# Sampling Attacks on Meta Reinforcement Learning: A Minimax Formulation and Complexity Analysis

**Tao Li, Haozhe Lei, and Quanyan Zhu**
Department of Electrical and Computer Engineering
New York University
NY 11201
{taoli, hl4155, qz494}@nyu.edu

## Abstract

Meta reinforcement learning (meta RL), as a combination of meta-learning ideas and reinforcement learning (RL), enables the agent to adapt to different tasks using a few samples. However, this sampling-based adaptation also makes meta RL vulnerable to adversarial attacks. By manipulating the reward feedback from sampling processes in meta RL, an attacker can mislead the agent into building wrong knowledge from training experience, which deteriorates the agent's performance when dealing with different tasks after adaptation. This paper provides a game-theoretical underpinning for understanding this type of security risk. In particular, we formally define the sampling attack model as a Stackelberg game between the attacker and the agent, which yields a minimax formulation. It leads to two online attack schemes: Intermittent Attack and Persistent Attack, which enable the attacker to learn an optimal sampling attack, defined by an $\epsilon$-first-order stationary point, within $\mathcal{O}(\epsilon^{-2})$ iterations. These attack schemes freeride the learning progress concurrently without extra interactions with the environment. By corroborating the convergence results with numerical experiments, we observe that a minor effort of the attacker can significantly deteriorate the learning performance, and the minimax approach can also help robustify the meta RL algorithms. Code and demos are available at `https://github.com/NYU-LARX/Attack_MetaRL`

## 1 Introduction

Meta Reinforcement Learning (meta RL), as a combination of meta learning and reinforcement learning, intends to learn a decision-making model that can quickly adapt to new tasks. Compared with learning from scratch, meta RL enables the agent to leverage previous training data, requiring fewer samples when dealing with new tasks. The successes of meta RL [Wang et al., 2016, Duan et al., 2016, Finn et al., 2017, Fallah et al., 2020] can be explained from a feature learning standpoint: meta RL aims at building an internal representation of the policy (meta policy) that is broadly suitable for a collection of similar tasks. This internal representation is learned by either recurrent neural networks [Wang et al., 2016, Duan et al., 2016] or stochastic optimization [Finn et al., 2017, Fallah et al., 2020], when agents are exposed to the collection of tasks via a unique sampling process. In meta RL, in addtion to producing sample trajectories within a single task, the sampling process also randomly samples environments for learning a common representation shared across these tasks.

However, this unique sampling process also makes meta RL vulnerable when facing adversarial attacks. By manipulating sample data collected in the meta training phase, the attacker can mislead the agent into learning a wrong meta policy that adapts poorly to individual tasks from the collection. For example, in Model-Agnostic Meta Reinforcement Learning (MAMRL) [Finn et al., 2017], a unique sampling process is required for optimizing the meta policy. As shown by our numerical

Preprint. Under review.

results, slightly perturbing the reward feedback from this sampling, the attacker can significantly degrade the performance of meta RL agents under different tasks.

**Contributions** This paper aims to provide a theoretical understanding of adversarial attacks on sampling processes of meta RL, which is referred to as sampling attacks. Due to its mathematical clarity, we choose MAMRL [Finn et al., 2017] as the departure point of our discussion. Yet, the idea of sampling attacks also applies to other meta RL formulations [Wang et al., 2016, Duan et al., 2016] (see Appendix B).

Our contributions are three-fold. 1) We characterize the sampling attack as a Stackelberg game [Von Stackelberg, 2010] between the attacker and the agent (the learner), which yields a minimax formulation. 2) Based on this formulation, two online black-box training-time attack schemes are proposed: Intermittent Attack and Persistent Attack, which enable the attacker to learn optimal sampling attacks concurrently with the meta learning without extra interactions with the environment. 3) The optimality of sampling attacks is defined by $\epsilon$-first-order stationarity, and our nonasymptotic analysis shows that the proposed attack schemes can achieve the optimality within $\mathcal{O}(\epsilon^{-2})$ iterations.

To the best of our knowledge, this work is among the first endeavor to investigate online training-time adversarial attacks on sampling processes of meta RL. In addition, our analysis of nonconvex-nonconcave minimax problem can be extended to other RL problems sharing the same nature of nonconvexity.

## 2 Related Works

This section reviews recent progress in the area of adversarial attacks on RL. We position our work using the following categorizations and comment on the differences between existing approaches and the proposed one in this paper, highlighting our contribution to the security study of RL and meta RL.

**Testing-time and Training-time Attacks** Unlike testing-time attacks, where the attacker intends to degrade the performance of a learned and deployed policy, training-time attacks aim to enforce a target policy in favor of the attacker by manipulating the learning process.

Our work falls within the class of training-time attack. Even though our attack methods are based on reward manipulations, which have also been explored in the literature [Ma et al., 2019, Huang and Zhu, 2019, Zhang et al., 2020b, 2021], we emphasize that previous works mainly target Q-learning algorithm [Ma et al., 2019, Huang and Zhu, 2019, Zhang et al., 2020b] or neural policy gradient [Zhang et al., 2021], which relies on $Q$ value estimation. For these value-based algorithms, the influence of the reward manipulation on the value function is more straightforward than in meta RL, where the adaptation in the meta training complicates the analysis. The optimality results regarding previous attacks do not apply to meta RL, and our paper's theoretical analysis of the optimality of the reward manipulation is more challenging.

**White-box and Black-box Attacks** For both testing-time and training-time attacks on RL agents, existing works have investigated both white-box [Ma et al., 2019, Zhang et al., 2020b, Huang and Zhu, 2019] and black-box settings [Xu et al., 2021, Russo and Proutiere, 2021]. For white-box attacks, the attacker needs to fully know the learner's learning algorithm and the policy model. Some attacks additionally require the knowledge of the environment, e.g., transition dynamics [Xu et al., 2021].

In contrast, the proposed attacks, as a black-box attack[1], do not require prior knowledge of the environment setup or learning algorithms. With access to the policy model, the attacker can adjust its attack by gradient updates, regardless of learning algorithms or the environment parameters.

Compared with previous black-box attacks [Xu et al., 2021, Russo and Proutiere, 2021], the proposed attacks do not require interactions with the training environment, e.g., the simulator. All it needs are the sample trajectories rolled out by the learner during the training process. In other words, the attacker considered in this paper is a free rider who does not actively collect information regarding the agent's learning algorithm or the environment.

**Offline and Online Attacks** Previous works mainly consider offline training time reward manipulations [Ma et al., 2019, Huang and Zhu, 2019], where the attacker has full knowledge of the training

[1]The fault line between white-box and black-box may vary in different contexts, We here follow the notion used in [Xu et al., 2021] and treat our proposed attack as a black-box attack.

dataset. The attacker makes one decision on reward manipulation and then provides the poisoned data to RL agent. In this work, online attacks are studied, where the attacker manipulates the reward feedback sequentially, adaptive to the learner's learning process.

## 3 Preliminaries

Let $\{\mathcal{T}_i\}_{i=1}^T$ be the set of finite-horizon Markov Decision Processes (MDP) representing $T$ different tasks/environments. Each task $\mathcal{T}_i$ is given by a tuple $\langle \mathcal{S}, \mathcal{A}, P_i, r_i, \rho_i \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denotes the the space of states and actions, respectively. For each $\mathcal{T}_i$, $P_i : \mathcal{S} \times \mathcal{A} \to \mathcal{B}(\mathcal{S})$ denotes the transition kernel that maps a state-action pair to $\mathcal{B}(\mathcal{S})$, a Borel probability measure over the state space. Finally, the agent receives a reward $r_i(s, a)$ when taking action $a$ at state $s$, and the initial state is drawn from the initial distribution $\rho_i$. For the sake of simplicity, we assume that all tasks share the same state and action spaces, as well as the same horizon length $H$, and our analysis applies to cases where different tasks have different MDP formulations [Fallah et al., 2020].

To facilitate our discussion, the following notations are introduced. A trajectory from an MDP $\mathcal{T}_i$ is defined as $\tau(\mathcal{T}_i) = (s_1, a_1, \ldots, s_H, a_H)$, where $H$ denotes the horizon length. A batch of trajectories $\tau(\mathcal{T}_i)$ under the same MDP is considered to be a training data set $\mathcal{D}_i$. Given a trajectory $\tau_i$ (a shorthand for $\tau(\mathcal{T}_i)$), the total reward received over this trajectory is $R(\tau_i) := \sum_{t=1}^H r_i(s_t, a_t)$.

Suppose that the agent's policy in $\mathcal{T}_i$, $\pi(\cdot|\cdot;\theta) : \mathcal{S} \to \mathcal{B}(\mathcal{A})$, is parameterized with $\theta \in R^d$, and $\pi(a|s;\theta)$ is the probability of choosing action $a$ at state $s$. Then, given the transition dynamics $P_i(s'|s,a)$ and the initial distribution $\rho_i(s)$, the probability that such a trajectory $\tau_i$ occurs is $q_i(\tau;\theta) := \rho_i(s_1)\prod_{t=1}^H \pi(a_t|s_t;\theta)\prod_{t=1}^{H-1} P_i(s_{t+1}|s_t,a_t)$. Therefore, the expected cumulative rewards under the policy $\pi(\cdot|\cdot;\theta)$ in the task $\mathcal{T}_i$ is $J_i(\theta) = \mathbb{E}_{\tau_i \sim q_i(\cdot;\theta)}[R(\tau_i)]$. RL algorithms, such as policy gradient [Sutton et al., 2000], seek to find a $\theta^*$ that maximizes $J_i(\theta)$.

The idea of policy gradient method is to apply gradient descent with respect to the objective function $J_i(\theta)$. Following the policy gradient theorem [Sutton et al., 2000], we obtain $\nabla J_i(\theta) = \mathbb{E}_{\tau \sim q_i(\cdot;\theta)}[g_i(\tau;\theta)]$, $g_i(\tau;\theta) = \sum_{h=1}^H \nabla_\theta \log \pi_i(a_h|s_h;\theta) R_i^h(\tau)$ where $R_i^h(\tau) = \sum_{t=h}^H r_i(s_t,a_t)$. In RL practice, the policy gradient $\nabla J_i(\theta)$ is replaced by its Monte Carlo (MC) estimation, since evaluating the exact value is intractable. Given a batch of trajectories $\mathcal{D}_i(\theta)$ under the policy $\pi_i(\cdot|\cdot;\theta)$, the MC estimation is $\hat{\nabla} J_i(\theta, \mathcal{D}_i(\theta)) := 1/|\mathcal{D}_i(\theta)| \sum_{\tau \in \mathcal{D}_i(\theta)} g_i(\tau;\theta)$.

Instead of focusing on individual tasks $\mathcal{T}_i$, MAMRL, as introduced in [Finn et al., 2017], aims to find a good initial policy, also called a meta policy that returns satisfying rewards in expectation when it is updated using limited number of gradient step updates under a new environment $\mathcal{T}'$, sampled from a task distribution $p$. In particular, for only one-step policy gradient update with a step size $\alpha$, the optimization problem of MAMRL [Fallah et al., 2020] is

$$\max_{\theta \in \mathbb{R}^d} L(\theta) = \mathbb{E}_{\mathcal{T}_i \sim p} \mathbb{E}_{\mathcal{D}_i(\theta) \sim_{q_i(\cdot;\theta)}} [J_i(\theta + \alpha \hat{\nabla} J_i(\theta, \mathcal{D}_i(\theta)))]. \tag{1}$$

A stochastic gradient method (SG-MRL) is proposed in [Fallah et al., 2020] for solving the maximization problem (1) (see Algorithm 1), where an unbiased estimator of $\nabla L(\theta)$ can be efficiently computed for performing gradient ascent. Suppose that for simplicity $\mathcal{D}_i^1 = \{\tau\}$, the data set sampled using $q_i(\cdot;\theta_t)$, only contains one trajectory, then $\nabla_\theta \mathbb{E}_{\mathcal{D} \sim q_i(\cdot;\theta)}[J_i(\theta + \alpha \hat{\nabla} J(\theta, \mathcal{D}))] = \mathbb{E}_{\tau \sim q_i(\cdot;\theta)}[\nabla J_i(\theta + \alpha g_i(\tau;\theta))(I + \alpha \nabla_\theta g_i(\tau;\theta)) + J_i(\theta + \alpha g_i(\tau;\theta)) \nabla_\theta \log \pi(\tau;\theta)]$. The MC estimation of $\nabla_\theta \mathbb{E}_{\mathcal{D} \sim q_i(\cdot;\theta)}[J_i(\theta + \alpha \hat{\nabla} J(\theta, \mathcal{D}))]$ is given below:

$$\nabla_\theta^i MC = \hat{\nabla} J_i(\theta + \alpha g_i(\tau;\theta))(I + \alpha \nabla_\theta g_i(\tau;\theta)) + \hat{J}_i(\theta + \alpha g_i(\tau;\theta)) \nabla_\theta \log \pi(\tau;\theta), \tag{2}$$

where $\hat{\nabla} J_i(\theta + \alpha g_i(\tau;\theta))$ and $\hat{J}_i(\theta + \alpha g_i(\tau;\theta))$ are estimated using another batch of trajectories $\mathcal{D}_i^2$ rolled out under the policy $\pi(\theta + \alpha g_i(\tau;\theta))$. The average of $\nabla_\theta^i MC$ constitutes an unbiased estimate of $\nabla L(\theta)$.

## 4 Sampling Attack Models

In this section, we propose three sampling attack models and characterize them as Stackelberg games between the attacker and the learner. As shown in Algorithm 1, MAMRL includes two sampling

processes. The first sampling (line 5 in Algorithm 1) is for the MC estimation of the corresponding policy gradient, while the second sampling (line 7 in Algorithm 1) is for the MC estimation of the gradient of the objective function $L(\theta)$. The main idea of the sampling attack is to manipulate the reward signal so that MC estimations are corrupted. Since the meta training includes two sampling processes, there are three possible sampling attacks: 1) Inner Sampling Attack (ISA): only attacking the first sampling process, where samples are used to compute the gradient adaptation in the inner loop; 2) Outer Sampling Attack (OSA): only attacking the second sampling process, where samples are used to perform one step meta optimization in the outer loop; 3) Combined Sampling Attack (CSA): attacking both the first and the second sampling process.

**Algorithm 1** SG-MARL [Fallah et al., 2020]

1: **Input** Initialization $\theta_0$, $t = 0$, step size $\alpha, \eta_1$.
2: **while** not converge **do**
3: Draw a batch of i.i.d tasks $\mathcal{T}_i, i \in \mathcal{I}$ with size $|\mathcal{I}| = I$;
4: **for** $i \in \mathcal{I}$ **do**
5: Sample a batch of trajectories $\mathcal{D}_i^1$ under $q_i(\cdot; \theta_t)$;
6: $\theta_{t+1}^i = \theta_t + \alpha \hat{\nabla} J_i(\theta_t, \mathcal{D}_i^1)$ ;
7: Sample a batch of trajectories $\mathcal{D}_i^2$ under $q_i(\cdot; \theta_{t+1}^i)$;
8: Compute $\nabla_\theta^i MC$ using (2)
9: $\theta_{t+1} = \theta_t + (\eta_1 / I) \sum_{i \in \mathcal{I}} \nabla_{MC}^i$
10: **Return** $\theta_{t+1}$

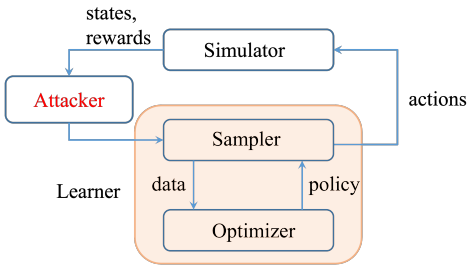



Figure 1: A schematic illustration of sampling attack. The attacker hijacks the reward feedback sent from the simulator to the learner, and poisons the rewards using attack methods (Intermittent and Persistent Attack) introduced in Section 5, in order to mislead the learner.

**Practicability of Sampling Attacks** Before delving into the details of these attacks, we first illustrate how this attack can happen in a training environment. For the training process in RL, a learner, as the orange box in Figure 1, usually consists of two components: a sampler and an optimizer. At each state, the sampler chooses an action based on the current policy, and sends it to the simulator, which returns a reward as well as the next state. The state-action-reward tuple will be stored within the sampler, and this process repeats until multiple trajectories are sampled. These sample trajectories make up a training data set fed to the optimizer, which performs one-step policy improvement.

To launch sampling attacks, the attacker first covertly infiltrates into the learning system shown in Figure 1 through Advanced Persistent Threats (APT) processes [Zhu and Rass, 2018]. Details on this infiltration are included in Appendix A.1. After the infiltration, the attacker intercepts communication between the sampler and the simulator and sends the learner poisoned reward signals, misleading the learner. One may argue that the attacker could gain control of the whole system, and its possible maneuvers are more than reward manipulation (e.g., compromising the simulator, corrupting the state input). In Appendix A.2, we justify our choice of reward manipulation as the first step of the investigation without losing the generality of sampling attacks.

The following summarizes three important aspects of the proposed sampling attacks. 1) **Knowledge of the attacker**: the attacker only has access to the sample trajectories rolled out by the learner and the current policy. The simulator and the learner's internal learning process remain black-box to the attacker. 2) **Attack Strategies**: the attacker is allowed to modify the rewards of sample trajectories provided by the simulator. It can adjust its attack sequentially depending on the meta learning process. Note that the attacker needs to choose the manipulation strategy from a constraint set, which can be interpreted as the attack budget (see Appendix A.3). 3) **The attacker's objective**: this paper considers a greedy attacker who misleads the learner into learning the worst possible meta policy, i.e., the value function of the learned meta policy [$L(\theta)$ in (1)] is minimized after sampling attacks. In the subsequent, we use ISA as an example to elaborate on these aspects of the proposed sampling attacks.

**Inner Sampling Attack** The first attack is to apply reward manipulations to the first sampling process. Denote the manipulated reward by $r^{adv}(s, a; \delta)$, which is parameterized by $\delta \in \Delta \subset \mathbb{R}^n, n \in \mathbb{N}$. We assume that $r^{adv}(s, a; \delta) := \delta \cdot r(s, a)$, i.e., the manipulated reward is obtained by applying the linear transformation to the original reward, and $\delta \in \Delta \subset \mathbb{R}$. It is a more general practice to represent

$r^{adv}$ by a neural network, $\delta$ being the weights of the network. Another remark is that the admissible manipulation strategies $\delta$ are confined to a bounded set $\Delta$ (see Assumption 6), limiting the attacker's ability. Appendix A.3 compares our attack constraints with existing constraints in the literature.

Under ISA, the policy adaptation $\theta_{t+1}^i = \theta_t + \alpha\hat{\nabla}J_i(\theta, \mathcal{D}_i^1)$ is compromised, as $\mathcal{D}_i^1$ is replaced by $\widetilde{\mathcal{D}}_i^1$, in which every reward is scaled by $\delta$. For simplicity, it is assumed that reward signals in $\mathcal{D}_i^1$ are uniformly scaled by $\delta$. More sophisticated manipulation strategies, such as adaptive poisoning [Zhang et al., 2020b], can also be considered, yet to keep the theoretical analysis neat, we limit ourselves to this uniform scaling within each batch. In the following, notations with $\widetilde{\cdot}$ either denote data corrupted by the manipulation or quantities computed using corrupted data.

Following the policy gradient theorem reviewed in Section 3, the policy gradient under ISA takes the following form $\widetilde{\nabla}J_i(\theta) = \mathbb{E}_{\tau\sim q_i(\cdot;\theta)}[g_i(\widetilde{\tau};\theta)]$, where $\widetilde{g}_i(\tau;\theta) = \sum_{h=1}^H \nabla_\theta \log \pi_i(a_h|s_h;\theta)R_i^h(\widetilde{\tau})$. Since $\widetilde{\tau}$ only differs from $\tau$ in that the rewards are scaled by $\delta$, $R_i^h(\widetilde{\tau}) = \delta R_i^h(\tau)$, the relationship between the corrupted policy gradient and the original one is given by $\widetilde{\nabla J_i}(\theta) = \delta\nabla J_i(\theta)$. Similarly, for the MC estimation, we have $\hat{\nabla}J(\theta, \widetilde{\mathcal{D}}) = \delta\hat{\nabla}J(\theta, \mathcal{D})$. Finally, the goal of the attack is to find a $\delta$ such that the performance of the learned meta policy [evaluated through the objective function $L(\theta)$, specified in (1).] is minimized. The optimization problem associated with (ISA) is given by

$$\begin{aligned} \min_{\delta\in\Delta} \quad & L(\theta^*) \\ \text{s.t.} \quad & \theta^* \in \arg\max_{\theta\in\mathbb{R}^d} \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta + \alpha\cdot\delta\hat{\nabla}J(\theta,\mathcal{D}))]. \end{aligned} \tag{ISA}$$

**Outer Sampling Attack** Another sampling attack is to consider manipulating the samples $\mathcal{D}_i^2$ in Algorithm 1. The outer sampling collects samples under the updated policy $\theta_{t+1}^i$, which are used to compute the MC estimation of the gradient of $L(\theta_t)$. In this case, the optimization problem in OSA share the same attack objective $\min_{\delta\in\Delta} L(\theta^*)$ as in ISA, but the constraint becomes

$$\text{s.t.} \quad \theta^* \in \arg\max_{\theta\in\mathbb{R}^d} \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[\delta J_i(\theta + \alpha\cdot\hat{\nabla}J(\theta,\mathcal{D}))]. \tag{OSA}$$

**Combined Sampling Attack** Finally, the inner and the outer sampling attack can be combined together. Following the same line of argument, the objective is $\min_{\delta\in\Delta} L(\theta^*)$, and the constraint is

$$\text{s.t.} \quad \theta^* \in \arg\max_{\theta\in\mathbb{R}^d} \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[\delta J_i(\theta + \alpha\cdot\delta\hat{\nabla}J(\theta,\mathcal{D}))]. \tag{CSA}$$

Some remarks are in order. Note that compared with the other two sampling attacks, OSA is rather trivial: a reward flipping, i.e., $\delta = -1$ is sufficient, which makes $\theta^*$ already a minimizer. Hence, for the rest of the paper, we mainly discuss (ISA) and (CSA). Our theoretical analysis focuses on (ISA), and its extension to (CSA) is straightforward (see Appendix E.1). Another remark is that even though there may exist multiple maximizers $\theta^*$, as the objective function is nonconvex differentiable, we assume that in our formulation, $\theta^*$ is properly selected, and hence unique. One way to justify this assumption is that $\theta^*$ can be viewed as the policy parameter returned by the RL algorithm in the training process, and hence, it is unique. We will elaborate on this selection more mathematically (see Lemma E1 and its proof) in Section 5.

Note that all three sampling attacks are formulated as bi-level optimization problems. Hence they amount to a Stackelberg game between the attacker (leader) and the learner (follower). The best response of the learner is to learn an optimal policy under the reward manipulation, and the three constraints on $\theta^*$ in the above formulations correspond to the learner's meta learning processes under ISA, OSA, and CSA, respectively. Knowing the learner's best response, the attacker aims to find a $\delta$ such that the learned meta policy adapts poorly in the testing phase without further attacks.

Since the attacker targets the testing performance of $\theta^*$, it needs to consider the policy gradient adaptation $(\theta^* + \alpha\hat{\nabla}J_i(\theta^*, \mathcal{D}(\theta^*))$ to evaluate $L(\theta^*)$, where $\mathcal{D}(\theta^*)$ is the genuine sample trajectories under $\theta^*$ without manipulation. However, in (ISA), the attacker can only access the learner's contaminated training data [i.e., sample trajectories under the policy $(\theta + \alpha\cdot\delta\hat{\nabla}J(\theta,\mathcal{D}))$]. In this case, the feedback of the attacker's manipulation [e.g., estimates of $\nabla L(\theta)$] is not directly available during the training time. Hence, searching for the optimal attack defined in (ISA) in an online manner is challenging, due the attacker's limited knowledge of the meta learning process.

**From Stackelberg to Minimax** As discussed above, it is a daunting task for the attacker to find the optimal $\delta$ by solving the bi-level optimization problem in (ISA). Thus, we reformulate the attacker's problem as a minimax problem (MMP), which is more tractable than (ISA), as the attacker now targets the learner's training performance as shown below.

$$\min_{\delta\in\Delta}\max_{\theta\in\mathbb{R}^d}\mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta+\alpha\cdot\delta\hat{\nabla}J(\theta,\mathcal{D}))]. \tag{MMP}$$

Note that in general the $\min$ and $\max$ operator in (MMP) is not changable, since the objective is nonconvex-nonconcave [Fallah et al., 2020]. Hence, (MMP) still amounts to a Stackelberg game, yet the attacker now aims to minimize the training performance under ISA, i.e., the value function under the corrupted dataset [the max part of (MMP)]. Sample trajectories rolled out by the learner can also help guide the attacker's search for the optimal attack, and no extra data or information is needed. In this case, the attack freerides the meta learing process, making the proposed sampling attack stealthy and lightweight. This free-ride nature will be more clear in Section 5, where two online attack schemes based on the gradient feedback are introduced.

Before concluding this section, we comment on the relationship between original (ISA) and (MMP). Even though the minimax value of (MMP) is by definition different from the Stackelberg equilibrium payoff given by (ISA), our following Proposition 1 states that under the assumption of bounded policy gradients (see Assumption 3 in Appendix D), the minimax value, when combined with a $\ell_2$-norm regularization $\|\delta-1\|$ serves as an upper-bound of the value function under the optimal attack.

**Proposition 1.** *Under the assumption of bounded policy gradients, there exists a constant $C$, such that for $\theta^*=\theta^*(\delta)$ defined in* (ISA)*, any batch of trajectories $\mathcal{D}$,*

$$J_i(\theta^*+\alpha\hat{\nabla}J_i(\theta^*,\mathcal{D}))]\leq J_i(\theta^*+\alpha\cdot\delta\hat{\nabla}J(\theta^*,\mathcal{D}))]+C\|\delta-1\|. \tag{3}$$

*As a consequence,*

$$\min_{\delta\in\Delta}L(\theta^*)\leq\min_{\delta\in\Delta}\mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta^*+\alpha\cdot\delta\hat{\nabla}J(\theta^*,\mathcal{D}))]+C\|\delta-1\|. \tag{4}$$

From Proposition 1, the free-rider nature can explained in the following way: as long as the training performance, i.e., the right-hand side of (4) deteriorates, the testing performance or the quality of the meta policy, i.e., the left-hand side of (4) cannot be any better. Even though the proposed sampling attack models are discussed in the context of MAMRL, they are also relevant to other meta RL frameworks, as discussed in Appendix B.

## 5 Sampling Attacks under Concurrent Learning

Based on (MMP), this section presents two onine attack schemes that enable the attacker to learn the optimal attack alongside with learner's learning process. We present the two schemes in the context of ISA, and the extension to CSA is straightforward (see Appendix E.1). To streamline the investigation into the optimality of proposed mechanisms, we skip the $\ell_2$ term, treating it as a regularization in numerical implementations, and mainly focus on (MMP).

**Intermittent Attack** In (MMP), the inner maximization problem $\max_{\theta\in\mathbb{R}^d}\mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta+\alpha\cdot\delta\hat{\nabla}J(\theta,\mathcal{D}))]$ corresponds to the meta RL under ISA, while the attacker's problem is given by $\min_{\delta\in\Delta}\mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta^*+\alpha\cdot\delta\hat{\nabla}J(\theta^*,\mathcal{D}))]$. A natural way to seek the minimizer is to apply gradient descent to the objective function. Using the similar argument in (2), we arrive at the following MC estimation of the gradient with respect to $\theta$,

$$\widetilde{\nabla}^i_\theta MC=\hat{\nabla}J_i(\theta+\delta g_i(\tau;\theta))(I+\delta\nabla_\theta g_i(\tau;\theta))+\hat{J}_i(\theta+\delta g_i(\tau;\theta))\nabla_\theta\log\pi(\tau;\theta). \tag{5}$$

Similarly for the gradient with respect to $\delta$ when fixing $\theta^*$, the MC estimate is given by

$$\nabla^i_\delta MC=\alpha\cdot\hat{\nabla}J_i(\theta^*+\alpha\cdot\delta g_i(\tau;\theta^*))g_i(\tau;\theta^*) \tag{6}$$

The attacker's learning proceeds as follows. At first, the attacker initializes the attack by setting $\delta=1$, and closely monitors the learning process. During the meta training process, the attacker remains dormant, i.e., $\delta$ remains constant, until policy parameter $\theta$ stabilizes. Then it is activated and updates $\delta$ using one-step stochastic gradient descent under the Euclidean projection $\Pi_\Delta(\cdot)$. It

repeats the above steps until the $\delta$ stabilizes. Since the attacker only adapts the reward manipulation intermittently, we refer to such attack as Intermittent Attack. The pseudocode is in Algorithm 2.

**Persistent Attack** Instead of waiting for the learner to converge to $\theta^*(\delta)$, the attacker can also implement a real-time attack, i.e., it performs gradient descent along with the learner's gradient ascent, no matter whether the current $\theta$ stabilizes or not. In this case, with the initialization $\delta = 1$, the attacker is spared from detecting the stabilization of the learner's training and updates $\delta$ every time $\theta$ is updated. Algorithm 3 summarizes this kind of attack, which we refer to as Persistent Attack, since the attacker keeps adjusting $\delta$ according to the learner's progress all the time. The intuition behind Persistent Attack is that if the attacker's learning rate is far smaller than the learner's, the learner views the attack as quasi-static, while the attacker treats $\theta_t$ as stabilized [Lin et al., 2019], which shares the same spirit as Intermittent Attack.

Before proceeding to the theoretical analysis, we comment on the differences between the two schemes in terms of cost and stealthiness. Since Intermittent Attack utilizes the stabilized policy parameter, the attacker's MC estimation (6) in Intermittent Attack tends to return a more accurate gradient descent direction than its counterpart in Persistent Attack when minimizing the objective function in (MMP). For the attacker's gradient estimation under two attack schemes, the reader is referred to Lemma E1, Lemma E5, and associated proofs. As a result, in practice, Intermittent Attack tends to find the optimal attack using fewer iterations (as shown in Figure 2(b)(d).) However, in Intermittent Attack, the attacker needs to detect the stabilization of the learner's training. This detection can be costly, as the attacker needs to keep a record of the average return of each iteration. In contrast, in Persistent Attack, the attacker simply updates the attack $\delta$ per iteration without detection. On the other hand, since Persistent Attack manipulates the rewards of sample trajectories during each iteration, it is less stealthy than Intermittent Attack.

**Algorithm 2** Intermittent Attack

**Input** maximum iterations $N, K$, meta learning step size $\eta_1$, meta attack step size $\eta_2$, initialization $\delta_0, \theta_0$, policy gradient step size $\alpha$.
**for** $t = 0, \dots, N-1$ **do**
  $\theta_0(\delta_t) = \theta_t$
  **for** $k = 0, \dots, K-1$ **do**
    Draw a batch of tasks $\mathcal{T}_i$, $i \in \mathcal{I}$;
    **for** $i \in \mathcal{I}$ **do**
      Sample $\mathcal{D}_i^1$ from $q_i(\cdot; \theta_t)$;
      $\theta_{k+1}^i = \theta_k + \alpha \cdot \delta \hat{\nabla} J_i(\theta_k, \mathcal{D}_i^1)$;
      Sample $\mathcal{D}_i^2$ from $q_i(\cdot; \theta_{k+1}^i)$;
    $\theta_{k+1} = \theta_k + (\eta_1/I) \sum_{i \in \mathcal{I}} \widetilde{\nabla}_\theta^i MC$;
  $\theta_{t+1} = \theta_K(\delta_t)$;
  $\delta_{t+1} = \Pi_\Delta \left( \delta_t + (\eta_2/I) \sum_{i \in \mathcal{I}} \nabla_\delta^i MC \right)$ ;
**Return** $\{\delta_t, \theta_K(\delta_t)\}$ for $t = 0, 1, \dots, N$.

**Algorithm 3** Persistent Attack

**Input** Initialization $(\delta_0, \theta_0)$,
step sizes $\eta_1 > \eta_2$.
**for** $t = 0, 1, 2, \dots, N-1$ **do**
  Draw a batch of iid tasks $\mathcal{T}_i$, $i \in \mathcal{I}$ with size $I$;
  **for** $i \in I$ **do**
    Sample $\mathcal{D}_i^1$ with respect to $q_i(\cdot; \theta_t)$;
    $\theta_{t+1}^i = \theta_t + \alpha \cdot \delta \hat{\nabla} J_i(\theta_t, \mathcal{D}_i^1)$;
    Sample $\mathcal{D}_i^2$ with respect to $q_i(\cdot; \theta_{t+1}^i)$;
    Compute $\widetilde{\nabla}_\theta^i MC$ using (5), and $\nabla_\delta^i MC$ using (6);
  $\theta_{t+1} = \theta_t + (\eta_1/I) \sum_{i \in \mathcal{I}} \widetilde{\nabla}_\theta^i MC$;
  $\delta_{t+1} = \Pi_\Delta \left( \delta_t - (\eta_2/I) \sum_{i \in \mathcal{I}} \nabla_\delta^i MC \right)$;
▷ $\Pi_\Delta(\cdot)$ is the Euclidean projection
**Return** $\{\delta_t, \theta_t\}$ for $t = 0, 1, \dots, N$.

**Optimality of Sampling Attacks** The objective in (MMP) is nonconvex-nonconcave. An appropriate optimality condition is the $\epsilon$-first order stationary point ($\epsilon$-FOSP), widely adopted in the study of programming and game theory [Nouiehed et al., 2019, Li et al., 2022].

**Definition 1** ($\epsilon$-FOSP). *$(\delta^*, \theta^*)$ is an $\epsilon$-FOSP of a continuously differentiable function $f(\delta, \theta)$ if $\min_{\delta \in B(\delta^*, 1)} \langle \nabla_\delta f(\delta^*, \theta^*), \delta - \delta^* \rangle \geq -\epsilon$, $\max_{\theta \in B(\theta^*, 1)} \langle \nabla_\theta f(\delta^*, \theta^*), \theta - \theta^* \rangle \leq \epsilon$, where $B(\delta^*, 1) := \{\delta \in \Delta : \|\delta - \delta^*\| \leq 1\}$, $B(\theta^*, 1) := \{\theta \in \mathbb{R}^d : \|\theta - \theta^*\| \leq 1\}$.*

Since the objective function in (MMP) is noncovex-nonconcave, in order to show the optimality of the attack, we need to impose some regularity conditions. For the ease of notation, we define $f(\delta, \theta) := \mathbb{E}_{i \sim p} \mathbb{E}_{\mathcal{D} \sim q_i(\cdot; \theta)}[J_i(\theta + \alpha \cdot \delta \hat{\nabla} J(\theta, \mathcal{D}))]$, and denote its MC estimation by $\hat{\nabla} f(\delta, \theta; \xi)$, where the random variable $\xi$ represents the a collection of datasets $\{\mathcal{D}_1^i\}_{i \in \mathcal{I}}, \{\mathcal{D}_i^2\}_{i \in \mathcal{I}}$ to be sampled in the sampling processes for the computation of $\nabla_\delta^i MC, \nabla_\theta^i MC$. Denote $\Phi(\delta) := \max_\theta f(\delta, \theta)$. The following assumptions extend Polyak-Łojasiewicz (PL) condition [Polyak, 1963, Łojasiewica, 1963] and restricted secant inequality (RSI) [Zhang and Yin, 2013] to the MMP. Note that both

Assumption 1 and Assumption 2 are weaker than commonly assumed strong convexity condition [Lin et al., 2019, Nouiehed et al., 2019], and detailed discussions are presented in Appendix D.

**Assumption 1** (Minmax PL)**.** *For* (MMP)*, it is assumed that there exists a constant* $\mu > 0$ *such that* $\frac{1}{2}\|\nabla_\theta f(\delta,\theta)\|^2 \geq \mu(\max_\theta f(\delta,\theta) - f(\delta,\theta))$.

**Assumption 2** (Minimax RSI)**.** *For* (MMP)*, it is assumed that there exists a constant* $\mu > 0$ *such that for any fixed* $\delta$, $\langle \nabla_\theta f(\delta,\theta), \theta^*(\delta) - \theta\rangle \geq \mu\|\theta^*(\delta) - \theta\|^2, \forall \theta \in \mathbb{R}^d$.

In addition to the above assumptions, some regularity assumptions are also needed to show the convergence of Intermittent and Persistent Attack. They are presented in Appendix D.

**Nonasymptotic Analysis of Intermittent Attack** The convergence result rests on that when the meta learning stabilizes, the attacker's gradient feedback $\nabla_\delta f(\delta_t, \theta_t^*(\delta_t))$ equals $\nabla\Phi(\delta_t)$ (see Lemma E1). Then, the standard analysis of gradient descent in nonconvex programming can be applied.

**Theorem 1.** *Under Assumption 1 and regularity assumptions, for any given* $\epsilon \in (0,1)$*, let the step sizes* $\eta_1, \eta_2$ *as well as the batch size* $M$ *be properly chosen (see Appendix E.2), if* $N \geq N(\epsilon) \sim \mathcal{O}(\epsilon^{-2})$*,* $K \geq K(\epsilon) \sim \mathcal{O}(\log \epsilon^{-1})$*, then there exists an index* $t$ *such that* $\{\delta_t, \theta_K(\delta_t)\}$ *generated by Algorithm 2 is an* $\epsilon$*-FOSP.*

**Nonasymptotic Analysis of Persistent Attack** Although Assumption 1 suffices for proving the optimality of Intermittent Attack, it is rather a weak condition when dealing with Persistent Attack. The gap is that in Persistent Attack, the attacker's gradient update does not utilize the exact gradient $\nabla\Phi(\delta_t) = \nabla_\delta f(\delta_t, \theta_t^*)$ or its approximation $\nabla_\delta f(\delta_t, \theta_K(\delta_t))$, instead, $\nabla_\delta f(\delta_t, \theta_t)$ is applied. To control the difference $d_t = \|\theta_t^* - \theta_t\|$, we show in Appendix E.3 that $d_t$ exhibits a linear contraction using Assumption 2, leading to the convergence result stated as follows.

**Theorem 2.** *Under Assumption 2 and regularity conditions, for any given* $\epsilon \in (0,1)$*, let the step sizes* $\eta_1, \eta_2$ *as well as the batch size* $M$ *be properly chosen (see Appendix E.3), then if* $N \geq N(\epsilon) \sim \mathcal{O}(\epsilon^{-2})$*, there exists an index* $t$*, such that* $\{\delta_t, \theta_t\}$ *generated by Algorithm 3 is an* $\epsilon$*-FOSP.*

## 6 Numerical Experiments

This section presents experimental evaluations of the proposed attack schemes for ISA and CSA. We consider two benchmark meta learning tasks, including a 2D-navigation task and a more complex locomotion problem: Half-cheetah (H-C) [Finn et al., 2017]. The following briefly introduces the two tasks, and more details can be found in Appendix C.1. In the 2D-navigation task, starting from a fixed initial position [the black triangle in Figure 2(a)] in a 2D plane, a point agent [the red dot in Figure 2(a)] needs to move to a goal position [the green star in Figure 2(a)] randomly sampled from a given task distribution. In the H-C task, a planar cheetah [as shown in Figure 2(c)] is required to run at a particular velocity sampled from the task distribution.

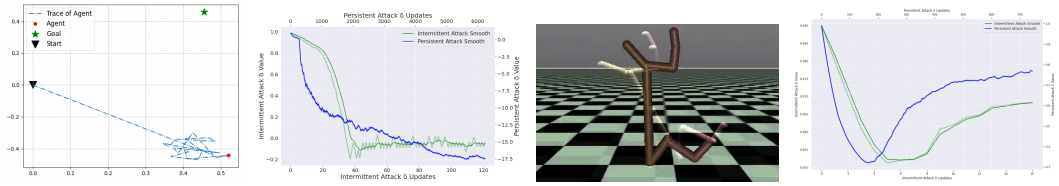
**Experimental Setup** Hyperparameter setup for the two algorithms is detailed in Appendix C.2. When evaluating the corrupted meta policy in the testing phase, we compare the adaptation performance [$L(\theta)$ defined in (1)] of meta policies learned under different attack settings and the benchmark setting without attacks. One-step policy adaption is considered throughout the paper. Additional experiments on multi-step adaption can be found in Appendix C.3. As in [Finn et al., 2017], the evaluation metric is the average of cumulative rewards of policies adapted from the meta policy.

**Experiment Results** Figure 2 presents an example of the poor adaptation of the corrupted meta policies and the convergence of two online attack schemes under ISA in a single experiment. In Figure 2(a), when using the policy adapted from the meta policy, the agent is misled into searching the lower-right corner [as the search paths concentrate on that corner], opposite to the true goal. Similarly, with a small perturbation $\delta = 0.967$, the cheetah agent fails to learn the desired running-forward policy as shown in Figure 2(c). In this example, it is observed in Figure 2(b)(d) that both attack schemes converge. Intermittent Attack takes fewer iterations since its gradient estimation is more accurate as we discussed in Section 5. To demonstrate the impact of sampling attacks, the first two columns in Table 1 summarizes the average rewards (and their standard deviations) of the policies after one-step adaptation from the meta policies corrupted by different attacks.

**Robust Training against Sampling Attacks** In Persistent Attack, the attacker and the learner operate in a two-timescale manner. Intuitively, the learner first achieves the inner maximization using a larger step size. On the contrary, if the learner's step size is smaller than the attacker's, then the

Table 1: Experimental evaluations of Intermittent Attack and Persistent Attack under ISA and CSA as well as robust training under ISA. Average cumulative rewards (and standard deviations) are chosen as the evaluation metric (higher the better), which is the sample mean of $L(\theta)$, $\theta$ being the meta policies corrupted by different attacks. As shown in the first two columns, Intermittent and Persistent Attacks significantly deteriorate the adaption performance of meta policies. Unlike Persistent Attack ($\eta_1 = 0.1 > 0.01 = \eta_2$), in the robust training, the learner's step size $\eta_1 = 0.01$ is smaller that the attacker's $\eta_2 = 0.1$. The decreased step size makes the meta leanring process robust to attacks.

| | | Intermittent Attack | Persistent Attack | Robust Training | Benchmark |
|---|---|---|---|---|---|
| 2D | ISA | $-30.06 \pm 7.37$ | $-298.77 \pm 34.09$ | $-11.50 \pm 0.57$ | $-10.92 \pm 0.85$ |
| | CSA | $-155.84 \pm 3.96$ | $-488.97 \pm 31.06$ | $---$ | |
| H-C | ISA | $-129.25 \pm 12.69$ | $-101.13 \pm 10.57$ | $-63.26 \pm 8.69$ | $-63.41 \pm 6.86$ |
| | CSA | $-98.92 \pm 10.85$ | $-93.65 \pm 9.81$ | $---$ | |




(a) The failure of 2D Navigation task under $\delta = -12.96$.

(b) Convergence of $\delta$ under two attacks schemes in 2D navigation.

(c) The failure of Half-cheetah task when $\delta = 0.967$.

(d) Convergence of $\delta$ under two attacks schemes in Half-cheetah.

Figure 2: Experimental results of meta RL agent in 2D navigation task and Half-cheetah task under Intermittent and Persistent Attack under ISA. (a)(c) demonstrates the poor performance of policies adapted from the corrupted meta policy. In (a), the agent is misled to search the lower-right corner, opposite to the true goal. In (c), the agent fails to run forward using the policy adapted from the meta policy. (b)(d) shows that Intermittent Attack need fewer iterations to converge than Persistent Attack.

attacker would first achieve the minimization. In this case, the Stackelberg game becomes a maxmin problem $\max_{\theta \in \mathbb{R}^d} \min_{\delta \in \Delta} f(\delta, \theta)$, inspiring a robust training method. During the meta learning process, the learner can decrease $\eta_1$ when it is suspicious of possible attacks. The obtained meta policy returns the best possible adaptation performance when the training data is corrupted. In one of our experiments, when $0.01 = \eta_1 < \eta_2 = 0.1$, it is observed that the meta policy is robust to Persistent Attack under ISA, as the adaptation performance is close to the benchmark. The column "Robust Training" in Table 1 compares the performance of the meta policies trained under Persistent Attack [$(\eta_1, \eta_2) = (0.1, 0.01)$], Robust Training [$(\eta_1, \eta_2) = (0.01, 0.1)$], and the benchmark.

## 7 Discussion

This work formally defines three sampling attack models on meta RL in the training phase. The optimal sampling attack is formulated as the solution to a minimax problem, which leads to two online black-box attack mechanisms: Intermittent Attack and Persistent Attack. Experiments show that simply scaling the reward feedback during the training, the attacker can significantly deteriorate the adaptation performance of the learned meta policy.

The main limitation of the proposed online attack schemes is that with the noised gradient feedback, the attacker searches for the $\epsilon$-FOSP (see Definition 1), which is a local notion. Since the objective function $f(\delta, \theta)$ is nonconvex-nonconcave, it is challenging to characterize these stationary points and compare the corresponding objective values with the global optimum. Hence, quantifying the impact of sampling attacks theoretically (e.g., the performance loss or robustness) remains an open problem. Finally, we comment on the potential negative societal impacts of the proposed sampling attacks. As meta RL has been applied to many real-world applications [Hospedales et al., 2021], the stealthy and lightweight sampling attack schemes pose a great threat to these learning systems. Our

robust training idea points out a promising way to combat the attack, yet more efforts are needed for a systematic investigation of defense mechanisms.

## References


A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift. *arXiv*, 2019.

J. Bannon, B. Windsor, W. Song, and T. Li. Causality and batch reinforcement learning: Complementary approaches to planning in unknown domains. *arXiv preprint arXiv:2006.02579*, 2020. doi: 10.48550/ARXIV.2006.02579. URL `https://arxiv.org/abs/2006.02579`.

P. Bernhard and A. Rapaport. On a theorem of danskin with an application to a theorem of von neumann-sion. *Nonlinear Analysis: Theory, Methods & Applications*, 24(8):1163–1181, 1995. ISSN 0362-546X. doi: https://doi.org/10.1016/0362-546X(94)00186-L. URL `https://www.sciencedirect.com/science/article/pii/0362546X9400186L`.

P. Dayan and C. J. Watkins. Q-Learning. *Machine Learning*, 8(3-4):279 292, 1992. ISSN 0885-6125. doi: 10.1007/bf00992698.

Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. RL2: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv*, 2016.

R. Fakoor, P. Chaudhari, S. Soatto, and A. J. Smola. Meta-q-learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SJeD3CEFPH`.

A. Fallah, K. Georgiev, A. Mokhtari, and A. Ozdaglar. On the Convergence Theory of Debiased Model-Agnostic Meta-Reinforcement Learning. *arXiv*, 2020.

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

T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, may 2021. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3079209.

Y. Huang and Q. Zhu. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *Decision and Game Theory for Security*, pages 217–237, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32430-8.

H. Karimi, J. Nutini, and M. Schmidt. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-ŁOjasiewicz Condition. In *European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 9851*, ECML PKDD 2016, page 795–811, Berlin, Heidelberg, 2016. Springer-Verlag. ISBN 9783319461274. doi: 10.1007/978-3-319-46128-1_50. URL `https://doi.org/10.1007/978-3-319-46128-1_50`.

T. Li and Q. Zhu. On Convergence Rate of Adaptive Multiscale Value Function Approximation for Reinforcement Learning. *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2019. doi: 10.1109/mlsp.2019.8918816.

T. Li, G. Peng, and Q. Zhu. Blackwell Online Learning for Markov Decision Processes. *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, 00:1–6, 2021. doi: 10.1109/ciss50987.2021.9400319.

T. Li, G. Peng, Q. Zhu, and T. Başar. The confluence of networks, games, and learning a game-theoretic framework for multiagent decision making over networks. *IEEE Control Systems Magazine*, 42(4):35–67, 2022. doi: 10.1109/MCS.2022.3171478.

T. Lin, C. Jin, and M. I. Jordan. On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems. *arXiv*, 2019.

T. Liu, Z. Liu, Q. Liu, W. Wen, W. Xu, and M. Li. StegoNet: Turn Deep Neural Network into a Stegomalware. *Annual Computer Security Applications Conference*, pages 928–938, 2020. doi: 10.1145/3427228.3427268.

S. Łojasiewica. A topological property of real analytic subsets (in french). *Coll. du CNRS, Leśequationsaux ́derivées partielles*, pages 87–89, 1963.

Y. Ma, X. Zhang, W. Sun, and J. Zhu. Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems*, 32:14570–14580, 2019.

B. Mehta, T. Deleu, S. C. Raparthy, C. J. Pal, and L. Paull. Curriculum in Gradient-Based Meta-Reinforcement Learning. *arXiv*, 2020.

M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn. Solving a Class of Non-Convex Min-Max Games Using Iterative First Order Methods. *arXiv*, 2019.

B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal vychislitel'noi matematiki i matematicheskoi fiziki*, 3(4):643–653, 1963.

K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5331–5340. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/rakelly19a.html`.

A. Russo and A. Proutiere. Towards Optimal Attacks on Reinforcement Learning Policies. *2021 American Control Conference (ACC)*, 00:4561–4567, 2021. doi: 10.23919/acc50511.2021.9483025.

J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1506.02438`.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. v. d. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–9, 2016. ISSN 0028-0836. doi: 10.1038/nature16961.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057—1063. MIT press, 2000.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

H. Von Stackelberg. Market structure and equilibrium. 2010. doi: 10.1007/978-3-642-12586-7.

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

H. Xu, R. Wang, L. Raizman, and Z. Rabinovich. Transferable environment poisoning: Training-time attack on reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, page 1398–1406, Richland, SC, 2021. ISBN 9781450383073.

H. Zhang and W. Yin. Gradient methods for convex minimization: better rates under weaker conditions. *arXiv preprint arXiv:1303.4645*, 2013.

H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33:21024–21037, 2020a.

X. Zhang, Y. Ma, A. Singla, and X. Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 11225–11234. PMLR, 2020b.

X. Zhang, Y. Chen, X. Zhu, and W. Sun. Robust Policy Gradient against Strong Data Corruption. *arXiv*, 2021.

Q. Zhu and S. Rass. On Multi-Phase and Multi-Stage Game-Theoretic Modeling of Advanced Persistent Threats. *IEEE Access*, 6:13958–13971, 2018. ISSN 2169-3536. doi: 10.1109/access.2018.2814481.

# A Practicability of Sampling Attacks

## A.1 Infiltration through Advanced Persistent Threats

In reality, sampling attacks can be achieved through Advanced Persistent Threats [Zhu and Rass, 2018], a class of multi-stage stealthy attack processes. In the first stage, the attacker embeds malware in neural network models by replacing the model parameters with malware bytes Liu et al. [2020]. Then, the malware is delivered covertly along with the model into the system in Figure 1. In the second stage, the malware performs reconnaissance and tailors its attack to the communication channel between the simulator and the sampler. In the final stage, the malware launches sampling attacks using either Intermittent Attack (see Algorithm 2) or Persistent Attack (see Algorithm 3). When implementing sampling attacks, the attacker stands between the simulator and the learner, cutting off the information transmission between the two and sending the learner manipulated reward feedback to mislead the learner.

## A.2 Justification of the Reward Manipulation

When it has covertly infiltrated the learning system, the attacker can corrupt any component within the learning system in Figure 1. This work considers sampling attacks, a particular case of man-in-the-middle attacks, where the attacker interrupts an existing conversation or data transfer between the simulator and the sampler. Sampling attacks only require sample trajectories under the current policy model to sabotage the learning process. In contrast, if the attacker were to control the simulator or other components of the system, it requires knowledge of how these components are built and operate. Therefore, sampling attacks are more lightweight and stealthy, requiring less knowledge of the learning system.

The attacker can manipulate everything in sample trajectories during sampling attacks, including states, actions, and rewards. We choose reward manipulation as the first step of the investigation because rewards are one-dimensional scalars, and they are more vulnerable to adversarial manipulations (state and action manipulations may require more advanced devices [Zhang et al., 2020a, Russo and Proutiere, 2021] than simply linear scaling). Its analysis leads to sufficient insights without losing the generality of sampling attacks, since the Stackelberg formulation also applies to state and action manipulations.

## A.3 Attack Budgets

The attack budget aims to limit the attacker's ability to sabotage the learning process; otherwise, it is not surprising that an omnipotent attacker can compromise the learning system of interest. Previous works mainly study two kinds of attack budgets. The first one is to restrict the amount of sample data the attacker can attack [Zhang et al., 2021, Huang and Zhu, 2019, Ma et al., 2019] which we refer to as the volume constraint. This volume constraint is often considered in offline RL or batch RL scenarios where the training dataset is prefixed [Zhang et al., 2021, Ma et al., 2019]. The other one is to confine the attacker's perturbation to a limited magnitude [Zhang et al., 2020a, Russo and Proutiere, 2021, Zhang et al., 2020b], which we refer to as the magnitude constraint.

This work considers the magnitude constraint as in other studies on online attacks [Zhang et al., 2020a,b]. Note that the proposed Intermittent and Persistent Attacks can accommodate the volume constraint by restricting the number of sample trajectories the attacker can poison. In this case, our Stackelberg formulations are still valid, yet the optimality analysis needs refinements as the gradient-based analysis is not directly available. The associated optimality analysis is beyond the nonconvex-nonconcave programming framework and remains an open problem.

# B Relevance to Broader Meta RL research

This work chooses a well-accepted meta RL framework [Finn et al., 2017, Fallah et al., 2020] and studies associated sampling attacks. In particular, our convergence analysis in Section 5 is built on the theoretical results regarding SG-MRL established in [Fallah et al., 2020].

Picking MAMRL serves as a starting point, and our findings can apply to many other variants of meta RL algorithms. The proposed minimax formulation in this work remains valid for other

non-MAMRL-based algorithms. Recall that in (MMP), the max part corresponds to the objective in MAMRL. We can replace this objective with its counterparts using other meta RL formulations, such as recurrent-network-based ones [Wang et al., 2016, Duan et al., 2016]. Similar to MAMRL, meta RL methods in [Wang et al., 2016, Duan et al., 2016] utilize on-policy data during both meta-training and adaptation. In this case, the attacker can also leverage these on-policy data to estimate its gradient feedback. Hence, Intermittent and Persistent Attacks also apply to these meta RL frameworks.

On the other hand, our online attack schemes need refinements when dealing with meta RL methods based on off-policy learning [Fakoor et al., 2020, Rakelly et al., 2019]. In off-policy learning, such as Q-learning [Dayan and Watkins, 1992] and its variants [Silver et al., 2016, Li and Zhu, 2019, Li et al., 2021], sample trajectories are collected using a behavior policy different from the target policy (the optimal policy). In this case, the attacker may not be able to assess its attack impact on the target policy if it can only access sample trajectories under the behavior policy. One possible remedy would be to reveal to the attacker some side information on the importance ratio between the target policy and the behavior policy [Bannon et al., 2020] so that the target policy's performance can be estimated using off-policy data. We leave this topic as future work.

# C Experiment Details

## C.1 Meta Reinforcement Learning Tasks

**2D Navigation** Consider in a 2D plane, a point agent must move to different goal positions that are sampled according to a task distribution. Goals are picked from a unit square, centered at the origin with length and width ranging from [-0.5, 0.5]. The state observation is the current 2D position, a 2-dimensional vector, and actions correspond to velocity commands clipped to the range [-0.1, 0.1]. The reward is the negative squared distance to the goal, and iterations terminate when the agent is within 0.01 of the goal or at the horizon of 100 steps. The discounting factor is $0.99$.

**Half-Cheetah** In the half-cheetah goal velocity problem, a planar cheetah is required to run at a particular velocity, randomly sampled from $[0, 2]$. The state variable is a 17-dimensional vector, representing the position and velocity of the cheetah's joints, and actions correspond to continuous momentum of six leg joints. The transition follows the physical rule simulated by MuJoCo Todorov et al. [2012], and the reward is defined as negative absolute value between the current velocity of the agent and a goal, which is chosen uniformly at random from [0, 2]. The horizon length is 200. The discounting factor is $0.99$.

## C.2 Experimental Setup

The attack program is built on the meta RL program developed in [Mehta et al., 2020]. We modify the original meta RL program to support parallel computing using graphics cards, saving a significant amount of training time. All the experiments are conducted using a Linux (Ubuntu 18.04) desktop with 4 Nvidia RTX8000 graphics cards and an AMD Threadripper 3990X (64 Cores, 2.90 GHz) processor. The experiments use a feedforward neural network policy with two 64-unit hidden layers and tanh activiations. The rest of this subsection summarizes the experimental setup of the meta training and sampling attack schemes.

**Meta RL Training** Recall that in Algorithm 1, a batch of i.i.d. tasks are first sampled from the task distribution. This batch of tasks is referred to as the meta batch [Finn et al., 2017, Fallah et al., 2020]. Then for each task $\mathcal{T}_i$, the sampler returns two batches of trajectories: the inner batch $\mathcal{D}_i^1$ and the outer batch $\mathcal{D}_i^2$. The batch sizes (the number of trajectories/ episodes) of the inner and the outer batch are the same in the experiments. This size is referred to as the batch size.

In the training phase, the total iteration number is 1000 for 2D navigation (2D) and half-cheetah (H-C) tasks, and the meta batch size is 20 for 2D and 40 for H-C. For both tasks, the batch size is 20 episodes. The learning rate is $\eta_1 = 0.1$. The policy gradient is obtained through the generalized advantage estimator (GAE) [Schulman et al., 2016], which is a refinement of the vanilla policy gradient reviewed in Section 3. Unlike the original policy gradient estimator

$\nabla J(\theta) = \mathbb{E}_{\tau\sim q(\cdot;\theta)}[\sum_{h=1}^{H} \nabla_\theta \log \pi(a_h|s_h;\theta) R^h(\tau)]$, the GAE estimator takes the following form

$$\begin{aligned}\nabla^{GAE} J(\theta) &= \mathbb{E}_{\tau\sim q(\cdot;\theta)}[\sum_{h=1}^{H} \nabla_\theta \log \pi(a_h|s_h;\theta) A(s_h)], \\ &= \nabla_\theta \mathbb{E}_{\tau\sim q(\cdot;\theta)}[\sum_{h=1}^{H} \log \pi(a_h|s_h;\theta) A(s_h)],\end{aligned}$$

where the advantage function $A(\cdot)$ is defined as $A(s_h) := R^h(\tau) - \hat{V}(s_h)$, and $\hat{V}(\cdot)$ is the baseline function. $\mathbb{E}_{\tau\sim q(\cdot;\theta)}[\sum_{h=1}^{H} \log \pi(a_h|s_h;\theta) A(s_h)]$ (its MC estimation) is referred to as the reinforce loss function (empirical reinforce loss), and this loss is the optimization objective maximizing the probability of taking good actions. In short, the higher the reinforce loss, the better policy is. The introduction of the advantage function helps reduce the estimation variance in the RL training. The reader is referred to [Schulman et al., 2016] for more details on computing the baseline function and interpretations of the advantage function.

**Sampling Attacks** For both Intermittent (Algorithm 2) and Persistent (Algorithm 3) Attacks, the learner's meta learning process uses the hyperparameters specified above, albeit the inner batch and the outer batch are poisoned according to sampling attack schemes. Unless otherwise specified, the learner's and the attacker's learning rates in Intermittent Attack are $\eta_1 = 0.1, \eta_2 = 0.1$, respectively, and $\eta_1 = 0.1, \eta_2 = 0.01$ in Persistent Attack. In sampling attack experiments, the regularization $|\delta - 1|$ is incorporated into the (MMP) objective function.

**Testing** To evaluate the quality of the learned meta policy, we conduct test experiments. In the experiment, a batch of tasks is first sampled. The meta policy first perform the gradient adaptation using the inner batch in each sampled task. Then, the adaptation performance $L(\theta)$ is given by the averaging cumulative rewards of episodes from the outer batch of every sampled task. Note that in the testing phase, no attacks are allowed. For a single meta policy, We repeat the test experiment with 10 random seeds and report mean rewards and standard deviations of these experimental results in Table 1 and other tables in the following subsection.

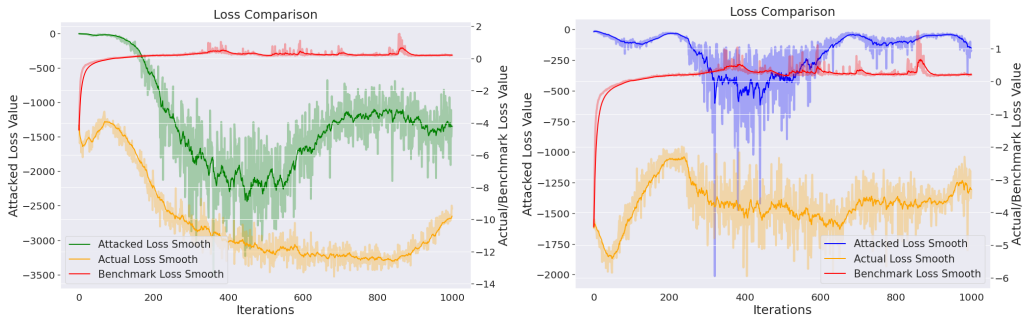
### C.3 Additional Experiment Results

**The intuition behind Sampling Attacks** By scaling the reward signals by $\delta$, the attacker distorts the learner's perception of its reinforce loss, misleading the meta learning process. This distortion can be explained using the reinforce loss plots in Figure 3. We choose one representative experiment for each meta task under ISA and plot three kinds of reinforce loss functions (under smoothing) reviwed in the paragraph "Meta Training". The benchmark loss (red curves) is obtained by averaging all empirical reinforce loss (ERL) of policies adapted from the meta policy during the training phase in the benchmark setting (no attack). Denote by $\theta_t$ the meta policy at the $t$-th iteration in the benchmark training, the benchmark loss at time $t$ is given by

$$\begin{aligned}\text{Bechmark Loss} =& \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} \text{ERL}(\theta_t^i), \quad \theta_t^i = \theta_t + \alpha \widehat{\nabla} J_i(\theta_t, \mathcal{D}_i^1) \\ \text{ERL}(\theta_t^i) :=& \frac{1}{|\mathcal{D}_i^2|} \sum_{\tau\in\mathcal{D}_i^2} [\sum_{h=1}^{H} \log \pi(a_h|s_h;\theta_t^i) A(s_h)], \quad (s_h, a_h) \in \tau,\end{aligned}$$

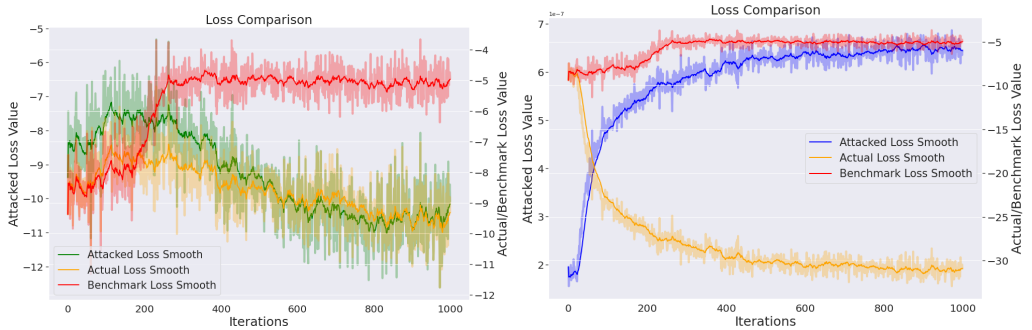
where the inner batch $\mathcal{D}_i^i$ the outer batch $\mathcal{D}_i^2$ are sampled using $\theta_t$ and $\theta_t^i$, respectively (see Algorithm 1). ERL is the MC estimation of the reinforce loss.

The benchmark loss shows the training progress of meta RL in a neutral environment and serves as the baseline. To compare the training progress under adversarial attacks with this baseline, we introduce two other loss functions. Denote by $\hat{\theta}_t$ the meta policy obtained under ISA and by $\delta$ the attacker's manipulation, then the attacked loss (green curves for Intermittent Attack and blue curves




(a) Reinforce loss under benchmark and Intermittent Attack in 2D navigation.

(b) Reinforce loss under benchmark and Persistent Attack in 2D navigation.




(c) Reinforce loss under benchmark and Intermittent Attack in half-cheetah.

(d) Reinforce loss under benchmark and Persistent Attack in half-cheetah.

Figure 3: Comparisons of the learner's empirical reinforce loss function under the benchmark and ISA settings.

for Persistent Attack) and the actual loss (yellow curves) are given by

$$\text{Attacked Loss} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \text{ERL}(\tilde{\theta}_t^i), \quad \tilde{\theta}_t^i = \hat{\theta}_t + \alpha\cdot\delta\widehat{\nabla} J_i(\theta_t, \mathcal{D}_i^1),$$

$$\text{Actual Loss} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \text{ERL}(\hat{\theta}_t^i), \quad \hat{\theta}_t^i = \hat{\theta}_t + \alpha\widehat{\nabla} J_i(\theta_t, \mathcal{D}_i^1).$$

The attacked loss implies the learner's distorted perception of its training progress as its policy gradient is corrupted. The actual loss, on the other hand, reflects the true learning progress in the adversarial environment since the ERL of $\hat{\theta}_t^i$ reflects the current meta policy's adaptation performance under actual sample data.

As shown in Figure 3, the agent mistakenly believes that it is making progress in the adversarial environment, as the green/blue curves go up (see Figures 3(a)(d)) or even above the benchmark (see Figures 3(b)). However, the actual performance of the learning either keeps deteriorating (see Figures 3(c)(d)) or remains at a modest level (see Figures 3(a)(b)).

**Additional Results on Intermittent and Persistent Attacks** This subsection presents additional experimental results on sampling attacks. We begin with experiments on attacking meta RL with two-step policy gradient adaptation (the one-step adaptation case is in Section 3 ). Algorithm 4 summarizes the MAMRL algorithm using two-step policy gradient adaptation. Note that when using two-step policy adaption, the inner sampling process requires two batches of trajectories $\mathcal{D}_i^1$ and $\mathcal{D}_i^2$, both of which are attacked in the experiments.

Similar to our findings in the one-step case, the proposed sampling attack can significantly deteriorate the two-step adaptation performance of meta policies. Since Intermittent and Persistent Attacks are based on stochastic gradient descent, and the objective function is nonconvex-nonconcave, the two

**Algorithm 4** SG-MARL-2

**Input** Initialization $\theta_0$, $t = 0$, step size $\alpha, \eta_1$.
**while** not converge **do**
  Draw a batch of i.i.d tasks $\mathcal{T}_i, i \in \mathcal{I}$ with size $|\mathcal{I}| = I$;
  **for** $i \in \mathcal{I}$ **do**
    Sample a batch of trajectories $\mathcal{D}_i^1$ under $q_i(\cdot; \theta_t)$;
    $\bar{\theta}_{t+1}^i = \theta_t + \alpha \hat{\nabla} J_i(\theta_t, \mathcal{D}_i^1)$ ;
    Sample a batch of trajectories $\mathcal{D}_i^2$ under $q_i(\cdot; \bar{\theta}_{t+1}^i)$;
    $\theta_{t+1}^i = \bar{\theta}_{t+1}^i + \alpha \hat{\nabla} J_i(\bar{\theta}_{t+1}^i, \mathcal{D}_i^2)$
    Sample a batch of trjectories $\mathcal{D}_i^3$ under $q_i(\cdot; \theta_{t+1}^i)$;
  Compute $\nabla_\theta^i MC$ using $\mathcal{D}_i^3$;
  $\theta_{t+1} = \theta_t + (\eta_1 / I) \sum_{i \in \mathcal{I}} \nabla_{\theta_t}^i MC$;
**Return** $\theta_{t+1}$

Table 2: Experimental evaluations of Intermittent Attack and Persistent Attack under ISA and CSA when meta RL uses two-step gradient adaptation. Average cumulative rewards (and standard deviations) are chosen as the evaluation metric, which is the sample mean of $L(\theta)$, $\theta$ being the meta policies corrupted by different attacks. Intermittent and Persistent Attacks significantly deteriorate the two-step adaption performance of meta policies.

| Task | Attack Model | Intermittent Attack | Persistent Attack | Benchmark |
|---|---|---|---|---|
| 2D | ISA | $-33.75 \pm 6.24$ | $-362.07 \pm 28.97$ | $-6.41 \pm 0.62$ |
| | CSA | $-141.45 \pm 3.96$ | $-151.20 \pm 36.60$ | |
| H-C | ISA | $-91.92 \pm 10.85$ | $-127.17 \pm 8.04$ | $-61.86 \pm 8.10$ |
| | CSA | $-94.11 \pm 10.82$ | $-104.53 \pm 9.46$ | |

attack schemes do not return the same optimal attack $\delta^*$ or the same corrupted meta policy in general. Table 3 summarizes optimal attacks $\delta^*$ under different attack settings. Finally, we comment on the robust training idea. As discussed in Section 6, decreasing the learner's step size turns the minimax into maxmin. For example, the maxmin problem in ISA is

$$\max_{\theta \in \mathbb{R}^d} \min_{\delta \in \Delta} \mathbb{E}_{i \sim p} \mathbb{E}_{\mathcal{D} \sim q_i(\cdot;\theta)} [J_i(\theta + \alpha \cdot \delta \hat{\nabla} J(\theta, \mathcal{D}))].$$

In this case, the learner aims to maximize its training performance under sampling attacks, which does not necessarily lead to the maximization of the testing performance. It is not guaranteed that the meta policy obtained through the robust training would achieve satisfying adaptation performance in the testing phase. Table 4 summarizes the robust training experiments where $\eta_1 = 0.01, \eta_2 = 0.1$. Some encouraging results (shown in bold in Table 4) do appear in some experiments where the obtained meta policies achieve comparable adaptation performance as the benchmark ones. However, the robust training does not succeed[2] in the half-cheetah task under CSA (the blank entries in Table 4). This maxmin training remains largely a heuristic, and more efforts are needed to investigate defense mechanisms for meta learning thoroughly.

## D Regularity Assumptions

In this section, we provide further justifications for the regularity conditions assumed in this paper. In our analysis, it is assumed that the policy gradient $\nabla J_i(\theta)$ is bounded as stated in Assumption 3.

**Assumption 3** (Bounded Policy Gradient)**.** *There exists a constant $G$, such that for any trajectory $\tau$ and policy parameter $\theta$, $\|g_i(\tau; \theta)\| \leq G$. Equivalently, for any $\theta$ and any batch of trajectories $\mathcal{D}$, $\|\nabla J_i(\theta)\| \leq G$, $\|\hat{\nabla} J_i(\theta, \mathcal{D})\| \leq G$.*

[2]We run multiple robust training experiments in half-cheetah, yet not every obtained meta policy shows adaptation performance comparable to the benchmark ones.

Table 3: A summary of optimal attacks $\delta^*$ under different attack settings.

| Task | gradient step | Attacl Model | Intermittent Attack | Persistent Attack |
|---|---|---|---|---|
| 2D | 1 | ISA | -0.24 | -39.56 |
| | | CSA | -12.96 | -38.58 |
| | 2 | ISA | 2.78 | -2.36 |
| | | CSA | -0.29 | -2.60 |
| H-C | 1 | ISA | 1.11 | 0.98 |
| | | CSA | -5.94 | 0.89 |
| | 2 | ISA | 5.81 | 0.06 |
| | | CSA | 4.23 | 0.66 |

Table 4: A summary of adaption performance of robust meta policy against Persistent Attack.

| Task | gradient step | Attacl Model | Persistent Attack | Robust Training | Benchmark |
|---|---|---|---|---|---|
| 2D | 1 | ISA | $-298.77 \pm 34.09$ | $\mathbf{-11.50 \pm 0.57}$ | $-10.92 \pm 0.85$ |
| | | CSA | $-488.97 \pm 31.06$ | $\mathbf{-11.34 \pm 0.48}$ | |
| | 2 | ISA | $-362.07 \pm 28.97$ | $-33.67 \pm 1.34$ | $-6.41 \pm 0.62$ |
| | | CSA | $-151.20 \pm 36.60$ | $-26.26 \pm 1.41$ | |
| H-C | 1 | ISA | $-101.13 \pm 10.57$ | $\mathbf{-63.26 \pm 8.69}$ | $-63.41 \pm 6.86$ |
| | | CSA | $-93.65 \pm 9.81$ | $---$ | |
| | 2 | ISA | $-127.17 \pm 8.04$ | $-75.09 \pm 9.29$ | $-61.04 \pm 8.10$ |
| | | CSA | $-104.53 \pm 9.46$ | $---$ | |

Moreover, the value function $f(\delta, \theta)$ is assumed to Lipschitz smooth in Assumption 4.

**Assumption 4** (Lipschitz Gradients)**.** *The function $f$ is continuously differentiable in both $\delta$ and $\theta$. Furthermore, there exists constants $L_{11}, L_{12}$ and $L_{22}$ such that for every $\delta, \delta_1, \delta_2$ and $\theta, \theta_1, \theta_2$,*

$$\begin{aligned} &\|\nabla_\delta f(\delta_1, \theta) - \nabla_\delta f(\delta_2, \theta)\| \le L_{11}\|\delta_1 - \delta_2\|, \\ &\|\nabla_\theta f(\delta, \theta_1) - \nabla_\theta f(\delta, \theta_2)\| \le L_{22}\|\theta_1 - \theta_2\|, \\ &\|\nabla_\delta f(\delta, \theta_1) - \nabla_\delta f(\delta, \theta_2)\| \le L_{12}\|\theta_1 - \theta_2\|, \\ &\|\nabla_\theta f(\delta_1, \theta) - \nabla_\theta f(\delta_2, \theta)\| \le L_{12}\|\delta_1 - \delta_2\|. \end{aligned} \tag{7}$$

The two assumptions are direct extensions of [Fallah et al., 2020, Lemma 1] in the context of adversarial MAMRL, which is established on customary assumptions (e.g., bounded rewards) in the policy gradient literature Agarwal et al. [2019]. On the other hand, the unbiasedness and finite variance of the stochastic gradient assumed in Assumption 5 is also valid in MAMRL, which has been proved in Fallah et al. [2020], in order to show the convergence of Algorithm 1.

**Assumption 5** (Stochastic Gradients)**.** *The stochastic gradient $\hat{\nabla} f(\delta, \theta, \xi)$, with $\xi$ being the random variable corresponding to the batch of trajectories with batch size $M$, satisfies*

$$\begin{aligned} &\mathbb{E}[\hat{\nabla} f(\delta, \theta, \xi) - \nabla f(\delta, \theta)] = 0 \\ &\mathbb{E}[\|\hat{\nabla} f(\delta, \theta, \xi) - \nabla f(\delta, \theta)\|^2] \le \sigma^2 / M. \end{aligned} \tag{8}$$

Independent of the above assumptions regarding the regularity of the value function $f(\delta, \theta)$, we require that the admissible set of attacks $\delta$ is also regular as stated in Assumption 6, which helps the derivation of complexity results.

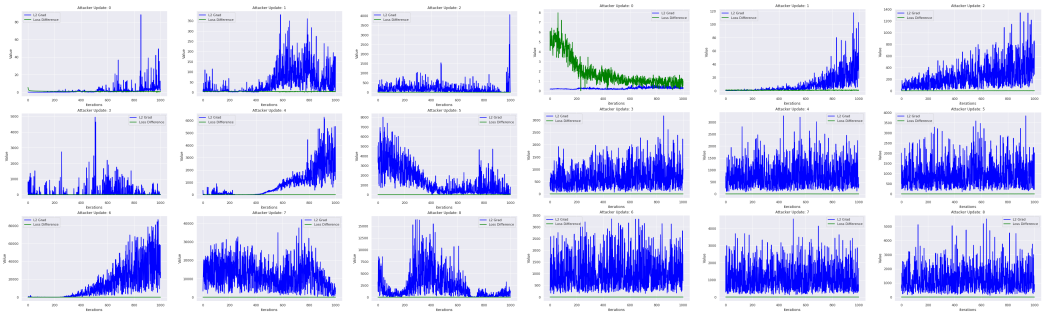
**Assumption 6.** *The set $\Delta$ is convex and compact, with its diameter upper bounded by $D \in \mathbb{R}_+$. Without loss of generality, it is assumed that $D \ge 1$.*

In addition to the assumptions discussed above, PL condition in Assumption 1 and RSI condition in Assumption 2 play an important role in showing the convergence of the proposed attack mechanisms.

Given their relationships with respect to strong convexity(SC) Karimi et al. [2016]:

$$\text{SC} \Rightarrow \text{RSI} \Rightarrow \text{PL},$$

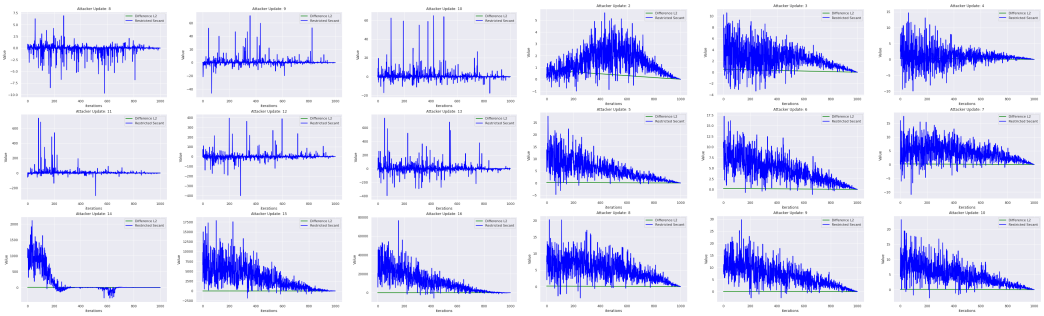
this work extends the convergence results of Gradient Descent Ascent [Karimi et al., 2016, Lin et al., 2019] to nonconvex situations under weaker conditions. The key point is that the linear contraction of $\mathbb{E}[\|\theta_t^* - \theta_t\|^2]$ (see Lemma E4), previously established under strong convexity Lin et al. [2019], also holds true under Assumption 2. Our theoretical treatment on nonconvex-nonconcave minimax problems can be helpful for future studies on this topic. Even though it is unclear under which condition, the value function of RL or meta RL will satisfy PL or RSI, we provide numerical evidence in Figure 4 and Figure 5, demonstrating their validity in the experiments.



(a) The comparison between $\|\nabla_\theta f(\delta_t, \theta_t)\|$ and $(\max_\theta f(\delta_t, \theta) - f(\delta_t, \theta_t))$ in 2D navigation

(b) The comparison between $\|\nabla_\theta f(\delta_t, \theta_t)\|$ and $(\max_\theta f(\delta_t, \theta) - f(\delta_t, \theta_t))$ in Half-cheetah

Figure 4: Numerical Justifications of Assumption 1. As the meta learning proceeds, the norm of gradients (blue curves) are above the loss difference (green curves).



(a) The comparison between $\langle \nabla_\theta f(\delta_t, \theta_t), \theta^*(\delta) - \theta_t \rangle$ and $\|\theta^* - \theta_t\|^2$ in 2D navigation

(b) The comparison between $\langle \nabla_\theta f(\delta_t, \theta_t), \theta^*(\delta) - \theta_t \rangle$ and $\|\theta^* - \theta_t\|^2$ in Half-cheetah

Figure 5: Numerical Justifications of Assumption 1. As the meta learning proceeds, the restricted secant $\langle \nabla_\theta f(\delta_t, \theta_t), \theta^*(\delta) - \theta_t \rangle$ (blue curves) are above the $\ell_2$ difference (green curves) up to a constant.

# E Proofs

## E.1 Proofs in Section 4

*Proof of Proposition 1.* As shown in [Fallah et al., 2020, lemma 1], under Assumption 3, for any $\theta_1, \theta_2 \in \mathbb{R}^d$, we have

$$|J_i(\theta_1) - J_i(\theta_2)| \leq C(G)\|\theta_1 - \theta_2\|,$$

where $C(G)$ is a constant depending on $G$. Fix a $\delta \in \Delta$, and let $\theta_1 = \theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D})$, $\theta_2 = \theta^* + \alpha \cdot \delta\widehat{\nabla} J_i(\theta^*, \mathcal{D})$, the above inequality leads to

$$\begin{aligned}|J_i(\theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D})) - J_i(\theta^* + \alpha \cdot \delta\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| &\leq C(G)\alpha\|\delta - 1\|\|\widehat{\nabla} J_i(\theta^*, \mathcal{D})\| \\ &\leq C(\alpha, G)\|\delta - 1\|,\end{aligned}$$

where the second inequality holds by the boundedness of $\|\widehat{\nabla} J_i(\theta^*, \mathcal{D})\|$ assumed in Assumption 3. Combining triangle inequality and the above one yields (3). Taking expectations on both sides of (3), and further taking the minimization of the left-hand side, we obtain

$$\begin{aligned}\min_{\delta\in\Delta} \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta^* + \alpha\hat{\nabla} J_i(\theta^*, \mathcal{D}))] \leq \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta^* + \alpha \cdot \delta\hat{\nabla} J(\theta^*, \mathcal{D}))] \\ + C(\alpha, G)\|\delta - 1\|, \qquad \forall \delta.\end{aligned}$$

Finally, taking the minimization of the right-hand side of the above inequality leads to (4). □

**Extensions to CSA**

**Proposition 2.** *Under Assumption 3 and Assumption 6, there exists a constant $C$ such that for $\theta^* = \theta^*(\delta)$ defined in* (CSA) *and any batch of trajectories $\mathcal{D}$,*

$$J_i(\theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D})) \leq \delta J_i(\theta^* + \alpha \cdot \delta\widehat{\nabla} J_i(\theta^*, \mathcal{D})) + C|\delta - 1|. \tag{E1}$$

*Hence,*

$$\min_{\delta\in\Delta} \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[J_i(\theta^* + \alpha\hat{\nabla} J_i(\theta^*, \mathcal{D}))] \tag{E2}$$

$$\leq \min_{\delta\in\Delta} \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[\delta J_i(\theta^* + \alpha \cdot \delta\hat{\nabla} J(\theta^*, \mathcal{D}))] + C|\delta - 1|. \tag{E3}$$

*Proof.* Note that

$$\begin{aligned}&|J_i(\theta^* + \alpha\hat{\nabla} J_i(\theta^*, \mathcal{D})) - \delta J_i(\theta^* + \alpha \cdot \delta\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| \\ &\leq |J_i(\theta^* + \alpha\hat{\nabla} J_i(\theta^*, \mathcal{D})) - \delta J_i(\theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| \\ &\quad + |\delta J_i(\theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D})) - \delta J_i(\theta^* + \alpha \cdot \delta\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| \\ &\leq |\delta - 1||J_i(\theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| + |\delta|C(\alpha, G)|\delta - 1|,\end{aligned}$$

where in the last inequality we use the following result obtained in the proof of Proposition 1

$$|J_i(\theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D})) - J_i(\theta^* + \alpha \cdot \delta\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| \leq C(\alpha, G)\|\delta - 1\|.$$

Let $\|r\|_\infty := \max_{s,a} |r(s, a)|$ (assumed to be bounded). According to the definition of the value function $J_i$, the horizon length $H$, and the discounting factor $\gamma$, we obtain

$$|J_i(\theta^* + \alpha\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| \leq \frac{1 - \gamma^H}{1 - \gamma}\|r\|_\infty.$$

Since $\delta \in \Delta$, $|\delta|$ can be controlled by the diameter $D$. Hence, we obtain

$$|J_i(\theta^* + \alpha\hat{\nabla} J_i(\theta^*, \mathcal{D})) - \delta J_i(\theta^* + \alpha \cdot \delta\widehat{\nabla} J_i(\theta^*, \mathcal{D}))| \leq C(\alpha, G, \|r\|_\infty, D)|\delta - 1|,$$

leading to (E1). Following the same argument in the proof of Proposition 1, we obtain (E3). □

The above proposition leads to the MMP formulation of CSA in below.

$$\min_{\delta\in\Delta} \max_{\theta\in\mathbb{R}^d} \mathbb{E}_{i\sim p}\mathbb{E}_{\mathcal{D}\sim q_i(\cdot;\theta)}[\delta J_i(\theta + \alpha \cdot \delta\hat{\nabla} J(\theta, \mathcal{D}))].$$

Since $\Delta$ is a bounded compact convex set, our nonasymptotic convergence analysis on Intermittent and Persistent Attacks under ISA also applies to CSA using the above MMP formulation. Finally, we comment on the relationship between MMP and ISA(CSA). The original Stackelberg formations in (ISA) and (CSA) target the testing performance, while in (MMP), the objective is the training performance. Hence, the two are different problems, even though the minimax values serves as the upper bounds for (ISA) and (CSA). When the access to the testing performance is not available during the training phase, this minimax upper bound helps guide the attacker's update.

### E.2 Proofs in Section 5

In this section, we present the full version of Theorem 1 with detailed choices of step sizes $\eta_1, \eta_2$ and batch size $M$. We begin with some technical lemmas.

**Lemma E1.** *Under Assumption 1, Assumption 4 and Assumption 5, there exists a* $\theta^* \in \arg\max_\theta f(\delta, \theta)$*, such that*

$$\nabla\Phi(\delta) = \nabla_\delta f(\delta, \theta^*) = \mathbb{E}[\hat{\nabla} f(\delta, \theta^*; \xi)].$$

*Moreover,* $\Phi(\delta)$ *is L-Lipschitz smooth with the Lipschitz constant* $L := L_{11} + \frac{L_{12}^2}{2\mu}$*,*

$$\|\nabla\Phi(\delta_1) - \nabla\Phi(\delta_2)\| \leq L\|\delta_2 - \delta_1\|.$$

*Proof of Lemma E1.* Our first step is to show that for any $\delta_1, \delta_2$ and $\theta_1 \in \arg\max_\theta f(\delta_1, \theta)$, there exists $\theta_2 \in \arg\max_\theta f(\delta, \theta)$ such that $\|\theta_1 - \theta_2\| \leq \frac{L_{12}}{2\mu}\|\delta_1 - \delta_2\|$.

First, based on the assumption of Lipschitz gradients in (7),

$$\|\nabla_\theta f(\delta_2, \theta_1) - \nabla_\theta f(\delta_1, \theta_1)\| \leq L_{12}\|\delta_2 - \delta_1\|,$$

which is equivalent to $\|\nabla_\theta f(\delta_2, \theta_1)\| \leq L_{12}\|\delta_2 - \delta_1\|$, since $\nabla_\theta f(\delta_1, \theta_1) = 0$. Then, using the PL condition in (1), we obtain

$$\max_\theta f(\delta_2, \theta) - f(\delta_2, \theta_1) \leq \frac{1}{2\mu}\|\nabla_\theta f(\delta_2, \theta_1)\|^2,$$

which implies that

$$\Psi_{\delta_2}(\theta_1) - \min_\theta \Psi_{\delta_2}(\theta) \leq \frac{L_{12}^2}{2\mu}\|\delta_2 - \delta_1\|^2. \tag{E4}$$

As shown in [Karimi et al., 2016, Theorem 2], PL condition implies the quadratic growth, that is, if $\Psi_\delta(\theta)$ satisfies $\mu$-PL condition, the following holds true

$$\Psi_\delta(\theta) - \min_\theta \Psi_\delta(\theta) \geq 2\mu\|\theta - \theta^*\|^2, \tag{E5}$$

where $\theta^* \in \arg\min_\theta \Psi_\delta(\theta)$ and $\|\theta - \theta^*\| = \min_{\theta' \in \arg\min_\theta \Psi_\delta(\theta)} \|\theta - \theta'\|$. In other words, $\theta^*$ is the projection of $\theta$ onto the set of optimal solution, and $\|\theta - \theta^*\|$ is the distance of the point $\theta$ to the optimal solution set.

Finally, let $\theta_2$ be the projection of $\theta_1$ onto the set $\{\theta' | \theta' \in argmax_\theta f(\delta_2, \theta)\}$, then combining (E4) and (E5) leads to the desired result

$$\|\theta_1 - \theta_2\| \leq \frac{L_{12}}{2\mu}\|\delta_1 - \delta_2\|. \tag{E6}$$

The second step is to show the existence of $\nabla\Phi(\delta)$ and $\nabla\Phi(\delta) = \nabla_\delta f(\delta, \theta^*)$, which is straightforward from [Nouiehed et al., 2019, lemma A.5]. Finally, we show that $\Phi(\delta)$ is also Lipschitz smooth because of Assumption 4. Notice that

$$\begin{aligned}
\|\nabla\Phi(\delta_1) - \nabla\Phi(\delta_2)\| &= \|\nabla_\delta f(\delta_1, \theta_1) - \nabla_\delta f(\delta_2, \theta_2)\| \\
&= \|\nabla_\theta f(\delta_1, \theta_1) - \nabla_\delta f(\delta_2, \theta_1) + \nabla_\delta f(\delta_2, \theta_1) - \nabla_\theta f(\delta_2, \theta_2)\| \\
&\leq L_{11}\|\delta_1 - \delta_2\| + L_{12}\|\theta_1 - \theta_2\|,
\end{aligned}$$

together with (E6), we have

$$\|\nabla\Phi(\delta_1) - \nabla\Phi(\delta_2)\| \leq (L_{11} + \frac{L_{12}^2}{2\mu})\|\delta_1 - \delta_2\|,$$

showing that $\Phi(\delta)$ is also Lipschitz smooth with the Lipschitz constant being $L_{11} + \frac{L_{12}^2}{2\mu}$. □

The above lemma, as an extension of [Nouiehed et al., 2019, Lemma A.1], serves as a substitute of Danskin's theorem in convex analysis [Bernhard and Rapaport, 1995], which states that the gradient of $\max_\theta f(\delta, \theta)$ can be directly evaluated using the gradient $\nabla_\delta f(\delta, \theta^*)$. Moreover, such gradient function $\nabla\Phi$ is Lipschitz smooth, reducing the attacker's convergence problem to a standard analysis of gradient descent of nonconvex objectives.

Our next lemma shows that $\theta_K$ returned by the inner loop properly approximates $\theta^*(\delta)$ in the sense that $\nabla_\delta f(\delta_t, \theta_t^K) \approx \nabla\Phi(\delta_t)$, where $\theta_t^K := \theta_K(\delta_t)$ denotes the final iterate of the inner loop under $\delta_t$. The proof of the following lemma relies on the assumption that there exists a constant $B$, such that $\Phi(\delta_t) - f(\delta_t, \theta_0) \le B$ holds for every $t$, which can be verified using Lemma E1, as shown in [Nouiehed et al., 2019, Lemma A.6]

**Lemma E2.** *Let $\rho = 1 - \frac{2\mu}{L_{22}}$, $\bar{L} = \max\{L_{12}, L_{22}, 1, \Phi_\infty\}$, where $\Phi_\infty := \max\{\max_\delta \|\nabla\Phi\|, 1\}$. Assume that there exists a constant $B > 0$ such that $\Phi(\delta_t) - f(\delta_t, \theta_0) \le B$. For any $\epsilon > 0$, if $K$ and $M$ are large enough such that*

$$K \ge \frac{1}{\log \rho^{-1}} \left( 4 \log \epsilon^{-1} + \log \frac{2^4 D^2 (2\Phi_\infty + LD)^2 \bar{L}^2 B}{L^2 \mu} \right)$$
$$M \ge \frac{24\sigma^2 D^2 (2\Phi_\infty + LD)^4}{2\bar{L} L^2 \epsilon^4}$$

*then for $z_t := \nabla_\delta f(\delta_t, \theta_t^K) - \nabla\Phi(\delta_t)$,*

$$\mathbb{E}[\|z_t\|] \le \frac{L\epsilon^2}{4D(2\Phi_\infty + LD)^2} \tag{E7}$$

*and*

$$\mathbb{E}[\|\nabla_\theta f(\delta_t, \theta_t^K)\|] \le \epsilon. \tag{E8}$$

*Proof of Lemma E2.* Since $f(\delta, \theta)$ is assumed to be Lipschitz-smooth, $-f(\delta, \theta)$ is also Lipschitz-smooth, and we have

$$-f(\delta_t, \theta_t^K) \le -f(\delta_t, \theta_{t-1}^K) - \left\langle \nabla_\theta f(\delta_t, \theta_t^{K-1}), \theta_t^K - \theta_t^{K-1} \right\rangle + \frac{L_{22}}{2} \|\theta_t^K - \theta_t^{K-1}\|^2.$$

Plugging the updating rule into the above inequality, we obtain

$$-f(\delta_t, \theta_t^K) \le -f(\delta_t, \theta_{t-1}^K) - \frac{1}{L_{22}} \left\langle \nabla_\theta f(\delta_t, \theta_t^{K-1}), \widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}, \xi_t^{K-1}) \right\rangle + \frac{1}{2L_{22}} \|\widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}, \xi_t^{K-1})\|^2,$$

which implies

$$\mathbb{E}[\Phi(\delta_t) - f(\delta_t, \theta_t^K) | \theta_t^{K-1}] \le \Phi(\delta_t) - f(\delta_t, \theta_t^{K-1}) - \frac{1}{L_{22}} \|\nabla_\theta f(\delta_t, \theta_t^{K-1})\|^2 + \frac{1}{2L_{22}} \mathbb{E}[\|\widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}, \xi_t^{K-1})\|^2 | \theta_t^{K-1}].$$

By Assumption 5, we have

$$\begin{aligned}
\mathbb{E}[\|\widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}, \xi_t^{K-1})\|^2 | \theta_t^{K-1}] &= \mathbb{E}[\|\widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}, \xi_t^{K-1}) - \nabla_\theta f(\delta_t, \theta_t) + \nabla_\theta f(\delta_t, \theta_t)\|^2 | \theta_t^{K-1}] \\
&= \mathbb{E}[\|\widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}) - \nabla_\theta f(\delta_t, \theta_t)\|^2 | \theta_t^{K-1}] + \mathbb{E}[\|\nabla_\theta f(\delta_t, \theta_t)\|^2 | \theta_t^{K-1}] \\
&\quad + 2\mathbb{E}[\left\langle \widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}, \xi_t^{K-1}) - \nabla_\theta f(\delta_t, \theta_t), \nabla_\theta f(\delta_t, \theta_t) \right\rangle | \theta_t^{K-1}] \\
&\le 3\mathbb{E}[\|\widehat{\nabla}_\theta f(\delta_t, \theta_t^{K-1}) - \nabla_\theta f(\delta_t, \theta_t)\|^2 | \theta_t^{K-1}] + \frac{3}{2} \mathbb{E}[\|\nabla_\theta f(\delta_t, \theta_t)\|^2 | \theta_t^{K-1}] \\
&\le \frac{3\sigma^2}{M} + \frac{3}{2} \mathbb{E}[\|\nabla_\theta f(\delta_t, \theta_t)\|^2 | \theta_t^{K-1}]
\end{aligned}$$

where the second last inequality follows from Young's inequality. Combing the two inequalities above, we arrive at

$$\begin{aligned}
\mathbb{E}[\Phi(\delta_t) - f(\delta_t, \theta_t^K) | \theta_t^{K-1}] &\le \Phi(\delta_t) - f(\delta_t, \theta_t^{K-1}) - \frac{1}{4L_{22}} \|\nabla_\theta f(\delta_t, \theta_t^{K-1})\|^2 + \frac{3\sigma^2}{2L_{22}M} \\
&\le (1 - \frac{2\mu}{L_{22}})(\Phi(\delta_t) - f(\delta_t, \theta_t^{K-1})) + \frac{3\sigma^2}{2L_{22}M}.
\end{aligned}$$

Recursively applying the above inequality, and let $\rho = 1 - \frac{2\mu}{L_{22}} < 1$, we obtain

$$\mathbb{E}[\Phi(\delta_t) - f(\delta_t, \theta_t^K)] \leq \rho^K(\Phi(\delta_t) - f(\delta_t, \theta_0)) + \left(\frac{1-\rho^K}{1-\rho}\right)\frac{3\sigma^2}{2L_{22}M}$$

As shown in [Nouiehed et al., 2019], there exists a constant $B$, such that $\Phi(\delta_t) - f(\delta_t, \theta_0) \leq B$ holds for every $t$. Then, using the quadratic growth shown in the proof of Lemma E1, we have

$$\mathbb{E}[\|\theta_t^K - \theta_t^*\|^2] \leq \frac{B}{2\mu}\rho^K + \left(\frac{1-\rho^K}{1-\rho}\right)\frac{3\sigma^2}{2L_{22}M}$$

Hence, with Jensen inequality and the choice of $K$ and $M$

$$\begin{aligned}
\mathbb{E}[\|z_t\|] &= \mathbb{E}[\|\nabla_\delta f(\delta_t, \theta_t^K) - \nabla\Phi(\delta_t)\|] \\
&\leq L_{12}\mathbb{E}[\|\theta_t^K - \theta_t^*\|] \\
&\leq L_{12}\sqrt{\frac{B}{2\mu}\rho^K + \left(\frac{1-\rho^K}{1-\rho}\right)\frac{3\sigma^2}{2L_{22}M}} \\
&\leq \frac{L\epsilon^2}{4D(2\Phi_\infty + LD)^2},
\end{aligned}$$

which proves (E7). To prove (E8), note that

$$\begin{aligned}
\mathbb{E}[\|\nabla_\theta f(\delta_t, \theta_t^K)\|] &= \mathbb{E}[\|\nabla_\theta f(\delta_t, \theta_t^K) - \nabla_\theta f(\delta_t, \theta_t^*)\|] \\
&\leq L_{22}\mathbb{E}[\|\theta_t^K - \theta_t^*\|] \leq \frac{L\epsilon^2}{4D(2\Phi_\infty + LD)^2} \leq \epsilon.
\end{aligned}$$

□

We are now ready to present the full version of Theorem 1 and its proof in the following.

**Theorem E1** (Full version of Theorem 1)**.** *Under Assumption 1, Assumption 4 and Assumption 5, for any given $\epsilon \in (0, 1)$, let the step sizes be chosen as $\eta_1 = 1/L_{22}, \eta_2 = 1/L$, if $K$, $N$ and $M$ are large enough,*

$$N \geq N(\epsilon) \sim \mathcal{O}(\epsilon^{-2}), \qquad K \geq K(\epsilon) \sim \mathcal{O}(\log \epsilon^{-1}), \qquad M \geq M(\epsilon) \sim \mathcal{O}(\epsilon^{-4})$$

*then there exists an index $t$ such that $(\delta_t, \theta_t^K)$ is an $\epsilon$-FOSP.*

*Proof of Theorem 1.* Due to projection,

$$\left\langle \delta_t - \frac{1}{L}\nabla_\delta f(\delta_t, \theta_t^K, \xi_t^K) - \delta_{t+1}, \delta - \delta_{t+1} \right\rangle \leq 0, \quad \forall \delta \in \Delta.$$

Let $\delta = \delta_t$, and take expectations on both sides, we have

$$\mathbb{E}[\left\langle \nabla_\delta f(\delta_t, \theta_t^K), \delta_{t+1} - \delta_t \right\rangle | \delta_t, \theta_t^K] \leq -L\mathbb{E}[\|\delta_t - \delta_{t+1}\|^2 | \delta_t, \theta_t^K],$$

which leads to

$$\begin{aligned}
\mathbb{E}[\langle \nabla_\delta f(\delta_t, \theta_t^*), \delta_{t+1} - \delta_t \rangle | \delta_t, \theta_t^K] &\leq \mathbb{E}[\left\langle \nabla_\delta f(\delta_t, \theta_t^*) - \nabla_\delta f(\delta_t, \theta_t^K), \delta_{t+1} - \delta_t \right\rangle | \delta_t, \theta_t^K] - L\mathbb{E}[\|\delta_t - \delta_{t+1}\|^2 | \delta_t, \theta_t^K] \\
&= \mathbb{E}[\langle z_t, \delta_t - \delta_{t+1} \rangle | \delta_t, \theta_t^K] - L\mathbb{E}[\|\delta_t - \delta_{t+1}\|^2 | \delta_t, \theta_t^K]. \quad \text{(E9)}
\end{aligned}$$

On the other hand, since $\Phi$ is L-Lipschitz smooth,

$$\begin{aligned}
\mathbb{E}[\Phi(\delta_{t+1})] &\leq \mathbb{E}[\Phi(\delta_t)] + \mathbb{E}[\langle \nabla_\delta f(\delta_t, \theta_t^*), \delta_{t+1} - \delta_t \rangle] + \frac{L}{2}\mathbb{E}[\|\delta_{t+1} - \delta_t\|^2] \\
&\leq \mathbb{E}[\Phi(\delta_t)] + \mathbb{E}[\langle z_t, \delta_t - \delta_{t+1} \rangle] - \frac{L}{2}\mathbb{E}[\|\delta_{t+1} - \delta_t\|^2], \quad \text{(E10)}
\end{aligned}$$

where the last inequality holds because of Equation (E9) and law of total probability.

Also by projection,

$$\begin{aligned}
\mathbb{E}[\langle \nabla_\delta f(\delta_t, \theta_t^K), \delta - \delta_{t+1} \rangle | \delta_t, \theta_t^K] &\geq L\mathbb{E}[\langle \delta_t - \delta_{t+1}, \delta - \delta_{t+1} \rangle | \delta_t, \theta_t^K] \\
&\geq \mathbb{E}[\langle \nabla_\delta f(\delta_t, \theta_t^K), \delta_{t+1} - \delta_t \rangle | \delta_t, \theta_t^K] + L\mathbb{E}[\langle \delta_t - \delta_{t+1}, \delta - \delta_{t+1} \rangle | \delta_t, \theta_t^K] \\
&\geq \mathbb{E}[\langle \nabla_\delta f(\delta_t, \theta_t^K - \Phi(\delta_t), \delta_{t+1} - \delta_t) \rangle | \delta_t, \theta_t^k] \\
&\quad + \mathbb{E}[\langle \nabla \Phi(\delta_t), \delta_{t+1} - \delta_t \rangle | \delta_t, \theta_t^k] + L\mathbb{E}[\langle \delta_t - \delta_{t+1}, \delta - \delta_{t+1} \rangle | \delta_t, \theta_t^k] \\
&\geq -\|z_t\| \mathbb{E}[\|\delta_{t+1} - \delta_t\| | \delta_t, \theta_t^k] - \|\nabla \Phi(\delta_t)\| \mathbb{E}[\|\delta_{t+1} - \delta_t\| | \delta_t, \theta_t^k] \\
&\quad - L\mathbb{E}[\|\delta - \delta_{t+1}\| \|\delta_{t+1} - \delta_t\| | \delta_t, \theta_t^k] \\
&\geq -(\Phi_\infty + LD + \|z_t\|) \mathbb{E}[\|\delta_{t+1} - \delta_t\| | \delta_t, \theta_t^K], \qquad \text{(E11)}
\end{aligned}$$

where the last inequality is obtained using our assumptions that $\Phi_\infty = \max\{\max_\delta \|\nabla \Phi\|, 1\}$, and the diameter of the set $\Delta$ is $D$.

Notice that by the choice of $K$ in Lemma E2,

$$\begin{aligned}
\mathbb{E}[\|z_t\|] &\leq L_{12} \mathbb{E}[\|\theta_t^K - \theta_t^*\|] \\
&\leq \frac{L\epsilon^2}{4D(2\Phi_\infty + LD)^2} \leq \Phi_\infty,
\end{aligned}$$

which reduces (E11) to the following

$$\mathbb{E}[\langle \nabla_\delta f(\delta_t, \theta_t^K), \delta - \delta_{t+1} \rangle] \geq -(2\Phi_\infty + LD) \mathbb{E}[\|\delta_{t+1} - \delta_t\|].$$

Let $e_t = -\mathbb{E}[\langle \nabla_\delta f(\delta_t, \theta_t^K), \delta - \delta_{t+1} \rangle]$, then we have

$$\mathbb{E}[\|\delta_{t+1} - \delta_t\|] \geq \frac{e_t}{2\Phi_\infty + LD}, \qquad \text{(E12)}$$

and combining (E12) with (E10) yields

$$\mathbb{E}[\Phi(\delta_{t+1})] - \mathbb{E}[\Phi(\delta_t)] \leq -\frac{L}{2} \frac{e_t^2}{(2\Phi_\infty + LD)^2} + D\|z_t\|.$$

Therefore, we have

$$\Phi(\delta_0) - \mathbb{E}[\Phi(\delta_N)] \geq -\frac{L}{2(2\Phi_\infty + LD)^2} \sum_{t=0}^{N-1} e_t^2 - D \sum_{t=0}^{N-1} \mathbb{E}[\|z_t\|].$$

Denote $D_\Phi := \Phi(\delta_0) - \min_\delta \Phi(\cdot)$, and the the above inequality can be rewritten as

$$\frac{2D_\phi(2\Phi_\infty + LD)^2}{LN} + \frac{2D(2\Phi_\infty + LD)^2}{LN} \sum_{t=0}^{N-1} \mathbb{E}[\|z_t\|].$$

Let $N \geq \frac{4D_\Phi(2\Phi_\infty + LD)^2}{L\epsilon^2}$, and then by Lemma E2, we obtain

$$\frac{1}{N} \sum_{t=0}^{N-1} e_t^2 \leq \epsilon^2,$$

which implies that there exists one index $t$ for which $e_t \leq \epsilon$. This completes the proof. □

### E.3 Proofs in Section 5

This section presents the full version of Theorem 2 with detailed choices of step sizes $\eta_1, \eta_2$ and batch size $M$, and the main results rest on the following lemmas.

**Lemma E3.** *The iterates $\{\delta_t\}$ generated by Algorithm 3 satisfies the following inequality,*

$$\begin{aligned}
\mathbb{E}[\Phi(\delta_{t+1})] \leq \mathbb{E}[\Phi(\delta_t)] &- (\frac{\eta_2}{2} - 2L\eta_2^2) \mathbb{E}[\|\nabla \Phi(\delta_t)\|^2] \\
&+ (\frac{\eta_2}{2} + 2L\eta_2^2) \mathbb{E}[\|\nabla \Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2] + \frac{L\eta_2^2 \sigma^2}{M}.
\end{aligned} \qquad \text{(E13)}$$

*Proof of Lemma E3.* Since $\Phi(\delta)$ is $L$-smooth (see Lemma E1), we have

$$\Phi(\delta_{t+1}) - \Phi(\delta_t) - (\delta_{t+1} - \delta_t)\nabla\Phi(\delta_t) \leq L/2\|\delta_{t+1} - \delta_t\|^2.$$

In Algorithm 3, gradient descent is used to update $\delta_t$, hence $\delta_{t+1} - \delta_t = \eta_2\hat{\nabla}_\delta f(\delta_t, \theta_t; \xi_t)$, and the above inequality can be rewritten as

$$\begin{aligned}\Phi(\delta_{t+1}) &\leq \Phi(\delta_t) - \eta_2\hat{\nabla}_\delta f(\delta_t, \theta_t; \xi_t)^\mathsf{T}\nabla\Phi(\delta_t) + \frac{L}{2}\|\delta_{t+1} - \delta_t\|^2 \\ &= \Phi(\delta_t) + \eta_2(\nabla\Phi(\delta_t) - \hat{\nabla}_\delta f(\delta_t, \theta_t; \xi_t))^\mathsf{T}\nabla\Phi(\delta_t) \\ &\quad - \eta_2\|\nabla\Phi(\delta_t)\|^2 + \frac{L\eta_2^2}{2}\|\hat{\nabla}_\delta f(\delta_t, \theta_t, \xi_t)\|^2.\end{aligned}$$

By taking a conditional expectation on both sides, we have

$$\begin{aligned}\mathbb{E}[\Phi(\delta_{t+1})|\delta_t, \theta_t] &\leq \Phi(\delta_t) + \eta_2(\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t))^\mathsf{T}\nabla\Phi(\delta_t) \\ &\quad - \eta_2\|\nabla\Phi(\delta_t)\|^2 + \frac{L\eta_2^2}{2}\mathbb{E}[\|\hat{\nabla}_\delta f(\delta_t, \theta_t, \xi_t)\|^2|\delta_t, \theta_t]. \end{aligned} \tag{E14}$$

With Cauchy-Schwartz inequality,

$$\|\hat{\nabla}_\delta f(\delta_t, \theta_t, \xi_t)\|^2 \leq 2\|\hat{\nabla}_\delta f(\delta_t, \theta_t, \xi_t) - \nabla f(\delta_t, \theta_t)\|^2 + 2\|\nabla_\delta f(\delta_t, \theta_t)\|^2 \tag{E15}$$

$$\|\nabla_\delta f(\delta_t, \theta_t)\|^2 \leq 2\|\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2 + 2\|\nabla\Phi(\delta_t)\|^2. \tag{E16}$$

Applying Young's inequality to the second term of the right-hand side in (E14), we have

$$(\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t))^\mathsf{T}\nabla\Phi(\delta_t) \leq \frac{\|\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2 + \|\nabla\Phi(\delta_t)\|^2}{2}. \tag{E17}$$

Therefore, plugging (E15),(E16) and (E17) into (E14) leads to the following

$$\begin{aligned}&\mathbb{E}[\Phi(\delta_{t+1})|\delta_t, \theta_t] \\ &\leq \Phi(\delta_t) + \frac{\eta_2}{2}\|\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2 - \frac{\eta_2}{2}\|\nabla\Phi(\delta_t)\|^2 \\ &+ L\eta_2^2\mathbb{E}[\|\hat{\nabla}_\delta f(\delta_t, \theta_t; \xi_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2|\delta_t, \theta_t] + L\eta_2^2\|\nabla_\delta f(\delta_t, \theta_t)\|^2 \\ &\leq \Phi(\delta_t) + (\frac{\eta_2}{2} + 2L\eta_2^2)\|\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2 \\ &- (\frac{\eta_2}{2} - 2L\eta_2^2)\|\nabla\Phi(\delta_t)\|^2 + L\eta_2^2\mathbb{E}[\|\hat{\nabla}_\delta f(\delta_t, \theta_t; \xi_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2|\delta_t, \theta_t].\end{aligned}$$

Finally, with Assumption 5 and law of total probability, we obtain the desired inequality □

**Lemma E4.** *Under Assumption 2, Assumption 4 and Assumption 5, for the iterates generated by Algorithm 3, let $d_t := \mathbb{E}[\|\theta_t^* - \theta_t\|^2]$, where $\theta_t^* \in \arg\max_\theta f(\delta_t, \theta)$, and let $\kappa = 2\bar{L}/\mu^2$, where $\bar{L}$ is defined in Lemma E2, then*

$$d_t \leq \left(1 - \frac{1}{2\kappa} + 16\kappa^3\bar{L}^2\eta_2^2\right)d_{t-1} + 16\kappa^3\eta_2^2\mathbb{E}[\|\nabla\Phi(\delta_{t-1})\|^2] + \frac{8\kappa^3\eta_2^2\sigma^2}{M} + \frac{\mu^2\sigma^2}{L_{22}^2 M} \tag{E18}$$

*Proof of Lemma E4.* According to the gradient update with respect to $\theta$,

$$\langle\theta_{t-1} + \eta_1\nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1}) - \theta_t, \theta - \theta_t\rangle \leq 0, \forall\theta.$$

Let $\theta = \theta_{t-1}^*$, then the above inequality leads to

$$\begin{aligned}&\left\langle\theta_t - \theta_{t-1} - \eta_1\nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1}), \theta_{t-1}^* - \theta_t\right\rangle \geq 0 \\ \Leftrightarrow &\underbrace{\left\langle\theta_{t-1}^* - \theta_{t-1} - \eta_1\nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1}), \theta_{t-1}^* - \theta_t\right\rangle}_{(*)} \geq \|\theta_{t-1}^* - \theta_t\|^2.\end{aligned} \tag{E19}$$

Denote the right-hand side of the above inequality by $(*)$, then

$$\begin{aligned}(*) &= \left\langle\theta_{t-1}^* - \theta_{t-1} - \eta_1\nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1}), \theta_{t-1}^* - \theta_{t-1} - \eta_1\nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1})\right\rangle \\ &= \|\theta_{t-1}^* - \theta_{t-1}\|^2 - 2\eta_1\left\langle\theta_{t-1}^* - \theta_{t-1}, \nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1})\right\rangle + \eta_1^2\|\nabla\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1})\|^2.\end{aligned} \tag{E20}$$

By Assumption 2, we have

$$\left\langle \nabla_\theta f(\delta_{t-1}, \theta_{t-1}), \theta^*_{t-1} - \theta_{t-1} \right\rangle \geq \mu \|\theta^*_{t-1} - \theta_{t-1}\|^2. \tag{E21}$$

Taking expectations on both sides of (E20) and plugging (E21) yields

$$\mathbb{E}[(*)] \leq (1 - 2\mu\eta_1)\mathbb{E}[\|\theta^*_{t-1} - \theta_{t-1}\|^2] + \eta_1^2 \mathbb{E}[\|\nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1})\|^2]. \tag{E22}$$

Similar to the argument in the proof of Lemma E2, by Young's inequality, Assumption 5 and Assumption 4,

$$\begin{aligned}\mathbb{E}[\|\nabla_\theta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1})\|^2] &\leq \frac{2\sigma^2}{M} + 2\|\nabla_\theta f(\delta_{t-1}, \theta_{t-1})\|^2 \\ &\leq \frac{2\sigma^2}{M} + 2L_{22}\|\theta^*_{t-1} - \theta_{t-1}\|^2.\end{aligned}$$

Hence, plug the above inequality and (E22) into (E19) after taking expectations, we obtain

$$\mathbb{E}[\|\theta^*_{t-1} - \theta_t\|^2] \leq (1 - 2\mu\eta_1 + 2L_{22}\eta_1^2) d_{t-1} + \frac{2\sigma^2\eta_1^2}{M}.$$

Let $\kappa = \frac{2\bar{L}}{\mu^2}$ and with the choice of $\eta_1 = \frac{\mu}{2L_{22}}$, the above inequality can be rewritten as

$$\mathbb{E}[\|\theta^*_{t-1} - \theta_t\|^2] \leq (1 - \frac{1}{\kappa}) d_{t-1} + \frac{\sigma^2\mu^2}{2L_{22}^2 M}. \tag{E23}$$

By Young's inequality

$$\begin{aligned} d_t &= \mathbb{E}[\|\theta^*_t - \theta_t\|^2] = \mathbb{E}[\|\theta^*_t - \theta^*_{t-1} + \theta^*_{t-1} - \theta_t\|^2] \\ &\leq \left(1 + \frac{1}{2(\max\{\kappa, 2\} - 1)}\right) \mathbb{E}[\|\theta^*_{t-1} - \theta_t\|^2] + (1 + 2(\max\{\kappa, 2\} - 1))\mathbb{E}[\|\theta^*_t - \theta^*_{t-1}\|^2] \\ &\leq \left(\frac{2\max\{\kappa, 2\} - 1}{2\max\{\kappa, 2\} - 2}\right) \mathbb{E}[\|\theta^*_{t-1} - \theta_t\|^2] + 4\kappa\mathbb{E}[\|\theta^*_t - \theta^*_{t-1}\|^2] \\ &\leq \left(1 - \frac{1}{2\kappa}\right) d_{t-1} + 4\kappa\mathbb{E}[\|\theta^*_t - \theta^*_{t-1}\|^2] + \frac{\sigma^2\mu^2}{L_{22}^2 M}. \end{aligned}$$

By the Lipschitz continuity of $\theta^*$ shown in (E6), $\|\theta^*_t - \theta^*_{t-1}\| \leq \frac{L_{12}}{2\mu}\|\delta_t - \delta_{t-1}\| \leq \kappa\|\delta_t - \delta_{t-1}\|$. Without loss of generality, it is assumed that $\mu$ are small enough. Furthermore, according to the stochastic gradient ascent, we have

$$\begin{aligned}\mathbb{E}[\|\delta_t - \delta_{t-1}\|^2] &= \eta_2^2 \mathbb{E}[\|\nabla \delta f(\delta_{t-1}, \theta_{t-1}, \xi_{t-1})\|^2] \\ &\leq 4\bar{L}^2\eta_2^2 d_{t-1} + 4\eta_2^2\mathbb{E}[\|\nabla\Phi(\delta_{t-1})\|^2] + \frac{2\sigma^2\eta_2^2}{M}.\end{aligned}$$

Putting these pieces together leads to (E18) in Lemma E4 □

**Lemma E5.** *Under Assumption 2, Assumption 4 and Assumption 5, for the iterates generated by Algorithm 3, and for $d_t$ defined in Lemma E4,*

$$\mathbb{E}[\Phi(\delta_{t+1})] \leq \mathbb{E}[\Phi(\delta_t)] - \frac{7\eta_2}{16}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2] + \frac{9\bar{L}^2\eta_2 d_t}{16} + \frac{L\eta_2^2\sigma^2}{M}. \tag{E24}$$

*Proof of Lemma E5.* By the choice of $\eta_2 = \min\{1/32L, 1/8\kappa^2\bar{L}\}$,

$$\frac{7}{16}\eta_2 \leq \frac{\eta_2}{2} - 2L\eta_2^2 \leq \frac{\eta_2}{2} + 2L\eta_2^2 \leq \frac{9}{16}\eta_2,$$

which reduces (E13) to

$$\begin{aligned}\mathbb{E}[\Phi(\delta_{t+1})] \leq \mathbb{E}[\Phi(\delta_t)] &- \frac{7\eta_2}{16}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2] \\ &+ \frac{9\eta_2}{16}\mathbb{E}[\|\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2] + \frac{L\eta_2^2\sigma^2}{M}.\end{aligned}$$

Note that

$$\mathbb{E}[\|\nabla\Phi(\delta_t) - \nabla_\delta f(\delta_t, \theta_t)\|^2] \leq \bar{L}^2\mathbb{E}[\|\theta^*_t - \theta_t\|^2] = \bar{L}^2 d_t,$$

Combining the above inequalities, we arrive at (E24). □

Lemma E5 implies that the difference $\mathbb{E}[\Phi(\delta_N)] - \mathbb{E}[\Phi(\delta_0)]$ is controlled by the sum of $d_t$, that is,

$$\mathbb{E}[\Phi(\delta_N)] - \mathbb{E}[\Phi(\delta_0)] \leq -\mathcal{O}(\eta_2)\left(\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2]\right) + \mathcal{O}(\eta_2)\sum_{t=0}^{N-1} d_t.$$

Since $d_t$ exhibits a linear contraction as shown in Lemma E4, $\sum_{t=0}^{N-1} d_t$ can be further controlled by $\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2]$. Eventually, one can show that $\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2]$ is bounded, implying the convergence of Algorithm 3. The full version of Theorem 2 is stated as follows.

**Theorem E2.** *Under Assumption 2, Assumption 4 and Assumption 5, for any given $\epsilon \in (0,1)$, let the step sizes be chosen as $\eta_1 = \mu/L_{22}$, $\eta_2 = \min\{1/32L, 1/8\kappa^2\bar{L}\}$, then if $N$ and $M$ are large enough,*

$$N \geq \left(\frac{(8\kappa^2+32)\bar{L}D_\Phi + \kappa D^2\bar{L}^2}{\epsilon^2}\right), \qquad M \geq \mathcal{O}(\frac{\sigma^2\kappa}{\epsilon^2}),$$

*there exists an index $t$, such that $(\delta_t, \theta_t)$ is an $\epsilon$-FOSP.*

*Proof of Theorem 2.* Let $\gamma = 1 - \frac{1}{2\kappa} + 16\kappa^3\bar{L}^2\eta_2^2$. Applying (E18) recursively yields

$$d_t \leq \gamma^t D^2 + 16\kappa^3\eta_2^2\left(\sum_{j=0}^{t-1}\mathbb{E}[\|\nabla\Phi(\delta_j)\|^2]\right) + \left(\frac{8\kappa^3\eta_2^2\sigma^2}{M} + \frac{\mu^2\sigma^2}{L_{22}^2 M}\right)\left(\sum_{j=0}^{t-1}\gamma^{t-1-j}\right). \quad \text{(E25)}$$

Combining (E24) and (E25), we obtain

$$\begin{aligned}\mathbb{E}[\Phi(\delta_t)] \leq \mathbb{E}[\Phi(\delta_{t-1})] &- \frac{7\eta_2}{16}\mathbb{E}[\|\nabla\Phi(\delta_{t-1})\|^2] + \frac{9\bar{L}^2\eta_2}{16}\gamma^{t-1}D^2 \\ &+ 9\bar{L}^2\kappa^3\eta_2^3\left(\sum_{j=0}^{t-2}\gamma^{t-2-j}\mathbb{E}[\|\nabla\Phi(\delta_j)\|^2]\right) + \frac{9\bar{L}^2\eta_2}{16}\left(\frac{8\kappa^3\eta_2^2\sigma^2}{M} + \frac{\mu^2\sigma^2}{L_{22}^2M}\right)\left(\sum_{j=0}^{t-2}\gamma^{t-2-j}\right) \\ &+ \frac{L\eta_2^2\sigma^2}{M}. \end{aligned} \quad \text{(E26)}$$

Applying (E26) recursively leads to

$$\begin{aligned}\mathbb{E}[\Phi(\delta_N)] \leq \Phi(\delta_0) &- \frac{7\eta_2}{16}\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2] + \frac{9\bar{L}^2\eta_2 D^2}{16}\left(\sum_{t=0}^{N-1}\gamma^t\right) \\ &+ \frac{NL\eta_2^2\sigma^2}{M} + 9\bar{L}^2\kappa^3\eta_2^3\left(\sum_{t=1}^{N-1}\gamma^{t-2-j}\sum_{j=0}^{t-2}\mathbb{E}[\|\nabla\Phi(\delta_j)\|^2]\right) \\ &+ \frac{9\bar{L}^2\eta_2}{16}\left(\frac{8\kappa^3\eta_2^2\sigma^2}{M} + \frac{\mu^2\sigma^2}{L_{22}^2M}\right)\left(\sum_{t=1}^{N}\sum_{j=0}^{t-2}\gamma^{t-2-j}\right).\end{aligned}$$

By the choice of $\eta_2$, $\gamma \leq 1 - \frac{1}{4\kappa}$, and hence $\sum_{t=0}^{N-1}\gamma^t \leq 4\kappa$, which implies

$$\begin{aligned}\sum_{t=1}^{N-1}\gamma^{t-2-j}\sum_{j=0}^{t-2}\mathbb{E}[\|\nabla\Phi(\delta_j)\|^2] &\leq 4\kappa\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2], \\ \sum_{t=1}^{N}\sum_{j=0}^{t-2}\gamma^{t-2-j} &\leq 4\kappa N.\end{aligned}$$

Combining all the inequalities above, and suppressing the constants in front of variance terms, we arrive at

$$\mathbb{E}[\Phi(\delta_N)] \leq \Phi(\delta_0) - \frac{47\eta_2}{128}\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2] + \mathcal{O}(\frac{\eta_2\sigma^2\kappa N}{M}) + \mathcal{O}(\kappa\eta_2 D^2\bar{L}^2).$$

By the definition of $D_\Phi$, the above inequality leads to

$$\frac{1}{N}\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2] \leq \frac{128D_\Phi}{47\eta_2 N} + \mathcal{O}(\frac{\kappa D^2\bar{L}^2}{N}) + \mathcal{O}(\frac{\sigma^2\kappa}{M})$$
$$\leq \mathcal{O}(\frac{(8\kappa^2+32)\bar{L}D_\Phi + \kappa D^2\bar{L}^2}{N}) + \mathcal{O}(\frac{\sigma^2\kappa}{M}).$$

Since $N \geq \mathcal{O}\left(\frac{(8\kappa^2+32)\bar{L}D_\Phi+\kappa D^2\bar{L}^2}{\epsilon^2}\right)$, and $M \geq \mathcal{O}(\frac{\sigma^2\kappa}{\epsilon^2})$,

$$\frac{1}{N}\sum_{t=0}^{N-1}\mathbb{E}[\|\nabla\Phi(\delta_t)\|^2] \leq \epsilon^2.$$

Following the same argument in the proof of Theorem 1, we arrive at the conclusion. □