Adversary-Resilient Distributed Estimation using Intermittent and Heterogeneous Data with Application to Network Tomography

Gugan Thoppe, Mihir Dhankshiru, Nibedita Roy, Alexandre Reiffers-Masson, Naman Naman, Alexandre Azor

Adversary-Resilient Distributed Estimation using Intermittent and Heterogeneous Data with Application to Network Tomography

Gugan Thoppe[*†], **Mihir Dhankshiru**[*], **Nibedita Roy**[*], **Alexandre Reiffers-Masson**[‡], **Naman**[*], **Alexandre Azor**[†]

Abstract

Robustness to adversarial attacks in online distributed learning within a single parameter server and multiple worker node framework is a critical research area. Traditional methods for this setup achieve resilience through a two-step process in each iteration. First, all worker nodes synchronize to compute and communicate identical quantities, such as gradients at a shared point, to the server. Then, the server uses a robust aggregate of these quantities—obtained via techniques like the median or trimmed mean—to update its solution estimate. However, this approach falls short in applications like network tomography, where the measurements across different nodes are sporadic and heterogeneous. A novel two-timescale algorithm was recently proposed to deal with such scenarios. In this study, we establish that its convergence rate is $O(1/\sqrt{n})$, which is optimal for non-strongly convex optimization. Separately, due to the sporadic nature of data, it is inevitable that this rate expression degrades when more honest agents are incorporated into the system. For a fixed number of adversaries, our work reveals that this degradation is of order $O(\sqrt{N})$, where N is the number of worker nodes. Lastly, we demonstrate the applicability of this algorithm and our theoretical results to network tomography.

1 Introduction

Distributed systems, involving a central server and multiple worker nodes, have emerged as the prevailing approach in large-scale estimation [Fan et al., 2019, Castro et al., 2004] and machine learning [Li et al., 2014, Verbraeken et al., 2020, Zhang et al., 2021]. In recent times, though, the focus has shifted towards robust estimation or learning when some of worker nodes are adversarial (Byzantine) or faulty [Yang et al., 2020, Bouhata et al., 2022]. Broadly, these adversary-resilient methods can be grouped into two categories: those that rely on some robust aggregation mechanism, such as trimmed mean or geometric median, to filter out outlier or malicious measurements [Blanchard et al., 2017, Yin et al., 2018], and those that employ coding schemes to add redundancy and enable recovery of the correct estimate Chen et al. [2018]. A major limitation of most of these methods is their requirement for worker nodes to synchronously compute and communicate their local measurements with the server. However, in applications such as network tomography [Tsang et al., 2003, Vardi, 1996] or Vertical Federated Learning (VFL) [Liu et al., 2024], data from various worker

[*]Indian Institute of Science, Bengaluru 560012, India
[†]Corresponding author
[‡]IMT Atlantique, Plouzané 29280, France

Preprint. Under review.

nodes is only available intermittently. Hence, the above constraint limits the utility of the aforementioned approaches to such applications.

The above discussion bring into focus robust distributed optimization methods that can work with asynchronous workers. To our knowledge, there are only four such methods documented: Kardam [Damaskinos et al., 2018], (ii.) Zeno++ [Xie et al., 2020], (iii.) AFLGuard [Fang et al., 2022], and (iv.) BASGD [Yang and Li, 2021]. These techniques adapt the principles of robust aggregation from synchronous methods to scenarios where inputs from only a few workers are available at any given time. They are designed to manage situations where worker nodes may lag due to varying computational or communication resources. Kardam, Zeno++, and AFLGuard employ a complex scoring system to exclude malicious inputs, but Kardam has the drawback of mistakenly filtering out correct estimates during attacks [Xie et al., 2020]. Meanwhile, both Zeno++ and AFLGuard depend on a separate validation dataset stored on the parameter server, which poses privacy concerns. Additionally, BASGD operates asynchronously only on the client-side, with server updates still governed by the slowest participating clients. On top of these limitations, all these methods need the worker node measurements to be based on homogeneous datasets. Consequently, these methods are also not useful to network tomography and VFL-type applications, wherein the data at different worker nodes could be completely different.

Recently, Ganesh et al. [2023] proposed a novel coding-based algorithm, building upon ideas from [Fawzi et al., 2014], for robustly estimating the mean of a random vector from an (abstract) linear measurement model and established its convergence. In this work, we study the *convergence rate* of this algorithm and discuss its utility as a robust distributed estimation scheme for network tomography. We now provide a brief background on network tomography and discuss how Ganesh et al. [2023]'s approach is applicable here.

Network tomography is a powerful tool to estimate the internal state of a closed network, which cannot be measured directly, by using end-to-end external measurements. It is useful for estimating network properties such as link delays or traffic volumes which are crucial in network management and troubleshooting. Typically, this problem can be cast as follows: Estimate statistics of a random vector X using samples of another random vector Y which relates to X via the relation $Y = PX$, where P is the path-link matrix. The matrix P is assumed to be a priori known and contains information about the different measurement paths and the links/edges contained in them. The challenge in solving the above problem is due to the fact that the $Y = PX$ relation is usually underdetermined. The difficulty becomes even more acute when the worker nodes responsible for path measurements are adversarial or faulty [Zhao et al., 2017, Chiu and He, 2021]. The literature on this latter problem is very sparse. There are a few works that study the case where some unknown set of links fail; hence, measurements along paths containing those links are no longer available. To deal with such scenarios, the algorithms proposed, e.g., [Tati et al., 2014], add redundancy to measurements by including additional paths so that estimates along non-failed links can be continued. These works do not consider the adversarial case. There are also a few works, e.g., [Yao et al., 2012], that look at characterizing the network under conditions of errors, which may be random or adversarial. These methods leverage network coding techniques to identify and localize the errors they introduce into the network. Specifically, they rely on the ability to observe the network's overall response to known or controlled inputs and compare these to the expected outcomes based on a network coding scheme. Due to these reasons, these methods have high computational complexity and are extremely sophisticated since they aim to identify the precise link that is the source of these errors.

Our main contributions are as follows. We first derive the convergence rate, in the L^2 sense, of the algorithm proposed in [Ganesh et al., 2023] (see Section 2). We show that its rate is $O(1/\sqrt{n})$, which is optimal for the family of optimization methods it falls under (non-strongly convex optimization). Note that we get this optimality even with adversaries, which is a first for a distributed estimation scheme to our knowledge. Next, we show that, due its fully asynchronous nature, there is a $O(\sqrt{N})$ degradation in its convergence rate as the number N of measurement paths increases. Finally, in Section 4, we demonstrate how this algorithm can be used as an extremely simple yet robust distributed estimation scheme for network tomography. The simplicity is there because this algorithm does not to detect the adversarial

Algorithm 1: Online distributed algorithm to estimate $\mathbb{E}X$ [Ganesh et al., 2023]

Input : stepsize sequences (α_n) and (β_n), projection set $\mathcal{X}$, and observation matrix A

Initialize estimates of $\mathbb{E}X$ and $\mathbb{E}Y$ at the server to $x_0 \in \mathcal{X}$ and $y_0 = 0 \in \mathbb{R}^N$, respectively

for *each iteration* $n \geq 0$

Central server

Sample agent index $i \equiv i_{n+1} \in [N]$ uniformly randomly

Update $\mathbb{E}X$ estimate using

$$x_{n+1} = \Pi_{\mathcal{X}}\big(x_n + \alpha_n a_i \text{sign}(y_n(i) - a_i^\top x_n)\big)$$

Chosen agent $i \in [N]$

if *agent* i *is honest* **then**

Obtain a sample $Y_{n+1}(i) \overset{\text{IID}}{\sim} Y(i)$ and sent it to the central server

else

Assign some (possibly malicious) value to $Y_{n+1}(i)$ and send it to the central server

end

Central server

Update $\mathbb{E}Y$ estimate using

$$y_{n+1}(j) = y_n(j) + \beta_n[NY_{n+1}(i)\mathbb{1}\{j = i\} - y_n(j)], \quad \forall j \in [N]$$

end

paths, but only ensures robustness in estimation. We also provide empirical evidence for tightness of our $O(\sqrt{N})$ degradation rate.

2 Setup, algorithm, and main result

Let $X \in \mathbb{R}^d$ and $Y = AX \in \mathbb{R}^N$ be two random vectors, where $A \in \mathbb{R}^{N \times d}$, with $N > d$, is some a priori known tall matrix. Ganesh et al. [2023] proposed a novel online algorithm to estimate $\mathbb{E}X$, the mean of X, in a distributed framework with adversaries and only sporadic access to samples of each $Y(i)$, the i-th coordinate of Y. They also established this algorithm's almost sure convergence. Below, we describe the setup, the algorithm, and this convergence result from [Ganesh et al., 2023]. Thereafter, we discuss this algorithm's convergence rate, which forms our main result. In Section 4, we additionally discuss how our work can be utilized to solve the network tomography problem.

Setup: The distributed setup consists of a single parameter server and N ($> d$) worker nodes, where a fixed but unknown subset $\mathcal{A} \subseteq [N] := \{1, \ldots, N\}$ of nodes, with $|\mathcal{A}| \leq m$, is adversarial. Further, agent $i \in [N]$ is equipped to obtain an independent and identically distributed (IID) sample of the random variable $Y(i) := a_i^\top X$, each time it is queried, where $a_i^\top$ is the i-th row of A.

Algorithm: The pseudo-code for the method proposed in [Ganesh et al., 2023] for estimating $\mathbb{E}X$ in the above framework is given in Algorithm 1. Each iteration of this algorithm has three phases. In the first phase, the server picks an agent index[4] $i_{n+1} \in [N]$ uniformly at random and updates the estimate of $\mathbb{E}X$ using Step 3. In this step, $\Pi_{\mathcal{X}}$ refers to the Euclidean projection on to the set $\mathcal{X}$, which is presumed to contain $\mathbb{E}X$. Further, for any $r \in \mathbb{R}$, $\text{sign}(r) = 1$ (resp. -1) if $r > 0$ (resp. $r < 0$) and $= 0$ when $r = 0$. In the second phase, agent i sets $Y_{n+1}(i)$ to be an independently obtained sample of $Y(i)$, if it is *honest*, and to some (potentially malicious) value, otherwise. Thereafter, agent i communicates this value to the server. In the final phase, the central server uses the value of $Y_{n+1}(i)$ to update its estimate of $\mathbb{E}Y$ as shown in Step 7. When all the agents are honest, the update rules for x_n

[4] At several places, we suppress i_{n+1}'s dependence on n for notational simplicity.

and y_n at the server can be viewed as a stochastic gradient descent algorithm for minimizing

$$f(x) := N^{-1}\|Ax - \mathbb{E}Y\|_1. \tag{1}$$

To deal with adversaries, this algorithm relies on the redundancy in the measurement model $Y = AX$ that is enforced on account of the observation matrix A being tall.

Remark 2.1. *In the distributed learning with adversaries literature, it is typically assumed that the data at different agents is roughly similar and that their estimates are more or less available at the same time. In contrast, the setup above has the following two significant differences:*

1. ***Heterogeneity**: For $i \neq j$, the measurement $Y(i)$ at agent i could have a very different distribution to that of $Y(j)$ based on the dissimilarity in the values of a_i and a_j.*
2. ***Sporadic or Intermittent Data**: At any instance $n \geq 0$, data from only one agent is available at the server. Furthermore, the data from any agent i is available only sporadically at the server with the mean time between two successive estimates being N.*

The main contribution of [Ganesh et al., 2023] is in using differential inclusion theory to show that $x_n \xrightarrow{a.s.} \mathbb{E}[X]$ for $\mathcal{X} = \mathbb{R}^d$, i.e., their result applies to the case where $\Pi_{\mathcal{X}}$ is identity. Their result holds under the following set of assumptions:

$\mathcal{A}_1$. ***Target vector***: There exist $\bar{\mu}, \bar{\sigma} > 0$ with $|\mathbb{E}X(j)| \leq \bar{\mu}$ and $\mathrm{Var}\big(X(j)\big) \leq \bar{\sigma}^2$ for all $j \in [d]$.

$\mathcal{A}_2$. ***Observation matrix***: The matrix A has full column rank and satisfies

$$\sum_{i \in S^c} |a_i^T x| > \sum_{i \in S} |a_i^T x| \tag{2}$$

for all $x \in \mathbb{R}^d \setminus 0$ and all $S \subseteq [N]$ with $|S| = m$.

$\mathcal{A}_3$. ***Stepsize***: The sequences $(\alpha_n)_{n\geq 0}$ and $(\beta_n)_{n\geq 0}$ are monotonically decreasing positive numbers such that $\max\{\alpha_0, \beta_0\} \leq 1$, $\sum_{n\geq 0} \alpha_n = \sum_{n\geq 0} \beta_n = \infty$, $\lim_{n\to\infty} \alpha_n/\beta_n = \lim_{n\to\infty} \beta_n = 0$, and $\max\{\sum_{n\geq 0} \alpha_n^2, \sum_{n\geq 0} \beta_n^2, \sum_{n\geq 0} \alpha_n\gamma_n\} < \infty$, where $\gamma_n = \sqrt{\beta_n \ln(\sum_{k=0}^n \beta_k)}$.

An example of the stepsizes satisfying $\mathcal{A}_3$ is $\alpha_n = n^{-\alpha}$, $\alpha \in (2/3, 1]$, and $\beta_n = n^{-\beta}$, $\beta \in (1/2, 1] \cap (2(1-\alpha), \alpha)$. Because $\alpha_n/\beta_n \to 0$, Algorithm 1 is a two-timescale algorithm with x_n denoting the slow update and y_n the fast one.

Main result: In Theorem 2.3, our main result, we state the convergence rate of Algorithm 1. Unlike [Ganesh et al., 2023], though, we need the following assumptions on stepsize sequences (α_n) and (β_n), instead of $\mathcal{A}_3$, and on $\mathcal{X}$:

$\mathcal{A}_3'$. ***Stepsize***: For $n \geq 0$, $\alpha_n = 1/\sqrt{n+1}$ and $\beta_n = 1/(n+1)$.

$\mathcal{A}_4'$. ***Projection set***: $\mathcal{X}$ is a non-empty, compact, and convex set containing $\mathbb{E}X$.

Remark 2.2. *With $\mathcal{A}_3'$, we have that $\alpha_n/\beta_n \to \infty$, i.e., β_n's decay rate is faster than that of α_n. Clearly, this stepsize choice does not satisfy $\mathcal{A}_3$. Hence, the almost sure convergence result from Ganesh et al. [2023] does not apply in this case. However, our main result below shows that this stepsize choice leads to the optimal convergence rate in the L^2 sense.*

We need a few notations to state our main result. For $0 \leq i \leq n$ and $i \leq j \leq n$, let

$$\tilde{\alpha}_j \equiv \tilde{\alpha}_j^{i,n} = \frac{\alpha_j}{\sum_{k=i}^n \alpha_k} \tag{3}$$

and

$$\tilde{x}_i^n = \sum_{j=i}^n \tilde{\alpha}_j x_j. \tag{4}$$

Further, let $\bar{A} := \max_{i\in[N]} \|a_i\|$ and

$$\eta := \min_{S:|S|=m} \min_{x\neq 0} \frac{1}{N\|x\|} \left(\sum_{i\in S^c} |a_i^\top x| - \sum_{i\in S} |a_i^\top x| \right), \tag{5}$$

where $\|\cdot\|$ denotes the Euclidean norm. From [Ganesh et al., 2023, Lemma 2], we have that $\eta > 0$. Due to this reason, $K := \frac{2m\bar{A}}{N\eta} + 1$ is well defined. Finally, let $D_{\mathcal{X}} := \max_{x\in\mathcal{X}} \|x - x_0\|$, where $x_0 \in \mathcal{X}$ is the initial estimate for $\mathbb{E}X$.

Theorem 2.3 (Main result). *Suppose $N \geq 4$, $m \leq N/2$, and $i = \lceil rn \rceil$ for some fixed $r \in (0,1)$, where $\lceil\cdot\rceil$ is the ceil function. Further, suppose $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3'$, and $\mathcal{A}_4'$ hold. Then, for any $n \geq 2$,*

$$0 \leq \mathbb{E}[f(\tilde{x}_i^n) - f(\mathbb{E}X)] \leq \left(\frac{N-m}{\sqrt{N}}\right) \frac{C_x}{\sqrt{n}}, \tag{6}$$

and, for any $n \geq 1$ and $i \in \mathcal{A}^c$,

$$\mathbb{E}|y_n(i) - \mathbb{E}Y(i)| \leq \sqrt{N} \frac{C_y}{\sqrt{n}}, \tag{7}$$

where $C_x, C_y \geq 0$ are two constants given by

$$C_y := \sqrt{d\bar{A}^2(\bar{\sigma}^2 + \bar{\mu}^2)} \qquad \text{and} \qquad C_x := \frac{K(4D_{\mathcal{X}}^2 - (4C_y + \bar{A}^2)\ln r)}{4 - 2\sqrt{2r+3}}. \tag{8}$$

Remark 2.4. *By substituting $\beta_n = 1/(n+1)$ in Step 7 of Algorithm 1, it follows that $y_n(i) = n^{-1}\sum_{k=1}^n NY_k(i_k)\mathbb{1}\{i_k = i\}$ for $n \geq 1$. The RHS shows that other agents' measurements do not influence $(y_n(i))_{n\geq 0}$'s behavior when agent i is honest. In this case, the simple average also confirms that the convergence rate of $O(\sqrt{N}/\sqrt{n})$ is optimal for how fast $|y_n(i) - \mathbb{E}Y(i)|$ decays, where $\sqrt{N}$ is due to the sporadic nature of obtaining $Y(i)$'s samples. In contrast, since adversaries can do anything, no guarantee can be provided about $(y_n(i))$'s behavior when agent i is adversarial.*

Remark 2.5. *Step 3 in Algorithm 1 shows that the sequence (x_n) is influenced by $(y_n(i))$'s behavior for all $i \in [N]$, including the adversarial ones. This influence is unavoidable since the central server cannot identify the adversarial agents. However, the inclusion of the sign function ensures that an adversary, at worst, can only invert the actual sign value at given time instance. Our main result shows that the rate at which $\mathbb{E}\big[f(\tilde{x}_i^n) - \mathbb{E}f(\mathbb{E}X)\big]$ decays to 0 is $O(1/\sqrt{n})$, which is optimal for the non-strongly convex optimization setting [Nemirovski et al., 2009, Section 2.2] even in the absence of adversaries. Additionally, due to the sporadic availability of measurements across agents and the fully asynchronous nature of the algorithm, a degradation in its convergence rate is inevitable as the number of worker nodes N increases. Our result shows that this degradation is $O(\sqrt{N})$ when N increases, provided there is a consistent lower (resp. upper) bound on η (resp. $\bar{A}$). The numerical study in Section4 provides compelling evidence that this degradation rate is indeed tight.*

3 Proof of our main result

In this section, we give the full proof of y_n's convergence rate as it is straightforward. For our proof of x_n's convergence rate, which is a bit involved, we only give an outline and focus on highlighting the key challenges and our novelty in handling them. The actual proof is given in Appendix A.

We first establish the convergence rate of the (y_n) sequence.

Proof of (7) in Theorem 2.3. From Remark 2.4, we have $y_n(i) = \frac{1}{n}\sum_{k=1}^n NY_k(i_k)\mathbb{1}\{i_k = i\}$ for any $n \geq 1$ and $i \in [N]$. Let $i \in \mathcal{A}^c$, i.e., suppose agent i is honest. Then, $(NY_k(i_k)\mathbb{1}\{i_k = i\})$ is a sequence of IID random variables with $\mathbb{E}NY_k(i_k)\mathbb{1}\{i_k = i\} = \mathbb{E}Y(i)$.

Hence, $\mathbb{E}|y_n(i) - \mathbb{E}Y(i)|^2 = n^{-2}\sum_{k=1}^n \mathbb{E}Z_k^2$, where $Z_k := NY_k(i_k)\mathbb{1}\{i_k = i\} - \mathbb{E}Y(i)$. Now, for any $k \geq 1$,

$$\begin{aligned}
\mathbb{E}Z_k^2 &\overset{(a)}{=} N\mathbb{E}[Y(i) - \mathbb{E}Y(i)]^2 + (N-1)[\mathbb{E}Y(i)]^2 \\
&\overset{(b)}{=} N\mathbb{E}[a_i^\top (X - \mathbb{E}X)]^2 + (N-1)[a_i^\top \mathbb{E}X]^2 \\
&\overset{(c)}{\leq} Nd \max_i \|a_i\|^2 \max_j \big(\mathrm{Var}(X(j)) + [\mathbb{E}X(j)]^2\big) \\
&\leq Nd\bar{A}^2(\bar{\sigma}^2 + \bar{\mu}^2),
\end{aligned}$$

where (a) holds because, on the event $\{i_k = i\}$, $Y_k(i_k) \sim Y(i)$ is generated with independent randomness, (b) holds since $Y(i) = a_i^\top X$, while (c) follows from the Cauchy-schwartz inequality.

The desired result now follows. □

We now discuss our approach to derive (x_n)'s convergence rate. We begin by rewriting Step 3 of Algorithm 1 as

$$x_{n+1} = \Pi_{\mathcal{X}}\left(x_n + \alpha_n[g_n' + \epsilon_n + M_{n+1}]\right), \tag{9}$$

where, for $n \geq 0$,

$$g_n' := \frac{1}{N}\left[\sum_{i\in\mathcal{A}^c} \mathrm{sign}(\mathbb{E}Y(i) - a_i^\top x_n)\ a_i + \sum_{i\in\mathcal{A}} \mathrm{sign}(y_n(i) - a_i^\top x_n)\ a_i\right] \tag{10}$$

$$\epsilon_n := \frac{1}{N}\sum_{i\in\mathcal{A}^c} \left[\mathrm{sign}(y_n(i) - a_i^\top x_n) - \mathrm{sign}(\mathbb{E}Y(i) - a_i^\top x_n)\right]\ a_i \tag{11}$$

and

$$M_{n+1} := a_{i_{n+1}}\ \mathrm{sign}(y_n(i_{n+1}) - a_{i_{n+1}}^\top x_n) - g_n' - \epsilon_n. \tag{12}$$

Separately, let

$$g_n := \frac{1}{N}\sum_{i=1}^N \mathrm{sign}(\mathbb{E}Y(i) - a_i^\top x_n) a_i. \tag{13}$$

Then, an intuitive description of g_n, g_n', ϵ_n, and M_{n+1} is as follows. First, $-g_n$ is a true sub-gradient of f at x_n, while $-g_n'$ is its corrupted cousin, where the corruption is due to the dependence on the $y_n(i)$ estimates given by the adversaries. Next, ϵ_n is the error in estimating the non-corrupted part of g_n' that appears due to our lack of knowledge of $\mathbb{E}Y$. Put differently, $g_n' + \epsilon_n$ is a corrupted approximation of the true sub-gradient of f at x_n. Finally, M_{n+1} is the noise in the estimate of $g_n' + \epsilon_n$ which arises since only one randomly picked coordinate of y_n is used to update x_n. Specifically, (M_n) is a martingale-difference sequence with respect to the filtration $(\mathcal{F}_n)$, where

$$\mathcal{F}_n := \sigma(x_0, y_0, i_1, x_1, y_1, \ldots, i_n, x_n, y_n). \tag{14}$$

Suppose the adversaries were not there and $y_n \equiv \mathbb{E}Y$ so that $-Ng_n' = -Ng_n$, $\epsilon_n = 0$, and $M_{n+1} = a_{i_{n+1}}\mathrm{sign}(\mathbb{E}Y(i_{n+1}) - a_{i_{n+1}}^\top x_n) - g_n$. Then, (9) would be the hypothetical algorithm

$$x_{n+1} = \Pi_{\mathcal{X}}\big(x_n + \alpha_n[g_n + M_{n+1}]\big), \tag{15}$$

which can be viewed as stochastic subgradient descent method for minimizing the non-strongly convex function f in (1). In which case, the analysis in [Nemirovski et al., 2009, Section 2.2] would have directly applied and given us the convergence rate. The challenge in our analysis stems from the corruption $g_n - g_n'$ and the approximation error ϵ_n in our sub-gradient estimate that is introduced by the adversaries and our lack of knowledge of $\mathbb{E}Y$, respectively. In particular, the adversaries could make g_n' an extremely poor estimate of g_n, while the discontinuity of the sign function implies showing $\epsilon_n \to 0$ is not trivial even though $y_n(i) \to \mathbb{E}Y(i)$ for any $i \in \mathcal{A}^c$.

We now recall the key ideas of [Nemirovski et al., 2009, Section 2.2] and briefly discuss their role in deriving the convergence rate of the *hypothetical* (x_n) sequence obtained from (15). Let $E_n := \frac{1}{2}\mathbb{E}\|x_n - \mathbb{E}X\|^2$. Since $-g_n$ is a subgradient of the convex function f, we have

$$\mathbb{E}[(x_n - \mathbb{E}X)^\top(-g_n)] \geq \mathbb{E}\big[f(x_n) - f(\mathbb{E}X)\big]. \tag{16}$$

Separately, since $\|g_n + M_{n+1}\| \leq \bar{A}$, $\mathbb{E}[M_{n+1}|\mathcal{F}_n] = 0$, and $\|\Pi_{\mathcal{X}}(x) - \Pi_{\mathcal{X}}(y)\| \leq \|x - y\|$, we get

$$E_{n+1} \leq E_n + \alpha_n\mathbb{E}[(x_n - \mathbb{E}X)^\top g_n] + \frac{1}{2}\alpha_n^2\bar{A}^2. \tag{17}$$

A combination of (16) and (17) then gives

$$\alpha_n\mathbb{E}[f(x_n) - f(\mathbb{E}X)] \leq E_n - E_{n+1} + \frac{1}{2}\alpha_n^2\bar{A}^2. \tag{18}$$

By exploiting the telescopic nature of $E_n - E_{n+1}$ above, it then follows that

$$\sum_{j=i}^{n}\mathbb{E}[\tilde{\alpha}_j f(x_j) - f(\mathbb{E}X)] \leq \frac{E_i - E_{n+1} + \frac{1}{2}\bar{A}^2\ \sum_{j=i}^n\alpha_j^2}{\sum_{j=i}^n\alpha_j} \leq \frac{E_i + \frac{1}{2}\bar{A}^2\ \sum_{j=i}^n\alpha_j^2}{\sum_{j=i}^n\alpha_j}, \tag{19}$$

where $\tilde{\alpha}_j$ is as defined in (3). Finally, by the convexity of f, we have $f(\tilde{x}_i^n) \leq \sum_{j=i}^n\mathbb{E}\tilde{\alpha}_j f(x_j)$, which shows that

$$\mathbb{E}\big[f(\tilde{x}_i^n) - f(\mathbb{E}X)\big] \leq \frac{E_i + \frac{1}{2}\bar{A}^2\ \sum_{j=i}^n\alpha_j^2}{\sum_{j=i}^n\alpha_j}. \tag{20}$$

The use of the projection operator $\Pi_{\mathcal{X}}$, along with the fact $x_0 \in \mathcal{X}$, ensures that $x_n \in \mathcal{X}$ for all $n \geq 0$, which implies $E_i \leq 2D_{\mathcal{X}}^2$ for any $i \geq 0$. By choosing i and α_n as in Theorem 2.3, it is easy to see that a similar convergence rate, as in (6), holds for our hypothetical (x_n) sequence.

Now, let us go back to (9). Recall that its key difference to (15) is that the hypothetical version directly uses a noisy estimate of g_n, while (9) relies on a noisy estimate of $g_n' + \epsilon_n$. Importantly, the latter quantity is only an approximate version of g_n, since $\mathbb{E}Y$ is unknown. Moreover, this approximation is corrupted, since g_n' also depends on the $y_n(i)$ estimates obtained from the adversaries. The main novelty in our proof is in showing that an inequality similar to (16) also holds in the context of $\mathbb{E}[(x_n - \mathbb{E}X)^\top(g_n' + \epsilon_n)]$. This result is summarized below.

Lemma 3.1. *Let g_n' and ϵ_n be defined as in* (10) *and* (11)*. Further, suppose the sequence (x_n) is generated using Algorithm 1 or, equivalently,* (9)*. Then, for K and C_y as in Theorem 2.3, we have*

$$\mathbb{E}(x_n - \mathbb{E}X)^\top g_n' \leq \frac{1}{K}\mathbb{E}(x_n - \mathbb{E}X)^\top g_n, \tag{21}$$

and

$$\mathbb{E}(x_n - \mathbb{E}X)^\top\epsilon_n \leq \frac{2(N-m)}{\sqrt{N}}\frac{C_y}{\sqrt{n}}. \tag{22}$$

The proof of this result is given in Appendix A.

Remark 3.2. *While g_n' could be corrupted by advesaries, we show that the redundancy and robustness induced by our observation matrix A (see $\mathcal{A}_2$) ensures* (21) *holds, which leads to a similar inequality like* (16)*, but with an additional $1/K$ factor. Since $m \leq N/2$, we get $K \leq \frac{\bar{A}}{\eta} + 1$, a constant, which implies $1/K$ cannot be too small. Similarly, while the sign function is discontinuous, we use the crucial inequality*

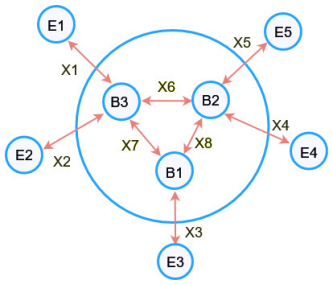
$$|\text{sign}(y_n(i) - a_i^\top x_n) - \text{sign}(\mathbb{E}Y(i) - a_i^\top x_n)| \leq \mathbb{1}\{|y_n(i) - \mathbb{E}Y(i)| \geq |a_i^\top x_n - \mathbb{E}Y(i)|\}$$

to show that

$$(x_n - \mathbb{E}X)^\top\epsilon_n \leq \frac{2}{N}\sum_{i\in\mathcal{A}^c}|y_n(i) - \mathbb{E}Y(i)|.$$

Our result about (y_n)'s convergence rate then leads to (22)*, as desired.*

Since (21) and (22) put together give the analogue of (16), the convergence rate in Theorem 2.3 now follows by repeating the arguments given in (17) (with $g_n' + \epsilon_n$ replacing g_n) to (20).



$$P := \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad A := \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 2 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 1 \end{bmatrix}$$

Figure 1: Edge nodes $E_i,\ i = 1, ..., 5$ has different link delay but has the same mean, $B_i,\ i = 1, 2, 3$ are backbone nodes.

4 Numerical illustrations

In this section of the paper, we show how to apply our algorithm to link delay tomography. The goal of link delay tomography is to estimate link delay statistics from path delay measurements (end-to-end measurement).

Consider a network $G = (\mathcal{V}, \mathcal{L})$ with nodes $\mathcal{V}$ and links $\mathcal{L}$. Each link $l \in \mathcal{L}$ is associated with a performance metric (e.g., delay, loss) drawn at each instant from a distribution with unknown mean $\mathbb{E}[x(l)]$. Let $\mathcal{P}$ be the set of probing paths. The network manager can inject probes on all paths in $\mathcal{P}$ and observes their end-to-end performance. More precisely, we assume that if at instant n, a probe on path $p \in \mathcal{P}$ is received, then the network manager observes the path delay measurement:

$$y_n(p) = \sum_{l \in p} x_n(l). \tag{23}$$

Let us define a $|\mathcal{P}| \times |\mathcal{L}|$ matrix P, where the pl-entry $P_{pl} = 1$ if the link l is on path p otherwise is equal to 0. At every time step n, the measurements can be summarized using the following vectorial form:

$$y_n = Px_n. \tag{24}$$

It is unreasonable to expect that all probe packets injected at time n reach their destination at the same time, or for the central server to wait until it receives measurements from all the injected probe packets before it updates its estimate. This motivates the assumption of sporadic or intermittent data in Section 2.

In standard network tomography, the true measurement y_n, given in (24), is returned to the central server. Here, we consider the possibility that one of the monitors can behave maliciously by returning any arbitrary real number instead. To mimic the worst case, let us consider the case where the adversary $i \in |\mathcal{P}|$, whose identity is unknown to the central server, always returns $r \in \mathbb{R}$ such that $\text{sign}(r - a_i^T x_n)$ is the opposite of $\text{sign}(y_n(i) - a_i^T x_n)$.

For our simulations, we consider the path matrix P and the network in fig. 1. For each $l \in \mathcal{L}$ and $n \geq 1$, we sample $\mu \sim exp(1)$ and $\epsilon \sim \mathcal{N}(0, 0.01)$ and set $x_n(l) = \mu + \epsilon$. We generate y_n according to (24) with the IID samples of x_n. Usually, the number of paths is much lower than the number of links (i.e. P is a wide matrix), so (24) is an under-determined system and directly estimating $\mathbb{E}[x]$ is impossible. Instead, we make the assumption that the delays on edge links (links $X1, \ldots, X5$ in fig. 1), which are links between edge nodes and backbone nodes, have the same mean. This assumption is reasonable when we assume that links can be of different classes with each class having a similar average delay. For instance, such an assumption has been taken in [Kinsho et al., 2019, 2017], where they assume that delays at neighboring base stations are comparable.

By taking A as in fig. 1, Y to be the expected value of y and X to be the vector whose first entry is the expected delay on the edge links and the remaining entries are the expected

delays on the non-edge links, it is easy to see that:

$$Y = AX.$$

The matrix A satisfies assumption $\mathcal{A}_2$ and the above system of equations is solvable. A proof of the robustness of A is given in Appendix B.

In our first experiment, we run Algorithm 1 with the samples $\{y_n(p)\}_{n\geq 1, p\in[|\mathcal{P}|]}$ generated using (24), $\alpha_n = (1 + \lfloor \frac{n}{100} \rfloor)^{-1/2}$ and $\beta_n = (1 + \lfloor \frac{n}{100} \rfloor)^{-1}$. Let z_n be the estimates from Algorithm 1 and $\tilde{z}_i^n$ be their weighted average with $i = \lfloor \frac{n}{2} \rfloor$ as in (4).

In Fig. 2, we have displayed the evolution of 10 different runs of Algorithm 1 along with their average $\|A(X - \tilde{z}_i^n)\|_1$ error with the confidence interval (5-95%). We observe that most runs begin converging towards 0 after approximately 2000 iterations. From 2000 iterations onward, the convergence continues, with the error reducing and approaching 0 as the number of iterations increases up to 5000. This verifies our claim of convergence of Algorithm 1 in Theorem 2.3.

For the second experiment, we study the impact of the number of agents in the convergence of Algorithm 1. We vary the number of paths, $N := |\mathcal{P}|$, by creating new matrices as follows: We start with the initial matrix A and generate 20 new matrices by adding 10 rows to A in each iteration. More precisely, for each matrix, each new row is added by randomly selecting a row from the previous matrix. This procedure ensures that all the matrices satisfy the robustness condition in Assumption $\mathcal{A}_2$.

For each of the 20 new matrices, we select n, the number of iterations, randomly between 2000 and 5000 and we perform 10 runs of Algorithm 1, similar to the first experiment. In fig. 3, for each value of N (which corresponds to one of the matrices), we plot $\sqrt{n}$ times the average $\|A(X - \tilde{z}_i^n)\|_1$. We use non-linear regression to fit the equation $a\sqrt{N} + b$ to the points obtained and plot the fitted curve. We obtain $a = 269.98$ and $b = -1065.23$ The closeness of the fitted curve to our data in fig. 3 empirically validates that the error $\sqrt{n}\|A(X - \tilde{z}_i^n)\|_1$ grows at a $\sqrt{N}$ rate.

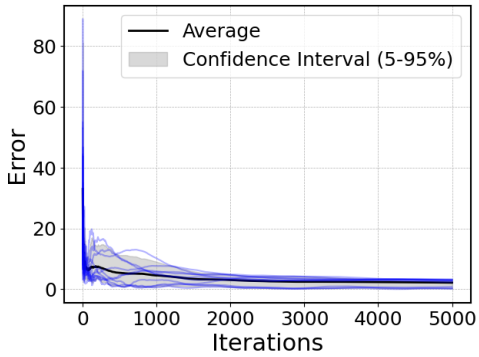


Figure 2: Evolution of $\|A(X - \tilde{z}_i^n)\|_1$ over 5000 iterations for 10 different runs of Algorithm 1 and their average.

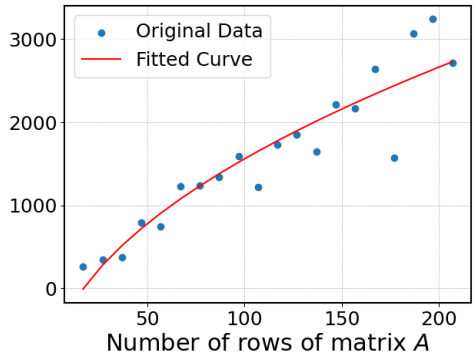


Figure 3: Nonlinear regression of $\sqrt{n} \times \mathbb{E}[\|A(X - \tilde{z}_i^n)\|_1]$ with respect to the number of lines in matrices A.

5 Conclusions and future directions

We establish the convergence rate of a two-timescale algorithm for online distributed learning in the presence of adversaries, and sporadic and heterogeneous measurements. We also show that for a fixed number of adversaries, the degradation in the convergence rate is of the order $O(\sqrt{N})$, where N is the number of worker nodes. Finally, we demonstrate the utility of our algorithm in the domain of network tomography.

There are two main limitations of this work. In this paper, we design a robust observation matrix using brute-force methods. Developing a technique that can generate observation

matrices for an arbitrary number of worker nodes is an unsolved challenge. The main results of our paper currently only apply to the distributed estimation setting. In the future, we hope to extend the techniques developed in the paper to solve related problems in distributed machine learning.

References

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.

Djamila Bouhata, Hamouma Moumen, Jocelyn Ahmed Mazari, and Ahcène Bounceur. Byzantine fault tolerance in distributed machine learning: a survey. *arXiv preprint arXiv:2205.02572*, 2022.

Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: Recent developments. 2004.

Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. In *International Conference on Machine Learning*, pages 903–912. PMLR, 2018.

Cho-Chun Chiu and Ting He. Stealthy dgos attack: Degrading of service under the watch of network tomography. *IEEE/ACM Transactions on Networking*, 29(3):1294–1307, 2021.

Georgios Damaskinos, Rachid Guerraoui, Rhicheek Patra, Mahsa Taziki, et al. Asynchronous byzantine machine learning (the case of sgd). In *International Conference on Machine Learning*, pages 1145–1154. PMLR, 2018.

Jianqing Fan, Dong Wang, Kaizheng Wang, and Ziwei Zhu. Distributed estimation of principal eigenspaces. *Annals of statistics*, 47(6):3009, 2019.

Minghong Fang, Jia Liu, Neil Zhenqiang Gong, and Elizabeth S Bentley. Aflguard: Byzantine-robust asynchronous federated learning. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 632–646, 2022.

Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic control*, 59 (6):1454–1467, 2014.

Swetha Ganesh, Alexandre Reiffers-Masson, and Gugan Thoppe. Online learning with adversaries: A differential-inclusion analysis. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 1288–1293. IEEE, 2023.

Hideaki Kinsho, Rie Tagyo, Daisuke Ikegami, Takahiro Matsuda, Akira Takahashi, and Tetsuya Takine. Heterogeneous delay tomography based on graph fourier transform in mobile networks. In *2017 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, pages 1–6. IEEE, 2017.

Hideaki Kinsho, Rie Tagyo, Daisuke Ikegami, Takahiro Matsuda, Jun Okamoto, and Tetsuya Takine. Heterogeneous delay tomography for wide-area mobile networks. *IEICE Transactions on Communications*, 102(8):1607–1616, 2019.

Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on operating systems design and implementation (OSDI 14)*, pages 583–598, 2014.

Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Srikar Tati, Simone Silvestri, Ting He, and Thomas La Porta. Robust network tomography in the presence of failures. In *2014 IEEE 34th International Conference on Distributed Computing Systems*, pages 481–492. IEEE, 2014.

Yolanda Tsang, Mark Coates, and Robert D Nowak. Network delay tomography. *IEEE Transactions on Signal Processing*, 51(8):2125–2136, 2003.

Yehuda Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American statistical association*, 91(433):365–377, 1996.

Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *Acm computing surveys (csur)*, 53(2):1–33, 2020.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno++: Robust fully asynchronous sgd. In *International Conference on Machine Learning*, pages 10495–10503. PMLR, 2020.

Yi-Rui Yang and Wu-Jun Li. Basgd: Buffered asynchronous sgd for byzantine learning. In *International Conference on Machine Learning*, pages 11751–11761. PMLR, 2021.

Zhixiong Yang, Arpita Gang, and Waheed U Bajwa. Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the byzantine threat model. *IEEE Signal Processing Magazine*, 37(3):146–159, 2020.

Hongyi Yao, Sidharth Jaggi, and Minghua Chen. Passive network tomography for erroneous networks: A network coding approach. *IEEE Transactions on Information Theory*, 58(9): 5922–5940, 2012.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. Pmlr, 2018.

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

Shangqing Zhao, Zhuo Lu, and Cliff Wang. When seeing isn't believing: On feasibility and detectability of scapegoating in network tomography. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 172–182. IEEE, 2017.

A Detailed proof of (x_n)'s convergence rate

As discussed in Section 3, the key ingredient in the derivation of (x_n)'s convergence rate is Lemma 3.1. We begin with its proof. First, we need the following result.

Lemma A.1. *Let K be as defined above Theorem 2.3. Then,*

$$\sum_{i\in S^c} |a_i^\top x| > \frac{K+1}{K-1} \sum_{i\in S} |a_i^\top x|$$

for every $x \neq 0$ and every $S \subseteq [N]$ such that $|S| = m$.

Proof. For any x such that $\sum_{i\in S} |a_i^\top x| = 0$, the result trivially holds from our assumption on A. Hence, suppose that $\sum_{i\in S} |a_i^\top x| \neq 0$.

From the definition of η given in (5), we have

$$\sum_{i\in S^c} |a_i^\top x| - \sum_{i\in S} |a_i^\top x| \geq N\eta\|x\|.$$

Therefore,

$$\frac{\sum_{i\in S^c} |a_i^\top x|}{\sum_{i\in S} |a_i^\top x|} \geq 1 + \frac{N\eta\|x\|}{\sum_{i\in S} |a_i^\top x|}$$

Now, from the Cauchy-Schwarz inequality, we have

$$\max_{x\neq 0} \frac{\sum_{i\in S} |a_i^\top x|}{\|x\|} \leq \sum_{i\in S} \|a_i\| \leq m\bar{A}$$

The desired result is now easy to see. □

We now derive Lemma 3.1.

Proof of (21) *in Lemma 3.1.* From Lemma A.1, we have

$$(K-1) \sum_{i\in\mathcal{A}^c} |(x_n - \mathbb{E}X)^\top a_i| \geq (K+1) \sum_{i\in\mathcal{A}} |(x_n - \mathbb{E}X)^\top a_i|.$$

Since $|z| \geq -z\text{sign}(r)$ for any real numbers z and r, the above relation implies

$$(K-1) \sum_{i\in\mathcal{A}^c} |(x_n - \mathbb{E}X)^\top a_i| \geq \sum_{i\in\mathcal{A}} \Big(|(x_n - \mathbb{E}X)^\top a_i| - K\big[(x_n - \mathbb{E}X)^\top a_i\big] \text{ sign}(y_n(i) - a_i^\top x_n) \Big).$$

Now, since $a_i^\top \mathbb{E}X = \mathbb{E}Y(i)$ and $z\text{sign}(z) = |z|$, the above relation implies

$$\begin{aligned}(K-1) \sum_{i\in\mathcal{A}^c} (x_n - \mathbb{E}X)^\top a_i \text{ sign}(\mathbb{E}Y(i) - a_i^\top x_n) \\ \leq \sum_{i\in\mathcal{A}} (x_n - \mathbb{E}X)^\top a_i \ \big[\text{sign}(\mathbb{E}Y(i) - a_i^\top x_n) - K\text{sign}(y_n(i) - a_i^\top x_n)\big].\end{aligned}$$

The desired relation in (21) now follows. □

Proof of (22) *in Lemma 3.1.* First, for any three real numbers r, r_1, and r_2, we have

$$\text{sign}(r_1 - r) - \text{sign}(r_2 - r) \leq 2 \times \mathbb{1}\{|r_1 - r_2| \geq |r - r_2|\}. \tag{25}$$

This inequality can be verified by considering all possible ordering of r, r_1, and r_2. Hence,

$$\begin{aligned}(x_n - \mathbb{E}X)^\top \epsilon_n &\overset{(a)}{=} \frac{1}{N}\sum_{i\in\mathcal{A}^c}(x_n - \mathbb{E}X)^\top a_i \ \left[\text{sign}(y_n(i) - a_i^\top x_n) - \text{sign}(\mathbb{E}Y(i) - a_i^\top x_n)\right] \\ &\overset{(b)}{\leq} \frac{2}{N}\sum_{i\in\mathcal{A}^c}(x_n - \mathbb{E}X)^\top a_i \ \mathbb{1}\{|y_n(i) - \mathbb{E}Y(i)| \geq |a_i^\top x_n - \mathbb{E}Y(i)|\} \\ &\overset{(c)}{\leq} \frac{2}{N}\sum_{i\in\mathcal{A}^c}|y_n(i) - \mathbb{E}Y(i)| \ \mathbb{1}\{|y_n(i) - \mathbb{E}Y(i)| \geq |a_i^\top x_n - \mathbb{E}Y(i)|\} \\ &\leq \frac{2}{N}\sum_{i\in\mathcal{A}^c}|y_n(i) - \mathbb{E}Y(i)|,\end{aligned}$$

where (a) holds from the definition of ϵ_n in (11), (b) is due to (25), while (c) follows since

$$|(x_n - \mathbb{E}X)^\top a_i| = |a_i^\top x_n - \mathbb{E}Y(i)| \leq |y_n(i) - \mathbb{E}Y(i)|$$

when $\mathbb{1}\{|y_n(i) - \mathbb{E}Y(i)| \geq |a_i^\top x_n - \mathbb{E}Y(i)|\} = 1$.
The claim in (22) now follows from (7). □

We now derive the convergence rate for the (x_n) sequence obtained from Algorithm 1.

Proof of (6) *in Theorem 2.3.* We have

$$\begin{aligned}\|x_{n+1} - \mathbb{E}X\|^2 &\overset{(a)}{=} \|\Pi_{\mathcal{X}}(x_n + \alpha_n(g_n' + \epsilon_n + M_{n+1})) - \Pi_{\mathcal{X}}(\mathbb{E}X)\|^2 \\ &\overset{(b)}{\leq} \|x_n + \alpha_n(g_n' + \epsilon_n + M_{n+1}) - \mathbb{E}X\|^2 \\ &= \|x_n - \mathbb{E}X\|_2^2 + 2\alpha_n(x_n - \mathbb{E}X)^\top(g_n' + \epsilon_n + M_{n+1}) + \alpha_n^2\|M_{n+1}\|^2 \\ &\overset{(c)}{\leq} \|x_n - \mathbb{E}X\|_2^2 + 2\alpha_n(x_n - \mathbb{E}X)^\top(g_n' + \epsilon_n + M_{n+1}) + \alpha_n^2\bar{A}^2,\end{aligned}$$

where (a) follows from (9) and fact that $\mathbb{E}X \in \mathcal{X}$ (which implies $\Pi_{\mathcal{X}}(\mathbb{E}X) = \mathbb{E}X$), (b) holds since $\|\Pi(x) - \Pi(y)\|_2 \leq \|x - y\|_2$ for any $x, y \in \mathbb{R}^d$, while (c) holds since $\|M_{n+1}\| \leq \bar{A}$.
Now, since $E_n = \frac{1}{2}\mathbb{E}\|x_n - \mathbb{E}X\|_2^2$, the inequality in (c) above implies

$$\begin{aligned}E_{n+1} &\leq E_n + \alpha_n\mathbb{E}[(x_n - \mathbb{E}X)^\top(g_n' + \epsilon_n + M_{n+1})] + \frac{1}{2}\alpha_n^2\bar{A} \\ &= E_n + \alpha_n\mathbb{E}[(x_n - \mathbb{E}X)^\top(g_n' + \epsilon_n)] + \frac{1}{2}\alpha_n^2\bar{A},\end{aligned}$$

where the last relation holds since $\mathbb{E}[M_{n+1}|\mathcal{F}_n] = 0$ with $\mathcal{F}_n$ being the σ-field defined in (14). By substituting (21) and (22) from Lemma 3.1 above, we then get

$$E_{n+1} \leq E_n + \frac{\alpha_n}{K}\mathbb{E}(x_n - \mathbb{E}X)^\top g_n + \alpha_n\frac{2(N-m)}{\sqrt{N}}\frac{C_y}{\sqrt{n}} + \frac{1}{2}\alpha_n^2\bar{A}^2.$$

Note that the above relation is similar in spirit to (17). By combining the above relation with (16), it then follows that

$$\mathbb{E}\alpha_n\left(f(x_n) - f(\mathbb{E}X)\right) \leq K(E_n - E_{n+1}) + \frac{2(N-m)KC_y}{\sqrt{N}}\frac{\alpha_n}{\sqrt{n}} + \frac{K}{2}\bar{A}^2\alpha_n^2.$$

Since the above relation is true for any n, we get

$$\begin{aligned}\mathbb{E}\sum_{k=i}^{n}\alpha_k(f(x_k) - f(\mathbb{E}X)) &\leq K(E_i - E_{n+1}) + \frac{2(N-m)KC_y}{\sqrt{N}}\sum_{k=i}^{n}\frac{\alpha_k}{\sqrt{k}} + \frac{K}{2}\bar{A}^2\sum_{k=i}^{n}\alpha_k^2 \\ &\leq KE_i + \frac{2(N-m)KC_y}{\sqrt{N}}\sum_{k=i}^{n}\frac{\alpha_k}{\sqrt{k}} + \frac{K}{2}\bar{A}^2\sum_{k=i}^{n}\alpha_k^2.\end{aligned}$$

Therefore,

$$\left[\sum_{j=i}^{n}\alpha_j\right]\left[\mathbb{E}\sum_{k=i}^{n}\tilde{\alpha}_k(f(x_k)-f(\mathbb{E}X))\right]\leq KE_i+\frac{2(N-m)KC_y}{\sqrt{N}}\sum_{k=i}^{n}\frac{\alpha_k}{\sqrt{k}}+\frac{K}{2}\bar{A}^2\sum_{k=i}^{n}\alpha_k^2,$$

which is the relation that is similar in spirit to (19).

Now, $E_i\leq 2D_{\mathcal{X}}^2$ and, by the convexity of f, we have $f(\tilde{x}_i^n)\leq\sum_{j=i}^{n}\mathbb{E}\tilde{\alpha}_j f(x_j)$. Therefore,

$$\begin{aligned}\left[\sum_{j=i}^{n}\alpha_j\right]\left[\mathbb{E}f(\tilde{x}_i^n)-f(\mathbb{E}X)\right]&\leq 2KD_{\mathcal{X}}^2+\frac{2(N-m)KC_y}{\sqrt{N}}\sum_{k=i}^{n}\frac{\alpha_k}{\sqrt{k}}+\frac{K}{2}\bar{A}^2\sum_{k=i}^{n}\alpha_k^2\\&\leq\left(\frac{N-m}{\sqrt{N}}\right)\left[2KD_{\mathcal{X}}^2+2KC_y\sum_{k=i}^{n}\frac{\alpha_k}{\sqrt{k}}+\frac{K}{2}\bar{A}^2\sum_{k=i}^{n}\alpha_k^2\right],\end{aligned}$$

where in the last relation we made use of the fact that $(N-m)\geq\sqrt{N}$ for any $m\leq N/2$ and $N\geq 4$. Next, since $\alpha_k=1/\sqrt{k+1}$, we have that, for $1\leq i\leq n$,

$$\max\left\{\frac{\alpha_k}{\sqrt{k}},\sum_{k=i}^{n}\alpha_k^2\right\}\leq\sum_{k=i}^{n}\frac{1}{k}\leq\ln\left(\frac{n}{i-1}\right).$$

Similarly,

$$\sum_{j=i}^{n}\alpha_j=\sum_{j=i}^{n}\frac{1}{\sqrt{j+1}}\geq 2\big[\sqrt{n+2}-\sqrt{i+1}\big].$$

By combining the last three relations, it now follows that

$$\mathbb{E}f(\tilde{x}_i^n)-f(\mathbb{E}X)\leq\left(\frac{N-m}{\sqrt{N}}\right)\frac{\left[2KD_{\mathcal{X}}^2+K(2C_y+\bar{A}^2/2)\ln\left(\frac{n}{i-1}\right)\right]}{2(\sqrt{n+2}-\sqrt{i+1})}.\tag{26}$$

Finally, by using the fact that $i=\lceil rn\rceil+1$, we have $rn+1\leq i\leq rn+2$, which implies

$$\ln\left(\frac{n}{i-1}\right)\leq-\ln r.$$

Furtheremore, for $n\geq 2$, we have that

$$\begin{aligned}2(\sqrt{n+2}-\sqrt{i+1})&\geq 2\sqrt{n+2}\left(1-\sqrt{\frac{rn+3}{n+2}}\right)\\&\geq 2\sqrt{n}\left(1-\sqrt{\frac{rn+3}{n+2}}\right)\\&\geq 2\sqrt{n}\left(1-\sqrt{r}\sqrt{1+\frac{3/r-2}{n+2}}\right)\\&\geq 2\sqrt{n}\left(1-\sqrt{r}\sqrt{1+\frac{3/r-2}{4}}\right)\\&=2\sqrt{n}\left(1-\sqrt{\frac{r}{2}+\frac{3}{4}}\right)\\&\geq\sqrt{n}\left(2-\sqrt{2r+3}\right).\end{aligned}$$

By substituting the above inequalities in (26), the desired result follows. □

B Robustness of the A matrix in fig. 1

.

Proposition B.1. *The matrix A is robust.*

Proof. We verify the robustness of the matrix A by checking condition 2 in the presence of one adversary. The adversary can attack any path/row of the matrix A. We will verify the robustness condition for one of the paths; the others can be verified in a similar way.

Case 1: Let us assume the adversary is on the second path. To verify A is robust we must show

$$|2x_1 + x_2| < |2x_1 + x_3| + |2x_1 + x_4| + |2x_1 + x_2 + x_3| + |2x_1 + x_3 + x_4| + |2x_1 + x_2 + x_4| + |2x_1 + x_2 + x_3 + x_4|. \tag{27}$$

Using the triangle inequality on the left hand side of (27), we get

$$|2x_1 + x_2| \leq |2x_1 + x_2 + x_3| + |x_3 + 2x_1 + x_4| + |2x_1 + x_4|. \tag{28}$$

(28) can also be rewritten as

$$|2x_1 + x_2| \leq |2x_1 + x_2 + x_3| + |x_3 + 2x_1 + x_4| + |2x_1 + x_4| + |2x_1 + x_3| + |2x_1 + x_2 + x_4| + |2x_1 + x_2 + x_3 + x_4|. \tag{29}$$

If equality occurs in (29), we have equality in (28) as well. Then equality in (28) and (29) give us the following equations,

$$2x_1 + x_3 = 0, 2x_1 + x_2 + x_4 = 0, 2x_1 + x_2 + x_3 + x_4 = 0. \tag{30}$$

(30) gives $x_3 = x_1 = 0$. Using these values and equality in (28) we get the simplified form,

$$|x_2| = |x_2| + |x_4| + |x_4|. \tag{31}$$

Hence from (31), $|x_4| = 0, i.e, x_4 = 0$, which implies $x_2 = 0$, from (30). Thus equality in (29) has the zero vector as the only solution. But, in $\mathcal{A}_2$, we assumed that $x = (x_1, x_2, x_3, x_4)$ is a non-zero vector, hence the strict inequality in (29) must hold.

In a similar way, $\mathcal{A}_2$ can be verified for the other paths. This shows that our matrix A is robust. □

C Compute Details

Our simulations were run on an Intel(R) Xeon(R) CPU and took approximately 3 hours to complete.