Biased Gradient Estimate with Drastic Variance Reduction for Meta Reinforcement Learning

Yunhao Tang [1]

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, we investigate such a discrepancy. In particular, (1) we show that unbiased gradient estimates have variance $\Theta(N)$ 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 $\mathcal{O}(1/\sqrt{N})$ and variance $\mathcal{O}(1/N)$; (3) we show that most empirical prior work in fact implements variants of the LSF gradient estimates. This implies that practical algorithms "accidentally" introduce bias to achieve better performance; (4) we establish theoretical guarantees for the LSF gradient estimates in meta-RL regarding its convergence to stationary points, showing better dependency on N than prior work when N is large.

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 (Mnih et al., 2013; Silver et al., 2016; Levine et al., 2016). 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 (Duan et al., 2016; Wang et al., 2016; Finn et al., 2017). 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. (Wang et al., 2015; Duan et al., 2016; Houthooft et al., 2018; Rakelly et al., 2019; Zintgraf et al., 2019; Fakoor et al., 2019; Ortega et al., 2019; Oh et al., 2020; Xu et al., 2020)), Our focus is meta-RL through gradient-based adaptations (Finn et al., 2017), where the agent carries out policy gradient (PG) inner loop updates (Sutton et al., 2000) 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 (Fallah et al., 2020a). However, since the inception of the field, meta-RL practitioners have almost always implemented biased gradient estimates (Finn et al., 2017; Al-Shedivat et al., 2017; Rothfuss et al., 2018; Liu et al., 2019; Tang et al., 2021). 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 (Al-Shedivat et al., 2017; Rothfuss et al., 2018; Fallah et al., 2020a), as opposed to the MAML objective (Finn et al., 2017) where the inner loop update is exact PG. This objective is of practical interest, because at meta-testing time, inner loop updates can only be implemented with N-sample PG estimates. See Sec 1 for details.

Summary of this work. We make a number of developments to bridge meta-RL theory and practice.

- **High variance of unbiased estimates.** By formulating the meta-RL problem as optimizing a generic N-sample additive Monte-Carlo objective, we show that the unbi-

[1]DeepMind. Correspondence to: Yunhao Tang <robintyh@deepmind.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

ased gradient estimates have variance on the order of $\Theta(N)$, rendering the estimates useless when N is large (see Sec 2).

- **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 $\mathcal{O}(1/N)$ and bias $\mathcal{O}(1/\sqrt{N})$. Its application to meta-RL enjoys better properties at large N (see Sec 3).
- **Prior work implements biased estimates.** We observe that despite their claims of unbiasedness, most prior work in fact implements variants of LSF gradient estimates. This implies they are both biased w.r.t. the MAML and the N-sample meta-RL objective (see Sec 4).
- **Performance guarantee with better dependency on N.** We provide performance guarantee of meta-RL algorithms with biased estimates. Such guarantee contrasts with results of unbiased estimates, where the guarantee degrades significantly at large N (Fallah et al., 2020a) (see Sec 5).

1. Background

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 $s_{t+1} \sim p(\cdot|s_t, a_t)$. Without loss of generality, we assume that the at time $t = 0$ 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. In general, we assume the policy is parameterized π_θ with parameter $\theta \in \mathbb{R}^D$.

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

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

Policy gradient and stochastic estimates. Policy gradient (PG) (Sutton et al., 2000) is the gradient of the value function with respect to policy parameter $\nabla_\theta V_g(\theta) = \mathbb{E}_{\tau\sim p_{\theta,g}}[R(\tau, g)\nabla_\theta \log p_{\theta,g}(\tau)]$. 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 $\tau \sim p_{\theta,g}$ is an unbiased PG estimate in that $\mathbb{E}[\hat{\nabla}_\theta V_g(\theta)] = \nabla_\theta V_g(\theta)$.

1.1. 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 an ascent step with N-sample PG estimates. This ascent step is also called the *inner loop update*. Here, $(\tau_i)_{i=1}^N \sim p_{\theta,g}$ i.i.d. and η is a fixed stepsize. Formally, consider the following optimization problem,

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

The expectations are over the goal distribution $g \sim p_{\mathcal{G}}$ and random trajectories $(\tau_i)_{i=1}^N \sim p_{\theta,g}$. 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 (Al-Shedivat et al., 2017; Rothfuss et al., 2018) under the name E-MAML and analyzed in (Fallah et al., 2020a) 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

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

In other words, the inner loop update becomes exact PG ascent. This objective was proposed in the initial MAML framework (Finn et al., 2017).

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 4). Instead, we start the discussion in Sec 2 on the N-sample meta-RL objective.

1.2. Stochastic Gradient Estimates 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 probabilistic machine learning problems (see, e.g., Blei et al. (2017); Mohamed et al. (2020) for related reviews). In its general form, MC objectives are defined as $L(\theta) \coloneqq \mathbb{E}_{X\sim p_\theta}[f(X)]$ where random variables X are drawn from a distribution p_θ whose density is a differentiable function of θ. For simplicity, we first consider when f depends explicitly on X only, though it can also depend on θ, which we will discuss shortly. To optimize $L(\theta)$, it is of interest to construct unbiased estimates to $\nabla_\theta L(\theta)$.

Score function (SF) gradient estimate. Assume f is bounded[1]. The SF gradient estimate is defined as follows

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

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 (PW) gradient estimate. Due to space limit, we present details on PW gradient estimate in Appendix B. The PW estimate is not applicable in meta-RL, though it can be used as a golden baseline for variance comparison.

2. Meta-RL as N-sample Additive Monte-Carlo Objective

We start by extending the MC objective to N-sample additive MC objective. This general framework encompasses meta-RL as a special case. It also allows us to naturally derive a novel estimate with significant variance reduction.

2.1. 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 that takes $x \in \mathcal{X}$ as input and outputs a h-dimensional feature $\phi(x)$. Let $f : \mathbb{R}^h \mapsto \mathbb{R}$ be a scalar function that maps from the feature space to a scalar value. We define the N-sample additive MC objective as follows,

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

The N-sample additive MC objective can be recovered as a special case of the MC objective by defining $X \coloneqq (X_i)_{i=1}^N$. However, we will find it useful to make clear how the property of $L(\theta)$ explicitly depends on N. Though the objective defines interactions between $\phi(X_i)$ in an additive manner, we will see that this seemingly restrictive definition generalizes the N-sample meta-RL objective $F_N(\theta, g)$ as a special case. In the following, 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.

2.2. Gradient Estimates for N-sample Additive MC Objective

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

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

where $(X_i)_{i=1}^N \sim p_\theta$. Since the SF gradient 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[2].

Lemma 1. In the toy N-sample MC objective example, $\mathbb{V}\left[\hat{\nabla}^{\text{SF}}_\theta L(\theta)\right] = \Theta(N)$.

The variance depends linearly on N! This makes the estimate very hard to use in applications with large N. Compared to the SF gradient estimate, when the PW gradient estimate $\hat{\nabla}^{\text{PW}}_\theta L(\theta)$ 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 2. In the toy N-sample MC objective example, $\mathbb{V}\left[\hat{\nabla}^{\text{PW}}_\theta L(\theta)\right] = 0$.

The zero variance is specialized to the toy example, though in general the PW gradient estimate also tends to achieve very small variance, making it a golden standard for unbiased gradient estimates. However, PW gradient estimates are not generally applicable, e.g., to RL and meta-RL objectives.

2.3. Gradient Estimates for Generalized N-sample Additive MC Objective

Next, we discuss the case where the functions f, ϕ depend also on the parameter θ. Define the *generalized* N-sample additive MC objective as follows

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

[1] Here, we assume f to be bounded for simplicity, though the SF gradient estimate is well defined and unbiased under more general assumptions (Mohamed et al., 2020). This boundedness assumption is satisfied for the meta-RL application to be discussed later.

[2] Throughout the presentation, we will use the "Big O" notations. See Appendix A for their detailed definitions.

We start by deriving exact gradient to the objective

Lemma 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.,

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

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

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

The two terms in the estimate echo the two terms in the exact gradient in Lemma 3. Term (i) corresponds to the SF gradient estimate in Eqn 4. Term (ii) is a direct result of how f,ϕ depends on θ. We provide a full derivation in Appendix C. 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 $\nabla_\theta \log p_\theta(X_i)$.

2.4. Meta-RL as Generalized N-sample Additive MC Objective

With the conversion: $X_i \coloneqq \tau_i, \phi(X_i,\theta) \coloneqq R(\tau_i,g)\nabla_\theta \log p_{\theta,g}(\tau_i)$ and $f(\bar{\phi}_N,\theta) = V_g(\theta+\eta\bar{\phi}_N)$, we cast meta-RL as a special instance of the generalized N-sample additive MC objective. We compute gradient of the N-sample objective $J_N(\theta,g) \coloneqq \nabla_\theta F_N(\theta,g)$ as a direct result of Lemma 3.

Lemma 4. Let $\theta'_N \coloneqq \theta + \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i,g)\nabla_\theta \log p_{\theta,g}(\tau_i)$ be the (random) updated parameter. Let $\tau_i \sim p_{\theta,g}$ i.i.d. and let $\nabla V_g(\theta'_N)$ denotes $[\nabla_\theta V_g(\theta)]_{\theta=\theta'_N}$. Then $J_N(\theta,g) \coloneqq \nabla_\theta F_N(\theta,g)$ is

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

We omit the notation that the expectation is with respect to N random trajectories $(\tau_i)_{i=1}^N$. We now reiterate intuitions about the two gradient terms in the context of meta-RL. The parameter θ influences the objective $F_N(\theta,g)$ in two different ways. The first term arises from the fact that the N random trajectories are sampled from $p_{\theta,g}$, which depends on θ. The second term is a result of how θ impacts $F_N(\theta,g)$ 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 5. First, sample $(\tau_i)_{i=1}^N \sim p_{\theta,g}$ and compute 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[3]. 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 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.

[3] For now, we just require the estimates to be unbiased. In Sec 5, we make these estimates concrete for refined convergence analysis.

In fact, building on the N-sample additive MC objective toy example, we can construct meta-RL examples where unbiased estimates have variance $\Theta(N)$. See Appendix D for more details. Our objective now is to develop new estimates which bypass the high variance of the unbiased estimate.

3. 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 ($\mathcal{O}(1/N)$) compared to the SF gradient estimate ($\Theta(N)$), making it attractive in practice when N is large.

3.1. Linearized Score Function 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,

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

In the above, we define $\nabla f(\bar{\phi}_N) \coloneqq [\nabla_x f(x)]_{x=\bar{\phi}_N}$. The naming *linearized* comes from how the estimate was derived in the first place, which we show in detail in Appendix C.

Variance of the estimate. The following result shows LSF achieves significant variance reduction.

Lemma 6. In the toy N-sample MC objective example, $\mathbb{V}\left[\hat{\nabla}_\theta^{\text{LSF}} L(\theta)\right] = \mathcal{O}(1/N)$.

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 $\mathcal{O}(1/N)$, which decays as N increases. This makes LSF applicable in large N regimes. However, unlike the SF gradient estimate which is by design unbiased, the LSF gradient estimate is in general biased. Nevertheless, when applying the LSF gradient estimate to the N-sample meta-RL objective, we can characterize the bias to be of order $\mathcal{O}(1/N)$ (see Proposition 11).

3.2. 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 gradient estimate by LSF gradient estimate in Eqn 6. This produces the generalized LSF gradient estimate $\hat{\nabla}_\theta^{\text{LSF}} G(\theta)$ as follows,

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

Due to the bias in the LSF gradient estimate, the generalized LSF gradient 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 gradient estimate.

3.3. Biased Gradient Estimate to Meta-RL Objective

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

Corollary 7. Let $u_i \coloneqq \nabla_\theta \log p_{\theta,g}(\tau_i)$. The generalized LSF gradient estimate $\hat{J}_{N,\text{LSF}}(\theta,g)$ to $F_N(\theta,g)$ can be expressed as follows,

$$\left(\eta\frac{1}{N}\sum_{i=1}^{N} R(\tau_i,g)u_i u_i^T\right)\nabla V_g(\theta'_N)\left(I+\eta H_\theta\right)\nabla V_g(\theta'_N),$$

where $H_\theta = \frac{1}{N}\sum_{i=1}^{N} R(\tau_i,g)\nabla_\theta^2 \log p_{\theta,g}(\tau_i)$. Define $\nabla\hat{V}_g, \hat{V}_g$ as unbiased estimates to $\nabla V_g, V_g$. The following estimate has the same expectation as $\hat{J}_{N,\text{LSF}}(\theta,g)$,

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

Since Eqn 11 can be computed via a similar procedure as Lemma 5, we call it the generalized LSF gradient estimate to the meta-RL objective. While the unbiased SF gradient estimate $\hat{J}^{(ii)}_{N,\text{SF}}$ has high variance when N is large, the LSF gradient 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 Sec 5.

Connections to Hessian estimation. We can rewrite the LSF gradient estimate in Eqn 11 as $(I + \eta \hat{H}_N(\theta))\nabla \hat{V}_g(\theta'_N)$, where $\hat{H}_N(\theta) = \frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\left(u_i u_i^T + \nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right)$. It is straightforward to verify that the matrix $I + \eta \hat{H}_N(\theta)$ is an unbiased estimate to the Hessian matrix $\mathbb{E}[\hat{H}_N(\theta)] = \nabla^2 V_g(\theta)$. Most prior work focus on variance reduction for estimating this function (Foerster et al., 2018; Mao et al., 2019; Rothfuss et al., 2018; Tang et al., 2021).

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

Corollary 8. Let $u_i \coloneqq \nabla_\theta \log p_{\theta,g}(\tau_i)$ 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

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

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.

4. Discussion on prior work

We provide a briefly discussion on prior work. See Appendix E for an extended discussion.

Unbiased gradient to $J_\infty(\theta, g)$. In the original implementation of the MAML gradient estimate (Finn et al., 2017), a term equivalent to $J_\infty^{(i)}(\theta, g)$ was dropped, resulting in a biased estimate. This motivates a number of follow-up work to derive unbiased gradients (Foerster et al., 2018; Liu et al., 2019). However, such follow-up estimates are also *biased* in general. This is mainly because practical algorithms can only estimate $\nabla_g V_g(\theta'_N)$ instead of $\nabla_g V_g(\theta')$, while the latter is required to estimate $J_\infty(\theta, g)$ in an unbiased way. This observation was also hinted at recently in (Tang et al., 2021).

Prior work constructs de-facto LSF estimate. Since most prior work derive meta-RL gradient estimates based on $J_\infty(\theta, g)$ (Foerster et al., 2018; Rothfuss et al., 2018; Liu et al., 2019; Tang et al., 2021), 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 8). 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 $\hat{H}_N(\theta)$. Though in theory reducing the variance of $\hat{H}_N(\theta)$ does not necessarily guarantee improvements, in practice, this seems to be very critical.

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 (Al-Shedivat et al., 2017; Rothfuss et al., 2018; Fallah et al., 2020a). A comprehensive derivation was carried out in (Rothfuss et al., 2018), where they contrasted $J_\infty(\theta, g)$ with $J_N(\theta, g)$. However, they claimed that $J_\infty^{(ii)}(\theta, g) = J_N^{(ii)}(\theta, g)$, which is *not* true. Our derivation shows that $J_\infty^{(ii)}(\theta, g) \neq J_N^{(ii)}(\theta, g)$ in general because $\mathbb{E}[\nabla V_g(\theta'_N)] \neq \nabla V_g(\theta')$.

5. Convergence Analysis with Linearized Score Function Gradient Estimate

We start by presenting the meta-RL full algorithm with generalized LSF gradient estimate. Note that the pseudocode also closely relates to how practical algorithms are implemented (see more in Appendix H).

5.1. Full Algorithm and Key Assumptions

The full meta-RL algorithm is in Algorithm 1. There are two important notes on the details: (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 averaged across tasks, which is then used for the final update $\theta_{t+1} = \theta_t + \alpha \hat{J}_{N,\text{LSF}}$ at each iteration.

We need a few common assumptions (Fallah et al., 2020a) for theoretical analysis.

Assumption 9. (Smooth parameterization) For all $s \in \mathcal{S}, a \in \mathcal{A}, g \in \mathcal{G}$ and $\theta \in \mathbb{R}^D$, $\|\nabla_\theta \log \pi_\theta(a|s, g)\|_2 \leq G_1$ and $\left\|\nabla_\theta^2 \log \pi_\theta(a|s, g)\right\|_2 \leq G_2$[4].

[4] See Appendix F for definitions of tensor norms and variance.

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 $\theta_1 = \theta$.
for $t = 1, 2...$ **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 $\theta'_{i,N} = \theta_t + \eta \frac{1}{N} \sum_{j=1}^N R(\tau_{i,j}, g_i) \nabla_\theta \log p_{\theta,g_i}(\tau_{i,j})$.
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 full estimate. Update outer loop $\theta_{t+1} = \theta_t + \alpha \hat{J}_{N,\text{LSF}}$.
end for
Output trained meta-RL policy π_θ.

In addition, we impose a smoothness condition on the value function. This could be converted into an equivalent assumption on the parameterization.

Assumption 10. (Smooth value function) For all $g \in \mathcal{G}, \theta \in \mathbb{R}^D$, $\left\|\nabla^3 V_g(\theta)\right\|_2 \leq L$.

All the above assumptions can be conveniently verified for e.g., tabular MDP (finite $\mathcal{X}, \mathcal{A}$ and $\mathcal{G}$) with soft-max parameterization of the policy, where $\pi_\theta(a|s, g) \propto \exp(\theta(s, a, g))$ with parameter $\theta = \{\theta(s, a, g)\}$.

5.2. Performance Guarantee

We now provide performance guarantees of the LSF gradient estimate. It is worth noting that since we are interested in the dependency on N, the analysis does not necessarily obtain the optimal dependency on other problem parameters (such as the parameter dimension D or horizon H). We leave potential improvements to future work.

The meta-RL objective takes an average over the parameter-independent distribution and hence its gradient $J_N(\theta) := \mathbb{E}_g[J_N(\theta, g)]$. As previously discussed, since $\mathbb{E}[\hat{J}_{N,\text{LSF}}(\theta, g)] \neq J_N(\theta, g)$, the generalized LSF gradient estimate $\hat{J}_{N,\text{LSF}}(\theta)$ is biased in general. We start by characterizing its bias against $J_N(\theta)$. Our results below characterize the dependency of various quantities on N, and folding other constants into $\mathcal{O}(1)$. See Appendix G for concrete dependencies on other constants in our analysis.

Proposition 11. For all values of the parameter $\theta \in \mathbb{R}^D$, $\left\|\mathbb{E}[\hat{J}_{N,\text{LSF}}(\theta)] - J_N(\theta)\right\|_2 = \mathcal{O}(1/\sqrt{N})$.

The bias is benign as it vanishes when N is large. We next characterize the variance of the estimate.

Proposition 12. For all $\theta \in \mathbb{R}^D$, $\mathbb{V}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] = \underbrace{\mathcal{O}(1/M) + \mathcal{O}(1/B)}_{\mathcal{O}(1)} + \mathcal{O}(1/N)$.

The bound $\mathcal{O}(1/M) + \mathcal{O}(1/B)$ means to show the dependency on the sample size B and M. When numerical quantities do not depend on N, they are considered $\mathcal{O}(1)$. The three terms on the upper bound above indicate sources of randomness that contribute the variance of the generalized LSF gradient estimate $\hat{J}_{N,\text{LSF}}(\theta)$: 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.

The bound is in general $\mathcal{O}(1)$ when N is large. This is because in general it is not possible to get rid of the variance induced by a finite B and M. However, when we let $B, M \to \infty$, the total variance is of order $\mathcal{O}(1/N)$. This is consistent with the variance of the LSF gradient estimate for the N-sample MC objective (see Lemma 6). Now we are ready present the convergence guarantee of Algorithm 1. We show its convergence to a stationary point of the objective $J_N(\theta)$.

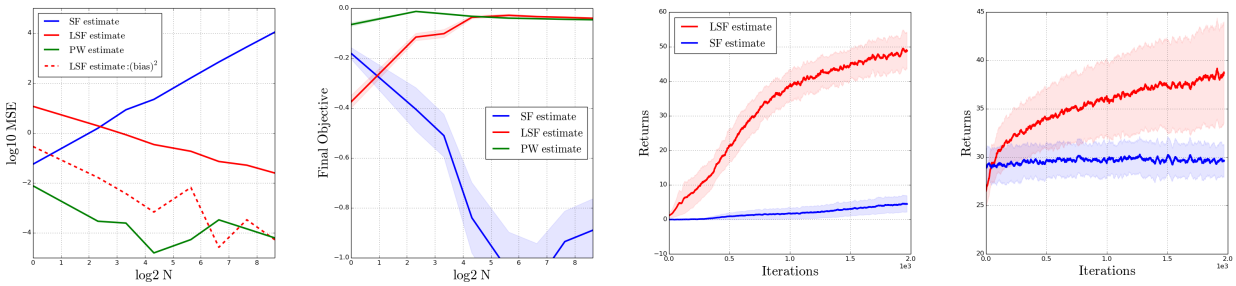
Proposition 13. With a properly chosen learning rate in Algorithm 1, for any $\epsilon > 0$, with $T_{\text{LSF}} = 2 \max\{\frac{1}{\epsilon^2 + \mathcal{O}(1/N)}, \frac{\mathcal{O}(1) + \mathcal{O}(1/N)}{\epsilon^4 + \mathcal{O}(1/N^2)}\}$ iterations of the algorithm, we have

$$\min_{1 \leq t \leq T_{\text{LSF}}} \mathbb{E}[\|J_N(\theta_t)\|_2^2] = \epsilon^2 + \mathcal{O}(1/N) =: \delta_{\text{LSF}}.$$

It is insightful to contrast with the result of Fallah et al. (2020a), where they they analyze the generalized SF gradient estimate. They show $T_{\text{SF}} = \mathcal{O}(1) \frac{1}{\alpha} \min\{\epsilon^{-2}, \Theta(N^{-2})\}$ and $\delta_{\text{SF}} = \epsilon^2 + \Theta(N^3 \alpha)$, where recall that α is the learning rate.

To see that the generalized LSF gradient estimate achieves a better dependency on N than the generalized SF gradient estimate, we fix ϵ and N, and adjust the learning rate α of the SF gradient estimate. First, we require the asymptotic error to have the same dependency on N by setting $\alpha = 1/N^4$, in which case $\delta_{\text{SF}} = \epsilon^2 + \Theta(1/N)$ while $\delta_{\text{LSF}} = \epsilon^2 + \mathcal{O}(1/N)$. This implies $T_{\text{SF}} = \mathcal{O}(1) N^4 \min\{\epsilon^{-2}, \Theta(N^{-2})\}$, which is significantly worse than T_{LSF} when N is large. Intuitively, this is because the generalized SF gradient estimate has much higher variance, which requires a very small α to achieve the same level of asymptotic error as the LSF gradient estimate. As a result, this takes the algorithm many more iterations to converge.

Equivalently, we can require both estimates to converge with the same number of iterations. Assuming ϵ is small enough such that N is the dominating factor in the asymp-



(a) Bias-variance trade-off (b) 1-D Optimization (c) Meta-RL HalfCheetah (d) Meta-RL Walker2D

Figure 1. (a) 1-D problem bias and variance trade-off: The MSE is computed at the initial update. At small N, LSF has higher MSE than SF. However, the MSE of SF increases with N due to variance; (b) 1-D problem learning curves: LSF outperforms SF. PW is the best when available. Each curve averages over 100 runs; (c)-(d) High-dimensional meta-RL problems: LSF outperforms SF consistently across all tasks. Each curve averages over 5 runs. See Appendix H for more results.

totic error, this means we should choose $\alpha = 1/N$. In this case, we have $\delta_{\text{SF}} = \epsilon^2 + \Theta(N^2)$, which is significantly worse than $\delta_{\text{LSF}} = \epsilon^2 + \mathcal{O}(1/N)$. This is because in order to converge with the same number of iterations, SF gradient estimate requires a relatively large learning rate. Large learning rates prevent the algorithm from dissipating the high variance in the gradient estimate, which ultimately lead to high asymptotic errors.

Convergence to stationary point of $F_\infty(\theta)$. Since $\|J_N(\theta) - J_\infty(\theta)\|_2 \leq \mathcal{O}(1/N)$ (Proposition 24), the above result also implies that with T_{LSF}, the algorithm also obtains the stationary point of $J_\infty(\theta)$ up to error $\epsilon^2 + \mathcal{O}(1/N)$. As such, we can interpret the optimization of $F_N(\theta)$ as a proxy to $F_\infty(\theta)$ when N is large.

6. Experiments

We carry out experiments to illustrate theoretical insights. We briefly discuss the results, see Appendix H for further details on the experiments.

6.1. 1-D optimization problem

To better understand the connection between the variance of the estimate and the optimization performance, we maximize a 1-D problem with objective $L(\theta) = \mathbb{E}_{(X_i)_{i=1}^N \sim p_\theta}[-(\bar{X}_N - 1)^2]$ where $\bar{X}_N$ is the average over N samples and $p_\theta = \mathcal{N}(\theta, 1^2)$. Fig 1(a) shows the bias-variance trade-off of different gradient estimates.

The mean-square error (MSE) of SF increases with N due to variance (because SF is unbiased), whereas the MSE of LSF decreases with N due to both decreasing variance and bias. Consistent with Lemma 2 and many empirical observations in prior work, PW achieves the lowest MSE due to low variance. Fig 1(b) compares the final objective after a fixed number of ascent updates. It shows as N increases, SF degrades significantly in performance whereas LSF improves. The performance of different estimates correlate strongly with the MSE: when N is small, LSF is outperformed by SF due to the bias; when N is large, LSF catches up with PW while the performance of SF degrades significantly. These observations are also consistent with discussions in Sec 5.

6.2. High-dimensional meta-RL problems

We contrast the SF and LSF gradient estimates in meta-RL. To implement both SF and LSF, we use the Markov structure to reduce variance compared to the original "trajectory-based" estimates. This turns out to be quite critical in practice. Fig 1(b)-(d) shows that the LSF gradient estimate outperforms the SF gradient estimate, where the high variance of the unbiased SF gradient estimate consistently hinders learning across all tasks we tested on.

At a first glance from results in Fig 1(d), the LSF estimate appears to exhibit higher variance than the SF estimate *across different runs*. We speculate that this is because under the SF estimate the algorithm barely learns, and consequently the difference of performance across independent runs is small, leading to small variance. On the other hand, the LSF estimate does provide better performance due to better stochastic gradient estimate with smaller variance *per run*. However, the optimization process overall is still fairly complicated, and different runs can achieve very different levels of performance (even though per-run they are all better than the SF estimate), leading to high variance across seeds.

Finally, we note that though the LSF estimate attempts to capture the variance reduction benefits of many practical algorithms with biased estimates (see also Sec 4 for further discussions), in practice it is common to introduce bisa to the estimates, to further improve empirical performance (Finn et al., 2017; Rothfuss et al., 2018; Tang et al., 2020). We leave the study of such observations to future work.

7. Conclusion

By formulating the N-sample meta-RL objective as a special case of N-sample additive MC objective, we identify the high variance ($\Theta(N)$) of naive SF gradient estimate. The LSF gradient estimate, which is biased but has low variance ($\mathcal{O}(1/N)$), achieves theoretical guarantees with much more benign dependency on N. As a result, our analysis suggests the necessity of employing biased gradient estimates in practice. Meanwhile, we also make the observation that many prior work turned out to implement variants of the LSF gradient estimate. This implies that despite their claim of unbiasedness, the practical gradient estimates are almost always biased. This is consistent with the empirical results. Overall, we believe our results help better understand the subtle design choices in meta-RL practice, and might entail the design of new algorithms in future work.

Acknowledgements. We thank the anonymous reviewers and meta-reviewer for helpful comments and suggestions. We also gratefully acknowledge the support of our DeepMind colleagues in the course of this work.

References

Ajalloeian, A. and Stich, S. U. (2020). Analysis of sgd with biased gradient estimators. *arXiv preprint arXiv:2008.00051*.

Al-Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. (2017). Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*.

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

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. (2016). Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Fakoor, R., Chaudhari, P., Soatto, S., and Smola, A. J. (2019). Meta-q-learning. *arXiv preprint arXiv:1910.00125*.

Fallah, A., Georgiev, K., Mokhtari, A., and Ozdaglar, A. (2020a). Provably convergent policy gradient methods for model-agnostic meta-reinforcement learning. *arXiv preprint arXiv:2002.05135*.

Fallah, A., Mokhtari, A., and Ozdaglar, A. (2020b). On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1082–1092. PMLR.

Farquhar, G., Whiteson, S., and Foerster, J. (2019). Loaded dice: Trading off bias and variance in any-order score function gradient estimators for reinforcement learning.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Foerster, J., Farquhar, G., Al-Shedivat, M., Rocktäschel, T., Xing, E., and Whiteson, S. (2018). Dice: The infinitely differentiable monte carlo estimator. In *International Conference on Machine Learning*, pages 1529–1538. PMLR.

Houthooft, R., Chen, R. Y., Isola, P., Stadie, B. C., Wolski, F., Ho, J., and Abbeel, P. (2018). Evolved policy gradients. *arXiv preprint arXiv:1802.04821*.

Ji, K., Yang, J., and Liang, Y. (2020). Multi-step model-agnostic meta-learning: Convergence and improved algorithms. *arXiv preprint arXiv:2002.07836*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

Liu, H., Socher, R., and Xiong, C. (2019). Taming maml: Efficient unbiased meta-reinforcement learning. In *International Conference on Machine Learning*, pages 4061–4071. PMLR.

Mao, J., Foerster, J., Rocktäschel, T., Al-Shedivat, M., Farquhar, G., and Whiteson, S. (2019). A baseline for any order gradient estimation in stochastic computation graphs. In *International Conference on Machine Learning*, pages 4343–4351. PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. (2020). Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21(132):1–62.

Oh, J., Hessel, M., Czarnecki, W. M., Xu, Z., van Hasselt, H., Singh, S., and Silver, D. (2020). Discovering reinforcement learning algorithms. *arXiv preprint arXiv:2007.08794*.

Ortega, P. A., Wang, J. X., Rowland, M., Genewein, T., Kurth-Nelson, Z., Pascanu, R., Heess, N., Veness, J., Pritzel, A., Sprechmann, P., et al. (2019). Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*.

Popoviciu, T. (1965). Sur certaines inégalités qui caractérisent les fonctions convexes. *Analele Stiintifice Univ."Al. I. Cuza", Iasi, Sectia Mat*, 11:155–164.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. (2019). Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR.

Ross, S. (2002). Simulation, 2002.

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., and Abbeel, P. (2018). Promp: Proximal meta-policy search. *arXiv preprint arXiv:1810.06784*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Tang, Y., Kozuno, T., Rowland, M., Munos, R., and Valko, M. (2021). Unifying gradient estimators for meta-reinforcement learning via off-policy evaluation. *arXiv preprint arXiv:2106.13125*.

Tang, Y., Valko, M., and Munos, R. (2020). Taylor expansion policy optimization. *arXiv preprint arXiv:2003.06259*.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., and De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*.

Xu, Z., van Hasselt, H., Hessel, M., Oh, J., Singh, S., and Silver, D. (2020). Meta-gradient reinforcement learning with an objective discovered online. *arXiv preprint arXiv:2007.08433*.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. (2019). Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*.

APPENDICES: Biased Gradient Estimate with Drastic Variance Reduction for Meta Reinforcement Learning

A. Definitions of "Big O" Notations

For non-negative functions f, g, the notation $f(N) = \mathcal{O}(g(N))$ implies that when N is large enough, there exists constant $C \geq 0$ such that $f(N) \leq Cg(N)$.

The notation $f(N) = \Theta(g(N))$ implies that N is large enough, there exists constant $C \geq c \geq 0$ such that $cg(N) \leq f(N) \leq Cg(N)$.

B. Reparameterized Gradient Estimate

If X is reparameterizable (RE), there exists an elementary distribution $\zeta \sim p_\zeta$ (e.g. gaussian $\mathcal{N}(0,1)$) and a function $\mathcal{T}_\theta$ such that $\mathcal{T}_\theta(\zeta)$ is equal in distribution to X.

When X is RE and f is differentiable, the PW gradient estimate is defined as

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

The PW gradient estimate is also unbiased. Intuitively, since PW gradient estimate makes use of the gradient $\nabla_X f(X)$, it enjoys lower variance compared to the SF gradient estimate in many applications (Kingma and Welling, 2013). 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.

C. Derivation of the Linearized Score Function Gradient Estimate

Since X_is are i.i.d., we expect the average $\bar{\phi}_N = \frac{1}{N}\sum_{i=1}^N \phi(X_i)$ to approach $\bar{\phi} \coloneqq \mathbb{E}\left[\phi(X_i)\right]$ as $N \to \infty$. This provides a direct motivation to consider the behavior of $f(\bar{\phi}_N)$ near the constant $\bar{\phi}$. In particular, consider the Taylor expansion of $f\left(\bar{\phi}_N\right)$ with $\bar{\phi}$ as its reference point. We decompose $f\left(\bar{\phi}_N\right)$ into three parts,

$$f\left(\bar{\phi}_N\right) = \underbrace{f\left(\bar{\phi}\right)}_{\text{constant term}} + \underbrace{\left[\nabla f\left(\bar{\phi}\right)\right]^T \left[\bar{\phi}_N - \bar{\phi}\right]}_{\text{linear term}} + \underbrace{\mathcal{O}\left(\left\|\bar{\phi} - \bar{\phi}_N\right\|_2^2\right)}_{\text{residual term}}$$

If we multiply the above with $\sum_{i=1}^N \nabla_\theta \log p_\theta(X_i)$, we recover the original SF gradient estimate on the LHS. Examining, the LHS, if we drop the residual term of the Taylor expansion, this yields a new estimate,

$$\underbrace{f\left(\bar{\phi}\right)\sum_{i=1}^N \nabla_\theta \log p_\theta(X_i)}_{\text{constant term}} + \underbrace{\left[\nabla f\left(\bar{\phi}\right)\right]^T \left[\bar{\phi}_N - \bar{\phi}\right]\left(\sum_{i=1}^N \nabla_\theta \log p_\theta(X_i)\right)}_{\text{linear term}}.$$

We will see that removing the residual term leads to a bias of order $\mathcal{O}(1/\sqrt{N})$ under some mild conditions on f.

Note now that the constant term has mean zero, but is nonzero in general. This implies that this term contributes a large portion of the total variance of this new estimate. It is therefore tempting to remove this term from the estimate. In fact, removing the constant term is equivalent to augmenting the original estimate with a baseline (or control variate) $\left(f(\bar{\phi}_N) - f(\bar{\phi})\right)\sum_{i=1}^N \nabla \log p_\theta(X_i)$ (Ross, 2002). We should expect the control variate to achieve significant variance reduction when N is large.

We now are left with the linear term alone as the new estimate. Note that if we count each $\nabla_\theta \log p_\theta(X_i)$ as a single term, there are a total of N^2 terms in the linear term. This is because we can write

$$\text{linear term} = \frac{1}{N}\sum_{i=1}^N \sum_{j=1}^N \left[\nabla f(\bar{\phi})\right]^T \left(\phi(X_j) - \bar{\phi}\right) \nabla_\theta \log p(X_j).$$

It is worth noting that the "off-diagonal" terms have mean zero. In other words,

$$\mathbb{E}\left[\left[\nabla f(\bar{\phi})\right]^T \left(\phi(X_i) - \bar{\phi}\right) \nabla_\theta \log p(X_j)\right] = 0, \;\; i \neq j.$$

We can remove all such terms, reducing the computations to only N "diagonal terms". We expect this removal step to reduce variance in practice as well, because it reduces the number of summations from N^2 to N. This produces a new estimate,

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

There are two places where $\bar{\phi}$ appears. Since $\bar{\phi}$ is not known in practice, we make two modifications to the above estimate: (1) We replace $f(\bar{\phi})$ by $f(\bar{\phi}_N)$. This will introduce further bias into the estimate; (2) We remove the second $\bar{\phi}$ for simplicity, as it will not impact the dependency of final estimates' bias and variance on N. Importantly, note that since $\bar{\phi}\nabla_\theta \log p(X_i)$ has mean zero, it does not change the mean of the overall estimate. However, since $\bar{\phi}$ can be seen as a control variate, its removal can potentially increase the variance of the estimate. Combining the above modifications, we arrive at the LSF gradient estimate,

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

D. Toy Example for Meta-RL

We construct a toy meta-RL example that illustrates the variance property of the SF gradient estimate vs. LSF gradient estimate. We consider a MDP where the value function $V_g(\theta) \coloneqq V_0$ is a constant for all θ. We also set the adaptation step size $\eta = 0$. We also assume a single starting state x_0, and that the trajectory terminates after $H = 1$ step. As a result, the trajectory consists of a single action $\tau_i \equiv a_i$. We can assume a 1-D action space $\mathcal{A} = \mathbb{R}$, and the policy $\pi(a|\theta, g) = \mathcal{N}(\theta, \sigma^2)$ is a Gaussian distribution with learnable mean θ and fixed σ. In this case, the effective objective is $L(\theta, g) = \mathbb{E}_{(a_i)_{i=1}^N \sim \mathcal{N}(\theta,\sigma^2)}[V_g(\theta)]$.

SF gradient estimate. In this case, the estimate is $V_g(\theta)\sum_{i=1}^N \nabla_\theta \log p_{\theta,g}(\tau_i) = V_0 \sum_{i=1}^N \nabla_\theta \log p_{\theta,g}(\tau_i)$. The SF gradient estimate bears close resemblance to the SF gradient estimate in the N-sample MC estimate. Its variance is of order $\Theta(N)$.

LSF gradient estimate. In this case, we can show that the LSF gradient estimate is effectively the PG estimate at the adapted parameter $\hat{\nabla} V_g(\theta')$ where $\theta' = \theta$. It then depends on how we construct the PG estimate. For example, since $\theta' = \theta$, we can use the N samples generated under θ to estimate the PG estimate. It then naturally follows that the variance is of order $\mathcal{O}(1/N)$.

E. Discussion on Prior Work

Here, we provide a more extended discussion on prior work.

N-sample meta-RL objective. As noted earlier, the N-sample meta-RL objective was considered in both empirical (Al-Shedivat et al., 2017; Rothfuss et al., 2018) and theoretical contexts (Fallah et al., 2020a). 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 (Finn et al., 2017).

Unbiased gradient to the limit case $J_\infty(\theta, g)$. In the author's original implementation of the MAML gradient estimate (Finn et al., 2017), 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 (Foerster et al., 2018; Liu et al., 2019). However, they are *biased* in general. This is mainly because practical algorithms can only estimate $\nabla_g V_g(\theta'_N)$ instead of $\nabla_g V_g(\theta')$, while the latter is required to estimate $J_\infty(\theta, g)$ in an unbiased way. This observation was also hinted at recently in (Tang et al., 2021).

Prior work in fact constructs the LSF gradient estimate. Since most prior work derive meta-RL gradient estimates based on $J_\infty(\theta, g)$ (Foerster et al., 2018; Rothfuss et al., 2018; Liu et al., 2019; Tang et al., 2021), 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 8). 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 $\hat{H}_N(\theta)$. Though in theory reducing the variance of $\hat{H}_N(\theta)$ does not necessarily guarantee improvements, in practice, this seems to be very critical. Variance reduction methods include control variates (Liu et al., 2019), as well as introducing further bias to the estimate of $\hat{H}_N(\theta)$ (Rothfuss et al., 2018; Tang et al., 2021).

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 (Al-Shedivat et al., 2017; Rothfuss et al., 2018; Fallah et al., 2020a). A comprehensive derivation was carried out in (Rothfuss et al., 2018), where they contrasted $J_\infty(\theta, g)$ with $J_N(\theta, g)$. However, they claimed that $J_\infty^{(ii)}(\theta, g) = J_N^{(ii)}(\theta, g)$, which 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. Due to the highly complex objective landscape of meta learning, most theoretical analysis focuses on convergence to stationary points. Recently, (Fallah et al., 2020b) established generic convergence guarantees for gradient-based meta-learning algorithms for supervised learning with one inner loop update. Recently, (Ji et al., 2020) extended the analysis to multi-step inner loop updates. For meta-RL, (Fallah et al., 2020a) established convergence for the N-sample meta-RL objective. They motivated the objective in a similar manner as (Al-Shedivat et al., 2017; Rothfuss et al., 2018) and constructed unbiased estimates exactly as the generalized SF gradient estimate $\hat{J}_{N,\mathrm{SF}}(\theta, g)$. 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 gradient estimate achieves performance guarantee with more desirable dependency on N.

F. Notations for Norms and Useful Inequalities

For any tensor (vector or matrix) X, we define its 2-norm as

$$\|X\|_2 \coloneqq \sqrt{\sum_i X_i^2},$$

where i sums over components of X. The variance is defined as the sum of the variance of its components,

$$\mathbb{V}[X] \coloneqq \sum_i \mathbb{V}\left[X_i\right].$$

We now introduce a number of useful inequalities, which will be heavily used in the proof section.

Operator norm and 2-norm for matrix. The oprator norm of a matrix X is defined as: $\|X\|_{\mathrm{op},2} \coloneqq \max_{\|u\|_2=1} \|Xu\|_2$. It is known that $\|X\|_{\mathrm{op},2} \leq \|X\|_2$.

Exchange of norm and expectation. For any random tensor X, $\|\mathbb{E}\left[X\right]\|_2 \leq \mathbb{E}\left[\|X\|_2\right]$. The proof is based on Jensen's inequality and the fact that 2-norm is a convex function of its argument.

Cauchy–Schwarz (CS) inequality for random variables. For any two random variables X, Y,

$$\mathbb{E}\left[|X||Y|\right] \leq \sqrt{\mathbb{E}[X^2]}\sqrt{\mathbb{E}[Y^2]}.$$

CS inequality for random matrix and vector. For any random matrix $X \in \mathbb{R}^{H\times H}$ and random vector $Y \in \mathbb{R}^H$, then

$$\|\mathbb{E}\left[XY\right]\|_2 \leq \mathbb{E}\left[\|XY\|_2\right] \leq \mathbb{E}\left[\|X\|_{\mathrm{op},2}\|Y\|_2\right] \leq \mathbb{E}\left[\|X\|_2\|Y\|_2\right] \leq \sqrt{\mathbb{E}\left[\|X\|_2^2\right]}\sqrt{\mathbb{E}\left[\|Y\|_2^2\right]}.$$

The last inequality comes from the CS inequality of scalar random variables.

Expected norm inequality. For any random tensor X, we have

$$\mathbb{E}[\|X\|_2] \leq \sqrt{\mathbb{E}[\|X\|_2^2]}.$$

The result follows by considering $Y \coloneqq \|X\|_2$ as a single random variable, we have $\mathbb{E}[Y] \leq \sqrt{\mathbb{E}[Y^2]}$.

G. Proof

Lemma 14. In the toy N-sample MC objective example,

$$\mathbb{V}\left[\hat{\nabla}_\theta^{\text{SF}} L(\theta)\right] = \Theta(N), \mathbb{V}\left[\hat{\nabla}_\theta^{\text{LSF}} L(\theta)\right] = \mathcal{O}(1/N), \mathbb{V}\left[\hat{\nabla}_\theta^{\text{PW}} L(\theta)\right] = 0.$$

Proof. We first reparameterize the random variable $X_i = \mu + \sigma \cdot \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0,1)$. With some calculations, we can show,

$$\hat{\nabla}_\theta^{\text{SF}} L(\theta) = \frac{1}{N}\left(\sum_{i=1}^N \epsilon_i\right)^2 + \frac{\mu}{\sigma}\frac{1}{N}\sum_{i=1}^N \epsilon_i,\ \hat{\nabla}_\theta^{\text{LSF}} L(\theta) = \frac{1}{N}\sum_{i=1}^N \epsilon_i^2 + \frac{\mu}{\sigma}\sum_{i=1}^N \epsilon_i,\ \hat{\nabla}_\theta^{\text{PW}} L(\theta) = 1.$$

It then follows that

$$\mathbb{V}\left[\hat{\nabla}_\theta^{\text{SF}} L(\theta)\right] = \frac{\mu^2}{\sigma^2}N + \frac{2}{N} = \Theta(N),\ \mathbb{V}\left[\hat{\nabla}_\theta^{\text{LSF}} L(\theta)\right] = \frac{\mu^2}{\sigma^2}\frac{1}{N} + \frac{2}{N} = \mathcal{O}(1/N),\ \mathbb{V}\left[\hat{\nabla}_\theta^{\text{PW}} L(\theta)\right] = 0.$$

□

Lemma 15. In the toy N-sample MC objective example, $\mathbb{V}\left[\hat{\nabla}_\theta^{\text{SF}} L(\theta)\right] = \Theta(N)$.

Proof. See the proof for Lemma 14. □

Lemma 16. In the toy N-sample MC objective example, $\mathbb{V}\left[\hat{\nabla}_\theta^{\text{PW}} L(\theta)\right] = 0$.

Proof. See the proof for Lemma 14. □

Lemma 17. 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.,

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

Proof. Recall the definition of $G(\theta)$,

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

The objective depends on θ in a few ways. It is straightforward to see that term (i) results from the fact that $X_i \sim p_\theta$. Another source of dependency is through the argument $\phi(X, \theta)$ and $f(\bar{\phi}_N, \theta)$. Fixing X_is, taking partial gradient of f with respect to θ, we get from chain rule,

$$\nabla f(\bar{\phi}_N, \theta) = \nabla_\theta f(\bar{\phi}_N, \theta) + \nabla_\theta \bar{\phi}_N \nabla_{\bar{\phi}_N} f(\bar{\phi}_N, \theta).$$

Expanding $\nabla_\theta \bar{\phi}_N$, we get the desired expression. □

Lemma 18. Let $\theta'_N \coloneqq \theta + \eta \frac{1}{N} \sum_{i=1}^N R(\tau_i, g) \nabla_\theta \log p_{\theta,g}(\tau_i)$ be the (random) updated parameter. Let $\tau_i \sim p_{\theta,g}$ i.i.d. and let $\nabla V_g(\theta'_N)$ denotes $[\nabla_\theta V_g(\theta)]_{\theta=\theta'_N}$. Then $J_N(\theta, g) \coloneqq \nabla_\theta F_N(\theta, g)$ is

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

Proof. As discussed in the main text, with the conversion: $X_i \coloneqq \tau_i, \phi(X_i, \theta) \coloneqq R(\tau_i, g) \nabla_\theta \log p_{\theta,g}(\tau_i)$ 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. By using the gradient of the N-sample additive MC objective shown in Lemma 3, we get the desired result. □

Corollary 19. First, sample $(\tau_i)_{i=1}^N \sim p_{\theta,g}$ and compute 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[5]. The final estimate is

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

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

Proof. Given θ'_N, as assumed, we can construct unbiased estimates $\nabla \hat{V}_g(\theta'_N)$ and $\hat{V}_g(\theta'_N)$ to $\nabla V_g(\theta'_N)$ and $V_g(\theta'_N)$. This is equivalent to the following statement,

$$\mathbb{E}\left[\nabla \hat{V}_g(\theta'_N) \,\middle|\, \theta'_N\right] = \nabla V_g(\theta'_N), \mathbb{E}\left[\hat{V}_g(\theta'_N) \,\middle|\, \theta'_N\right] = V_g(\theta'_N).$$

We now have the following

$$\begin{aligned}
&\mathbb{E}\left[\hat{V}_g(\theta'_N) \sum_{i=1}^N \nabla_\theta \log p_{\theta,g}(\tau_i) \left(I + \eta \frac{1}{N} \sum_{i=1}^N R(\tau_i, g) \nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right) \nabla \hat{V}_g(\theta'_N)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\hat{V}_g(\theta'_N) \sum_{i=1}^N \nabla_\theta \log p_{\theta,g}(\tau_i) \left(I + \eta \frac{1}{N} \sum_{i=1}^N R(\tau_i, g) \nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right) \nabla \hat{V}_g(\theta'_N) \,\middle|\, \theta'_N\right]\right] \\
&= \mathbb{E}\left[V_g(\theta'_N) \sum_{i=1}^N \nabla_\theta \log p_{\theta,g}(\tau_i) \left(I + \eta \frac{1}{N} \sum_{i=1}^N R(\tau_i, g) \nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right) \nabla V_g(\theta'_N)\right] = J_N(\theta, g).
\end{aligned}$$

This shows that the finale estimate is unbiased. □

Lemma 20. In the toy N-sample MC objective example, $\mathbb{V}\left[\hat{\nabla}_\theta^{\mathrm{LSF}} L(\theta)\right] = \mathcal{O}(1/N)$.

[5] For now, we just require the estimates to be unbiased. In Sec 5, we make these estimates concrete for refined convergence analysis.

Proof. See the proof for Lemma 14. □

Corollary 21. Let $u_i \coloneqq \nabla_\theta \log p_{\theta,g}(\tau_i)$. The generalized LSF gradient estimate $\hat{J}_{N,\text{LSF}}(\theta, g)$ to $F_N(\theta, g)$ can be expressed as follows,

$$\left(\eta \frac{1}{N} \sum_{i=1}^{N} R(\tau_i, g) u_i u_i^T\right) \nabla V_g(\theta'_N) \left(I + \eta H_\theta\right) \nabla V_g(\theta'_N),$$

where $H_\theta = \frac{1}{N} \sum_{i=1}^{N} R(\tau_i, g) \nabla_\theta^2 \log p_{\theta,g}(\tau_i)$. Define $\nabla \hat{V}_g, \hat{V}_g$ as unbiased estimates to $\nabla V_g, V_g$. The following estimate has the same expectation as $\hat{J}_{N,\text{LSF}}(\theta, g)$,

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

Proof. With the conversion $X_i \coloneqq \tau_i, \phi(X_i, \theta) \coloneqq R(\tau_i, g) \nabla_\theta \log p_{\theta,g}(\tau_i)$ and $f(\bar{\phi}_N, \theta) = V_g(\theta + \eta \bar{\phi}_N)$, we can derive the generalized LSF gradient estimate to the meta-RL objective as a special instance of Eqn 10,

$$\left(\eta \frac{1}{N} \sum_{i=1}^{N} R(\tau_i, g) u_i u_i^T\right) \nabla V_g(\theta'_N) \left(I + \eta \frac{1}{N} \sum_{i=1}^{N} R(\tau_i, g) \nabla_\theta^2 \log p_{\theta,g}(\tau_i)\right) \nabla V_g(\theta'_N).$$

For the second part of the result, by replacing V_g and ∇V_g by their unbiased estimate, we can show that the overall estimate has the same expectation. This is because $\mathbb{E}[\hat{V}_g(\theta'_N) | \theta'_N] = V_g(\theta'_N)$ and $\mathbb{E}[\nabla \hat{V}_g(\theta'_N) | \theta'_N] = \nabla V_g(\theta'_N)$, we can show the desired result via the law of total expectation as in the proof of Lemma 5. □

Corollary 22. Let $u_i \coloneqq \nabla_\theta \log p_{\theta,g}(\tau_i)$ 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

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

Proof. We directly compute the gradient $J_\infty(\theta, g) = \nabla_\theta L(\theta, g)$ via chain rule,

$$J_\infty(\theta, g) = \left(I + \eta \nabla^2 V_g(\theta)\right) \nabla V(\theta').$$

Now, if we write $V_g(\theta) = \mathbb{E}_\tau\left[R(\tau, g)\right]$ with $\tau \sim p_{\theta,g}$. We can compute its Hessian,

$$\begin{aligned} \nabla^2 V_g(\theta) &= \mathbb{E}_\tau\left[R(\tau, g) \nabla_\theta^2 \log p_{\theta,g}(\tau) + R(\tau, g) \nabla_\theta \log p_{\theta,g}(\tau) \left(\nabla_\theta \log p_{\theta,g}(\tau)\right)^T\right] \\ &= \mathbb{E}_{(\tau_i)_{i=1}^N}\left[\frac{1}{N} \sum_{i=1}^{N} R(\tau_i, g) \nabla_\theta^2 \log p_{\theta,g}(\tau_i) + \frac{1}{N} \sum_{i=1}^{N} R(\tau_i, g) \nabla_\theta \log p_{\theta,g}(\tau_i) \left(\nabla_\theta \log p_{\theta,g}(\tau_i)\right)^T\right]. \end{aligned}$$

Rearranging Eqn 12, we get the desired result.

□

Proposition 23. For all θ, θ' and g, the value function satisfies the smoothness condition $\left\|\nabla^2 V_g(\theta)\right\|_2 \leq L_1$. The following variances are bounded,

$$\mathbb{V}\left[R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\right] \leq \sigma_1^2, \mathbb{V}\left[R(\tau,g)\nabla_\theta^2 \log p_{\theta,g}(\tau)\right] \leq \sigma_{21}^2,$$
$$\mathbb{V}\left[R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)^T\right] \leq \sigma_{22}^2,$$
$$\mathbb{E}\left[\left\|R(\tau_i)\nabla_\theta \log p_{\theta,g}(\tau_i) - \mathbb{E}\left[R(\tau_i)\nabla_\theta \log p_{\theta,g}(\tau_i)\right]\right\|_2^3\right] \leq \sigma_3^3.$$

All quantities above $L_1, \sigma_1^2, \sigma_{21}^2, \sigma_{22}^2, \sigma_3^2$ can be expressed as functions of $(R, \gamma, H, G_1, G_2, D)$.

Proof. First, consider L_1. To express L_1 as a function of $(R, \gamma, H, G_1, G_2, D)$, note that we can derive an upper bound for all θ, g and τ,

$$\left\|R(\tau,g)\nabla_\theta^2 \log p_{\theta,g}(\tau) + R(\tau,g)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)^T\right\|_2 \leq RG_2H\frac{1-\gamma^H}{1-\gamma} + RG_1^2H^2\frac{1-\gamma^H}{1-\gamma}.$$

This implies

$$\begin{aligned}\left\|\nabla^2 V_g(\theta)\right\|_2 &= \left\|\mathbb{E}_\tau\left[R(\tau,g)\nabla_\theta^2 \log p_{\theta,g}(\tau) + R(\tau,g)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)^T\right]\right\|_2 \\ &\leq \mathbb{E}_\tau\left[\left\|R(\tau,g)\nabla_\theta^2 \log p_{\theta,g}(\tau) + R(\tau,g)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)^T\right\|_2\right] \\ &\leq RG_2H\frac{1-\gamma^H}{1-\gamma} + RG_1^2H^2\frac{1-\gamma^H}{1-\gamma}.\end{aligned}$$

Then we can write

$$\left\|\nabla V_g(\theta) - V_g(\theta')\right\|_2 = \left\|\nabla^2 V_g(\tilde{\theta})(\theta-\theta')\right\|_2 \leq \left\|\nabla^2 V_g(\tilde{\theta})\right\|_{\text{op},2}\left\|\theta-\theta'\right\|_2 \leq \left\|\nabla^2 V_g(\tilde{\theta})\right\|_2\left\|\theta-\theta'\right\|_2 \leq L_1\left\|\theta-\theta'\right\|_2.$$

Hence, we can set $L_1 = RG_2H\frac{1-\gamma^H}{1-\gamma} + RG_1^2H^2\frac{1-\gamma^H}{1-\gamma}$.

Regarding the variances, note that since all the random variables

$$R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau),\ R(\tau,g)\nabla_\theta^2 \log p_{\theta,g}(\tau),\ R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\left(\nabla_\theta \log p_{\theta,g}(\tau)\right)^T,$$

are bounded almost surely (the bounds are a function of $(R, \gamma, H, G_1, G_2, D)$), their variances are also bounded, and can be expressed as a function of such bounds. As one way to derive such bounds, we can upper bound each entry of the random tensor. For example, take $R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)$ as an example, we can write $\left|\left[R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\right]_i\right| \leq RG_1H\frac{1-\gamma^H}{1-\gamma}$ for all component i. For any random variable X such that $|X| \leq C$, we have $\mathbb{V}[X] \leq C^2$ (Popoviciu, 1965). This implies $\mathbb{V}\left[R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\right] \leq D \cdot \left(RG_1H\frac{1-\gamma^H}{1-\gamma}\right)^2$ and we can set $\sigma_1 = \sqrt{D} \cdot RG_1H\frac{1-\gamma^H}{1-\gamma}$. We refer to such bounds as the loose bounds.

Such bounds might not have an optimal dependency on $(R, \gamma, H, G_1, G_2, D)$. For example, since for all θ, g and τ, we can bound

$$\left\|R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\right\|_2 \leq RG_1H\frac{1-\gamma^H}{1-\gamma}.$$

This implies that the random vector $R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)$ has bounded norm almost surely. By definition of the vector variance, this also implies that σ_1^2 is bounded. We can bound

$$\mathbb{V}\left[R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\right] \leq \mathbb{E}\left[\left\|R(\tau,g)\nabla_\theta \log p_{\theta,g}(\tau)\right\|_2^2\right] \leq \left(RG_1H\frac{1-\gamma^H}{1-\gamma}\right)^2.$$

We can hence set $\sigma_1 = RG_1H\frac{1-\gamma^H}{1-\gamma}$, which is an improvement over the naive approach with a factor of $\sqrt{D}$. Nevertheless, it is straightforward to derive the loose bounds for $\sigma_{21}^2, \sigma_{22}^2$ and σ_3^2 and conclude the result, though tighter bounds require more refined analysis.

□

Proposition 24. For all $\theta \in \mathbb{R}^D$, $\left\| \mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] - J_\infty(\theta) \right\|_2 \leq \mathcal{O}(1/\sqrt{N})$.

Proof. Since $\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] - J_\infty(\theta) = \mathbb{E}_g\left[\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta, g)\right] - J_\infty(\theta, g)\right]$, we focus on the bias of the task-conditioned bias $\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta, g)\right] - J_\infty(\theta, g)$. Henceforth we will suppress the dependency of the trajectories on the task variable, still denoting the N trajectories as $(\tau_i)_{i=1}^N$. We write $\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta, g)\right] - J_\infty(\theta, g)$ as follows,

$$\mathbb{E}_{(\tau_i)_{i=1}^N}\left[\left(I + \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\nabla_\theta^2 \log p_{\theta,g}(\tau_i) + \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i, g)u_i u_i^T\right)\left(\nabla V_g(\theta'_N) - \nabla V_g(\theta')\right)\right].$$

For notational simplicity, we define the following

$$X_N \coloneqq \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i, g)\nabla_\theta^2 \log p_{\theta,g}(\tau_i) + \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i, g)u_i u_i^T$$
$$Y_N \coloneqq \nabla V_g(\theta'_N) - \nabla V_g(\theta').$$

Note that $X \coloneqq \mathbb{E}[X_N] = \eta\nabla_\theta^2 V_g(\theta)$. We write the above difference as

$$\mathbb{E}\left[(I + X_N)Y_N\right] = \underbrace{\mathbb{E}\left[(I + X)Y_N\right]}_{\text{part (i)}} + \underbrace{\mathbb{E}[(X_N - X)Y_N]}_{\text{part (ii)}}.$$

To bound the norms of each term, note that we have due to the Lipschitz smoothness of the value function parameterization,

$$\mathbb{E}\left[\|Y_N\|_2^2\right] \leq L_1^2\mathbb{E}\left[\|\theta'_N - \theta'\|_2^2\right] \leq L_1^2\frac{\eta^2\sigma_1^2}{N}.$$

The Lipschitz smoothness also implies $\|X\|_2 = \eta\left\|\nabla_\theta^2 V_g(\theta)\right\|_2 \leq \eta L_1$. This entails a bound on part (i) of the difference,

$$\|\text{part (i)}\|_2 \leq \|(I + X)\mathbb{E}[Y_N]\|_2 \leq \|I + X\|_{\text{op},2}\|\mathbb{E}[Y_N]\|_2 \leq (1 + \eta L_1)\sqrt{\mathbb{E}\left[\|Y_N\|_2^2\right]} \leq (1 + \eta L_1)L_1\frac{\eta\sigma_1}{\sqrt{N}}.$$

We have exchanged the norms and expectation, and applied the expected norm inequality.

To bound the second part, first note that for any two random variables X, Y, the sum of the variance is upper bounded as: $\mathbb{V}[X + Y] \leq 2\mathbb{V}[X] + 2\mathbb{V}[Y]$. This inequality extends to general random tensor X, Y. This implies

$$\mathbb{V}\left[R(\tau_i, g)\nabla_\theta^2 \log p_{\theta,g}(\tau_i) + R(\tau_i, g)\nabla_\theta \log p_{\theta,g}(\tau_i)\left(\nabla_\theta \log p_{\theta,g}(\tau_i)\right)^T\right] \leq \sigma_{21}^2 + \sigma_{22}^2$$

This further implies,

$$\mathbb{E}\left[\|X_N - X\|_2^2\right] = \mathbb{V}\left[X_N\right] \leq \frac{\eta^2}{N}(\sigma_{21}^2 + \sigma_{22}^2).$$

Before bounding the norm of part (ii), we recall that that for any matrix X and vector Y, we have $\|XY\|_2 \leq \|X\|_{\text{op},2}\|Y\|_2$, and that $\|X\|_{\text{op},2} \leq \|X\|_2$. Now, we can upper bound part (ii) as follows,

$$\|\text{part (ii)}\|_2 \leq \mathbb{E}\left[\|(X_N - X)Y_N\|_2\right] \leq \mathbb{E}\left[\|(X_N - X)\|_{\text{op},2}\|Y_N\|_2\right] \leq \mathbb{E}\left[\|(X_N - X)\|_2\|Y_N\|_2\right].$$

The final RHS is upper bounded by the following due to Cauchy–Schwarz inequality, which implies

$$\|\text{part (ii)}\|_2 \leq \ldots \leq \sqrt{\mathbb{E}\left[\|(X_N - X)\|_2^2\right]}\sqrt{\mathbb{E}\left[\|Y_N\|_2^2\right]} \leq \frac{\eta}{\sqrt{N}}\sqrt{\sigma_{21}^2 + \sigma_{22}^2} \cdot L_1\frac{\eta\sigma_1}{\sqrt{N}}.$$

Combining the two results above with a triangle inequality due to the vector 2-norm, we have

$$\left\|\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta, g)\right] - J_\infty(\theta, g)\right\|_2 \leq (1 + \eta L_1)\frac{L_1\eta\sigma_1}{\sqrt{N}} + \sqrt{\sigma_{21}^2 + \sigma_{22}^2}\frac{L_1\eta^2\sigma_1}{N}.$$

This induces the final result,

$$\left\|\mathbb{E}[\hat{J}_{N,\text{LSF}}(\theta)] - J_\infty(\theta)\right\|_2 \leq \mathbb{E}_g\left[\left\|\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta,g)\right] - J_\infty(\theta,g)\right\|_2\right] \leq (1+\eta L_1)\frac{L_1\eta\sigma_1}{\sqrt{N}} + \sqrt{\sigma_{21}^2+\sigma_{22}^2}\frac{L_1\eta^2\sigma_1}{N}.$$

□

Proposition 25. For all $\theta \in \mathbb{R}^D$, $\|J_\infty(\theta) - J_N(\theta)\|_2 \leq \mathcal{O}(1/\sqrt{N})$.

Proof. We seek to bound the difference $\left\|\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] - J_N(\theta)\right\|_2$. We first focus on bounding task-conditional gradient $\left\|\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta,g)\right] - J_N(\theta,g)\right\|_2$. To this end, recall that $u_i \coloneqq \nabla_\theta \log p_{\theta,g}(\tau_i)$, then we can write

$$\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta,g)\right] - J_N(\theta,g) = \mathbb{E}_{(\tau_i)_{i=1}^N}\left[\left(\eta\frac{1}{N}\sum_{i=1}^N R(\tau_i,g)u_iu_i^T\right)\nabla V_g(\theta'_N)\right] - \mathbb{E}_{(\tau_i)_{i=1}^N}\left[\left(\eta\sum_{i=1}^N u_i\right)V_g(\theta'_N)\right].$$

We start with Taylor expansion of $V_g(\theta'_N)$ with respect to the reference point $\theta' \coloneqq \theta + \eta\mathbb{E}[R(\tau,g)\nabla\log p_{\theta,g}(\tau)]$.

$$V_g(\theta'_N) = V_g(\theta') + \left(\nabla V_g(\theta')\right)^T(\theta'_N - \theta') + 1/2\cdot(\theta'_N-\theta')^T\nabla^2 V_g(\theta)(\theta'_N-\theta')^T + 1/6\cdot\nabla^3 V_g(\tilde{\theta})(\theta'_N-\theta')^3,$$

where $\tilde{\theta}$ is a random vector between θ'_N and θ'. Here, for $A \in \mathbb{R}^{H\times H\times H}$ and $x\in\mathbb{R}^H$ we define the notation $Ax^3 \coloneqq \sum_{ijk}A_{ijk}x_ix_jx_k$. Plugging in the expansion, $\mathbb{E}\left[\hat{J}_{N,\text{LSF}}(\theta,g)\right] - J_N(\theta,g)$ evaluates to

$$\begin{aligned}
&= \mathbb{E}\left[\left(\eta\frac{1}{N}\sum_{i=1}^N R(\tau_i,g)u_iu_i^T\right)\nabla V_g(\theta'_N)\right] - \underbrace{\mathbb{E}\left[\left(\eta\sum_{i=1}^N u_i\right)V_g(\theta')\right]}_{=0}\\
&\quad - \mathbb{E}\left[\left(\eta\sum_{i=1}^N u_i\right)\left(\nabla V_g(\theta')\right)^T(\theta'_N-\theta')\right] - \mathbb{E}\left[\left(\eta\sum_{i=1}^N u_i\right)1/2\cdot(\theta'_N-\theta')^T\nabla^2V_g(\tilde{\theta})(\theta'_N-\theta')^T\right]\\
&= \underbrace{\mathbb{E}\left[\left(\eta\frac{1}{N}\sum_{i=1}^N R(\tau_i,g)u_iu_i^T\right)\left(\nabla V_g(\theta'_N)-\nabla V_g(\theta')\right)\right]}_{\text{part (i)}} - \underbrace{\mathbb{E}\left[\left(\eta\sum_{i=1}^N u_i\right)1/2\cdot(\theta'_N-\theta')^T\nabla^2V_g(\theta')(\theta'_N-\theta')^T\right]}_{\text{part (ii)}}\\
&\quad - \underbrace{\mathbb{E}\left[\left(\eta\sum_{i=1}^N u_i\right)1/6\cdot\nabla^3V_g(\tilde{\theta})(\theta'_N-\theta')^3\right]}_{\text{part (iii)}}.
\end{aligned}$$

Below, we bound each of the three parts above. For part (i), let $X_N = \eta\frac{1}{N}\sum_{i=1}^N R(\tau_i,g)u_iu_i^T$ and $Y_N = \nabla V_g(\theta'_N) - \nabla V_g(\theta')$. Let $X = \mathbb{E}[X_N]$. Then,

$$\mathbb{E}[\|X_N\|_2^2] = \|\mathbb{E}[X_N]\|_2^2 + \mathbb{V}[X_N].$$

To obtain a bound of the norm on $\mathbb{E}[X_N]$, let $u = \nabla_\theta\log p_{\theta,g}(\tau)\left(\nabla_\theta\log p_{\theta,g}(\tau)\right)^T$ for any τ, then

$$\left\|R(\tau,g)uu^T\right\|_2 \leq R\frac{1-\gamma^H}{1-\gamma}\left\|uu^T\right\|_2 = R\frac{1-\gamma^H}{1-\gamma}\sqrt{\text{Trace}(uu^Tuu^T)} = R\frac{1-\gamma^H}{1-\gamma}(u^Tu) \leq H^2G^2R\frac{1-\gamma^H}{1-\gamma}.$$

In the above we used the assumption that $\|\nabla_\theta\log\pi_\theta(a|s,g)\|_2 \leq G$. This implies

$$\|\mathbb{E}[X_N]\|_2 = \eta\left\|\mathbb{E}[R(\tau,g)uu^T]\right\|_2 \leq \eta\mathbb{E}\left[\left\|R(\tau,g)uu^T\right\|_2\right] \leq \eta H^2G^2R\frac{1-\gamma^H}{1-\gamma}.$$

We hence conclude

$$\mathbb{E}[\|X_N\|_2^2] \leq \eta^2 H^4 G^4 R^2 \left(\frac{1-\gamma^H}{1-\gamma}\right)^2 + \frac{\eta^2\sigma_{22}^2}{N}.$$

Combining with the previous result on the norm bound of $\mathbb{E}[\|Y_N\|_2^2]$, we get via the CS inequality,

$$\|\text{part (i)}\|_2 \leq \sqrt{\mathbb{E}[\|X_N\|_2^2]}\sqrt{\mathbb{E}[\|Y_N\|_2^2]} \leq \sqrt{\eta^2 H^4 G^4 R^2 \left(\frac{1-\gamma^H}{1-\gamma}\right)^2 + \frac{\eta^2\sigma_{22}^2}{N}} \cdot L_1 \frac{\eta\sigma_1}{N}.$$

Now, consider part (ii). For notational simplicity, we denote $x_i \coloneqq R(\tau_i, g)\nabla_\theta \log p_{\theta,g}(\tau_i)$ and $y_i \coloneqq \nabla_\theta \log p_{\theta,g}(\tau_i)$. Let $A \coloneqq \nabla^2 V_g(\theta + \eta\mu)$ be the Hessian.

$$\begin{aligned}\text{part (ii)} &= \mathbb{E}\left[\frac{1}{2}\left(\frac{1}{N}\sum_i x_i - \mu\right)^T A \left(\frac{1}{N}\sum_i x_i - \mu\right)\sum_i y_i\right] \\ &= \mathbb{E}\left[\frac{1}{2}\frac{1}{N^2}\sum_{ijk} x_i^T A x_j y_k - \frac{1}{2}\frac{1}{N}\left(\sum_{ij} x_i^T A \mu y_j + \sum_{ij}\mu^T A x_i y_j\right)\right].\end{aligned}$$

Recall that $\mathbb{E}[y_i] = 0$. We can simplify the above as follows

$$\mathbb{E}\left[\frac{1}{2}\frac{1}{N^2}\left(\sum_{j\neq i} x_i^T A x_j y_i + \sum_{j\neq i} x_i^T A x_j y_j + \sum_i x_i^T A x_i y_i\right) - \frac{1}{2}\frac{1}{N}\left(\sum_i x_i^T A\mu y_i + \sum_i \mu^T A x_i y_i\right)\right].$$

Let $\mu_1 \coloneqq \mathbb{E}\left[x_i^T A\mu y_i\right]$, $\mu_2 \coloneqq \mathbb{E}\left[\mu^T A x_i y_i\right]$ and $\mu_3 \coloneqq \mathbb{E}[x_i^T A x_i y_i]$. Then

$$\begin{aligned}\left\|\mathbb{E}\left[\frac{1}{N^2}\sum_{j\neq i} x_i^T A x_j y_i\right] - \frac{1}{N}\sum_i x_i^T A x_j y_i\right\|_2 &= \frac{1}{N}\|\mu_1\|_2 \\ \left\|\mathbb{E}\left[\frac{1}{N^2}\sum_{j\neq i} x_i^T A x_j y_j\right] - \frac{1}{N}\sum_i \mu^T A x_i y_i\right\|_2 &= \frac{1}{N}\|\mu_2\|_2. \\ \left\|\frac{1}{N^2}\sum_i x_i^T A x_i y_i\right\|_2 &\leq \frac{1}{N}\|\mu_3\|_2.\end{aligned}$$

Overall, we can bound the norm of part (ii) by $\frac{1}{N}\left(\|\mu_1\|_2 + \|\mu_2\|_2 + \|\mu_3\|_2\right)$. We now provide bounds to the norms of μ_1, μ_2, μ_3 above. Take μ_1 as an example,

$$\|\mu_1\|_2 \leq \mathbb{E}\left[\left\|x_i^T A\mu y_i\right\|_2\right] = \mathbb{E}\left[\left|x_i^T A\mu\right| \|y_i\|_2\right] \leq L_1 \mathbb{E}\left[\left|x_i^T\mu\right| \|y_i\|_2\right] \leq L_1 \mathbb{E}\left[\|x_i\|_2 \|\mu\|_2 \|y_i\|_2\right].$$

Finally, note that we have $\|x_i\|_2 \leq HGR\frac{1-\gamma^H}{1-\gamma}$, $\|\mu\|_2 \leq HGR\frac{1-\gamma^H}{1-\gamma}$ and $\|y_i\|_2 \leq HG$, we can conclude

$$\|\mu_1\|_2 \leq L_1 H^3 G_1^3 R^2 \left(\frac{1-\gamma^H}{1-\gamma}\right)^2.$$

We can derive similar bounds

$$\|\mu_2\|_2 \leq L_1 H^3 G_1^3 R^2 \left(\frac{1-\gamma^H}{1-\gamma}\right)^2, \|\mu_3\|_2 \leq L_1 H^3 G_1^3 R^2 \left(\frac{1-\gamma^H}{1-\gamma}\right)^2.$$

Overall, this implies a bound on part (ii).

$$\|\text{part (ii)}\|_2 \leq \frac{3}{N} \cdot L_1 H^3 G_1^3 R^2 \left(\frac{1-\gamma^H}{1-\gamma}\right)^2.$$

Now, finally consider part (iii). We first note that Assumption 10 implies that $\left|\nabla^3 V_g(\theta)\right| \leq L$. This further implies that for any vector $x \in \mathbb{R}^D$,

$$\nabla^3 V_g(\theta) x^3 \coloneqq \sum_{ijk} \left(\nabla^3 V_g(\theta)\right)_{ijk} x_i x_j x_k \leq \underbrace{D^3 L}_{=:L_2} \|x\|_2^3 .$$

Recall that $\mu \coloneqq \mathbb{E}[R(\tau, g) \nabla_\theta \log p_{\theta,g}(\tau)]$ is the expected PG at θ. We have the following,

$$\begin{aligned}
\|\text{part (iii)}\|_2 &\leq \mathbb{E}\left[\left|\nabla^3 V_g(\tilde{\theta})\left(\frac{1}{N}\sum_{i=1}^N R(\tau_i)\nabla_\theta \log p_{\theta,g}(\tau_i) - \mu\right)^3\right| \left\|\sum_i \nabla_\theta \log p_{\theta,g}(\tau_i)\right\|_2\right] \\
&\leq \mathbb{E}\left[L_2 \left\|\left(\frac{1}{N}\sum_{i=1}^N R(\tau_i)\nabla_\theta \log p_{\theta,g}(\tau_i) - \mu\right)\right\|_2^3 \left\|\sum_i \nabla_\theta \log p_{\theta,g}(\tau_i)\right\|_2\right] \\
&\leq L_2 NHG\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N R(\tau_i)\nabla_\theta \log p_{\theta,g}(\tau_i) - \mu\right\|_2^3\right] \leq L_2 NHG_1 \cdot \frac{1}{N^{3/2}}\sigma_3^3 = \frac{1}{\sqrt{N}} L_2 H G_1 \sigma_3^3.
\end{aligned}$$

By combining the bounds in part (i)-(iii) via a triangle inequality, we arrive at the desired result.

□

Proposition 26. For all values of the parameter $\theta \in \mathbb{R}^D$, $\left\|\mathbb{E}[\hat{J}_{N,\text{LSF}}(\theta)] - J_N(\theta)\right\|_2 = \mathcal{O}(1/\sqrt{N})$.

Proof. Combining Proposition 24 and Proposition 25 with a triangle inequality, we get the desired result. □

Proposition 27. The objective $J_\infty(\theta)$ is Lipschitz with constant $(1 + \eta L_1) R G_1 H \frac{1-\gamma^H}{1-\gamma}$.

Proof. By construction $\nabla_\theta J_\infty(\theta) = \mathbb{E}_g\left[\nabla_\theta J_\infty(\theta, g)\right]$, we can derive

$$\nabla_\theta J_\infty(\theta) = \mathbb{E}_g\left[\left(I + \eta \nabla^2 V_g(\theta)\right) \nabla V_g(\theta'_g)\right],$$

where $\theta'_g \coloneqq \theta + \eta \mathbb{E}[R(\tau, g) \nabla \log p_{\theta,g}(\tau)]$. Recall that $\left\|\nabla^2 V_g(\theta)\right\|_{\text{op},2} \leq \left\|\nabla^2 V_g(\theta)\right\|_2 \leq L_1$ by the assumption and $\|\nabla V_g(\theta)\|_2 \leq R G_1 H \frac{1-\gamma^H}{1-\gamma}$. We conclude

$$\|\nabla_\theta J_\infty(\theta)\|_2 \leq \mathbb{E}_g\left[(1 + \eta L_1) R G_1 H \frac{1-\gamma^H}{1-\gamma}\right] = (1 + \eta L_1) R G_1 H \frac{1-\gamma^H}{1-\gamma}.$$

□

Proposition 28. For all $\theta \in \mathbb{R}^D$, $\mathbb{V}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] = \underbrace{\mathcal{O}(1/M) + \mathcal{O}(1/B)}_{\mathcal{O}(1)} + \mathcal{O}(1/N)$.

Proof. Recall that the LSF gradient estimate is constructed with three sources of randomness: B sampled task variables g_i, N sampled trajectories per task for the inner loop τ_{ij} and M sampled trajectories per task for the outer loop PG estimate τ'_{ik}. The variance of $\hat{J}_{N,\text{LSF}}$ comes from these three sources of randomness. Recall that given random variable Z_1, Z_2, Y, the variance of Y can be decomposed into three parts,

$$\mathbb{V}[Y] = \underbrace{\mathbb{E}\left[\mathbb{V}[Y|Z_1, Z_2]\right]}_{\text{first}} + \underbrace{\mathbb{E}\left[\mathbb{V}\left[\mathbb{E}\left[Y|Z_1, Z_2\right]|Z_1\right]\right]}_{\text{second}} + \underbrace{\mathbb{V}\left[\mathbb{E}[Y|Z_1]\right]}_{\text{third}}.$$

By definition of the variance of general random tensors, the above formula can be extended to the case where Y is a random tensor. Recall the LSF gradient estimate

$$\frac{1}{B}\sum_{i=1}^{B}\left(\left(I+\eta\sum_{j=1}^{N}R(\tau_{ij},g_i)\nabla_\theta^2\log p_{\theta,g_i}(\tau_{ij})\right)\nabla\hat{V}_{g_i}(\theta'_{i,N})+\eta\frac{1}{N}\sum_{j=1}^{N}R(\tau_{ij},g_i)u_{ij}u_{ij}^T\nabla\hat{V}_{g_i}(\theta'_{i,N})\right),$$

where $u_{ij}\coloneqq\nabla_\theta\log p_{\theta,g_i}(\tau_{ij})$. Here, recall that $\hat{V}_{g_i}(\theta'_{i,N})$ is the M-sample estimate of outer loop PG based on τ'_{ik}. Using the variance decomposition formula, we set $Y=$ part (i) and let $Z_1=(g_i)_{i=1}^B$, $Z_2=(\tau_{ij})_{i=1,j=1}^{B,N}$.

First part. For notational simplicity, let $A_i\coloneqq\left(I+\eta\frac{1}{N}\sum_{j=1}^{N}R(\tau_{ij},g_i)\nabla_\theta^2\log p_{\theta,g_i}(\tau_{ij})\right)$ and let $B_i=\eta\frac{1}{N}\sum_{j=1}^{N}R(\tau_{ij},g_i)u_{ij}u_{ij}^T$. Note that both A_i and B_i are random matrices. The first part of the variance is

$$\mathbb{E}_{Z_1,Z_2}\left[\mathbb{V}\left[\frac{1}{B}\sum_{i=1}^{B}(A_i+B_i)\hat{V}_{g_i}(\theta'_{i,N})|Z_1,Z_2\right]\right]=\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{Z_1,Z_2}\left[\|(A_i+B_i)\|_2^2\,\mathbb{V}\left[\hat{V}_{g_i}(\theta'_{i,N})|Z_1,Z_2\right]\right],$$

where we have used the conditional independence across different i indices and the fact that for any constant matrix A and zero mean vector x:

$$\begin{aligned}\mathbb{V}[Ax]&\le\mathbb{E}\left[\|Ax\|_2^2\right]=\mathbb{E}\left[x^TA^TAx\right]\le\mathbb{E}\left[x^Tx\cdot\max_i\left|\sigma_i\left(A^TA\right)\right|\right]\\&\le\mathbb{E}[x^Tx\sqrt{\sum_i\sigma_i^2(A^TA)}]=\left\|A^TA\right\|_2\mathbb{E}\left[x^Tx\right]\le\|A\|_2^2\,\mathbb{E}[x^Tx]=\|A\|_2^2\,\mathbb{V}[x].\end{aligned}$$

In the above, $\sigma_i(A)$ denotes the i-th eigenvalue of matrix A. We have also used the fact that $\sqrt{\sum_i\sigma_i^2(A^TA)}=\left\|A^TA\right\|_2\le\|A\|_2^2$. Since $\hat{V}_{g_i}(\theta'_{i,N})$ is M-sample estimate of PG from $\theta'_{i,N}$, from previous proof, we conclude $\mathbb{V}\left[\hat{V}_{g_i}(\theta'_{i,N})|Z_1,Z_2\right]\le\frac{L_1^2\sigma_1^2}{M}$. To bound the norm of each A_i+B_i, note that

$$\mathbb{E}[\|A_i+B_i\|_2^2]\le\|\mathbb{E}[A_i+B_i]\|_2^2+\mathbb{V}[A_i+B_i].$$

Now, note $\mathbb{E}[A_i+B_i]=I+\eta\nabla^2V_g(\theta)$ whose 2-norm is bounded as $\left\|I+\eta\nabla^2V_g(\theta)\right\|_2\le\|I\|_2+\left\|\eta\nabla^2V_g(\theta)\right\|_2\le\sqrt{D}+\eta L_1$. Next, by recalling $\mathbb{V}[X+Y]\le2\mathbb{V}[X]+2\mathbb{V}[Y]$, we have

$$\mathbb{V}[A_i+B_i]\le2\eta^2\frac{\sigma_{21}^2+\sigma_{22}^2}{N}.$$

Combining all previous results, we have the first part of the variance is upper bounded as follows

$$\mathbb{E}_{Z_1,Z_2}\left[\mathbb{V}\left[\frac{1}{B}\sum_{i=1}^{B}(A_i+B_i)\hat{V}_{g_i}(\theta'_{i,N})|Z_1,Z_2\right]\right]\le\left((\sqrt{D}+\eta L_1)^2+2\eta^2\frac{\sigma_{21}^2+\sigma_{22}^2}{N}\right)\cdot\frac{L_1^2\sigma_1^2}{M}.$$

Second part. Using notations above, we first integrate over the randomness in trajectories τ'_{ik},

$$\mathbb{E}[Y|Z_1,Z_2]=\frac{1}{B}\sum_{i=1}^{B}(A_i+B_i)\,\nabla V_{g_i}(\theta'_{i,N}).$$

Using the conditional independence of i given Z_1, we deduce

$$\mathbb{V}\left[\mathbb{E}[Y|Z_1,Z_2]|Z_1\right]=\frac{1}{B}\sum_{i=1}^{B}\underbrace{\mathbb{V}[(A_i+B_i)\,\nabla V_{g_i}(\theta'_{i,N})|g_i]}_{p_i}.$$

Now, consider each term p_i above. We can let $X_i = A_i + B_i, Y_i = \nabla V_{g_i}(\theta'_{i,N})$. We also define $X \coloneqq \mathbb{E}[X_i]$ and $Y \coloneqq \nabla V_{g_i}(\theta'_i)$ where $\theta' = \theta + \eta \mathbb{E}[R(\tau, g_i)\nabla_\theta \log p_{\theta,g_i}(\tau)]$. Importantly, note that $\mathbb{E}[Y_i] \neq Y$. By using the definition of variance and Cauchy–Schwarz inequality, we have

$$p_i \leq \mathbb{E}\left[\|X_iY_i - XY\|_2^2\right] \leq \underbrace{\mathbb{E}\left[\|X_iY_i - XY_i\|_2^2\right]}_{\text{term (a)}} + \underbrace{\mathbb{E}\left[\|XY_i - XY\|_2^2\right]}_{\text{term (b)}} + 2\sqrt{\mathbb{E}\left[\|X_iY_i - XY_i\|_2^2\right]}\sqrt{\mathbb{E}\left[\|XY_i - XY\|_2^2\right]}.$$

Consider term (a),

$$\begin{aligned}\mathbb{E}\left[\|X_iY_i - XY_i\|_2^2\right] &\leq \mathbb{E}\left[\|X_i - X\|_2^2 \|Y_i\|_2^2\right] \leq \left(RHG_1\frac{1-\gamma^H}{1-\gamma}\right)^2 \mathbb{E}\left[\|X_i - X\|_2^2\right] \\ &\leq \left(RHG_1\frac{1-\gamma^H}{1-\gamma}\right)^2 \frac{2\eta^2(\sigma_{21}^2 + \sigma_{22}^2)}{N}.\end{aligned}$$

Now we consider term (b)

$$\mathbb{E}\left[\|XY_i - XY_i\|_2^2\right] \leq \mathbb{E}\left[\|X\|_2^2 \|Y_i - Y\|_2^2\right] \leq (1+\eta L_1)^2 \mathbb{E}\left[\|Y_N - Y\|_2^2\right] \leq (1+\eta L_1)^2 \frac{L_1^2\sigma_1^2}{N},$$

where we have applied a bound on the norm of the PG $\leq RHG_1\frac{1-\gamma^H}{1-\gamma}$ and on the Hessian operator norm $\leq L_1$ implied by the assumptions. We thus conclude the following bound on the second variance term,

$$\begin{aligned}\mathbb{E}\left[\mathbb{V}\left[\mathbb{E}\left[Y|Z_1, Z_2\right]|Z_1\right]\right] \leq \mathbb{E}\left[\frac{1}{B}\sum_{i=1}^B p_i\right] \leq \left(RHG_1\frac{1-\gamma^H}{1-\gamma}\right)^2 \frac{2\eta^2(\sigma_{21}^2 + \sigma_{22}^2)}{N} + (1+\eta L_1)^2 \frac{L_1^2\sigma_1^2}{N} + \\ 2\left(RHG_1\frac{1-\gamma^H}{1-\gamma}\right)(1+\eta L_1)L_1\eta\sigma_1\sqrt{2(\sigma_{21}^2+\sigma_{22}^2)}\frac{1}{N}.\end{aligned}$$

Third part. Let $X_i = I + \eta\sum_{j=1}^N R(\tau_{ij}, g_i)\nabla_\theta^2 \log p_{\theta,g_i}(\tau_{ij}) + \eta\frac{1}{N}\sum_{j=1}^N R(\tau_{ij}, g_i)u_{ij}u_{ij}^T$ and $Y_i = \nabla V_{g_i}(\theta'_{i,N})$. Also let $Y = Y_i = \nabla V_{g_i}(\theta'_i)$. We can write

$$\mathbb{V}\left[\mathbb{E}[Y|Z_1]\right] = \frac{1}{B}\mathbb{V}\left[\mathbb{E}[X_iY_i]\right]$$

where we have used the independence across different i. For clarity, the expectation is w.r.t. all randomness in τ_{ij} and τ'_{ik}, whereas the variance is w.r.t. the randomness in i. Now, for any i, consider the following

$$\begin{aligned}\|\mathbb{E}\left[X_iY_i\right]\|_2^2 &\leq \mathbb{E}\left[\|X_iY_i\|_2^2\right] \\ &\leq \mathbb{E}\left[\|X_i\|_2^2 G_1^2R^2H^2\left(\frac{1-\gamma^H}{1-\gamma}\right)^2\right] \\ &= G_1^2R^2H^2\left(\frac{1-\gamma^H}{1-\gamma}\right)^2 \cdot \left(\mathbb{V}\left[X_i\right] + \|\mathbb{E}\left[X_i\right]\|_2^2\right) \\ &\leq G_1^2R^2H^2\left(\frac{1-\gamma^H}{1-\gamma}\right)^2 \cdot \left(2\eta^2\frac{\sigma_{21}^2+\sigma_{22}^2}{N} + \|\mathbb{E}\left[X_i\right]\|_2^2\right).\end{aligned}$$

We need to upper bound the 2-norm of $\mathbb{E}[X_i] = I + \eta\nabla^2 V_g(\theta)$. Note that we have $\left\|I + \eta\nabla^2 V_g(\theta)\right\|_2 \leq \|I\|_2 + \left\|\eta\nabla^2 V_g(\theta)\right\|_2 = \sqrt{D} + \eta L_1$. We hence have

$$\mathbb{V}\|\mathbb{E}[Y|Z_1]\|_2 \leq \frac{1}{B}\mathbb{E}\left[\|\mathbb{E}[X_iY_i]\|_2^2\right] \leq \frac{1}{B}G_1^2R^2H^2\left(\frac{1-\gamma^H}{1-\gamma}\right)^2 \cdot \left(2\eta^2\frac{\sigma_{21}^2+\sigma_{22}^2}{N} + \left(\sqrt{D}+\eta L_1\right)^2\right).$$

Combining all parts. Combining all the three parts above, we have

$$\mathbb{V}\left[\hat{J}_{N,\text{LSF}}(\theta)\right] \leq \underbrace{\mathcal{O}(1/M)}_{\text{first part}} + \underbrace{\mathcal{O}(1/N)}_{\text{second part}} + \underbrace{\mathcal{O}(1/B)}_{\text{third part}}.$$

□

Lemma 29. (Adapted from (Ajalloeian and Stich, 2020)) Let $F : \mathbb{R}^H \mapsto \mathbb{R}$ be a L-Lipschitz function. Let $\hat{g}(x)$ be an estimate to $\nabla F(x)$. Its bias and variance properties are the following,

$$\|\mathbb{E}\left[\hat{g}(x)\right] - \nabla F(x)\|_2^2 \leq \psi^2, \mathbb{V}\left[\hat{g}(x)\right] \leq \sigma^2,$$

for all $x \in \mathbb{R}^H$. Now consider the recursion: $x_{t+1} = x_t + \alpha\hat{g}(x_t)$. For any $\epsilon > 0$, if we choose the learning rate $\alpha = \min\{\frac{1}{L}, \frac{\epsilon+\psi^2}{2L\sigma^2}\}$, then for $T = \max\{\frac{1}{\epsilon+\psi^2}, \frac{\sigma^2}{\epsilon^2+\psi^4}\}$ iterations, we have

$$\min_{1\leq t\leq T} \mathbb{E}\left[\|\nabla F(x_t)\|_2^2\right] \leq \epsilon + \psi^2.$$

Proof. Please refer to (Ajalloeian and Stich, 2020) for detailed proof and the complete result.

□

Proposition 30. With a properly chosen learning rate in Algorithm 1, for any $\epsilon > 0$, with $T_{\text{LSF}} = 2\max\{\frac{1}{\epsilon^2+\mathcal{O}(1/N)}, \frac{\mathcal{O}(1)+\mathcal{O}(1/N)}{\epsilon^4+\mathcal{O}(1/N^2)}\}$ iterations of the algorithm, we have

$$\min_{1\leq t\leq T_{\text{LSF}}} \mathbb{E}[\|J_N(\theta_t)\|_2^2] = \epsilon^2 + \mathcal{O}(1/N) =: \delta_{\text{LSF}}.$$

Proof. We directly draw results from (Ajalloeian and Stich, 2020) where they establish convergence to stationary point using biased stochastic gradient estimates. See Lemma 29 for a simplified version of their result which will be useful for our analysis below.

Instead of directly characterizing the convergence to a stationary point of J_N, we consider how fast the algorithm converges to a stationary point of J_∞. Proposition 27 shows that J_∞ is Lipschitz-smooth with a Lipschitz constant L independent of N, hence we can write $L = \mathcal{O}(1)$. Proposition 25 shows that the estimate bias ψ is of order $\mathcal{O}(1/\sqrt{N})$ such that

$$\left\|J_\infty(\theta) - \mathbb{E}[\hat{J}_{N\text{LSF}}(\theta)]\right\|_2^2 \leq \psi^2.$$

Proposition 12 shows that the variance $\mathbb{V}\left[\hat{J}_{N\text{LSF}}(\theta)\right] \leq \sigma^2 = \mathcal{O}(1) + \mathcal{O}(1/N)$ (note that we treat B, M as $\mathcal{O}(1)$ here). Directly using results from Proposition 29, we obtain the following: after $T_{\text{LSF}} = \max\{\frac{1}{\epsilon^2+\mathcal{O}(1/N)}, \frac{\mathcal{O}(1)+\mathcal{O}(1/N)}{\epsilon^4+\mathcal{O}(1/N^2)}\}$ iterations,

$$\min_{1\leq t\leq T_{\text{LSF}}} \mathbb{E}[\|\nabla_\theta J_\infty(\theta_t)\|_2^2] \leq \epsilon^2 + \mathcal{O}(1/N).$$

Finally, recall that Proposition 24 upper bounds the bias between $\nabla_\theta J_\infty(\theta)$ and $J_N(\theta)$ by $\mathcal{O}(1/N)$, we obtain via a CS inequality $\|a+b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2$,

$$\min_{1\leq t\leq T_{\text{LSF}}} \mathbb{E}[\|J_N(\theta_t)\|_2^2] \leq \min_{1\leq t\leq T_{\text{LSF}}} 2\mathbb{E}[\|\nabla_\theta J_\infty(\theta_t)\|_2^2] + 2\mathbb{E}[\|J_N(\theta_t) - J_\infty(\theta_t)\|_2^2] \leq 2\epsilon^2 + \mathcal{O}(1/N).$$

By properly scaling ϵ, the above is equivalent to: after $T_{\text{LSF}} = 2 \cdot \max\{\frac{1}{\epsilon^2+\mathcal{O}(1/N)}, \frac{\mathcal{O}(1)+\mathcal{O}(1/N)}{\epsilon^4+\mathcal{O}(1/N^2)}\}$ iterations,

$$\min_{1\leq t\leq T_{\text{LSF}}} \mathbb{E}[\|J_N(\theta_t)\|_2^2] \leq \epsilon^2 + \mathcal{O}(1/N).$$

□

H. Additional experiments

H.1. Toy 1-D Optimization Problem

The variable X comes from the distribution $p_\theta = \mathcal{N}(\theta, 1^2)$. This implies a natural reparameterization, $X = \theta + \zeta, \zeta \sim \mathcal{N}(0, 1)$, which we use for the PW gradient estimate.

Bias-variance trade-off. To generate Fig 1(a), we compute the MSE of different gradient estimates against the true gradient, evaluated at the initial parameter of the algorithm. In particular, given the ground truth gradient g, we generate $M = 100$ gradient estimates $\hat{g}_i$ for each type, and compute

$$\frac{1}{M}\sum_{i=1}^{M} \|\hat{g}_i - g\|_2^2 ,$$

as an estimate to the MSE. For LSF, the bias is computed as

$$\frac{1}{M}\sum_{i=1}^{M} \|\hat{g}_i - \bar{g}\|_2^2 ,$$

where $\bar{g} = \frac{1}{M}\sum_{i=1}^{M} \hat{g}_i$ is an estimate of the expected gradient estimate. SF and PW gradient estimates have zero variance, so that all their MSE consists of variance. Finally, we use the average over 1000 PW gradient estimates as g (because PW has low variance and zero bias, we expect the approximation to be reasonably accurate).

Optimization. To generate Fig 1(b), for each type of gradient estimate, at each iteration $1 \le t \le T$, we construct an average gradient estimate $\bar{g} = \frac{1}{B}\sum_{i=1}^{B} \hat{g}_i$ where $\hat{g}_i$ is an one-sample gradient estimate of the N-sample MC objective. The parameter is updated with $\hat{g}$ at each iteration, with Adam optimizer (Kingma and Ba, 2014) and learning rate 0.1.

At $T = 100$, we record the objective $L(\theta_T)$ for each type of gradient estimate. We repeat the same experiment 100 times, and compare mean $\pm$ std averaged over such repeated trials.

H.2. High-dimensional Meta-RL Problems

The following specifies details of generating Fig 1(c)-(d).

Environments. The environments of the meta-RL experiments are based on MuJoCo (Todorov et al., 2012), and imported directly from the open source projects of (Rothfuss et al., 2018). These are robotics control tasks where the states s_t are sensory inputs and actions a_t are controls applied to the robots. Across all three tasks we considered, the task g corresponds to different directions in which the robot should aim to run. See (Rothfuss et al., 2018) for further details.

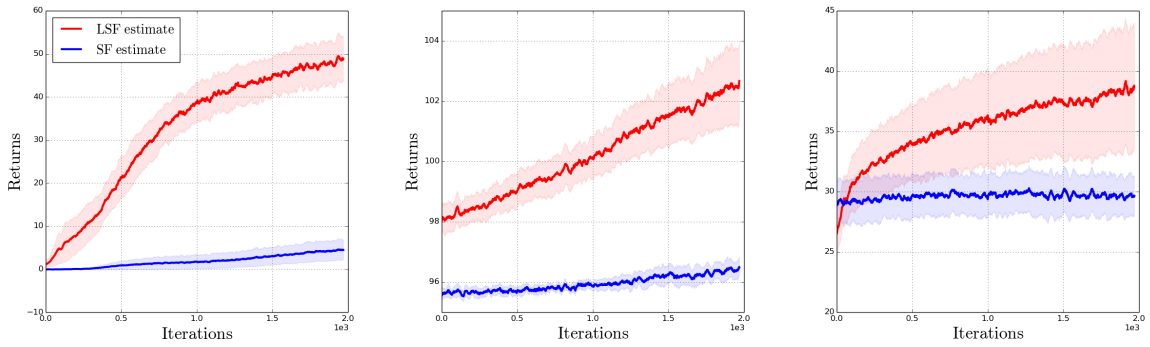
Trust region outer loop optimization. After obtaining the gradient estimate $\hat{J}$, Algorithm 1 suggests that we update $\theta_{t+1} = \theta_t + \alpha\hat{J}$. In practice, we adopt trust region policy optimization (Schulman et al., 2015), which enforces a trust region constraint between θ_t and θ_{t+1} when updating the parameter. See the open sourced code base for hyper-parameter settings of the TRPO optimizer.

Hyper-parameters of algorithms. We use a batch of $B = 20$ tasks per iteration, $N = M = 20$ trajectories per task for both inner loop adaptation and outer loop rollouts for PG estimates. Each trajectory is truncated at $H = 100$ steps. We adapt only one step throughout the experiments. Please refer to the open sourced code base for other default hyper-parameters whose details we omit here.

Important implementation details. Though all algorithms are based on the open source project of (Rothfuss et al., 2018), it is worth noting a number of important modifications that we make to ensure that the implementation adheres to our theoretical setups as much as possible.

The unbiased generalized SF gradient estimate is very closely related to the gradient estimate used in E-MAML algorithm. In fact, when implemented exactly, the E-MAML algorithm utilizes the SF gradient estimate $\hat{J}^{(i)}_{N,\text{SF}}(\theta, g) + \hat{J}^{(ii)}_{N,\text{SF}}(\theta, g)$ defined in Eqn 8. However, the code base in (Rothfuss et al., 2018) effectively uses the following gradient estimate,

$$\frac{1}{N}\hat{J}^{(i)}_{N,\text{SF}}(\theta, g) + \hat{J}^{(ii)}_{N,\text{SF}}(\theta, g).$$



(a) Meta-RL HalfCheetah (b) Meta-RL Ant (c) Meta-RL Walker2D

Figure 2. Full results of high-dimensional meta-RL problems: LSF outperforms SF consistently across all tasks. Each curve averages over 5 runs.

The factor $1/N$ biases the overall estimate, yet reduces the variance introduced by $\hat{J}^{(i)}_{N,\text{SF}}(\theta, g)$. With this, we see that the practical implementation of E-MAML already introduces bias for variance reduction, albeit in a more opaque way. To obtain results of the SF gradient estimate in Fig 1, we remove the $1/N$ factor and use the unbiased SF gradient estimate.

H.3. Practical (prior) implementations of generalized LSF gradient estimate

To implement the generalized LSF estimate, we need to construct unbiased estimate to value function Hessian $\nabla^2 V_g(\theta)$ evaluated at the initial policy parameter θ. On a high level, one first constructs an estimate $\hat{H}_N \approx \nabla^2 V_g(\theta)$ and then computes the meta-RL gradient estimate as

$$\hat{H}_N \nabla V_g(\theta'_N),$$

where θ'_N is the (random) updated parameter. A number of prior work discusses on how to construct unbiased estimates (Foerster et al., 2018; Mao et al., 2019; Farquhar et al., 2019) or biased estimates (Rothfuss et al., 2018; Tang et al., 2020) to the value function Hessian. A major desiderata is that such estimates should lead to variance reduction compared to the naive "trajectory-based" estimate. Concretely, the "trajectory-based" Hessian estimate is (derived from Eqn 11)

$$\frac{1}{N}\sum_{i=1}^{N} R(\tau_i, g)\nabla^2_\theta \log p_{\theta,g}(\tau_i) + \frac{1}{N}\sum_{i=1}^{N} R(\tau_i, g)\left(\nabla_\theta \log p_{\theta,g}(\tau_i)\right)\left(\nabla_\theta \log p_{\theta,g}(\tau_i)\right)^T. \tag{13}$$

Arguably, the variance of the above estimate could be further improved by exploiting the Markov property of trajectories τ_i. Taking computing PG estimate as an analogy, when computing "trajectory-based" PG estimate $R(\tau_i)\nabla_\theta \log p_{\theta,g}(\tau_i)$, we can instead use its "stepwise-based" variant $\sum_{t=0}^{H-1} \gamma^t \hat{Q}_t \nabla_\theta \log \pi_\theta(a_t|s_t, g)$, where $\hat{Q}_t$s are unbiased estimates to Q-functions. The "stepwise-based" estimate usually has much lower variance than the "trajectory-based" estimate, because it is constructed based on the Markov structure of the trajectory. Constructing variance-reduced estimates for the Hessian is more complicated, but is better understood through the lens of off-policy evaluation. We refer readers to (Tang et al., 2021) for further details.

In our experiments, we always use such "stepwise-based" PG and Hessian estimates when computing the meta-RL gradient estimates. Specifically, we use DiCE (Foerster et al., 2018) to compute the LSF gradient estimate, which can be interpreted as building an unbiased variant of Eqn 13 with variance reduction via the Markov structure of the trajectory. Please refer to the code base of (Rothfuss et al., 2018) for further implementation details.

Full results. See Fig 2 for full results on the high-dimensional meta-RL problems. Overall, the LSF estimate achieves significant performance gains over the SF estimate.