2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Distributed Statistical Min-Max Learning in the Presence of Byzantine Agents

Arman Adibi*, Aritra Mitra*, George J. Pappas and Hamed Hassani

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993078

***Abstract*— Recent years have witnessed a growing interest in the topic of min-max optimization, owing to its relevance in the context of generative adversarial networks (GANs), robust control and optimization, and reinforcement learning. Motivated by this line of work, we consider a multi-agent min-max learning problem, and focus on the emerging challenge of contending with worst-case Byzantine adversarial agents in such a setup. By drawing on recent results from robust statistics, we design a robust distributed variant of the extra-gradient algorithm - a popular algorithmic approach for min-max optimization. Our main contribution is to provide a crisp analysis of the proposed robust extra-gradient algorithm for smooth convex-concave and smooth strongly convex-strongly concave functions. Specifically, we establish statistical rates of convergence to approximate saddle points. Our rates are near-optimal, and reveal both the effect of adversarial corruption and the benefit of collaboration among the non-faulty agents. Notably, this is the first paper to provide formal theoretical guarantees for large-scale distributed min-max learning in the presence of adversarial agents.**

I. Introduction

We consider a min-max learning problem of the form

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} f(x,y) \triangleq \mathbb{E}_{\xi\sim\mathcal{D}}[F(x,y;\xi)]. \tag{1}$$

Here, $\mathcal{X}$ and $\mathcal{Y}$ are convex, compact sets in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively; $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are model parameters; ξ is a random variable representing a data point sampled from the distribution $\mathcal{D}$; and $f(x,y)$ is the population function corresponding to the stochastic function $F(x,y;\xi)$. Throughout this paper, we assume that $f(x,y)$ is continuously differentiable in x and y, and is *convex-concave* over $\mathcal{X}\times\mathcal{Y}$. Specifically, $f(\cdot,y) : \mathcal{X} \to \mathbb{R}$ is convex for every $y \in \mathcal{Y}$, and $f(x,\cdot) : \mathcal{Y} \to \mathbb{R}$ is concave for every $x \in \mathcal{X}$. Our goal is to find a saddle point (x^*, y^*) of $f(x,y)$ over the set $\mathcal{X}\times\mathcal{Y}$, where a saddle point is defined as a vector pair $(x^*, y^*) \in \mathcal{X}\times\mathcal{Y}$ that satisfies

$$f(x^*,y) \le f(x^*,y^*) \le f(x,y^*), \forall x \in \mathcal{X}, y \in \mathcal{Y}. \tag{2}$$

The min-max optimization problem described above features in a variety of applications: from classical developments in game theory [1] and online learning [2], to robust optimization [3] and reinforcement learning [4]. More recently, in the context of machine learning, min-max problems have found important applications in training generative adversarial networks (GANs) [5], and in robustifying deep neural networks against adversarial attacks [6]. Motivated by this recent line of work, we consider a min-max learning problem of the form in Eq. (1), where the data samples required for finding a saddle-point are distributed across multiple devices (agents). Specifically, we focus on a large-scale distributed setup comprising of M agents, each of which can access i.i.d. data samples from the distribution $\mathcal{D}$. The agents collaborate under the orchestration of a central server to compute an approximate saddle point of statistical accuracy higher relative to the setting when they act alone. The intuition here is simple: since all agents receive data samples from the *same* distribution, exchanging information via the server can help reduce the randomness (variance) associated with these samples.[1] An example of the above setup that aligns with the modern federated learning paradigm is one where multiple devices (e.g., cell phones or tablets) collaborate via a server to train a robust statistical model; see, for instance, [7].

To reap the benefits of collaboration in modern distributed computing systems, one needs to contend with the critical challenge of *security*. In particular, this challenge arises from the fact that the individual agents in such systems are easily susceptible to adversarial attacks. In fact, unless appropriately accounted for, even a single malicious agent can severely degrade the overall performance of the system by sending corrupted messages to the central server.

Objective. Thus, given the emerging need for security in large-scale computing, *our objective in this paper is to design an algorithm that achieves near-optimal statistical performance in the context of distributed min-max learning, while being robust to worst-case attacks*. To that end, we consider a setting where a fraction of the agents is Byzantine [8]. Each Byzantine agent is assumed to have complete knowledge of the system and learning algorithms; moreover, leveraging such knowledge, the Byzantine agents can send arbitrary messages to the server and collude with each other.

Challenges. Even in the absence of noise or attacks, recent work [9] has shown that algorithms such as gradient descent ascent (`GDA`) can diverge for simple convex-concave functions. We have to contend with both noise (due to our statistical setup) *and* worst-case attacks - this makes the analysis for our setting non-trivial. In particular, the adversarial agents can introduce complex probabilistic dependencies across iterations that need to be carefully accounted for; we do so in this work by making the following contributions.

*Arman Adibi and Aritra Mitra contributed equally.

The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania. Email: {aadibi, amitra20, hassani, pappasg}@seas.upenn.edu. This work was supported by NSF CPS Grant 1837253, ARL CRA DCIST, NSF CAREER award CIF 1943064, and the Air Force Office of Scientific Research Young Investigator Program (AFOSR-YIP) under award FA9550-20-1-0111.

[1] This intuition will be made precise in Section IV.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:31 UTC from IEEE Xplore. Restrictions apply.

Contributions. Our contributions are summarized below.

• *Problem.* Given the importance and relevance of security, several recent works have studied distributed optimization/learning in the face of adversarial agents. However, we are unaware of any analogous paper for adverarially-robust distributed *min-max* learning. Our work closes this gap.

• *Algorithm.* In Section III, we develop an algorithm for finding an approximate saddle point to the min-max learning problem in Eq. (1), subject to the presence of Byzantine agents. Our proposed algorithm - called Robust Distributed Extra-Gradient (RDEG) - brings together two separate algorithmic ideas: (i) the classical extra-gradient algorithm due to Korpelevich [10] that has gained a lot of popularity due to its empirical performance in training GANs, and (ii) the recently proposed univariate trimmed mean estimator due to Lugosi and Mendelson [11].

• *Theoretical Results.* Our main contribution is to provide a rigorous theoretical analysis of the performance of RDEG for smooth convex-concave (Theorem 2), and smooth strongly convex-strongly concave (Theorem 3) settings. In each case, we establish that as long as the fraction of corrupted agents is "small", RDEG guarantees convergence to approximate saddle points at *near-optimal* statistical rates with high probability. The rates that we derive precisely highlight the benefit of collaboration in effectively reducing the variance of the noise model. At the same time, they indicate the (unavoidable) additive bias introduced by adversarial corruption. Notably, our results in the context of min-max learning complement those of a similar flavor in [12] for stochastic optimization under attacks. However, our analysis differs significantly from that in [12]: unlike the covering argument employed in [12], our proofs rely on a simpler, and more direct probabilistic analysis. An immediate benefit of such an analysis is that one can build on it for the more challenging nonconvex-nonconcave setting as future work.

Related Work. In what follows, we discuss connections to relevant strands of literature.

• *Min-Max Optimization.* Convergence guarantees of first-order algorithms for saddle point problems over compact sets were studied in [13] and [14]. More recently, there has been a surge of interest in analyzing the performance of such algorithms from different perspectives: a dynamical systems approach in [15], [16], and a proximal point perspective in [17]. We refer to [18] for a detailed survey on this topic.

• *Robust Distributed Optimization and Learning.* Robustness to adversarial agents in distributed optimization has been extensively studied in [19]–[21]. However, these works consider deterministic settings, and do not provide statistical error rates like we do. In the context of statistical learning over a server-client computing architecture, several works have proposed and analyzed robust algorithms [12], [22]–[25]. Notably, none of the above works consider the min-max learning problem studied in this paper.

• *Robust Statistics.* Robust mean estimation in the presence of outliers is a classical topic in statistics pioneered by Huber [26], with follow-up work in [27]. In our work, we exploit some recent results on this topic from [11].

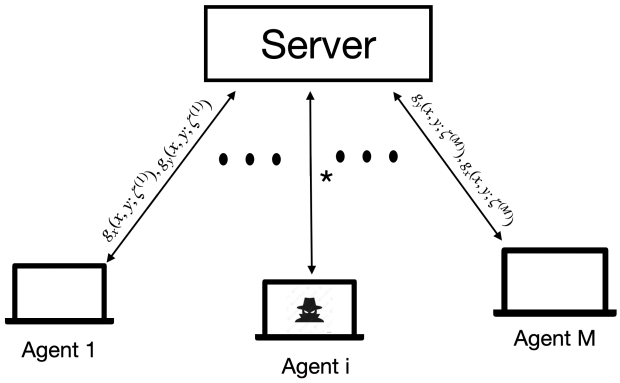


Fig. 1. A group of M agents collaborate to find a saddle point for the min-max learning problem in Eq. (1). A fraction α of the agents is adversarial and upload arbitrarily corrupted messages (denoted by $*$) to the server. All the remaining good agents upload noisy partial gradients of $f(x, y)$.

II. Problem Formulation

In this section, we formally set up the problem of interest by first introducing some notation. Our setting comprises of M agents, αM of whom are Byzantine; see Fig. 1. We denote the adversarial agents by $\mathcal{B} \in [M]$.[2] For any $\bar{x} \in \mathcal{X}$ and $\bar{y} \in \mathcal{Y}$, let $\nabla_x f(\bar{x}, \bar{y})$ and $\nabla_y f(\bar{x}, \bar{y})$ denote the gradient of $f(x, y)$ with respect to x and y, respectively, at $(\bar{x}, \bar{y})$. Upon drawing a sample $\xi \sim \mathcal{D}$ at a point $(\bar{x}, \bar{y})$, each normal agent receives noisy estimates of $\nabla_x f(\bar{x}, \bar{y})$ and $\nabla_y f(\bar{x}, \bar{y})$ denoted by $g_x(\bar{x}, \bar{y}; \xi)$ and $g_y(\bar{x}, \bar{y}; \xi)$, respectively. For each normal agent in $[M] \setminus \mathcal{B}$, these noisy estimates satisfy the following for all $\bar{x} \in \mathcal{X}$ and $\bar{y} \in \mathcal{Y}$:

$$\begin{aligned} \mathbb{E}_{\xi\sim\mathcal{D}}[g_x(\bar{x}, \bar{y}; \xi)] &= \nabla_x f(\bar{x}, \bar{y}) \\ \mathbb{E}_{\xi\sim\mathcal{D}}[g_y(\bar{x}, \bar{y}; \xi)] &= \nabla_y f(\bar{x}, \bar{y}). \end{aligned} \tag{3}$$

Furthermore, $\forall j \in [n]$ and $\forall k \in [m]$, we have

$$\begin{aligned} \mathbb{E}_{\xi\sim\mathcal{D}}\left[\left\|[g_x(\bar{x}, \bar{y}; \xi)]_j - [\nabla_x f(\bar{x}, \bar{y})]_j\right\|^2\right] &\le \sigma_x^2(j) \\ \mathbb{E}_{\xi\sim\mathcal{D}}\left[\left\|[g_y(\bar{x}, \bar{y}; \xi)]_k - [\nabla_y f(\bar{x}, \bar{y})]_k\right\|^2\right] &\le \sigma_y^2(k), \end{aligned} \tag{4}$$

where we used $[a]_j$ to represent the j-th component of a vector a.[3] In words, each normal agent receives unbiased estimates of the gradients of $f(x, y)$ (w.r.t. x and y) with component-wise bounded variance - essentially, a standard stochastic oracle model. With a slight abuse of notation, we will continue to use $\{g_x(x, y; \xi), g_y(x, y; \xi)\}$ to denote the gradients transmitted by an adversarial agent as well; these could, however, be arbitrary corrupted vectors. Our problem of interest can now be stated as follows.

Problem 1. *Given access to the stochastic oracle model described by equations (3) and (4), design a distributed algorithm that finds a saddle point (in the sense of Eq. (2)) for the function $f(x, y)$ in Eq. (1), despite the presence of the Byzantine adversarial set $\mathcal{B}$.*

In the next section, we will develop our proposed algorithm to address Problem 1.

[2]Given a positive integer N, we use $[N]$ to represent the set $\{1, \ldots, N\}$.
[3]We use $\|\cdot\|$ to represent the Euclidean norm.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:31 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Robust Distributed Extra-Gradient (RDEG)

Require: Initial vectors $x_1 \in \mathcal{X}$, $y_1 \in \mathcal{Y}$; algorithm parameters: step-size $\eta > 0$ and trimming parameter ϵ.

1: **for** $t = 1, \ldots, T$ **do**

2: Server sends (x_t, y_t) to each agent.

3: Each *normal* agent i draws an i.i.d. sample $\xi_{1,t}^{(i)} \sim \mathcal{D}$, and transmits $g_x(x_t, y_t; \xi_{1,t}^{(i)})$, $g_y(x_t, y_t; \xi_{1,t}^{(i)})$ to server.[4]

4: Server computes robust gradients:

$$\begin{aligned} \tilde{g}_x(x_t, y_t) &\leftarrow \texttt{Trim}_\epsilon\{g_x(x_t, y_t; \xi_{1,t}^{(i)}) : i \in [M]\} \\ \tilde{g}_y(x_t, y_t) &\leftarrow \texttt{Trim}_\epsilon\{g_y(x_t, y_t; \xi_{1,t}^{(i)}) : i \in [M]\}. \end{aligned} \tag{5}$$

5: Server computes mid-points $(\hat{x}_t, \hat{y}_t)$ as follows, and transmits them to each agent.

$$\begin{aligned} \hat{x}_t &\leftarrow \Pi_{\mathcal{X}}\left(x_t - \eta \tilde{g}_x(x_t, y_t)\right) \\ \hat{y}_t &\leftarrow \Pi_{\mathcal{Y}}\left(y_t + \eta \tilde{g}_y(x_t, y_t)\right). \end{aligned} \tag{6}$$

6: Each *normal* agent i draws an i.i.d. sample $\xi_{2,t}^{(i)} \sim \mathcal{D}$, and transmits $g_x(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)})$, $g_y(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)})$ to server.

7: Server computes robust gradients:

$$\begin{aligned} \tilde{g}_x(\hat{x}_t, \hat{y}_t) &\leftarrow \texttt{Trim}_\epsilon\{g_x(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)}) : i \in [M]\} \\ \tilde{g}_y(\hat{x}_t, \hat{y}_t) &\leftarrow \texttt{Trim}_\epsilon\{g_y(\hat{x}_t, \hat{y}_t; \xi_{2,t}^{(i)}) : i \in [M]\}. \end{aligned} \tag{7}$$

8: Server computes new updates x_{t+1} and y_{t+1}:

$$\begin{aligned} x_{t+1} &\leftarrow \Pi_{\mathcal{X}}\left(x_t - \eta \tilde{g}_x(\hat{x}_t, \hat{y}_t)\right) \\ y_{t+1} &\leftarrow \Pi_{\mathcal{Y}}\left(y_t + \eta \tilde{g}_x(\hat{x}_t, \hat{y}_t)\right). \end{aligned} \tag{8}$$

9: **end for**

III. Robust Distributed Extra-Gradient

In this section, we develop the Robust Distributed Extra-Gradient (RDEG) algorithm outlined in Algorithm 1. Our algorithm evolves in discrete-time iterations $t \in [T]$, where T is the total number of iterations. There are two main steps in RDEG. In the first step, the server computes robust gradient estimates $\{\tilde{g}_x(x_t, y_t), \tilde{g}_y(x_t, y_t)\}$ at the current iterate (x_t, y_t) by applying a Trim operator to the gradients collected from all agents (line 4); we will describe this operator shortly. The robust gradient estimates are then used to compute a mid-point $(\hat{x}_t, \hat{y}_t)$ by performing a projected primal-dual update (line 5). In the second step, the server now computes robust gradients at the mid-point (line 7), and performs a projected primal-dual update using these gradients to generate the next iterate (x_{t+1}, y_{t+1}). We now describe the Trim operation.

The Trim operator in equations (5) and (7) takes as input M vectors, and applies the univariate trimmed mean estimator in [11] - described in Algorithm 2 - to each coordinate of these vectors separately. To describe the trimmed mean estimator, suppose the data comprises of M independent copies of a scalar random variable Z with mean μ_Z and variance σ_Z^2. An adversary corrupts at most αM of these copies; the corrupted data-set is then made available to the estimator. The estimator splits the corrupted data set into two equal chunks, denoted by $Z_1, \ldots, Z_{M/2}$, $\tilde{Z}_1, \ldots \tilde{Z}_{M/2}$. One of the chunks is used to compute appropriate quantile levels for truncation (line 2 of Algo. 2). The robust estimate $\hat{\mu}_Z$ of μ_Z is an average of the data points in the other chunk, with those data points falling outside the estimated quantile levels truncated prior to averaging (line 3 of Algo. 2).

[4]Recall that $\{g_x(x_t, y_t; \xi_{1,t}^{(i)}), g_y(x_t, y_t; \xi_{1,t}^{(i)})\}$ could be arbitrary vectors for an adversarial agent $i \in \mathcal{B}$.

Algorithm 2 Univariate Trimmed-Mean Estimator [11]

Require: Corrupted data set $Z_1, \ldots, Z_{M/2}$, $\tilde{Z}_1, \ldots \tilde{Z}_{M/2}$, corruption fraction α, and confidence level δ.

1: Set $\epsilon = 8\alpha + 24\frac{\log(4/\delta)}{M}$.

2: Let $Z_1^* \le Z_2^* \le \cdots \le Z_{M/2}^*$ represent a non-decreasing arrangement of $\{Z_i\}_{i \in [M/2]}$. Compute quantiles: $\gamma = Z_{\epsilon M/2}^*$ and $\beta = Z_{(1-\epsilon)M/2}^*$.

3: Compute robust mean estimate $\hat{\mu}_Z$ as follows:

$$\hat{\mu}_Z = \frac{2}{M}\sum_{i=1}^{M/2} \phi_{\gamma,\beta}(\tilde{Z}_i); \phi_{\gamma,\beta}(x) = \begin{cases} \beta & x > \beta \\ x & x \in [\gamma, \beta] \\ \gamma & x < \gamma \end{cases}$$

The following result on the performance of Algorithm 2 will play a key role in our subsequent analysis of RDEG.

Theorem 1. *[11, Theorem 1] Consider the trimmed mean estimator in Algorithm 2. Suppose $\alpha \in [0, 1/16)$, and let $\delta \in (0, 1)$ be such that $\delta \ge 4e^{-M/2}$. Then, there exists an universal constant c, such that with probability at least $1-\delta$,*

$$|\hat{\mu}_Z - \mu_Z| \le c\sigma_Z\left(\sqrt{\alpha} + \sqrt{\frac{\log(1/\delta)}{M}}\right).$$

In the next section, we will provide rigorous guarantees on the performance of our proposed algorithm RDEG.

IV. Performance Guarantees for RDEG

Before stating our main results, we first make a standard smoothness assumption on the function $f(x, y)$.

Assumption 1. *There exists a constant $L > 0$ such that the following holds for all $x_1, x_2 \in \mathcal{X}$, and all $y_1, y_2 \in \mathcal{Y}$:*

$$\begin{aligned} \|\nabla_x f(x_1, y_1) - \nabla_x f(x_2, y_2)\| &\le L\left(\|x_1 - x_2\| + \|y_1 - y_2\|\right), \\ \|\nabla_y f(x_1, y_1) - \nabla_y f(x_2, y_2)\| &\le L\left(\|x_1 - x_2\| + \|y_1 - y_2\|\right). \end{aligned}$$

We now define a few key quantities that will show up in our main results. Let $\sigma_x = \sqrt{\sum_{j \in [n]} \sigma_x^2(j)}$, $\sigma_y = \sqrt{\sum_{k \in [m]} \sigma_y^2(k)}$, and $\sigma = \max\{\sigma_x, \sigma_y\}$. Moreover, let $d = \max\{n, m\}$, and $D = \max\{D_x, D_y\}$, where D_x and D_y are the diameters of the sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. With the above notations in place, we state our first main result that provides a bound on the primal-dual gap $\phi_T \triangleq \max_{y \in \mathcal{Y}} f(\bar{x}_T, y) - \min_{x \in \mathcal{X}} f(x, \bar{y}_T)$, where

$$\bar{x}_T = (1/T)\sum_{t \in [T]} \hat{x}_t, \text{ and } \bar{y}_T = (1/T)\sum_{t \in [T]} \hat{y}_t.$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:31 UTC from IEEE Xplore. Restrictions apply.

Theorem 2. *Suppose Assumption 1 holds, the fraction α of corrupted devices satisfies $\alpha \in [0, 1/16)$, and the number of agents M is sufficiently large: $M \geq 48\log(16dT^2)$. Then, with a step-size η satisfying $\eta \leq 1/(2L)$, and the confidence parameter δ in Algorithm 2 set to $\delta = 1/(4dT^2)$, RDEG guarantees the following with probability at least $1 - 1/T$:*

$$\phi_T \leq \frac{D^2}{\eta T} + \tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{\frac{1}{M}}\right)\right). \tag{9}$$

Discussion. Theorem 2 tells us that with high probability, the primal-dual gap ϕ_T converges to a ball of radius $\tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{1/M}\right)\right)$ at a $O(1/T)$ rate.[5] Notably, the primal-dual gap is zero if and only if $(\bar{x}_T, \bar{y}_T)$ is a saddle point of $f(x, y)$ over the set $\mathcal{X} \times \mathcal{Y}$. Thus, RDEG provably generates approximate saddle points. The following result is one of the main implications of Theorem 2.

Corollary 1. *Suppose the conditions in Theorem 2 hold. Then, RDEG guarantees the following with probability at least $1 - 1/T$:*

$$|f(\bar{x}_T, \bar{y}_T) - f(x^*, y^*)| \leq \frac{D^2}{\eta T} + \tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{\frac{1}{M}}\right)\right). \tag{10}$$

Corollary 1 tells us that with high probability, the function values $f(\hat{x}_t, \hat{y}_t)$ of the averaged iterates generated by RDEG converge to the saddle-point value $f(x^*, y^*)$ up to an error-floor of $\tilde{O}\left(\sigma D\left(\sqrt{\alpha} + \sqrt{1/M}\right)\right)$, at a $O(1/T)$ rate. There are several key messages from this result. First, in the absence of adversaries (i.e., when $\alpha = 0$), the classical extra-gradient algorithm with a constant step-size would yield convergence to the saddle-point value with an error floor of $\tilde{O}(\sigma(\sqrt{1/M}))$ at a $O(1/T)$ rate. Thus, modulo the biasing effect of the adversaries, the statistical performance of RDEG is *near-optimal*. Second, the additive biasing effect due to adversarial corruption shows up even in the context of stochastic minimization [12]. In fact, the authors in [12] argue that an additive biasing effect of order $\tilde{\Omega}(\alpha)$ is unavoidable, albeit for the minimization setting. This is all to say that the dependence of our rate on the corruption level in Eq. (10) is only to be expected. Third, when the corruption level is small, the benefit of collaboration is evident from the second term in Eq. (10): the variance σ arising from the noise term is effectively reduced by a factor of $\sqrt{M}$ due to the averaging effect of the normal agents. This effect will be aptly demonstrated by the simulations in Section VI.

We now turn to the goal of achieving faster convergence rates than those in Theorem 2. To that end, we study the performance of the RDEG algorithm for strongly convex-strongly concave (SC-SC) functions. Accordingly, we first make the following assumption on $f(x, y)$.

Assumption 2. *The function $f(x, y)$ is μ-strongly convex-μ-strongly concave (SC-SC) over $\mathcal{X} \times \mathcal{Y}$, i.e., for all $x_1, x_2 \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$, the following holds:*

$$f(x_2, y_1) \geq f(x_1, y_1) + \langle \nabla_x f(x_1, y_1), x_2 - x_1 \rangle + \frac{\mu}{2}\|x_2 - x_1\|^2,$$
$$f(x_1, y_2) \leq f(x_1, y_1) + \langle \nabla_y f(x_1, y_1), y_2 - y_1 \rangle - \frac{\mu}{2}\|y_2 - y_1\|^2.$$

For $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, define $z \triangleq [x; y]$. We have the following result for functions satisfying Assumption 2.

Theorem 3. *Suppose Assumptions 1 and 2 hold in conjunction with the assumptions on α and M in Theorem 2. Then, with $\delta = 1/(4dT^2)$ and step-size $\eta \leq 1/(4L)$, RDEG guarantees the following with probability at least $1 - 1/T$:*

$$\|z^* - z_{T+1}\|^2 \leq 2e^{-\frac{T}{4\kappa}} D^2 + \tilde{O}\left(\frac{\sigma D \kappa}{L}\left(\sqrt{\alpha} + \sqrt{\frac{1}{M}}\right)\right), \tag{11}$$

where $\kappa = \mu/L$.

Theorem 3 says that for smooth strongly convex-strongly concave functions, the iterates generated by RDEG converge linearly to a ball around the saddle point (x^*, y^*) with high probability. The size of the ball is dictated by the second term in Eq. (11).

Remark 1. *(Comments on α) The requirement that the fraction of corruption $\alpha \in [0, 1/16)$ in our results is inherited from the analysis of the trimmed mean estimator in [11]. One can potentially tolerate a larger fraction of corruption (up to $\alpha < 1/2$) by using the robust estimators in [12]. However, this would likely come at a price: the authors in [12] impose additional statistical assumptions on the partial gradients; we do not make such assumptions.*

Remark 2. *(Comments on M) In our results, we need the number of agents M to scale with $\log(dT)$. We note that similar conditions show up in the context of adversarially-robust distributed statistical learning; see, for instance, [12] and [25]. In fact, the covering argument in [12] requires M to scale linearly with the model dimension d. By avoiding such an argument in our analysis, we can get by with a far milder logarithmic dependence on d. As an example, for $d = 100$, and number of iterations $T = 2^{10}$ (which should suffice for all practical purposes), $\log(dT) \approx 12$. This is a very reasonable requirement for large-scale computing systems where the number of devices is of the order of thousands. Furthermore, with $T = 2^{10}$, our guarantees in Theorems 2 and 3 hold with probability $1 - 1/T \approx 1$.*

V. Proof Sketch of Theorem 2

In this section, we provide a sketch of the proof of Theorem 2. Due to space constraints, detailed proofs of our main results (including that of Theorem 3) are omitted here, but can be found in [28]. Essentially, our proofs comprise of a perturbation analysis of the extra-gradient algorithm, where the perturbations arise due to adversarial corruption. As the starting point of such an analysis, we establish some simple relations in the following lemma.

[5] In the statement of our results, we will use the $\tilde{O}(\cdot)$ notation to hide terms that are logarithmic in n, m, and T.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:31 UTC from IEEE Xplore. Restrictions apply.

Lemma 1. *For the RDEG algorithm, the following inequalities hold for all $t \in [T], x \in \mathcal{X}$, and $y \in \mathcal{Y}$:*

$$\begin{aligned}
2\eta\langle \tilde{g}_x(x_t,y_t), \hat{x}_t - x\rangle &\le \|x-x_t\|^2 - \|x-\hat{x}_t\|^2 - \|\hat{x}_t - x_t\|^2\\
-2\eta\langle \tilde{g}_y(x_t,y_t), \hat{y}_t - y\rangle &\le \|y-y_t\|^2 - \|y-\hat{y}_t\|^2 - \|\hat{y}_t - y_t\|^2\\
2\eta\langle \tilde{g}_x(\hat{x}_t,\hat{y}_t), x_{t+1} - x\rangle &\le \|x-x_t\|^2 - \|x-x_{t+1}\|^2 - \|x_{t+1} - x_t\|^2\\
-2\eta\langle \tilde{g}_y(\hat{x}_t,\hat{y}_t), y_{t+1} - y\rangle &\le \|y-y_t\|^2 - \|y-y_{t+1}\|^2 - \|y_{t+1} - y_t\|^2.
\end{aligned} \tag{12}$$

Using the previous result, our next goal is to track the progress made by the mid-point vector $(\hat{x}_t, \hat{y}_t)$ in each iteration, as a function of the errors introduced by adversarial corruption. To that end, for each $\bar{x} \in \mathcal{X}$ and $\bar{y} \in \mathcal{Y}$, we define the following errors vectors:

$$e_x(\bar{x},\bar{y}) \triangleq \tilde{g}_x(\bar{x},\bar{y}) - \nabla_x f(\bar{x},\bar{y}); e_y(\bar{x},\bar{y}) \triangleq \tilde{g}_y(\bar{x},\bar{y}) - \nabla_y f(\bar{x},\bar{y}). \tag{13}$$

We have the following key lemma.

Lemma 2. *Suppose Assumption 1 holds and $\eta \le 1/(2L)$. For the RDEG algorithm, the following then holds for all $t \in [T], x \in \mathcal{X}$, and $y \in \mathcal{Y}$:*

$$\begin{aligned}
&\eta\langle \nabla_x f(\hat{x}_t,\hat{y}_t), \hat{x}_t - x\rangle - \eta\langle \nabla_y f(\hat{x}_t,\hat{y}_t), \hat{y}_t - y\rangle\\
&\le \frac{1}{2}\left(\|x-x_t\|^2 - \|x-x_{t+1}\|^2 + \|y-y_t\|^2 - \|y-y_{t+1}\|^2\right)\\
&+ \eta D\left(\|e_x(x_t,y_t)\| + \|e_x(\hat{x}_t,\hat{y}_t)\| + \|e_y(x_t,y_t)\| + \|e_y(\hat{x}_t,\hat{y}_t)\|\right).
\end{aligned} \tag{14}$$

In our next result, we establish high-probability bounds on the error vectors by leveraging Theorem 1.

Lemma 3. *Consider the event $\mathcal{H}_t$ defined as follows:*

$$\mathcal{H}_t \triangleq \{\max\{\|e_x(x_t,y_t)\|, \|e_x(\hat{x}_t,\hat{y}_t)\|, \|e_y(x_t,y_t)\|, \|e_y(\hat{x}_t,\hat{y}_t)\|\} \le \Delta\},$$

where

$$\Delta = c\sigma\left(\sqrt{\alpha} + \sqrt{\frac{\log(4dT^2)}{M}}\right). \tag{15}$$

For the RDEG algorithm, we have:

$$\mathbb{P}(\mathcal{H}_t) \ge 1 - \frac{1}{T^2}, \text{ for each } t \in [T]. \tag{16}$$

Proof. We begin by defining certain "good" events:

$$\begin{aligned}
\mathcal{G}_{x,t} &\triangleq \{\|e_x(x_t,y_t)\| \le \Delta\}, \mathcal{G}_{y,t} \triangleq \{\|e_y(x_t,y_t)\| \le \Delta\},\\
\bar{\mathcal{G}}_{x,t} &\triangleq \{\|e_x(\hat{x}_t,\hat{y}_t)\| \le \Delta\}, \bar{\mathcal{G}}_{y,t} \triangleq \{\|e_y(\hat{x}_t,\hat{y}_t)\| \le \Delta\}.
\end{aligned}$$

To analyze the probability of occurrence of the above events, we need to next define an appropriate filtration. Accordingly, let $\mathcal{F}_t$ denote the sigma field generated by $\{x_k, y_k\}_{k\in[t]}$ and $\{\hat{x}_k, \hat{y}_k\}_{k\in[t-1]}$; and $\bar{\mathcal{F}}_t$ denote the sigma field generated by $\{x_k, y_k\}_{k\in[t]}$ and $\{\hat{x}_k, \hat{y}_k\}_{k\in[t]}$. From definition, we have

$$\mathcal{F}_1 \subset \bar{\mathcal{F}}_1 \subset \mathcal{F}_2 \subset \bar{\mathcal{F}}_2 \subset \cdots \subset \mathcal{F}_T \subset \bar{\mathcal{F}}_T.$$

Clearly, (x_t, y_t) is $\mathcal{F}_t$-measurable. Thus, conditioned on $\mathcal{F}_t$, for each coordinate $j \in [n]$, the data set $\{[g_x(x_t,y_t;\xi_{1,t}^{(i)})]_j : i \in [M]\}$ has the following properties: (i) at most αM of the samples are corrupted; and (ii) the uncorrupted samples are i.i.d. scalar random variables with mean $[\nabla_x f(x_t,y_t)]_j$ and variance bounded above by $\sigma_x^2(j)$. Invoking Theorem 1 for the trimmed mean estimator in Algorithm 2, we conclude that conditioned on $\mathcal{F}_t$, with probability at least $1 - 1/(4dT^2)$,

$$|[\tilde{g}_x(x_t,y_t)]_j - [\nabla_x f(x_t,y_t)]_j| \le c\sigma_x(j)\left(\sqrt{\alpha} + \sqrt{\tfrac{\log(4dT^2)}{M}}\right).$$

Now union-bounding over each of the n coordinates, we have that conditioned on $\mathcal{F}_t$, with probability at least $1 - \frac{n}{4dT^2} \ge 1 - \frac{1}{4T^2}$,

$$\|\tilde{g}_x(x_t,y_t) - \nabla_x f(x_t,y_t)\| \le \Delta.$$

Here, we used the fact that $d = \max\{n,m\} \ge n$, and $\sqrt{\sum_{j\in[n]} \sigma_x^2(j)} = \sigma_x \le \sigma$. We have thus shown that $\mathbb{P}(\mathcal{G}_{x,t}|\mathcal{F}_t) \ge 1 - 1/(4T^2)$. Using an identical argument, we can establish an analogous result for the event $\mathcal{G}_{y,t}$. An union bound thus yields $\mathbb{P}(\mathcal{G}_t|\mathcal{F}_t) \ge 1 - 1/(2T^2)$, where $\mathcal{G}_t = \mathcal{G}_{x,t} \cap \mathcal{G}_{y,t}$. Noting that $(\hat{x}_t, \hat{y}_t)$ is $\bar{\mathcal{F}}_t$-measurable, we can similarly show that $\mathbb{P}(\bar{\mathcal{G}}_t|\bar{\mathcal{F}}_t) \ge 1 - 1/(2T^2)$, where $\bar{\mathcal{G}}_t = \bar{\mathcal{G}}_{x,t} \cap \bar{\mathcal{G}}_{y,t}$. Our next task is to analyze the probability of occurrence of the event $\mathcal{H}_t = \mathcal{G}_t \cap \bar{\mathcal{G}}_t$ by exploiting the nested sigma-field structure: $\mathcal{F}_t \subset \bar{\mathcal{F}}_t$. To that end, observe:

$$\begin{aligned}
\mathbb{P}(\bar{\mathcal{G}}_t|\mathcal{F}_t) &= \mathbb{E}[1_{\bar{\mathcal{G}}_t}|\mathcal{F}_t]\\
&\overset{(a)}{=} \mathbb{E}[\mathbb{E}[1_{\bar{\mathcal{G}}_t}|\bar{\mathcal{F}}_t]|\mathcal{F}_t]\\
&= \mathbb{E}[\mathbb{P}(\bar{\mathcal{G}}_t|\bar{\mathcal{F}}_t)|\mathcal{F}_t]\\
&\overset{(b)}{\ge} 1 - \frac{1}{2T^2}.
\end{aligned} \tag{17}$$

Here, we used $1_{\mathcal{A}}$ to represent the indicator random variable for an event $\mathcal{A}$. For (a), we used the fact that given a random variable X and two sigma-fields $\mathcal{B}_1$ and $\mathcal{B}_2$ with $\mathcal{B}_1 \subset \mathcal{B}_2$, it holds that $\mathbb{E}[\mathbb{E}[X|\mathcal{B}_2]|\mathcal{B}_1] = \mathbb{E}[X|\mathcal{B}_1]$, i.e., the smaller sigma-field "wins" [29, Theorem 5.1.6]. For (b), we used the previously established fact that $\mathbb{P}(\bar{\mathcal{G}}_t|\bar{\mathcal{F}}_t) \ge 1 - 1/(2T^2)$. Using (17) and an union bound, we conclude that $\mathbb{P}(\mathcal{H}_t|\mathcal{F}_t) = \mathbb{P}(\mathcal{G}_t \cap \bar{\mathcal{G}}_t|\mathcal{F}_t) \ge 1 - 1/T^2$. To complete the proof, we note that

$$\mathbb{P}(\mathcal{H}_t) = \mathbb{E}[1_{\mathcal{H}_t}] = \mathbb{E}[\mathbb{E}[1_{\mathcal{H}_t}|\mathcal{F}_t]] = \mathbb{E}[\mathbb{P}(\mathcal{H}_t|\mathcal{F}_t)] \ge 1 - \frac{1}{T^2}.$$

□

The proof of Theorem 2 is a fairly simple consequence of Lemma's 1, 2, and 3. In particular, it follows by conditioning on the clean event $\mathcal{H} = \bigcap_{t\in[T]} \mathcal{H}_t$, where $\mathcal{H}_t$ is as defined in Lemma 3, and exploiting the convex-concave property of $f(x,y)$ in a standard way. For details, see [28].

VI. Simulations

In this section, we study a specific instance of problem (1), namely, a bilinear game of the following form:

$$\min_{\|x\|\le\rho} \max_{\|y\|\le\rho} f(x,y) \triangleq \mathbb{E}[x^T A y + 2(b+\zeta)^T x - 2(c+\zeta)^T y].$$

Here, $x, y, b, c \in \mathbb{R}^{10}$, $A \in \mathbb{R}^{10\times 10}$, and $\rho = 100$. The parameters A, b, c are fixed, and $\zeta \sim N(0, \sigma^2 I)$. As our measure of performance, we consider the instantaneous primal-dual gap $\phi_t = \max_{y\in\mathcal{Y}} f(\bar{x}_t, y) - \min_{x\in\mathcal{X}} f(x, \bar{y}_t)$. We simulate two algorithms: the vanilla extra-gradient algorithm that does not

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:31 UTC from IEEE Xplore. Restrictions apply.

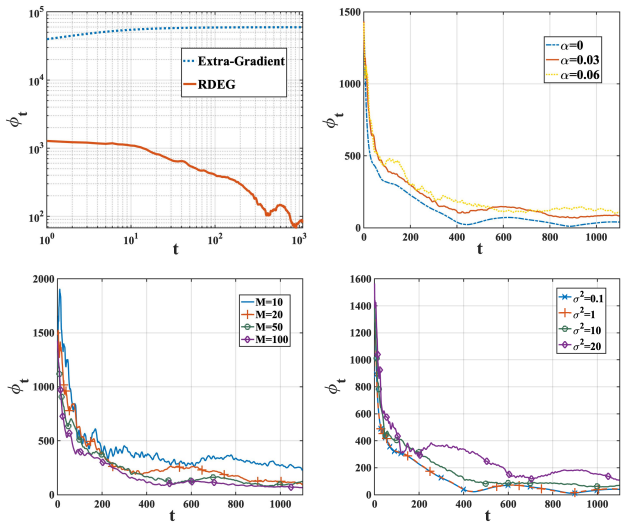


Fig. 2. **Top Left (a).** Comparison between vanilla extra-gradient and RDEG. **Top Right (b).** Performance of RDEG vs. level of corruption fraction. **Bottom Left (c).** Performance of RDEG vs. number of agents. **Bottom Right (d).** Performance of RDEG vs. level of noise variance.

account for adversaries, and the proposed RDEG algorithm. In Fig. 2(a), we plot the performance of these algorithms with $\alpha = 0.06$, $M = 100$, and $\sigma^2 = 10$. We observe that even a small number of Byzantine workers can cause the extra-gradient algorithm to diverge from the saddle point. In Fig. 2(b), with $M = 100$ and $\sigma^2 = 10$, we explore the impact of varying the corruption fraction α. Complying with Theorem 2, the error floor of RDEG increases as a function of α. Next, in Fig. 2(c), to demonstrate the benefit of collaboration, we fix $\alpha = 0.06$ and $\sigma^2 = 10$, and plot the performance of RDEG as a function of the number of agents M. As expected, by increasing M, RDEG converges to a smaller ball around the saddle point, highlighting the benefit of collaboration in reducing the variance of the noise model. Finally, in Fig. 2(d), we fix $M = 100$ and $\alpha = 0.06$, and change the variance of the noise σ^2. We observe that increasing σ^2 leads to a higher error-floor. Importantly, all of the above plots verify the bound in Theorem 2.

VII. Conclusion

We studied the problem of distributed min-max learning under adversarial agents for the first time. By exploiting recent ideas from robust statistics, we developed a novel robust distributed extra-gradient algorithm. For both smooth convex-concave and smooth strongly convex-strongly concave functions, we showed that with high probability, our proposed approach guarantees convergence to approximate saddle points at near-optimal statistical rates.

References

[1] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton university press, 2007.

[2] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[3] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton university press, 2009.

[4] B. Dai, N. He, Y. Pan, B. Boots, and L. Song, "Learning from conditional distributions via dual embeddings," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1458–1467.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[7] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust federated learning: The case of affine distribution shifts," *Advances in Neural Information Proc. Systems*, vol. 33, pp. 21 554–21 565, 2020.

[8] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 203–226.

[9] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, "Training gans with optimism," *arXiv preprint arXiv:1711.00141*, 2017.

[10] G. M. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, pp. 747–756, 1976.

[11] G. Lugosi and S. Mendelson, "Robust multivariate mean estimation: the optimality of trimmed mean," *The Annals of Statistics*, vol. 49, no. 1, pp. 393–410, 2021.

[12] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[13] A. Nemirovski, "Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 229–251, 2004.

[14] A. Nedić and A. Ozdaglar, "Subgradient methods for saddle-point problems," *Journal of optimization theory and applications*, vol. 142, no. 1, pp. 205–228, 2009.

[15] T. Liang and J. Stokes, "Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks," in *AISTATS*. PMLR, 2019, pp. 907–915.

[16] C. Daskalakis and I. Panageas, "The limit points of (optimistic) gradient descent in min-max optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[17] A. Mokhtari, A. E. Ozdaglar, and S. Pattathil, "Convergence rate of o(1/k) for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems," *SIAM Journal on Optimization*, vol. 30, no. 4, pp. 3230–3251, 2020.

[18] T. Lin, C. Jin, and M. I. Jordan, "Near-optimal algorithms for minimax optimization," in *Conference on Learning Theory*. PMLR, 2020, pp. 2738–2779.

[19] L. Su and N. H. Vaidya, "Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms," in *Proc. of the 2016 ACM symposium on principles of distributed computing*, 2016, pp. 425–434.

[20] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.

[21] N. Ravi, A. Scaglione, and A. Nedić, "A case of distributed optimization in adversarial environment," in *ICASSP*, 2019, pp. 5252–5256.

[22] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "Draco: Byzantine-resilient distributed training via redundant gradients," in *ICML*. PMLR, 2018, pp. 903–912.

[24] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. of the ACM on Measurement and Anal. of Comp. Sys.*, vol. 1, no. 2, pp. 1–25, 2017.

[25] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, 2022.

[26] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[27] Y. Cheng, I. Diakonikolas, and R. Ge, "High-dimensional robust mean estimation in nearly-linear time," in *Proc. of the thirtieth annual ACM-SIAM symp. on discrete algorithms*. SIAM, 2019, pp. 2755–2771.

[28] A. Adibi, A. Mitra, G. J. Pappas, and H. Hassani, "Distributed statistical min-max learning in the presence of Byzantine agents," *arXiv preprint arXiv:2204.03187*, 2022.

[29] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2019, vol. 49.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:31 UTC from IEEE Xplore. Restrictions apply.