right)\right]^T\left[\bar{\phi} - \left(\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right)\right] + o\left(\left\|\bar{\phi} - \frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right\|_2\right)$$

Rearranging terms, we get

$$f\left(\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right) = f\left(\bar{\phi}\right) + \left[\nabla f\left(\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right)\right]^T\left[\left(\frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right) - \bar{\phi}\right] + o\left(\left\|\bar{\phi} - \frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\right\|_2\right)$$

Now consider each term above. The constant term $f(\bar{\phi})$ is independent of θ and produces zero gradient. If we drop the residual term, we are left with the central term .

Note that the central term contains $\bar{\phi}$, which we do not have access to. If we multiply the above terms with $\sum_i \nabla \log p(X_i)$, and by dropping terms with expectation zero as well as $\bar{\phi}$ (when dropping $\bar{\phi}$ the expected gradient does not change), we arrive at the LSF estimate

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