Decentralized Federated Policy Gradient with Byzantine Fault-Tolerance and Provably Fast Convergence

Philip Jordan
ETH Zürich
Switzerland
jordanph@ethz.ch

Florian Grötschla
ETH Zürich
Switzerland
fgroetschla@ethz.ch

Flint Xiaofeng Fan
National University of Singapore
Singapore
fxf@u.nus.edu

Roger Wattenhofer
ETH Zürich
Switzerland
wattenhofer@ethz.ch

ABSTRACT

In *Federated Reinforcement Learning (FRL)*, agents aim to collaboratively learn a common task, while each agent is acting in its local environment without exchanging raw trajectories. Existing approaches for FRL either (a) do not provide any fault-tolerance guarantees (against misbehaving agents), or (b) rely on a trusted central agent (a single point of failure) for aggregating updates. We provide thefi rst decentralized Byzantine fault-tolerant FRL method. Towards this end, wefi rst propose a new centralized Byzantine fault-tolerant policy gradient (PG) algorithm that improves over existing methods by relying only on assumptions standard for non-fault-tolerant PG. Then, as our main contribution, we show how a combination of robust aggregation and Byzantine-resilient agreement methods can be leveraged in order to eliminate the need for a trusted central entity. Since our results represent thefi rst sample complexity analysis for Byzantine fault-tolerant decentralized federated non-convex optimization, our technical contributions may be of independent interest. Finally, we corroborate our theoretical results experimentally[1] for common RL environments, demonstrating the speed-up of decentralized federations w.r.t. the number of participating agents and resilience against various Byzantine attacks.

KEYWORDS

Federated Reinforcement Learning; Policy Gradient; Byzantine Fault-Tolerance; Decentralization; Robust Optimization

ACM Reference Format:
Philip Jordan, Florian Grötschla, Flint Xiaofeng Fan, and Roger Wattenhofer. 2024. Decentralized Federated Policy Gradient with Byzantine Fault-Tolerance and Provably Fast Convergence. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 25 pages.

[1]Our code is available at https://github.com/philip-jordan/decentralized-byzantine-RL.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Many real-world reinforcement learning (RL) systems consist of a group of agents (e.g. afl eet of autonomous vehicles), in which all agents aim to learn the same task, each in its local environment. Since RL models often suffer from poor sample complexity, collaboration is highly desirable. However, as in the autonomous driving example, trajectories of environment interactions may be made up of large amounts of video and sensor data (too large to transfer between agents), and possibly also with privacy restrictions. This motivates the need for distributed algorithms that can leverage the power of collaboration without sharing raw trajectories. In the broader machine learning context, this setting has been widely studied under the name *Federated Learning* (FL) [1, 2], and has inspired *Federated Reinforcement Learning* (FRL) [3–6] as an analogous concept in RL.

Policy gradient methods are among the most popular algorithms in model-free RL. Existing work studies how to generalize such approaches to FRL in a fully trusted setting [7]. In many practical situations, however, there may not be any guarantee on the trustworthiness of information provided by the participating agents, be it due to e.g. communication failure, or malicious attempts at trying to prevent the system from learning. Methods that tolerate the presence of some fraction of Byzantine agents have previously been proposed and demonstrated to defend against some attacks in practice [5]. As we are going to discuss in related work, so far, FRL algorithms need additional non-standard assumptions regarding gradient estimation noise.

Moreover, a crucial limitation of previous methods for achieving Byzantine fault-tolerance is the need for one trusted party responsible for aggregating updates,fi ltering out potentially malicious inputs, and broadcasting results back to all participants. Introducing such a single point of failure seems like a high price to pay for achieving Byzantine resilience—and is going against the very idea of a trustless and robust design. Hence we are aiming for a decentralized system, i.e., a system in which Byzantine behavior of *any* participant can be tolerated, as long as only a reasonable number of such bad actors occur simultaneously. Algorithms achieving both Byzantine fault-tolerance and decentralization have previously been proposed for general non-convex optimization [8] but analyzed only w.r.t. infinite-time asymptotic behavior. We propose a novel method and givefi nite-time sample complexity guarantees for decentralized federated PG with Byzantine fault-tolerance.

More concretely, **our contributions** can be summarized as follows:

- As a starting point, we provide a new centralized Byzantine fault-tolerant federated PG algorithm ByzPG, and prove competitive sample complexity guarantees under assumptions that are standard in non-fault-tolerant PG literature. In particular, unlike previous approaches, we do not rely on deterministic bounds on gradient estimation noise.
- For our main contribution, we extend the above (centralized) approach to the significantly more challenging decentralized setting: With DecByzPG, we propose a decentralized Byzantine fault-tolerant PG method. To the best of our knowledge, this is thefi rst decentralized Byzantine fault-tolerant algorithm for non-convex optimization with sample complexity guarantee.
- Technically, we leverage the favorable interplay of fault-tolerant aggregation and agreement mechanisms that so far have only been studied in separation. Key to our analysis is a novel lemma on the concentration of random parameter vectors that helps control the bias incurred from agreement.
- To corroborate our theoretical results regarding both ByzPG and DecByzPG experimentally, we demonstrate speed-up as more agents join a federation, as well as tolerance against different Byzantine attack scenarios.

The rest of this paper is organized as follows: In Section 2, we provide the necessary background on RL while covering related work on FRL and fault-tolerance. Section 3 introduces our setup regarding communication, the Byzantine failure model, and technical assumptions needed for the sample complexity analysis. As a warm-up, in Section 4, we introduce our centralized algorithm ByzPG which in Section 5 is generalized to the decentralized DecByzPG. Finally, our experiments are described and evaluated in Section 6.

2 BACKGROUND & RELATED WORK

Reinforcement learning (RL) and policy gradient (PG). The RL setup is commonly modeled as a Markov Decision Process (MDP, see also [9]) $\mathcal{M} := \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho\}$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $\mathcal{P}(s' \mid s,a)$, and reward $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto [\,0, R]$ where $R > 0$, $\gamma \in (0,1)$ is the discount factor, and ρ is the initial state distribution. Let π denote the policy controlling an agent's behavior, i.e., $\pi(a \mid s)$ is the probability that the agent chooses action a at state s. A trajectory $\tau := \{s_0, a_0, \ldots, s_{H-1}, a_{H-1}\}$ is a sequence of state-action pairs followed by an agent according to a stationary policy π, where $s_0 \sim \rho$. We define $\mathcal{R}(\tau) := \sum_{t=0}^{H-1} \gamma^t \mathcal{R}(s_t, a_t)$ as the cumulative discounted reward for a trajectory τ. Note that here we study episodic MDPs withfi xed trajectory horizon H.

PG methods are popular in model-free RL [10, 11]. Compared to deterministic value-function-based methods, PG is more effective when applied to tasks with high-dimensional or infinite action spaces. Let π_θ denote the policy parameterized by $\theta \in \mathbb{R}^d$, and $p(\tau \mid \pi_\theta)$ the trajectory distribution induced by policy π_θ. The expected discounted future reward when following π_θ is given by $J(\theta) := \mathbb{E}_{\tau \sim p(\cdot|\theta)} [\mathcal{R}(\tau) \mid \mathcal{M}]$ whose gradient w.r.t. θ is

$$\begin{aligned} \nabla_\theta J(\theta) &= \int_\tau \mathcal{R}(\tau) \nabla_\theta p(\tau \mid \theta)\, d\tau \\ &= \mathbb{E}_{\tau \sim p(\cdot|\theta)} [\nabla_\theta \log p(\tau \mid \theta) \mathcal{R}(\tau) \mid \mathcal{M}]\,. \end{aligned} \tag{1}$$

Hence we can use gradient ascent in order to optimize $J(\theta)$ over $\theta \in \mathbb{R}^d$. Since (1) involves computing an integral over all possible trajectories, we typically use stochastic gradient ascent. In each iteration, a batch of trajectories $\{\tau_i\}_{i=1}^M$ is sampled at the current policy θ. Then, the policy is updated by $\theta \leftarrow \theta + \eta \widehat{\nabla}_M J(\theta)$, where η is the step size and $\widehat{\nabla}_M J(\theta)$ is an estimate of (1) based on the sampled trajectories $\{\tau_i\}_{i=1}^M$: $\widehat{\nabla}_M J(\theta) = \frac{1}{M} \sum_{i=1}^M \nabla_\theta \log p(\tau_i \mid \theta) \mathcal{R}(\tau_i)$. Commonly used policy gradient estimators, e.g. REINFORCE [12] and GPOMDP [13], can be written as

$$\widehat{\nabla}_M J(\theta) = \frac{1}{M} \sum_{i=1}^M g(\tau_i \mid \theta)$$

where $\tau_i = \{s_0^i, a_0^i, \ldots, s_{H-1}^i, a_{H-1}^i\}$ and $g(\tau_i \mid \theta)$ is an unbiased estimate of $\nabla_\theta \log p(\tau_i \mid \theta) \mathcal{R}(\tau_i)$. For more details on gradient estimation and sampling, we refer to Appendix A.1.

Non-convex optimization. Despite PG's additional challenges of non-stationarity and the non-finite-sum structure, improvements in convergence results in non-convex optimization have generally led to similar progress for optimizing the non-concave $J(\theta)$ in PG. The $O(\varepsilon^{-4})$ sample complexity for reaching an ε-stationary point, i.e., θ such that $\mathbb{E}[\|\nabla J(\theta)\|^2] \leq \epsilon^2$, of SGD [14] and vanilla PG [15] has been lowered to $O(\varepsilon^{-10/3})$ by SVRG, and SVRPG [16] respectively. These methods rely on an inner loop that reuses old gradient estimates for reduced variance which in the case of SVRPG is implemented via importance sampling. The recently proposed PAGE estimator [17], and its PG adaptation PAGE-PG [18], replace the inner loop by a probabilistic switch, lowering the sample complexity to $O(\varepsilon^{-3})$.

Fault-tolerance. Byzantine fault-tolerance [19] has long been established as the strongest notion of resilience against arbitrary failure or deliberate manipulation of distributed systems. Regarding previous work in the federated optimization literature, we distinguish between the rather common *centralized*, and the far less studied *decentralized*, sometimes called collaborative, setting.

(a) *Centralized:* In the presence of a trusted coordinator, Byzantine fault-tolerant non-convex optimization has been widely studied—with approaches differing mostly in terms offilter-ing techniques and problem assumptions [20–22]. We refer to [23] for an overview of such commonly used Byzantine-resilient methods for aggregating potentially malicious updates at a central server. Regarding Byzantine-tolerant PG, [5] shows promising empirical results. However, theoretical guarantees are proven only under deterministic noise bounds which makes results difficult to appreciate in comparison with non-fault-tolerant methods that do not rely on such assumptions. Recently, [24] has proposed a non-convex optimization algorithm leveraging the favorable interplay of certain robust aggregators and the above-mentioned variance-reduced PAGE estimator. Our centralized ByzPG extends their ideas into the PG setting, with a modified algorithm and tightened analysis.

(b) *Decentralized:* In the PG context, there is no previous work studying decentralized Byzantine-tolerant methods. More generally, [8] proposes a fault-tolerant algorithm for decentralized non-convex optimization. While convergence is only proven in an infinite-time asymptotic sense, their notion of averaging agreement is shown to be of crucial importance for decentralized learning. Indeed, the notion of ϵ-approximate Byzantine agreement on d-dimensional inputs had previously been proposed [25, 26]. However, unlike averaging agreement, such methods show poor applicability in our setting since the fraction of tolerable Byzantines goes to zero as d increases.

3 SETUP AND ASSUMPTIONS

3.1 Distributed Computing Setup

Communication is assumed to happen in a round-based, synchronous, all-to-all manner among K agents, and the exchange of raw trajectories is prohibited. In particular, our algorithms will only involve sending current values of local policy parameters and respective gradients.

In order to model both system failure as well as malicious agent behavior, we tolerate a fraction of Byzantine agents, in particular:

Assumption 1 (Byzantine agents). *Let $\alpha_{\max} = 1/2$ in the centralized setting, and $\alpha_{\max} = 1/4$ in the decentralized setting, respectively. Denote by $\mathcal{H}_t \subset [K]$ the set of honest (i.e. non-Byzantine) agents in iteration t of an algorithm. Then there exist $\alpha, \bar{\epsilon} > 0$ such that $\bar{\alpha} := \alpha + \bar{\epsilon} < \alpha_{\max}$ and for all t, $|\mathcal{H}_t| \geq (1-\alpha)K$.*

We point out that $\mathcal{H}_t$ may be different for each iteration t, hence it is of no use for any agent to remember past communication in order to infer who might be Byzantine.

Instead of sending updates as prescribed by our algorithms, Byzantine agents may send arbitrary values. In particular, these values may be chosen by an omniscient entity with access to all information (e.g. agents' local state, messages that have been sent, the definition of the algorithm, who is Byzantine, etc.) and controlling all Byzantine agents. This means Byzantine agents may collude or base their behavior on any other non-public information. However, Byzantine agents are not omnipotent, e.g. they cannot interfere in communication between honest agents by changing or delaying messages. Moreover, we assume that Byzantines cannot alter local state, not even their own state. In the centralized case, this assumption does not change anything, since our algorithm ByzPG only maintains cross-iteration state at the trusted central agent. In the decentralized case, however, corrupted local state may otherwise be passed on from a Byzantine agent to an honest agent across iterations. Note also that in particular, any agent not sending messages in the required format or omitting updates, potentially due to failure of the communication network, can be modeled as Byzantine.

3.2 Reinforcement Learning Assumptions

Our theoretical analysis aims to bound the required number of sampled trajectories required per agent in order to reach an ϵ-stationary solution. In the centralized case, this refers to the central agentfinding $\theta \in \mathbb{R}^d$ such that $\|\nabla J(\theta)\| \leq \epsilon$ which can then be broadcast to all participants. A generalized solution concept for the decentralized setting is presented in Section 5.

In the following, we state the set of assumptions our analysis is based on, which is standard in the study of PG, see e.g. [16, 18, 27, 28]. In particular, we do not require a more restrictive version of Assumption 4 made in [5]. Hence, our sample complexity results are amenable to comparison with non-fault tolerant counterparts.

Note that we are assuming homogeneity of all agents' local environments, and all agents hence share the same objective $J(\cdot)$.

Assumption 2 (Log-policy gradient norm). *For any $a \in \mathcal{A}$ and $s \in \mathcal{S}$, there exists a constant $G > 0$ such that for any $\theta \in \mathbb{R}^d$ we have $\|\nabla_\theta \log \pi_\theta(a \mid s)\| \leq G$.*

Assumption 3 (Log-policy smoothness). *For any $\theta \in \mathbb{R}^d$, π_θ is twice differentiable, and for any $a \in \mathcal{A}$ and $s \in \mathcal{S}$, there exists a constant $M > 0$ such that $\left\|\nabla^2_\theta \log \pi_\theta(a \mid s)\right\| \leq M$.*

Assumption 4 (Gradient estimator variance). *There exists a constant $\sigma > 0$ such that for any $\theta \in \mathbb{R}^d$, we have $\mathrm{Var}\left[g(\tau \mid \theta)\right] = \mathbb{E}\|g(\tau \mid \theta) - \nabla J(\theta)\|^2 \leq \sigma^2$.*

Assumption 5 (Importance weight variance). *For any policy pair $\theta_a, \theta_b \in \mathbb{R}^d$ and $\tau \sim p(\cdot \mid \theta_b)$, the importance weight $\omega(\tau \mid \theta_b, \theta_a) = \frac{p(\tau|\theta_a)}{p(\tau|\theta_b)}$ is well-defined. In addition, there exists a constant $W > 0$ such that $\mathrm{Var}\left[\omega(\tau \mid \theta_b, \theta_a)\right] \leq W\|\theta_a - \theta_b\|^2$.*

For completeness, we restate the following commonly used proposition from [16].

Proposition1. *Under the above assumptions 2, 3, 4, and 5, with $g(\tau \mid \theta)$ denoting the REINFORCE or GPOMDP gradient estimator, we have for all $\theta, \theta_1, \theta_2 \in \mathbb{R}^d$:*

1. *$\|g_k(\tau \mid \theta)\| \leq C_g$ with $C_g = HG(R + |C_b|)/(1-\gamma)$ and C_b is the baseline reward,*
2. *$\|g(\tau \mid \theta_1) - g(\tau \mid \theta_2)\| \leq L_g \|\theta_1 - \theta_2\|$ with $L_g = HM(R + |C_b|)/(1-\gamma)$, and*
3. *$J(\theta)$ is L-smooth with $L = HR(M + HG^2)/(1-\gamma)$.*

4 CENTRALIZED BYZANTINE-TOLERANT FEDERATED POLICY GRADIENT

In this section, we describe ByzPG, given by Algorithm 1, our centralized method for Byzantine fault-tolerant PG. This also serves as a warm-up for introducing parts of our method that are going to reappear in Section 5. Note that in Algorithm 1, $\mathrm{Be}(p)$ denotes a Bernoulli distribution with success probability p and $\mathcal{U}(S)$ denotes a uniform distribution over afi nite set S.

4.1 Method

Instead of the usual inner loop seen in variance-reduced methods such as SVRPG [27], we probabilistically switch between update types, as in PAGE-PG [18]. Concretely, in each iteration, we either (a) sample a large batch of N trajectories at θ_t for gradient estimation, or (b) sample a small batch of B trajectories and use a variance-reduced estimator incorporating the previous iteration's gradient estimate, and employ importance sampling to correct for non-stationarity. For details on gradient estimation, importance

Algorithm 1 BYZPG at server agent

input: $\theta_0 \in \mathbb{R}^d$, large batch size N, small batch size B, step size η, probability $p \in (0,1]$
for $t = 0$ **to** $T-1$ **do**
  $c_t \leftarrow$ **Sample** (Be (p))
  if $c_t = 1$ or $t = 0$ **then**
    for worker agent $k \in [K]$ in parallel **do**
      sample trajectories $\{\tau_{t,i}^{(k)}\}_{i=1}^N$ from $p(\cdot \mid \theta_t)$
      $\widetilde{v}_t^{(k)} = \frac{1}{N}\sum_{i=1}^N g(\tau_{t,i}^{(k)} \mid \theta_t)$
    $v_t \leftarrow$ **Aggregate** $\left(\langle \widetilde{v}_t^{(k)} \rangle_{k=1}^K\right)$ ▷ $\widetilde{v}_t^{(k)}$ received from worker agent k, $\forall k \in [K]$
  else
    sample trajectories $\{\tau_{t,i}\}_{i=1}^B$ from $p(\cdot \mid \theta_t)$
    $v_t = \frac{1}{B}\sum_{i=1}^B g(\tau_{t,i} \mid \theta_t) + v_{t-1} - \frac{1}{B}\sum_{i=1}^B g^{\omega_{\theta_t}}(\tau_{t,i} \mid \theta_{t-1})$
  $\theta_{t+1} = \theta_t + \eta v_t$ ▷ broadcast θ_{t+1} to worker agents
output: $\theta_{\widehat{T}}$ with $\widehat{T} \sim \mathcal{U}([T])$

sampling, and the definition of $g^{\omega_{\theta_t}}(\tau_{t,i} \mid \theta_{t-1})$ we refer to Appendix A.1. Note that with probability p we use (a), and (b) otherwise—except for thefi rst iteration, in which only (a) is well-defined. Furthermore, (a) is performed at all worker agents in parallel which have previously received the current parameters θ_t from the server agent. Then, the individual estimates $\widetilde{v}_t^k$ are aggregated at the server via the **Aggregate** subroutine described below. In the case of (b), we sample and estimate the gradient only at the server, hence there is no need for aggregation.

We point out that loopless variance-reduction has previously been used in conjunction with Byzantine fault-tolerant aggregation in [24]'s BYZ-VR-MARINA. However, BYZPG distinguishes itself in two ways:

(1) BYZPG only samples at workers in case (a) while BYZ-VR-MARINA does so in either case. Our analysis suggests that unlike in case (a), in (b), the bias introduced by Byzantine filtering outweighs the benefits from the reduced variance of the aggregated sample. In our PG setting, this modification is key to achieving sample complexity competitive with non-fault-tolerant methods.
(2) BYZ-VR-MARINA is designed for general non-convexfinite-sum optimization. BYZPG handles the additional challenges of non-stationarity and not having access to the full gradient by relying on importance sampling and switching between a large and small batch size.

The following notion of robust aggregation specifying our requirements on the **Aggregate** subroutine is adopted from [24] and hasfi rst appeared in a similar form in [29].

DEFINITION 1 (ROBUST AGGREGATION). *Let $C_{ra} > 0$ and $\alpha \in [0, 1/2)$. A function **Aggregate** $: (\mathbb{R}^d)^K \to \mathbb{R}^d$ is an (α, C_{ra})-robust aggregator if for any tuple of inputs $\langle \theta^{(k)} \rangle_{k\in[K]}$ with $\theta^{(k)} \in \mathbb{R}^d$, and for any $\mathcal{H} \subseteq K$ with $|\mathcal{H}| \geq (1-\alpha)K$, denoting $\hat{\theta} :=$ **Aggregate**$(\theta^{(1)}, \ldots, \theta^{(K)})$, it holds that*

$$\mathbb{E}\left[\|\hat{\theta} - \bar{\theta}\|^2\right] \leq \frac{C_{ra}\alpha}{|\mathcal{H}|(|\mathcal{H}| - 1)} \sum_{i,l \in \mathcal{H}} \mathbb{E}\left[\|\theta^{(i)} - \theta^{(l)}\|^2\right]$$

where $\bar{\theta} := \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \theta^{(i)}$. Expectations are taken over the randomness of the input.

Known implementations satisfying Definition 1 are discussed in Appendix A.2. In particular, there exist (α, C_{ra})-robust aggregators for constant C_{ra} and any $\alpha \in [0, 1/2)$.

4.2 Convergence Analysis and Sample Complexity

Next, we present the convergence guarantees for BYZPG, with proofs deferred to Appendix B.

THEOREM1. *Let Assumptions 2, 3, 4, and 5 hold. Suppose **Aggregate** is an (α, C_{ra})-robust aggregator with constant $C_{ra} > 0$ and α satisfying Assumption 1. Then the following holds for the output of BYZPG, i.e., Algorithm 1: For $\eta = \Theta(\min\{\sqrt{pK}, 1/L\})$, there exists a constant $C > 0$, such that for any $T \geq 1$,*

$$\mathbb{E}\left[\|\nabla J\left(\theta_{\widehat{T}}\right)\|^2\right] \leq \frac{2\mathbb{E}\left[\Phi_0\right]}{\eta T} + \frac{C\sigma^2}{N}\left(\alpha + \frac{1}{K}\right)$$

with $\Phi_0 := J^* - J(\theta_0) + \frac{\eta}{p}\|v_0 - \nabla J(\theta_0)\|^2$ and $J^* := \max_{\theta\in\mathbb{R}^d} J(\theta)$.

COROLLARY1. *In the setting of Theorem 1, by choosing $p = 1/N$, the expected number of trajectories that need to be sampled per agent, to achieve $\mathbb{E}[\|\nabla J\left(\theta_{\widehat{T}}\right)\|^2] \leq \epsilon^2$, is*

$$O\left(\frac{\alpha^{1/2}}{K^{1/2}\epsilon^3} + \frac{1}{K\epsilon^3}\right).$$

Observe that in particular, if $\alpha = 0$, we need $O(K^{-1}\epsilon^{-3})$ trajectories in expectation, and for constant $\alpha > 0$, we need $O(K^{-1/2}\epsilon^{-3})$ trajectories in expectation. We hence recover the SOTA sample complexity of PAGE-PG [18] (which is proven under assumptions equivalent to ours) for $K = 1$, and asymptotically improve for larger K, despite the presence of Byzantines.

5 DECENTRALIZED BYZANTINE-TOLERANT FEDERATED POLICY GRADIENT

5.1 Method

In the decentralized setting, instead of having a centrally maintained $\theta_t \in \mathbb{R}^d$, the state at each iteration t is given by a tuple $\langle \theta_t^{(k)} \rangle_{k\in[K]}$ of each agent's local parameters with $\theta_t^{(k)} \in \mathbb{R}^d$. We are interested in the following solution concept.

DEFINITION 2 (K-AGENT α-TOLERANT ϵ-APPROXIMATE SOLUTION). *For $\epsilon > 0$, we call $\langle \theta^{(k)} \rangle_{k\in[K]}$ with $\theta^{(k)} \in \mathbb{R}^d$ a K-agent α-tolerant ϵ-stationary point if $\exists \mathcal{G} \subset [K]$ such that $|\mathcal{G}| \geq (1-\alpha)K$ and $\forall k \in \mathcal{G}$, we have $\|\nabla J(\theta^{(k)})\| \leq \epsilon$. We say a decentralized algorithm achieves a K-agent α-tolerant ϵ-approximate solution in T rounds if $\exists \mathcal{G}_T \subset [K]$ such that $|\mathcal{G}_T| \geq (1-\alpha)K$ and $\forall k \in \mathcal{G}_T$, we have $\mathbb{E}[\|\nabla J(\theta_T^{(k)})\|^2] \leq \epsilon^2$, where $\theta_T^{(k)}$ is the output of agent k after T rounds and the expectation is taken w.r.t. all randomness of the algorithm.*

Algorithm 2 DecByzPG at the k-th agent

1: **input:** $\theta_0 \in \mathbb{R}^d$, large batch size N, small batch size B, step size η, probability $p \in (0,1]$
2: initialize $\theta_0^{(k)} = \theta_0$
3: **for** $t = 0$ **to** $T-1$ **do**
4: $\quad c_t \leftarrow$ **Common-Sample** (Be (p))
5: $\quad M \leftarrow \begin{cases} N & \text{if } c_t = 1 \text{ or } t = 0 \\ B & \text{else} \end{cases}$
6: $\quad$ sample trajectories $\{\tau_{t,i}^{(k)}\}_{i=1}^{M}$ from $p(\cdot \mid \theta_t^{(k)})$
7: $\quad \widetilde{v}_t^{(k)} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} g(\tau_{t,i}^{(k)} \mid \theta_t^{(k)}) & \text{if } c_t = 1 \text{ or } t = 0 \\ \frac{1}{B}\sum_{i=1}^{B} g(\tau_{t,i}^{(k)} \mid \theta_t^{(k)}) & \\ \quad + \frac{1}{\eta}(\theta_t^{(k)} - \theta_{t-1}^{(k)}) & \\ \quad - \frac{1}{B}\sum_{i=1}^{B} g^{\omega_{\theta_t^{(k)}}}(\tau_{t,i}^{(k)} \mid \theta_{t-1}^{(k)}) & \text{else} \end{cases}$
8: $\quad v_t^{(k)} \leftarrow$ **Aggregate** $\left(\langle \widetilde{v}_t^{(k')} \rangle_{k'=1}^{K}\right)$
9: $\quad \widetilde{\theta}_{t+1}^{(k)} = \theta_t^{(k)} + \eta v_t^{(k)}$
10: $\quad \theta_{t+1}^{(k)} \leftarrow$ **Avg-Agree**$_\kappa$ $\left(\langle \widetilde{\theta}_{t+1}^{(k')} \rangle_{k'=1}^{K}\right)$
11: **output:** $\theta_{\widehat{T}}^{(k)}$ with $\widehat{T} \sim$ **Common-Sample** $(\mathcal{U}([T]))$

As afi rst step towards decentralizing ByzPG, suppose all agents simultaneously execute ByzPG, each with itself in the role of the server, and denote the k-th agent's resulting local parameters in iteration t by $\tilde{\theta}_t^{(k)}$. Since Byzantines may send inconsistent gradient estimates to different agents, already after thefi rst iteration, we may have $\widetilde{\theta}_1^{(k)} \neq \widetilde{\theta}_1^{(k')}$ for $k \neq k'$. Such disagreement on parameters across agents may be detrimental to convergence at each agent. As a remedy, we adopt the notion of averaging agreement that has been proposed by [8] in the context of Byzantine fault-tolerant collaborative learning.

Definition 3 (Averaging Agreement). *Let* ***Avg-Agree***$_\kappa$ *be a decentralized algorithm that as input receives* $\langle \theta^{(k)} \rangle_{k\in[K]}$ *where* $\theta^{(k)} \in \mathbb{R}^d$ *is known only to agent* k*. Under Assumption 1, let* $\mathcal{G}_t \subseteq \mathcal{H}_t$ *be such that* $|\mathcal{G}_t| \geq (1-\tilde{\alpha})K$*. Suppose after* κ *rounds of communication, where* $\kappa \in \mathbb{N}$ *is a parameter of the algorithm,* ***Avg-Agree***$_\kappa$ *terminates with output* $\langle \hat{\theta}^{(k)} \rangle_{k\in\mathcal{G}_t}$ *in the form of* $\hat{\theta}^{(k)} \in \mathbb{R}^d$ *being known to agent* k*. Then, we say* ***Avg-Agree***$_\kappa$ *achieves* C_{avg}*-averaging agreement for some* $C_{avg} > 0$*, if for any input it is guaranteed that*

$$\max_{i,l\in\mathcal{G}_t} \|\hat{\theta}^{(i)} - \hat{\theta}^{(l)}\| \leq \frac{\max_{i,l\in\mathcal{G}_t} \|\theta^{(i)} - \theta^{(l)}\|}{2^\kappa} \quad \text{and}$$

$$\|\bar{\hat{\theta}} - \bar{\theta}\| \leq C_{avg} \cdot \max_{i,l\in\mathcal{G}_t} \|\theta^{(i)} - \theta^{(l)}\|$$

where $\bar{\theta} = \frac{1}{|\mathcal{G}_t|}\sum_{k\in\mathcal{G}_t} \theta^{(k)}$ *and* $\bar{\hat{\theta}} = \frac{1}{|\mathcal{G}_t|}\sum_{k\in\mathcal{G}_t} \hat{\theta}^{(k)}$.

Known implementations satisfying the above definition are stated and discussed in Appendix A.3. Our algorithm DecByzPG, as described by Algorithm 2, employs an **Avg-Agree**$_\kappa$ subroutine at the end of each iteration to ensure averaging agreement on agents' local parameters.

We point out that while [8] makes use of averaging agreement in a similar context, their analysis does not yield sample complexity results. Our improved results rely upon the following insights:

(1) Careful analysis of bias and variance of the *realized* gradient estimates, which we define as $\hat{v}_t^{(k)} := \frac{1}{\eta}\left(\theta_{t+1}^{(k)} - \theta_t^{(k)}\right)$, reveal that variance-reduced methods combined with the notion of robust aggregation from Definition 1 show favorable interplay with averaging agreement. In particular, the low variance of intermediate estimates $\widetilde{v}_t^{(k)}$ and $v_t^{(k)}$ keep the bias introduced by **Avg-Agree**$_\kappa$ small.
(2) Controlling this bias introduced by **Avg-Agree**$_\kappa$ further requires a bound on the expected diameter of agents' parameters before agreement, i.e., the $\tilde{\theta}_{t+1}^{(k)}$'s. We leverage the fact that only the diameter of some large subset of parameters needs to be bounded, allowing us to apply strong concentration bounds instead of a weak union bound.

In place of **Sample** in Line 3 of Algorithm 1, DecByzPG requires a distributed Byzantine fault-tolerant sampling procedure. While such implementations have been studied in theory [30], in practice, we may simply use a pseudorandom generator with a seed derived from the common initialization θ_0.

5.2 Convergence Analysis and Sample Complexity

We next present sample complexity guarantees for DecByzPG, and provide a proof sketch outlining key ideas required for the analysis.

Theorem 2. *Let Assumptions 2, 3, 4, and 5 hold. Suppose* ***Aggregate*** *is an* (α, C_{ra})*-robust aggregator for constant* $C_{ra} > 0$ *and* α *as in Assumption 1. Let further* ***Avg-Agree***$_\kappa$ *achieve* C_{avg}*-averaging agreement for constant* $C_{avg} > 0$*. For* $A = \Theta\left(\frac{\alpha}{p^2} + \frac{1}{pK}\right)$*, choose* $\eta = \frac{1}{2}\min\left\{\frac{1}{\sqrt{A}}, \frac{1}{L}\right\}$*, and* $\kappa = \Theta\left(\log\frac{NK}{p^2}\right)$*. Then the following holds for the output of DecByzPG, i.e., Algorithm 2: There exists a constant* $C > 0$ *such that for any* $T \geq 1$*,* $\exists \mathcal{G}_{\widehat{T}} \subset [K]$ *with* $|\mathcal{G}_{\widehat{T}}| \geq (1-\tilde{\alpha})K$ *and* $\forall k \in \mathcal{G}_{\widehat{T}}$*,*

$$\mathbb{E}\left[\left\|\nabla J\left(\theta_{\widehat{T}}^{(k)}\right)\right\|^2\right] \leq \frac{4\mathbb{E}\left[\Phi_0\right]}{\eta T} + \frac{C\sigma^2}{N}\left(\alpha + \frac{1}{K}\right) + O\left(2^{-\kappa}\right)$$

where we define $\Phi_0 := J^* - J(\theta_0) + \frac{2\eta}{p}\left\|\frac{1}{K}\sum_{k\in[K]} \hat{v}_0^{(k)} - \nabla J(\theta_0)\right\|^2$ *with* $J^* := \max_{\theta\in\mathbb{R}^d} J(\theta)$.

Corollary 2. *In the setting of Theorem 2, by choosing* $p = 1/N$ *and* $\kappa = \Theta\left(\max\left\{\log(NK), \log\left(\epsilon^{-1}\right)\right\}\right)$*, the expected number of trajectories that need to be sampled per agent, to achieve a* K*-agent* $\tilde{\alpha}$*-tolerant* ϵ*-approximate solution as in Definition 2, is*

$$O\left(\frac{\alpha^{3/2}}{\epsilon^4} + \frac{\alpha^{1/2}}{K\epsilon^4} + \frac{\alpha^{1/2}}{K^{1/2}\epsilon^3} + \frac{1}{K\epsilon^3}\right).$$

In particular, if $\alpha = 0$, we need $O(K^{-1}\epsilon^{-3})$ trajectories in expectation which matches with our respective result from Corollary 1. The same sample complexity has been obtained in [7] for a momentum-based decentralized PG method that, however, lacks fault-tolerance. For constant $\alpha > 0$, we need $O(\epsilon^{-4})$ trajectories

in expectation which in our setting matches for example the complexity of single-agent vanilla PG [15]. If, e.g., a constant number of agents are Byzantine, i.e., $\alpha = \Theta(K^{-1})$, we get a complexity of $O(K^{-3/2}\epsilon^{-4} + K^{-1}\epsilon^{-3})$. Hence asymptotic speed-up w.r.t. the number of agents is possible despite the presence of Byzantine agents.

Remark. *Besides sample complexity, we prefer algorithms with low communication complexity. Due to* ***Avg-Agree***$_\kappa$*, each of the T iterations of* DecByzPG *involves* $\kappa = \Theta(\max\{\log(NK), \log(\epsilon^{-1})\})$ *rounds of all-to-all communication, each consisting of* $O(K^2)$ *messages containing a vector in* $\mathbb{R}^d$*. We point out that the logarithmic number of rounds is crucial for the practicality of our decentralized algorithm, as otherwise the cost of communication may outweigh the benefits of the lower sample complexity gained from collaboration.*

Due to space constraints, the full proofs of Theorem 2 and Corollary 2, as well as all required lemmas are deferred to Appendix C. Here, we want to focus on one key argument of the proof responsible for controlling the diameter of agents' local parameters. Before stating and proving the two respective lemmas, we introduce additional notation: Recall that $\mathcal{H}_t \subset [K]$ is the set of *honest*, i.e., non-Byzantine agents as in Assumption 1 with $|\mathcal{H}_t| \geq (1-\alpha)K$. In addition, with $\bar{\alpha} = \alpha + \bar{\epsilon} < \alpha_{\max} = 1/4$, denote the diameter of a tuple of vectors by $\Delta_2(\cdot)^2$, e.g., for some $S \subseteq [K]$, let

$$\Delta_2\left(\langle \theta_t^{(i)} \rangle_{i\in S}\right) := \max_{i,j\in S} \|\theta_t^{(i)} - \theta_t^{(j)}\|$$

and consider the set

$$\mathcal{G}_t := \operatorname*{arg\,min}_{S\subset\mathcal{H}_t, |S|\geq(1-\bar{\alpha})K} \Delta_2\left(\langle \tilde{\theta}_t^{(i)} \rangle_{i\in S}\right) \subset \mathcal{H}_t$$

which we will call the set of *good* agents. As we will show below, the diameter of good agents' parameters exhibits good concentration in the sense that we obtain stronger bounds as would hold for the expected diameter of all honest agents' parameters. The diameter of good agents' parameters after agreement will frequently occur as an error term which we denote by

$$\mathcal{E}_t^\Delta := \Delta_2\left(\langle \theta_t^{(i)} \rangle_{i\in\mathcal{G}_t}\right)^2. \tag{2}$$

Finally, we abbreviate

$$\widetilde{\mathcal{T}}_{1,t} := \frac{1}{|\mathcal{G}_t|(|\mathcal{G}_t|-1)} \sum_{i,l\in\mathcal{G}_t} \mathbb{E}\left[\|\tilde{v}_t^{(i)} - \tilde{v}_t^{(l)}\|^2 \mid c_t = 1\right],$$

$$\widetilde{\mathcal{T}}_{0,t} := \frac{1}{|\mathcal{G}_t|(|\mathcal{G}_t|-1)} \sum_{i,l\in\mathcal{G}_t} \mathbb{E}\left[\|\tilde{v}_t^{(i)} - \tilde{v}_t^{(l)}\|^2 \mid c_t = 0\right].$$

The following lemma bounds the diameter of good agents' parameters after aggregation and before agreement in iteration t, distinguishing between the two cases given by the probabilistic switch.

Lemma1. *For any* $\bar{\epsilon} > 0$*, it holds that*

$$\mathbb{E}\left[\Delta_2\left(\langle \tilde{\theta}_{t+1}^{(i)} \rangle_{i\in\mathcal{G}_t}\right)^2 \mid c_t = 1\right] \leq 2\mathbb{E}[\mathcal{E}_t^\Delta] + \frac{10\eta^2 C_{ra}\alpha\widetilde{\mathcal{T}}_{1,t}}{\bar{\epsilon}},$$

$$\mathbb{E}\left[\Delta_2\left(\langle \tilde{\theta}_{t+1}^{(i)} \rangle_{i\in\mathcal{G}_t}\right)^2 \mid c_t = 0\right] \leq 2\mathbb{E}[\mathcal{E}_t^\Delta] + \frac{10\eta^2 C_{ra}\alpha\widetilde{\mathcal{T}}_{0,t}}{\bar{\epsilon}}.$$

Proof. First, we define

$$S_t := \operatorname*{arg\,min}_{S\subset\mathcal{G}_t, |S|\geq(1-(\alpha+\bar{\epsilon}))K} \Delta_2\left(\langle v_t^{(i)} \rangle_{i\in S_t}\right) \subset \mathcal{H}_t \subset [K]$$

and aim to bound $\mathbb{E}\left[\Delta_2(\langle v_t^{(i)} \rangle_{i\in S_t})^2\right]$. Observe that we have

$$\Delta_2\left(\langle v_t^{(i)} \rangle_{i\in S_t}\right)^2 \leq \max_{i\in S_t} \|\bar{\tilde{v}}_t - v_t^{(i)}\|^2.$$

For $i \in \mathcal{H}_t$, let $\overline{\mathcal{T}}_i$ be such that $\mathbb{E}[\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2] \leq \overline{\mathcal{T}}_i$ (we will plug in the right bound $\overline{\mathcal{T}}_i$ later) where $\bar{\tilde{v}}_t := \frac{1}{|\mathcal{G}_t|}\sum_{i\in\mathcal{G}_t} \tilde{v}_t^{(i)}$, and let X_i be indicator random variables for the events

$$\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2 \geq \frac{2}{\bar{\epsilon}} \cdot \overline{\mathcal{T}}_i.$$

Let $X = \sum_{i\in\mathcal{H}_t} X_i$. Our goal is to upper bound $\mathbb{E}X_i = \Pr[X_i = 1]$ in order to use Chernoff concentration bounds on X.

By Lemma 8 (see Appendix A.4), we get

$$\begin{aligned}
\Pr[X_i = 1] &= \Pr\left[\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2 \geq \frac{2}{\bar{\epsilon}} \cdot \overline{\mathcal{T}}_i\right] \\
&\leq \Pr\left[\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2 \geq \frac{2}{\bar{\epsilon}} \cdot \mathbb{E}\left[\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2\right]\right] \\
&= \Pr\left[\|\bar{\tilde{v}}_t - v_t^{(i)}\| \geq \sqrt{\frac{2}{\bar{\epsilon}}} \cdot \sqrt{\mathbb{E}\left[\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2\right]}\right] \\
&\leq \frac{\mathbb{E}\left[\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2\right]}{\frac{2}{\bar{\epsilon}}\mathbb{E}\left[\|\bar{\tilde{v}}_t - v_t^{(i)}\|^2\right]} \\
&= \frac{\bar{\epsilon}}{2}.
\end{aligned}$$

Let E be the "bad case", i.e., the event that we have $X \geq (\bar{\epsilon}+\alpha)K$. By Lemma 9 (see Appendix A.4), with $\delta = 1 + \frac{2\alpha}{\bar{\epsilon}}$ and $\hat{p}$ as bounded above, we get

$$\begin{aligned}
\Pr[E] = \Pr[X \geq (\bar{\epsilon}+\alpha)K] &= \Pr\left[X \geq (1+\delta)\frac{\bar{\epsilon}K}{2}\right] \\
&\leq \exp\left(-\frac{\delta^2\bar{\epsilon}K}{4+2\delta}\right) \\
&= \exp\left(-\frac{K(2\alpha+\bar{\epsilon})^2}{4\alpha+6\bar{\epsilon}}\right) \\
&\leq \exp\left(-\frac{\bar{\epsilon}K}{6}\right).
\end{aligned}$$

With $\overline{\mathcal{T}} := \max_{i\in\mathcal{H}_t} \overline{\mathcal{T}}_i$, by the law of total expectation, we then have

$$\begin{aligned}
\mathbb{E}\left[\Delta_2\left(\langle v_t^{(i)} \rangle_{i\in S_t}\right)^2\right] &\leq \mathbb{E}\left[\Delta_2\left(\langle v_t^{(i)} \rangle_{i\in S_t}\right)^2 \mid \bar{E}\right] \cdot \underbrace{\Pr[\bar{E}]}_{\leq 1} \\
&\quad + \mathbb{E}\left[\Delta_2\left(\langle v_t^{(i)} \rangle_{i\in S_t}\right)^2 \mid E\right] \cdot \Pr[E] \\
&\leq \frac{2}{\bar{\epsilon}}\overline{\mathcal{T}} + K\overline{\mathcal{T}} \cdot \exp\left(-\frac{\bar{\epsilon}K}{6}\right) \\
&\leq \frac{5}{\bar{\epsilon}}\overline{\mathcal{T}}
\end{aligned}$$

where in thefi rst step, for the expectation conditioned on E we union-bound the max by introducing a factor K, and in the second

step we use the fact that the function $f(x) = xe^{-\beta x}$ has a global maximum with value $\frac{1}{\beta e}$.

Remains to use this bound on $\mathbb{E}[\Delta_2(\langle v_t^{(i)}\rangle_{i\in S_t})^2]$ in order to obtain the desired bound on $\mathbb{E}[\Delta_2(\langle \tilde{\theta}_{t+1}^{(i)}\rangle_{i\in \mathcal{G}_t})^2]$ which follows straightforwardly since

$$\begin{aligned}
&\mathbb{E}\left[\Delta_2\left(\langle \tilde{\theta}_{t+1}^{(i)}\rangle_{i\in\mathcal{G}_t}\right)^2\right] \\
&= \mathbb{E}\left[\max_{i,l\in\mathcal{G}_t} \|\tilde{\theta}_{t+1}^{(i)} - \tilde{\theta}_{t+1}^{(l)}\|^2\right] \\
&\leq 2\mathbb{E}\left[\max_{i,l\in\mathcal{G}_t} \|\theta_t^{(i)} - \theta_t^{(l)}\|^2\right] + 2\eta^2\mathbb{E}\left[\max_{i,l\in\mathcal{G}_t} \|v_t^{(i)} - v_t^{(l)}\|^2\right] \\
&\leq 2\mathbb{E}[\mathcal{E}_t^{\Delta}] + 2\eta^2\mathbb{E}\left[\Delta_2\left(\langle v_t^{(i)}\rangle_{i\in S_t}\right)^2\right] \\
&\leq 2\mathbb{E}[\mathcal{E}_t^{\Delta}] + \frac{10\eta^2}{\bar{\epsilon}}\overline{\mathcal{T}}.
\end{aligned}$$

What can we plug in for $\overline{\mathcal{T}}$? Observe that $v_t^{(i)}$ is the result of aggregation of inputs with average $\bar{\tilde{v}}_t := \frac{1}{|\mathcal{G}_t|}\sum_{i\in\mathcal{G}_t}\tilde{v}_t^{(i)}$. Therefore, by Definition 1, for any $i \in \mathcal{G}_t$, we have

$$\mathbb{E}\left[\|v_t^{(i)} - \bar{\tilde{v}}_t\|^2\right] \leq \frac{C_{ra}\alpha}{|\mathcal{G}_t|(|\mathcal{G}_t|-1)}\sum_{i,l\in\mathcal{G}_t}\mathbb{E}\left[\|\tilde{v}_t^{(i)} - \tilde{v}_t^{(l)}\|^2\right].$$

Thus, we can distinguish between conditioning our expectation on $c_t = 0$ and $c_t = 1$ and using the respective bounds $C_{ra}\alpha\widetilde{\mathcal{T}}_{0,t}$ and $C_{ra}\alpha\widetilde{\mathcal{T}}_{1,t}$, the result follows. □

With this bound on the diameter of intermediate local parameters $\tilde{\theta}_t^{(i)}$ at hand, we can now derive the desired bound on parameters after agreement that only depends on the averaging agreement parameter κ.

Lemma2. *For all iterations $t \leq T$, there exists $\overline{\mathcal{E}}^{\Delta,\kappa}$ such that $\mathbb{E}[\mathcal{E}_t^{\Delta}] \leq \overline{\mathcal{E}}^{\Delta,\kappa}$ and*

$$\overline{\mathcal{E}}^{\Delta,\kappa} \leq \mathcal{O}\left(2^{-\kappa}\right).$$

Proof. We proceed by induction on t. For $t = 0$, $\mathcal{E}_t^{\Delta} = 0$ due to the common initialization $\theta^{(k)} = \theta_0$ for all $k \in [K]$. Suppose for some $t < T-1$, $\mathbb{E}[\mathcal{E}_t^{\Delta}] \leq \overline{\mathcal{E}}^{\Delta,\kappa} \leq \mathcal{O}(2^{-\kappa})$. In iteration t, applying Lemma 1 and the bounds on $\widetilde{\mathcal{T}}_{0,t}$ and $\widetilde{\mathcal{T}}_{0,t}$ from Lemma 15 (see Appendix C), one can observe that in both cases $c_{t-1} = 1$ and $c_{t-1} = 0$, the expected diameter of $\tilde{\theta}_t^{(i)}$'s for good agents $i \in \mathcal{G}_t$ is (loosely) bounded by $\mathcal{O}(1)$. Hence by the definition of averaging agreement, see Definition 3, we have $\mathbb{E}\left[\mathcal{E}_t^{\Delta}\right] \leq \overline{\mathcal{E}}^{\Delta,\kappa}$ with

$$\overline{\mathcal{E}}^{\Delta,\kappa} \leq \frac{\mathbb{E}\left[\Delta_2\left(\langle \tilde{\theta}_t^{(i)}\rangle_{i\in\mathcal{G}_t}\right)\right]}{2^{\kappa}} \leq \mathcal{O}\left(2^{-\kappa}\right).$$

□

6 EXPERIMENTS

In order to corroborate our theoreticalfi ndings, we empirically study the performance of the proposed methods w.r.t. the properties suggested by Corollary 2, i.e., (a) speed-up when increasing the number of agents K, and (b) resilience against various Byzantine attacks. We focus on our main contribution regarding the more challenging decentralized setting here (i.e. DecByzPG), and defer experiments for ByzPG to Appendix E.

Environments and Setup. We consider two common RL benchmarks, CartPole [31] and LunarLander. For all experiments, we report average returns of honest agents (y-axis) in terms of the trajectories that have been sampled per agent (x-axis). To visualize potential variance in our experiments, all plots show the respective mean and standard deviation across 10 independent runs. Further details, including hyperparameters, can be found in Appendix D.

6.1 DecByzPG without Byzantine Agents

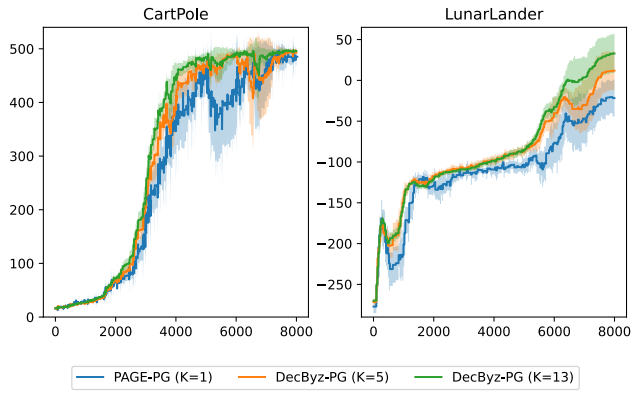

Figure 1: Performance of DecByzPG for different federation sizes when all agents behave honestly (i.e. $\alpha = 0$).

In Figure 1, we consider DecByzPG in the case $\alpha = 0$, with $K = 1$ (which is equivalent to PAGE-PG [18]), $K = 5$, and $K = 13$. Speed-up with increasing number of agents is observable in both environments, as suggested by Corollary 2. Such faster convergence provides empirical evidence motivating agents to join a decentralized federation.

6.2 DecByzPG under Attack

Choice of attacks. In previous work [5], Byzantine attacks are constructed by making random modifications to an agent's interaction with its environment, by e.g. choosing an action u.a.r. instead of following the current policy (here denoted **RandomAction**), adding noise to the reward, or randomlyfl ipping the reward's sign. Wefi nd that in our setting, for simple environments such as CartPole, robustness to such attacks is often already given for naively collaborating agents. This behavior is exemplified by our experiments under the **RandomAction** attack. Thus, even though DecByzPG is also resilient to such attacks, a stronger adversary is needed to demonstrate DecByzPG's advantage over naive methods. **LargeNoise** lets Byzantine agents directly send noise instead of gradients obtained from noisy interactions. Even though introducing noise may generally also have beneficial effects on convergence speed (e.g. due to improved exploration), by choosing the variance large enough, such benefits are outweighed. The third attack, **AvgZero** leverages the power of Byzantine knowledge and collaboration. Gradients sent by Byzantines are chosen such that when

averaged with gradients sent by honest agents, the result will be close to zero.

In Figure 2 and 3, we compare DecByzPG under above attacks to (a) PAGE-PG [18], the SOTA single-agent PG method that DecByzPG reduces to when $K = 1$, and (b) Dec-PAGE-PG, a naive decentralized (but not fault-tolerant) version of PAGE-PG where aggregation of gradients is done by averaging, and no agreement mechanism is used. Note that for experiments involving Byzantine agents, we choose their quantity to be the largest for which Assumption 1, and hence the guarantees of Theorem 2, still hold (i.e. 3 out of 13 agents are Byzantine).

For both environments and all attacks, we can observe that DecByzPG performs nearly on par with the unattacked Dec-PAGE-PG. This empirically supports the Byzantine fault-tolerance of DecByzPG. Furthermore, for CartPole, as expected, **LargeNoise** and **AvgZero** are highly effective against the non-fault-tolerant method, while as previously remarked, **RandomAction** barely shows any effect. For the more difficult task of LunarLander, already **RandomAction** breaks Dec-PAGE-PG. Lastly, we point out that in all cases DecByzPG with $K = 13$ and $\alpha > 0$ outperforms PAGE-PG with $K = 1$ (and $\alpha = 0$), meaning that in our experiments, despite the presence of Byzantines, joining the federation is empirical beneficial for faster convergence.

7 CONCLUSION

We described and analyzed a federated decentralized Byzantine fault-tolerant PG algorithm. As a warm-up, we combined variance-reduced PG methods with results from Byzantine-tolerant non-convex optimization to obtain a new centralized algorithm under standard assumptions. We then use ideas from Byzantine robust aggregation and agreement to generalize our approach to the significantly more challenging decentralized setting. As a result, we obtained thefi rst sample complexity guarantees for Byzantine fault-tolerant decentralized federated non-convex optimization. We thus believe that our technical contributions are more generally applicable and may therefore open up directions for future research. Moreover, the provided empirical results for standard RL benchmark tasks support our theory and promise practical relevance of our method.

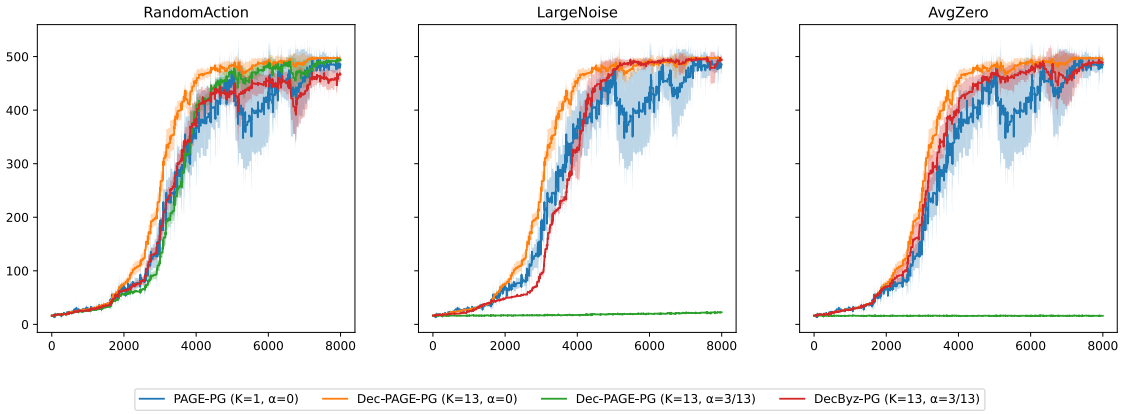

Figure 2: Performance & resilience of DecByzPG for CartPole w.r.t. our three attack types.

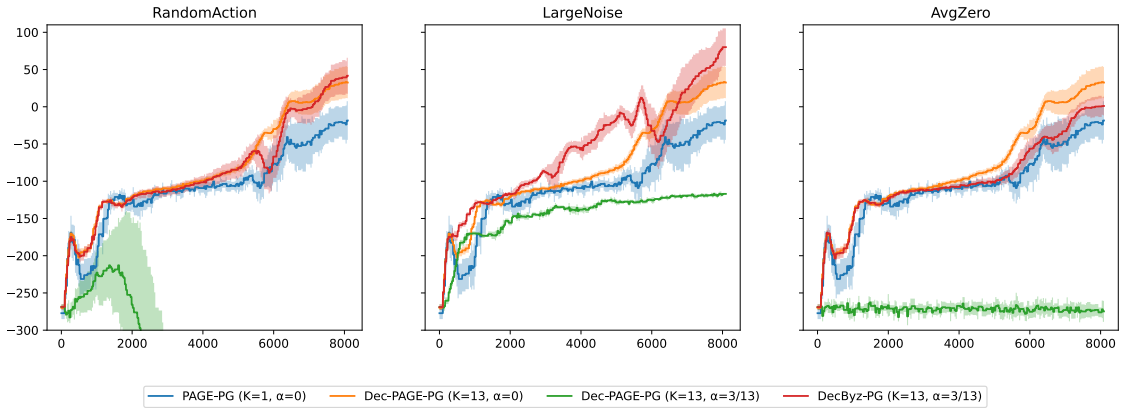

Figure 3: Performance & resilience of DecByzPG for LunarLander w.r.t. our three attack types.

REFERENCES

[1] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[2] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[3] Jiaju Qi, Qihao Zhou, Lei Lei, and Kan Zheng. Federated reinforcement learning: Techniques, applications, and open challenges. *arXiv preprint arXiv:2108.11887*, 2021.

[4] Hankz Hankui Zhuo, Wenfeng Feng, Yufeng Lin, Qian Xu, and Qiang Yang. Federated deep reinforcement learning. *arXiv preprint arXiv:1901.08277*, 2019.

[5] Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Wei Jing, Cheston Tan, and Bryan Kian Hsiang Low. Fault-tolerant federated reinforcement learning with theoretical guarantee. In *Advances in Neural Information Processing Systems*, volume 34, pages 1007–1021, 2021.

[6] Flint Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Cheston Tan, Bryan Kian Hsiang Low, and Roger Wattenhofer. Fedhql: Federated heterogeneous q-learning. arXiv:2301.11135, 2023.

[7] Zhanhong Jiang, Xian Yeow Lee, Sin Yong Tan, Kai Liang Tan, Aditya Balu, Young M Lee, Chinmay Hegde, and Soumik Sarkar. Mdpgt: momentum-based decentralized policy gradient tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9377–9385, 2022.

[8] El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). *Advances in Neural Information Processing Systems*, 34:25044–25057, 2021.

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

[10] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.

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

[13] Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[14] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323, 2016.

[15] Matteo Papini. Safe policy optimization. 2021.

[16] Pan Xu, Felicia Gao, and Quanquan Gu. An improved convergence analysis of stochastic variance-reduced policy gradient. In *Uncertainty in Artificial Intelligence*, pages 541–551. PMLR, 2020.

[17] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International conference on machine learning*, pages 6286–6295, 2021.

[18] Matilde Gargiani, Andrea Zanelli, Andrea Martinelli, Tyler Summers, and John Lygeros. Page-pg: A simple and loopless variance-reduced policy gradient method with probabilistic gradient estimation. In *International Conference on Machine Learning*, pages 7223–7240, 2022.

[19] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3), 1982.

[20] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128, 2017.

[21] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. *Advances in Neural Information Processing Systems*, 31, 2018.

[22] Zeyuan Allen-Zhu, Faeze Ebrahimian, Jerry Li, and Dan Alistarh. Byzantine-resilient non-convex stochastic gradient descent. arXiv:2012.14368, 2020.

[23] Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*, pages 6246–6283, 2022.

[24] Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel. Variance reduction is an antidote to byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top. In *The Eleventh International Conference on Learning Representations*, 2023.

[25] Hammurabi Mendes and Maurice Herlihy. Multidimensional approximate agreement in byzantine asynchronous systems. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 391–400, 2013.

[26] Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K Garg. Multidimensional agreement in byzantine systems. *Distributed Computing*, 28(6): 423–441, 2015.

[27] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, pages 4026–4035, 2018.

[28] Huizhuo Yuan, Xiangru Lian, Ji Liu, and Yuren Zhou. Stochastic recursive momentum for policy gradient methods. *arXiv preprint arXiv:2003.04302*, 2020.

[29] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, pages 5311–5319, 2021.

[30] C Cachin and V Shoup. Random oracles in constantinople: Practical asynchronous byzantine agreement using. In *Proceedings of the 19th ACM Symposium on Principles of Distributed Computing, no*, pages 1–26, 2000.

[31] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, pages 834–846, 1983.

[32] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

[33] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*, 2022.

[34] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[35] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[36] Endre Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386, 1937.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.