A Theoretical Understanding of Gradient Bias in Meta-Reinforcement Learning

Bo Liu[*]
Institute of Automation,
Chinese Academy of Sciences
benjaminliu.eecs@gmail.com

Xidong Feng[*]
University College London
xidong.feng.20@ucl.ac.uk

Jie Ren
University of Edinburgh
jieren9806@gmail.com

Luo Mai
University of Edinburgh
luo.mai@ed.ac.uk

Rui Zhu
DeepMind
ruizhu@google.com

Haifeng Zhang
Institute of Automation,
Chinese Academy of Sciences
haifeng.zhang@ia.ac.cn

Jun Wang
University College London
jun.wang@cs.ucl.ac.uk

Yaodong Yang[†]
Institute for AI, Peking University
Beijing Institute for General AI
yaodong.yang@pku.edu.cn

Abstract

Gradient-based Meta-RL (GMRL) refers to methods that maintain two-level optimisation procedures wherein the outer-loop meta-learner guides the inner-loop gradient-based reinforcement learner to achieve fast adaptations. In this paper, we develop a unified framework that describes variations of GMRL algorithms and points out that existing stochastic meta-gradient estimators adopted by GMRL are actually **biased**. Such meta-gradient bias comes from two sources: 1) the compositional bias incurred by the two-level problem structure, which has an upper bound of $O\big(K\alpha^K\hat{\sigma}_{\text{In}}|\tau|^{-0.5}\big)$ *w.r.t.* inner-loop update step K, learning rate α, estimate variance $\hat{\sigma}^2_{\text{In}}$ and sample size $|\tau|$, and 2) the multi-step Hessian estimation bias $\hat{\Delta}_H$ due to the use of autodiff, which has a polynomial impact $O\big((K-1)(\hat{\Delta}_H)^{K-1}\big)$ on the meta-gradient bias. We study tabular MDPs empirically and offer quantitative evidence that testifies our theoretical findings on existing stochastic meta-gradient estimators. Furthermore, we conduct experiments on Iterated Prisoner's Dilemma and Atari games to show how other methods such as off-policy learning and low-bias estimator can help fix the gradient bias for GMRL algorithms in general.

1 Introduction

Meta Learning, also known as learning to learn, is proposed to equip intelligent agents with meta knowledge for fast adaptations [30]. Meta-Reinforcement Learning (RL) algorithms aim to train RL

[*]Equal contribution, the order is determined by flipping a coin. See Appendix J for more details.
[†]Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

agents that can adapt to new tasks with only few examples [5, 10, 16]. For example, MAML-RL [10], a typical Meta-RL algorithm, learns the initial parameters of an agent's policy so that the agent can rapidly adapt to new environments with a limited number of policy-gradient updates. Recently, Meta-RL algorithms have been further developed beyond the scope of learning fast adaptations. An important direction is to conduct online meta-gradient learning for adaptively tuning algorithmic hyper-parameters [39, 41] or designing intrinsic reward [42] during training in one single task. Besides, there are Meta-RL developments that manage to discover new RL algorithms by learning algorithmic components or other fundamental concepts in RL, such as the policy gradient objective [26] or the TD-target [40], which can generalise to solve a distribution of different tasks.

In general, gradient-based Meta-RL (GMRL) tasks can be formulated by a two-level optimisation procedure. This procedure optimises the parameters of an outer-loop meta-learner, whose objective is dependent on a K-step policy update process (e.g., stochastic gradient descent) conducted by an inner-loop learner. Formally, this procedure can be written as:

$$\begin{aligned} &\max_{\boldsymbol{\phi}} J^K(\boldsymbol{\phi}) := J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K), \\ &\text{s.t. } \boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i + \alpha \nabla_{\boldsymbol{\theta}^i} J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i), i \in \{0, 1 \dots K-1\}, \end{aligned} \tag{1}$$

where $\boldsymbol{\theta}$ are inner-loop policy parameters, $\boldsymbol{\phi}$ are meta parameters, α is the learning rate, J^{In} and J^{Out} are value functions for the inner and the outer-loop learner. In solving Eq. (1), estimating the meta-gradient of $\nabla_{\boldsymbol{\phi}} J^K(\boldsymbol{\phi})$ from a two-level optimisation process is non-trivial; how to conduct *accurate* estimation on $\nabla_{\boldsymbol{\phi}} J^K(\boldsymbol{\phi})$ has been a critical yet challenging research problem [23, 29, 33].

In this paper, we point out that the meta-gradient estimators adopted by many recent GMRL methods [26, 40] are in fact biased. We conclude that such bias comes from two sources: (1) the **compositional bias** and (2) the **multi-step Hessian bias**. The compositional bias origins from the discrepancy between the sampled policy gradient $\nabla_{\boldsymbol{\theta}} \hat{J}^{\text{In}}$ and expected policy gradient $\nabla_{\boldsymbol{\theta}} J^{\text{In}}$, and the multi-step Hessian estimation bias occurs due to the biased Hessian estimation $\nabla^2_{\boldsymbol{\theta}} \hat{J}^{\text{In}}$ resulting from the employment of automatic differentiation in modern GMRL implementations.

There are very few prior work that investigates the bias in GMRL, most of them limited to the Hessian estimation bias in the MAML-RL setting [29, 33]. Our paper investigates the above two biases in a broader setting and applies on generic meta-gradient estimators. For the compositional bias term, we offer the first theoretical analysis on its quantity, based on which we then investigate current mitigation solutions. For the multi-step Hessian bias, we provide rigorous analysis in GMRL settings particularly for those GMRL tasks where complex inner-loop optimisations are involved.

For the rest of the paper, we first introduce a *unified* Meta-RL framework that can describe variations of existing GMRL algorithms. Building on this framework, we offer two theoretical results that **1)** the compositional bias has an upper bound of $O\big(K\alpha^K \hat{\sigma}_{\text{In}} |\tau|^{-0.5}\big)$ with respect to the inner-loop update step K, the learning rate α, the estimate variance $\hat{\sigma}^2_{\text{In}}$ and the sample size $|\tau|$, and **2)** the multi-step Hessian bias $\hat{\Delta}_H$ has a polynomial impact of $O\big((K-1)(\hat{\Delta}_H)^{K-1}\big)$. To validate our theoretical insights on these two biases, we conduct a comprehensive list ablation studies. Experiments over tabular MDP with MAML-RL [10] and LIRPG [42] demonstrate how quantitatively these two biases influence the estimation accuracy, which consolidates our theories.

Furthermore, our theoretical results help understand to what extent existing methods can mitigate theses two bias empirically. For the compositional bias, we show that off-policy learning methods can reduce the inner-loop policy gradient variance and the resulting compositional bias. For the multi-step Hessian bias, we study how the low-variance curvature technique based on Rothfuss et al. [29] can help correct the Hessian bias for general GMRL problems. We test these solutions on environments including iterated Prisoner's Dilemma with off-policy corrected LOLA-DiCE [13], and eight Atari games based on MGRL [39] with the multi-step Hessian correction technique. Experimental results confirm that those bias-correction methods can substantially decrease the meta-gradients bias and improve the overall performance on rewards.

2 Related Work

A pioneering work that studies meta-gradient estimation is Al-Shedivat et al. [1] who discussed the biased estimation problem of MAML and proposed the E-MAML sample based formulation to fix

the meta-gradient bias. Following work includes Rothfuss et al. [29], Liu et al. [23], Tang et al. [33] that tried to fix the meta-gradient estimation error by reducing the estimation variance so as to improve the performance. Moreover, Foerster et al. [12], Farquhar et al. [8], Mao et al. [24] discussed the higher-order gradient estimation in RL. Recently, Bonnet et al. [4], Vuorio et al. [36] proposed algorithms to balance the bias-variance trade-off for meta-gradient estimates. Within theoretical context of GMRL, most theoretical analysis focuses on convergence to stationary points. Fallah et al. [6] established convergence guarantees of gradient-based meta-learning algorithms for supervised learning with one-step inner-loop update. Ji et al. [18] extended the analysis to multi-step inner loop updates. Fallah et al. [7] established convergence for the E-MAML formulation. Our work is different from all the above prior work from three folds: (1) we study the additional bias term (the compositional bias) (2) we consider different formulation (expected update while sample-based update in [36], refer to Appendix B for more discussions) (3) we study a broader scope of applications that include different GMRL algorithm instantiations and settings.

In the following part, we review existing GMRL algorithms and categorise them into four research topics. We offer one typical example for each topic in Table 1 and further discussed their relationship and how they can be unified into one framework in Sec. 3. We also provide a more self-contained explanation for each topic in Appendix A for readers that are not familiar with GMRL.

Few-shot RL. The idea of few-shot RL is to enable RL agent with fast learning ability. Specifically, the RL agent is only allowed to interact with the environment for a few trajectory to conduct task-specific adaptation. Conducting few-shot RL has two approaches: gradient based and context based. Context based Meta-RL involves works [5, 15, 27, 37], which uses neural network to embed the information from few-shot interactions so as to obtain task-relevant context. Gradient based few-shot RL [1, 23, 29] focus on meta-learning the model's initial parameters through meta-gradient descent.

Opponent Shaping. Foerster et al. [13], Letcher et al. [22], Kim et al. [19] explicitly models the learning process of opponents in multi-agent learning problems, which can be thought of as GMRL problems since meta-gradient estimation in these works involve differentiation over opponent's policy updates. By modelling opponents' learning process, the multi-agent learning process can reach better social welfare [13, 22] or the ego agent can adapt to a new peer agent [19].

Single-lifetime Meta-gradient RL. This line of research focuses on learning online adaptation over algorithmic hyper-parameters to enhance the performance of an RL agent in one single task, such as discount factor [39, 41], intrinsic reward generator [42], auxiliary loss [34], reward shaping mechanism [17], and value correction [44]. The main feature of online meta-gradient RL is that they are under single-lifetime setting [40], meaning that the algorithm only iterates through the whole RL learning procedure in one task rather a distribution of tasks.

Multi-lifetime Meta-gradient RL. Compared to the previous single-lifetime settings, "multi-lifetime" refers to settings where agents learn to adapt on a distribution of tasks or environments.The meta-learning target includes policy gradient or TD learning objectives [3, 20, 26], intrinsic reward [43], target value function [40], options in hierarchical RL [35], and recently the design of curriculum in multi-agent learning [9].

3 A Unified Framework for Meta-gradient Estimation

In this section, we derive a general formulation for meta-gradient estimation in GMRL. This formulation enables us to conduct general analysis about meta-gradient estimation and we will show how algorithms in the four research topics mentioned in Sec. 2 can be described through it.

We propose that a general GMRL objective with K-step inner-loop policy gradient update can be written as the following objective

$$\max_{\boldsymbol{\phi}} J^K(\boldsymbol{\phi}) := J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K), \boldsymbol{\theta}^K = \boldsymbol{\theta}^0 + \alpha \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\theta}^i} J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i). \tag{2}$$

We denote the meta-parameters as $\boldsymbol{\phi}$, and the pre- and post-adapt inner parameters as $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^K$, respectively. The meta objective as J^{Out}, the inner loop objective as J^{In}. The general objective for GMRL is to maximise the meta objective $J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)$, where the inner-loop post-adapt parameters are obtained by taking K policy gradient steps. Note that all inner-loop updates refer to an expected policy gradient (EPG) and all bias term we discuss hereafter is the bias *w.r.t* the **exact** meta-gradient

Table 1: Four typical gradient-based Meta-RL (GMRL) algorithms.

Category	Algorithms	Meta parameter ϕ	Inner parameter θ
Few-shot RL	MAML [10]	Initial Parameter	Initial Parameter
Opponent Shaping	LOLA [13]	Ego-agent Policy	Other-agent Policy
Single-lifetime MGRL	MGRL [39]	Discount Factor	RL Agent Policy
Multi-lifetime MGRL	LPG [26]	LSTM Network	RL Agent Policy

in this K-step EPG inner-loop setting. We discuss the **truncated** K-step EPG setting in Appendix B. The form of exact meta-gradient $\nabla_{\boldsymbol{\phi}} J^K(\boldsymbol{\phi})$ is given by the following proposition via the chain rule.

Proposition 3.1 (K-step Meta-Gradient)**.** *The exact meta-gradient to the objective in Eq.* (2) *can be written as:*

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} J^K(\boldsymbol{\phi}) &= \nabla_{\boldsymbol{\phi}} J^{Out}(\boldsymbol{\phi}, \boldsymbol{\theta}^K) + \alpha \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^K \nabla_{\boldsymbol{\theta}^K} J^{Out}(\boldsymbol{\phi}, \boldsymbol{\theta}^K), \\ \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^K &= \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{In}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{In}(\boldsymbol{\phi}, \boldsymbol{\theta}^j) \right).\end{aligned} \tag{3}$$

The derivation of the above proposition is in Appendix E.1.

For each topic mentioned in Sec. 2, we pick one GMRL algorithm example to illustrate how they can be fit into this framework. To describe meta-gradient estimation in RL, we start with basic notations. Consider a discrete-time finite horizon Markov Decision Process (MDP) defined by $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma, H \rangle$. At each time step t, the RL agent observes a state $\boldsymbol{s}_t \in \mathcal{S}$, takes an action $\boldsymbol{a}_t \in \mathcal{A}$ based on the policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$ parametrised with $\boldsymbol{\theta} \in \mathbb{R}^d$, transits to the next state $\boldsymbol{s}_{t+1} \in \mathcal{S}$ according to the transition function $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ and receives the reward $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$. We define the return $\mathcal{R}(\boldsymbol{\tau}) = \sum_{t=0}^{H} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ as the discounted sum of rewards along a trajectory $\boldsymbol{\tau} := (\boldsymbol{s}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{s}_{H-1}, \boldsymbol{a}_{H-1}, \boldsymbol{s}_H)$ sampled by agent policy. The objective for the RL agent is to maximise the expected discounted sum of rewards $V(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau};\boldsymbol{\theta})}[\mathcal{R}(\boldsymbol{\tau})]$. Then the RL agent updates parameter $\boldsymbol{\theta}$ using policy gradient given by $\nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau};\boldsymbol{\theta})}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \mathcal{R}(\boldsymbol{\tau})\right]$ where $\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}) = \sum_{t=0}^{H} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$. For two-agent RL problems, we can extend the MDP to two-agent MDP (or, Stochastic games [31]) defined by $\langle \mathcal{S}_1, \mathcal{S}_2, \mathcal{A}_1, \mathcal{A}_2, P, r_1, r_2, \gamma, H \rangle$, the learning objective for agent i is to maximise its value function of $V_i(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau}_i \sim p(\boldsymbol{\tau}_i;\boldsymbol{\theta}_i)}\left[\mathcal{R}_i(\boldsymbol{\tau}_i)\right]$.

MAML. Finn et al. [10] optimized over meta initial parameters to maximize the return of one-step adapted policy: $\boldsymbol{\theta}^1 = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta})$. In MAML-RL, $J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^1)$ degenerates to $V(\boldsymbol{\theta}^1)$ and $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ represent the same initial parameters. The meta-gradient can be derived in the form of Eq. (3): $\nabla_{\boldsymbol{\theta}} \boldsymbol{\theta}^1 \nabla_{\boldsymbol{\theta}^1} V(\boldsymbol{\theta}^1)$, where $\nabla_{\boldsymbol{\theta}} \boldsymbol{\theta}^1 = I + \alpha \nabla^2_{\boldsymbol{\theta}} V(\boldsymbol{\theta})$.

LOLA. Foerster et al. [13] proposed a new learning objective by including an additional term that accounts for the impact of the ego policy to the anticipated opponent's gradient update. For LOLA-agent with parameters $\boldsymbol{\phi}$, it will optimise its return over one-step-lookahead opponent parameters $\boldsymbol{\theta}^1$. The meta-gradient can be shown as: $\nabla_{\boldsymbol{\phi}} V_1(\boldsymbol{\phi}, \boldsymbol{\theta}^1) + \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^1 \nabla_{\boldsymbol{\theta}^1} V_1(\boldsymbol{\phi}, \boldsymbol{\theta}^1)$, where $\nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^1 = \alpha \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} V_2(\boldsymbol{\phi}, \boldsymbol{\theta})$

MGRL. Xu et al. [39] proposed to tune the discount factor γ and bootstrapping parameter λ in an online manner. The main feature of MGRL is to conduct inner-loop RL policy $\boldsymbol{\theta}$ update and outer-loop meta parameters $\boldsymbol{\phi} = (\gamma, \lambda)$ alternately. In MGRL, $J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^1)$ degenerates to $V(\boldsymbol{\theta}^1)$. The meta-gradient takes the form: $\nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^1 \nabla_{\boldsymbol{\theta}^1} V(\boldsymbol{\theta}^1)$, $\nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^1 = \alpha \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\phi}, \boldsymbol{\theta})$.

LPG. Oh et al. [26] aimed to learn a neural network based RL algorithm, by which a RL agent can be properly trained. In LPG, $\boldsymbol{\theta}$ represents the RL agent policy parameters and $\boldsymbol{\phi}$ is the meta-parameter of neural LSTM RL algorithm, $J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ denotes $f(\boldsymbol{\phi}, \boldsymbol{\theta})$, which is the output of meta-network $\boldsymbol{\phi}$ for conducting inner-loop neural policy gradients. The meta-gradient can be shown as: $\nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^1 \nabla_{\boldsymbol{\theta}^1} V(\boldsymbol{\theta}^1)$, $\nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^1 = \alpha \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\phi}, \boldsymbol{\theta})$.

Remark. The analytical form of exact meta-gradient given in Eq. (3) involves computation of first-order gradient $\nabla_{\boldsymbol{\theta}} J^{\text{In}}$, $\nabla_{\boldsymbol{\phi}} J^{\text{Out}}$ and$\nabla_{\boldsymbol{\theta}} J^{\text{Out}}$, Jacobian $\nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} J^{\text{In}}$ and Hessian $\nabla^2_{\boldsymbol{\theta}} J^{\text{In}}$. In practice, these four quantities can be estimated by random samples from inner-loop update step 0 to $K-1$, which denoted by $\boldsymbol{\tau}_0^{0:K-1}, \boldsymbol{\tau}_1^{0:K-1}, \boldsymbol{\tau}_2^{0:K-1}, \boldsymbol{\tau}_3$. As a result, the estimated gradient $\nabla_{\boldsymbol{\phi}} \hat{J}^K(\boldsymbol{\phi})$ can be

derived as:

$$\nabla_{\boldsymbol{\phi}} \hat{J}^K(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \boldsymbol{\tau}_3) + \alpha \nabla_{\boldsymbol{\phi}} \hat{\boldsymbol{\theta}}^K \nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \boldsymbol{\tau}_3),$$
$$\nabla_{\boldsymbol{\phi}} \hat{\boldsymbol{\theta}}^K = \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} J^{\text{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_1^i) \prod_{j=t+1}^{K-1} \left(I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} J^{\text{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \right). \tag{4}$$

The post-adapt inner parameter estimate takes the form $\hat{\boldsymbol{\theta}}^K = \boldsymbol{\theta}^0 + \alpha \sum_{i=0}^{K-1} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\text{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_0^i)$.

4 Theoretical Analysis of Meta-gradient Estimators

In this section, we systematically discuss and theoretically analyse the bias and variance terms for meta-gradient estimations in the current GMRL literature. We highlight two important sources of biases in meta-gradient estimations: the compositional bias and the multi-step Hessian bias. Our analysis builds on the following three assumptions:

Assumption 4.1 (Lipschitz continuity)**.** The outer-loop objective function J^{Out} satisfies that $J^{\text{Out}}(\cdot, \boldsymbol{\theta})$ and $J^{\text{Out}}(\boldsymbol{\phi}, \cdot)$ are Lipschitz continuous with constants $m_{\boldsymbol{\theta}}$ and $m_{\boldsymbol{\phi}}$ respectively, $m_1 = \max_{\boldsymbol{\theta}} m_{\boldsymbol{\theta}}$, $m_2 = \max_{\boldsymbol{\phi}} m_{\boldsymbol{\phi}}$. $\nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\cdot, \boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} J^{\text{Out}}(\boldsymbol{\phi}, \cdot)$ are Lipschitz continuous with constants $\mu_{\boldsymbol{\theta}}$ and $\mu_{\boldsymbol{\phi}}$ respectively, $\mu_1 = \max_{\boldsymbol{\theta}} \mu_{\boldsymbol{\theta}}$, $\mu_2 = \max_{\boldsymbol{\phi}} \mu_{\boldsymbol{\phi}}$. The inner-loop objective function J^{In} satisties that $\nabla_{\boldsymbol{\theta}} J^{\text{In}}(\cdot, \boldsymbol{\theta})$and $\nabla_{\boldsymbol{\theta}} J^{\text{In}}(\boldsymbol{\phi}, \cdot)$ are Lipschitz continuous with constants $c_{\boldsymbol{\theta}}$ and $c_{\boldsymbol{\phi}}$ respectively, $c_1 = \max_{\boldsymbol{\theta}} c_{\boldsymbol{\theta}}$, $c_2 = \max_{\boldsymbol{\phi}} c_{\boldsymbol{\phi}}$. $\nabla^2_{\boldsymbol{\theta}} J^{\text{In}}(\boldsymbol{\phi}, \cdot)$ is Lipschitz continuous with constants $\rho_{\boldsymbol{\phi}}$, $\rho_2 = \max_{\boldsymbol{\phi}} m_{\boldsymbol{\phi}}$.

Assumption 4.2 (Bias of estimators)**.** Outer-loop stochastic gradient estimator $\nabla_{\boldsymbol{\phi}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau})$ and $\nabla_{\boldsymbol{\theta}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau})$ are unbiased estimator of $\nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta})$. Inner-loop stochastic gradient estimator $\nabla_{\boldsymbol{\theta}} \hat{J}^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau})$ is unbiased estimator of $\nabla_{\boldsymbol{\theta}} J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta})$.

Assumption 4.3 (Variance of estimators)**.** The outer-loop stochastic gradient estimator $\nabla_{\boldsymbol{\phi}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau})$ and $\nabla_{\boldsymbol{\theta}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\tau})$ has bounded variance, i.e., $\mathbb{E}_{\tau}[\|\nabla_{\boldsymbol{\phi}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \cdot, \boldsymbol{\tau}) - \nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\boldsymbol{\phi}, \cdot)\|^2] \leq (\sigma_1)^2$, and $\mathbb{E}_{\tau}[\|\nabla_{\boldsymbol{\theta}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \cdot, \boldsymbol{\tau}) - \nabla_{\boldsymbol{\theta}} J^{\text{Out}}(\boldsymbol{\phi}, \cdot)\|^2] \leq (\sigma_2)^2$.

The above three assumptions are all common ones adopted by existing work [6, 7, 18]. We futher discussed the limitation of assumptions, which are presented in Appendix D.

4.1 The Compositional Bias

Recall the K-step inner-loop meta-gradient estimate in Eq. (4), existing GMRL methods usually get unbiased outer-loop gradient estimator $\nabla_{\boldsymbol{\phi}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3)$ and $\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3)$, then the algorithm can get unbiased meta-gradient estimation by plugging in unbiased inner-loop gradient estimation $\hat{\boldsymbol{\theta}}^K$, where $\mathbb{E}[\hat{\boldsymbol{\theta}}^K] = \boldsymbol{\theta}^K$. However, this is not true because of the compositional optimisation structure.

Consider a non-linear compositional scalar objective $f(\boldsymbol{\theta}^K)$, the gradient estimation bias comes from the fact that

$$f(\boldsymbol{\theta}^K) = f(\mathbb{E}[\hat{\boldsymbol{\theta}}^K]) \neq \mathbb{E}[f(\hat{\boldsymbol{\theta}}^K)].$$

If one substitutes the non-linear function $f(\boldsymbol{\theta}^K)$ with $\nabla_{\boldsymbol{\theta}^K} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)$ and $\nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)$, then a typical meta-gradient estimation in GMRL introduces compositional bias:

$$\mathbb{E}[\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3)] = \mathbb{E}[\nabla_{\hat{\boldsymbol{\theta}}^K} J^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K)] \neq \nabla_{\boldsymbol{\theta}^K} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K),$$
$$\mathbb{E}[\nabla_{\boldsymbol{\phi}} \hat{J}^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3)] = \mathbb{E}[\nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K)] \neq \nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K), \tag{5}$$

which leads to meta-gradient estimation bias. The following lemma characterises compositional bias.

Lemma 4.4 (Compositional Bias)**.** *Suppose that Assumption 4.1 and 4.2 hold, let* $\hat{\Delta}_C = \mathbb{E}[\|f(\hat{\boldsymbol{\theta}}^K) - f(\boldsymbol{\theta}^K)\|]$ *be the compositional bias and* C_0 *the Lipschitz constant of* $f(\cdot)$, $|\boldsymbol{\tau}|$ *denote number of trajectories used to estimate inner-loop gradient in each inner-loop update step,* α *the learning rate, then we have,*

$$\hat{\Delta}_C \leq C_0 \mathbb{E}[\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\|] \leq C_0 \left((1 + \alpha c_2)^K - 1 \right) \frac{\hat{\sigma}_{In}}{c_2 \sqrt{|\boldsymbol{\tau}|}}, \tag{6}$$

where $\hat{\sigma}_{In} = \max_i \sqrt{\mathbb{V}[\nabla_{\boldsymbol{\theta}^i} \hat{J}^{In}(\boldsymbol{\phi}, \boldsymbol{\theta}^i, \boldsymbol{\tau}_0^i)]}$, $i \in \{0, .., K-1\}$.

Proof. See Appendix F.1 for a detailed proof. □

Lemma 4.4 indicates that the compositional bias comes from the inner-loop policy gradient estimate, concerning the learning rate α, the sample size $|\tau|$ and the variance of policy gradient estimator $\hat{\sigma}_{\text{In}}$. This is a fundamental issue in many existing GMRL algorithms [13, 39] since applying stochastic policy gradient update can introduce estimation errors, possibly due to large sampling variance, therefore $\hat{\boldsymbol{\theta}}^K \neq \boldsymbol{\theta}^K$. It also implies that the bias issue becomes more serious under the multi-step formulation since each policy gradient step introduces estimation error, resulting in composite biases.

4.2 The Multi-step Hessian Bias

Recall the analytical form of the exact meta-gradient in Eq. (3), estimating $\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}^K$ involves computing Hessian $\nabla^2_{\boldsymbol{\theta}^j} J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^j)$. In Eq. (4), the Hessian term is estimated by $\nabla^2_{\hat{\boldsymbol{\theta}}^j} J^{\text{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \tau_2^j)$ where $j \in \{1, \ldots, K-1\}$. Hessian estimation is a non-trivial problem in GMRL, biased Hessian estimation issue has been brought up in various MAML-RL papers[23, 29, 33], we offer a brief summary in Appendix C due to page limit. Beyond MAML-RL, many recent GMRL work suffers from the same bias due to applying direct automatic differentiation. For example, existing work such as Eq. (3) in Oh et al. [26], Eq. (4) in Zheng et al. [43], Eq. (12), (13), (14) in Xu et al. [39] and Eq. (5), (6), (7) in Xu et al. [40] suffers from this issue. Interestingly, most of them are **coincidentally** unbiased if they only conduct only one-step policy gradient update in the inner-loop. For K-step GMRL when $K = 1$, $\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}^1$ in meta-gradient writes as:

$$\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}^1 = \nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^1} J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^1). \tag{7}$$

We can see from Eq. (7) that it would not involve Hessian $\nabla^2 J^{\text{In}}$ computation if $\boldsymbol{\phi} \neq \boldsymbol{\theta}$. To further illustrate, in one-step MGRL, we can show that the estimation of $\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}^1$ are unbiased because it takes derivatives *w.r.t* meta-parameters $\boldsymbol{\phi} = (\gamma, \lambda)$ which don't have gradient dependency on the trajectory distribution. However, when it takes more than one-step inner-loop policy gradient updates, the meta-gradient estimation will get the hessian estimation bias. As a result, for the reason that when $K > 1$ in K-step GMRL, the $\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}^K$ in meta-gradient takes the form:

$$\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}^K = \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^i} J^{\text{In}}\left(\boldsymbol{\phi}, \boldsymbol{\theta}^i\right) \prod_{j=t+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\text{In}}\left(\boldsymbol{\phi}, \boldsymbol{\theta}^j\right)\right). \tag{8}$$

This is the reason why we name it by **multi-step** Hessian bias.

4.3 Theoretical Bias-Variance Analysis

Based on Lemma 4.4 and the discussion in Sec. 4.2, we can derive the upper bound on the bias and variance of the meta-gradient with K-step inner-loop updates.

Theorem 4.5 (Upper bound for the bias and the variance)**.** *Suppose that Assumption 4.1 and 4.2 and 4.3 hold. Let $J_{\boldsymbol{\phi},\boldsymbol{\theta}}$ denote $\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}} J^{In}$, $H_{\boldsymbol{\theta},\boldsymbol{\theta}}$ denote $\nabla^2_{\boldsymbol{\theta}} J^{In}$, $\hat{\Delta}^K = \|\mathbb{E}[\nabla_{\boldsymbol{\phi}}\hat{J}^K(\boldsymbol{\phi})] - \nabla_{\boldsymbol{\phi}} J^K(\boldsymbol{\phi})\|$ be the meta-gradient estimation bias, set $B = 1 + \alpha c_2$. Then the bound of bias hold:*

$$\hat{\Delta}^K \leq O\Big((B + \hat{\Delta}_H)^{K-1}\left(\mathbb{E}[\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\|] + \hat{\Delta}_J + (K-1)\right)\Big). \tag{9}$$

Let $(\hat{\sigma}^K)^2 = \mathbb{V}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^K(\boldsymbol{\phi})\right]$ be the meta-gradient estimation variance, set $V_1 = (1 + \alpha c_2)^2$, $V_2 = 2\alpha^2(m_1^2 + 3\sigma_2^2)$ the estimation variance is given by

$$\begin{aligned}(\hat{\sigma}^K)^2 \leq O\Bigg(&(V_1 + \hat{\Delta}_H^2)^{K-1}\left(\mathbb{E}[\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\|^2] + (K-1)\right) \\ &+ \left(V_2 + (V_1 + \hat{\Delta}_H^2 + \hat{\sigma}_H^2)^{K-1} - (V_1 + \hat{\Delta}_H^2)^{K-1}\right)(\hat{\Delta}_J^2 + \hat{\sigma}_J^2)\Bigg).\end{aligned} \tag{10}$$

where $\hat{\Delta}_J = \max_{\boldsymbol{\phi}\times\boldsymbol{\theta}} \|\mathbb{E}[\hat{J}_{\boldsymbol{\phi},\boldsymbol{\theta}}] - J_{\boldsymbol{\phi},\boldsymbol{\theta}}\|$. $\hat{\Delta}_H = \max_{\boldsymbol{\theta}} \|\mathbb{E}[\hat{H}_{\boldsymbol{\theta},\boldsymbol{\theta}}] - H_{\boldsymbol{\theta},\boldsymbol{\theta}}\|$. $(\hat{\sigma}_J)^2 = \frac{\max_{\boldsymbol{\phi}\times\boldsymbol{\theta}} \mathbb{V}[\hat{J}_{\boldsymbol{\phi},\boldsymbol{\theta}}]}{|\tau|}$. $(\hat{\sigma}_H)^2 = \frac{\max_{\boldsymbol{\theta}} \mathbb{V}[\hat{H}_{\boldsymbol{\theta},\boldsymbol{\theta}}]}{|\tau|}$.

Proof. See Appendix G.1 for a detailed proof. □

Theorem 4.5 shows that the upper bound of bias and variance consists of two parts: the first term indicates compositional bias $\mathbb{E}[\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\|]$ in Lemma 4.4, while the second term refers to the second-order estimation bias $(\hat{\Delta}_J, \hat{\Delta}_H)$ and the variance $(\hat{\sigma}_J, \hat{\sigma}_H)$. Several observations can be made based on Theorem 4.5: 1) the compositional bias exists in both upper bound of bias and variance; 2) the multi-step Hessian bias has polynomial impact on the upper bound of bias and variance; 3) most importantly, many existing GMRL algorithms suffer from the compositional bias; moreover, the Hessian bias can significantly increase meta-gradient bias in the multi-step inner-loop setting.

4.4 Understanding Existing Mitigations for Meta-gradient Biases

Based on the above theoretical analysis, we now try to explore and understand how existing methods can handle these two estimation biases.

Off-policy Learning. From the Lemma 4.4 and the discussion in Sec. 4.1, we know that the compositional bias is caused by the estimation error between $\hat{\theta}_t^K$ and θ_t^K. In this case, a simple idea is that one can leverage off-policy learning technique [32] to handle the compositional bias problem by reusing samples $\tau_{0:t-1,0}^i$ together with $\tau_{t,0}^i$ to approximate θ_t^K, we want $\hat{\theta}_t^K$ to stay close to θ_t^K. The intuition behind this correction is to enlarge the sample size $|\boldsymbol{\tau}|$ so that the compositional bias can be minimised according to Lemma 4.4. Specifically, we can apply importance sampling technique to correct the compositional bias, i.e., $\mathbb{E}_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau};\boldsymbol{\theta})}\left[\sum_{t=0}^{H-1} \frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\mu(\boldsymbol{a}_t|\boldsymbol{s}_t)}\mathcal{R}_{\boldsymbol{\phi}}(\boldsymbol{\tau})\right]$. In practice, we also need to manually control the level of off-policyness to prevent the variance introduced by the importance sampling process from increasing the estimation variance conversely.

Mutli-step Hessian Estimator. From the theoretical analysis in Sec. 4.3, we can know that the Hessian estimation bias can significantly increase the meta-gradient estimation bias in the multi-step inner-loop setting. Many low-bias hessian estimator have been proposed in the scope of MAML-RL. However, the effect of them have never been verified in the general GMRL context. Here we have a second look at the Low Variance Curvature (LVC) method [29], one can replace the original log-likelihood with the LVC operator in the policy gradient step. As such, the Hessian estimator $\nabla_\theta^2 J_{\text{LVC}}^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ takes the form: $\nabla_\theta \mathbb{E}_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau};\boldsymbol{\theta})}\left[\sum_{t=0}^{H-1} \frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\perp\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}\mathcal{R}_{\boldsymbol{\phi}}(\boldsymbol{\tau})\right]$ where $\perp$ is the stop-gradient operation which detaches the gradient dependency from the computation graph. As shown in [29], the LVC operator will ensure an unbiased first-order policy gradient and low-biased low-variance second-order policy gradient. Essentially, this operator only corrects the meta-gradient update and leaves the inner-loop gradient estimation formula untouched.

5 Experiments

In this section we conduct empirical evaluation of our proposed bias analysis and the proposed methods to mitigate the biases, and our experiments cover all 3 GMRL fields listed in Table 1. In particular, we conduct a tabular MDP experiment to show the existence of two biases discussed in Sec. 4 using MAML [10] and LIRPG [42]. In order to show how the proposed methods can mitigate the compositional bias, we consider a Iterated Prisoner Dilemma (IPD) problem and use LOLA [13] with off-policy corrections in Sec. 4.4; Similarly, we conduct evaluations on Atari games using MGRL [39] with LVC corrections in Sec. 4.4 to mitigate the Hessian estimation bias. We open source our code at `https://github.com/Benjamin-eecs/Theoretical-GMRL`.

5.1 Investigate the correlation and bias of meta-gradient estimators

Firstly, we conduct experiments to study different meta-gradient estimators using a tabular example using MAML and LIRPG. To align with existing works in the literature, we adopt the settings of random MDPs in [33] with the focus on meta-gradient estimation. We refer readers to Appendix I.1.1 for more experimental details.

MAML-RL. Recall that the meta-gradient in MAML is $\nabla_{\boldsymbol{\theta}^0} J^{\text{Out}}(\boldsymbol{\theta}^K)$. To control the effect of gradient estimations, we use the three estimators to estimate following three terms: (I) inner-loop policy gradient $\nabla_{\boldsymbol{\theta}} J^{\text{In}}(\boldsymbol{\theta})$; (II) Jacobian/Hessian $\nabla_{\boldsymbol{\theta}}^2 J^{\text{In}}(\boldsymbol{\theta})$; (III) outer-loop policy gradient $\nabla_{\boldsymbol{\theta}^K} J^{\text{Out}}(\boldsymbol{\theta}^K)$. Refer to Appendix I.1.2 for the implementation of decomposing meta-gradient estimation with different estimators.

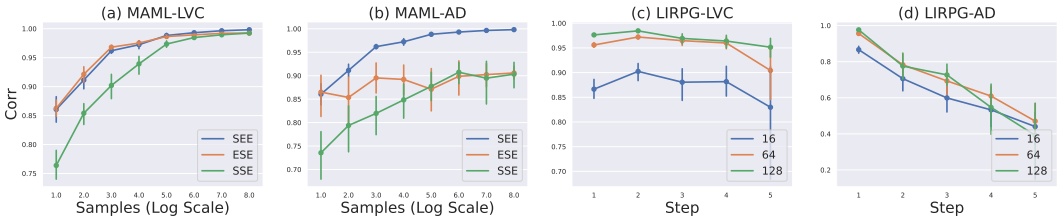


Figure 1: (a, b) Ablation study for sample size and estimators in MAML-RL. “S" is for stochastic estimation while “E" is for exact solution. AD refers to automatic differentiation. (c,d) Ablation study on sample size, steps and estimators in LIRPG.

To show how the estimation is biased, we use a stochastic estimator denoted as S, and exact analytic calculator denoted as E, for all three derivative terms. Thus, we can have 7 valid permutations in the experiment to validate the estimation, where the rest EEE estimator is the exact gradient.

Firstly, we conduct our ablation studies by comparing the correlation between of meta-gradient with the exact one. The correlation metric, which is determined by bias and variance, can show how the final estimation quality is influenced by these two bias terms. As the quality of stochastic estimators vary from many factors, we conduct this ablation study under extensive combinations of estimation algorithms (including DiCE [12], Loaded-DiCE [8], LVC [29], and pure automatic differentiation in original MAML [10]), learning rates, sample sizes, .etc. Due to the page limit, the results illustrated in Fig. 1(a,b) only include ablation study on sample size using LVC/automatic differentiation and estimation using exact gradients for (III), the outer-loop policy gradient. For the rest ablation study and more evaluation metrics (variance of estimation), refer to Appendix I.1.3.

We start our evaluation by increasing the sample size. For simplicity, we only conduct one inner step here. In Fig. 1(a), we leverage the LVC [29] estimator. We can see that a correct inner-loop policy estimation and/or a Hessian estimation can significantly improve the estimation quality (as SEE $\approx$ ESE $>$ SSE in low sample size case). By increasing the sample size, we can see the gap between them is shrinking, which verifies the finding in Lemma 4.4. In this case the compositional bias correction shares the same importance with the Hessian bais correction (SEE $\approx$ ESE). We also compare it with the original yet biased gradient estimator of MAML in Fig. 1(b), in which SEE $>$ ESE in all sample size settings. In fact, since the gradient estimation is biased, only SEE achieves near 1.0 correlation provided sufficient samples, again confirms the importance of Hessian bias corrections in Sec. 4.2.

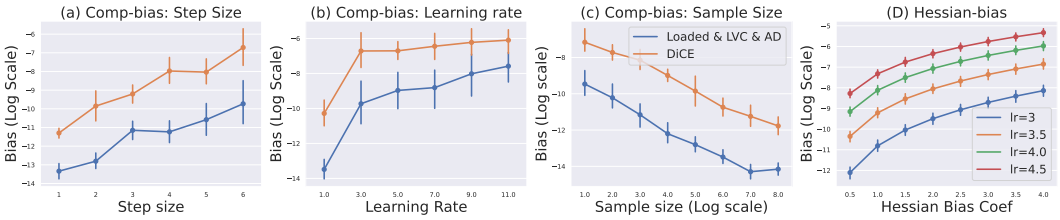


Figure 2: (a, b, c) Ablation study of meta-gradient bias due to the compositional bias in different estimators, step sizes, learning rates. Loaded-DiCE, LVC and AD achieve exactly the same compositional bias because they have the same first-order gradient, (d) Ablation study of meta-gradient bias due to the Hessian bias in different learning rates and Hessian bias coefficients.

Beside the correlation result above, we also add additional experimental results in Fig. 2 over the pure meta-gradient bias term introduced by compositional bias and Hessian bias. It can also be regarded as an empirical verification of our Lemma 4.4 and Theorem 4.5. In Fig. 2 (a, b, c), we mainly study How (a) the inner-loop step size, (b) learning rate and (c) sample size influence final meta-gradient bias. It successfully validates our Lemma 4.4 ($O\big(K\alpha^K\hat{\sigma}_{\text{In}}|\tau|^{-0.5}\big)$) about the exponential impact from the inner-loop step K (approximately linear relationship between log-scale bias and step size in (a)), the polynomial impact from the learning rate α (approximately Concave downward relationship between log-scale bias and learning rate in (b)) and the polynomial impact

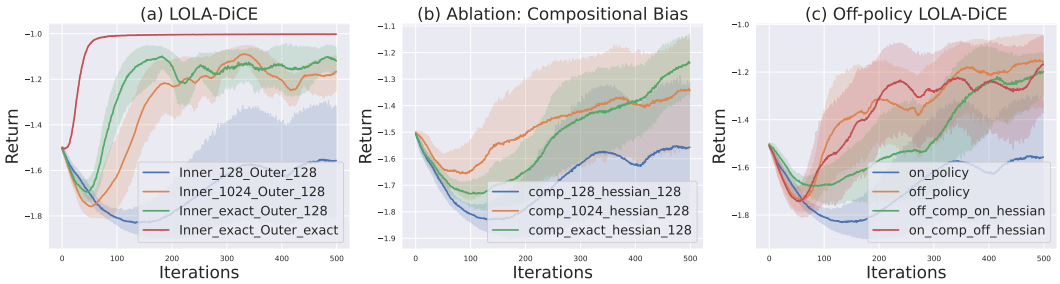


Figure 3: Experiment result of LOLA-DiCE over 10 seeds. The Inner_A_Outer_B legend means we use A samples to estimate inner-loop gradient while B samples to estimate outer-loop gradient. The 'exact' means we use analytical solution of policy gradient instead of estimation.

from the sample size α (approximately negative linear relationship between log-scale bias and log-scale sample size in (c)). In the second MAML-Hessian experiment, we conduct experiments to verify the polynomial impact $O\big((K-1)(\hat{\Delta}_H)^{K-1}\big)$ on the meta-gradient bias introduced by the multi-step Hessian estimation bias $\hat{\Delta}_H$ (The Concave downward relationship in (d)). In our implementation, we manually add the Hessian bias error into the estimation and control the quantity of it by multiplying different coefficients.

LIRPG. In this setting, we follow the algorithm of intrinsic reward generator presented in [42]. In tabular MDP, we have an additional meta intrinsic reward matrix ϕ. Starting from $\boldsymbol{\theta}^0$, the inner-loop process takes policy gradient based on the new reward matrix $R_{\text{new}} = R + \boldsymbol{\phi}$: $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i + \alpha \nabla_{\boldsymbol{\theta}^i} J^{\text{In}}(\boldsymbol{\theta}^i, \phi), i \in \{0, 1 \ldots K-1\}$. The meta-gradient estimation of the intrinsic reward matrix $\nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\boldsymbol{\theta}^K)$ is needed in this case. Note that in the outer loss we use the original reward matrix R so the outer loss is $J^{\text{Out}}(\boldsymbol{\theta}^K)$ rather than $J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)$. Compared with MAML-RL, the object of meta-update (intrinsic matrix) and the object of inner-update (policy parameters) are different, which help us identify the problem mentioned in Sec. 4.2.

In this case, we choose the LVC and AD estimator. We conduct ablation study on inner-step and sample size shown in Fig. 1(c,d). With more sample size and less step size, the correlation increases for both estimator. Two important features are: (1) With 1-step inner-loop setting, both estimator performs similarly in the correlation. (2) With multi-step inner-loop setting, LVC based estimator can still reach relatively high correlation while MAML-biased estimator directly reaches low correlation after 5-step inner-loop. The phenomenon shown here corresponds exactly to the Hessian estimation issue we discuss in Sec. 4.2 and the bias issue will be more severe with multi-step inner-loop setting.

5.2 Compositional bias/off-policy learning in LOLA

In this subsection, we conduct three experiments on Iterated Prisoner Dilemma (IPD) with the LOLA algorithm to show: (1) The effect brought by different inner/outer estimators. (2) The effect brought compisitional bias (Sec. 4.1) (3) How off-policy correction (Sec. 4.4) can help the LOLA algorithm. Please refer to Appendix I.2.1, I.2.2 for more experimental setting and results.

Ablation on LOLA-DiCE inner/outer estimation. We report the result of conducting ablation study for different inner/outer-loop estimation of LOLA-DiCE in the Fig. 3(a). Here the inner-loop estimation refers to $\nabla_{\boldsymbol{\theta}} J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ while outer-loop estimation refers to $\nabla_{\boldsymbol{\theta}^1} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^1)$ and $\nabla_{\boldsymbol{\phi}} J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^1)$. The return shown in Fig. 3(a) reveals us two findings: 1) The inner-loop gradient estimation plays an important role for making LOLA work—the default batch size 128 fails while the batch size 1024 succeeds. 2) The outer-loop gradient estimation is also crucial to the performance of LOLA-Exact. Furthermore, we continue conducting ablation studies on the inner-loop gradient update.

Ablation on compositional bias. Since the unbiased DiCE estimator is used in LOLA-DiCE algorithm, there is no Hessian estimation bias in the LOLA algorithm. Thus, we mainly discuss the problem of compositional bias brought in Fig. 3(b). We also apply the implementation in Sec. 5.1 to decompose meta-gradient estimation with different estimators. Fig. 3(b) show us the ablation study over compositional bias, which reveals that: compositional bias may decrease the performance and by adding more samples or using analytical solution, the performance can start to improve.

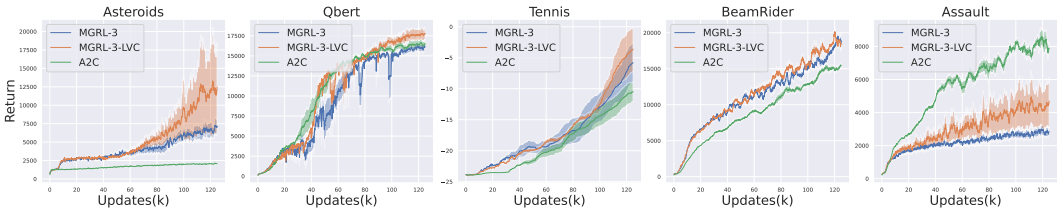


Figure 4: Experimental results on Atari game over 5 random seeds.

Off-policy DiCE and ablation study. We use the off-policy learning to conduct inner-loop update and keep the outer-loop gradient same as before. By combing DiCE and off-policy learning, we have off-policy DiCE $J^{\text{OFF-DICE}}$: $\mathbb{E}_{\boldsymbol{\tau}}\left[\sum_{t=0}^{H-1}\left(\prod_{t'=0}^{t}\frac{\pi_\phi(\boldsymbol{a}^1_{t'}|\boldsymbol{s}^1_{t'})\pi_\theta(\boldsymbol{a}^2_{t'}|\boldsymbol{s}^2_{t'})}{\mu_1(\boldsymbol{a}^1_{t'}|\boldsymbol{s}^1_{t'})\mu_2(\boldsymbol{a}^2_{t'}|\boldsymbol{s}^2_{t'})}\right)R_t\right]$, where ϕ, θ refer to the current policy, μ_1, μ_2 refer the behaviour policy for agent 1 and agent 2, respectively. H is the trajectory length and R refers to the reward for agent. Note that the off-policy DiCE here can not only lower the compositional bias by lowering the first-order policy gradient error, but also helps lower the Hessian variance theoretically. By the decomposition trick, we conduct experiments by traversing over all learning settings for (off-off/off-on/on-off/on-on), which are shown at Fig. 3(d). Comparisons between different settings verify that off-policy DiCE can increase performance by either lowering the compositional bias, or the hessian variance, or both.

5.3 Multi-step Hessian correction on MGRL

Finally, we conduct experiment over MGRL [39]. See Appendix I.3.1 and I.3.2 for experimental settings. When applying the LVC estimator in MGRL, we get the new inner-loop update equation: $\mathbb{E}_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau};\boldsymbol{\theta})}\left[\sum_{t=0}^{H-1}\frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\perp\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}\left(g_{\boldsymbol{\phi}}(\boldsymbol{\tau})-v_{\boldsymbol{\theta}}(\boldsymbol{s}_t)\right)\right]$, where $\boldsymbol{\phi} = (\gamma, \lambda)$ refer to meta-paramters and θ refers to the RL policy, $g_{\boldsymbol{\phi}}(\tau)$ denotes λ-return, $v_{\boldsymbol{\theta}}(s_t)$ denotes value prediction. We conduct experiment on eight environments of Atari games. We follow previous work [4] to use the "discard" strategy in which we conduct multiple virtual inner-loop updates for meta gradient estimation. This strategy is designed to keep the inner learning update unchanged.

In Fig. (4), we show five environments comparing three variants of algorithm: 1) Baseline Advantage Actor-critic(A2C) algorithm [25]; 2) 3-step MGRL + A2C; 4) 3-step MGRL + A2C + LVC correction. The "3-step" means we take 3 inner-loop RL virtual updates for calculating meta-gradient. Refer to Appendix I.3.3 for experimental results on all eight environments. Compared with 3-step MGRL, the MGRL with LVC correction can substantially improves the performance, which validates the effectiveness of the multi-step Hessian correction in Sec. 4.4 for handling meta-gradient estimation bias and bring in better hyperparameter-tuning in RL. Note that the fact that A2C algorithms can achieve better results compared with 3-step MGRL is consistent with the results in [39].

6 Conclusion

In this paper, we introduce a unified framework for studying generic meta-gradient estimations in gradient-based Meta-RL. Based on this framework, we offer two theoretical insights that **1)** the compositional bias has an upper bound of $O\big(K\alpha^K\hat{\sigma}_{\text{In}}|\tau|^{-0.5}\big)$ with respect to the inner-loop update step K, the learning rate α, the estimate variance $\hat{\sigma}^2_{\text{In}}$ and the sample size $|\tau|$, and **2)** the multi-step Hessian bias $\hat{\Delta}_H$ has a polynomial impact of $O\big((K-1)(\hat{\Delta}_H)^{K-1}\big)$. To validate our theoretical discoveries, we conduct a comprehensive list of ablation studies. Empirical results over tabular MDP, LOLA-DiCE and MGRL validate our theories and the effectiveness of correction methods. We believe our work can inspire more future work on unbiased meta-gradient estimations in GMRL.

Acknowledgements

We would like to thank Yunhao Tang for insightful discussion and help for tabular experiments.

References

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

[2] Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. *arXiv preprint arXiv:1703.04782*, 2017.

[3] Sarah Bechtle, Artem Molchanov, Yevgen Chebotar, Edward Grefenstette, Ludovic Righetti, Gaurav Sukhatme, and Franziska Meier. Meta learning via learned loss. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4161–4168. IEEE, 2021.

[4] Clément Bonnet, Paul Caron, Thomas Barrett, Ian Davies, and Alexandre Laterre. One step at a time: Pros and cons of multi-step meta-gradient reinforcement learning. *arXiv preprint arXiv:2111.00206*, 2021.

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

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

[7] Alireza Fallah, Kristian Georgiev, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of debiased model-agnostic meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3096–3107, 2021.

[8] Gregory Farquhar, Shimon Whiteson, and Jakob Foerster. Loaded dice: trading off bias and variance in any-order score function estimators for reinforcement learning. *arXiv preprint arXiv:1909.10549*, 2019.

[9] Xidong Feng, Oliver Slumbers, Ziyu Wan, Bo Liu, Stephen McAleer, Ying Wen, Jun Wang, and Yaodong Yang. Neural auto-curricula in two-player zero-sum games. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[11] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130, 2018.

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

[13] Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

[14] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR, 2017.

[15] Haotian Fu, Hongyao Tang, Jianye Hao, Chen Chen, Xidong Feng, Dong Li, and Wulong Liu. Towards effective context for meta-reinforcement learning: an approach based on contrastive learning. *arXiv preprint arXiv:2009.13891*, 2020.

[16] Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *Advances in Neural Information Processing Systems*, 31:5302–5311, 2018.

[17] Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems*, 33, 2020.

[18] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Multi-step model-agnostic meta-learning: Convergence and improved algorithms. *CoRR*, abs/2002.07836, 2020.

[19] Dong-Ki Kim, Miao Liu, Matthew Riemer, Chuangchuang Sun, Marwa Abdulhai, Golnaz Habibi, Sebastian Lopez-Cot, Gerald Tesauro, and Jonathan P How. A policy gradient algorithm for learning to learn in multiagent reinforcement learning. *arXiv preprint arXiv:2011.00382*, 2020.

[20] Louis Kirsch, Sjoerd van Steenkiste, and Juergen Schmidhuber. Improving generalization in meta reinforcement learning using learned objectives. In *International Conference on Learning Representations*, 2019.

[21] Ilya Kostrikov. Pytorch implementations of reinforcement learning algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

[22] Alistair Letcher, Jakob Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson. Stable opponent shaping in differentiable games. In *International Conference on Learning Representations*, 2018.

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

[24] Jingkai Mao, Jakob Foerster, Tim Rocktäschel, Maruan Al-Shedivat, Gregory Farquhar, and Shimon Whiteson. A baseline for any order gradient estimation in stochastic computation graphs. In *International Conference on Machine Learning*, pages 4343–4351. PMLR, 2019.

[25] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

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

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

[28] Jie Ren, Xidong Feng, Bo Liu, Xuehai Pan, Luo Mai, and Yaodong Yang. Torchopt. https://github.com/metaopt/torchopt, 2022.

[29] Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[30] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[31] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

[32] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[33] Yunhao Tang, Tadashi Kozuno, Mark Rowland, Remi Munos, and Michal Valko. Unifying gradient estimators for meta-reinforcement learning via off-policy evaluation. *Advances in Neural Information Processing Systems*, 34, 2021.

[34] Vivek Veeriah, Matteo Hessel, Zhongwen Xu, Richard Lewis, Janarthanan Rajendran, Junhyuk Oh, Hado van Hasselt, David Silver, and Satinder Singh. Discovery of useful questions as auxiliary tasks. *arXiv preprint arXiv:1909.04607*, 2019.

[35] Vivek Veeriah, Tom Zahavy, Matteo Hessel, Zhongwen Xu, Junhyuk Oh, Iurii Kemaev, Hado van Hasselt, David Silver, and Satinder Singh. Discovery of options via meta-learned subgoals. *arXiv preprint arXiv:2102.06741*, 2021.

[36] Risto Vuorio, Jacob Austin Beck, Gregory Farquhar, Jakob Nicolaus Foerster, and Shimon Whiteson. No DICE: An investigation of the bias-variance tradeoff in meta-gradients. In *Deep RL Workshop NeurIPS 2021*, 2021.

[37] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[38] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[39] Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. *arXiv preprint arXiv:1805.09801*, 2018.

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

[41] Tom Zahavy, Zhongwen Xu, Vivek Veeriah, Matteo Hessel, Junhyuk Oh, Hado van Hasselt, David Silver, and Satinder Singh. A self-tuning actor-critic algorithm. *arXiv preprint arXiv:2002.12928*, 2020.

[42] Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. *Advances in Neural Information Processing Systems*, 31:4644–4654, 2018.

[43] Zeyu Zheng, Junhyuk Oh, Matteo Hessel, Zhongwen Xu, Manuel Kroiss, Hado Van Hasselt, David Silver, and Satinder Singh. What can learned intrinsic rewards capture? In *International Conference on Machine Learning*, pages 11436–11446. PMLR, 2020.

[44] Wei Zhou, Yiying Li, Yongxin Yang, Huaimin Wang, and Timothy Hospedales. Online meta-critic learning for off-policy actor-critic methods. *Advances in Neural Information Processing Systems*, 33, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix B.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sec. 4
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix F and Appendix G
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] , See Appendix I
(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] , See Fig. 1, 3, 4.
(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix I

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material

The supplementary material is organized as follows. Appendix A offers more algorithm illustration for the 4 topics we discuss in Sec. 3. In Appendix B we discuss our choice for EPG formulation and the truncated setting in GMRL. In Appendix C we briefly summarise biased Hessian estimation issue in MAML-RL mentioned in Section 4.2, In Appendix D we illustrate how realistic are Assumption 4.1-4.3 of Section 4. Appendices E to G contain the proofs for the results presented in the paper. In Appendix H we provide statements and proofs for some auxiliary lemmas which are instrumental for the main results. For convenience of the reader, before each proof we also restate the corresponding theorem. Finally, in Appendix I we present additional experiments results.

A More topics on GMRL

A.1 Few-shot Reinforcement Learning

One important research field in Meta Reinforcement Learning is few-shot Reinforcement Learning. The main objective of this research field is to enable Reinforcement Learning agent with fast adaptation ability. Instead of thousands of interactions in traditional Reinforcement Learning algorithms, agent in few-shot setting is only allowed to interact with the new environment for a few trajectories. One of the most classical gradient based algorithms in this field is **Model Agnostic Meta Learning (MAML-RL)**. [10] aims at learning neural network's initial parameters for fast adaptation on new environments. It assumes distribution $\rho(\mathcal{T})$ over RL environment $\mathcal{T}$ and tries to optimise $\boldsymbol{\theta}$ which leads to high-performing updated policy $\boldsymbol{\theta}'$. The objective equation for one-step MAML-RL can be shown as follows:

$$\begin{aligned} J(\boldsymbol{\theta}) &= \mathbb{E}_{\mathcal{T}\sim\rho(\mathcal{T})}\left[\mathbb{E}_{\boldsymbol{\tau}'\sim P_{\mathcal{T}}(\boldsymbol{\tau}'|\boldsymbol{\theta}')}\left[R\left(\boldsymbol{\tau}'\right)\right]\right] \\ \text{with} \quad \boldsymbol{\theta}' &= \boldsymbol{\theta} + \alpha\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\tau}\sim P_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{\theta})}[R(\boldsymbol{\tau})] \end{aligned} \tag{11}$$

where in practice we use the limited trajectories sampled from the new environment to estimate $\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\tau}\sim P_{\mathcal{T}}(\boldsymbol{\tau}|\boldsymbol{\theta})}[R(\boldsymbol{\tau})]$. During training, by estimating meta policy gradient $\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta})$, MAML can conduct meta update on the initial policy parameters.

In the scope of Eq. (2), MAML-RL optimizes over meta initial parameters to maximize the return of one-step adapted policy: $\boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha\nabla_{\boldsymbol{\theta}}J^{\text{In}}(\boldsymbol{\theta})$. In MAML-RL, $J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}')$ degenerates to $J^{\text{Out}}(\boldsymbol{\theta}')$ and $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ represent the same initial parameters. The meta-gradient can be derived with the following equation:

$$\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\boldsymbol{\theta}'\nabla_{\boldsymbol{\theta}'}J^{\text{Out}}\left(\boldsymbol{\theta}'\right), \nabla_{\boldsymbol{\theta}}\boldsymbol{\theta}' = I + \alpha\nabla^2_{\boldsymbol{\theta}}J^{\text{In}}(\boldsymbol{\theta}) \tag{12}$$

A.2 Meta-gradient in Opponent Shaping

Opponent shaping [11, 19, 22] is a powerful tool in multi-agent learning process for different purposes. For instance, Foerster et al. [11] and Letcher et al. [22] have shown that putting other-players learning dynamic into self-learning process can bring in cooperation behaviors, which may help to reach better social welfare compared with purely independent learning. Meta-gradient estimation is needed when ego-agent takes derivatives of other-agent policy gradient step. **Learning with Opponent-Learning Awareness (LOLA)** [11] proposed a new learning objective by including an additional term accounting for the impact of ego policy to the anticipated opponent gradient update. Specifically, in the two-player setting, with agent 1 policy $\boldsymbol{\phi}$ and agent 2 policy $\boldsymbol{\theta}$, the traditional independent learning (IL) and 1-step LOLA algorithm can result in different updates for agent 1:

$$\begin{aligned} &\boldsymbol{\phi}'_{\text{IL}} = \boldsymbol{\phi} + \beta\nabla_{\boldsymbol{\phi}}J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}) \\ &\boldsymbol{\phi}'_{\text{LOLA}} = \boldsymbol{\phi} + \beta\nabla_{\boldsymbol{\phi}}J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}') \\ &\text{where } \boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha\nabla_{\boldsymbol{\theta}}J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}) \end{aligned} \tag{13}$$

Where β refers to the outer learning rate and $J^{\text{In/Out}}$ refers to the value function for agent 2 and agent 1 respectively. For meta-agent 1 with parameters $\boldsymbol{\phi}$, it will optimise its return over one-step-lookahead opponent parameters $\boldsymbol{\theta}'$. Thus the meta-gradient of meta-agent corresponds exactly to Eq. (2) with $\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}' = \alpha\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}}J^{\text{In}}(\boldsymbol{\phi}, \boldsymbol{\theta})$. Note that this one-step-lookahead is a just virtual update considered in the optimisation of agent 1. Agent 2 can also choose this LOLA update by conducting one-step-lookahead over agent 1.

A.3 Single-lifetime Meta-gradient RL

In this setting, the main objective is to self-tune the meta parameters (γ in [39]) or meta models (intrinsic model in [42]) along with the underlying normal RL updates. It is called online because it only involves one single RL life-time. This research field is also related with online hyperparameter optimisation in supervised learning such as [2, 14]. Xu et al. [39] proposed meta-gradient reinforcement learning (MGRL) to tune the discount factor γ and bootstrapping parameter λ in an online manner. It tries to differentiate through one RL inner update to optimize the meta-parameters and maximise one-step policy return.

$$\begin{aligned} &\max_{\boldsymbol{\eta}} V^{\pi_{\theta'}}, \text{where } \boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\tau}, \boldsymbol{\theta}, \boldsymbol{\eta}), \text{and} \\ \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\tau}, \boldsymbol{\theta}, \boldsymbol{\eta}) &= \left(g_{\boldsymbol{\eta}}(\boldsymbol{\tau}) - v_{\boldsymbol{\theta}}(S)\right) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A \mid S) \\ &\quad + \left(g_{\boldsymbol{\eta}}(\boldsymbol{\tau}) - v_{\boldsymbol{\theta}}(S)\right) \nabla_{\boldsymbol{\theta}} v_{\boldsymbol{\theta}}(S) \\ &\quad + \nabla_{\boldsymbol{\theta}} H\left(\pi_{\boldsymbol{\theta}}(\cdot \mid S)\right) \end{aligned} \tag{14}$$

where η refers to (γ, λ), $\boldsymbol{\tau}$ refers to trajectories, $g_{\boldsymbol{\eta}}$, $v_{\boldsymbol{\theta}}$, H represent GAE estimation, value function and entropy respectively. Eq. (14) combines actor loss, critic loss and entropy loss, which are commonly used in typical Actor-Critic [25] algorithms. Specifically, the meta parameters (γ, λ) corresponds to $\boldsymbol{\phi}$ in Eq. (2) . After the policy parameters $\boldsymbol{\theta}$ take one policy gradient update to become $\boldsymbol{\theta}'(\boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J^{\text{In}}(\boldsymbol{\theta}, \boldsymbol{\phi}))$, we can calculate the meta-gradient by backpropogating from J^{Out} to meta parameters. In MGRL, $J^{\text{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}')$ degenerates to $J^{\text{Out}}(\boldsymbol{\theta}')$. The meta-gradient can be shown as:

$$\nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}' \nabla_{\boldsymbol{\theta}'} J^{\text{Out}}\ (\boldsymbol{\theta}'), \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}' = \alpha \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} J^{\text{In}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \tag{15}$$

Here for simplicity we omit the critic and entropy loss. Usually work in this research field only conduct one-step inner-loop update before taking meta update. Some recent works such as [34, 4] have also shown that multi-step online meta-gradient can achieve better performance.

A.4 Multi-lifetime Meta-gradient RL

Existing work like [26, 43, 40, 9] are trying to learn some fundamental/generalizable meta module across different environments such as a neural RL algorithm in [26](LPG). An important feature of multi-lifetime Meta-gradient RL is that it inherently needs multi-step inner-loop to account for the effect of fundamental meta module over the RL process. The objective of LPG is to learn a neural network based RL algorithm, by which a RL agent can be properly trained. The mathmatical formulation can be shown as follows:

$$J(\boldsymbol{\phi}) = \mathbb{E}_{\mathcal{T} \sim \rho(\mathcal{T})} \left[\mathbb{E}_{\boldsymbol{\tau}^K \sim P_{\mathcal{T}}(\boldsymbol{\tau}^K \mid \theta^K)} \left[R\left(\boldsymbol{\tau}^K\right) \right] \right], \text{with} \tag{16}$$

$$\theta^i = \theta^{i-1} + \alpha \nabla_{\theta^{i-1}} \mathbb{E}_{\boldsymbol{\tau} \sim P_{\mathcal{T}}(\boldsymbol{\tau} \mid \theta^{i-1})} \left[f_{\boldsymbol{\phi}}(\boldsymbol{\tau}) \right] \tag{17}$$

where $f_{\boldsymbol{\phi}}(\boldsymbol{\tau})$ is the output of meta-network $\boldsymbol{\phi}$ for conducting inner-loop neural policy gradient and k can be large to show the long-range impact brought by neural RL algorithm. We omit the kl inner loss used in [26] for simplicity. In the scope of Eq. (1), $J^{\text{In/Out}}$ refers to the value function, $\boldsymbol{\theta}$ represents the RL agent policy parameters and $\boldsymbol{\phi}$ is the meta-parameter of neural RL algorithm. Most of works are under a multi-task/environment (or a distribution over environment) and multi-lifetime setting. [40] is a special case in these work because it is also under the online setting. We believe the main reason is that the training iterations/sample complexity in [40] is real large (1e9) and makes it become a special case of 'multi-lifetime' setting.

B Discussion of expected policy gradient (EPG) formulation and truncated setting

We discuss 4 research topics in Section 3: few-shot RL(MAML-RL), opponent shaping(LOLA-DiCE), online meta gradient RL(MGRL) and meta gradient based inverse design(LPG). And we need to discuss how this multi-step EPG inner-loop formulation differs in these topics. Though they all need meta policy gradient estimation, the differences between setting and final objective require us to discuss them separately.

Different setting: MAML-RL and most inverse design algorithms are under multi-lifetime setting which can renew an environment and restart the RL training from the very beginning. Work in online

meta gradient RL/LOLA only happen in a single lifetime RL process. There only exists one RL training process.

Different objectives: For MAML-RL, the main objective is to maximise the return of few-step adapted policy. Thus the objective corresponds exactly to few-step inner-loop formulation. However, for topics beyond few-shot RL, in most case they need to measure the influence of meta module over RL final (after thousands of steps) performance.

There are two important issues in this EPG formulation. The first one is that it assumes an expected policy gradient inner-loop update. And the second one is because we only consider few-step inner-loop update so they are under a truncated estimation setting which might bring in bias. Recently, one work [36] argues that: (1) the general unbiased meta gradient for MAML-RL ([10]) and Online Meta Gradient ([39],[42]) should be the K-sample inner-loop meta gradient shown in E-MAML [1] rather than the expected policy gradient inner-loop meta gradient used in many recent work [23, 29, 33]. (2) The gradient estimator in online meta gradient utilise truncated optimization and the unbiased meta gradient should be the one in untruncated setting.

Overall we agree that: (1) The K-sample inner-loop meta gradient estimator is unbiased for MAML-RL problem when sampled policy gradient are used. (2) To learn an schedule (rather than a global meta module) of meta-parameter/meta-module for MGRL or to learn some fundamental concepts in inverse-design, the gradient estimator in untruncated setting is unbiased. However, we argue that (1) For MAML-related problem, the variance of sampling correction term in K-sample inner-loop meta gradient estimator is large because it needs to sum up all k terms and that is why [36] proposes to use one coefficient to control. The EPG can achieve lower variance estimation and perform better empirically [29] (2) For meta gradient based inverse design with multi-lifetime, the few-step meta gradient estimation under truncated setting is biased.

However, in online meta gradient setting (MGRL) or online opponent modelling (LOLA) with single-lifetime, things are completely different thus a direct transform of K-sample inner-loop formulation from MAML to MGRL might not be that straightforward. There exists a large gap between the implementation of online meta gradient algorithm and the final objective (meta-module/hyperparameters schedule) we may wish. First, it's an online setting so the multiple lifetime setting where the algorithm can restart from the very beginning and reiterate the whole process is banned here. This makes the estimation of unbiased meta gradient impossible because the algorithm cannot access to the future dynamic for gradient estimation. The experiments with multi-lifetime training in [36] is in fact out of the scope of online meta gradient setting and are more like meta gradient based inverse design. Second, in implementation of MGRL they only maintain one running γ or intrinsic model rather than multiple meta modules as a real schedule needs. Also, recently there exist one work [4] discussing multi-step MGRL and use one fixed meta parameters rather than a schedule for multi-step inner-loop, which may show a different understanding about untruncated gradient. In all, we believe that what online meta algorithm/opponent shaping like MGRL or LOLA optimizes and what the best they can achieve in such online setting are still open questions and remain to be further explored. It is really hard to simply formulate the unbiased meta gradient since the gap between implementation and objective is still not clear.

Thus, in our paper, we still focuses on the previous work (MAML/MGRL and LOLA) objectives with EPG inner-loop setting and use its meta gradient as our target gradient. All bias term we discuss is the bias w.r.t. the expected meta gradient in this EPG inner-loop and truncated setting. That is our work's limitation and we leave more things for future work: (1)The gap between EPG inner-loop meta gradient and K-sample inner-loop meta gradient in MAML-RL related problem. (2) The gap between truncated EPG inner-loop meta gradient and what the best gradient estimation we can get in online meta gradient/opponent shaping. (3) The gap between truncated EPG inner-loop meta gradient and the untruncated gradient in meta gradient based inverse design.

C Brief summary on biased Hessian estimation in MAML-RL

We will briefly introduce the reasons of biased Hessian estimation with automatic differentiation in one-step MAML-RL. Firstly, we can derive the analytic form of $\boldsymbol{\theta}^1$ and $\nabla_{\boldsymbol{\theta}^0}\boldsymbol{\theta}^1$

$$\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0 + \alpha \mathbb{E}_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau};\boldsymbol{\theta}^0)}[\nabla_{\boldsymbol{\theta}^0} \log \pi(\boldsymbol{\tau})\mathcal{R}(\boldsymbol{\tau})] \tag{18}$$

$$\nabla_{\theta^0}\theta^1 = I + \mathbb{E}_{\tau\sim p(\tau;\theta^0)}\left[\mathcal{R}(\tau)\left(\nabla^2_{\theta^0}\log\pi_{\theta^0}(\tau) + \nabla_{\theta^0}\log\pi_{\theta^0}(\tau)\nabla_{\theta^0}\log\pi_{\theta^0}(\tau)^\top\right)\right] \tag{19}$$

Typically we need to use trajectory samples τ_n to estimate the policy gradient, we can get the adapted policy estimate.

$$\hat{\theta}^1 = \theta^0 + \alpha\frac{1}{N}\sum_{\tau_n}\sum_{t=0}^{H-1}\nabla_\theta\log\pi_\theta(a_t^n \mid s_t^n)\left(\sum_{t'=0}^{H}\gamma^t r\left(s_{t'}^n, a_{t'}^n\right)\right) \tag{20}$$

Finally, implementation of MAML-RL derives the gradient estimate by automatic differentation. The corresponding estimation is biased:

$$\begin{aligned}\mathbb{E}[\nabla_{\theta^0}\hat{\theta}^1] &= I + \alpha\mathbb{E}_{\tau\sim p(\tau;\theta^0)}\left[\frac{1}{N}\sum_{\tau_n}\sum_{t=0}^{H-1}\nabla^2_{\theta^0}\log\pi_{\theta^0}(a_t^n \mid s_t^n)\left(\sum_{t'=0}^{H}\gamma^t r(s_{t'}^n, a_{t'}^n)\right)\right] \\ &= I + \alpha\mathbb{E}_{\tau\sim p(\tau;\theta^0)}\left[\mathcal{R}(\tau)\nabla^2_\theta\log\pi_{\theta^0}(\tau)\right] \neq \nabla_{\theta^0}\theta^1\end{aligned} \tag{21}$$

The main reason of biased Hessian estimation is that automatic differentiation tools only consider the dependency of θ in $\nabla_\theta\log\pi_\theta$ while ignoring the dependency in expectation $\mathbb{E}_{\tau\sim p(\tau;\theta^0)}$. In practice, the $\mathbb{E}_{\tau\sim p(\tau;\theta^0)}$ is represented by trajectory sampling so the gradient term $\nabla_\theta\mathbb{E}_{\tau\sim p(\tau;\theta^0)}$ is 0 using automatic differentiation. We need to add additional terms to further derive the gradient $\nabla_\theta\mathbb{E}_{\tau\sim p(\tau;\theta^0)}$ brought by sampling dependency.

D Limitations on Assumptions

Assumption 4.1-4.3 are standard assumptions used in various theoretical MAML-RL papers [6, 7, 18]. The Lipschitz continuity assumptions in Assumption 4.1 make sure we can work with nonconvex inner and outer objectives. The unbiased first-order gradient estimators assumptions in Assumption 4.2 can highlight our findings on two source of biases, which is also a plausible assumption in GMRL settings. As typically adopted in the analysis for stochastic optimization, we make the bounded-variance assumption in Assumption 4.3. Assumption 4.1-4.3 can be conveniently verified for e.g., inner-loop RL optimization in tabular MDP settings (finite state space and action space) with soft-max parameterisation of the policy, where $\pi_\theta(a \mid s) \propto \exp(\theta(s,a))$ with parameter $\theta = \theta(s,a)$. But in large-scale RL settings like atari games, Assumption 4.1-4.3 will not hold anymore.

E Proof of Proposition in Section 3

In this section, we provide the proof for Proposition 3.1 in Section 3.

E.1 Proof of Proposition 3.1

Proposition 3.1 (K-step Meta-Gradient)**.** *The exact meta-gradient to the objective in Eq. (2) can be written as:*

$$\begin{aligned}\nabla_\phi J^K(\phi) &= \nabla_\phi J^{Out}(\phi,\theta^K) + \alpha\nabla_\phi\theta^K\nabla_{\theta^K}J^{Out}(\phi,\theta^K), \\ \nabla_\phi\theta^K &= \sum_{i=0}^{K-1}\nabla_\phi\nabla_{\theta^i}J^{In}(\phi,\theta^i)\prod_{j=i+1}^{K-1}\left(I + \alpha\nabla^2_{\theta^j}J^{In}(\phi,\theta^j)\right).\end{aligned} \tag{3}$$

Proof. According to post-update inner parameters $\theta^K = \theta^0 + \alpha\sum_{i=0}^{K-1}\nabla_{\theta^i}J^{\text{In}}(\phi,\theta^i)$ and the fact that $\nabla_{\theta^i}J^{\text{In}}(\phi,\theta^i)$ is differentiable w.r.t. ϕ, we can treat θ^K as a differentiable function w.r.t. ϕ. Based on the chain rule, we can get

$$\nabla_\phi J^K(\phi) = \nabla_\phi J^{\text{Out}}(\phi,\theta^K) + \nabla_\phi\theta^K\nabla_{\theta^K}J^{\text{Out}}(\phi,\theta^K) \tag{22}$$

Based on the iterative updates that $\theta^{i+1} = \theta^i + \alpha\nabla_{\theta^i}J^{In}(\phi,\theta^i)$, for $i = 0,\ldots,K-1$ and similarly treat θ^i as a differentiable function w.r.t. ϕ, we have

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^{i+1} =& \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^{i} + \alpha \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) + \alpha \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^{i} \nabla^2_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \\
=& \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^{i} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \right) + \alpha \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i)
\end{aligned} \tag{23}
$$

Telescoping the above equality over i from 0 to $K-1$, we can get

$$
\begin{aligned}
\nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^{K} =& \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}^{0} \prod_{i=0}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \right) + \alpha \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^j) \right) \\
=& \alpha \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^j) \right)
\end{aligned} \tag{24}
$$

Combining Eq. (22) and Eq. (24) finishes the proof of Proposition 3.1. □

F Proof of Lemma in Section 4

F.1 Proof of Lemma 4.4

Lemma 4.4 (Compositional Bias)**.** *Suppose that Assumption 4.1 and 4.2 hold, let* $\hat{\Delta}_C = \mathbb{E}[\|f(\hat{\boldsymbol{\theta}}^K) - f(\boldsymbol{\theta}^K)\|]$ *be the compositional bias and* C_0 *the Lipschitz constant of* $f(\cdot)$*,* $|\boldsymbol{\tau}|$ *denote number of trajectories used to estimate inner-loop gradient in each inner-loop update step,* α *the learning rate, then we have,*

$$
\hat{\Delta}_C \leq C_0 \mathbb{E}[\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\|] \leq C_0 \left((1 + \alpha c_2)^K - 1 \right) \frac{\hat{\sigma}_{In}}{c_2 \sqrt{|\boldsymbol{\tau}|}}, \tag{6}
$$

where $\hat{\sigma}_{In} = \max_i \sqrt{\mathbb{V}[\nabla_{\boldsymbol{\theta}^i} \hat{J}^{In}(\boldsymbol{\phi}, \boldsymbol{\theta}^i, \boldsymbol{\tau}_0^i)]}$*,* $i \in \{0, .., K-1\}$*.*

Proof. In expected policy gradient inner-loop update setting, the iterative updates takes the form

$$
\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i + \alpha \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i), \ i = 0, \ldots, K-1 \tag{25}
$$

In Eq. (4), $\boldsymbol{\theta}^{i+1}$ are estimated using samples $\boldsymbol{\tau}_0^{0:i}$, then we have

$$
\hat{\boldsymbol{\theta}}^{i+1} = \hat{\boldsymbol{\theta}}^i + \alpha \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_0^i), \ \hat{\boldsymbol{\theta}}^0 = \boldsymbol{\theta}^0, \ i = 0, \ldots, K-1 \tag{26}
$$

According to the assumption that non-linear compositional vector-valued $f(\cdot)$ is Lipschitz continuous with constant C_0, we can get

$$
\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}[\|f(\hat{\boldsymbol{\theta}}^K) - f(\boldsymbol{\theta}^K)\|] \leq C_0 \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}} \left[\left\| \hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K \right\| \right] \tag{27}
$$

Based on Eq. (25) and Eq. (26), we can get

$$
\begin{aligned}
&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K-\boldsymbol{\theta}^K\right\|\right]\\
=&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^{K-1}-\boldsymbol{\theta}^{K-1}+\alpha\nabla_{\boldsymbol{\theta}^{K-1}}J^{\mathrm{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^{K-1})-\alpha\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})\right\|\right]\\
\overset{(i)}{\leq}&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^{K-1}-\boldsymbol{\theta}^{K-1}\right\|\right]+\alpha\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\nabla_{\boldsymbol{\theta}^{K-1}}J^{\mathrm{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^{K-1})-\mathbb{E}_{\tau_0^{K-1}}[\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})]\right\|\right]+\\
&\alpha\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\mathbb{E}_{\tau_0^{K-1}}[\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})]-\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})\right\|\right]\\
\leq&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^{K-1}-\boldsymbol{\theta}^{K-1}\right\|\right]+\alpha c_2\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^{K-1}-\boldsymbol{\theta}^{K-1}\right\|\right]+\\
&\alpha\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\mathbb{E}_{\tau_0^{K-1}}[\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})]-\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})\right\|\right]\\
\leq&(1+\alpha c_2)\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^{K-1}-\boldsymbol{\theta}^{K-1}\right\|\right]+\\
&\alpha\mathbb{E}_{\tau_0^{0:K-2}}\left[\mathbb{E}_{\tau_0^{K-1}}\left[\left\|\mathbb{E}_{\tau_0^{K-1}}[\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})]-\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1})\right\|\mid\boldsymbol{\tau}_0^{0:K-2}\right]\right]\\
\leq&(1+\alpha c_2)\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^{K-1}-\boldsymbol{\theta}^{K-1}\right\|\right]+\alpha\mathbb{E}_{\tau_0^{0:K-2}}\left[\sqrt{\frac{\mathbb{V}\left[\nabla_{\hat{\boldsymbol{\theta}}^{K-1}}\hat{J}^{\mathrm{In}}\left(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^{K-1},\boldsymbol{\tau}_0^{K-1}\right)\mid\boldsymbol{\tau}_0^{0:K-2}\right]}{|\boldsymbol{\tau}_0^{K-1}|}}\right]
\end{aligned}
\tag{28}
$$

where (i) follows from Lemma H.3 and Assumption 4.2.

Let $\hat{\sigma}_{\mathrm{In}}=\max_i\sqrt{\mathbb{V}[\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\tau_0^i)]}$, $|\boldsymbol{\tau}|=|\tau_0^i|$, $i\in\{0,\ldots,K-1\}$

$$
\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K-\boldsymbol{\theta}^K\right\|\right]\leq(1+\alpha c_2)\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^{K-1}-\boldsymbol{\theta}^{K-1}\right\|\right]+\alpha\frac{\hat{\sigma}_{\mathrm{In}}}{\sqrt{|\boldsymbol{\tau}|}}
\tag{29}
$$

Iteratively, we can get

$$
\begin{aligned}
\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K-\boldsymbol{\theta}^K\right\|\right]&\leq\left(1+\ldots+(1+\alpha c_2)^{K-1}\right)\alpha\frac{\hat{\sigma}_{\mathrm{In}}}{\sqrt{|\boldsymbol{\tau}|}}\\
&=\left((1+\alpha c_2)^K-1\right)\frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}}
\end{aligned}
\tag{30}
$$

which concludes the proof of Lemma 4.4. □

G Proof of Theorem in Section 4

G.1 Proof of Theorem 4.5

Theorem 4.5 (Upper bound for the bias and the variance)**.** *Suppose that Assumption 4.1 and 4.2 and 4.3 hold. Let $J_{\boldsymbol{\phi},\boldsymbol{\theta}}$ denote $\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}}J^{In}$, $H_{\boldsymbol{\theta},\boldsymbol{\theta}}$ denote $\nabla^2_{\boldsymbol{\theta}}J^{In}$, $\hat{\Delta}^K=\|\mathbb{E}[\nabla_{\boldsymbol{\phi}}\hat{J}^K(\boldsymbol{\phi})]-\nabla_{\boldsymbol{\phi}}J^K(\boldsymbol{\phi})\|$ be the meta-gradient estimation bias, set $B=1+\alpha c_2$. Then the bound of bias hold:*

$$
\hat{\Delta}^K\leq O\left((B+\hat{\Delta}_H)^{K-1}\left(\mathbb{E}[\|\hat{\boldsymbol{\theta}}^K-\boldsymbol{\theta}^K\|]+\hat{\Delta}_J+(K-1)\right)\right).
\tag{9}
$$

Let $(\hat{\sigma}^K)^2 = \mathbb{V}\left[\nabla_{\boldsymbol{\phi}} \hat{J}^K(\boldsymbol{\phi})\right]$ *be the meta-gradient estimation variance, set* $V_1 = (1+\alpha c_2)^2$, $V_2 = 2\alpha^2(m_1^2 + 3\sigma_2^2)$ *the estimation variance is given by*

$$
\begin{aligned}
(\hat{\sigma}^K)^2 \leq O\Bigg(& (V_1 + \hat{\Delta}_H^2)^{K-1}\left(\mathbb{E}[\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\|^2] + (K-1)\right) \\
& + \left(V_2 + (V_1 + \hat{\Delta}_H^2 + \hat{\sigma}_H^2)^{K-1} - (V_1 + \hat{\Delta}_H^2)^{K-1}\right)(\hat{\Delta}_J^2 + \hat{\sigma}_J^2)\Bigg).
\end{aligned} \tag{10}
$$

where $\hat{\Delta}_J = \max_{\boldsymbol{\phi}\times\boldsymbol{\theta}} \|\mathbb{E}[\hat{J}_{\boldsymbol{\phi},\boldsymbol{\theta}}] - J_{\boldsymbol{\phi},\boldsymbol{\theta}}\|$. $\hat{\Delta}_H = \max_{\boldsymbol{\theta}} \|\mathbb{E}[\hat{H}_{\boldsymbol{\theta},\boldsymbol{\theta}}] - H_{\boldsymbol{\theta},\boldsymbol{\theta}}\|$. $(\hat{\sigma}_J)^2 = \frac{\max_{\boldsymbol{\phi}\times\boldsymbol{\theta}} \mathbb{V}[\hat{J}_{\boldsymbol{\phi},\boldsymbol{\theta}}]}{|\tau|}$. $(\hat{\sigma}_H)^2 = \frac{\max_{\boldsymbol{\theta}} \mathbb{V}[\hat{H}_{\boldsymbol{\theta},\boldsymbol{\theta}}]}{|\tau|}$.

Proof. According to Proposition 3.1, exact meta-gradient $\nabla_{\boldsymbol{\phi}} J^{\mathrm{K}}(\boldsymbol{\phi})$ takes the form

$$
\nabla_{\boldsymbol{\phi}} J^{\mathrm{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K) + \alpha \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^j)\right) \nabla_{\boldsymbol{\theta}^K} J^{\mathrm{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K) \tag{31}
$$

where

$$
\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i + \alpha \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i),\ i = 0, \ldots, K-1 \tag{32}
$$

Acoordingly, in Eq. (4), K-step meta-gradient estimator $\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{K}}(\boldsymbol{\phi})$ takes the form

$$
\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \boldsymbol{\tau}_3) + \alpha \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_1^i) \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j)\right) \nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3) \tag{33}
$$

where

$$
\hat{\boldsymbol{\theta}}^{i+1} = \hat{\boldsymbol{\theta}}^i + \alpha \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_0^i),\ \hat{\boldsymbol{\theta}}^0 = \boldsymbol{\theta}^0,\ i = 0, \ldots, K-1 \tag{34}
$$

Hence the expectation of meta-gradient estimator takes the form

$$
\begin{aligned}
& \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}, \boldsymbol{\tau}_1^{0:K-1}, \boldsymbol{\tau}_2^{1:K-1}, \boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{K}}(\boldsymbol{\phi})] \\
=& \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\Bigg[\mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \boldsymbol{\tau}_3) \mid \boldsymbol{\tau}_0^{0:K-1}] + \alpha \sum_{i=0}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_1^i) \mid \boldsymbol{\tau}_0^{0:i-1}] \times \\
& \prod_{j=i+1}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \mid \boldsymbol{\tau}_0^{0:j-1}] \times \mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3) \mid \boldsymbol{\tau}_0^{0:K-1}]\Bigg]
\end{aligned} \tag{35}
$$

we can then derive meta-gradient bias in K-step expected policy gradient setting,

$$
\begin{aligned}
& \left\|\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}, \boldsymbol{\tau}_1^{0:K-1}, \boldsymbol{\tau}_2^{0:K-1}, \boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{K}}(\boldsymbol{\phi})] - \nabla_{\boldsymbol{\phi}} J^{\mathrm{K}}(\boldsymbol{\phi})\right\| \\
\leq& \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\Bigg[\Bigg\|\mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \boldsymbol{\tau}_3) \mid \boldsymbol{\tau}_0^{0:K-1}] + \alpha \sum_{i=0}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_1^i) \mid \boldsymbol{\tau}_0^{0:i-1}] \times \\
& \prod_{j=i+1}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \mid \boldsymbol{\tau}_0^{0:i-1}] \times \mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3) \mid \boldsymbol{\tau}_0^{0:K-1}] - \\
& \nabla_{\boldsymbol{\phi}} J^{\mathrm{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K) - \alpha \sum_{i=0}^{K-1} \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i) \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^j)\right) \nabla_{\boldsymbol{\theta}^K} J^{\mathrm{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)\Bigg\|\Bigg]
\end{aligned} \tag{36}
$$

$$
\begin{aligned}
&\leq \mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\mathbb{E}_{\tau_3}[\nabla_{\boldsymbol{\phi}}\hat{J}^{\text{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)\mid \boldsymbol{\tau}_0^{0:K-1}]-\nabla_{\boldsymbol{\phi}}J^{\text{Out}}(\boldsymbol{\phi},\boldsymbol{\theta}^K)\right\|\right]\\
&\quad+\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\alpha\sum_{i=0}^{K-1}\mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\mid \boldsymbol{\tau}_0^{0:i-1}]\times\prod_{j=i+1}^{K-1}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid \boldsymbol{\tau}_0^{0:i-1}]\right.\\
&\quad\times\mathbb{E}_{\tau_3}[\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\text{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)\mid \boldsymbol{\tau}_0^{0:K-1}]\\
&\quad\left.-\alpha\sum_{i=0}^{K-1}\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^i}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^i)\prod_{j=i+1}^{K-1}\left(I+\alpha\nabla^2_{\boldsymbol{\theta}^j}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^j)\right)\nabla_{\boldsymbol{\theta}^K}J^{\text{Out}}(\boldsymbol{\phi},\boldsymbol{\theta}^K)\right\|\right]\\
&\overset{(i)}{\leq}\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\nabla_{\boldsymbol{\phi}}J^{\text{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K)-\nabla_{\boldsymbol{\phi}}J^{\text{Out}}(\boldsymbol{\phi},\boldsymbol{\theta}^K)\right\|\mid \boldsymbol{\tau}_0^{0:K-1}\right]\\
&\quad+\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\alpha\sum_{i=0}^{K-1}\mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\mid \boldsymbol{\tau}_0^{0:i-1}]\times\prod_{j=i+1}^{K-1}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid \boldsymbol{\tau}_0^{0:i-1}]\right.\\
&\quad\times\mathbb{E}_{\tau_3}[\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\text{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)\mid \boldsymbol{\tau}_0^{0:K-1}]-\\
&\quad\left.\alpha\sum_{i=0}^{K-1}\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^i}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^i)\prod_{j=i+1}^{K-1}\left(I+\alpha\nabla^2_{\boldsymbol{\theta}^j}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^j)\right)\nabla_{\boldsymbol{\theta}^K}J^{\text{Out}}(\boldsymbol{\phi},\boldsymbol{\theta}^K)\right\|\right]
\end{aligned}
\tag{37}
$$

where (i) follows from Assumption 4.2.

$$
\begin{aligned}
&\overset{(ii)}{\leq}\mu_1\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K-\boldsymbol{\theta}^K\right\|\mid \boldsymbol{\tau}_0^{0:K-1}\right]\\
&\quad+\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\alpha\sum_{i=0}^{K-1}\mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\mid \boldsymbol{\tau}_0^{0:i-1}]\times\prod_{j=i+1}^{K-1}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid \boldsymbol{\tau}_0^{0:i-1}]\right.\\
&\quad\left.\times\mathbb{E}_{\tau_3}[\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\text{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)\mid \boldsymbol{\tau}_0^{0:K-1}]-\alpha\sum_{i=0}^{K-1}\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^i}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^i)\prod_{j=i+1}^{K-1}\left(I+\alpha\nabla^2_{\boldsymbol{\theta}^j}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^j)\right)\nabla_{\boldsymbol{\theta}^K}J^{\text{Out}}(\boldsymbol{\phi},\boldsymbol{\theta}^K)\right\|\right]
\end{aligned}
\tag{38}
$$

where (ii) follows from Assumption 4.1 on Lipschitz Continuity of $\nabla_{\boldsymbol{\phi}}J^{\text{Out}}$

$$
\begin{aligned}
&\overset{(iii)}{\leq}\mu_1\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K-\boldsymbol{\theta}^K\right\|\mid \boldsymbol{\tau}_0^{0:K-1}\right]\\
&\quad+\alpha\sum_{i=0}^{K-1}\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\mid \boldsymbol{\tau}_0^{0:i-1}]\times\right.\\
&\quad\prod_{j=i+1}^{K-1}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\text{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid \boldsymbol{\tau}_0^{0:i-1}]\times\mathbb{E}_{\tau_3}[\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\text{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)\mid \boldsymbol{\tau}_0^{0:K-1}]-\\
&\quad\left.\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^i}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^i)\prod_{j=i+1}^{K-1}\left(I+\alpha\nabla^2_{\boldsymbol{\theta}^j}J^{\text{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^j)\right)\nabla_{\boldsymbol{\theta}^K}J^{\text{Out}}(\boldsymbol{\phi},\boldsymbol{\theta}^K)\right\|\right]
\end{aligned}
\tag{39}
$$

where (iii) follows from Lemma H.3. Using the similar add-minus trick in the proof of Lemma 4.4, we can have

$$
\begin{aligned}
&\overset{(iv)}{\leq} \mu_1 \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\right\| \mid \boldsymbol{\tau}_0^{0:K-1}\right] \\
&\quad + \alpha \sum_{i=0}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\| \prod_{j=i+1}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \mid \boldsymbol{\tau}_0^{0:i-1}] \times \mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3) \mid \boldsymbol{\tau}_0^{0:K-1}] - \right.\right. \\
&\quad \left.\left. \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}\left(\boldsymbol{\phi}, \boldsymbol{\theta}^j\right)\right) \nabla_{\boldsymbol{\theta}^K} J^{\mathrm{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)\right\|\right] \times \|\nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i)\| + \\
&\quad \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\|\mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_1^i) \mid \boldsymbol{\tau}_0^{0:i-1}] - \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i)\right\|\right] \times \\
&\quad \left\|\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_2^j}[I - \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \mid \boldsymbol{\tau}_0^{0:i-1}] \times \mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3) \mid \boldsymbol{\tau}_0^{0:K-1}]\right]\right\|
\end{aligned}
\tag{40}
$$

Based on Assumption 4.1 and Assumption 4.2, we can change the expectation of unbiased first-order stochastic estimator to respective first-order gradient function, then we can replace it with Lipschitz constants.

$$
\begin{aligned}
&\leq \mu_1 \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\right\| \mid \boldsymbol{\tau}_0^{0:K-1}\right] \\
&\quad + \alpha \sum_{i=0}^{K-1} \left[\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\| \prod_{j=i+1}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \mid \boldsymbol{\tau}_0^{0:i-1}] - \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}\left(\boldsymbol{\phi}, \boldsymbol{\theta}^j\right)\right)\right\|\right] \times \right. \\
&\quad \left\|\nabla_{\boldsymbol{\theta}^K} J^{\mathrm{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)\right\| + \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\|\mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3) \mid \boldsymbol{\tau}_0^{0:K-1}] - \nabla_{\boldsymbol{\theta}^K} J^{\mathrm{Out}}(\boldsymbol{\phi}, \boldsymbol{\theta}^K)\right\|\right] \times \\
&\quad \left.\left\|\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \mid \boldsymbol{\tau}_0^{0:i-1}]\right]\right\|\right] \times \|\nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i)\| + \\
&\quad \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\|\mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \boldsymbol{\tau}_1^i) \mid \boldsymbol{\tau}_0^{0:i-1}] - \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i)\right\|\right] \times \\
&\quad \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\| \prod_{j=i+1}^{K-1} \mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j) \mid \boldsymbol{\tau}_0^{0:i-1}]\right\| \times \left\|\mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K} \hat{J}^{\mathrm{Out}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^K, \tau_3) \mid \boldsymbol{\tau}_0^{0:K-1}]\right\|\right]
\end{aligned}
\tag{41}
$$

$$
\begin{aligned}
&\leq \mu_1 \mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\right\|\right] \\
&+ \alpha \sum_{i=0}^{K-1} c_1 \Bigg[m_2 \mathbb{E}_{\tau_0^{0:K-1}}\Bigg[\Bigg\| \prod_{j=i+1}^{K-1} \mathbb{E}_{\tau_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \tau_2^j) \mid \tau_0^{0:i-1}] - \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}\left(\boldsymbol{\phi}, \boldsymbol{\theta}^j\right)\right) \Bigg\|\Bigg] \\
&+ \mu_2 \mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\right\|\right] \times \left\| \mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1} \mathbb{E}_{\tau_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \tau_2^j) \mid \tau_0^{0:i-1}]\right]\right\| \Bigg] + \\
&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\| \mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \tau_1^i) \mid \tau_0^{0:i-1}] - \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i)\right\|\right] \times \\
&m_2 \left\| \mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1} \mathbb{E}_{\tau_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \tau_2^j) \mid \tau_0^{0:i-1}]\right]\right\|
\end{aligned}
\tag{42}
$$

$$
\begin{aligned}
&\leq \mu_1 \mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\right\|\right] \\
&+ \alpha \sum_{i=0}^{K-1} c_1 m_2 \underbrace{\mathbb{E}_{\tau_0^{0:K-1}}\Bigg[\Bigg\| \prod_{j=i+1}^{K-1} \mathbb{E}_{\tau_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \tau_2^j) \mid \tau_0^{0:i-1}] - \prod_{j=i+1}^{K-1} \left(I + \alpha \nabla^2_{\boldsymbol{\theta}^j} J^{\mathrm{In}}\left(\boldsymbol{\phi}, \boldsymbol{\theta}^j\right)\right) \Bigg\|\Bigg]}_{\textbf{Term (i)}} \\
&+ \Bigg[c_1 \mu_2 \mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\right\|\right] + m_2 \underbrace{\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\| \mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}} \nabla_{\hat{\boldsymbol{\theta}}^i} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^i, \tau_1^i) \mid \tau_0^{0:i-1}] - \nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}^i} J^{\mathrm{In}}(\boldsymbol{\phi}, \boldsymbol{\theta}^i)\right\|\right]}_{\textbf{Term (ii)}} \Bigg] \times \\
&\underbrace{\left\| \mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1} \mathbb{E}_{\tau_2^j}[I + \alpha \nabla^2_{\hat{\boldsymbol{\theta}}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \tau_2^j) \mid \tau_0^{0:i-1}]\right]\right\|}_{\textbf{Term (iii)}}
\end{aligned}
\tag{43}
$$

Let $J_{\boldsymbol{\phi},\boldsymbol{\theta}}$ denote $\nabla_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\theta}} J^{\mathrm{In}}$, $H_{\boldsymbol{\theta},\boldsymbol{\theta}}$ denote $\nabla^2_{\boldsymbol{\theta}} J^{\mathrm{In}}$,$\hat{\Delta}_J = \max \|\mathbb{E}[\hat{J}_{\boldsymbol{\phi},\boldsymbol{\theta}}] - J_{\boldsymbol{\phi},\boldsymbol{\theta}}\|$. $\hat{\Delta}_H = \max \|\mathbb{E}[\hat{H}_{\boldsymbol{\theta},\boldsymbol{\theta}}] - H_{\boldsymbol{\theta},\boldsymbol{\theta}}\|$. $(\hat{\sigma}_J)^2 = \frac{\max \mathbb{V}[\hat{J}_{\boldsymbol{\phi},\boldsymbol{\theta}}]}{|\tau|}$. $(\hat{\sigma}_H)^2 = \frac{\max \mathbb{V}[\hat{H}_{\boldsymbol{\theta},\boldsymbol{\theta}}]}{|\tau|}$. We upper bound terms (i)-(ii) in Eq. (43) respectively, that is,

Term (i). According to

$$
\begin{aligned}
&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\prod_{j=i+1}^{K-1}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\theta}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid\boldsymbol{\tau}_0^{0:i-1}]-\prod_{j=i+1}^{K-1}\left(I+\alpha\nabla^2_{\theta^j}J^{\mathrm{In}}\left(\boldsymbol{\phi},\boldsymbol{\theta}^j\right)\right)\right\|\right]\\
\leq&\mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-2}(I+\alpha c_2+\alpha\hat{\Delta}_H)\right]\left(\alpha\hat{\Delta}_H+\alpha\rho_2\left((1+\alpha c_2)^{K-1}-1\right)\frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}}\right)\\
&+(1+\alpha c_2)\times\\
&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\prod_{j=i+1}^{K-2}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\theta}^j}\hat{J}^{In}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid\boldsymbol{\tau}_0^{0:i-1}]-\prod_{j=i+1}^{K-2}\left(I+\alpha\nabla^2_{\theta^j}J^{In}\left(\boldsymbol{\phi},\boldsymbol{\theta}^j\right)\right)\right\|\right]\\
\leq&\alpha\left[(1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-i-1}-(1+\alpha c_2)^{K-i-1}\right]\\
&+\frac{\rho_2}{c_2}\left((1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-i-1}-1\right)(1+\alpha c_2)^{K-1}\frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}}
\end{aligned}
\tag{44}
$$

Term (ii).

$$
\begin{aligned}
&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\theta}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\mid\boldsymbol{\tau}_0^{0:i-1}]-\nabla_{\boldsymbol{\phi}}\nabla_{\theta^i}J^{\mathrm{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^i)\right\|\right]\\
\leq&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\theta}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\mid\boldsymbol{\tau}_0^{0:i-1}]-\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\theta}^i}J^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i)\right\|\right]+\\
&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\theta}^i}J^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i)-\nabla_{\boldsymbol{\phi}}\nabla_{\theta^i}J^{\mathrm{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^i)\mid\boldsymbol{\tau}_0^{0:i-1}]\right\|\right]
\end{aligned}
\tag{45}
$$

$$
\begin{aligned}
&\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\theta}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\mid\boldsymbol{\tau}_0^{0:i-1}]-\nabla_{\boldsymbol{\phi}}\nabla_{\theta^i}J^{\mathrm{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^i)\right\|\right]\\
\leq&\mathbb{E}_{\tau_0^{0:K-1}}\left[\hat{\Delta}_J\right]+\lambda_2\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^i-\boldsymbol{\theta}^i\right\|\right]\\
\leq&\hat{\Delta}_J+\lambda_2\mathbb{E}_{\tau_0^{0:K-1}}\left[\left\|\hat{\boldsymbol{\theta}}^i-\boldsymbol{\theta}^i\right\|\right]
\end{aligned}
\tag{46}
$$

Term (iii).

$$
\begin{aligned}
&\left\|\mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\theta}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid\boldsymbol{\tau}_0^{0:i-1}]\right]\right\|\\
\leq&\mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1}\left(I+\alpha\left\|\mathbb{E}_{\tau_2^j}[\nabla^2_{\hat{\theta}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid\boldsymbol{\tau}_0^{0:i-1}]-\nabla^2_{\hat{\theta}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j)\right\|+\alpha\left\|\nabla^2_{\hat{\theta}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j)\right\|\right)\right]
\end{aligned}
\tag{47}
$$

$$
\begin{aligned}
&\left\|\mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1}\mathbb{E}_{\tau_2^j}[I+\alpha\nabla^2_{\hat{\theta}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\mid\boldsymbol{\tau}_0^{0:i-1}]\right]\right\|\\
\leq&\mathbb{E}_{\tau_0^{0:K-1}}\left[\prod_{j=i+1}^{K-1}(1+\alpha c_2+\alpha\hat{\Delta}_H)\mid\boldsymbol{\tau}_0^{0:i-1}]\right]\\
\leq&(1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-i-1}
\end{aligned}
\tag{48}
$$

Then combine terms (i)-(iii) together, that is

$$
\begin{aligned}
&\left\|\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1},\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{0:K-1},\boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})]-\nabla_{\boldsymbol{\phi}}J^{\mathrm{K}}(\boldsymbol{\phi})\right\| \\
\leq&\mu_1\left((1+\alpha c_2)^K-1\right)\frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}} \\
&+\alpha\sum_{i=0}^{K-1}\alpha c_1 m_2\left[(1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-i-1}-(1+\alpha c_2)^{K-i-1}\right] \\
&+c_1 m_2\frac{\rho_2}{c_2}\left((1+\alpha c_2+\hat{\Delta}_H)^{K-i-1}-1\right)(1+\alpha c_2)^{K-1}\frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}} \\
&+(1+\alpha c_2+\hat{\Delta}_H)^{K-i-1}\left[c_1\mu_2\left((1+\alpha c_2)^K-1\right)\frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}}+m_2\hat{\Delta}_J+m_2\lambda_2\left((1+\alpha c_2)^i-1\right)\frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}}\right] \\
\leq&(\mu_1+\alpha(c_1 m_2\frac{\rho_2}{c_2}+c_1\mu_2+m_2\lambda_2))\left(((1+\alpha c_2)^K-1\right)\left((1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-1}-1\right)\frac{\hat{\sigma}_{In}}{c_2\sqrt{|\boldsymbol{\tau}|}} \\
&+(\alpha m_2)\left((1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-1}-1\right)\hat{\Delta}_J \\
&+(\alpha^2 c_1 m_2)\left((1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-1}-(1+\alpha c_2)^{K-1}\right)
\end{aligned}
\tag{49}
$$

$$
\begin{aligned}
&\left\|\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1},\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{0:K-1},\boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})]-\nabla_{\boldsymbol{\phi}}J^{\mathrm{K}}(\boldsymbol{\phi})\right\| \\
\leq&\mathcal{O}\left((1+\alpha c_2+\alpha\hat{\Delta}_H)^{K-1}\left(\mathbb{E}[\|\hat{\boldsymbol{\theta}}^K-\boldsymbol{\theta}^K\|]+\hat{\Delta}_J+(K-1)\right)\right)
\end{aligned}
\tag{50}
$$

which concludes the proof of upper bound of meta-gradient bias.

According to Lemma H.7,

$$
\mathbb{V}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\right]=\underbrace{\mathbb{V}\left[\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\mid\boldsymbol{\tau}_0^{0:K-1}\right]\right]}_{\textbf{Term (i)}}+\underbrace{\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\mathbb{V}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\mid\boldsymbol{\tau}_0^{0:K-1}\right]\right]}_{\textbf{Term (ii)}}
\tag{51}
$$

We upper bound terms (i)-(ii) in Eq. (51) respectively, that is,

Term (i).

$$
\begin{aligned}
&\mathbb{V}\left[\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\mid\boldsymbol{\tau}_0^{0:K-1}\right]\right] \\
=&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\|\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\mid\boldsymbol{\tau}_0^{0:K-1}\right]-\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1},\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\right]\right\|^2\right] \\
\leq&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\left\|\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\mid\boldsymbol{\tau}_0^{0:K-1}\right]-\nabla_{\boldsymbol{\phi}}J^{\mathrm{K}}(\boldsymbol{\phi})\right\|^2\right]
\end{aligned}
\tag{52}
$$

According to proof of upper bound of bias term, together with Lemma H.5 (ii).

$$
\begin{aligned}
&\left\| \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1},\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{0:K-1},\boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{K}}(\boldsymbol{\phi})] - \nabla_{\boldsymbol{\phi}} J^{\mathrm{K}}(\boldsymbol{\phi}) \right\| \\
\leq & \mu_1 \left((1+\alpha c_2)^K - 1 \right) \frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}} \\
&+ \alpha \sum_{i=0}^{K-1} \alpha c_1 m_2 \left[(1+\alpha c_2 + \hat{\Delta}_H)^{K-i-1} - (1+\alpha c_2)^{K-i-1} \right] \\
&+ c_1 m_2 \frac{\rho_2}{c_2} \left((1+\alpha c_2 + \hat{\Delta}_H)^{K-i-1} - 1 \right) (1+\alpha c_2)^{K-1} \frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}} \\
&+ (1+\alpha c_2 + \hat{\Delta}_H)^{K-i-1} \left[c_1 \mu_2 \left((1+\alpha c_2)^K - 1 \right) \frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}} + m_2 \hat{\Delta}_J + m_2 \lambda_2 \left((1+\alpha c_2)^i - 1 \right) \frac{\hat{\sigma}_{\mathrm{In}}}{c_2\sqrt{|\boldsymbol{\tau}|}} \right]
\end{aligned} \tag{53}
$$

$$
\begin{aligned}
&\mathbb{V}\left[\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3} \left[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{K}}(\boldsymbol{\phi}) \mid \boldsymbol{\tau}_0^{0:K-1} \right] \right] \\
\leq & 4^K \left(2\mu_1^2 + 2K\alpha^2 (c_1 m_2 \frac{\rho_2}{c_2} + c_1\mu_2 + m_2\lambda_2)^2 \right) \times \\
&\left(\left((1+\alpha c_2)^2 + \alpha^2 c_2^2 \right)^K - 1 \right) \left(\left((1+\alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-1} - 1 \right) \frac{(\hat{\sigma}_{\mathrm{In}})^2}{c_2^2 |\boldsymbol{\tau}|} \\
&+ (2K\alpha^2 m_2^2) \left(\left((1+\alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-1} - 1 \right) (\hat{\Delta}_J)^2 \\
&+ (2K\alpha^4 c_1^2 m_2^2) \left(\left((1+\alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-1} - \left((1+\alpha c_2)^2 + \alpha^2 c_2^2 \right)^{K-1} \right)
\end{aligned} \tag{54}
$$

Term (ii).

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\mathbb{V}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi}) \mid \boldsymbol{\tau}_0^{0:K-1}\right]\right] \\
=&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi}) - \mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{K}}(\boldsymbol{\phi})\right]\right\|^2 \mid \boldsymbol{\tau}_0^{0:K-1}\right]\right] \\
\leq&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[2\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\boldsymbol{\tau}_3) - \mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\boldsymbol{\phi}}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\boldsymbol{\tau}_3)]\right\|^2\right]+\right. \\
&2\alpha^2\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\sum_{i=0}^{K-1}\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\prod_{j=i+1}^{K-1}\left(I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\right)\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)\right. \\
&\left.\left.-\sum_{i=0}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)]\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)]\right\|^2\right] \mid \boldsymbol{\tau}_0^{0:K-1}\right] \\
\leq&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[2(\sigma_1)^2+2K\alpha^2\sum_{i=0}^{K-1}\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\right\|^2\times\right. \\
&\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\prod_{j=i+1}^{K-1}\left(I-\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\right)\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)-\right.\right. \\
&\left.\left.\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)]\right\|^2\right]+ \\
&\left\|\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\mathbb{E}_{\boldsymbol{\tau}_3}[\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)]\right\|^2 \\
&\left.\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)-\mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)]\right\|^2\right] \mid \boldsymbol{\tau}_0^{0:K-1}\right]
\end{aligned}
\tag{55}
$$

$$
\begin{aligned}
\leq&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\left[2(\sigma_1)^2+2K\alpha^2\sum_{i=0}^{K-1}\left[\left\|\nabla_{\hat{\boldsymbol{\theta}}^K}\hat{J}^{\mathrm{Out}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^K,\tau_3)\right\|^2\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\right\|^2\times\right.\right. \\
&\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\prod_{j=i+1}^{K-1}\left(I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\right)-\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right\|^2\right]+ \\
&(\sigma_2)^2\left\|\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right\|^2\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\right\|^2\right]+ \\
&m_2^2\left\|\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I+\alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right\|^2\times \\
&\left.\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)-\mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)]\right\|^2\right] \mid \boldsymbol{\tau}_0^{0:K-1}\right]
\end{aligned}
\tag{56}
$$

$$
\begin{aligned}
&\leq \mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}}\Bigg[2(\sigma_1)^2 + 2K\alpha^2 \sum_{i=0}^{K-1}\Bigg[\left(2m_2^2 + 2(\sigma_2)^2\right) \times \left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\right\|^2 \times \\
&\underbrace{\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\prod_{j=i+1}^{K-1}\left(I + \alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\right) - \prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right\|^2\right]}_{\text{Part (I)}} + \\
&(\sigma_2)^2\left\|\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right\|^2 \times \underbrace{\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\right\|^2}_{\text{Part (II)}} + \\
&\underbrace{m_2^2\left\|\prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right\|^2}_{\text{Part (III)}} \times \\
&\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i) - \mathbb{E}_{\boldsymbol{\tau}_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)]\right\|^2\right] \mid \boldsymbol{\tau}_0^{0:K-1}\Bigg]
\end{aligned} \tag{57}
$$

Part (I) According to

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}_1^{0:K-1},\boldsymbol{\tau}_2^{1:K-1},\boldsymbol{\tau}_3}\left[\left\|\prod_{j=i+1}^{K-1}\left(I + \alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\right) - \prod_{j=i+1}^{K-1}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right\|^2\right] \\
&\leq 2\prod_{j=i+1}^{K-2}\left(3(1+\alpha c_2)^2 + 3\alpha^2(\hat{\Delta}_H)^2 + 3\alpha^2(\hat{\sigma}_H)^2\right) \times \alpha^2(\hat{\sigma}_H)^2 + \\
&4\left((1+\alpha c_2)^2 + \alpha^2(\hat{\Delta}_H)^2\right) \times \\
&\mathbb{E}_{\boldsymbol{\tau}_2^{1:K-1}}\left[\left\|\left(\prod_{j=i+1}^{K-2}\left(I + \alpha\nabla^2_{\hat{\boldsymbol{\theta}}^j}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)\right) - \prod_{j=i+1}^{K-2}\mathbb{E}_{\boldsymbol{\tau}_2^j}[I + \alpha\nabla^2_{\boldsymbol{\theta}}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^j,\boldsymbol{\tau}_2^j)]\right)\right\|^2\right] \\
&\leq 6^{K-i-1}\left[\left((1+\alpha c_2)^2 + \alpha^2(\hat{\Delta}_H)^2 + \alpha^2(\hat{\sigma}_H)^2\right)^{K-i-1} - \left((1+\alpha c_2)^2 + \alpha^2(\hat{\Delta}_H)^2\right)^{K-i-1}\right]
\end{aligned} \tag{58}
$$

Part (II) According to the supporting Lemma

$$
\begin{aligned}
&\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^0}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^0,\tau_1)\right\|^2 \\
&\leq \Bigg\|\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}J^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i) - \nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}J^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i) + \mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}\left(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i\right)] \\
&\quad - \mathbb{E}_{\tau_1^i}[\nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)] + \nabla_{\boldsymbol{\phi}}\nabla_{\hat{\boldsymbol{\theta}}^i}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\hat{\boldsymbol{\theta}}^i,\boldsymbol{\tau}_1^i)\Bigg\|^2 \\
&\leq 3c_1^2 + 3\left((\hat{\Delta}_J)^2 + (\hat{\sigma}_J)^2\right)
\end{aligned} \tag{59}
$$

$$
\left\|\nabla_{\boldsymbol{\phi}}\nabla_{\boldsymbol{\theta}^0}\hat{J}^{\mathrm{In}}(\boldsymbol{\phi},\boldsymbol{\theta}^0,\boldsymbol{\tau}_1^i)\right\|^2 \leq 3c_1^2 + 3\left((\hat{\Delta}_J)^2 + (\hat{\sigma}_J)^2\right) \tag{60}
$$

Part (III)

$$
\begin{aligned}
&\left\| \prod_{j=i+1}^{K-1} \mathbb{E}_{\tau_2^j}[I + \alpha \nabla^2_{\hat{\theta}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j)] \right\|^2 \\
&\leq \prod_{j=i+1}^{K-1} \left\| I - \alpha \nabla^2_{\hat{\theta}^j} J^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j) + \alpha \nabla^2_{\hat{\theta}^j} J^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j) - \alpha \mathbb{E}_{\tau_2^j}[\nabla^2_{\hat{\theta}^j} \hat{J}^{\mathrm{In}}(\boldsymbol{\phi}, \hat{\boldsymbol{\theta}}^j, \boldsymbol{\tau}_2^j)] \right\|^2 \\
&\leq \left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-i-1}
\end{aligned}
\tag{61}
$$

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}} \left[\mathbb{V}\left[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{K}}(\boldsymbol{\phi}) \mid \boldsymbol{\tau}_0^{0:K-1} \right] \right] \\
\leq &\mathbb{E}_{\boldsymbol{\tau}_0^{0:K-1}} \Bigg[2(\sigma_1)^2 + 2K\alpha^2 \sum_{i=0}^{K-1} \Bigg[\left(2m_2^2 + 2\sigma_2^2 \right) \times \left(3c_1^2 + 3\left((\hat{\Delta}_J)^2 + (\hat{\sigma}_J)^2 \right) \right) \times \\
&6^{K-i-1} \Bigg[\left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 + \alpha^2 (\hat{\sigma}_H)^2 \right)^{K-i-1} - \left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-i-1} \Bigg] + \\
&(\sigma_2)^2 \left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-i-1} \times \left(3c_1^2 + 3\left((\hat{\Delta}_J)^2 + (\hat{\sigma}_J)^2 \right) \right) + \\
&m_2^2 \left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-i-1} \times (\hat{\sigma}_J)^2 \mid \boldsymbol{\tau}_0^{0:K-1} \Bigg] \\
\leq &2(\sigma_1)^2 \\
&+ (6K\alpha^2 \sigma_2^2) \left(\left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-1} - 1 \right) \left(c_1^2 + (\hat{\Delta}_J)^2 + (\hat{\sigma}_J)^2 \right) \\
&+ (2K\alpha^2 m_2^2) \left(\left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-1} - 1 \right) (\hat{\sigma}_J)^2 \\
&+ 2\alpha^2 (m_1^2 + 3\sigma_2^2) \left(c_1^2 + (\hat{\Delta}_J)^2 + (\hat{\sigma}_J)^2 \right) \\
&+ (6^K K\alpha^2 (12m_2^2 + 12\sigma_2^2)) \left(c_1^2 + (\hat{\Delta}_J)^2 + (\hat{\sigma}_J)^2 \right) \times \\
&\left(\left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 + \alpha^2 (\hat{\sigma}_H)^2 \right)^{K-1} - \left((1 + \alpha c_2)^2 + \alpha^2 (\hat{\Delta}_H)^2 \right)^{K-1} \right)
\end{aligned}
\tag{62}
$$

Then combine terms (i)-(ii) together, that is

$$
\begin{aligned}
&\mathbb{V}\left[\nabla_{\boldsymbol{\phi}} \hat{J}^{\mathrm{K}}(\boldsymbol{\phi}) \right] \\
\leq &O\Bigg((V_1 + \hat{\Delta}_H^2)^{K-1} \left(\mathbb{E}[\|\hat{\boldsymbol{\theta}}^K - \boldsymbol{\theta}^K\|^2] + (K-1) \right) \\
&+ \left(V_2 + (V_1 + \hat{\Delta}_H^2 + \hat{\sigma}_H^2)^{K-1} - (V_1 + \hat{\Delta}_H^2)^{K-1} \right) (\hat{\Delta}_J^2 + \hat{\sigma}_J^2) \Bigg)
\end{aligned}
\tag{63}
$$

which concludes the proof of Theorem 4.5. □

H Supporting Lemmas

In this section, we present the supporting lemmas.

Definition H.1. Let X be a random vector in $\mathbb{R}^d$. Then the norm of X is

$$
\|X\| := \sqrt{\sum_i X_i^2} \tag{64}
$$

Lemma H.2. *Let X be a random vector in $\mathbb{R}^d$ with finite second moment, where $\mathbb{E}[\|X\|^2] \leq +\infty$. Then $\|\mathbb{E}[X]\| \leq \mathbb{E}[\|X\|]$, $\|\mathbb{E}[X]\|^2 \leq \mathbb{E}[\|X\|^2]$.*

Proof. Due to the convexity of norm operator, we can have $\|\mathbb{E}[X]\| \leq \mathbb{E}[\|X\|]$ using Jensen's inequality. Further we can get $\|\mathbb{E}[X]\|^2 \leq (\mathbb{E}[\|X\|^2])^2 \leq \mathbb{E}[\|X\|^2]$ and the statement follows. □

Lemma H.3. *Let X and Y be two random variables in $\mathbb{R}^d$ with finite second moment. Then $\mathbb{E}[\|X+Y\|] \leq \mathbb{E}[\|X\|] + \mathbb{E}[\|Y\|]$.*

Proof. According to Minkowski's inequality that $(\mathbb{E}[\|X+Y\|^p])^{1/p} \leq (\mathbb{E}[\|X\|^p])^{1/p} + (\mathbb{E}[\|Y\|^p])^{1/p}$, set $p = 1$ and the statement follows. □

Definition H.4. Let X be a random vector with values in $\mathbb{R}^d$. Then the variance of X is

$$\mathbb{V}[X] := \mathbb{E}[\|X - \mathbb{E}[X]\|^2] \tag{65}$$

Lemma H.5 (Properties of the variance)**.** *Let X and Y be two independent random variables in $\mathbb{R}^d$. We also assume that X, Y, have finite second moment. Then the following hold.*

(i) $\mathbb{V}[X] = \mathbb{E}[\|X\|^2] - \|\mathbb{E}[X]\|^2$,

(ii) *For every $x \in \mathbb{R}^d$, $\mathbb{E}[\|X - x\|^2] = \mathbb{V}[X] + \|\mathbb{E}[X] - x\|^2$. Hence, $\mathbb{V}[X] = \min_{x \in \mathbb{R}^d} \mathbb{E}[\|X - x\|^2]$,*

(iii) $\mathbb{V}[X+Y] = \mathbb{V}[X] + \mathbb{V}[Y]$.

Proof. (i)-(ii): Let $x \in \mathbb{R}^d$. Then, $\|X-x\|^2 = \|X-\mathbb{E}[X]\|^2 + \|\mathbb{E}[X]-x\|^2 + 2(X-\mathbb{E}[X])^\top(\mathbb{E}[X]-x)$. Hence, taking the expectation we get $\mathbb{E}[\|X-x\|^2] = \mathbb{V}[X] + \|\mathbb{E}[X]-x\|^2$. Therefore, $\mathbb{E}[\|X-x\|^2] \geq \mathbb{V}[X]$ and for $x = \mathbb{E}[X]$ we get $\mathbb{E}[\|X-x\|^2] = \mathbb{V}[X]$. Finally, for $x = 0$ we get (i).

(iii): Let $\bar{X} := \mathbb{E}[X]$ and $\bar{Y} := \mathbb{E}[Y]$, we have

$$\begin{aligned}
\mathbb{V}[X+Y] &= \mathbb{E}[\|X - \bar{X} + Y - \bar{Y}\|^2] \\
&= \mathbb{E}[\|X - \bar{X}\|^2] + \mathbb{E}[\|Y - \bar{Y}\|^2] + 2\mathbb{E}[X - \bar{X}]^\top \mathbb{E}[Y - \bar{Y}] \\
&= \mathbb{E}[\|X - \bar{X}\|^2] + \mathbb{E}[\|Y - \bar{Y}\|^2]
\end{aligned}$$

Recalling the definition of $\mathbb{V}[X]$ the statement follows. □

Definition H.6. *(Conditional Variance)*. Let X be a random variable with values in $\mathbb{R}^d$ and Y be a random variable with values in a measurable space $\mathcal{Y}$. We call *conditional variance* of X given Y the quantity

$$\mathbb{V}[X \mid Y] := \mathbb{E}[\|X - \mathbb{E}[X \mid Y]\|^2 \mid Y].$$

Lemma H.7. *(Law of total variance) Let X and Y be two random variables, we can prove that*

$$\mathbb{V}[X] = \mathbb{E}[\mathbb{V}[X \mid Y]] + \mathbb{V}[\mathbb{E}[X \mid Y]] \tag{66}$$

Proof.

$$\begin{aligned}
\mathbb{V}[X] &= \mathbb{E}[\|X - \mathbb{E}[X]\|^2] \\
&= \mathbb{E}[\|X\|^2] - \|\mathbb{E}[X]\|^2 \\
&= \mathbb{E}[\mathbb{E}[\|X\|^2 \mid Y]] - \|\mathbb{E}[\mathbb{E}[X \mid Y]]\|^2 \\
&= \mathbb{E}[\mathbb{V}[X \mid Y] + \|\mathbb{E}[X \mid Y]\|^2] - \|\mathbb{E}[\mathbb{E}[X \mid Y]]\|^2 \\
&= \mathbb{E}[\mathbb{V}[X \mid Y]] + \left(\mathbb{E}[\|\mathbb{E}[X \mid Y]\|^2] - \|\mathbb{E}[\mathbb{E}[X \mid Y]]\|^2\right)
\end{aligned}$$

recognizing that the term inside the parenthesis is the conditional variance of $\mathbb{E}[X|Y]$ gives the result. □

Lemma H.8. *Let ζ and η be two independent random variables with values in $\mathcal{Z}$ and $\mathcal{Y}$ respectively. Let $\psi\colon \mathcal{Y} \to \mathbb{R}^{m\times n}$, $\phi\colon \mathcal{Z} \to \mathbb{R}^{n\times p}$, and $\varphi\colon \mathcal{Y} \to \mathbb{R}^{p\times q}$ matrix-valued measurable functions. Then*

$$\mathbb{E}[\psi(\eta)(\phi(\zeta) - \mathbb{E}[\phi(\zeta)])\varphi(\eta)] = 0 \tag{67}$$

Proof. Since, for every $y \in \mathcal{Y}$, $B \mapsto \psi(y)B\varphi(y)$ is linear and ζ and η are independent, we have

$$\mathbb{E}[\psi(\eta)(\psi(\zeta) - \mathbb{E}[\psi(\zeta)])\varphi(\eta)\,|\eta] = \psi(\eta)\mathbb{E}\big[\phi(\zeta) - \mathbb{E}[\phi(\zeta)]\big]\varphi(\eta) = 0.$$

Taking the expectation the statement follows. □

I Experiment

Computational resources. For compute resources, We used one internal compute servers which consists consisting of $2x$ Tesla A100 cards and 256 CPUs, however each model is trained on at most 1 card.

I.1 Tabular MDP

I.1.1 Experimental Settings

We adopt the tabular random MDP setting presented in [33]. The dimension is 20 for state space and 5 for action space, so we have the reward matrix $R \in \mathbb{R}^{20\times 5}$. The transition probability matrix is generated from independent Dirichlet distributions. The policy is a matrix $\theta^0 \in \mathbb{R}^{20\times 5}$. The final policy π_θ is obtained by adopting Softmax activation on this policy matrix: $\pi_\theta(a \mid s) = \exp(\theta(s,a))/\sum_b \exp(\theta(s,b))$. We set the initial policy as the uniform policy (by setting θ^0 as zero matrix) in MAML and LIRPG experiment. We conduct the inner-loop update stating from the same point for several times when estimating the meta-gradient correlation and variance. For accuracy measurement between estimation $x \in \mathbb{R}^L$ and ground truth $y \in \mathbb{R}^L$, we use the following equation:

$$\mathrm{Acc}(x,y) := \frac{x^T y}{\sqrt{x^T x}\sqrt{y^T y}}. \tag{68}$$

I.1.2 Implementation for decomposing Gradient estimation

To decompose the gradient estimation effects brought by different sources, such as outer estimation variance and inner estimation bias (compositional bias, hessian estimation error), we utilise the following implementation trick: Using estimator I to estimate $\theta'_c = \theta + \alpha\nabla J(\theta)$, estimator II to estimate $\theta'_h = \theta + \alpha\nabla J(\theta)$ and finally combine them with: $\theta' = \perp \theta'_c + \theta'_h - \perp \theta'_h$, where $\perp$ is the "stop gradient" operator. By this implementation trick, we can have the following property: $\theta' \to \theta'_c$ and $\nabla_\theta \theta' = \nabla_\theta \theta'_h$, where $\to$ is the "evaluates to" operator. "Evaluates to" operator $\to$ is in contrast with $=$, which also brings the equality of gradients. By "Evaluates to" operator, the "stop gradient" operator means that $\perp(f_\theta(x)) \to f_\theta(x)$ but $\nabla_\theta \perp(f_\theta(x)) \to 0$. This property guarantee that the compositional bias is only influenced by estimator I while hessian estimation error is controlled by estimator II. Besides estimator I and estimator II, an extra estimator III is used for outer-loop policy gradient $\nabla_{\theta^k} J^{\mathrm{out}}(\pi^k)$ estimation, which helps us understand the effect of outer-loop policy gradient.

I.1.3 Additional Experimental Results on Tabular MAML-RL

We offer additional experimental results on more estimators (DiCE/ Loaded-DiCE)/settings (All 7 permutations)/metrics (variance of Meta-gradient estimation).

Ablation study on sample size and estimator. Additional experimental results are shown in Fig. 5. The comparison between SSS, SES, ESS and SSE, SEE, ESE reveals the importance of the outer-loop gradient estimation. Accurate outer-loop policy gradient estimation brings more significant improvement over the correlation compared with the correction of Hessian error or compositional bias. In addition, with estimated outer-loop policy gradient, the correction of these two terms also helps ($EES > SES > ESS > SSS$).

Next we discuss the comparison between different estimators. The DiCE estimator have real high variance on first-order and second-order, and its first-order gradient corresponds to the REINFORCE algorithm [38] while the rest 3 estimators' first-order gradient corresponds to the Actor-critic algorithm. That is why DiCE performs the worst in all cases. With stochastic outer-loop estimation, the LVC and Loaded-DiCE estimator have comparable correlation while the variance of LVC is smaller than Loaded-DiCE. The AD estimator performs worse than LVC and Loaded-DiCE when the Hessian is estimated (SSE, ESE, SSS, ESS). This corresponds to the conclusion in [29] that

the LVC estimator introduces low-bias and low-variance Hessian estimation while AD estimator has large-bias and low-varaince Hessian estimation. With exact outer-loop estimation, the LVC has relatively great Hessian estimation so the correction of compositional bias has the same effect with Hessian correction ($ESE = SEE > SSE$), while the Hessian correction is still important in Loaded-DiCE ($SEE > ESE > SSE$).

Ablation study on inner learning rate, step and estimator. Additional ablation study on inner learning rate and number of steps are shown in Fig. 6, 7. The results show that: With more steps and larger learning rates, the inner-loop estimation can become more important than outer-loop policy gradient (the correlation decreases a lot in SSE in all estimators). Also in multi-step and large learning rate setting, the importance of Hessian estimation and compositional bias become comparable in LVC and Loaded-DiCE ($SEE \approx ESE, SES \approx ESS$).

Meta-gradient variance In all three plots Fig. 5, 6, 7, we report additional metric on variance of the meta-gradient estimation. We observe that the correction of compositional bias increases the variance especially when outer-loop policy gradient estimator is poor (estimator III uses stochastic samples) or Hessian variance is large (in DiCE and Loaded-DiCE). Only with low Hessian variance (LVC/AD) and great outer-loop policy gradient (estimator III uses analytical solution), the correction of compositional bias can decrease the variance.

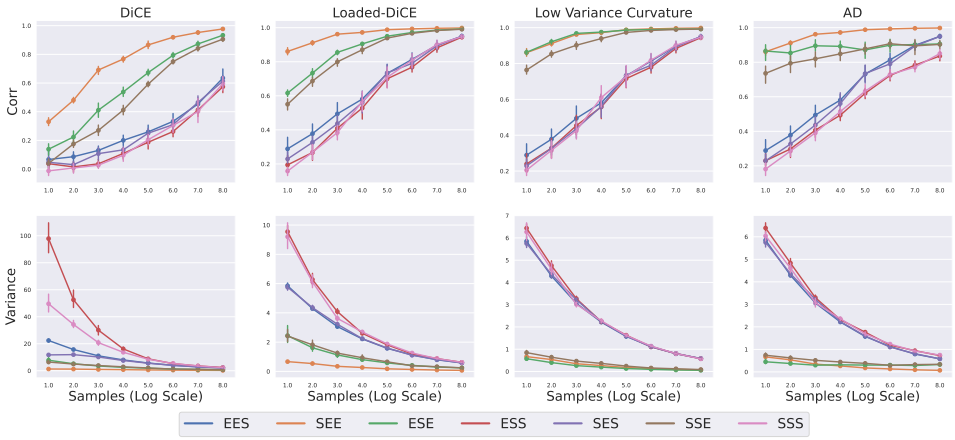


Figure 5: Ablation study on sample size and estimator in 1-step inner-loop setting. (1) Outer-loop policy gradient is important for estimation (2) Compositional bias correction helps increase the correlation (3) The LVC and Loaded-DiCE can achieve higher correlation compared with AD when the Hessian matrix is estimated.

I.1.4 Additional Experimental Results on Tabular LIRPG

In Fig. 8 we offer additional experimental Results on estimation variance. Basically the AD based estimation in LIRPG setting tend to have higher variance.

I.1.5 Hyperparameters

We offer the hyperparameter settings for our Tabular MDP experiment in Table 2.

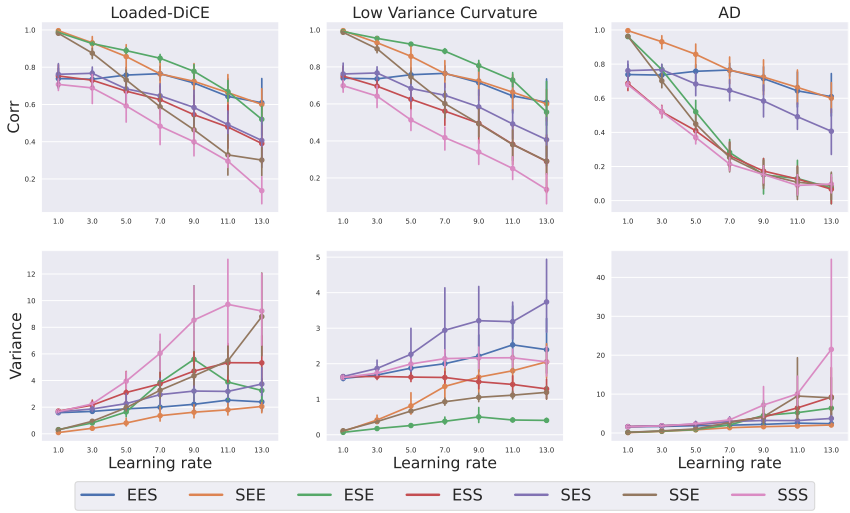


Figure 6: Ablation study on inner learning rate and estimator. (1) In Loaded-DiCE and LVC, With larger learning rate, the compositional bias basically shares the same importance with Hessian estimation error. (2) With larger learning rate, the Hessian estimation problem in AD largely decreases the correlation.

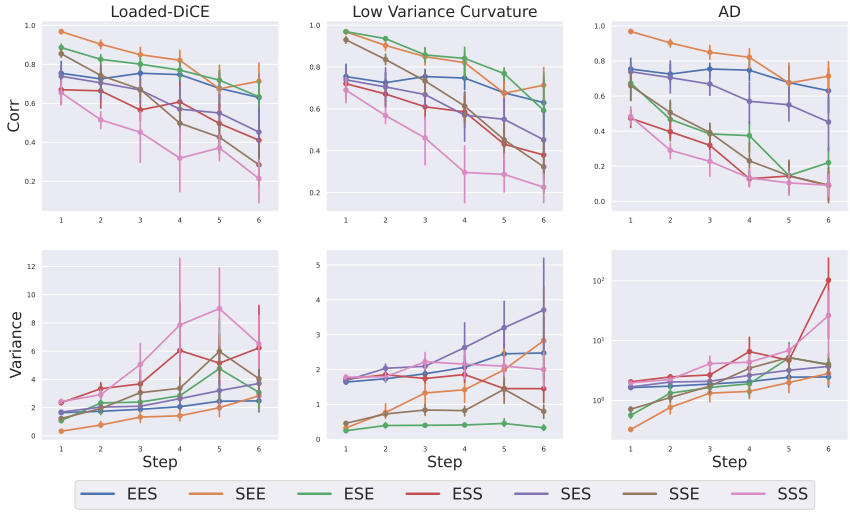


Figure 7: Ablation study on inner step and estimator. Results of larger steps show similar phenomenon with larger inner-loop learning rate.

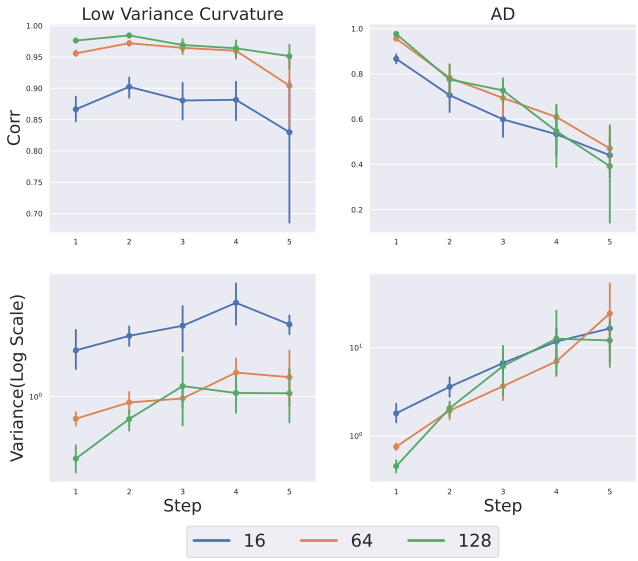


Figure 8: Additional experiment results on. Different color refers to different trajectory sample size.

Table 2: Hyper-parameter settings for Tabular MDP.

SETTINGS	VALUE	DESCRIPTION
TRAJECTORY LENGTH	20	RL TRAJECOTRY LENGTH
DISCOUNT FACTOR	0.8	LEARNING RATE FOR META-SOLVER UPDATES
INNER LEARNING RATE	10	LEARNING RATE FOR INNER-LOOP UPDATE
INNER STEP	1	STEP NUMBER OF INNER-UPDATE
INDEPENDENT TRIALS	10	NUMBER OF INDEPENDENT TRIALS ON ENVIRONMENTS
SAME TRIALS	20	NUMBER OF INDEPENDENT TRIALS ON THE SAME POINT
DIMENSION OF STATE	20	DIMENSION OF STATE
DIMENSION OF ACTION	5	DIMENSION OF ACTION
NOISE COEFFICIENT	1.0	NOISE FACTOR FOR SIMULATING ESTIMATED VALUE FUNCTION
DENSITY	0.001	PARAMETERS OF DIRICHILET DISTRIBUTION

I.2 LOLA-DiCE on Iterated Prisoner Dilemma (IPD)

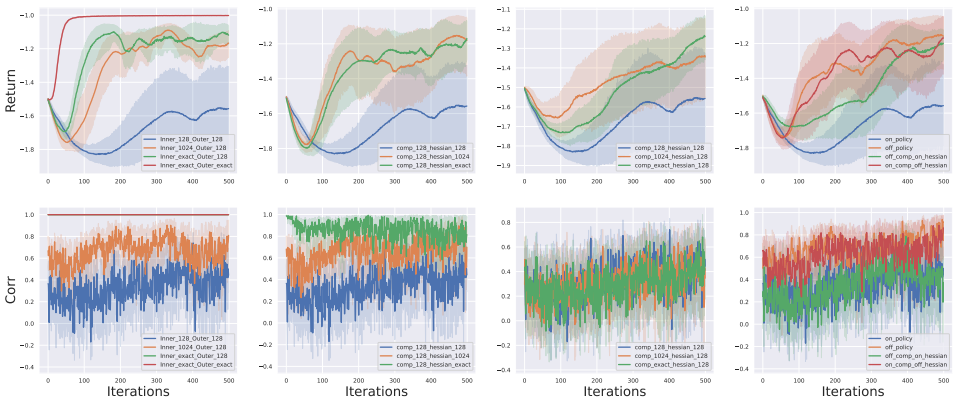
I.2.1 Experimental Settings

In Iterated Prisoner Dilemma, the Prisoner Dilemma game is played repeatedly by the same players. The payoffs of Prisoner Dilemma for players are shown as follows.

$$\mathbf{R}^1 = \begin{bmatrix} -2 & 0 \\ -3 & -1 \end{bmatrix} \quad \mathbf{R}^2 = \begin{bmatrix} -2 & -3 \\ 0 & -1 \end{bmatrix},$$

where the action 0 (correpsonds to column/row 0) as "cooperation" (don't confess) and the action 1 (correpsonds to column/row 1) as "defection" (confess). Agent in Iterated Prisoner Dilemma aims at maximising the cumulative Discounted reward. By LOLA-DiCE algorithm, it is possible for both agent to reach social welfare: (-1, -1). Refer to Appendix A.2 for how the algorithm is formulated.

We conduct our experiment by adapting code from the official codebase*. The official code only conducts the experiment using one fixed seed and the performance is highly sensitive to different random seeds using default hyperparameters. To evaluate the performance reliably, we conduct all the experiments for 10 random seeds and report the average result.



(a) LOLA-DiCE (b) Ablation: Hessian Variance (c) Ablation: Compositional Bias (d) off-policy LOLA-DiCE

Figure 9: Experiment result of LOLA-DiCE. (a) Poor inner-loop estimation can fail the LOLA-DiCE algorithm. (b) Hessian estimation variance is the main problem in LOLA-DiCE. (c) The correction of compositional bias also helps increase the average return. (d) The off-policy correction can both decrease the compositional bias and Hessian estimation variance, which largely increases the final return.

I.2.2 Additional Experimental Results

Ablation on LOLA-DiCE inner/outer estimation. We report the correlation result of conducting ablation study for different inner/outer-loop estimation of LOLA-DiCE in the Fig. 9(a). Higher correlation does not guarantee higher return. The bonus brought by setting inner-loop as exact solution have a really large improvement over correlation (from 0.7 to 1.0) but have limited improvement on return. We believe it is because the outer-loop gradient estimation becomes the main issue when inner-loop estimation is really well.

Ablation on LOLA-DiCE Hessian variance and compositional bias. We show additional experimental result in Fig. 9(c). An interesting thing is that we find out the gradient correlation of these three settings are comparable. An possible explanation is that the main issue here is the hessian variance and this is why the performance gain by lowering hessian variance is larger that lowering compositional bias. Though by correcting compositional bias LOLA can have better estimation with performance gain, the gain is not obvious in the aspect of gradient correlation because the hessian variance is still large.

Off-policy DiCE and ablation study The correlation gain for off-policy comp&on-policy hessian is still limited like that in Fig. 9(c). But the performance gain verifies the bonus brought by correcting compositional bias.

I.2.3 Hyperparameters

We offer the hyperparameter settings for our LOLA experiment in Table 3.

*https://github.com/alexis-jacq/LOLA_DiCE

Table 3: Hyper-parameter settings for LOLA-DiCE.

SETTINGS	VALUE	DESCRIPTION
OUTER LEARNING RATE	0.1	OUTER LEARNING RATE
INNER LEARNING RATE	0.3	INNER LEARNING RATE
DISCOUNT FACTOR	0.96	DISCOUNT FACTOR
UPDATE	500	STEP NUMBER OF META-UPDATE
ROLLOUT LENGTH	100	LENGTH OF IPD ROLLOUT
INNER STEP	1	NUMBER OF VIRTUAL INNER-STEP LOOK-AHEAD
VALUE FUNCTION LEARNING RATE	0.1	VALUE FUNCTION LEARNING RATE
OFF-POLICY BUFFER SIZE	1024	BUFFER SIZE
SAMPLE BATCH SIZE	128	COMP/HESSIAN/OUTER SAMPLE BATCH SIZE

I.3 MGRL on Atari games

I.3.1 Experimental setting

We reimplement the MGRL algorithm based on A2C baseline. In this case, Meta-parameters ϕ involves 4 hyperparameters: Discount factor, value loss coefficient, entropy loss coefficient and GAE ratio. The procedures of 'discard' strategy we use is summarized as follows: Starting from the inner-policy parameters θ^0, we utilise take 3 A2C updates and get the 3-step updated policy θ^3. Then we can calculate the meta-gradient by backpropogating from $R(\theta^3)$ to the meta parameters. Finally we reset the inner-loop policy parameters back to θ^1 so the rest 2 updates are in fact virtual update. It is only used for the meta-gradient estimation.

I.3.2 Discussion on the 'Discard' strategy

In the MGRL experiment, we follow previous work [4] for conducting multi-step MGRL. So the inner-loop policy will take multi-step virtual updates for meta-parameters update. As mentioned in Section 4.2 in their paper, this strategy can only keep the RL update times unchanged among different algorithms and is not particularly sample efficient because they need to take virtual look-ahead for the update of meta-parameters. However, one benefit of adopting such strategy is that we can keep the amount of meta-update large enough to verify the effect brought by the LVC correction. We also take some experiments on another setting where we take meta-update after each 3-step inner-loop update. Note that they are no longer virtual inner-loop updates. However, we find out that in many environment this setting largely decrease the meta-update times and make the comparison of different meta-gradient estimation less meaningful.

I.3.3 Additional Experimental results

We offer the full experiments results on all 8 Atari games: Asteroids, Qbert, Tennis, BeamRider, Alien, Assault, DoubleDunk, Seaquest. The reward performance is shown in Fig. 10. We also offer trajectories for all 4 meta-parameters on these experiments in Fig. 11. From Fig. 10 it can show that MGRL with LVC correction can achieve comparable or better performance in almost all 8 environments. Note that we need to clarify that in some RL experiments the MGRL cannot achieve better performance compared with A2C baseline. This also corresponds to the experimental results in original MGRL paper [39]. However, since our main comparison only happens between MGRL and MGRL with LVC correction, it is still a fair comparison to verify the effectiveness of LVC hessian correction. Fig. 11 reveals that even we have only 4 meta-parameters, different meta-gradient estimation can still results in large gap between the meta optimisation trajectory and final GMRL performance.

I.3.4 Implementations and hyperparameters

We adopt the codebase of A2C from [21] and differentiable optimization library [28] to implement MGRL algorithms. We use a shared CNN network (3 Conv layers and one fully connected (FC) layer) for the policy network and critic network. The (out-channel, filters, stride) for each Conv layer is (32, 8×8, 4), (64, 4×4, 2) and (64, 3×3, 1) respectively while the hidden size is 512 for the FC layer. For the training loss, we adopt additional entropy regularisation for policy loss and Mean Square Error (MSE) for the value loss. We adopt the Generalized Advantage estimation (GAE) for

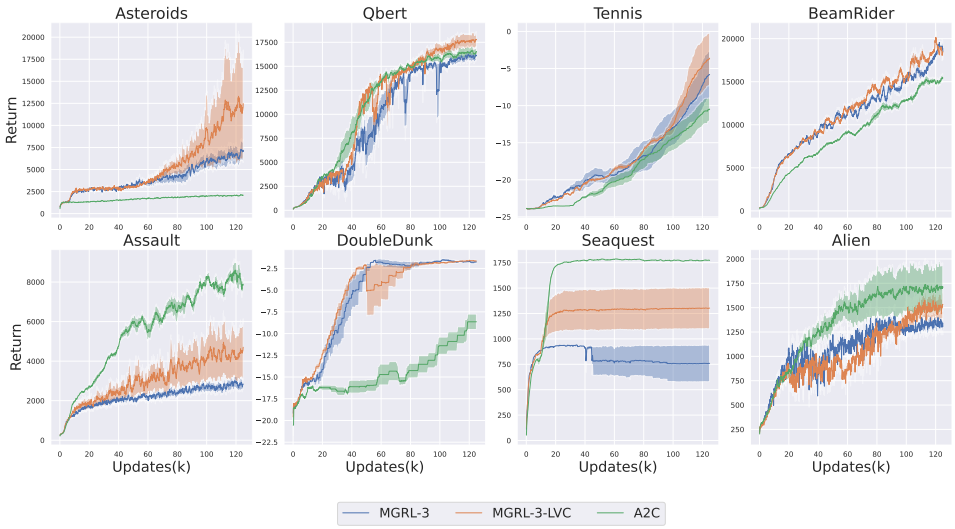


Figure 10: Experimental results on Atari game over 5 seeds. 3-step MGRL with LVC correction can achieve at least the same performance compared with 3-step MGRL in basically all environments.

advantage estimation. We offer the hyperparameter settings for our experiment in Table 4. We tun our algorithm for 125k inner updates, which corresponds to 40M environment steps for baseline A2C.

Table 4: Hyper-parameter settings for MGRL.

Settings	Value	Description
Inner Learning rate	7e-4	Inner Learning rate
Learning rate Scheduling	Linear decay	linearly decrease to 0
Discount factor	0.99	Discount factor
GAE lambda	0.95	ratio of generalized advantage estimation
Value coef	0.5	coefficient of value loss
entropy coef	0.01	coefficient of entropy loss
Update	125k	number of inner update
Number of process	64	number of multi process
number of step per update	5	number of step per update
Meta update	3	number of inner-update for conducting meta-update
Meta learning rate	0.001	meta learning rate
Inner optimizer	Adam	Inner-loop optimizer
Outer optimizer	Adam	Outer-loop optimizer

J Author contribution

We summarise the main contributions from each of the authors as follows:

Bo Liu: Algorithm design, main theoretical proof, some code implementation and experiments running (on tabular MDP and MGRL), and paper writing.

Xidong Feng: Idea proposing, algorithm design, part of theoretical proof, main code implementation and experiments running (on tabular MDP, LOLA and MGRL), and paper writing.

Jie Ren: Code implementation and experiments running for MGRL.

Luo Mai: Project discussion and paper writing.

Rui Zhu: Project discussion and paper writing.

Jun Wang: Project discussion and overall project supervision.

Yaodong Yang: Project lead, idea proposing, experiment supervision, and whole manuscript writing.

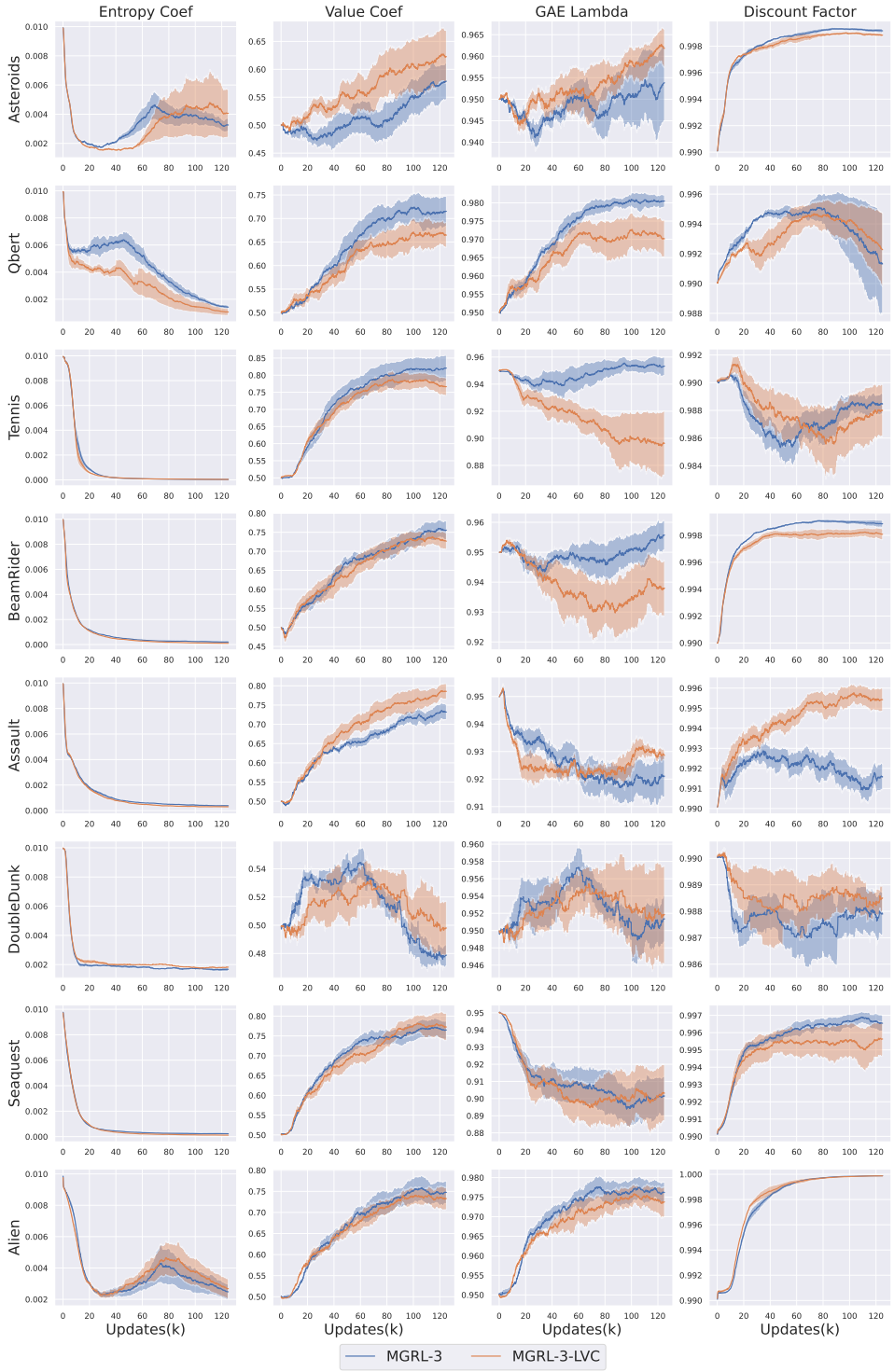


Figure 11: 4 Meta parameters trajectories on Atari game for 3-step MGRL and 3-step MGRL with LVC correction.