Settling the Bias and Variance of Meta-Gradient Estimation for Meta-Reinforcement Learning

Bo Liu\textsuperscript{1,*}, Xidong Feng\textsuperscript{2,*}, Haifeng Zhang\textsuperscript{1}, Jun Wang\textsuperscript{2}, Yaodong Yang\textsuperscript{3,†}

\textsuperscript{1}Institute of Automation, CAS \textsuperscript{2}University College London \textsuperscript{3}Peking University

Abstract

Meta-Reinforcement Learning (Meta-RL) refers to the methods that conduct meta-learning on reinforcement learning tasks. In the meta-RL framework, an outer-loop learning procedure is maintained to help the inner-loop reinforcement learner learn fast adaptations thus better generalisation performance. When the inner-loop is optimised by policy gradient methods, it is also called gradient based Meta-RL (GMRL). In recent years, GMRL methods have achieved remarkable successes in either discovering effective online hyperparameter for one single task (Xu et al., 2018) or learning good initialisation for multi-task transfer learning (Finn et al., 2017). Despite the empirical successes, it is often neglected that computing meta-gradients via vanilla backpropagation is ill-defined. In this paper, we argue that the stochastic meta-gradient estimation adopted by many existing MGRL methods are in fact biased; the bias comes from two sources: 1) the compositional bias that is inborn in the structure of compositional optimisation problems and 2) the bias of multi-step Hessian estimation caused by direct automatic differentiation. To better understand

\*Equal Contributions. † Corresponding Author. Email to: benjaminliu.eecs@gmail.com, xidong.feng.20@ucl.ac.uk, yaodong.yang@pku.edu.cn
the meta-gradient biases, we perform the first of its kind study to quantify the amount for each of them. We start by providing a unifying derivation for existing GMRL algorithms, and then theoretically analyse both the bias and the variance of existing gradient estimation methods. On understanding the underlying principles of bias, we propose two mitigation solutions based on off-policy correction and multi-step Hessian estimation techniques. The off-policy inner-loop policy gradient correction can introduce more samples to lower the inner-loop variance and finally the compositional bias. The multi-step Hessian estimation techniques changes the form of inner-loop policy gradient and this new form can enable us with low/zero-bias Hessian estimation with automatic differentiation. Comprehensive ablation studies have been conducted and results reveals: (1) The existence of these two biases and how they influence the meta-gradient estimation when combined with different estimator/sample size/step and learning rate. (2) The effectiveness of these mitigation approaches for meta-gradient estimation and thereby the final return on two practical Meta-RL algorithms: LOLA-DiCE and Meta-gradient Reinforcement Learning.

1 Introduction

Reinforcement Learning (RL) has achieved remarkable success in many different fields like such as games (Mnih et al. (2013)), chess (Schrittwieser et al. (2020)), and control (Levine et al. (2016)). Despite the large success, RL is still suffering from the low sample-efficiency problem even using advanced algorithms (Schulman et al. (2017), Haarnoja et al. (2018)). It typically takes far more environment interactions to achieve comparable performance with human intelligence. Yet, human demonstrates remarkable adaptation ability — human players can learn to play Atari games in minutes (Tsividis et al. (2017)) while it takes hours or days for a RL agent (Schulman et al. (2017), Haarnoja et al. (2018)). It is believe that one main reason behind humans’ fast learning ability is that humans possess thousands of years’ prior knowledge about the world, which enables to transfer previous skills or knowledge to new scenarios with no need of extensive repeated learning (Barbey, 2018). As highlighted by
Sutton (2019), the AI algorithms should build in some meta-methods; AI agents should learn the fundamental of learning: the meta-knowledge and skills about how to learn just like what human intelligence does, rather than rediscover the knowledge humans have already known.

Meta Learning, also known as learning to learn, is proposed to equip machine with the same meta knowledge (Schmidhuber (1987)). Meta-Reinforcement Learning (Meta-RL) in the early stage aims at learning RL agent with prior knowledge about environments to achieve the same fast reinforcement learning ability in unseen scenario (Duan et al. (2016), Finn et al. (2017), Gupta et al. (2018)). These work focuses on few-shot setting where the agent can only get very few trajectories for adaptation in new environment. One classical algorithm, MAML-RL (Finn et al., 2017), meta-learns the initial parameters and can enable a RL agent to adapt to a new environment with the reinforcement learner takes few policy gradient steps.

Although many existing Meta-RL works (Rakelly et al. (2019), Fu et al. (2020)) are still concentrating on the fast adaptation ability, the purpose of Meta-RL algorithms have been further explored beyond the scope of learning fast adaptations. An important direction is to conduct online meta-gradient RL for adaptively tuning the hyper-parameters (Xu et al. (2018), Zahavy et al. (2020)) or the intrinsic reward (Zheng et al. (2018)). One main feature of these online meta-gradient RL works is the alternate optimisation between an inner-loop policy optimisation and an outer-loop meta optimisation. Additionally, works such as (Oh et al. (2020), Zheng et al. (2020), Feng et al. (2021)) try to meta-learn some fundamental concepts or algorithmic components of RL algorithms by inverse-design. Such meta models are trained for generalisation ability over new RL environments.

In general, Meta-RL refers to the algorithms that involve bi-level optimisation procedure with the inner-loop being a RL subroutine. Gradient based Meta-RL (GMRL) is one important implementation framework for Meta-RL where policy gradient updates are often taken in the inner-loop. A general GMRL formulation with a \( k \)-step inner-loop policy gradient subroutine can be written as follows:

\[
\max_{\phi} J(\phi) := J^{\text{Out}}(\phi, \theta^k), \theta^i = \theta^0 + \alpha \nabla_{\theta^i-1} J^{\text{In}}(\phi, \theta^{i-1}), i \in \{1, 2...k\} \tag{1.1}
\]

where \( \theta \) refers to policy parameters, \( \phi \) refers to meta-parameters, \( \alpha \) represents the learning
rate and \( J_{\text{In/Out}} \) usually refers to the value function over RL agent (see a detailed illustration in Section 3). The general objective for GMRL is to optimise the meta-parameters to maximise the final outer objective function through \( k \) inner-step RL updates. Since the meta gradient estimation needs to differentiate through all policy gradient updates, the meta gradient estimation becomes more non-trivial than the traditional policy gradient, therefore becoming one of the major research challenges in Meta-RL literature Rothfuss et al. (2018), Tang et al. (2021).

Existing works (Rothfuss et al. (2018), Tang et al. (2021), Liu et al. (2019)) focus on the Hessian estimation problem in the scope of MAML-RL and proposed different Hessian estimator or higher-order baselines to decrease the Hessian estimation error in MAML-RL. In this paper, we argue that there exist two main issues in a broad range of GMRL: Plug-in compositional bias and Multi-step Hessian estimation bias beyond the scope of MAML-RL in many new Meta-RL algorithms. Specifically, the first compositional bias comes from the compositional optimisation structure of gradient based Meta-RL problem — the outer meta-gradient estimation needs to differentiate through inner-loop stochastic optimisation subroutine. To the best of our knowledge, this is the first work in Meta-RL that offers both theoretical analysis and remedies to this compositional bias. The second bias occurs in the high-order policy gradient estimation especially Hessian estimation problem. This problem has already been discussed in (Rothfuss et al., 2018; Tang et al., 2021) in the scope of MAML-RL. Beyond MAML-RL, we argue that many new gradient based Meta-RL algorithms are still suffering from the same Hessian estimation bias because they directly use automatic differentiation to obtain gradient estimation. Coincidentally, though most of works neglect this potential problem, they can still get unbiased meta-gradient if they set \( k = 1 \) in Eq. (1.1). However, the same issue happens again if they set \( k > 1 \) and conduct multiple inner-loop policy gradient updates. And that is why we call it Multi-step Hessian estimation bias.

To better understand the meta-gradient bias, we first present a unifying framework to interpret different Meta-RL research topics. This framework enables us to conduct general theoretical analysis over existing GMRL algorithms and to understand the impact of these
two issues over the final quality of meta-gradient estimation. To mitigate these two issues, we propose to adopt off-policy correction and multi-step Hessian estimator for meta-gradient estimation. We conduct empirical results over an tabular MDP to validate the influences of these two biases combined with different estimator/sample size/step and learning rate. Experiments on two practical Meta-RL algorithms further validates influence of these two estimation biases and by using the mitigation approaches we propose, the estimation error can decrease and bring in final reward improvement.

The paper is structured as follows: we firstly offer related work in Section 2. In Section 3, we present the framework for better interpreting GMRL. Based on this, we offer systematical discussion and theoretical analysis on meta-gradient estimation bias/variance in Section 4. Starting from the discussion and theoretical analysis, we offer remedies to fix the problem in Section 5. And finally in Section 6, we validate the influence on meta-gradient estimation by these two biases and the effectiveness of our proposed remedies with experimental results.

2 Related Work

In this section we summary related work for four active research fields in Meta-RL. We will detail the specific formulation for each topic in the next section.

**Few-shot RL.** The idea of few-shot RL is to enable RL agent with fast adaptation ability. The RL agent is only allow to interact with the environment for a few trajectory to conduct task-specific adaptation. The research of few-shot RL mainly consists of two topics: gradient based and context based few-shot Meta-RL. Context based Meta-RL involves works Duan et al. (2016), Wang et al. (2016), Rakelly et al. (2019), Fu et al. (2020), which utilise neural network such as recurrent neural network to embed the information from few-shot interactions to obtain task-relevant context. The task-specific information can reduce the environment uncertainty and enable the fast adaptation of RL agent. Rothfuss et al. (2018), Al-Shedivat et al. (2017), Liu et al. (2019) in gradient based few-shot Meta-RL follow the setting of Finn et al. (2017) to meta-learn the model’s initial parameters through meta-gradient descent with better meta-gradient estimation.
Opponent Shaping. Existing work Foerster et al. (2017), Letcher et al. (2018), Kim et al. (2020) tried to consider opponent learning process and incorporate it into self-learning process. By modelling opponents’ learning process, the multi-agent learning process can reach better social welfare (Foerster et al. (2017), Letcher et al. (2018)) or the ego-agent can adapt to a new and learning peer agent (Kim et al. (2020)). Meta-gradient estimation in such works involve the differentiation over other-agent RL updates. This research topic extends the setting of meta-gradient RL to multi-agent scenarios.

Online Meta-gradient RL. Existing work proposed different online meta-gradient RL algorithms for online adaptation over meta-parameters to enhance the RL performance in single RL learning process, such as hyperparameters in Xu et al. (2018) and Zahavy et al. (2020), intrinsic reward generator in Zheng et al. (2018), auxiliary loss in Veeriah et al. (2019), reward shaping mechanism in Hu et al. (2020), and value correction in Zhou et al. (2020). The main feature of online meta-gradient RL is that they are under single-lifetime setting (Xu et al. (2020)), so the algorithm basically only iterates through one RL learning process. This is consistent with traditional RL algorithms.

Meta-gradient based Inverse Design. There are lots of previous attempts to meta-learn some fundamental concepts in RL. In this research field, most works are under multi-lifetime setting and the final meta model can generalize to different environments. The term 'multi-lifetime' refers to multiple tasks or environments (or a task distribution) in contrast to 'single-lifetime' in online meta-gradient RL. Existing work tried to learn meta modules such as loss function/RL algorithm in Oh et al. (2020), Bechtle et al. (2021) and Kirsch et al. (2019), intrinsic reward generator in Zheng et al. (2020), target value function in Xu et al. (2020), options in Veeriah et al. (2021), and multi-agent auto-curricula in Feng et al. (2021).

Meta-gradient Estimation in RL. Starting from Al-Shedivat et al. (2017), existing works began to focus on the meta-gradient estimation problem. Al-Shedivat et al. (2017) discussed the biased estimation problem of MAML and proposed the E-MAML formulation to fix the meta-gradient bias. Following works Rothfuss et al. (2018), Liu et al. (2019), Tang et al. (2021) tried to fix the estimation bias or lower meta-gradient estimation variance to increase
the performance. Also there exist some works Foerster et al. (2018b), Farquhar et al. (2019), Mao et al. (2019) that mainly discussing the high-order gradient estimation in RL. Some recent work (Bonnet et al. (2021), Vuorio et al. (2021)) proposed algorithms to balance the variance and bias tradeoff for meta-gradient estimation. Recently we find a concurrent work Tang (2021), which analyzed the discrepancy between estimation in practical algorithm and the real unbiased meta-gradient in the MAML-RL setting. Our analysis of compositional bias happens to corresponds to the proposition E.3 in Tang (2021). However, we focus one a broader range of settings: (1) one-step to multi-step inner-loop (2) broader meta-gradient RL algorithm (3) Analysis on more estimation error term, while Tang (2021) mainly focus on the problem in MAML-RL.

3 A Unified Framework for Meta-gradient Estimation

In this section, we derive a general formulation for meta-gradient estimation in gradient based Meta-RL and show how different Meta-RL algorithms can be fit into our formulation. This formulation enables us to conduct general analysis about meta-gradient estimation in Meta-RL. We will show how 4 research topics mentioned in the Section 2 can be instantiated under this formulation.

We formulate the single agent RL using Markov Decision Processes (MDPs), defined by \( \langle S, A, P, R, \gamma \rangle \). At each time-step, the RL agent observes a state \( s \in S \), takes an action \( a \in A \) based on the policy \( \pi(a|s) \), transits to the next state \( s' \in S \) and receives the reward function \( r(s, a) \) according to the transition matrix \( P \) and reward matrix \( R \). We define the return \( R(\tau) \) as the discounted sum of rewards along a trajectory \( \tau := (s_0, a_0, \ldots, s_{H-1}, a_{H-1}, s_H) \):

\[
\mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s \right]
\]

The discounted state value function \( V \) and state-action value function \( Q \) is defined by

\[
V^\pi(s) := \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s \right] \quad \text{and} \quad Q^\pi(s, a) := \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a \right]
\]

respectively. For multi-agent RL, we can similarly extend the MDPs to multi-agent MDPs defined by \( \langle S, A, P, R, \gamma \rangle_n \), where \( n \) is the number of the agents. For simplicity, we use \( J^{\text{In/Out}} \) to represent the value function \( V(s) \).

Without the loss of generality, we denote the meta parameters as \( \phi \), and pre/post-adapt
inner parameters as \( \theta \) and \( \theta' \), the meta objective as \( J^{\text{Out}} \), inner loop objective as \( J^{\text{In}} \). So the general objective for gradient based meta reinforcement learning is to maximize the meta objective \( J^{\text{Out}}(\phi, \theta') \), where the post-adapt inner parameters are obtained by taking \( k \) policy gradient steps. For simplicity we omit the expectation over environments/tasks for algorithms that learn generalizable meta modules.

\[
\max_{\phi} \mathcal{J}(\phi) := J^{\text{Out}}(\phi, \theta^{k}), \theta^{i} = \theta + \alpha \nabla_{\theta_{i-1}} J^{\text{In}}(\phi, \theta^{i-1}), i \in \{1, 2 \ldots k\}
\]

(3.1)

where \( \alpha \) refers to the learning rate. Note that all inner-loop updates here refer to expected policy gradient (EPG)and all bias term we discuss is the bias w.r.t the expected meta-gradient in this \( k \)-step EPG inner-loop. In practice, this process can only be approximated with stochastic policy gradient estimation. This discrepancy is further discussed in Al-Shedivat et al. (2017) and a recent work Vuorio et al. (2021). We further discuss our choice of \( k \)-step EPG inner-loop setting in Appendix A.

For instance, when we take one-step inner-loop, we can derive the objective \( J(\phi) \) and corresponding meta-gradient \( \nabla J(\phi) \) as follows.

\[
\max_{\phi} \mathcal{J}(\phi) := J^{\text{Out}}(\phi, \theta^{1}), \theta^{1} = \theta^{0} + \alpha \nabla_{\theta^{0}} J^{\text{In}}(\phi, \theta^{0})
\]

\[
\nabla J(\phi) = \nabla_{\phi} J^{\text{Out}}(\phi, \theta^{1}) + \nabla_{\phi} \theta^{1} \nabla_{\theta^{1}} J^{\text{Out}}(\phi, \theta^{1})
\]

(3.2)

Next, we show how different gradient based meta reinforcement learning algorithms can be instantiated under this formulation. We choose four main research fields mentioned in the related work. For each topic, we pick one typical algorithm to illustrate its formulation.

### 3.1 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). Finn et al. (2017) aims at learning neural network’s initial parameters for fast adaptation on new environments. It assumes distribution $\rho(T)$ over RL environment $T$ and tries to optimise $\theta$ which leads to high-performing updated policy $\theta'$. The objective equation for one-step MAML-RL can be shown as follows:

$$J(\theta) = \mathbb{E}_{T \sim \rho(T)} \left[ \mathbb{E}_{\tau' \sim p_T(\tau' | \theta')} [R(\tau')] \right]$$

with $\theta' := U(\theta, T) = \theta + \alpha \nabla_{\theta} \mathbb{E}_{\tau \sim p_T(\tau | \theta)} [R(\tau)]$ (3.3)

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

In the scope of Eq. (3.2), MAML-RL optimizes over meta initial parameters to maximize the return of one-step adapted policy: $\theta' = \theta + \alpha \nabla_{\theta} J^{\text{in}}(\theta)$. In MAML-RL, $J^{\text{Out}}(\phi, \theta')$ degenerates to $J^{\text{Out}}(\theta')$ and $\phi$ and $\theta$ represent the same initial parameters. The meta-gradient can be derived with the following equation:

$$\nabla J(\theta) = \nabla_{\theta} \theta' \nabla_{\theta} J^{\text{Out}}(\theta'), \nabla_{\theta} \theta' = I + \alpha \nabla_{\theta}^2 J^{\text{in}}(\theta)$$ (3.4)

### 3.2 Meta-gradient in Opponent Shaping

Opponent shaping (Foerster et al. (2018a), Letcher et al. (2018), Kim et al. (2020)) is a powerful tool in multi-agent learning process for different purposes. For instance, Foerster et al. (2018a) and Letcher et al. (2018) 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) (Foerster et al. (2018a)) 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 $\phi$ and agent 2 policy $\theta$, the traditional independent learning (IL) and 1-step LOLA
algorithm can result in different updates for agent 1:

\[
\phi'_{\text{IL}} = \phi + \beta \nabla_\phi J^{\text{Out}}(\phi, \theta)
\]

\[
\phi'_{\text{LOLA}} = \phi + \beta \nabla_\phi J^{\text{Out}}(\phi, \theta'), \text{ where } \theta' = \theta + \alpha \nabla_\theta J^{\text{In}}(\phi, \theta)
\]

(3.5)

Where \( \beta \) 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 \( \phi \), it will optimise its return over one-step-lookahead opponent parameters \( \theta' \). Thus the meta-gradient of meta-agent corresponds exactly to Eq. (3.2) with \( \nabla_\phi \theta' = \alpha \nabla_\phi \nabla_\theta J^{\text{In}}(\phi, \theta) \). 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.

### 3.3 Online Meta-gradient Reinforcement Learning

In this setting, the main objective is to self-tune the meta parameters (\( \gamma \) in Xu et al. (2018)) or meta models (intrinsic model in Zheng et al. (2018)) 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 Baydin et al. (2017), Franceschi et al. (2017). Xu et al. (2018) proposed meta-gradient reinforcement learning (MGRL) to tune the discount factor \( \gamma \) and bootstrapping parameter \( \lambda \) in an online manner. It tries to differentiate through one RL inner update to optimize the meta-parameters and maximise one-step policy return.

\[
\max_{\eta} V^{\pi_{\theta'}} \text{, where } \theta' = \theta - \alpha \frac{\partial J(\tau, \theta, \eta)}{\partial \theta}, \text{ and } \\
-\frac{\partial J(\tau, \theta, \eta)}{\partial \theta} = (g_\eta(\tau) - v_\theta(S)) \frac{\partial \log \pi_\theta(A | S)}{\partial \theta} + (g_\theta(\tau) - v_\theta(S)) \frac{\partial v_\theta(S)}{\partial \theta} + \frac{\partial H(\pi_\theta(\cdot | S))}{\partial \theta} 
\]

(3.7)

where \( \eta \) refers to (\( \gamma, \lambda \)), \( \tau \) refer to trajectories and Eq. (3.7) combines actor loss, critic loss and entropy loss, which are commonly used in typical Actor-Critic (Mnih et al. (2016)) algorithms. Specifically, the meta parameters (\( \gamma, \lambda \)) corresponds to \( \phi \) in Eq. (3.2) . After the policy parameters \( \theta \) take one policy gradient update to become \( \theta'(\theta' = \theta + \alpha \nabla_\theta J^{\text{In}}(\theta, \phi)) \),
we can calculate the meta-gradient by backpropogating from $J_{\text{Out}}$ to meta parameters. In MGRL, $J_{\text{Out}}(\phi, \theta')$ degenerates to $J_{\text{Out}}(\theta')$. The meta-gradient can be shown as:

$$\nabla J(\phi) = \nabla_{\theta'} \nabla_{\phi'} J_{\text{Out}}(\theta'), \nabla_{\phi'} \theta' = \alpha \nabla_{\phi} \nabla_{\theta} J_{\text{In}}(\theta, \phi)$$ \hspace{1cm} (3.8)

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 Veeriah et al. (2019), Bonnet et al. (2021) have also shown that multi-step online meta-gradient can achieve better performance.

### 3.4 Meta-gradient based Inverse Design in RL

There exist many work like Oh et al. (2020), Zheng et al. (2020), Xu et al. (2020), Feng et al. (2021) that are trying to learn some fundamental/generalizable meta module across different environments such as a neural RL algorithm in Oh et al. (2020)(LPG). An important feature of meta-gradient based inverse design is that it inherently needs multi-step inner-loop to account for the eect 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:

$$J(\phi) = \mathbb{E}_{\tau \sim \rho(\tau)} \left[ \mathbb{E}_{\tau^k \sim p(\tau | \theta^k)} \left[ R \left( \tau^k \right) \right] \right]$$

with

$$\theta' = \theta^{t-1} + \alpha \nabla_{\theta^{t-1}} \mathbb{E}_{\tau \sim p(\tau | \theta^{t-1})} [f_{\phi}(\tau)]$$ \hspace{1cm} (3.9)

where $f_{\phi}(\tau)$ is the output of meta-network $\phi$ 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 Oh et al. (2020) for simplicity. In the scope of Eq. (3.1), $J^{\text{In/Out}}$ refers to the value function, $\theta$ represents the RL agent policy parameters and $\phi$ is the meta-parameter of neural RL algorithm. The formula can also be viewed as a multi-step extension from that in online meta-gradient reinforcement learning. Most of works are under a multi-task/environment (or a distribution over environment) and multi-lifetime setting. Xu et al. (2020) 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 Xu et al. (2020) is real large (1e9) and makes it become a special case of ‘multi-lifetime’ setting.

4 The Meta-gradient Estimation

In this section, we systematically discuss and analyze the bias and variance problems for meta-gradient estimation in current literature. We stress out two important problems in meta-gradient estimation: Plug-in compositional bias and Hessian estimation error. We will use the three Meta-RL algorithms presented in section 3 as examples but the analysis can generalise to all gradient based Meta-RL algorithms.

4.1 Plug-in Compositional Bias

Commonly, most gradient based Meta-RL literature get unbiased outer-loop gradient estimation (for estimating $\nabla_{\theta}J_{\phi}^{Out}(\phi, \theta^1)$ and $\nabla_{\phi}J_{\theta}^{Out}(\phi, \theta^1)$), and believe by plugging in unbiased inner-loop gradient estimation (for estimating $\nabla_{\phi}\theta^1$), the algorithm can get unbiased meta-gradient estimation. However, this is not true because of the compositional optimisation structure. We use one-step inner-loop meta-gradient as an example to illustrate the problem.

The expectation of meta-gradient is shown in Eq. (3.2). Note that the parameter $\theta^1$ is the expectation of conducting one-step policy gradient from $\theta$. In practice, we use samples to estimate all stochastic gradient. Assume we sample $\tau_1$ for inner-loop policy gradient estimate, conduct one-step gradient update to get $b\theta^1$, and finally sample $\tau_2$ for outer-loop policy gradient estimate, we have the following meta-gradient estimate:

$$\nabla \tilde{J}(\phi) = \nabla_{\phi}J_{\phi}^{Out}(\phi, \tilde{\theta}^1, \tau_1) + \nabla_{\phi}\tilde{\theta}^1 \nabla_{\tilde{\theta}^1}J_{\phi}^{Out}(\phi, \tilde{\theta}^1, \tau_1), \tilde{\theta}^1 = \theta + \alpha \nabla_{\theta}J_{\phi}^{In}(\theta, \phi, \tau_0)$$ (4.1)

where $\tilde{J}$ denotes the empirical estimate of corresponding variable. With unbiased first and second-order policy gradient estimator, we can get the expectation of the estimate above:

$$\mathbb{E}[\nabla \tilde{J}(\phi)] = \mathbb{E}_{\tilde{\theta}^1} \left[ \nabla_{\phi}J_{\phi}^{Out}(\phi, \tilde{\theta}^1) \right] + \nabla_{\phi}\tilde{\theta}^1 \mathbb{E}_{\tilde{\theta}^1} \left[ \nabla_{\tilde{\theta}^1}J_{\phi}^{Out}(\phi, \tilde{\theta}^1) \right]  \neq \nabla J(\phi)$$ (4.2)

Recall Eq. (3.2) and we can find out the plug-in estimation is biased. This estimation can
get unbiased inner and outer loop gradient but the outer loop gradient is evaluated at \( \hat{\theta}^1 \) rather than \( \theta^1 \). The bias comes from the fact that \( \nabla f(\mathbb{E}(x)) \neq \mathbb{E}_x(\nabla_x f(x)) \) unless \( f \) is a linear function for random variable \( x \). Because of the compositional structure of gradient based Meta-RL, typical meta-gradient estimation introduces estimation bias:

\[
\mathbb{E}[\nabla_{\hat{\theta}} J_{\text{Out}}^\theta (\phi, \hat{\theta}^1)] \neq \nabla_{\theta^1} J_{\text{Out}}^\theta (\phi, \theta^1), \quad \mathbb{E}[\nabla_{\phi} J_{\text{Out}}^\theta (\phi, \hat{\theta}^1)] \neq \nabla_{\phi} J_{\text{Out}}^\theta (\phi, \theta^1)
\] (4.3)

This is a fundamental problem in all gradient meta reinforcement learning algorithms because all policy gradient inner-loop update introduces estimation error caused by variance so \( \hat{\theta}^1 \neq \theta^1 \). The problem becomes more severe under multi-step formulation since each policy gradient step introduces estimation error and results in compositional bias.

### 4.2 One and Multi-step Hessian Estimation Bias

Many paper (Rothfuss et al. (2018), Liu et al. (2019), Tang et al. (2021)) tries to address the Hessian estimation in MAML. Rothfuss et al. (2018) theoretically and empirically validates that original MAML implementation can result in biased Hessian and meta-gradient estimation. However, we argue that beyond the scope of MAML-RL, many new proposed Meta-RL algorithms are still ignoring Hessian Estimation bias by directly calculating meta-gradient directly with automatic differentiation.

Here 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 \( \theta^1 \) and \( \nabla_{\theta} \theta^1 \)

\[
\theta^1 = \theta + \alpha \mathbb{E}_{\tau \sim \pi_\theta} [\nabla_\theta \log \pi (\tau | \theta) R(\tau)]
\] (4.4)

\[
\nabla_{\theta} \theta^1 = I + \mathbb{E}_{\tau \sim \pi_\theta} \left[ R(\tau) \left( \nabla_\theta^2 \log \pi_\theta(\tau) + \nabla_\theta \log \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau)^\top \right) \right]
\] (4.5)

Typically we need to use trajectory samples \( \tau_k \) to estimate the policy gradient. Denote the trajectory length as \( H \), we can get the adapted policy estimate.

\[
\hat{\theta}^1 = \theta + \frac{\alpha}{K} \sum_{k=1}^{M-1} \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta (a_t | s_t) \left( \sum_{t'=0}^{H} r(s_{t'}, a_{t'}) \right)
\] (4.6)
Finally, implementation of MAML-RL derives the gradient estimate by automatic differentiation. The corresponding estimation is biased:

\[
\mathbb{E}[\nabla_\theta \tilde{\theta}^1] = I + \alpha \mathbb{E}_{\tau \sim \pi_\theta} \left[ \frac{1}{K} \sum_{k=0}^{H-1} \sum_{t=0}^{H-1} \nabla_\theta^2 \log \pi_\theta (a_t | s_t) \left( \sum_{t'=0}^{H-1} r(s_{t'}, a_{t'}) \right) \right] = I + \alpha \mathbb{E}_{\tau \sim \pi_\theta} \left[ R(\tau) \nabla_\theta^2 \log \pi_\theta(\tau) \right] \neq \nabla_\theta \theta^1
\] (4.7)

The main reason of biased Hessian estimation is that automatic differentiation tools only consider the dependency of \( \theta \) in \( \nabla_\theta \log \pi_\theta \) while ignoring the dependency in expectation \( \mathbb{E}_{\tau \sim \pi_\theta} \).

In traditional practice, the \( \mathbb{E}_{\tau \sim \pi_\theta} \) is represented by trajectory sampling so the gradient term \( \nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta} \) is 0 using automatic differentiation. We need to add additional terms to further derive the gradient \( \nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta} \) brought by sampling dependency. Many new gradient based Meta-RL papers are still making the same biased estimation by using direct automatic differentiation. Existing works such as (Eq. (3) in Oh et al. (2020), Equ. (4) in Zheng et al. (2020), Eq. (12, 13, 14) in Xu et al. (2018) and Eq. (5, 6, 7) in Xu et al. (2020)) theoretically suffer from this problem. However, most of them are coincidentally unbiased if they only conduct one-step policy gradient update in the inner-loop. For one-step MGRL, this is true because it takes derivatives w.r.t meta parameters \( \phi = (\gamma, \lambda) \) which won’t have gradient dependency on \( \mathbb{E}_{\pi_\theta} \), rather than initial policy parameters. We can show that their meta-gradient estimation are unbiased as follows:

\[
\nabla_\phi \theta^1 = \alpha \mathbb{E}_{\tau \sim \pi_\theta} \left[ \nabla_\phi R(\tau, \phi) \nabla_\theta \log \pi_\theta(\tau) \right]
\] (4.8)

The corresponding estimate and the expectation of that are:

\[
\hat{\theta}^1 = \theta + \alpha \frac{1}{K} \sum_{k=0}^{H-1} \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta (a_t | s_t) R(\tau_k, \phi)
\] (4.9)

\[
\mathbb{E} \left[ \nabla_\phi \hat{\theta}^1 \right] = \alpha \mathbb{E}_{\tau \sim \pi_\theta} \left[ \nabla_\phi R(\tau, \phi) \nabla_\theta \log \pi_\theta(\tau) \right] = \nabla_\phi \theta^1
\] (4.10)

However, when they take more than one-step policy gradient update, the gradient estimation still gets the hessian estimation bias. For example, for MGRL methods that take two-step policy gradient update (\( \theta^2 = \theta^1 + \alpha \nabla_\theta J^{\text{In}}(\theta^1) \)) in the inner loop, the gradient term \( \nabla \theta^2(\phi) \)
of MGRL involves $\nabla_{\theta^1} \theta^2(\theta^1, \phi)$:

$$\nabla \theta^2(\phi) = \nabla_{\phi} \theta^2(\theta^1, \phi) + \nabla_{\phi} \theta^1 \nabla_{\theta^1} \theta^2(\theta^1, \phi)$$  \hspace{1cm} (4.11)

It corresponds exactly to the same structure of Eq. (4.7). So the same Hessian estimation bias happens here.

## 5 Theoretical Analysis

In this section, we establish some important properties of the meta-gradient which are useful for characterizing compositional bias and hessian estimation error in meta-gradient computation. Throughout the section we rely on following three assumptions,

**Assumption 5.1** (On objective functions).

The outer-loop objective function $J^{\text{Out}}$ satisfies

1. $J^{\text{Out}} (\cdot, \theta)$ and $J^{\text{Out}} (\phi, \cdot)$ are Lipschitz continuous with constants $m_\theta$ and $m_\phi$ respectively, $m_1 = \max \theta m_\theta$, $m_2 = \max \phi m_\phi$.

2. $\nabla_\phi J^{\text{Out}} (\cdot, \theta)$ and $\nabla_\theta J^{\text{Out}} (\phi, \cdot)$ are Lipschitz continuous with constants $\mu_\theta$ and $\mu_\phi$ respectively, $\mu_1 = \max \theta \mu_\theta$, $\mu_2 = \max \phi \mu_\phi$.

The inner-loop objective function $J^{\text{In}}$ satisfies

1. $\nabla_\theta J^{\text{In}} (\cdot, \theta)$ and $\nabla_\phi J^{\text{In}} (\phi, \cdot)$ are Lipschitz continuous with constants $c_\theta$ and $c_\phi$ respectively, $c_1 = \max \theta c_\theta$, $c_2 = \max \phi c_\phi$.

2. $\nabla_\phi \nabla_\theta J^{\text{In}} (\phi, \cdot)$ is Lipschitz continuous with constants $\lambda_\phi$, $\lambda_2 = \max \phi m_\phi$.

3. $\nabla_\theta^2 J^{\text{In}} (\phi, \cdot)$ is Lipschitz continuous with constants $\rho_\phi$, $\rho_2 = \max \phi m_\phi$.

**Assumption 5.2** (On bias of estimators).

1. Outer loop unbiased stochastic gradient estimator $\nabla_\phi J^{\text{Out}} (\phi, \theta, \tau)$ and $\nabla_\theta J^{\text{Out}} (\phi, \theta, \tau)$

2. Inner loop unbiased stochastic gradient estimator $\nabla_\theta J^{\text{In}} (\phi, \theta, \tau)$


**Assumption 5.3.** (On variance of estimators)

1. Outer loop stochastic gradient estimator $\nabla_\phi \hat{J}^{Out} (\phi, \theta, \tau)$ has bounded variance,
   \[ \mathbb{E}_\tau [\| \nabla_\phi \hat{J}^{Out} (\phi, \cdot, \tau) - \nabla_\phi J^{Out} (\phi, \cdot) \|^2] \leq (\sigma_1)^2 \]

2. Outer loop stochastic gradient estimator $\nabla_\theta \hat{J}^{Out} (\phi, \theta, \tau)$ has bounded variance,
   \[ \mathbb{E}_\tau [\| \nabla_\theta \hat{J}^{Out} (\phi, \cdot, \tau) - \nabla_\theta J^{Out} (\phi, \cdot) \|^2] \leq (\sigma_2)^2 \]

Note that Assumption 5.1 and 5.3 are standard assumptions mentioned in Fallah et al. (2020), also unbiasedness of first-order gradient estimators are made in 5.2 because this paper mainly focus on the situation where only bias of high-order gradient estimators exists.

## 5.1 Warm-up: One-step Inner Loop

We first give out a warm up regarding one-step inner-loop meta-gradient analysis. The following lemma characterizes the inner-loop adaptation error between $\theta^1_k$ and $\hat{\theta}^1_k$, which directly results in the compositional bias mentioned in Section 4.1

**Lemma 5.4** (Adaptation Error). Suppose that Assumptions 5.1 and 5.2 hold. Let $|\tau|$ denote number of trajectories used to estimate inner loop gradient, $\alpha$ the learning rate, then we have, at outer loop iteration $k \in \mathbb{N}$

\[ \mathbb{E} [\| \theta^1_k - \hat{\theta}^1_k \|] \leq \alpha \frac{\sigma_{In}^k}{\sqrt{|\tau|}} \tag{5.1} \]

where $\sigma_{In}^k = \sqrt{\mathbb{V} [\nabla_\theta \hat{J}^{In}(\phi_k, \theta^0_k, \tau)]}$.

**Proof.** See Appendix B.1 for a detailed proof. \qed

Lemma 5.4 implies that, given unbiased inner loop gradient estimator, the adaptation error is fully caused by estimation variance, or in specific, the policy gradient estimation variance. Due to this adaptation error, all of the outer loop stochastic gradients like $\nabla_\phi J^{Out}$ and $\nabla_\theta J^{Out}$ are evaluated at $\hat{\theta}^1_k$ instead of $\theta^1_k$.

Based on Lemma 5.4 and the discussion in Section 4.2, we can then theoretically derive upper bound on bias and variance of meta-gradient with one-step inner loop update.
**Theorem 5.5** (Bias of meta-gradient estimator). Suppose that Assumption 5.1 and 5.2 hold. Let $H_{\phi, \theta}$ denote $\nabla_\phi \nabla_\theta J^{\text{ln}}$. Then the following bounds hold at outer loop iteration $k \in \mathbb{N}$:

$$
\| \mathbb{E}[\nabla \hat{J}(\phi_k)] - \nabla J(\phi_k) \| \leq \alpha (\mu_1 + \alpha \mu_2 c_1) \frac{\sigma^{\text{In}}_k}{\sqrt{|\tau|}} + \alpha m_1 \underbrace{\hat{H}_k^H}_{\text{Compositional Bias}} + \underbrace{k \Delta H}_{\text{Hessian Estimation Bias}}
$$

where $\hat{H}_k^H = \| \mathbb{E}_{\tau_1} [\nabla_{\phi, \theta} (\phi_k, \theta_{k}^0, \tau_1)] - H_{\phi, \theta}(\phi_k, \theta_{k}^0) \|$.

**Proof.** See Appendix C.1 for a detailed proof. \qed

Theorem 5.5 shows that the upper bound of bias contains two parts: the first term indicates compositional bias caused by the adaptation error in Lemma 5.4, while the second term refers to the hessian estimation bias caused by direct automatic differentiation. All gradient based Meta-RL algorithm suffer from the first compositional bias while they have different Hessian estimation bias. The Infinitely Differentiable Monte Carlo Estimator (DiCE) used in LOLA can bring out unbiased hessian estimation: $\hat{H}_k^H = 0$. The Low-variance Curvature (LVC) estimator proposed by Rothfuss et al. (2018) introduce additional Hessian estimation term.

$$
\mathcal{H}_1 = \mathbb{E}_{\tau \sim p_{T(\tau|\theta)}} \left[ \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(a_t, s_t) \nabla_{\theta} \log \pi_{\theta}(a_t, s_t)^\top \left( \sum_{t=1}^{H-1} r(s_t, a_t, q_t) \right) \right]
$$

(5.3)

The following theorem derives the bound for the variance of meta-gradient estimator.

**Theorem 5.6** (Variance of meta-gradient estimator). Suppose that Assumptions 5.1, 5.2 and 5.3 hold. Set

$$
C_1 = 2 \sigma_1^2 + 6 \alpha \sigma_1^2 c_1^2 \sigma_2^2, \quad C_2 = 2 \alpha^2 (\mu_1 + \alpha \mu_2 c_1)^2, \quad C_3 = 2 \alpha^2 (m_1^2 + 3 \sigma_2^2)
$$

(5.4)
Then the following bounds hold at outer loop iteration $k \in \mathbb{N}$:

$$
\mathbf{V} \left[ \nabla \widehat{J}(\phi_k) \right] \leq C_1 + C_2 \frac{(\sigma^{In})^2}{|\tau|} + C_3 \left( (\widehat{\Delta}^H_k)^2 + (\sigma^H_k)^2 \right)
$$

(5.5)

where $(\sigma^H_k)^2 = \frac{\mathbf{V}[\nabla \phi \cdot \phi^{\text{In}}(\phi_k, \theta^t_k, \tau_t)]}{|\tau|}$.

**Proof.** See Appendix C.2 for a detailed proof. □

Theorem 5.6 implies that variance of meta-gradient estimator is bounded by weighted sum of squares of compositional bias, hessian estimation bias and variance. PROMP (Rothfuss et al. (2018)) achieves low meta-gradient variance because the LVC estimator can lower both $\widehat{\Delta}^H_k$ (compared with original MAML) and $\sigma^H_k$ (compared with DiCE). We can also know remedies that fix the meta-gradient estimation bias will additionally decrease the estimation variance.

### 5.2 Multi-step Inner Loop

We can now give $D$-step inner-loop meta-gradient analysis. The following lemma characterizes the multi-step inner-loop adaptation error, which directly results in the multi-step compositional bias.

**Lemma 5.7** ($D$-step Adaptation Error). Suppose that Assumptions 5.1 and 5.2 hold. Let $|\tau|$ denote number of trajectories used to estimate inner loop gradient in each inner loop update step, $\alpha$ the learning rate, then we have, at outer loop iteration $k \in \mathbb{N}$

$$
\mathbb{E}[\|\hat{\theta}_k^D - \theta_k^D\|] \leq ((1 + \alpha c_2)^D - 1) \frac{\sigma_{M-In}^k}{c_2 \sqrt{|\tau|}}
$$

(5.6)

where $\sigma_{M-In}^k = \max_t \sqrt{\mathbb{V}[\nabla \phi \cdot \phi^{\text{In}}(\phi_k, \theta^t_k, \tau_t)]}$, $t \in \{0, \ldots, D - 1\}$.

**Proof.** See Appendix B.2 for a detailed proof. □

**Theorem 5.8** ($D$-step Bias). Suppose that Assumptions 5.1 and 5.2 hold. Let $H_{\phi, \theta}$ denote $\nabla \phi \nabla J^{\text{In}}$, $H_{\theta, \theta}$ denote $\nabla^2 \theta J^{\text{In}}$. 


Set

\( B^D = \frac{((1 + \alpha c_2)^D - 1)}{D}\)  
\[ (5.7) \]

\( B_k^D = \frac{((1 + \alpha c_2^2 + \tilde{\Delta}_{k}^{H_{\theta,\varphi}})^D - 1)}{D} \)  
\[ \text{D-step Hessian Estimation Bias Scaling Factor} \]

\[ B_1 = \mu_1 + \alpha (c_1 m_2 \frac{\rho_2}{c_2} + c_1 \mu_2 + m_2 a_2), ~ B_2 = \alpha m_2, ~ B_3 = \alpha^2 c_1 m_2 \]  
\[ (5.8) \]

Then the following bounds hold at outer loop iteration \( k \in \mathbb{N} \):

\[ \mathbb{E}[\nabla J^D(\phi_k)] - \nabla J^D(\phi_k) \leq B_1 B_k^D B_D^\frac{\sigma_k^{ln}}{c_2 \sqrt{\tau}} + B_2 B_k^D \tilde{\Delta}_{k}^{H_{\theta,\varphi}} + B_3 (B_k^D - B^D) \]  
\[ (5.9) \]

where \( \tilde{\Delta}_{k,\varphi}^{H_{\theta,\varphi}} = \mathbb{E}_{\tau_1}[\tilde{H}_{\phi,\varphi}(\phi_k, \tilde{\theta}_k^t, \tau_1)] - H_{\phi,\varphi}(\phi_k, \tilde{\theta}_k^t) \), \( \tilde{\Delta}_{k}^{H_{\theta,\varphi}} = \max_j \tilde{\Delta}_{k,\varphi}^{H_{\theta,\varphi}} \), same with \( \tilde{\Delta}_{k,\varphi}^{H_{\theta,\varphi}} \).

\[ \text{Compositional Bias} \]

\[ \text{Hessian Estimation Bias} \]

**Proof.** See Appendix C.3 for a detailed proof.

Theorem 5.8 shows that the upper bound of \( D \)-step bias contains three parts: the first term indicates multi-step compositional bias caused by the adaptation error in Lemma 5.7, while the second term refers to the multi-step hessian estimation bias i.e. caused by direct automatic differentiation, the last term specifically refers to multi step hessian estimation bias \( \tilde{\Delta}_{k,\varphi}^{H_{\theta,\varphi}} \) term during inner update, which has a scaling effect on upper bound of first and second term.

**Theorem 5.9 (\( D \)-step Variance).** Suppose that Assumption 5.1, 5.2 and 5.3 hold. Set

\[ V^D = \frac{(((1 + \alpha c_2)^2 + \alpha^2 c_2^2)^D - 1)}{D}\]  
\[ \text{D-step Scaling Factor} \]

\[ V_k^D = \frac{(((1 + \alpha c_2^2 + \tilde{\Delta}_{k}^{H_{\theta,\varphi}})^2)^D - 1)}{D} \]  
\[ \text{D-step Hessian Estimation Bias Scaling Factor} \]

\[ H_k^D = (((1 + \alpha c_2^2 + (\tilde{\Delta}_{k}^{H_{\theta,\varphi}})^2 + (\sigma_k^{H_{\theta,\varphi}})^2)^D - 1) \]
\[ V_1 = 2\sigma_1^2, \ V_2 = 4D(2\mu_2^2 + 2\alpha^2\left(c_1m_2\frac{D^2}{c_2} + c_1\mu_2 + m_2\lambda_2\right)^2), \ V_3 = 6D\alpha^2(12m_2^2 + 12\sigma_2^2), \]
\[ V_4 = 2D\alpha^2(m_2^2 + 3\sigma_2^2), \ V_5 = 2D\alpha^4c_1^2m_2^2, \]

Then the following bounds hold at outer loop iteration \( k \in \mathbb{N} \):
\[
\mathbb{V}\left[ \nabla J^D(\phi_k) \right] \\
\leq V_1 + V_2V_k^D \left( \frac{\sigma_{In}^2}{|\tau|} \right)^2 + \left( V_3 \left( H_k^D - V_k^D \right) + V_4 \right) \left( c_1^2 + (\tilde{\Delta}_k^{H,\phi})^2 + (\sigma_k^{H,\phi})^2 \right) + V_5 \left( V_k^D - V_k^D \right)
\]

where \( \tilde{\Delta}_k^{H,\phi} = \|E_{\pi}[\tilde{H}_{\phi,\theta}(\phi_k, \tilde{\theta}_k, \tau_1)] - H_{\phi,\theta}(\phi_k, \tilde{\theta}_k)\|, \tilde{\Delta}_k^{H,\phi} = \max_j \tilde{\Delta}_{k,j}^{H,\phi}, \) same with \( \tilde{\Delta}_{k,j}^{H,\phi} \).

Proof. See Appendix C.4 for a detailed proof.

Theorem 5.9 shows that the upper bound of \( D \)-step variance contains three parts: the first part indicates multi-step compositional bias, while the second part refers to sum of the hessian estimation bias term \( \tilde{\Delta}_k^{H,\phi} \) and variance term \( \sigma_k^{H,\phi} \), the last part specifically refers to multi-step hessian estimation bias \( \tilde{\Delta}_k^{H,\phi} \) and variance \( \sigma_k^{H,\phi} \) during inner update, which has a scaling effect on upper bound of first and second term.

6 Mitigations for Meta-gradient Bias

We offer two remedies to handle these two estimation bias we argue respectively.

6.1 Off-policy Corrections

From the Lemma 5.4 and the discussion in Section 4.1, we know that the fundamental problem comes from the variance of policy gradient estimator. The directest way for handling compositional bias is to decrease the policy gradient estimation variance.

Many methods have been proposed to decrease the policy gradient estimation variance like value function approximation (Sutton et al. (2000)), GAE (Schulman et al. (2015)) or
off-policy learning (Degris et al. (2012)). We borrow the insight from off-policy learning to propose off-policy correction to handle the compositional bias problem. Combing off-policy learning with specific any-order policy gradient estimator, we can conduct off-policy correction to lower the policy gradient estimation variance. For instance, by combing DiCE and off-policy learning, we have off-policy DiCE (two-agent version):

\begin{equation}
J_{OFF-DICE} = \sum_{t=0}^{H-1} \left( \prod_{t'=0}^{t} \frac{\pi_{\phi_1} (a_{t'}^1 | s_{t'}^1) \pi_{\phi_2} (a_{t'}^2 | s_{t'}^2)}{\mu_1 (a_{t'}^1 | s_{t'}^1) \mu_2 (a_{t'}^2 | s_{t'}^2)} \right) r(s_t, a_t) \tag{6.1}
\end{equation}

We can also combine it with the LVC estimator and have:

\begin{equation}
J_{OFF-LVC} = \sum_{t=0}^{H-1} \frac{\pi_{\phi_1} (a_t^1 | s_t^1) \pi_{\phi_2} (a_t^2 | s_t^2)}{\mu_1 (a_t^1 | s_t^1) \mu_2 (a_t^2 | s_t^2)} \left( \sum_{t'=t}^{H-1} r(s_{t'}, a_{t'}) \right) \tag{6.2}
\end{equation}

where \( \phi_1, \phi_2 \) refer to the current policy, \( \mu_1, \mu_2 \) refer the behaviour policy for agent 1 and agent 2 respectively. \( H \) is the trajectory length and \( r \) refers to the reward for agent 1.

Note that though our main purpose is to lower the first order policy gradient estimation variance based on off-policy learning, the off-policy any-order gradient estimator can theoretically lower the variance of Hessian estimation, which can further increase the quality of meta-gradient estimation. Coincidentally, Tang et al. (2021) has shown some similar equations by off-policy evaluation framework and the OFF-DICE corresponds exactly to the step-wise importance sampling estimate. But our purposes are completely different. Tang et al. (2021) utilised the off-policy framework to help interpret prior work on higher-order derivative estimation while we are really using it for variance reduction to decrease the compositional bias.

### 6.2 Mutli-Step Hessian Estimator Correction

From the theoretical analysis in Section 5.2, we can know that hessian estimation bias can significantly increase meta-gradient estimation bias in multi-step inner-loop setting. For instance, existing work under the topic we mentioned in Section 3.3 and 3.4, are in fact biased. To lower the bias, we can use low-biased hessian estimator other than direct automatic differentiation and apply them on each-step inner-loop actor loss. Most prior work proposed
substitution on low-biased hessian estimation like DiCE in Foerster et al. (2018b), Loaded DiCE in Farquhar et al. (2019), Low Variance Curvature(LVC) in Rothfuss et al. (2018) and TayPO-\(K\) in Tang et al. (2021).

Here we show an example of applying the LVC method in multi-step MGRL algorithm. In the original MGRL, the inner-loop update follows:

\[
- \frac{\partial J(\tau, \theta, \eta)}{\partial \theta} = (g_\eta(\tau) - v_\theta(S)) \frac{\partial \log \pi_\theta(A \mid S)}{\partial \theta} + b (g_\eta(\tau) - v_\theta(S)) \frac{\partial v_\theta(S)}{\partial \theta} + c \frac{\partial H(\pi_\theta(\cdot \mid S))}{\partial \theta}
\]  
(6.3)

By using the LVC operator, we have the following equation:

\[
- \frac{\partial J(\tau, \theta, \eta)}{\partial \theta} = (g_\eta(\tau) - v_\theta(S)) \nabla_\theta \pi_\theta(A \mid S) + b (g_\eta(\tau) - v_\theta(S)) \frac{\partial v_\theta(S)}{\partial \theta} + c \frac{\partial H(\pi_\theta(\cdot \mid S))}{\partial \theta}
\]  
(6.4)

As shown in Rothfuss et al. (2018), the LVC operator will ensure an unbiased first-order policy gradient and low-biased/low-variance second-order policy gradient. So this operator only corrects the meta-gradient update and leave the inner-loop gradient estimation formula the same.

7 Experiments

We now carry out several empirical studies to complement the framework developed above. Firstly, we conduct ablation experiment over sample size/learning rate and step size on tabular MDP to empirically investigate the property of meta-gradient estimation with different settings of estimators, and it also helps us to verify the bias term mentioned in Section 4.1 and 4.2. Our second experiment is taken over LOLA algorithm, where we empirically verify how the relationship between meta-gradient estimation and performance, and we also validate and the effect of off-policy correction (Section 6.1). Finally, we conduct experiment on Atrai games to show the effect brought by offering additional sampling correction term (Section 6.2) on the MGRL algorithm.
7.1 Experiment on Tabular MDP

Firstly, we conduct experiments on the setting of Tabular MDP to quantify the effect from different factors to the meta-gradient estimation. We adopt the tabular random MDP setting presented in Tang et al. (2021). The dimension is 10 for state space and 5 for action space, so we have the reward matrix $R \in \mathbb{R}^{10 \times 5}$. The policy is a logits matrix $\theta^0 \in \mathbb{R}^{10 \times 5}$. The final policy is obtained by adopting softmax activation on this logits matrix. Different from the experiments on gradient/Hessian estimation in Tang et al. (2021), we mainly focus on the final meta-gradient estimation performance. We utilise component-wise correlation and the variance as the metrics to evaluate the performance of estimation. The correlation is calculated between oracle meta-gradient and estimated meta-gradient.

In the first setting, we adopt the setting of MAML-RL and estimate the meta-gradient $\nabla_{\theta^0} J^\text{Out}(\theta^k)$, where $\theta^k$ is obtained by conducting $k$ policy gradient steps from $\theta^0$: $\theta^i = \theta^{i-1} + \alpha \nabla J^\text{In}(\theta^{i-1}), i \in \{1, 2...k\}$. In this setting, $J^\text{In}$ and $J^\text{Out}$ refer to the same thing: value function $\mathbb{V}_\theta(s_0)$ over the initial state $s_0$. 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: $\theta' = \nabla \theta'_c + \theta'_h$, where $\nabla$ is the ”stop gradient” operator. By this implementation trick, we can have the following property: $\theta' \rightarrow \theta'_c$ and $\nabla \theta \theta' = \nabla \theta \theta'_c$, where $\rightarrow$ is the ”evaluates to” operator. ”Evaluates to” operator $\rightarrow$ is in contrast with $=$, which also brings the equality of gradients. By ”Evaluates to” operator, the ”stop gradient” operator means that $\nabla (f_\theta(x)) \rightarrow f_\theta(x)$ but $\nabla \theta \nabla (f_\theta(x)) \rightarrow 0$. 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 $\nabla_{\theta^k} J^\text{Out}(\pi^k)$ estimation, which can also help us understand the effect of outer-loop policy gradient over final meta-gradient.

since we have three estimators and also can calculate the analytic value of each gradient term, we run the following experiments by trying all 8 permutations over these three estimators.
For instance, when using estimated inner-loop policy gradient to get $\tilde{\theta}_i$, $i \in \{1, 2...k\}$, estimated hessian $\nabla_{\theta_i}^2 J_{\text{In}}(\theta_i)$, $i \in \{1, 2...k\}$ and exact outer-loop policy gradient $\nabla_{\theta^k} J^{\text{Out}}(\tilde{\theta}^i)$, we denote the setting as $\text{ESE}$, where $E$ refers to exact solution, $S$ refers to stochastic estimation. We use $\nabla_{\theta_i}^2 J_{\text{In}}(\theta_i)$ and $\hat{\theta}^i$ to emphasize these two terms are calculated by estimation while $\nabla_{\theta^k} J^{\text{Out}}(\tilde{\theta}^k)$ is the exact gradient on $\tilde{\theta}^k$. We use 4 different estimator in the following experiment: DiCE (Foerster et al. (2018b)), Loaded-DiCE (Farquhar et al. (2019)), Low variance curvature (LVC in Rothfuss et al. (2018)) and the biased estimator used in original MAML.

Figure 1: 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 biased-MAML when the Hessian matrix is estimated.

**Ablation study on sample size and estimator.** The first ablation study is taken upon sample size and different estimators and the result is shown in fig. (1). This experiment is taken under 1-step inner-loop setting. We can get several conclusions from the figure. Firstly, 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. Secondly, with accurate outer-loop policy gradient, the correction
of Hessian estimation error and compositional bias help increase the correlation and decrease the variance ($\text{SEE} > \text{ESE} > \text{SSE}$). With estimated outer-gradient, the correction of these two terms also helps ($\text{EES} > \text{SES} > \text{ESS} > \text{SSS}$). In this one-step setting, the correction of Hessian estimation error is more crucial than the compositional bias.

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 Williams (1992) 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 MAML biased estimator performs worse than LVC and Loaded-DiCE when the Hessian is estimated ($\text{SSE, ESE, SSS, ESS}$). This corresponds to the conclusion in Rothfuss et al. (2018) that the LVC estimator introduces low-bias and low-variance Hessian estimation while original MAML estimator has large-bias and low-variance 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 ($\text{ESE} = \text{SEE} > \text{SSE}$), while the Hessian correction is still important in Loaded-DiCE ($\text{SEE} > \text{ESE} > \text{SSE}$).

Ablation study on inner learning rate, step and estimator. We also conduct ablation study on inner learning rate and number of steps shown in fig. (2) and fig. (3). The results show that: (1) 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 $\text{SSE}$ in all estimators). Also in multi-step and large learning rate setting, the importances of Hessian estimation and compositional bias become comparable in LVC and Loaded-DiCE ($\text{SEE} \approx \text{ESE}, \text{SES} \approx \text{ESS}$). (2) The Hessian estimation problem becomes especially severe for MAML estimator in multi-step or larger learning rate setting ($\text{SSE, ESE}$ decreases a lot).
Figure 2: 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 MAML largely decreases the correlation.

Figure 3: Ablation study on inner step and estimator. Results of larger steps show similar phenomenon with larger inner-loop learning rate.

In the second setting, we follow the algorithm of intrinsic reward generator presented in
Zheng et al. (2018). In tabular MDP, we have an additional meta intrinsic reward matrix $\phi \in \mathbb{R}^{10 \times 5}$. Starting from $\theta^0$, the inner-loop process takes policy gradient based on the new reward matrix $R_{\text{new}} = R + \phi$: $\theta^i = \theta^{i-1} + \alpha \nabla J^\text{In}(\theta^{i-1}, \phi), i \in \{1, 2\ldots k\}$. The meta-gradient estimation of the intrinsic reward matrix $\nabla_{\phi} J^\text{Out}(\theta^k)$ is needed in this case. Note that in the outer loss we use the original reward matrix $R$ instead of $R_{\text{new}}$ so the outer loss is $J^\text{Out}(\theta^k)$ rather than $J^\text{Out}(\theta^k, \phi)$. Compared with the first setting, this setting has different learning objectives: the intrinsic reward matrix. The object of meta-update (intrinsic matrix) and the object of inner-update (policy parameters) are different now and it can help us identify the problem mentioned in Section 4.2.

**Ablation study on inner step, estimator and sample size.** In this case, we choose the LVC and original MAML-biased estimator. We conduct ablation study on inner-step and sample size shown in fig. (4). With more sample size and less step size, the correlation increases and the variance decreases for both estimator. Two important features are: (1) With 1-step inner-loop setting, both estimator performs similarly in the correlation and variance. (2) With multi-step inner-loop setting, LVC based estimator can still reach relatively high correlation and low variance while MAML-biased estimator directly reaches low correlation even with 2-step inner-loop. The phenomenon shown here corresponds exactly to the Hessian estimation issue we discuss in Section 4.2 and verify the importance of unbiased or low-biased Hessian estimator in multi-step meta gradient estimation.
Figure 4: Experiment result of Tabular MDP. The LVC estimation can maintain relatively great correlation with step size larger than 2, while the MAML-biased decreases to 0 easily.

### 7.2 Experiment on LOLA

In this subsection, we conduct experiments in LOLA algorithm. We conduct 3 experiments to understand: (1) the effect brought by different inner/outer estimation. (2) the effect brought by Hessian estimation error and compositional bias (3) Off-policy LOLA and ablation study, where we utilise the remedies of off-policy correction we mention in Section 5.1.

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. For each experiment, we report both the average reward (upper side) and the correlation (lower side) between estimated meta-gradient and the oracle expected meta-gradient. To mainly analyze the problem brought by inner-loop update, we recalculate the meta-gradient by resetting the

---

\(^*\)https://github.com/alexis-jacq/LOLA_DiCE
outer-loop gradient in analytic solution when visualising the correlation.

Figure 5: 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 average return.

**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 figure 5(a). Here the inner-loop estimation refers to $\nabla_\theta J^{\text{In}}(\theta, \phi)$ while outer-loop estimation refers to $\nabla_\phi J^{\text{Out}}(\phi, \theta')$ and $\nabla_\theta J^{\text{Out}}(\phi, \theta')$. 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 analytic solution of policy gradient instead of estimation. The return and correlation shown in figure 5(a) reveals us three things: (1) The inner-loop gradient estimation plays an important role for making LOLA algorithm in RL setting work - default batch size 128 leads to fail while batch size 1024 leads to work. (2) The outer-loop gradient estimation is also quite crucial and the performance is really well for LOLA-Exact. (3) 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. The outer-loop first-order policy gradient estimation is a classical topic in RL and in this paper we mainly focus on the particular problem in Meta-RL brought by inner-loop gradient estimation. Thus, we focus on point (1) and further conduct ablation study on inner-loop gradient update to understand it.

Ablation on LOLA-DiCE Hessian variance and compositional bias. Since the unbiased DiCE estimator is used in LOLA-DiCE algorithm, there does not exist any Hessian estimation bias in the LOLA algorithm. Thus, we mainly discuss the problem of compositional bias and Hessian estimation variance brought by inner-loop update in figure 5(b,c). To decompose the effect brought by different factors, we also utilise the inner-loop implementation trick mentioned in Section 6.1. The naming convention of the legend is similar with the previous one. For instance, comp_128_hessian_1024 means that we use 128 batch size for estimator I to get $\theta'_c$ and 1024 batch size for estimator II to get $\theta'_h$. Fig. 5(b) shows us the ablation study about Hessian variance over the LOLA algorithm. The correlation and reward plot shows us that the hessian variance is the main reason of estimation error and by reducing the hessian variance either by adding batch size or using analytic solution can largely increase the gradient correlation and performance. Fig. 5(c) shows us the ablation study about compositional bias over the LOLA algorithm. The compositional bias do influence the algorithm performance and the algorithm can gain performance by bias correction. An interesting thing in Fig. 5(c) is that we find out the gradient correlation of these three settings are comparable. An possible explanation is that the main problem here is the hessian variance that 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 We use the off-policy correction we mention in Section 6.1 to conduct the inner-loop gradient update and maintain the outer-loop gradient estimation with on-policy estimation. Note that the off-policy DiCE here can not only lower the compositional bias by lowering the first-order policy gradient error, but also is helpful for
lowering the higher-order policy gradient error theoretically. So we also conduct two ablation study by the same implementation trick to decompose the inner update with different settings: (1) off-policy (both are estimated in off-policy) (2) On-policy composition estimation with off-policy hessian estimation (3) Off-policy composition estimation with off-policy hessian estimation. The result is shown at Fig. 5(d). For the reward plot, all three off-policy settings achieve comparable final reward (around -1.2) and gain large improvement over on-policy baseline. The off-policy and on_comp_off_hessian setting achieve higher efficiency. For the correlation plot. For the gradient correlation, the relationship is: off-policy = on_comp_off_ > off_comp_on_hessian > on_policy. Off-policy DiCE can increase the LOLA performance by both lowering compositional bias and hessian variance. The correlation gain for off-policy comp&on-policy hessian is still limited like that in Fig. 5(c). But the performance gain verifies the bonus brought by correcting compositional bias.

7.3 Experiment on MGRL

Finally, we conduct experiment over meta-gradient Reinforcement Learning algorithm. We conduct experiment on the 3 environments of Atari 2600 video games. Note that there exists some differences between our multi-step MGRL and that in Bonnet et al. (2021). They use the ”dispatch” strategy in which they conduct multiple virtual inner-loop update just for meta update, while our experiments just follow the real RL update as the training goes. So unlike Bonnet et al. (2021), we can conduct fair comparison between MGRL and baseline algorithm under the same sample complexity. In all, we compare 4 variants of algorithm: (1) Baseline Advantage Actor-critic(A2C) algorithm (Mnih et al. (2016)) (2) one-step MGRL + A2C (3) 5-step MGRL + A2C. (4)5-step MGRL + A2C + LVC correction.

From the fig. (6) we can know two things: Firstly, multi-step MGRL always perform better than 1-step MGRL. Even with fewer meta updates, the multi-step MGRL performs better than the baseline and one-step MGRL, which is also somehow compatible with the result in Bonnet et al. (2021). Secondly, the LVC correction upon multi-step MGRL can increase the performance on the BeamRider environment and behave comparably on the rest
two environments, which validates the effectiveness of this LVC correction.

Figure 6: Experiment results on MGRL. The LVC correction can achieve at least as well as original estimation.

8 Conclusion

In this paper, we introduce a unifying framework for meta-gradient estimation in gradient-based Meta-RL algorithms. Based on the framework, we successfully show that many existing Meta-RL works are suffering from biased meta-gradient estimation because of the compositional bias and Hessian estimation bias problem. Empirical results over tabular MDP, LOLA-DiCE and MGRL verify our theoretical analysis and the effectiveness of the correction methods. We believe our work can offer intuitions for more work over meta-gradient estimation problem in Meta-RL.

References

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

Barbey, A. K. (2018). Network neuroscience theory of human intelligence. Trends in cognitive sciences, 22 8–20.
Baydin, A. G., Cornish, R., Rubio, D. M., Schmidt, M. and Wood, F. (2017). Online learning rate adaptation with hypergradient descent. *arXiv preprint arXiv:1703.04782*.

Bechtle, S., Molchanov, A., Chebotar, Y., Grefenstette, E., Righetti, L., Sukhatme, G. and Meier, F. (2021). Meta learning via learned loss. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE.

Bonnet, C., Caron, P., Barrett, T., Davies, I. and Laterre, A. (2021). One step at a time: Pros and cons of multi-step meta-gradient reinforcement learning. *arXiv preprint arXiv:2111.00206*.

Degris, T., White, M. and Sutton, R. S. (2012). Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*.

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

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

Farquhar, G., Whiteson, S. and Foerster, J. (2019). Loaded dice: trading off bias and variance in any-order score function estimators for reinforcement learning. *arXiv preprint arXiv:1909.10549*.

Feng, X., Slumbers, O., Wan, Z., Liu, B., McAleer, S., Wen, Y., Wang, J. and Yang, Y. (2021). Neural auto-curricula in two-player zero-sum games. *Advances in Neural Information Processing Systems, 34*.

Finn, C., Abbeel, P. and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR.
Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P. and Mordatch, I. (2018a). Learning with opponent-learning awareness. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems.

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

Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P. and Mordatch, I. (2017). Learning with opponent-learning awareness. arXiv preprint arXiv:1709.04326.

Franceschi, L., Donini, M., Frasconi, P. and Pontil, M. (2017). Forward and reverse gradient-based hyperparameter optimization. In International Conference on Machine Learning. PMLR.

Fu, H., Tang, H., Hao, J., Chen, C., Feng, X., Li, D. and Liu, W. (2020). Towards effective context for meta-reinforcement learning: an approach based on contrastive learning. arXiv preprint arXiv:2009.13891.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P. and Levine, S. (2018). Meta-reinforcement learning of structured exploration strategies. Advances in Neural Information Processing Systems, 31 5302–5311.

Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning. PMLR.

Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F. and Fan, C. (2020). Learning to utilize shaping rewards: A new approach of reward shaping. Advances in Neural Information Processing Systems, 33.

Kim, D.-K., Liu, M., Riemer, M., Sun, C., Abdulhai, M., Habibi, G., Lopez-Cot, S., Tesauro, G. and How, J. P. (2020). A policy gradient algorithm for learning to learn in multiagent reinforcement learning. arXiv preprint arXiv:2011.00382.
Kirsch, L., van Steenkiste, S. and Schmidhuber, J. (2019). Improving generalization in meta-reinforcement learning using learned objectives. In *International Conference on Learning Representations*.

Letcher, A., Foerster, J., Balduzzi, D., Rocktäschel, T. and Whiteson, S. (2018). Stable opponent shaping in differentiable games. In *International Conference on Learning Representations*.

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

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

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

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR.

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

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

Rakelly, K., Zhou, A., Finn, C., Levine, S. and Quillen, D. (2019). Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*. PMLR.
Rothfuss, J., Lee, D., Clavera, I., Asfour, T. and Abbeel, P. (2018). Promp: Proximal meta-policy search. \textit{arXiv preprint arXiv:1810.06784}.

Schmidhuber, J. (1987). \textit{Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook}. Ph.D. thesis, Technische Universität München.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T. et al. (2020). Mastering atari, go, chess and shogi by planning with a learned model. \textit{Nature}, \textbf{588} 604–609.

Schulman, J., Moritz, P., Levine, S., Jordan, M. and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. \textit{arXiv preprint arXiv:1506.02438}.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017). Proximal policy optimization algorithms. \textit{arXiv preprint arXiv:1707.06347}.

Sutton, R. (2019). The bitter lesson. \texttt{http://www.incompleteideas.net/IncIdeas/BitterLesson.html}.

Sutton, R. S., McAllester, D. A., Singh, S. P. and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In \textit{Advances in neural information processing systems}.

Tang, Y. (2021). Biased gradient estimate with drastic variance reduction for meta reinforcement learning. \textit{arXiv preprint arXiv:2112.07328}.

Tang, Y., Kozuno, T., Rowland, M., Munos, R. and Valko, M. (2021). Unifying gradient estimators for meta-reinforcement learning via off-policy evaluation. \textit{Advances in Neural Information Processing Systems}, \textbf{34}.

Tsividis, P. A., Pouny, T., Xu, J. L., Tenenbaum, J. B. and Gershman, S. J. (2017). Human learning in atari. In \textit{2017 AAAI Spring Symposium Series}. 36
Veeriah, V., Hessel, M., Xu, Z., Lewis, R., Rajendran, J., Oh, J., van Hasselt, H., Silver, D. and Singh, S. (2019). Discovery of useful questions as auxiliary tasks. *arXiv preprint arXiv:1909.04607*.

Veeriah, V., Zahavy, T., Hessel, M., Xu, Z., Oh, J., Kemaev, I., van Hasselt, H., Silver, D. and Singh, S. (2021). Discovery of options via meta-learned subgoals. *arXiv preprint arXiv:2102.06741*.

Vuorio, R., Beck, J. A., Farquhar, G., Foerster, J. N. and Whiteson, S. (2021). No DICE: An investigation of the bias-variance tradeoff in meta-gradients. In *Deep RL Workshop NeurIPS 2021*.

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

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8 229–256.

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

Xu, Z., van Hasselt, H. and Silver, D. (2018). Meta-gradient reinforcement learning. *arXiv preprint arXiv:1805.09801*.

Zahavy, T., Xu, Z., Veeriah, V., Hessel, M., Oh, J., van Hasselt, H., Silver, D. and Singh, S. (2020). A self-tuning actor-critic algorithm. *arXiv preprint arXiv:2002.12928*.

Zheng, Z., Oh, J., Hessel, M., Xu, Z., Kroiss, M., Van Hasselt, H., Silver, D. and Singh, S. (2020). What can learned intrinsic rewards capture? In *International Conference on Machine Learning*. PMLR.

Zheng, Z., Oh, J. and Singh, S. (2018). On learning intrinsic rewards for policy gradient methods. *Advances in Neural Information Processing Systems*, 31 4644–4654.
Zhou, W., Li, Y., Yang, Y., Wang, H. and Hospedales, T. (2020). Online meta-critic learning for off-policy actor-critic methods. *Advances in Neural Information Processing Systems, 33.*
A Discussion of 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 Vuorio et al. (2021) argues that: (1) the general unbiased meta gradient for MAML-RL (Finn et al. (2017)) and Online Meta Gradient (Xu et al. (2018), Zheng et al. (2018)) should be the K-sample inner-loop meta gradient shown in E-MAML Al-Shedivat et al. (2017) rather than the expected policy gradient inner-loop meta gradient used in many recent work (Rothfuss et al. (2018), Liu et al. (2019), Tang et al. (2021)). (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 Vuorio et al. (2021) proposes to use one coefficient to control. The EPG can achieve lower variance estimation and perform better empirically Rothfuss et al. (2018) (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 Vuorio et al. (2021) 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 $\gamma$ or intrinsic model rather than multiple meta modules as a real schedule needs. Also, recently there exist one work Bonnet et al. (2021) 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. We leave
these two 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.

B Proofs of Lemmas in Section 5

B.1 Proof of Lemma 5.4

Proof. Based on the iterative updates that

\[ \theta_k^1 = \theta_k^0 - \alpha \nabla J^{In} \left( \phi_k, \theta_k^0 \right) \]  

(B.1)

\[ \widehat{\theta}_k^1 = \theta_k^0 - \alpha \nabla \widehat{J}^{In} \left( \phi_k, \theta_k^0, \tau \right) \]  

(B.2)

\[ \mathbb{E}_\tau [||\widehat{\theta}_k^1 - \theta_k^1||] = \sqrt{\mathbb{E}_\tau [||\widehat{\theta}_k^1 - \theta_k^1||^2]} \leq \sqrt{\mathbb{E}_\tau [||\theta_k^1 - \theta_k^0||^2]} = \alpha \sqrt{\frac{\nabla \left[ \nabla \widehat{J}^{In} \left( \phi_k, \theta_k^0, \tau \right) \right]}{\tau}} \]  

(B.3)

which concludes the proof of Lemma 5.4. \qed

B.2 Proof of Lemma 5.7

Proof. Based on the iterative updates

\[ \theta_k^{i+1} = \theta_k^i + \alpha \nabla J^{In} \left( \phi_k, \theta_k^i \right), \; t = 0, \ldots, D - 1 \]  

(B.4)

\[ \widehat{\theta}_k^{i+1} = \widehat{\theta}_k^i + \alpha \nabla \widehat{J}^{In} \left( \phi_k, \widehat{\theta}_k^i, \tau_k^i \right), \; \widehat{\theta}_k^0 = \theta_k^0, \; t = 0, \ldots, D - 1 \]  

(B.5)
Iteratively, we can get

\[
\mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| \right]
\]

\[
= \mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| + \alpha \nabla_{\theta} J_{\text{In}} (\phi_k, \theta^D_k) - \alpha \nabla_{\theta} J_{\text{In}} (\phi_k, \theta^D_k, \tau_0) \right] \]
\]

\[
\leq \mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| \right] + \alpha \mathbf{E}_{\tau_0:D-1} \left[ \left\| \nabla_{\theta} J_{\text{In}} (\phi_k, \theta^D_k) - \nabla_{\theta} J_{\text{In}} (\phi_k, \theta^D_k, \tau_0) \right\| \right] + \alpha \mathbf{E}_{\tau_0:D-1} \left[ \left\| \nabla_{\theta} J_{\text{In}} (\phi_k, \theta^D_k, \tau_0) \right\| - \nabla_{\theta} J_{\text{In}} (\phi_k, \bar{\theta}^D_k, \tau_0) \right] \]
\]

\[
\leq (1 + \alpha c_2) \mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| \right] + \alpha \mathbf{E}_{\tau_0:D-1} \left[ \left\| \nabla_{\theta} J_{\text{In}} (\phi_k, \bar{\theta}^D_k, \tau_0) \right\| - \nabla_{\theta} J_{\text{In}} (\phi_k, \theta^D_k, \tau_0) \right] \]
\]

\[
\leq (1 + \alpha c_2) \mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| \right] + \alpha \mathbf{E}_{\tau_0:D-2} \left[ \left\| \nabla_{\theta} J_{\text{In}} (\phi_k, \bar{\theta}^D_k, \tau_0) \right\| - \nabla_{\theta} J_{\text{In}} (\phi_k, \theta^D_k, \tau_0) \right] \]
\]

(B.6)

Let \( \sigma^\text{M-In}_k = \max_t \sqrt{\nabla [\nabla_{\theta} J_{\text{In}} (\phi_k, \bar{\theta}^t_k, \tau_0^t)]}, |\tau| = |\tau_0^t|, t \in \{0, \ldots, D - 1\} \)

\[
\mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| \right] \leq (1 + \alpha c_2) \mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| \right] + \alpha \frac{\sigma^\text{M-In}_k}{\sqrt{|\tau|}} \quad \text{(B.7)}
\]

Iteratively, we can get

\[
\mathbf{E}_{\tau_0:D-1} \left[ \left\| \theta^D_k - \theta_k \right\| \right] \leq (1 + \ldots + (1 + \alpha c_2)^{D-1}) \alpha \frac{\sigma^\text{M-In}_k}{\sqrt{|\tau|}} \quad \text{(B.8)}
\]

\[
= (1 + \alpha c_2)^{D-1} \frac{\sigma^\text{M-In}_k}{c_2 \sqrt{|\tau|}}
\]
which concludes the proof of Lemma 5.7.

\[ \square \]

\section*{C Proof of Theorems in Section 5}

\subsection*{C.1 Proof of Theorem 5.5}

\textit{Proof.} In one-step scenario, meta-gradient estimator takes the form

\[ \nabla \hat{J}(\phi_k) = \nabla \phi J^{\text{Out}} \left( \phi_k, \tilde{\theta}^1_k, \tau_2 \right) - \alpha \hat{H}_{\phi, \theta} \left( \phi_k, \theta^0_k, \tau_1 \right) \nabla \theta J^{\text{Out}} \left( \phi_k, \tilde{\theta}^1_k, \tau_2 \right) \]  

(C.1)

\[ \tilde{\theta}^1_k = \theta^0_k - \alpha \nabla \theta \hat{J}^{\text{In}}(\phi_k, \theta^0_k, \tau_0) \]  

(C.2)

exact meta-gradient takes the form

\[ \nabla J(\phi_k) = \nabla \phi J^{\text{Out}} \left( \phi_k, \theta^1_k \right) - \alpha H_{\phi, \theta} \left( \phi_k, \theta^0_k \right) \nabla \theta J^{\text{Out}} \left( \phi_k, \theta^1_k \right) \]  

(C.3)

\[ \theta^1_k = \theta^0_k - \alpha \nabla \theta J^{\text{In}} \left( \phi_k, \theta^0_k \right) \]  

(C.4)

\[ \| \mathbb{E}_{\tau_0, \tau_1, \tau_2} [\nabla \hat{J}(\phi_k)] - \nabla J(\phi_k) \| \]

\[ \leq \mathbb{E}_{\tau_0} \left[ \left\| \mathbb{E}_{\tau_1} \left[ H_{\phi, \theta} \left( \phi_k, \theta^0_k \right) \nabla \theta J^{\text{Out}} \left( \phi_k, \tilde{\theta}^1_k, \tau_2 \right) \right] \right\| \right] - \mathbb{E}_{\tau_0} \left[ \left\| \nabla \phi J^{\text{Out}} \left( \phi_k, \tilde{\theta}^1_k, \tau_2 \right) \right\| \right] + \alpha \mathbb{E}_{\tau_0} \left[ \left\| \nabla \theta J^{\text{Out}} \left( \phi_k, \tilde{\theta}^1_k, \tau_2 \right) \right\| \right] - \mathbb{E}_{\tau_0} \left[ \left\| \nabla \phi J^{\text{Out}} \left( \phi_k, \theta^1_k \right) \right\| \right] \]

\[ \leq \mathbb{E}_{\tau_0} \left[ \left\| \nabla \phi J^{\text{Out}} \left( \phi_k, \tilde{\theta}^1_k \right) - \nabla \phi J^{\text{Out}} \left( \phi_k, \theta^1_k \right) \right\| \right] + \alpha \mathbb{E}_{\tau_0} \left[ \left\| H_{\phi, \theta} \left( \phi_k, \theta^0_k \right) \right\| \right] \left\| \nabla \theta J^{\text{Out}} \left( \phi_k, \tilde{\theta}^1_k \right) - \nabla \theta J^{\text{Out}} \left( \phi_k, \theta^1_k \right) \right\| \]

\[ \leq \mu_1 \mathbb{E}_{\tau_0} \left[ \left\| \tilde{\theta}^1_k - \theta^1_k \right\| \right] + \alpha \mu_2 c_1 \mathbb{E}_{\tau_0} \left[ \left\| \tilde{\theta}^1_k - \theta_k \right\| \right] + \alpha m_1 \mathbb{E}_{\tau_1} \left[ \left\| \hat{H}_{\phi, \theta} \left( \phi_k, \theta^0_k, \tau_1 \right) - H_{\phi, \theta} \left( \phi_k, \theta^0_k \right) \right\| \right] \]

\[ \leq \mu_1 \mathbb{E}_{\tau_0} \left[ \left\| \tilde{\theta}^1_k - \theta^1_k \right\| \right] + \alpha \mu_2 c_1 \mathbb{E}_{\tau_0} \left[ \left\| \tilde{\theta}^1_k - \theta_k \right\| \right] + \alpha m_1 \mathbb{E}_{\tau_1} \left[ \left\| \hat{H}_{\phi, \theta} \left( \phi_k, \theta^0_k, \tau_1 \right) - H_{\phi, \theta} \left( \phi_k, \theta^0_k \right) \right\| \right] \]

(C.5)

\[ \text{Let } \hat{\Delta}^H_k = \| \mathbb{E}_{\tau_1} [\hat{H}_{\phi, \theta}(\phi_k, \theta^0_k, \tau_1)] - H_{\phi, \theta}(\phi_k, \theta^0_k) \| \]
\[ \| E_{\tau_0, \tau_2} [\nabla \widehat{J}(\phi_k)] - \nabla J(\phi_k) \| \leq \alpha (\mu_1 + \alpha \mu_2 c_1) \frac{\sigma_{ln}^k}{|\tau|} + \alpha m_1 \widehat{\Delta}_k^H \] (C.6)

which concludes the proof of Theorem 5.5. \(\square\)

C.2 Proof of Theorem 5.6

Proof. According to the law of total variance

\[ \nabla J(\phi_k) = \nabla \widehat{J}(\phi_k) | \tau_0 \] + \[ \nabla \widehat{J}(\phi_k) | \tau_0 \] (C.7)

We upper bound terms (i)-(ii) in (C.7) respectively, that is,

Term (i).

\[ \nabla \left[ E_{\tau_1, \tau_2} \left[ \nabla \widehat{J}(\phi_k) | \tau_0 \right] \right] \]

\[ = E_{\tau_0} \left[ \left\| E_{\tau_1, \tau_2} \left[ \nabla \widehat{J}(\phi_k) | \tau_0 \right] - E_{\tau_0, \tau_1, \tau_2} \left[ \nabla \widehat{J}(\phi_k) \right] \right\|^2 \right] \] (C.8)

According to Theorem 1.

\[ \left\| \nabla J(\phi_k) - E_{\tau_0, \tau_1, \tau_2} [\nabla \widehat{J}(\phi_k)] \right\| \leq E_{\tau_0} \left[ \left\| E_{\tau_1, \tau_2} \left[ \nabla \widehat{J}(\phi_k) | \tau_0 \right] - \nabla J(\phi_k) \right\| \right] \] (C.9)

\[ \left\| E_{\tau_1, \tau_2} \left[ \nabla \widehat{J}(\phi_k) | \tau_0 \right] - \nabla J(\phi_k) \right\| \leq \alpha (\mu_1 + \alpha \mu_2 c_1) \left\| \theta_k^1 - \theta_k^1 \right\| + \alpha m_1 \widehat{\Delta}_k^H \]

\[ \nabla \left[ E_{\tau_0} \left[ \nabla \widehat{J}(\phi_k) | \tau_0 \right] \right] \leq 2\alpha^2 (\mu_1 + \alpha \mu_2 c_1)^2 \frac{\nabla [\nabla \widehat{J}(\phi_k, \theta_k^0, \tau)]}{|\tau|} + 2\alpha^2 m_1^2 (\widehat{\Delta}_k^H)^2 \] (C.10)

Term (ii).
According to supporting Lemma,

\[
\|\hat{H}_{\phi,\theta} (\phi_k, \theta_0^1, \tau_1)\|^2 \leq 3c_1^2 + 3 \left( (\hat{\Delta}^H_k)^2 + (\hat{\sigma}_k^H)^2 \right)
\]  

(C.12)

\[
\mathbb{E}_{\tau_0} \left[ \mathbf{V} \left[ \nabla \bar{J}^\text{Out} (\phi_k) \mid \tau_0 \right] \right] \leq 2\sigma_1^2 + 2\alpha^2 \left( 3c_1^2 + 3 \left( (\hat{\Delta}^H_k)^2 + (\hat{\sigma}_k^H)^2 \right) \right) \sigma_2^2 + m_1^2 (\hat{\sigma}_k^H)^2
\]

(C.13)

\[
\leq 2\sigma_1^2 + 2\alpha^2 \left( m_2^2 + 3\sigma_2^2 (\hat{\sigma}_k^H)^2 + (3\sigma_2^2 c_1^2 + 3\sigma_2^2 (\hat{\Delta}^H_k)^2) \right)
\]
Then combine terms (i)-(ii) together, that is
\[
\nabla \mathbb{E}_{\tau_0} \left[ \mathbb{E}_{\tau_1} \left[ \nabla \tilde{J}(\phi_k) \mid \tau_1 \right] \right] + \mathbb{E}_{\tau_0} \left[ \mathbb{E}_{\tau_1} \left[ \nabla \tilde{J}(\phi_k) \mid \tau_1 \right] \right]
\]
\[
\leq 2\alpha^2 (\mu_1 + \alpha \mu_2 c_1)^2 \frac{(\sigma_{ln})^2}{|\tau|} + 2\alpha^2 m_1^2 (\Delta_k^H)^2
\]
\[
+ 2\alpha^2 (m_1^2 + 3\sigma_2^2) (\sigma_k^H)^2 + (3\sigma_2^2 c_1^2 + 3\sigma_2^2 (\Delta_k^H)^2)
\]
\[
\leq 2\alpha^2 + 6\alpha^2 c_1^2 \sigma_2^2
\]
\[
+ 2\alpha^2 (\mu_1 + \alpha \mu_2 c_1)^2 \frac{(\sigma_{ln})^2}{|\tau|}
\]
\[
+ 2\alpha^2 (m_1^2 + 3\sigma_2^2) (\sigma_k^H)^2 + (\Delta_k^H)^2)
\]
which concludes the proof of Theorem 5.6. \hfill \Box

### C.3 Proof of Theorem 5.8

**Proof.** According Lemma D.1, exact meta-gradient \( \nabla J^D(\phi_k) \) takes the form
\[
\nabla J^\text{Out} \left( \phi_k, \theta^j_k \right) = \alpha \sum_{t=0}^{D-1} H_{\phi,\theta} \left( \phi_k, \theta^j_k \right) \prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta,\theta} \left( \phi_k, \theta^j_k \right) \right) \nabla J^\text{Out} \left( \phi_k, \theta^D_k \right)
\]
(C.15)

where
\[
\theta^{t+1} = \theta^t + \alpha \nabla J^\text{In} \left( \phi_k, \theta^j_k \right), \ t = 0, \ldots, D - 1
\]
(C.16)

Accordingly, \( D \)-step meta-gradient estimator \( \nabla \tilde{J}^D(\phi_k) \) takes the form
\[
\nabla \tilde{J}^\text{Out} \left( \phi_k, \tilde{\theta}^D_k \right) = \alpha \sum_{t=0}^{D-1} \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_t \right) \prod_{j=t+1}^{D-1} \left( I - \alpha \tilde{H}_{\theta,\theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_t \right) \right) \nabla \tilde{J}^\text{Out} \left( \phi_k, \tilde{\theta}^D_k, \tau^3 \right)
\]
(C.17)

where
\[
\tilde{\theta}^{t+1} = \tilde{\theta}^t + \alpha \nabla \tilde{J}^\text{In} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_t \right), \ \tilde{\theta}^0 = \theta^D_k, \ t = 0, \ldots, D - 1
\]
(C.18)

Hence the expectation of meta-gradient estimator takes the form
\[
\mathbb{E}_{\tau_{0:D-1}, \tau_{1:D-1}, \tau_{3}} \left[ \nabla \tilde{J}^D (\phi_k) \right] = \\
\mathbb{E}_{\tau_{0:D-1}} \left[ \mathbb{E}_{\tau_{3}} \left[ \nabla \tilde{J}^{\text{Out}} (\phi_k, \hat{\theta}_k^D, \tau_3) \mid \tau_{0:D-1} \right] \right] - \\
\alpha \sum_{t=0}^{D-1} \mathbb{E}_{\tau_{t+1}^t} \left[ \tilde{H}_{\theta,\theta} \left( \phi_k, \hat{\theta}_k^t \mid \tau_{0:t-1} \right) \right] \times \\
\prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_{j+1}} \left[ I - \alpha \tilde{H}_{\theta,\theta} \left( \phi_k, \hat{\theta}_k^j \mid \tau_{0:j-1} \right) \right] \times \\
\mathbb{E}_{\tau_{3}} \left[ \nabla \tilde{J}^{\text{Out}} (\phi_k, \hat{\theta}_k^D, \tau_3) \mid \tau_{0:D-1} \right] - \\
\nabla \tilde{J}^{\text{Out}} (\phi_k, \hat{\theta}_k^D) + \alpha \sum_{t=0}^{D-1} H_{\phi,\theta} \left( \phi_k, \hat{\theta}_k^t \right) \prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta,\theta} \left( \phi_k, \hat{\theta}_k^j \right) \right) \nabla \tilde{J}^{\text{Out}} (\phi_k, \hat{\theta}_k^D),
\]
\[
\leq \mathbb{E}_{0:D-1} \left[ \left\| \mathbb{E}_{\tau_3} \left[ \nabla_\phi J^\text{Out} (\phi_k, \tilde{\theta}_D, \tau_3) \mid \tau_0^{0:D-1} \right] - \nabla_\phi J^\text{Out} (\phi_k, \theta_D^j) \right\| \right]
+ \mathbb{E}_{0:D-1} \left[ \left\| \alpha \sum_{t=0}^{D-1} \mathbb{E}_{\tau_1} \left[ \tilde{H}_{\phi, \theta} (\phi_k, \tilde{\theta}_k^j, \tau_1) \mid \tau_0^{0:t-1} \right] \times \right. \right.
\prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2} \left[ I - \alpha \tilde{H}_{\theta, \theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \mid \tau_0^{0:t-1} \right] \times \mathbb{E}_{\tau_3} \left[ \nabla_\theta J^\text{Out} (\phi_k, \tilde{\theta}_D^j, \tau_3) \mid \tau_0^{0:D-1} \right] -
\alpha \sum_{t=0}^{D-1} H_{\phi, \theta} J^\text{In} (\phi_k, \theta^j_k) \prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta, \theta} (\phi_k, \theta^j_k) \right) \nabla_\theta J^\text{Out} (\phi_k, \theta_D^j) \right\| \right]
\leq \mathbb{E}_{0:D-1} \left[ \left\| \nabla_\phi J^\text{Out} (\phi_k, \tilde{\theta}_D^j) - \nabla_\phi J^\text{Out} (\phi_k, \theta_D^j) \right\| \mid \tau_0^{0:D-1} \right]
+ \mathbb{E}_{0:D-1} \left[ \left\| \alpha \sum_{t=0}^{D-1} \mathbb{E}_{\tau_1} \left[ \tilde{H}_{\phi, \theta} (\phi_k, \tilde{\theta}_k^j, \tau_1) \mid \tau_0^{0:t-1} \right] \times \right. \right.
\prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2} \left[ I - \alpha \tilde{H}_{\theta, \theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \mid \tau_0^{0:t-1} \right] \times \mathbb{E}_{\tau_3} \left[ \nabla_\theta J^\text{Out} (\phi_k, \tilde{\theta}_D^j, \tau_3) \mid \tau_0^{0:D-1} \right] -
\alpha \sum_{t=0}^{D-1} H_{\phi, \theta} (\phi_k, \theta^j_k) \prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta, \theta} (\phi_k, \theta^j_k) \right) \nabla_\theta J^\text{Out} (\phi_k, \theta_D^j) \right\| \right]
\leq \mu_1 \mathbb{E}_{0:D-1} \left[ \left\| \tilde{\theta}_D^D - \theta_D^D \right\| \mid \tau_0^{0:D-1} \right]
+ \mathbb{E}_{0:D-1} \left[ \left\| \alpha \sum_{t=0}^{D-1} \mathbb{E}_{\tau_1} \left[ \tilde{H}_{\phi, \theta} (\phi_k, \tilde{\theta}_k^j, \tau_1) \mid \tau_0^{0:t-1} \right] \times \right. \right.
\prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2} \left[ I - \alpha \tilde{H}_{\theta, \theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \mid \tau_0^{0:t-1} \right] \times \mathbb{E}_{\tau_3} \left[ \nabla_\theta J^\text{Out} (\phi_k, \tilde{\theta}_D^j, \tau_3) \mid \tau_0^{0:D-1} \right] -
\alpha \sum_{t=0}^{D-1} H_{\phi, \theta} (\phi_k, \theta^j_k) \prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta, \theta} (\phi_k, \theta^j_k) \right) \nabla_\theta J^\text{Out} (\phi_k, \theta_D^j) \right\| \right]
\]
\[ \leq \mu_1 \mathbb{E}_{\tau_0^{0:D-1}} \left[ \left\| \widehat{\theta}_k^D - \theta_k^D \right\| | \tau_0^{0:D-1} \right] \\
+ a \sum_{t=0}^{D-1} \mathbb{E}_{\tau_0^{0:D-1}} \left[ \left\| \mathbb{E}_{\tau_1^t} [\widehat{H}_{\theta,\theta} (\phi_k, \widehat{\theta}_k^t, \tau_1^t) | \tau_0^{0:t-1}] \times \\
\prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2^j} [I - \alpha H_{\theta,\theta} (\phi_k, \widehat{\theta}_k^j, \tau_2^j) | \tau_0^{0:t-1}] \times \mathbb{E}_{\tau_3^j} [\nabla_{\theta} J^\text{Out} (\phi_k, \widehat{\theta}_k^D, \tau_3) | \tau_0^{0:D-1}] - \\
H_{\phi,\theta} (\phi_k, \theta_k^D) \right\| ight] \times \mathbb{E}_{\tau_0^{0:t-1}} \left[ \left\| \mathbb{E}_{\tau_1^t} [\widehat{H}_{\phi,\theta} (\phi_k, \widehat{\theta}_k^t, \tau_1^t) | \tau_0^{0:t-1}] - H_{\phi,\theta} (\phi_k, \theta_k^t) \right\| \right] \times \\
\prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta,\theta} (\phi_k, \theta_k^j) \right) \nabla_{\theta} J^\text{Out} (\phi_k, \theta_k^D) \times ||H_{\phi,\theta} (\phi_k, \theta_k^t) || + \]

\[ \mathbb{E}_{\tau_0^{0:D-1}} \left[ \left\| \mathbb{E}_{\tau_1^t} [\widehat{H}_{\phi,\theta} (\phi_k, \widehat{\theta}_k^t, \tau_1^t) | \tau_0^{0:t-1}] - H_{\phi,\theta} (\phi_k, \theta_k^t) \right\| \right] \times \\
\prod_{j=t+1}^{D-1} \left( I - \widehat{H}_{\theta,\theta} (\phi_k, \widehat{\theta}_k^j, \tau_2^j) \right) \nabla_{\theta} J^\text{Out} (\phi_k, \widehat{\theta}_k^D, \tau_3) \times \mathbb{E}_{\tau_0^{0:D-1}} \left[ \right] \left[ \right] \right] \]

\text{(C.23)}

\[ \leq \mu_1 \mathbb{E}_{\tau_0^{0:D-1}} \left[ \left\| \widehat{\theta}_k^D - \theta_k^D \right\| | \tau_0^{0:D-1} \right] \\
+ a \sum_{t=0}^{D-1} \mathbb{E}_{\tau_0^{0:D-1}} \left[ \left\| \mathbb{E}_{\tau_1^t} [\widehat{H}_{\phi,\theta} (\phi_k, \widehat{\theta}_k^t, \tau_1^t) | \tau_0^{0:t-1}] \times \mathbb{E}_{\tau_3^j} [\nabla_{\theta} J^\text{Out} (\phi_k, \widehat{\theta}_k^D, \tau_3) | \tau_0^{0:D-1}] - \\
H_{\phi,\theta} (\phi_k, \theta_k^D) \right\| \right] \times \mathbb{E}_{\tau_0^{0:t-1}} \left[ \left\| \mathbb{E}_{\tau_1^t} [\widehat{H}_{\phi,\theta} (\phi_k, \widehat{\theta}_k^t, \tau_1^t) | \tau_0^{0:t-1}] - H_{\phi,\theta} (\phi_k, \theta_k^t) \right\| \right] \times \\
\prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta,\theta} (\phi_k, \theta_k^j) \right) \nabla_{\theta} J^\text{Out} (\phi_k, \theta_k^D) \times ||H_{\phi,\theta} (\phi_k, \theta_k^t) || + \]

\[ \mathbb{E}_{\tau_0^{0:D-1}} \left[ \left\| \mathbb{E}_{\tau_1^t} [\widehat{H}_{\phi,\theta} (\phi_k, \widehat{\theta}_k^t, \tau_1^t) | \tau_0^{0:t-1}] - H_{\phi,\theta} (\phi_k, \theta_k^t) \right\| \right] \times \\
\prod_{j=t+1}^{D-1} \left( I - \widehat{H}_{\theta,\theta} (\phi_k, \widehat{\theta}_k^j, \tau_2^j) \right) \nabla_{\theta} J^\text{Out} (\phi_k, \widehat{\theta}_k^D, \tau_3) \times \mathbb{E}_{\tau_0^{0:D-1}} \left[ \right] \left[ \right] \right] \]

\text{(C.24)}
\[
\leq \mu_1 E_{0:D-1}^0 \left[ \left\| \theta^D_k - \theta^D_k \right\| \mid \tau^D_0 \right] \\
+ \alpha \sum_{i=0}^{D-1} \left[ \sum_{i=0}^{D-1} \left[ \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] - \sum_{j=0}^{D-1} \left( I - \alpha H_{\theta_i, \theta} \left( \phi_k, \theta^i_k \right) \right) \right] \times \left\| \nabla \theta J^{\text{Out}} \left( \phi_k, \theta^D_k \right) \right\| + E_{0:D-1}^0 \left[ \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \times \left\| \nabla \theta J^{\text{Out}} \left( \phi_k, \theta^D_k \right) \right\| \times \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \times \left\| \nabla \theta J^{\text{Out}} \left( \phi_k, \theta^D_k \right) \right\| \times \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right\| \right) \\
+ \mu_2 E_{0:D-1}^0 \left[ \left\| \tilde{H}_{\phi_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] + E_{0:D-1}^0 \left[ \left\| \tilde{H}_{\phi_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \times \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \right) \\
+ m_2 \left[ \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \right] \right) \\
(C.25)
\]

\[
\leq \mu_1 E_{0:D-1}^0 \left[ \left\| \theta^D_k - \theta^D_k \right\| \right] \\
+ \alpha \sum_{i=0}^{D-1} \left[ m_2 E_{0:D-1}^0 \left[ \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \right] \right) \\
+ \mu_2 E_{0:D-1}^0 \left[ \left\| \tilde{H}_{\phi_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \times \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \right) \\
+ \left[ \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \right) \times \\
\leq \mu_1 E_{0:D-1}^0 \left[ \left\| \theta^D_k - \theta^D_k \right\| \right] \\
+ \mu_2 E_{0:D-1}^0 \left[ \left\| \tilde{H}_{\phi_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \times \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \right) \\
+ m_2 \left[ \left\| \sum_{j=0}^{D-1} \left[ I - \alpha \tilde{H}_{\theta_i, \theta} \left( \phi_k, \tilde{\theta}^j_k, \tau^j_2 \right) \mid \tau^D_0 \right] \right] \right) \\
(C.26)
\]
\[
\begin{align*}
\leq & \mu_1 E_{\tau_0:D-1} \left[ \left\| \hat{\theta}_k^D - \theta_k^D \right\| \right] \\
+ & \alpha \sum_{t=0}^{D-1} c_1 m_2 E_{\tau_0:D-1} \left[ \prod_{j=t+1}^{D-1} E_{\tau_2} \left[ I - \alpha \widehat{H}_{\theta,\theta} (\phi_k, \hat{\theta}_k^j, \tau_1^j) \mid \tau_0^{0:t-1} \right] - \prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta,\theta} (\phi_k, \theta_k^j) \right) \right] \\
+ & c_1 \mu_2 E_{\tau_0:D-1} \left[ \left\| \hat{\theta}_k^D - \theta_k^D \right\| \right] + m_2 E_{\tau_0:D-1} \left[ \left\| E_{\tau_2} \left[ \widehat{H}_{\phi,\theta} (\phi_k, \hat{\theta}_k^j, \tau_1^j) \mid \tau_0^{0:t-1} \right] - H_{\phi,\theta} (\phi_k, \theta_k^j) \right\| \right] \\
& \left( C.27 \right)
\end{align*}
\]

We upper bound terms (i)-(ii) in Eq. (C.27) respectively, that is,

**Term (i).** According to the supporting Lemma D.3

\[
\begin{align*}
E_{\tau_0:D-1} \left[ \prod_{j=t+1}^{D-1} E_{\tau_2} \left[ I - \alpha \widehat{H}_{\theta,\theta} (\phi_k, \hat{\theta}_k^j, \tau_1^j) \mid \tau_0^{0:t-1} \right] - \prod_{j=t+1}^{D-1} \left( I - \alpha H_{\theta,\theta} (\phi_k, \theta_k^j) \right) \right] \\
\leq & \alpha \left[ (1 + \alpha c_2 + \alpha \Delta_k^{H_{\theta,\theta}})^{D-t-1} - (1 + \alpha c_2)^{D-t-1} \right] + \frac{\rho_2}{\alpha c_2} \left( (1 + \alpha c_2 + \alpha \Delta_k^{H_{\theta,\theta}})^{D-t-1} - 1 \right) \left( 1 + \alpha c_2 \right)^{D-1} \sigma_k^{M-In} \\
& \quad \left( C.28 \right)
\end{align*}
\]

**Term (ii).**

\[
\begin{align*}
E_{\tau_0:D-1} \left[ E_{\tau_2} \left[ \widehat{H}_{\phi,\theta} (\phi_k, \hat{\theta}_k^j, \tau_1^j) \mid \tau_0^{0:t-1} \right] - H_{\phi,\theta} (\phi_k, \theta_k^j) \right] \\
\leq & E_{\tau_0:D-1} \left[ E_{\tau_2} \left[ \widehat{H}_{\phi,\theta} (\phi_k, \hat{\theta}_k^j, \tau_1^j) \mid \tau_0^{0:t-1} \right] - H_{\phi,\theta} (\phi_k, \theta_k^j) \right] + \left( C.29 \right)
\end{align*}
\]
Let $\tilde{\Delta}^{\mathcal{H}_{\phi,\theta}}_{k,j} = \left\| \mathbb{E}_{\tau_1} \left[ \tilde{H}_{\phi,\theta}(\phi_k, \tilde{\theta}_k^j, \tau_1^j) \right] - H_{\phi,\theta}(\phi_k, \tilde{\theta}_k^j) \right\|$, $\tilde{\Delta}^{\mathcal{H}_{\phi,\theta}}_k = \max_j \tilde{\Delta}^{\mathcal{H}_{\phi,\theta}}_{k,j}$, $j \in \{0, \ldots, D - 1\}$

\[
\mathbb{E}^{0:D-1}_{\tau_0} \left[ \left\| \mathbb{E}_{\tau_1} \left[ \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_1^j \right) \mid \tau_0^{0:t-1} \right] - H_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^j \right) \right\| \right] \\
\leq \mathbb{E}^{0:D-1}_{\tau_0} \left[ \tilde{\Delta}^{\mathcal{H}_{\phi,\theta}}_{k,j} \right] + \lambda_2 \mathbb{E}^{0:D-1}_{\tau_0} \left[ \left\| \tilde{\theta}_k^j - \theta_k^j \right\| \right]
\]

(C.30)

Term (iii).

\[
\mathbb{E}^{0:D-1}_{\tau_0} \left[ \prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2} \left[ I - \alpha \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \mid \tau_0^{0:t-1} \right] \right] \\
\leq \mathbb{E}^{0:D-1}_{\tau_0} \left[ \prod_{j=t+1}^{D-1} \left( I + \alpha \left\| \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \mid \tau_0^{0:t-1} \right] - \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^j \right) \right] \right] \\
+ \alpha \left\| \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^j \right) \right\|
\]

(C.31)

Let $\tilde{\Delta}^{H_{\theta,\theta}}_{k,j} = \left\| \mathbb{E}_{\tau_2} \left[ \tilde{H}_{\phi,\theta}(\phi_k, \tilde{\theta}_k^j, \tau_2^j) \right] - H_{\phi,\theta}(\phi_k, \tilde{\theta}_k^j) \right\|$, $\tilde{\Delta}^{H_{\theta,\theta}}_k = \max_j \tilde{\Delta}^{H_{\theta,\theta}}_{k,j}$, $j \in \{0, \ldots, D - 1\}$

\[
\mathbb{E}^{0:D-1}_{\tau_0} \left[ \prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2} \left[ I - \alpha \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \mid \tau_0^{0:t-1} \right] \right] \\
\leq \mathbb{E}^{0:D-1}_{\tau_0} \left[ \prod_{j=t+1}^{D-1} \left( 1 + \alpha c_2 + \alpha \tilde{\Delta}^{H_{\theta,\theta}}_{k,j} \right) \mid \tau_0^{0:t-1} \right] \\
\leq (1 + \alpha c_2 + \alpha \tilde{\Delta}^{H_{\theta,\theta}}_k)^{D-t-1}
\]

Then combine terms (i)-(iii) together, that is
\[ \| \mathbb{E}_{\tau_0:D-1, \tau_1^{0:D-1}, \tau_2^{0:D-1}, \tau_3} [\nabla J^D(\phi_k)] - \nabla J^D(\phi_k) \| \]
\[ \leq \mu_1 ((1 + \alpha c_2)^D - 1) \frac{\sigma_{M-In}^k}{c_2 \sqrt{|\tau|}} \]
\[ + \alpha \sum_{t=0}^{D-1} \alpha c_1 m_2 \left( (1 + \alpha c_2 + \Delta H_{k}^{\theta, \theta})_t - (1 + \alpha c_2)^D - 1 \right) \]
\[ + c_1 m_2 \frac{p_2}{c_2} ((1 + \alpha c_2 + \Delta H_{k}^{\theta, \theta})_t - 1)(1 + \alpha c_2)^D - 1) \frac{\sigma_{M-In}^k}{c_2 \sqrt{|\tau|}} \]
\[ + (1 + \alpha c_2 + \Delta H_{k}^{\theta, \theta})_t - 1 \left( c_1 \mu_2 ((1 + \alpha c_2)^D - 1) \frac{\sigma_{M-In}^k}{c_2 \sqrt{|\tau|}} + m_2 \Delta H_{k}^{\theta, \theta} + m_2 \lambda_2 ((1 + \alpha c_2)^D - 1) \frac{\sigma_{M-In}^k}{c_2 \sqrt{|\tau|}} \right) \]
\[ \leq (\mu_1 + \alpha (c_1 m_2 \frac{p_2}{c_2} + c_1 \mu_2 + m_2 \lambda_2)) ((1 + \alpha c_2)^D - 1)(1 + \alpha c_2 + \Delta H_{k}^{\theta, \theta})_t - 1 \frac{\sigma_{M-In}^k}{c_2 \sqrt{|\tau|}} \]
\[ + (am_2)((1 + \alpha c_2 + \Delta H_{k}^{\theta, \theta})_t - 1) \Delta H_{k}^{\theta, \theta} \]
\[ + (\alpha^2 c_1 m_2) (((1 + \alpha c_2 + \Delta H_{k}^{\theta, \theta})_t - 1) - ((1 + \alpha c_2)^D - 1)) \]

(C.33)

\[ \| \mathbb{E}_{\tau_0:D-1, \tau_1^{0:D-1}, \tau_2^{0:D-1}, \tau_3} [\nabla J^D(\phi_k)] - \nabla J^D(\phi_k) \| \]
\[ \leq B_1 B^D_1 B^D_2 \frac{\sigma_{I^n}^k}{c_2 \sqrt{|\tau|}} + B_2 B^D_1 \Delta H_{k}^{\theta, \theta} + B_3 (B^D_1 - B^D_2) \]

(C.34)

which concludes the proof of Theorem 5.8.

\[ \square \]

C.4 Proof of Theorem 5.9

Proof. According to the law of total variance

\[ \mathbb{V} \left[ \nabla J^D(\phi_k) \right] = \mathbb{V} \left[ \mathbb{E}_{\tau_1^{0:D-1}, \tau_2^{0:D-1}, \tau_3} \left[ \nabla J^D(\phi_k) \mid \tau_0^{0:D-1} \right] \right] + \mathbb{E}_{\tau_0^{0:D-1}} \left[ \mathbb{V} \left[ \nabla J^D(\phi_k) \mid \tau_0^{0:D-1} \right] \right] \]

(C.35)

We upper bound terms (i)-(ii) in (C.35) respectively, that is,
According to Theorem 3.

\[
\mathbb{V} \left[ \mathbb{E}_{\tau_1^{0:D-1}, \tau_2^{0:D-1}, \tau_3^{0:D-1}} \left[ \nabla J^D (\phi_k) \mid \tau_0^{0:D-1} \right] \right] \\
= \mathbb{E}_{\tau_1^{0:D-1}} \left[ \mathbb{E}_{\tau_2^{0:D-1}, \tau_3^{0:D-1}} \left[ \nabla J^D (\phi_k) \mid \tau_0^{0:D-1} \right] - \mathbb{E}_{\tau_1^{0:D-1}} \left[ \nabla J^D (\phi_k) \right] \right]^2 \\
\leq \mathbb{E}_{\tau_1^{0:D-1}} \left[ \mathbb{E}_{\tau_2^{0:D-1}, \tau_3^{0:D-1}} \left[ \nabla J^D (\phi_k) \mid \tau_0^{0:D-1} \right] - \nabla J^D (\phi_k) \right]^2 \\
\leq \mu_1 \left( (1 + \alpha c_2)^D - 1 \right) \frac{\sigma^M_{\text{In}}}{c_2\sqrt{|\tau|}} \\
+ \alpha \sum_{i=0}^{D-1} \alpha c_1 m_2 \left[ (1 + \alpha c_2 + \Delta_k^{H_{\theta, \varphi}})^{D-i-1} - (1 + \alpha c_2)^{D-i-1} \right] \\
+ \frac{\rho^2}{c_2} \left( (1 + \alpha c_2 + \Delta_k^{H_{\theta, \varphi}})^{D-i-1} - 1 \right) \left( 1 + \alpha c_2 \right)^{D-i-1} \frac{\sigma^M_{\text{In}}}{c_2\sqrt{|\tau|}} \\
+ (1 + \alpha c_2 + \Delta_k^{H_{\theta, \varphi}})^{D-i-1} \left[ c_1 m_2 ((1 + \alpha c_2)^D - 1) \frac{\sigma^M_{\text{In}}}{c_2\sqrt{|\tau|}} + m_2 \Delta_k^{H_{\theta, \varphi}} + m_2 \lambda_2 ((1 + \alpha c_2)^D - 1) \frac{\sigma^M_{\text{In}}}{c_2\sqrt{|\tau|}} \right] \\
(C.36)
\]

\[
\mathbb{V} \left[ \mathbb{E}_{\tau_1^{0:D-1}, \tau_2^{0:D-1}, \tau_3^{0:D-1}} \left[ \nabla J^D (\phi_k) \mid \tau_0^{0:D-1} \right] \right] \\
\leq 4^D \left( 2\mu_1^2 + 2D\alpha^2 (c_1 m_2 \frac{\rho^2}{c_2} + c_1 \mu_2 + m_2 \lambda_2)^2 \right) \times \\
(((1 + \alpha c_2)^2 - \alpha^2 c_2^2)^D - 1)(((1 + \alpha c_2)^2 + \alpha^2 (\Delta_k^{H_{\theta, \varphi}})^2)^D - 1) \frac{\sigma^M_{\text{In}}}{c_2^2|\tau|} \\
+ (2D\alpha^2 m_2^2)(((1 + \alpha c_2)^2 + \alpha^2 (\Delta_k^{H_{\theta, \varphi}})^2)^D - 1) \Delta_k^{H_{\theta, \varphi}}^2 \\
+ (2D\alpha^4 c_1 m_2^2) (((1 + \alpha c_2)^2 + \alpha^2 (\Delta_k^{H_{\theta, \varphi}})^2)^D - ((1 + \alpha c_2)^2 + \alpha^2 c_2^2)^D) \\
+C.37
\]

Term (ii).
\[
\begin{align*}
\mathbb{E}_{0:D-1} & \left[ \nabla J^D(\phi_k) \mid \tau_0^{0:D-1} \right] \\
= & \mathbb{E}_{0:D-1} \left[ \mathbb{E}_{1:D-1, \tau_1:D-1, \tau_2} \left[ \nabla J^D(\phi_k) - \mathbb{E}_{1:D-1, \tau_1:D-1, \tau_3} \left[ \nabla J^D(\phi_k) \right] \right]^2 \mid \tau_0^{0:D-1} \right] \\
\leq & \mathbb{E}_{0:D-1} \left[ 2 \mathbb{E}_{1:D-1, \tau_2} \left[ \nabla_{\phi} J^{Out} \left( \phi_k, \tilde{\theta}_k^D, \tau_3 \right) - \mathbb{E}_{\tau_3} \left[ \nabla_{\phi} J^{Out} \left( \phi_k, \tilde{\theta}_k^D, \tau_3 \right) \right] \right]^2 \mid \tau_0^{0:D-1} \right] \\
2\alpha^2 & \mathbb{E}_{1:D-1, \tau_2} \left[ \sum_{t=0}^{D-1} \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^i, \tau_1^i \right) \prod_{j=t+1}^{D-1} \left( 1 - \alpha \tilde{H}_{\theta,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \right) \nabla_{\theta} J^{Out} \left( \phi_k, \tilde{\theta}_k^D, \tau_3 \right) \\
& - \sum_{t=0}^{D-1} \mathbb{E}_{\tau_1^i} \left[ \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^i, \tau_1^i \right) \prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2^j} \left[ 1 - \alpha \tilde{H}_{\theta,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \right] \mathbb{E}_{\tau_3} \left[ \nabla_{\theta} J^{Out} \left( \phi_k, \tilde{\theta}_k^D, \tau_3 \right) \right] \right] \mid \tau_0^{0:D-1} \right] \\
\leq & \mathbb{E}_{0:D-1} \left[ 2(\sigma_1)^2 + 2D\alpha^2 \sum_{t=0}^{D-1} \left\| \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^i, \tau_1^i \right) \right\|^2 \times \\
\mathbb{E}_{1:D-1, \tau_2} \left[ \prod_{j=t+1}^{D-1} \left( 1 - \alpha \tilde{H}_{\theta,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \right) \nabla_{\theta} J^{Out} \left( \phi_k, \tilde{\theta}_k^D, \tau_3 \right) \\
& - \prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2^j} \left[ 1 - \alpha \tilde{H}_{\theta,\theta} \left( \phi_k, \tilde{\theta}_k^j, \tau_2^j \right) \right] \mathbb{E}_{\tau_3} \left[ \nabla_{\theta} J^{Out} \left( \phi_k, \tilde{\theta}_k^D, \tau_3 \right) \right] \right]^2 \mid \tau_0^{0:D-1} \right] \\
\mathbb{E}_{1:D-1, \tau_2} \left[ \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^i, \tau_1^i \right) \right] - \mathbb{E}_{\tau_1^i} \left[ \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^i, \tau_1^i \right) \right] \mid \tau_0^{0:D-1} \right] \\
\leq & \mathbb{E}_{0:D-1} \left[ \left\| \tilde{H}_{\phi,\theta} \left( \phi_k, \tilde{\theta}_k^i, \tau_1^i \right) \right\|^2 \mid \tau_0^{0:D-1} \right]
\end{align*}
\]
\begin{align*}
&\leq \mathbb{E}_{t_0} \left[ 2(\sigma_1)^2 + 2D \alpha^2 \sum_{r=0}^{D-1} \left\| \nabla_{\theta} J_{\text{Out}}^r (\phi_k, \Theta_k^D, \tau_3) \right\|^2 \right] \times \mathbb{E}_{t_1} \left[ \mathbb{E}_{t_2} \left[ \left\| \hat{H}_{\phi, \theta} (\phi_k, \Theta_k^l, \tau_1^l) \right\|^2 \right] \right] + \\
&\mathbb{E}_{t_1} \left[ \left\| \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \right\|^2 \right]
\end{align*}

Part (I)

\begin{align*}
&\leq \mathbb{E}_{t_0} \left[ 2(\sigma_1)^2 + 2D \alpha^2 \sum_{r=0}^{D-1} \left( 2m^2_2 + 2(\sigma_2)^2 \right) \times \left\| \hat{H}_{\phi, \theta} (\phi_k, \Theta_k^l, \tau_1^l) \right\|^2 \right] \times \\
&\mathbb{E}_{t_1} \left[ \left\| \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \right\|^2 \right] + \\
&\mathbb{E}_{t_1} \left[ \left\| \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \right\|^2 \right]
\end{align*}

Part (II)

\begin{align*}
&\leq \mathbb{E}_{t_0} \left[ 2(\sigma_1)^2 + 2D \alpha^2 \sum_{r=0}^{D-1} \left( 2m^2_2 + 2(\sigma_2)^2 \right) \times \left\| \hat{H}_{\phi, \theta} (\phi_k, \Theta_k^l, \tau_1^l) \right\|^2 \right] \times \\
&\mathbb{E}_{t_1} \left[ \left\| \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \mathbb{E}_{t_2} \left[ I - \alpha \hat{H}_{\theta, \theta} (\phi_k, \Theta_k^l, \tau_2^l) \right] \right\|^2 \right]
\end{align*}

Part (III)

\begin{align*}
&\leq \mathbb{E}_{t_1} \left[ \left\| \hat{H}_{\phi, \theta} (\phi_k, \Theta_k^l, \tau_1^l) - \mathbb{E}_{t_1} \left[ \hat{H}_{\phi, \theta} (\phi_k, \Theta_k^l, \tau_1^l) \right] \right\|^2 \right] \times \tau_0^{0:D-1}
\end{align*}

\textbf{Part (I)} According to the supporting Lemma D.4

56
\[
\mathbb{E}_{\tau_1^{D-1}, \tau_2^{D-1}, \tau_2} \left[ \prod_{j=t+1}^{D-1} \left( I - \alpha \widehat{H}_{\theta, \theta} \left( \phi_k, \widehat{\theta}_k^j, \tau_2^j \right) \right) - \prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2^j} \left[ I - \alpha \widehat{H}_{\theta, \theta} \left( \phi_k, \widehat{\theta}_k^j, \tau_2^j \right) \right] \right]^2
\leq 6^{D-t-1} \left[ (1 + \alpha c_2)^2 + \alpha^2 (\widehat{\Delta}_{\theta}^H)^2 + \alpha^2 (\sigma_{\theta}^H)^2 \right]^{D-t-1} - (1 + \alpha c_2)^2 + \alpha^2 (\widehat{\Delta}_{\theta}^H)^2 \right]^{D-t-1} \right]
\]

\textbf{Part (II)} According to the supporting Lemma D.2

\[\left\| \widehat{H}_{\phi, \theta} \left( \phi_k, \widehat{\theta}_k^0, \tau_1^j \right) \right\|^2 \leq 3c_1^2 + 3 \left( (\widehat{\Delta}_{\theta}^H)^2 + (\sigma_{\theta}^H)^2 \right) \]

\textbf{Part (III)}

\[\left\| \prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2^j} \left[ I - \alpha \widehat{H}_{\theta, \theta} \left( \phi_k, \widehat{\theta}_k^j, \tau_2^j \right) \right] \right\|^2 \]

\[\leq \prod_{j=t+1}^{D-1} \left\| I - \alpha \widehat{H}_{\theta, \theta} \left( \phi_k, \widehat{\theta}_k^j \right) + \alpha H_{\theta, \theta} \left( \phi_k, \widehat{\theta}_k^j \right) - \alpha \mathbb{E}_{\tau_2^j} \left[ \widehat{H}_{\theta, \theta} \left( \phi_k, \widehat{\theta}_k^j, \tau_2^j \right) \right] \right\|^2 \]

\[\leq (1 + \alpha c_2)^2 + \alpha^2 (\widehat{\Delta}_{\theta}^H)^2 \right]^{D-t-1} \]
Lemma D.1.

In this section, we present the supporting lemmas.

D Supporting Lemmas

In this section, we present the supporting lemmas.

Lemma D.1. \((D\text{-step Meta-Gradient})\)
D-step Meta-Gradient takes the form

\[
\nabla J^D(\phi_k) = \nabla \phi J^{Out}(\phi_k, \theta_k^D) - \alpha \sum_{i=0}^{D-1} H_{\phi,\theta}(\phi_k, \theta_k^i) \prod_{j=i+1}^{D-1} \left(I - \alpha H_{\theta,\theta}(\phi_k, \theta_k^j)\right) \nabla \theta J^{Out}(\phi_k, \theta_k^D) \tag{D.1}
\]

**Proof.** Based on the chain rule, we can get

\[
\nabla J(\phi_k) = \nabla \phi J^{Out}(\phi_k, \theta_k^D) + \nabla \theta J^{Out}(\phi_k, \theta_k^D) \frac{\partial}{\partial \theta} \nabla \theta J^{Out}(\phi_k, \theta_k^D) \tag{D.2}
\]

Based on the iterative updates that \(\theta_k^t = \theta_k^{t-1} - \alpha \nabla \theta J^{ln}(\phi_k, \theta_k^{t-1})\), for \(t = 1, \ldots, D\), we have

\[
\nabla \theta_k^t = \nabla \theta_k^{t-1} \left(I - \alpha H_{\theta,\theta}(\phi_k, \theta_k^{t-1})\right) - \alpha \tag{D.3}
\]

Telescoping the above equality over \(t\) from 1 to \(D\)

\[
\nabla \theta_k^D = \nabla \theta_k^0 \prod_{i=0}^{D-1} \left(I - \alpha H_{\theta,\theta}(\phi_k, \theta_k^i)\right) - \alpha \sum_{i=0}^{D-1} H_{\phi,\theta}(\phi_k, \theta_k^i) \prod_{j=i+1}^{D-1} \left(I - \alpha H_{\theta,\theta}(\phi_k, \theta_k^j)\right) \tag{D.4}
\]

Combining Eq. (D.2) and Eq. (D.4) finishes the proof of Lemma D.1. \(\square\)

**Lemma D.2.**

\[
\left\| \hat{H}_{\phi,\theta}(\phi_k, \theta_k^0, \tau_1) \right\|^2 \leq 3c_1^2 + 3 \left( (\Delta_k^H)^2 + (\tau_k^H)^2 \right) \tag{D.5}
\]

**Proof.**

\[
\left\| \hat{H}_{\phi,\theta}(\phi_k, \theta_k^0, \tau_1) \right\|^2 \\
\leq \left\| H_{\phi,\theta}(\phi_k, \bar{\theta}_k) - H_{\phi,\theta}(\phi_k, \hat{\theta}_k) + \mathbb{E}_{\tau_1} [\hat{H}_{\phi,\theta}(\phi_k, \bar{\theta}_k, \tau_1)] \right\|
\\
- \mathbb{E}_{\tau_1} [\hat{H}_{\phi,\theta}(\phi_k, \bar{\theta}_k, \tau_1)] + \hat{H}_{\phi,\theta}(\phi_k, \bar{\theta}_k, \tau_1) \right\|^2 \\
\leq 3c_1^2 + 3 \left( (\Delta_k^H)^2 + (\tau_k^H)^2 \right) \tag{D.6}
\]

59
Thus, we conclude the proof of Lemma D.2.

**Lemma D.3.**

\[
\mathbb{E}_{r_0:D-1} \left[ \prod_{j=t+1}^{D-1} \mathbb{E}_{r_2} [I - \alpha \tilde{H}_{\theta,\theta} (\phi_k, \hat{\theta}_k^j, \tau_2^j) | \tau_0^{0:t-1}] - \prod_{j=t+1}^{D-1} (I - \alpha H_{\theta,\theta} (\phi_k, \theta_k^j)) \right] \\
\leq \alpha \left[(1 + \alpha c_2 + \alpha \Delta_k^{H_{\theta,\theta}})^{D-t-1} - (1 + \alpha c_2)^{D-t-1} \right] \\
+ \frac{\rho^2}{c_2} ((1 + \alpha c_2 + \alpha \Delta_k^{H_{\theta,\theta}})^{D-t-1} - 1)(1 + \alpha c_2)^{D-1} \frac{\sigma_{M-In}^2}{c_2 \sqrt{\tau}} \tag{D.7}
\]

**Proof.**

\[
\mathbb{E}_{r_0:D-1} \left[ \prod_{j=t+1}^{D-1} \mathbb{E}_{r_2} [I - \alpha \tilde{H}_{\theta,\theta} (\phi_k, \hat{\theta}_k^j, \tau_2^j) | \tau_0^{0:t-1}] - \prod_{j=t+1}^{D-1} (I - \alpha H_{\theta,\theta} (\phi_k, \theta_k^j)) \right] \\
\leq \mathbb{E}_{r_0:D-1} \left[ \prod_{j=t+1}^{D-2} \left( I + \alpha c_2 + \alpha \Delta_k^{H_{\theta,\theta}} \right) \right] \left( \alpha \Delta_k^{H_{\theta,\theta}} + \alpha \rho_2 ((1 + \alpha c_2)^{D-1} - 1) \frac{\sigma_{M-In}^2}{c_2 \sqrt{\tau}} \right) \\
+ (1 + \alpha c_2)^{D-t-1} \\
+ \frac{\rho^2}{c_2} ((1 + \alpha c_2 + \alpha \Delta_k^{H_{\theta,\theta}})^{D-t-1} - 1)(1 + \alpha c_2)^{D-1} \frac{\sigma_{M-In}^2}{c_2 \sqrt{\tau}} \tag{D.8}
\]

Thus, we conclude the proof of Lemma D.3.

**Lemma D.4.**

\[
\mathbb{E}_{r_1:D-1, \tau_2^{D-1}, \tau_3} \left[ \prod_{j=t+1}^{D-1} \left( I - \alpha \tilde{H}_{\theta,\theta} (\phi_k, \hat{\theta}_k^j, \tau_2^j) \right) \right] - \prod_{j=t+1}^{D-1} \left( I - \alpha \tilde{H}_{\theta,\theta} (\phi_k, \theta_k^j, \tau_2^j) \right) \right] \\
\leq 6^{D-t-1} \left[ (1 + \alpha c_2)^2 + \alpha^2 (\Delta_k^{H_{\theta,\theta}})^2 + \alpha^2 (\sigma_{M-In}^2)^2 \right]^{D-t-1} - ((1 + \alpha c_2)^2 + \alpha^2 (\Delta_k^{H_{\theta,\theta}})^2)^{D-t-1} \\
\tag{D.9}
\]
Proof.

\[
\mathbb{E}_{\tau_1^{0:D-1}, \tau_2^{1:D-1}, \tau_3} \left[ \left\| \prod_{j=t+1}^{D-1} \left( I - \alpha \hat{H}_{\theta,\theta} \left( \phi_k, \hat{\phi}_k^i, \tau_2^j \right) \right) - \prod_{j=t+1}^{D-1} \mathbb{E}_{\tau_2^j} \left[ I - \alpha \hat{H}_{\theta,\theta} \left( \phi_k, \hat{\phi}_k^i, \tau_2^j \right) \right] \right\|^2 \right]
\]

\[
\leq 2 \prod_{j=t+1}^{D-2} \left( 3(1 + \alpha c_2)^2 + 3\alpha^2 (\Delta_k^{H_{\theta,\theta}})^2 + 3\alpha^2 (\sigma_k^{H_{\theta,\theta}})^2 \right) \times \alpha^2 (\sigma_k^{H_{\theta,\theta}})^2 + \\
4 \left( 1 + \alpha c_2 \right)^2 + \alpha^2 (\Delta_k^{H_{\theta,\theta}})^2 \times \\
\mathbb{E}_{\tau_2^{1:D-1}} \left[ \left\| \prod_{j=t+1}^{D-2} \left( I - \alpha \hat{H}_{\theta,\theta} \left( \phi_k, \hat{\phi}_k^i, \tau_2^j \right) \right) - \prod_{j=t+1}^{D-2} \mathbb{E}_{\tau_2^j} \left[ I - \alpha \hat{H}_{\theta,\theta} \left( \phi_k, \hat{\phi}_k^i, \tau_2^j \right) \right] \right\|^2 \right]
\]

\[
\leq 6^{D-t-1} \left[ (1 + \alpha c_2)^2 + \alpha^2 (\Delta_k^{H_{\theta,\theta}})^2 + \alpha^2 (\sigma_k^{H_{\theta,\theta}})^2 \right]^{D-t-1} - (1 + \alpha c_2)^2 + \alpha^2 (\Delta_k^{H_{\theta,\theta}})^2 \right]^{D-t-1}
\]

(D.10)

Thus, we conclude the proof of Lemma D.4

\[ \Box \]