Robust Aggregation for Federated Learning

Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui

***Abstract*—We present a novel approach to federated learning that endows its aggregation process with greater robustness to potential poisoning of local data or model parameters of participating devices. The proposed approach, Robust Federated Aggregation (RFA), relies on the aggregation of updates using the geometric median, which can be computed efficiently using a Weiszfeld-type algorithm. RFA is agnostic to the level of corruption and aggregates model updates without revealing each device's individual contribution. We establish the convergence of the robust federated learning algorithm for the stochastic learning of additive models with least squares. We also offer two variants of RFA: a faster one with one-step robust aggregation, and another one with on-device personalization. We present experimental results with additive models and deep networks for three tasks in computer vision and natural language processing. The experiments show that RFA is competitive with the classical aggregation when the level of corruption is low, while demonstrating greater robustness under high corruption.**

***Index Terms*—Federated learning, robust aggregation, corrupted updates, distributed learning, data privacy.**

I. Introduction

FEDERATED learning is a key paradigm for machine learning and analytics on mobile, wearable and edge devices [1], [2] over wireless networks of 5G and beyond as well as edge networks and the Internet of Things. The paradigm has found widespread applications ranging from mobile apps deployed on millions of devices [3], [4], to sensitive healthcare applications [5], [6].

In federated learning, a number of devices with privacy-sensitive data collaboratively optimize a machine learning model under the orchestration of a central server, while keeping the data fully decentralized and private. Recent work has looked beyond supervised learning to domains such as data analytics but also semi-, self- and un-supervised learning, transfer learning, meta learning, and reinforcement learning [2], [7]–[9].

We study a question relevant in all these areas: robustness to corrupted updates. Federated learning relies on aggregation of updates contributed by participating devices, where the aggregation is privacy-preserving. Sensitivity to corrupted updates, caused either by adversaries intending to attack the system or due to failures in low-cost hardware, is a vulnerability of the usual approach. The standard arithmetic mean aggregation in federated learning is not robust to corruptions, in the sense that even a single corrupted update in a round is sufficient to degrade the global model for all devices. In one dimension, the median is an attractive aggregate for its robustness to outliers. We adopt this approach to federated learning by considering a classical multidimensional generalization of the median, known variously as the geometric or spatial or L_1 median [10].

Our robust approach preserves the privacy of the device updates by iteratively invoking the secure multi-party computation primitives used in typical non-robust federated learning [11], [12]. A device's updates are information theoretically protected in that they are computationally indistinguishable from random noise and the sensitivity of the final aggregate to the contribution of each device is bounded. Our approach is scalable, since the underlying secure aggregation algorithms are implemented in production systems across millions of mobile users across the planet [13]. The approach is communication-efficient, requiring a modest 1-3× the communication cost of the non-robust setting to compute the non-linear aggregate in a privacy-preserving manner.

Contributions: The main take-away message of this work is:

> *Federated learning can be made robust to corrupted updates by replacing the weighted arithmetic mean aggregation with an approximate geometric median at 1-3 times the communication cost.*

To this end, we make the following concrete contributions.

a) *Robust Aggregation:* We design a novel robust aggregation oracle based on the classical geometric median. We analyze the convergence of the resulting federated learning algorithm, RFA, for least-squares estimation and show that the proposed method is robust to update corruption in up to half the devices in federated learning with bounded heterogeneity. We also describe an extension of the framework to handle arbitrary heterogeneity via personalization.

b) *Algorithmic Implementation:* We show how to implement this robust aggregation oracle in a practical and privacy-preserving manner. This relies on an alternating minimization algorithm which empirically exhibits rapid convergence. This algorithm can be interpreted as a numerically stable version of the classical algorithm of Weiszfeld [14], thus shedding new light on it.

c) *Numerical Simulations:* We demonstrate the effectiveness of our framework for data corruption and parameter update corruption, on federated learning tasks from computer

Manuscript received July 24, 2021; revised December 20, 2021; accepted February 3, 2022. Date of publication February 24, 2022; date of current version March 11, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yue M. Lu. This work was supported in part by NSF under Grants CCF-1740551, CCF-1703574, and DMS-1839371, in part by the Washington Research Foundation for innovation in data-intensive discovery, in part by the CIFAR program Learning in Machines and Brains, faculty research awards, and in part by JP Morgan Ph.D. Fellowship. This work was first presented at the Workshop on Federated Learning and Analytics in June 2019. *(Corresponding author: Krishna Pillutla.)*

Krishna Pillutla and Zaid Harchaoui are with the University of Washington, Seattle, WA 98195 USA (e-mail: pillutla@cs.washington.edu; zaid@uw.edu).

Sham M. Kakade is with the Harvard University, Cambridge, MA 02138 USA (e-mail: sham@seas.harvard.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2022.3153135, provided by the authors.

Digital Object Identifier 10.1109/TSP.2022.3153135

vision and natural language processing, with linear models as well as convolutional and recurrent neural networks. In particular, our results show that the proposed RFA algorithm (i) outperforms the standard FedAvg [1], in high corruption and (ii) nearly matches the performance of the FedAvg in low corruption, both at *1-3 times the communication cost*. Moreover, the proposed algorithm is agnostic to the actual level of corruption in the problem instance.

We open source an implementation of the proposed approach in TensorFlow Federated[1]; cf. Appendix II for a template implementation. The Python code and scripts used to reproduce experimental results are publicly available online.[2]

Overview: Section II describes related work, and Section III describes the problem formulation and tradeoffs of robustness. Section IV proposes a robust aggregation oracle and presents a convergence analysis of the resulting robust federated learning algorithm. Finally, Section V gives comprehensive numerical simulations demonstrating the robustness of the proposed federated learning algorithm compared to standard baselines.

II. Related Work

We now survey some related work.

Federated Learning was introduced in [1] as a distributed optimization approach to handle on-device machine learning, with secure multi-party averaging algorithms given in [11], [17]. Extensions were proposed in [18]–[26]; see also the recent surveys [2], [27]. We address robustness to corrupted updates, which is broadly applicable in these settings.

Distributed optimization has a long history [28]. Recent work includes primal-dual frameworks [29], [30] and variants suited to decentralized [31], and asynchronous [32] settings.

From the lens of *learning in networks* [33], federated learning comprises a star network where agents (i.e., devices) with private data are connected to a server with no data, which orchestrates the cooperative learning. Further, for privacy, model updates from individual agents cannot be shared directly, but must be aggregated securely.

Robust estimation was pioneered by Huber [34], [35]. Robust median-of-means were introduced in [36], with follow ups in [37]–[41]. Robust mean estimation, in particular, received much attention [42]–[44]. Robust estimation in networks was considered in [45]–[47]. These works consider the statistics of robust estimation in the i.i.d. case, while we focus on distributed optimization with privacy preservation.

Byzantine robustness, resilience to arbitrary behavior of some devices [48], was studied in distributed optimization with gradient aggregation [49]–[54]. Byzantine robustness of federated learning is a priori not possible without additional assumptions because the secure multi-party computation protocols require faithful participation of the devices. Thus, we consider a more nuanced and less adversarial corruption model where devices participate faithfully in the aggregation loop; see Section III for practical examples. Further, it is unclear how to securely implement the nonlinear aggregation algorithms of these works. Lastly, the use of, e.g., secure enclaves [55] in conjunction with our approach could guarantee Byzantine robustness in federated learning. We aggregate *model parameters* in a robust manner, which is more suited to the federated setting. We note that [56] also aggregate model parameters rather than gradients by framing the problem in terms of consensus optimization. However, their algorithm requires devices to be always available and participate in multiple rounds, which is not practical in the federated setting [2].

Weiszfeld's algorithm [14] to *compute the geometric median*, has received much attention [57]–[59]. The Weiszfeld algorithm is also known to exhibit asymptotic linear convergence [60]. However, unlike these variants, ours is numerically stable. A theoretical proposal of a near-linear time algorithm for the geometric median was recently explored in [61].

Frameworks to guarantee *privacy* of user data include differential privacy [62], [63] and homomorphic encryption [64]. These directions are orthogonal to ours, and could be used in conjunction. See [2], [11], [27] for a broader discussion.

III. Problem Setup: Federated Learning With Corruptions

We begin this section by recalling the setup of federated learning (without corruption) and the standard FedAvg algorithm [1] in Section III-A. We then formally setup our corruption model and discuss the trade-offs introduced by requiring robustness to corrupted updates in Section III-B.

A. Federated Learning Setup and Review

Federated learning consists of n client devices which collaboratively train a machine learning model under the orchestration of a central server or a fusion center [1], [2]. The data is local to the client devices while the job of the server is to orchestrate the training.

We consider a typical federated learning setting where each device i has a distribution D_i over some data space such that the data on the client is sampled i.i.d. from D_i. Let the vector $w \in \mathbb{R}^d$ denote the parameters of a (supervised) learning model and let $f(w; z)$ denote the loss of model w on input-output pair z, such as the mean-squared-error loss. Then, the objective function of device i is $F_i(w) = \mathbb{E}_{z \sim D_i}[f(w; z)]$.

Federated learning aims to find a model $w^\star$ that minimizes the average objective across all the devices,

$$\min_{w \in \mathbb{R}^d} \left[F(w) := \sum_{i=1}^{n} \alpha_i \, F_i(w) \right] , \tag{1}$$

where device i is weighted by $\alpha_i > 0$. In practice, the weight α_i is chosen proportional to the amount of data on device i. For instance, in an empirical risk minimization setting, each D_i is the uniform distribution over a finite set $\{z_{i,1}, \dots, z_{i,N_i}\}$ of size N_i. It is common practice to choose $\alpha_i = N_i/N$ where $N = \sum_{i=1}^n N_i$ so that the objective $F(w) =$

[1] https://github.com/google-research/federated/tree/master/robust_aggregation

[2] https://github.com/krishnap25/rfa

$(1/N)\sum_{i=1}^{n}\sum_{j=1}^{N_i} f(w; z_{i,j})$ is simply the unweighted average over all samples from all n devices.

Federated Learning Algorithms: Typical federated learning algorithms run in synchronized rounds of communication between the server and the devices with some local computation on the devices based on their local data, and aggregation of these updates to update the server model. The de facto standard training algorithm is FedAvg [1], which runs as follows.

a) The server samples a set S_t of m clients from $[n]$ and broadcasts the current model $w^{(t)}$ to these clients.

b) Staring from $w_{i,0}^{(t)} = w^{(t)}$, each client $i \in S_t$ makes τ local gradient or stochastic gradient descent steps for $k = 0, \ldots, \tau - 1$ with a learning rate γ:

$$w_{i,k+1}^{(t)} = w_{i,k}^{(t)} - \gamma \nabla F_i(w_{i,k}^{(t)}) \,. \tag{2}$$

c) Each device $i \in S_t$ sends to the server a vector $w_i^{(t+1)}$ which is simply the final iterate, i.e., $w_i^{(t+1)} = w_{i,\tau}^{(t)}$. The server updates its global model using the weighted average

$$w^{(t+1)} = \frac{\sum_{i \in S_t} \alpha_i w_i^{(t+1)}}{\sum_{i \in S_t} \alpha_i} \,. \tag{3}$$

The federated learning algorithm, and in particular, the choice of aggregation, impacts the following three factors [2], [27], [65]: communication efficiency, privacy, and robustness.

Communication Efficiency: Besides the computation cost, the communication cost is an important parameter in distributed optimization. While communication is relatively fast in the datacenter, that is not the case of federated learning. The repeated exchange of massive models between the server and client devices over resource-limited wireless networks makes communication over the network more of a bottleneck in federated learning than local computation on the devices. Therefore, training algorithms should be able to trade-off more local computation for lower communication, similar to step (b) of FedAvg above. While the exact benefits (or lack thereof) of local steps is an active area of research, local steps have been found empirically to reduce the amount of communication required for a moderately accurate solution [1], [66].

Accordingly, we set aside the local computation cost for a first order approximation, and compare algorithms in terms of their total communication cost [2]. Since typical federated learning algorithms proceed in synchronized rounds of communication, we measure the complexity of the algorithms in terms of the number of communication rounds.

Privacy: While the privacy-sensitive data $z \sim D_i$ is kept local to the device, the model updates $w_i^{(t+1)}$ might also leak privacy. To add a further layer of privacy protection, the server is not allowed to inspect individual updates $w_i^{(t+1)}$ in the aggregation step (c); it can only access the aggregate $w^{(t+1)}$.

We make this precise through the notion of a *secure average oracle*. Given m devices with each device i containing $w_i \in \mathbb{R}^d$ and a scalar $\beta_i > 0$, a secure average oracle computes the average $\sum_{i=1}^{m} \beta_i w_i / \sum_{i=1}^{m} \beta_i$ at a total communication of $\mathcal{O}(md + m \log m)$ bits such that no w_i or β_i are revealed to either the server or any other device.

In practice, a secure average oracle is implemented using cryptographic protocols based on secure multi-party computation [11], [12]. These require a communication overhead of $O(m \log m)$ in addition to $O(md)$ cost of sending the m vectors. First, the vector $\beta_i w_i$ is dimension-wise discretized on the ring $\mathbb{Z}_M^d$ of integers modulo M in d-dimensions. Then, a noisy version $\tilde{w}_i$ is sent to the server, where the noise is designed to satisfy:

- correctness up to discretization, by ensuring $\sum_{i=1}^{m} \tilde{w}_i \mod M = \sum_{i=1}^{m} \beta_i w_i \mod M$ with probability 1, and,
- privacy preservation from honest-but-curious devices and server in the information theoretic sense, by ensuring that $\tilde{w}_i$ is computationally indistinguishable from $\zeta_i \sim \mathrm{Uniform}(\mathbb{Z}_M^d)$, irrespective of w_i and β_i.

As a result, we get the correct average (up to discretization) while not revealing any further information about a w_i or β_i to the server or other devices, beyond what can be inferred from the average. Hence, no further information about the underlying data distribution D_i is revealed either. In this work, we assume for simplicity that the secure average oracle returns the exact update, i.e., we ignore the effects of discretization on the integer ring and modular wraparound. This assumption is reasonable for a large enough value of M.

Robustness: We would like a federated learning algorithm to be robust to corrupted updates contributed by malicious devices or hardware/software failures. FedAvg uses an arithmetic mean to aggregate the device updates in (3), which is known to not be robust [34]. This can be made precise by the notion of a breakdown point [67], which is the smallest fraction of the points which need to be changed to cause the aggregate to take on arbitrary values. The breakdown point of the mean is 0, since only one point needs to changed to arbitrarily change the aggregate [10]. This means in federated learning that a single corrupted update, either due to an adversarial attack or a failure, can arbitrarily change the resulting aggregate in each round. We will give examples of adversarial corruptions in Section III-B.

In the rest of this work, we aim to address the lack of robustness of FedAvg. A popular robust aggregation of scalars is the median rather than the mean. We investigate a multidimensional analogue of the median, while respecting the other two factors: communication efficiency and privacy. While the non-robust mean aggregation can be computed with secure multi-party computation via the secure average oracle, it is unclear if a robust aggregate can also satisfy this requirement. We discuss this as well as other tradeoffs involving robustness in the next section.

B. Corruption Model and Trade-Offs of Robustness

We start with the corruption model used in this work. We allow a subset $\mathcal{C} \subset [n]$ of *corrupted devices* to, unbeknownst to the server, send arbitrary vectors $w_i^{(t+1)} \in \mathbb{R}^d$ rather than the updated model $w_{i,\tau}^{(t)}$ from local data as expected by the server. Formally, we have,

$$w_i^{(t+1)} = \begin{cases} w_{i,\tau}^{(t)}, & \text{if } i \notin \mathcal{C}, \\ H_i\left(w^{(t)}, \{(w_{j,\tau}^{(t)}, D_j)\}_{j \in S_t}\right) & \text{if } i \in \mathcal{C}, \end{cases} \tag{4}$$

TABLE I

Examples Corruptions and Capability of an Adversary They Require, as Measured Along the Following Axes: **Data Write**, Where a Device $i \in \mathcal{C}$ Can Replace Its Local Distribution D_i by Any Arbitrary Distribution $\tilde{D}_i$; **Model Read**, Where a Device $i \in \mathcal{C}$ Can Read the Server Model $w^{(t)}$ and Replace Its Local Distribution D_i by an Adaptive Distribution $\tilde{D}_i^{(t)}$ Depending on $w^{(t)}$; **Model Write**, Where a Device $i \in \mathcal{C}$ Can Return an Arbitrary Vector to the Server for Aggregation as in (4), And, **Aggregation**, Where a Device $i \in \mathcal{C}$ Can Behave Arbitrarily During the Computation of an Iterative Secure Aggregate. the Last Column Indicates Whether the Proposed RFA Algorithm is Robust to Each Type of Corruption

Corruption Type	**Data write**	**Model read**	**Model write**	**Aggregation**	**RFA applicable?**
Non-adversarial	-	-	-	-	✓
Static data poisoning	Yes	-	-	-	✓
Adaptive data poisoning	Yes	Yes	-	-	✓
Update poisoning	Yes	Yes	Yes	-	✓
Byzantine	Yes	Yes	Yes	Yes	N/A

where H_i is an arbitrary $\mathbb{R}^d$-valued function which is allowed to depend on the global model $w^{(t)}$, the uncorrupted updates $w_{j,\tau}^{(t)}$ as well as the data distributions D_j of each device $j \in S_t$.

This encompasses situations where the corrupted devices are individually or collectively trying to "attack" the global model, that is, reduce its predictive power over uncorrupted data. We define the *corruption level* ρ as the total fraction of the weight of the corrupted devices:

$$\rho = \frac{\sum_{i \in \mathcal{C}} \alpha_i}{\sum_{i=1}^{n} \alpha_i} \, . \tag{5}$$

Since the corrupted devices can only harm the global model through the updates they contribute in the aggregation step, we aim to robustify the aggregation in federated learning. However, it turns out that robustness is not directly compatible with the two other desiderata of federated learning, namely communication efficiency and privacy.

The Tension Between Robustness, Communication and Privacy: We first argue that any federated learning algorithm can only have two out of the three of robustness, communication and privacy under the existing techniques of secure multi-party computation. The standard approach of FedAvg is communication-efficient and privacy-preserving but not robust, as we discussed earlier. In fact, any aggregation scheme $A(w_1, \ldots, w_m)$ which is a linear function of $w_1, \ldots, w_m$ is similarly non-robust. Therefore, any robust aggregate A must be a non-linear function of the vectors it aggregates.

The approach of sending the updates to the server at a communication of $O(md)$ and utilizing one of the many robust aggregates studied in the literature [e.g. [50], [52], [53]] has robustness and communication efficiency but not privacy. If we try to make it privacy-preserving, however, we lose communication efficiency. Indeed, the secure multi-party computation primitives based on secret sharing, upon which privacy-preservation is built, are communication efficient only for linear functions of the inputs [68]. The additional $O(m \log m)$ overhead of secure averaging for linear functions becomes $\Omega(md \log m)$ for general non-linear functions required for robustness; this makes it impractical for large-scale systems [11]. Therefore, one cannot have both communication efficiency and privacy preservation along with robustness.

In this work, we strike a compromise between robustness, communication and privacy. We will approximate a non-linear robust aggregate as an *iterative secure aggregate*, i.e., as a sequence of weighted averages, computed with a secure average oracle with weights being adaptively updated.

Definition 1: A function $A : (\mathbb{R}^d)^m \to \mathbb{R}^d$ is said to be an iterative secure aggregate of $w_1, \ldots, w_m$ with R communication rounds and initial iterate $v^{(0)}$ if for $r = 0, \ldots, R-1$, there exist weights $\beta_1^{(r)}, \ldots, \beta_m^{(r)}$ such that

i) $\beta_i^{(r)}$ depends only on $v^{(r)}$ and w_i,

ii) $v^{(r+1)} = \sum_{i=1}^{m} \beta_i^{(r)} w_i / \sum_{i=1}^{m} \beta_i^{(r)}$, and,

iii) $A(w_1, \cdots w_m) = v^{(R)}$.

Further, the iterative secure aggregate is said to be s-privacy preserving for some $s \in (0,1)$ if

iv) $\beta_i^{(r)} / \sum_{j=1}^{m} \beta_j^{(r)} \le s$ for all $i \in [m]$ and $r \in [R]$.

If we have an iterative secure aggregate with R communication rounds which is also robust, we gain robustness at a R-fold increase in communication cost. Condition (iv) ensures privacy preservation because it reveals only weighted averages with weights at most s, so a user's update is only available after being mixed with those from a large cohort of devices.

The Tension Between Robustness and Heterogeneity: Heterogeneity is a key property of federated learning. The distribution D_i of device i can be quite different from the distribution D_j of some other device j, reflecting the heterogeneous data generated by a diverse set of users.

To analyze the effect of heterogeneity on robustness, consider the simplified scenario of robust mean estimation in Huber's contamination model [34]. Here, we wish to estimate the mean $\mu \in \mathbb{R}^d$ given samples $w_1, \ldots, w_m \sim (1-\rho)\mathcal{N}(\mu, \sigma^2 I) + \rho Q$, where Q denotes some outlier distribution that ρ-fraction of the points (designated as outliers) are drawn from. Any aggregate $\bar{w}$ must satisfy the lower bound $\|\bar{w} - \mu\|^2 \ge \Omega(\sigma^2 \max\{\rho^2, d/m\})$ with constant probability [69, Theorem 2.2]. In the federated learning setting, more heterogeneity corresponds to a greater variance σ^2 among the inlier points, implying a larger error in mean estimation. This suggests a tension between robustness and heterogeneity, where increasing heterogeneity makes robust mean estimation harder in terms of ℓ_2 error.

In this work, we strike a compromise between robustness and heterogeneity by considering a family $\mathcal{D}$ of allowed data

Algorithm 1: The RFA Algorithm.

```
Input: Initial iterate $w^{(0)}$, number of communication rounds $T$, number of clients per round $m$, number of local updates $\tau$, local step size $\gamma$, approximation threshold $\epsilon$
1:  for $t = 0, 1, \dots, T-1$ do
2:      Sample $m$ clients from $[n]$ without replacement in $S_t$
3:      for each selected client $i \in S_t$ in parallel do
4:          Initialize $w_{i,0}^{(t)} = w^{(t)}$
5:          for $k = 0, \dots, \tau - 1$ do
6:              Sample data $z_{i,k}^{(t)} \sim D_i$
7:              Update $w_{i,k+1}^{(t)} = w_{i,k}^{(t)} - \gamma \nabla f(w_{i,k}^{(t)}; z_{i,k}^{(t)})$
8:          Set $w_i^{(t+1)} = w_{i,\tau}^{(t)}$
9:      $w^{(t+1)} = \mathrm{GM}((w_i^{(t+1)})_{i \in S_t}, (\alpha_i)_{i \in S_t}, \epsilon)$ (Algo. 2)
10: return $w_T$
```

Algorithm 2: The Smoothed Weiszfeld Algorithm.

```
Input: $w_1, \dots, w_m \in \mathbb{R}^d$ with $w_i$ on device $i$, $\alpha_1, \dots, \alpha_m > 0$, $\nu > 0$, budget $R$, $v^{(0)} \in \mathbb{R}^d$, secure average oracle $\mathcal{A}$
for $r = 0, 1, \dots, R-1$ do
    Server broadcasts $v^{(r)}$ to devices $1, \dots, m$
    Device $i$ computes $\beta_i^{(r)} = \alpha_i / (\nu \vee \|v^{(r)} - w_i\|)$
    $v^{(r+1)} \leftarrow (\sum_{i=1}^m \beta_i^{(r)} w_i) / \sum_{i=1}^m \beta_i^{(r)}$ using $\mathcal{A}$
return $v^{(R)}$
```

distributions such that any device i with $D_i \notin \mathcal{D}$ will be regarded as a corrupted device, i.e., $i \in \mathcal{C}$. We will be able to guarantee convergence up to the degree of heterogeneity in $\mathcal{D}$; we call this $\mathrm{width}(\mathcal{D})$ and make it precise in Section IV. In the i.i.d. case, $\mathcal{D}$ is a singleton and $\mathrm{width}(\mathcal{D}) = 0$.

Examples: Next, we consider some examples of update corruption — see [2] for a comprehensive treatment. Corrupted updates could be non-adversarial in nature, such as sensor malfunctions or hardware bugs in unreliable and heterogeneous devices (e.g., mobile phones) which are outside the control of the orchestrating server. On the other hand, we could also have adversarial corruptions of the following types:

a) *Static data poisoning:* The corrupted devices $\mathcal{C}$ are allowed to modify their training data prior to the start of the training, and the data is fixed thereafter. Formally, the objective function of device $i \in \mathcal{C}$ is now $\tilde{F}_i(w) = \mathbb{E}_{z \sim \tilde{D}_i}[f(w; z)]$ where $\tilde{D}_i$ has been modified from the original D_i. These devices then participate in the local updates (2) with $\nabla \tilde{F}_i$ rather than ∇F_i. We consider device i to contribute corrupted updates only if $\tilde{D}_i \notin \mathcal{D}$ (for instance, $\mathcal{D}$ is the set of natural RGB images).

b) *Adaptive data poisoning:* The corrupted devices $\mathcal{C}$ are allowed to modify their training data in each round of training depending on the current model $w^{(t)}$. Concretely, the objective function of device $i \in \mathcal{C}$ in round t is $\tilde{F}_i^{(t)}(w) = \mathbb{E}_{z \sim \tilde{D}_i^{(t)}}[f(w; z)]$ where $\tilde{D}_i^{(t)}$ has been modified from the original D_i using knowledge of $w^{(t)}$. As previously, these devices then participate in the local updates (2) with $\nabla \tilde{F}_i^{(t)}$ rather than ∇F_i in round t.

c) *Update Poisoning:* The corrupted devices can send an arbitrary vector to the server for aggregation, as described by (4) in its full generality. This setting subsumes all previous examples as special cases.

The corruption model in (4) precludes the *Byzantine setting* [e.g., [2], Sec. 5.1], which refers to the worst-case model where a corrupted client device $i \in \mathcal{C}$ can behave arbitrarily, such as for instance, changing the weights $\beta_i^{(r)}$ or the vector w_i between each of the rounds of the iterative secure aggregate, as defined in Definition 1. It is provably impossible to design a Byzantine-robust iterative secure aggregate in this sense. The examples listed above highlight the importance of robustness to the corruption model under consideration.

Table I compares the various corruptions in terms of the capability of an adversary required to induce the corruption.

IV. Robust Aggregation and the RFA Algorithm

In this section, we design a robust aggregation oracle and analyze the convergence of the resulting federated algorithm.

Robust Aggregation with the Geometric Median: The geometric median (GM) of $w_1, \dots, w_m \in \mathbb{R}^d$ with weights $\alpha_1, \dots, \alpha_m > 0$ is the minimizer of

$$g(v) := \sum_{i=1}^m \alpha_i \|v - w_i\| \,, \tag{6}$$

where $\|\cdot\| = \|\cdot\|_2$ is the Euclidean norm. As a robust aggregation oracle, we use an ϵ-approximate minimizer $\widehat{v}$ of g which satisfies $g(\widehat{v}) - \min_z g(v) \le \epsilon$. We denoted it by $\widehat{v} = \mathrm{GM}((w_i)_{i=1}^m, (\alpha_i)_{i=1}^m, \epsilon)$. Further, when $\alpha_i = 1/m$, we write $\mathrm{GM}((w_i)_{i=1}^m, \epsilon)$.

The GM has an optimal breakdown point of 1/2 [70]. That is, to get the geometric median to equal an arbitrary point, at least half the points (in total weight) must be modified. We assume that $w_1, \cdots w_m$ are non-collinear, which is reasonable in the federated setting. Then, g admits a unique minimizer $v^\star$. Further, we assume $\sum_i \alpha_i = 1$ w.l.o.g.[3]

Robust Federated Aggregation (RFA): The RFA algorithm is obtained by replacing the mean aggregation of FedAvg with this GM-based robust aggregation oracle – the full algorithm is given in Algorithm 1. Similar to FedAvg, RFA also trades-off some communication for local computation by running multiple local steps in line 6. The communication efficiency and privacy preservation of RFA follow from computing the GM as an iterative secure aggregate, which we turn to next. Note that RFA is agnostic to the *actual* level of corruption in the problem and the aggregation is robust regardless of the convexity of the local objectives F_i.

Geometric Median as an Iterative Secure Aggregate: While the GM is a natural robust aggregation oracle, the key

[3] One could apply the results to $\widetilde{g}(v) := g(v) / \sum_{i=1}^m \alpha_i$.

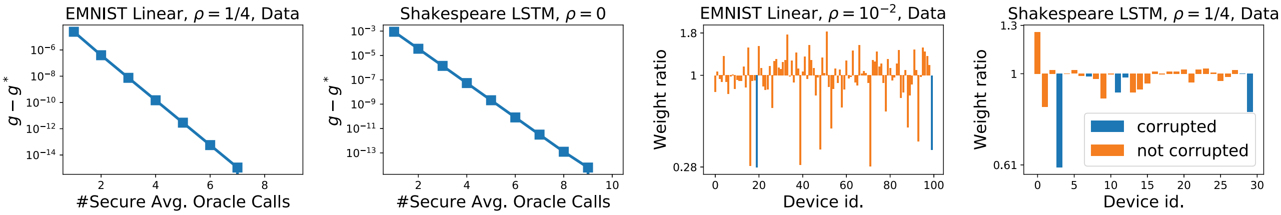

Fig. 1. **Left two**: Convergence of the smoothed Weiszfeld algorithm. **Right two**: Visualization of the re-weighting β_i/α_i, where β_i is the weight of w_i in $\mathrm{GM}((w_i),(\alpha_i)) = \sum_i \beta_i w_i$. See Appendix IV-D for details.

challenge in the federated setting is to implement it as an iterative secure aggregate. Our approach, given in Algorithm 2, iteratively computes a new weight $\beta_i^{(r)} \propto 1/\|v^{(r)} - w_i\|$, up to a tolerance $\nu > 0$, whose role is to prevent division by zero. This endows the algorithm with greater stability. We call it the smoothed Weiszfeld algorithm as it is a variation of Weiszfeld's classical algorithm [14]. The smoothed Weiszfeld algorithm satisfies the following convergence guarantee, proved in Appendix III.

Proposition 2: The iterate $v^{(R)}$ of Algorithm 2 with input $v^{(0)} \in \mathrm{conv}\{w_1, \dots, w_m\}$ and $\nu > 0$ satisfies

$$g(v^{(R)}) - g(v^\star) \le \frac{2\|v^{(0)} - v^\star\|^2}{\overline{\nu} R} + \frac{\nu}{2},$$

where $v^\star = \arg\min g$ and $\overline{\nu} = \min_{r\in[R], i\in[m]} \nu \vee \|v^{(r-1)} - w_i\| \ge \nu$. Furthermore, if $0 < \nu \le \min_{i=1,\dots,m} \|v^\star - w_i\|$, then it holds that $g(v^{(R)}) - g(v^\star) \le 2\|v^{(0)} - v^\star\|^2/\overline{\nu} R$.

For a ϵ-approximate GM, we set $\nu = O(\epsilon)$ to get a $O(1/\epsilon^2)$ rate. However, if the GM $v^\star$ is not too close to any w_i, then the same algorithm automatically enjoys a faster $O(1/\epsilon)$ rate. The algorithm enjoys plausibly an even faster convergence rate *locally*, and we leave this for future work.

The proof relies on constructing a jointly convex surrogate $G : \mathbb{R}^d \times \mathbb{R}^m_{++} \to \mathbb{R}$ defined using $\eta = (\eta_1, \dots, \eta_m) \in \mathbb{R}^m$ as

$$G(v, \eta) := \frac{1}{2} \sum_{k=1}^{m} \alpha_k \left(\frac{\|v - w_k\|^2}{\eta_k} + \eta_k \right).$$

Instead of minimizing $g(v)$ directly using the equality $g(v) = \inf_{\eta > 0} G(v, \eta)$, we impose the constraint $\eta_i \ge \nu$ instead to avoid division by small numbers. The following alternating minimization leads to Algorithm 2:

$$\eta^{(r)} = \arg\min_{\eta \ge \nu} G(v^{(r)}, \eta), \text{ and,}$$
$$v^{(r+1)} = \arg\min_{v \in \mathbb{R}^d} G(v, \eta^{(r)}).$$

Numerically, we find in Fig. 1 that Algorithm 2 is rapidly convergent, giving a **high quality solution in 3 iterations**. This ensures that the approximate GM as an iterative secure aggregate provides robustness at a modest 3× increase in communication cost over regular mean aggregation in FedAvg.

Privacy Preservation: While we can compute the geometric median as an iterate secure aggregate, privacy preservation also requires that the effective weights $\beta_i^{(r)} / \sum_j \beta_j^{(r)}$ are bounded away from 1 for each i. We show this holds for m large.

Proposition 3: Consider $\beta^{(r)}, v^{(r)}$ produced by Algorithm 2 when given $w_1, \cdots w_m \in \mathbb{R}^d$ with weights $\alpha_i = 1/m$ for each i as inputs. Denote $B = \max_{i,j} \|w_i - w_j\|$ and $\overline{\nu}$ as in Proposition 2. Then, we have for all $i \in [m]$ and $r \in [R]$ that

$$\frac{\beta_i^{(r)}}{\sum_{j=1}^m \beta_j^{(r)}} \le \frac{B}{B + (m-1)\overline{\nu}}.$$

Proof: Since $v^{(r)} \in \mathrm{conv}\{w_1, \dots, w_m\}$, we have $\overline{\nu} \le \|v^{(r)} - w_i\| \le B$. Hence, $\alpha_i / B \le \beta_i^{(r)} \le \alpha_i / \overline{\nu}$ for each i and r and the proof follows.

A. *Convergence Analysis of RFA*

We present a convergence analysis of RFA under two simplifying assumptions. First, we focus on least-squares fitting of additive models, as it allows us to leverage sharp analyses of SGD [71]–[73] and focus on the effect of the aggregation. Second, we assume w.l.o.g. that each device is weighted by $\alpha_i = 1/n$ to avoid technicalities of random sums $\sum_{i \in S_t} \alpha_i$. This assumption can be lifted with standard reductions; see Remark 5.

Setup: We are interested in the supervised learning setting where $z_i \equiv (x_i, y_i) \sim D_i$ is an input-output pair. We assume that the output y_i satisfies $\mathbb{E}[y_i] = 0$ and $\mathbb{E}[y_i^2] < \infty$. Denote the marginal distribution of input x_i as $D_{X,i}$. The goal is to estimate the regression function $\overline{x} \mapsto \mathbb{E}[y_i | x_i = \overline{x}]$ from a training sequence of independent copies of $(x_i, y_i) \sim D_i$ in each device. The corresponding objective is the square loss minimization

$$F(w) = \frac{1}{n} \sum_{i=1}^{n} F_i(w), \quad \text{where} \tag{7}$$

$$F_i(w) = \frac{1}{2} \mathbb{E}_{(x,y)\sim D_i} \left(y - w^\top \phi(x) \right)^2 \text{ for all } i \in [n], \tag{8}$$

where, $\phi(x) = (\phi_1(x), \dots, \phi_d(x)) \in \mathbb{R}^d$ where $\phi_1, \dots, \phi_d$ are a fixed basis of measurable, centered functions. The basis functions may be nonlinear, thus encompassing random feature approximations of kernel feature maps and pre-trained deep network feature representations.

We state our results under the following assumptions: (a) the feature maps are bounded as $\|\phi(x)\| \le R$ with probability one under $D_{X,i}$ for each device i; (b) each F_i is μ-strongly convex; (c) the additive model is well-specified on each device: for each device i, there exists $w_i^\star \in \mathbb{R}^d$ such that $y_i = \phi(x_i)^\top w_i^\star + \zeta_i$ where $\zeta_i \sim \mathcal{N}(0, \sigma^2)$. The second assumption is equivalent to requiring that $H_i = \nabla^2 F_i(w) = \mathbb{E}_{x\sim D_{X,i}}[\phi(x)\phi(x)^\top]$, the covariance of x on device i, has eigenvalues no smaller than μ.

Quantifying Heterogeneity: We quantify the heterogeneity in the data distributions D_i across devices in terms of the heterogeneity of marginals $D_{X,i}$ and of the conditional expectation $\mathbb{E}[y_i|x_i = x] = \phi(x)^\top w_i^\star$. Let $H = \nabla^2 F(w) = (1/n)\sum_{i=1}^n H_i$ be the covariance of x under the mixture distribution across devices, where H_i is the covariance of x_i in device i. We measure the dissimilarities $\Omega_X, \Omega_{Y|X}$ of the marginal and the conditionals respectively as

$$\Omega_X = \max_{i\in[n]} \lambda_{\max}(H^{-1/2} H_i H^{-1/2})\,, \tag{9}$$

$$\Omega_{Y|X} = \max_{i,j\in[n]} \|w_i^\star - w_j^\star\|\,, \tag{10}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue. Note that $\Omega_X \ge 1$ and it is equal to 1 iff each $H_i = H$. It measures the spectral misalignment between each H_i and H. The second condition is related to the Wasserstein-2 distance [74] between the conditionals $D_{Y|X,i}$ as $W_2(D_{Y|X,i}, D_{Y|X,j}) \le R\Omega_{Y|X}$. We define the degree of heterogeneity between the various $D_i = D_{X,i} \otimes D_{Y|X,i}$ as $\mathrm{width}(\mathcal{D}) = \Omega_X \Omega_{Y|X} =: \Omega$. That is, if the conditionals are the same ($\Omega_{Y|X} = 0$), we can tolerate arbitrary heterogeneity in the marginals $D_{X,i}$.

Convergence: We now analyze RFA where the local SGD updates are equipped with "tail-averaging" [73] so that $w_i^{(t+1)} = (2/\tau)\sum_{k=\tau/2}^{\tau} w_{i,k}^{(t)}$ is averaged over the latter half of the trajectory of iterates instead of line 9 of Algorithm 1. We show that this variant of RFA converges up to the dissimilarity level $\Omega = \Omega_X\Omega_{Y|X}$ when the corruption level $\rho < 1/2$.

Theorem 4: Consider F defined in (7) and suppose the corruption level satisfies $\rho < 1/2$. Consider Algorithm 1 run for T outer iterations with a learning rate $\gamma = 1/(2R^2)$, and the local updates are run for τ_t steps in outer iteration t with tail averaging. Fix $\delta > 0$ and $\theta \in (\rho, 1/2)$, and set the number of devices per iteration, m as

$$m \ge \frac{\log(T/\delta)}{2(\theta-\rho)^2}\,. \tag{11}$$

Define $C_\theta := (1-2\theta)^{-2}, w^\star = \arg\min F, F^\star = F(w^\star), \kappa := R^2/\mu$ and $\Delta_0 := \|w^{(0)} - w^\star\|^2$. Let $\tau \ge 4\kappa\log(128C_\theta\kappa)$. We have that the event $\mathcal{E} = \bigcap_{t=0}^{T-1}\{|S_t \cap \mathcal{C}| \le \theta m\}$ holds with probability at least $1-\delta$. Further, if $\tau_t = 2^t\tau$ for each iteration t, then the output $w^{(T)}$ of Algorithm 1 satisfies,

$$\mathbb{E}\left[\|w^{(T)}) - w^\star\|^2 \,\middle|\, \mathcal{E}\right] \le \frac{\Delta_0}{2^T} + CC_\theta\left(\frac{d\sigma^2 T}{\mu\tau 2^T} + \frac{\epsilon^2}{m^2} + \Omega^2\right)$$

where C is a universal constant. If $\tau_t = \tau$ instead, then, the noise term above reads $d\sigma^2/\mu\tau$.

Theorem 4 shows near-linear convergence $O(T/2^T)$ up to two error terms in the case that ρ is bounded away from $1/2$ (so that θ and C_θ can be taken to be constants). The increasing local computation $\tau_t = 2^t\tau$ required by this rate is feasible since local computation is assumed to be cheaper than communication.

The first error term is ϵ^2/m^2 due to approximation ϵ in the GM, which can be made arbitrarily small by increasing the number m of devices sampled per round. The second error term Ω^2 is due to heterogeneity. Indeed, exact convergence as $T\to\infty$ is not possible in the presence of corruption: lower bounds for robust mean estimation [e.g. 69, Theorem 2.2] imply that $\|w^{(T)} - w^\star\|^2 \ge C\rho^2\Omega_{Y|X}^2$ w.p. at least 1/2. Consistent with our theory, we find in real heterogeneous datasets in Section V that RFA can lead to *marginally* worse performance than FedAvg in the corruption-free regime ($\rho = 0$). Finally, while we focus on the setting of least squares, our results can be extended to the general convex case.

Remark 5: For unequal weights, we can perform the reduction $\tilde{F}_i(w) = n\alpha_i F_i(w)$, so the theory applies with the substitution $(R^2, \sigma^2, \mu, \Omega_X) \mapsto (c_1\, R^2, c_1\, \sigma^2, c_2\, \mu, (c_1/c_2)\Omega_X)$, where $c_1 = n\max_i \alpha_i$ and $c_2 = n\min_i \alpha_i$.

We use the following convergence result of SGD [72, Theorem 1], [73, Corollary 2].

Theorem 6 ([72], [73]): Consider a F_k from (7). Then, defining $\kappa := R^2/\mu$, the output $\overline{v}_\tau$ of τ steps of tail-averaged SGD starting from $v_0 \in \mathbb{R}^d$ using learning rate $(2R^2)^{-1}$ satisfies

$$\mathbb{E}\|\overline{v}_\tau - w^\star\|^2 \le 2\kappa\exp\left(-\frac{\tau}{4\kappa}\right)\|v_0 - w^\star\|^2 + \frac{8d\sigma^2}{\mu\tau}\,.$$

Proof of Theorem 4: Define the event $\mathcal{E}_t = \{|S_t \cap \mathcal{C}| \le \theta m\}$ so that $\mathcal{E} = \bigcap_{t=0}^{T-1}\mathcal{E}_t$. Hoeffding's inequality gives $\mathbb{P}(\overline{\mathcal{E}_t}) \le \delta/T$ for each t so that $\mathbb{P}(\overline{\mathcal{E}}) \le \delta$ using the union bound. Below, let $\mathcal{F}_t$ denote the sigma algebra generated by $w^{(t)}$.

Consider the local updates on an uncorrupted device $i \in S_t \setminus \mathcal{C}$, starting from $w^{(t)}$. Theorem 6 gives, upon using $\tau_t \ge \tau \ge 4\kappa\log(128C_\theta\kappa)$,

$$\mathbb{E}\left[\|w_i^{(t+1)} - w_i^\star\|^2 \,\middle|\, \mathcal{E}, \mathcal{F}_t\right] \le \frac{1}{64C_\theta}\|w^{(t)} - w_i^\star\|^2 + \frac{8d\sigma^2}{\mu\tau_t}\,.$$

Note that $w^\star = (1/n)\sum_{j=1}^n H^{-1}H_j w_j^\star$, so that

$$\|w^\star - w_i^\star\| \le \frac{1}{n}\sum_{j=1}^n \|H^{-1}H_j(w_j^\star - w_i^\star)\| \le \Omega\,.$$

Using $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$, we get,

$$\begin{aligned}\mathbb{E}\left[\|w_i^{(t+1)} - w^\star\|^2 \,|\, \mathcal{E}, \mathcal{F}_t\right] &\le 2\mathbb{E}\left[\|w_i^{(t+1)} - w_i^\star\|^2 \,\middle|\, \mathcal{E}, \mathcal{F}_t\right] \\ &\qquad + 2\Omega^2 \\ &\le \frac{1}{32C_\theta}\|w^{(t)} - w_i^\star\|^2 + \frac{16d\sigma^2}{\mu\tau_t} + 2\Omega^2 \\ &\le \frac{q}{16C_\theta}\|w^{(t)} - w^\star\|^2 + \frac{16d\sigma^2}{\mu\tau_t} + 4\Omega^2\,.\end{aligned}$$

We now apply the robustness property of the GM ([70, Thm. 2.2] or [75, Lem. 3]) to get,

$$\begin{aligned}&\mathbb{E}\left[\|w^{(t+1)} - w^\star\|^2 \,\middle|\, \mathcal{E}, \mathcal{F}_t\right] \\ &\qquad \le \frac{1}{2}\|w^{(t)} - w^\star\|^2 + \frac{128C_\theta d\sigma^2}{\mu\tau_t} + \Gamma\,,\end{aligned}$$

Algorithm 3: One-step Smoothed Weiszfeld Algorithm.

```
Input: Same as Algorithm 2
Device $i$ sets $\beta_i = \alpha_i / (\nu \vee \|w_i\|)$
return $(\sum_{i=1}^m \beta_i w_i) / \sum_{i=1}^m \beta_i$ using $\mathcal{A}$
```

Algorithm 4: RFA with Personalization.

```
Replace lines 4 to 8 of Algorithm 1 with the following:
Set $u_{i,0}^{(t)} = u_i^{(t)}$ and $w_{i,0}^{(t)} = w^{(t)}$
for $k = 0, \ldots, \tau - 1$ do
    $u_{i,k+1}^{(t)} = u_{i,k}^{(t)} - \gamma \nabla f(w^{(t)} + u_{i,k}^{(t)}; z_{i,k}^{(t)})$ with $z_{i,k}^{(t)} \sim D_i$
for $k = 0, \ldots, \tau - 1$ do
    $w_{i,k+1}^{(t)} = w_{i,k}^{(t)} - \gamma \nabla f(w_{i,k}^{(t)} + u_{i,\tau}^{(t)}; \tilde{z}_{i,k}^{(t)})$ with $\tilde{z}_{i,k}^{(t)} \sim D_i$
Set $w_i^{(t+1)} = w_{i,\tau}^{(t)}$ and $u_i^{(t+1)} = u_{i,\tau}^{(t)}$
```

where $\Gamma = 2C_\theta(\epsilon^2/m^2 + 16\Omega^2)$. Taking an expectation conditioned on $\mathcal{E}$ and unrolling this inequality gives

$$\mathbb{E}\left[\|w^{(T)} - w^\star\|^2 \,\middle|\, \mathcal{E}\right] \le \frac{\Delta_0}{2^T} + \frac{128 C_\theta d \sigma^2}{\mu} \sum_{t=1}^{T} \frac{1}{2^{T-t}\tau_t} + 2\Gamma \,.$$

When $\tau_t = 2^t \tau$, the series sums to $2^{-(T-1)}T/\tau$, while for $\tau_t = \tau$, the series is upper bounded by $2/\tau$.

We now consider RFA in connection with the three factors mentioned in Section III-A.

i) **Communication Efficiency**: Similar to FedAvg, RFA performs multiple local updates for each aggregation round, to save on the total communication. However, owing to the trade-off between communication, privacy and robustness, RFA requires a modest 3× more communication for robustness per aggregation. In the next section, we present a heuristic to reduce this communication cost to one secure average oracle call per aggregation.

ii) **Privacy Preservation**: Algorithm 2 computes the aggregation as an iterative secure aggregate. This means that the server only learns the intermediate parameters after being averaged over all the devices, with effective weights bounded away from 1 (Proposition 3). The noisy parameter vectors sent by individual devices are uniformly uninformative in information theoretic sense with the use of secure multi-party computation.

iii) **Robustness**: The geometric median has a breakdown point of 1/2 [70, Theorem 2.2], which is the highest possible [70, Theorem 2.1]. In the federated learning context, this means that convergence is still guaranteed by Theorem 4 when up to half the points in terms of total weight are corrupted. RFA is resistant to both data or update poisoning, while being privacy preserving. On the other hand, FedAvg has a breakdown point of 0, where a single corruption in each round can cause the model to become arbitrarily bad.

B. *Extensions to RFA*

We now discuss two extensions to RFA to reduce the communication cost (without sacrificing privacy) and better accommodate statistical heterogeneity in the data with model personalization.

One-step RFA: Reducing the Communication Cost: Recall that RFA results in a 3-5× increase in the communication cost over FedAvg. Here, we give a heuristic variant of RFA in an extremely communication-constrained setting, where it is infeasible to run multiple iterations of Algorithm 2. We simply run Algorithm 2 with $v^{(0)} = 0$ and a communication budget of $R = 1$; see Algorithm 3 for details. We find in Section V-C that one-step RFA retains most of the robustness of RFA.

Personalized RFA: Offsetting Heterogeneity: We now show RFA can be extended to better handle heterogeneity in the devices with the use of personalization. The key idea is that predictions are made on device i by summing the shared parameters w maintained by the server with personalized parameters $U = \{u_1, \ldots, u_n\}$ maintained individually on-device. In particular, the optimization problem we are interested in solving is

$$\min_{w,U} \left[F(w,U) := \sum_{i=1}^{n} \alpha_k \, \mathbb{E}_{z\sim D_i}\left[f(w + u_i; z)\right] \right] .$$

We outline the algorithm in Algorithm 4. We train the shared and personalized parameters on each other's residuals, following the residual learning scheme of [76]. Each selected device first updates its personalized parameters u_i while keeping the shared parameters w fixed. Next, the updates to the shared parameter are computed on the residual of the personalized parameters. The updates to the shared parameter are aggregated with the geometric median, identical to RFA. Experiments in Section V-C show that personalization is effective in combating heterogeneity.

V. Numerical Simulations

We now conduct simulations to compare RFA with other federated learning algorithms. The simulations were run using TensorFlow and the data was preprocessed using LEAF [77]. We first describe the experimental setup in Section V-A, then study the robustness and convergence of RFA in Section V-B. We study the effect of the extensions of RFA in Section V-C. The full details from this section and more simulation results are given in Appendix IV. The code and scripts to reproduce these experiments can be found online [16].

A. *Setup*

We consider three machine learning tasks. The datasets are described in Table II. As described in Section III-A, we take the weight α_i of device i to be proportional to the number of datapoints N_i on the device.

a) *Character Recognition:* We use the EMNIST dataset [78], where the input x is a 28×28 grayscale image of a handwritten character and the output y is its identification (0-9, a-z, A-Z). Each device is a writer of the handwritten character x. We use two models — a linear model

TABLE II
Dataset Description and Statistics

Dataset	Task	#Classes	#Train	#Test	#Devices	#Train per Device		
						Median	Max	Min
EMNIST	Image Classification	62	$204K$	$23K$	1000	160	418	92
Shakespeare	Character-level Language Modeling	53	$2.2M$	$0.25M$	628	1170	70600	90
Sent140	Sentiment Analysis	2	$57K$	$15K$	877	55	479	40

$\varphi(x; w) = w^\top x$ and a convolutional neural network (ConvNet). We use as objective $f(w; (x,y)) = \ell(y, \varphi(x; w))$, where ℓ is the multinomial logistic loss ℓ. We evaluate performance using the classification accuracy.

b) *Character-Level Language Modeling:* We learn a character-level language model over the Complete Works of Shakespeare [79]. We formulate it as a multiclass classification problem, where the input x is a window of 20 characters, the output y is the next (i.e., 21st) character. Each device is a role from a play (e.g., Brutus from The Tragedy of Julius Caesar). We use a long-short term memory model (LSTM) [80] together with the multinomial logistic loss. The performance is evaluated with the classification accuracy of next-character prediction.

c) *Sentiment Analysis:* We use the Sent140 dataset [81] where the input x is a tweet and the output $y = \pm 1$ is its sentiment. Each device is a distinct Twitter user. We use a linear model using average of the GloVe embeddings [82] of the words of the tweet. It is trained with the binary logistic loss and evaluated with the classification accuracy.

Corruption Models: We consider the following corruption models for corrupted devices $\mathcal{C}$, cf. Section III-B:

a) *Data Poisoning*: The distribution D_i on a device $k \in \mathcal{C}$ is replaced by some fixed $\tilde{D}_i$. For EMNIST, we take the negative of an image so that $\tilde{D}_i(x,y) = D_i(1 - x, y)$. For the Shakespeare dataset, we reverse the text so that $\tilde{D}_i(c_1, \cdots c_{20}, c_{21}) = D_i(c_{21}, \cdots c_2, c_1)$. In both these cases, the labels are unchanged. For the Sent140 dataset, we flip the label while keeping x unchanged.

b) *Update poisoning with Gaussian corruption:* Each corrupted device $i \in \mathcal{C}$ returns $w_i^{(t+1)} = w_{i,\tau}^{(t)} + \zeta_i^{(t)}$, where $\zeta_i^{(t)} \sim \mathcal{N}(0, \sigma^2 I)$, where σ^2 is the variance across the components of $w_{i,\tau}^{(t)} - w^{(t)}$.[4]

c) *Update poisoning with omniscient corruption:* The parameters $w_i^{(t+1)}$ returned by devices $i \in \mathcal{C}$ are modified so that the weighted arithmetic mean $\sum_{i \in S_t} \alpha_i w_i^{(t+1)}$ over the selected devices S_t is set to $-\sum_{i \in S_t} \alpha_i w_{i,\tau}^{(t)}$, the negative of what it would to have been without the corruption. This is designed to hurt the weighted arithmetic mean aggregation.

Hyperparameters: The hyperparameters are chosen similar to the defaults of [1]. A learning rate schedule was tuned on a validation set for FedAvg with no corruption. The same schedule was used for RFA. The aggregation in RFA is implemented using the smoothed Weiszfeld algorithm with a budget of $R = 3$ calls to the secure average oracle, thanks to its rapid empirical convergence (cf. Fig. 1), and $\nu = 10^{-6}$ for numerical stability. Each simulation was repeated 5 times and the shaded area denotes the minimum and maximum over these runs. Appendix IV gives details on hyperparameter, and a sensitivity analysis of the Weiszfeld communication budget.

B. Robustness and Convergence of RFA

First, we compare the robustness of RFA as opposed to vanilla FedAvg to different types of corruption across different datasets in Fig. 2. We make the following observations.

RFA gives improved robustness to linear models with data corruption: For instance, consider the EMNIST linear model at $\rho = 1/4$. RFA achieves 52.8% accuracy, over 10% better than FedAvg at 41.2%.

RFA performs similarly to FedAvg in deep nets with data corruption: RFA and FedAvg are within one standard deviation of each other for the Shakespeare LSTM model, and nearly equal for the EMNIST ConvNet model. We note that the behavior of the training of a neural network when the data is corrupted is not well-understood in general [e.g., 83].

RFA gives improved robustness to omniscient corruptions for all models: For the omniscient corruption, the test accuracy of the FedAvg is close to 0% for the EMNIST linear model and ConvNet, while RFA still achieves over 40% at $\rho = 1/4$ for the former and well over 60% for the latter. A similar trend holds for the Shakespeare LSTM model.

RFA almost matches FedAvg in the absence of corruption: Recall from Section III-B that robustness comes at the cost of heterogeneity; this is also reflected in the theory of Section IV. Empirically, we find that the performance hit of RFA due to heterogeneity is quite small: 1.4% for the EMNIST linear model (64.3% vs. 62.9%), under 0.4% for the Shakespeare LSTM, and 0.3% for Sent140 (65.0% vs. 64.7%). Further, we demonstrate in Appendix IV-E that, consistent with the theory, this gap completely vanishes in the i.i.d. case.

RFA is competitive with other robust aggregation schemes while being privacy-preserving. We now compare RFA with: (a) coordinate-wise median [52] and ℓ_2 norm clipping [84] which are agnostic to the actual corruption level ρ like RFA, and, (b) trimmed mean [52] and multi-Krum [49], that require exact knowledge of the level of corruption ρ in the problem. We find that RFA is more robust than the two agnostic algorithms coordinate-wise median and norm clipping. Perhaps surprisingly, RFA is also more robust than the trimmed mean which

[4] Model updates $w_i^{(t)} - w^{(t)}$ are aggregated, not the models $w_i^{(t)}$ directly [2].

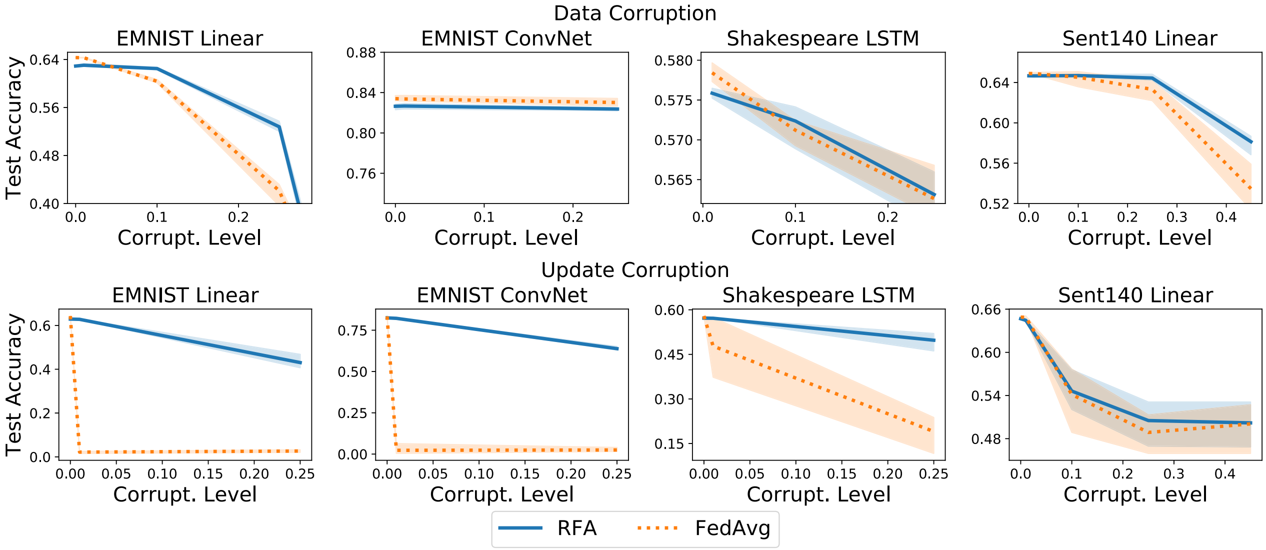

Fig. 2. Comparison of robustness of RFA and FedAvg under data corruption (**top**) and update corruption (**bottom**). The left three plots for update corruption show omniscient corruption while the rightmost one shows Gaussian corruption. The shaded area denotes minimum and maximum over 5 random seeds.

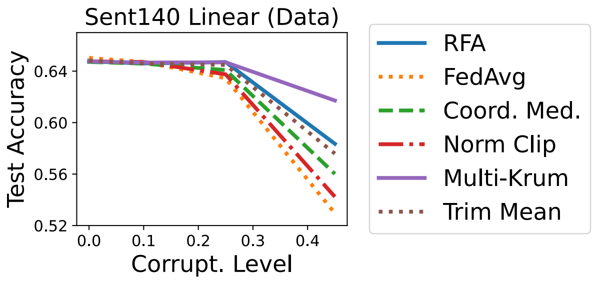

Fig. 3. Comparison of RFA with other robust aggregation algorithms on Sent140 with data corruption.

uses perfect knowledge of the corruption level ρ. We note that multi-Krum is more robust than RFA. That being said, RFA has the advantage that it is fully agnostic to the actual corruption level ρ and is privacy-preserving, while the other robust approaches are not.

Summary: robustness of RFA: Overall, we find that RFA is no worse than FedAvg in the presence of corruption and is often better, while being almost as good in the absence of corruption. Furthermore, RFA degrades more gracefully as the corruption level increases.

RFA requires only $3\times$ the communication of FedAvg: Next, we plot in Fig. 4 the performance versus the number of rounds of communication as measured by the number of calls to the secure average oracle. We note that in the low corruption regime of $\rho = 0$ or $\rho = 10^{-2}$ under data corruption, RFA requires $3\times$ the number of calls to the secure average oracle to reach the same performance. However, it matches the performance of FedAvg when measured in terms of the number of outer iterations, with the additional communication cost coming from multiple Weiszfeld iterations for computation of the average.

RFA exhibits more stable convergence under corruption: We also see from Fig. 4 ($\rho = 1/4$, Data) that the variability of accuracy across random runs, denoted here by the shaded region, is much smaller for RFA. Indeed, by being robust to the corrupted updates sent by random sampling of corrupted clients, RFA exhibits a more stable convergence across iterations.

C. *Extensions of RFA*

We now study the proposed extensions: one-step RFA and personalization.

One-step RFA gives most of the robustness with no extra communication: From Fig. 5, we observe that for one-step RFA is quite close in performance to RFA across different levels of corruption for both data corruption on an EMNIST linear model and omniscient corruption on an EMNIST ConvNet. For instance, in the former, one-step RFA gets 51.4% in accuracy, which is 10% better than FedAvg while being almost as good as full RFA (52.8%) at $\rho = 0.25$. Moreover, for the latter, we find that one-step RFA (67.9%) actually achieves higher test accuracy than full RFA (63.0%) at $\rho = 0.25$.

Personalization helps RFA offset effects of heterogeneity: Fig. 6 plots the effect of RFA with personalization. First, we observe that personalization leads to an improvement with no corruption for both FedAvg and RFA. For the EMNIST linear model, we get 70.1% and 69.9% respectively from 64.3% and 62.9%. Second, we observe that RFA exhibits greater robustness to corruption with personalization. At $\rho = 1/4$ with the EMNIST linear model, RFA with personalization gives 66.4% (a reduction of 3.4%) while no personalization gives 52.8% (a reduction of 10.1%). The results for Sent140 are similar, with the exception that FedAvg with personalization is nearly identical to RFA with personalization.

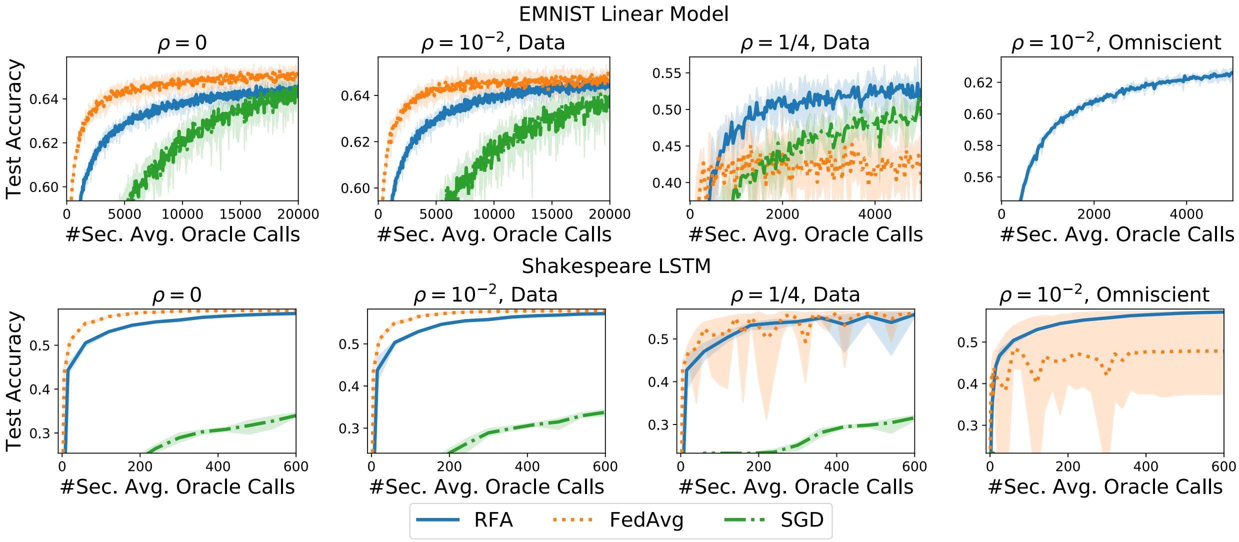

Fig. 4. Comparison of methods plotted against number of calls to the secure average oracle for different corruption settings. For the case of omniscient corruption, FedAvg and SGD are not shown in the plot if they diverge. The shaded area denotes the maximum and minimum over 5 random seeds.

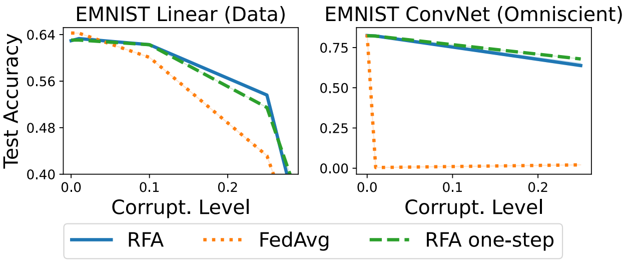

Fig. 5. Robustness of one-step RFA.

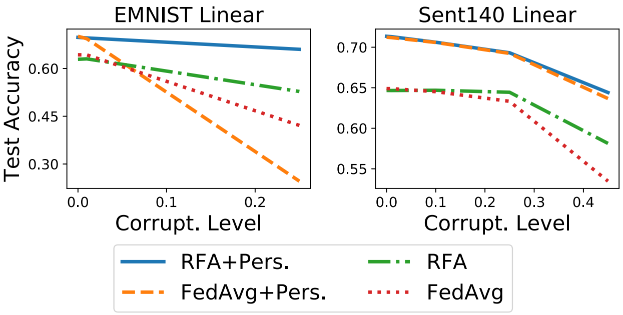

Fig. 6. Effect of personalization on the robustness of RFA and FedAvg under data corruption.

VI. Conclusion

We presented a robust aggregation approach, based on the geometric median and the smoothed Weiszfeld algorithm to efficiently compute it, to make federated learning more robust to settings where a fraction of the devices may be sending corrupted updates to the orchestrating server. The robust aggregation oracle preserves the privacy of participating devices, operating with calls to secure multi-party computation primitives enjoying privacy preservation theoretical guarantees. RFA is available in several variants, including a fast one with a single step of robust aggregation and a one adjusting to heterogeneity with on-device personalization. All variants are readily scalable while preserving privacy, building off secure multi-party computation primitives already used at planetary scale. The theoretical analysis of RFA with personalization is an interesting venue for future work. The further analysis of robustness under heterogeneity is also an interesting venue for future work.

Acknowledgment

The authors would like to thank Zachary Garrett, Peter Kairouz, Jakub Konečný, Brendan McMahan, Krzysztof Ostrowski and Keith Rush for fruitful discussions, as well as help with the implementation of RFA on Tensorflow Federated.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[2] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found.Trends Mach. Learn.*, vol. 14, no. 1-2, pp. 1–210, 2021.

[3] T. Yang *et al.*, "Applied federated learning: Improving google keyboard query suggestions," *CoRR*, vol. abs/1812.02903, 2018, [Online]. Available: http://arxiv.org/abs/1812.02903.

[4] M. Ammad-ud din *et al.*, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *CoRR*, vol. abs/1901.09888, 2019, [Online]. Available: http://arxiv.org/abs/1901.09888

[5] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. Systems, Man, Cybernet., Part C (Appl. Reviews)*, vol. 40, no. 1, pp. 1–12, Jan. 2009.

[6] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *J. Biomed. Informat.*, vol. 99, 2019, Art. no. 103291.

[7] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported Internet of Things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019.

[8] S. Lin, G. Yang, and J. Zhang, "A collaborative learning framework via federated meta-learning," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 289–299.

[9] W. Zhuang, X. Gan, Y. Wen, S. Zhang, and S. Yi, "Collaborative unsupervised visual representation learning from decentralized data," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 4912–4921.

[10] R. A. Maronna, D. R. Martin, and V. J. Yohai, *Robust Statistics: Theory and Methods*. Hoboken, NJ, USA: Wiley, 2018.

[11] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[12] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure single-server aggregation with (Poly) Logarithmic overhead," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 1253–1269.

[13] K. A. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, 2019, pp. 374–388.

[14] E. Weiszfeld, "On the point for which the sum of the distances to n given points is minimum," (in French), *Tohoku Math. J., First Ser.*, vol. 43, pp. 355–386, 1937.

[15] 2019. [Online]. Available: https://github.com/google-research/federated/tree/master/robust_aggregation

[16] 2019. [Online]. Available: https://github.com/krishnap25/rfa

[17] B. Balle, G. Barthe, M. Gaboardi, J. Hsu, and T. Sato, "Hypothesis testing interpretations and renyi differential privacy," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2496–2506.

[18] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Informat. Process. Syst.*, 2017, vol. 30, pp. 4424–4434.

[19] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 4615–4625.

[20] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.

[21] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[22] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21394–21405.

[23] A. Fallah, A. Mokhtari, and A. E. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 3557–3568.

[24] Y. Laguel, K. Pillutla, J. Malick, and Z. Harchaoui, "A superquantile approach to federated learning with heterogeneous devices," in *Proc. IEEE Conf. Inf. Sci. Syst.*, 2021, pp. 1–6.

[25] D. Avdiukhin and S. P. Kasiviswanathan, "Federated learning under arbitrary communication patterns," in *Proc. Int. Conf. Mach. Learn.*, 2021, vol. 139, pp. 425–435.

[26] S. J. Reddi *et al.*, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Representations*, 2021, [Online]. Available: https://openreview.net/forum?id=LkFG3lB13U5.

[27] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[28] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Belmont, MA, USA: Athena Scientific, 1997.

[29] V. Smith, S. Forte, M. Chenxin, M. Takáč, M. I. Jordan, and M. Jaggi, "COCOA: A general framework for communication-efficient distributed optimization," *J. Mach. Learn. Res.*, vol. 18, p. 230:1–230:49, 2018.

[30] C. Ma *et al.*, "Distributed optimization with arbitrary local solvers," *Optim. Methods Softw.*, vol. 32, no. 4, pp. 813–848, 2017.

[31] L. He, A. Bian, and M. Jaggi, "COLA: Decentralized linear learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 4541–4551.

[32] R. Leblond, F. Pedregosa, and S. Lacoste-Julien, "Improved asynchronous parallel optimization analysis for stochastic incremental methods," *J. Mach. Learn. Res.*, vol. 19, pp. 81:1–81:68, 2018.

[33] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4-5, pp. 311–801, 2014.

[34] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.

[35] P. J. Huber, *Robust Statistics*. in *International Encyclopedia of Statistical Science*, Berlin, Heidelberg: Springer, 2011, pp. 1248–1251. doi: 10.1007/978-3-642-04898-2_594.

[36] A. S. Nemirovski and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. Hoboken, NJ, USA: Wiley, 1983.

[37] S. Minsker, "Geometric median and robust estimation in banach spaces," *Bernoulli*, vol. 21, no. 4, pp. 2308–2335, 2015.

[38] D. J. Hsu and S. Sabato, "Loss minimization and parameter estimation with heavy tails," *J. Mach. Learn. Res.*, vol. 17, pp. 18:1–18:40, 2016.

[39] G. Lugosi and S. Mendelson, "Risk minimization by median-of-means tournaments," *J. Eur. Math. Soc.*, vol. 22, no. 3, pp. 925–965, 2019.

[40] G. Lecué and M. Lerasle, "Robust machine learning by median-of-means: Theory and practice," *Ann. Statist.*, vol. 48, no. 2, pp. 906–931, 2020.

[41] G. Lugosi and S. Mendelson, "Regularization, sparse recovery, and median-of-means tournaments," *Bernoulli*, vol. 25, no. 3, pp. 2075–2106, 2019, doi: 10.3150/18-BEJ1046.

[42] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart, "Robust estimators in high dimensions without the computational intractability," in *Proc. Symp. Foundations Comput. Sci.*, 2016, pp. 655–664.

[43] S. Minsker, "Uniform bounds for robust mean estimators," 2019.

[44] Y. Cheng, I. Diakonikolas, and R. Ge, "High-dimensional robust mean estimation in nearly-linear time," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 2019, pp. 2755–2771.

[45] S. Al-Sayed, A. M. Zoubir, and A. H. Sayed, "Robust distributed estimation by networked agents," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 3909–3921, Aug. 2017.

[46] Y. Yu, H. Zhao, R. C. de Lamare, Y. Zakharov, and L. Lu, "Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1566–1581, Mar. 2019.

[47] Y. Chen, S. Kar, and J. M. Moura, "Resilient distributed parameter estimation with heterogeneous data," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4918–4933, Oct. 2019.

[48] L. Lamport, R. E. Shostak, and M. C. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[49] P. Blanchard, R. Guerraoui, E. M. El Mhamdi, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 119–129.

[50] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, 2017, pp. 44:1–44:25.

[51] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "DRACO: Byzantine-resilient distributed training via redundant gradients," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 902–911.

[52] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5636–5645.

[53] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4618–4628.

[54] X. Cao and L. Lai, "Distributed gradient descent algorithm robust to an arbitrary number of Byzantine attackers," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5850–5864, Nov. 2019.

[55] P. Subramanyan, R. Sinha, I. Lebedev, S. Devadas, and S. A. Seshia, "A formal foundation for secure remote execution of enclaves," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 2435–2450.

[56] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1544–1551.

[57] H. W. Kuhn, "A note on Fermat's problem," *Math. Program.*, vol. 4, no. 1, pp. 98–107, Dec. 1973.

[58] Y. Vardi and C.-H. Zhang, "A modified Weiszfeld algorithm for the Fermat-Weber location problem," *Math. Program.*, vol. 90, no. 3, pp. 559–566, 2001.

[59] A. Beck and S. Sabach, "Weiszfeld's method: Old and new results," *J. Optim. Theory Appl.*, vol. 164, no. 1, pp. 1–40, 2015.

[60] I. N. Katz, "Local convergence in fermat's problem," *Math. Program.*, vol. 6, no. 1, pp. 89–104, 1974.

[61] M. B. Cohen, Y. T. Lee, G. L. Miller, J. Pachocki, and A. Sidford, "Geometric median in nearly linear time," in *Proc. Symp. Theory Comput.*, 2016, pp. 9–21.

[62] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceeding Theory Cryptography Conference*, Berlin, Heidelberg, Germany: Springer, 2006, vol. 3876, pp. 265–284.

[63] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete Gaussian mechanism for federated learning with secure aggregation," in *Proc. ICML*, 2021, vol. 139, pp. 5201–5212.

[64] C. Gentry, “Computing arbitrary functions of encrypted data,” *Commun. ACM*, vol. 53, no. 3, pp. 97–105, 2010.

[65] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, “Federated learning: A signal processing perspective,” *CoRR*, vol. abs/2103.17150, 2021, [Online]. Available: https://arxiv.org/abs/2103.17150.

[66] J. Wang *et al.*, “A field guide to federated optimization,” *CoRR*, vol. abs/2107.06917, 2021, [Online]. Available: https://arxiv.org/abs/2107.06917.

[67] D. L. Donoho and P. J. Huber, “The notion of breakdown point,” *A festschrift Erich L. Lehmann*, vol. 157184, pp. 57–184, 1983.

[68] D. Evans *et al.*, “A pragmatic introduction to secure multi-party computation,” *Found. Trends Privacy Secur.*, vol. 2, no. 2-3, pp. 70–246, 2018.

[69] M. Chen, C. Gao, and Z. Ren, “Robust covariance and scatter matrix estimation under Huber’s contamination model,” *Ann. Statist.*, vol. 46, no. 5, pp. 1932–1960, 2018.

[70] H. P. Lopuhaa and P. J. Rousseeuw, “Breakdown points of affine equivariant estimators of multivariate location and covariance matrices,” *Ann. Statist.*, vol. 19, no. 1, pp. 229–248, Mar. 1991.

[71] F. Bach and E. Moulines, “Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 773–781.

[72] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, V. K. Pillutla, and A. Sidford, “A Markov chain theory approach to characterizing the minimax optimality of stochastic gradient descent (for least squares),” in *Proc. Conf. Found. Softw. Technol. Theor. Comput. Sci.*, 2017, vol. 2, pp 2:1–2:10.

[73] P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford, “Parallelizing stochastic gradient descent for least squares regression: Mini-batching, averaging, and model misspecification,” *J. Mach. Learn. Res.*, vol. 18, pp. 223:1–223: 42, 2017.

[74] V. M. Panaretos and Y. Zemel, *An Invitation to Statistics in Wasserstein Space*. Berlin, Germany: Springer Nature, 2020.

[75] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, “Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks,” *IEEE Trans. Signal Process.*, vol. 68, pp. 4583–4596, 2020.

[76] A. Agarwal, J. Langford, and C.-Y. Wei, “Federated residual learning,” *CoRR*, vol. abs/2003.12880, 2020, [Online]. Available: https://arxiv.org/abs/2003.12880.

[77] S. Caldas *et al.*, “LEAF: A benchmark for federated settings,” *CoRR*, vol. abs/1812.01097, http://arxiv.org/abs/1812.01097.

[78] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “EMNIST: An extension of MNIST to handwritten letters,” *CoRR*, vol. abs/1702.05373, 2017, [Online]. Available: http://arxiv.org/abs/1702.05373.

[79] W. Shakespeare, “The complete works of William Shakespeare.” [Online]. Available: https://www.gutenberg.org/ebooks/100

[80] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[81] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” CS224N Project Report, pp. 1–12, 2009.

[82] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in *Proc. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[83] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://dblp.org/rec/conf/iclr/ZhangBHRV17.html?view=bibtex.

[84] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, “Can you really backdoor federated learning?,” *CoRR*, vol. abs/1911.07963, 2019, [Online]. Available: http://arxiv.org/abs/1911.07963.

[85] A. Beck and M. Teboulle, “Smoothing and first order methods: A unified framework,” *SIAM J. Optim.*, vol. 22, no. 2, pp. 557–580, 2012.

[86] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 2016.

[87] J. Mairal, “Optimization with first-order surrogate functions,” in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 783–791.

[88] J. Mairal, “Incremental majorization-minimization optimization with application to large-scale machine learning,” *SIAM J. Optim.*, vol. 25, no. 2, pp. 829–855, 2015.

[89] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Berlin, Germany: Springer, 2018.

[90] A. Beck, “On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes,” *SIAM J. Optim.*, vol. 25, no. 1, pp. 185–209, 2015.

[91] Y. LeCun *et al.*, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.