Securing Distributed Gradient Descent in High Dimensional Statistical Learning

LILI SU[*], CSAIL, EECS, MIT, USA
JIAMING XU, The Fuqua School of Business, Duke University, USA

We consider unreliable distributed learning systems wherein the training data is kept confidential by external workers, and the learner has to interact closely with those workers to train a model. In particular, we assume that there exists a system adversary that can adaptively compromise some workers; the compromised workers deviate from their local designed specifications by sending out arbitrarily malicious messages.

We assume in each communication round, up to q out of the m workers suffer Byzantine faults. Each worker keeps a local sample of size n and the total sample size is $N = nm$. We propose a secured variant of the gradient descent method that can tolerate up to a constant fraction of Byzantine workers, i.e., $q/m = O(1)$. Moreover, we show the statistical estimation error of the iterates converges in $O(\log N)$ rounds to $O(\sqrt{q/N} + \sqrt{d/N})$, where d is the model dimension. As long as $q = O(d)$, our proposed algorithm achieves the optimal error rate $O(\sqrt{d/N})$. Our results are obtained under some technical assumptions. Specifically, we assume strongly-convex population risk. Nevertheless, the empirical risk (sample version) is allowed to be non-convex. The core of our method is to robustly aggregate the gradients computed by the workers based on the filtering procedure proposed by Steinhardt et al. [29]. On the technical front, deviating from the existing literature on robustly estimating a finite-dimensional mean vector, we establish a *uniform* concentration of the sample covariance matrix of gradients, and show that the aggregated gradient, as a function of model parameter, converges uniformly to the true gradient function. To get a near-optimal uniform concentration bound, we develop a new matrix concentration inequality, which might be of independent interest.

CCS Concepts: • **Security and privacy → Distributed systems security**; **Mobile and wireless security**; • **Computing methodologies → Batch learning**; **MapReduce algorithms**;

Additional Key Words and Phrases: Distributed systems, learning, security, Byzantine adversaries, high-dimensional statistics

ACM Reference Format:
Lili Su and Jiaming Xu. 2019. Securing Distributed Gradient Descent in High Dimensional Statistical Learning. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 1, Article 12 (March 2019), 41 pages. https://doi.org/10.1145/3311083

1 INTRODUCTION

Distributed machine learning has been an attractive solution to large-scale problems for years [5]. At the same time, learning in the presence of (possibly malicious) outliers has a deep root in robust statistics [15] and has become an extremely active area recently [7, 9, 10, 19, 29]. However, most of the previous work implicitly assumes that the systems used to carry out the learning task are reliable, i.e., each computing device follows some designed specification. In this work, we consider

[*]This is the corresponding author

Authors' addresses: Lili Su, CSAIL, EECS, MIT, 32 Vassar St, Cambridge, MA, 02139, USA, lilisu@mit.edu; Jiaming Xu, The Fuqua School of Business, Duke University, 100 Fuqua Drive, Durham, NC, USA, jx77@duke.edu.

2476-1249/2019/3-ART12 $15.00
https://doi.org/10.1145/3311083

unreliable distributed learning systems [2, 4, 8, 20, 30, 38] that are prone to system failures or even adversarial attacks. In particular, we assume that there exists a system adversary that can adaptively choose some computing devices to compromise; the compromised devices deviate from their local designed specifications and behave maliciously in an arbitrary manner.

Our consideration of unreliable distributed learning systems is motivated by the recent trends in a new learning framework wherein the training data is kept confidential by external computing devices, and the learner interacts with the external computing devices to train a model. In classical learning frameworks, data is collected from its providers (who may or may not be voluntary) and is stored by the learner. Such data collection immediately leads to data providers' serious privacy concerns, which root in not only purely psychological reasons but also the poor real-world practice of privacy-preserving solutions. In fact, privacy breaches occur frequently, with recent examples including Facebook data leak scandal, iCloud leaks of celebrity photos, and PRISM surveillance program. Putting this privacy risk aside, data providers often benefit from the learning outputs. For example, in medical applications, although participants may be embarrassed about their use of drugs, they might benefit from good learning outputs that can provide high-accuracy predictions of developing diseases.

To resolve this dilemma of data providers, researchers and practitioners have proposed an alternative learning framework wherein the training data is kept confidential by its providers from the learner and these providers function as workers [12, 17, 22]. This framework has been implemented in practical systems such as Google's *Federated Learning* [17, 22], wherein Google tries to learn a model with the training data kept confidential on the users' mobile devices. We refer to this new learning framework as *learning with external workers*. In contrast to the traditional learning framework under which models are trained within data-centers, in *learning with external workers* the learner faces serious *security* risk: (1) some external workers may be highly unreliable or even be malicious (hacked by the system adversary); (2) the learner lacks enough administrative power over those external workers. In this paper, we aim to develop strategies to safeguard distributed machine learning against adversarial workers while keeping the following two key practical constraints in mind: [1]

- Small local samples versus high model dimensions: While the total volume of data over all workers may be large, individual workers may keep only small samples comparing to model dimensions. That is, the training data is *locally* a scarce resource.
- Communication constraints: Similar to other large distributed systems, the external workers are typically highly heterogeneous in terms of computation powers, real-time local computation environments, etc. As a result of this, each round of communication requires synchronization [40]; the transmission between the external workers and the learner typically suffers from high latency and low throughout.

These two constraints together raise significant challenges for designing securing strategies. Without the first constraint, a one-shot outlier-resilient aggregation procedure suffices: each worker separately performs learning based on the local sample and sends the local estimates to the learner who aggregates these estimates to output a final global estimate. This procedure is straightforward to implement and is communication-efficient [13, 40]. However, the correctness of these algorithms crucially relies on the assumption that the local sample size is sufficiently large. In particular, $n = N/m \gg d$, where m is the number of workers, n is the local sample size, $N = nm$ is the total sample size, and d is the model dimension. In contrast, practical distributed learning systems often

[1] Depending on applications, there might be many other constraints such as unevenly distributed training data, intermittent availability of external workers, etc. In addition, different applications might even call for different performance metrics. We would like to explore these more richer settings in our future work.

operate in the regime where $n \ll d$. Two immediate consequences are: (1) to learn an accurate model, the learner has to interact closely with those external workers, and such close interaction gives the adversary more chances to foil the learning process; (2) identifying the adversarial workers based on abnormality is highly challenging, as it becomes difficult to distinguish the statistical errors from the adversarial errors when the sample sizes are small. In addition, due to the randomness of the training data, the estimates computed at different rounds are highly dependent on each other. See Section 1.1.1 for further explanation.

There have been attempts to robustify stochastic gradient descent (SGD) [2, 4] with different focus from what we consider here. In particular, [4] assumes all the workers can access the whole data sample. Similar to ours, the concurrent work [2] considers the scenario where data is generated and stored in a distributed fashion at the workers. However, [2] assumes that in each iteration the workers are able to use *fresh data* to compute the gradients. However, fresh data in each round implies that the local sample size grows with time, which is not necessarily true in some applications. The fresh data assumption is crucial in their analysis: with fresh data, conditioning on the current model parameter estimator, the local gradients computed at different workers become independent, and the existing analysis of robust mean estimation may suffice. In this work, we assume that the sample size is fixed over time[2], and the training data is stored in a distributed fashion [8, 38].

Contributions: In this work, we propose a robust gradient descent method that tolerates up to a constant fraction of adversarial workers (i.e., $\frac{q}{m} = O(1)$) and converges to a statistical estimation error $O(\sqrt{q/N} + \sqrt{d/N})$ in $O(\log N)$ communication rounds; whereas, the minimax-optimal error rate in the failure-free and centralized setting is $O(\sqrt{d/N})$. [3] As long as $q = O(d)$, our proposed algorithm achieves the optimal error rate $O(\sqrt{d/N})$, matching the failure-free optimal error rate. Our results are obtained under some technical assumptions that we hope to relax in the future. Specifically, we assume that the population risk is strongly-convex. Nevertheless, we do allow the empirical risk (sample version) to be non-convex.

On the technical front, to deal with the interplay of the randomness of the data and the iterative updates of the model choice θ, we first establish the concentration of sample covariance matrix of gradients *uniformly*[4] at all possible model parameters; then we prove that our aggregated gradient, as a function of θ, converges uniformly to the population gradient function $\nabla F(\cdot)$. Similar uniform concentration of sample covariance matrix has been derived in [7, Lemma 2.1] under the assumption that the gradients are sub-gaussian. While sub-gaussian *data distribution* is commonly assumed in statistical learning literature, the resulting *gradients* may be sub-exponential or even heavier tailed. For example, in the simplest linear regression example, the gradients are sub-exponential instead of sub-gaussian. Note that standard routine to bounding the spectral norm of the sample covariance matrix is available, see [33, Theorem 5.44] and [1, Corollary 3.8] for example. However, it turns out that using these existing results, the uniform concentration bound obtained is far from being optimal. To this end, we develop a new concentration inequality for matrices with i.i.d. sub-exponential column vectors. This new inequality leads to a near-optimal uniform bound. Relaxing the distributional assumption from sub-gaussian to sub-exponential is highly nontrivial. See [34, Section 1.3] and [1] for details. Our analysis framework developed in this work is not tied to sub-exponential assumptions. With different gradient distributional assumptions such as bounded second moment, one can follow our analysis roadmap to obtain different uniform concentration

[2] Fixed sample size arises, for example, when the model training speed is significantly faster than the data generation speed.
[3] See [37, Section 3.2] for a proof. Note that $O(\sqrt{d/N})$ is the minimax optimal estimation error rate without any additional structure on the model parameter. When the model parameter has additional structure, such as sparsity, the $\sqrt{d}$ factor can possibly be improved.
[4] See [26, 27] and reference therein for details about uniform convergence of functions.

bounds for the sample covariance matrix of gradient vectors, which in turn implies different error bounds to our robust gradient descent method.

Note that in our algorithm, we let each non-faulty worker compute the local gradient based on the *entire* local sample (all n data points). Since n is small, the computational burdens of the workers are reasonable. It has been demonstrated numerically in [18] that in the adversary-free setting, there is a performance improvement when each worker performs a few epochs of SGD before the model updates are aggregated. Whether there will be similar performance improvement in our adversary-prone setting is unclear, and we would like to leave this direction for future exploration.

1.1 Comparison with Robustly Estimating a Finite-dimensional Mean Vector

Our work is closely related to high-dimensional robust mean estimation – a research area that has been intensively studied [7, 9, 10, 19, 29]. High-dimensional[5] robust mean estimation focuses on estimating the mean of a d-dimensional random vector from a contaminated dataset whose $\epsilon > 0$ fraction of data is arbitrarily corrupted.

Our algorithm uses robust mean estimation – in particular, the procedure developed in [29] – as a sub-routine to aggregate the gradients computed by the workers in each iteration. We provide a way to leverage robust mean estimation primitive to design an optimization algorithm that is resilient to adversarial system failures. Similar attempts were made in concurrent work [11, 16, 25]. In particular, [16] focuses on the linear or polynomial regression model, and [11, 25] consider a general machine learning model similar to ours. The correctness of the robust gradient descent method proposed in [25] relies on uniform approximation of the aggregated gradient to the true population gradient [25, Def. 1]; however, only point-wise approximation bound is proved [25, Lemma 1]. In contrast, we prove a high-probability, uniform approximation bound by assuming the local gradients are sub-exponential (See Theorem 3.3). A similar uniform approximation bound is proved in [11, Prop. B.5] with stronger assumptions. In fact, in proving [11, Prop. B.5], they essentially assume the local gradients are bounded[6] by L' in ℓ_2 norm and restrict $N \geq q \gg d^2(L')^2$, where L' is the Lipschitz continuous parameter of local gradients which may scale polynomially with N, d. See Remark 3 for detailed comparisons.

1.1.1 Why uniform convergence? The existing analysis of robust mean estimation assumes that the good data vectors are independently and identically distributed, and hence can only guarantee the performance of the gradient estimation at a *given* θ. However, in our problem, we need to take into account the fact that θ_t is updated iteratively based on the training data; due to the randomness of the training data, though θ_0 might be treated as given, $\{\theta_t\}_{t\geq 1}$ are random. As $\{\theta_t\}_{t\geq 1}$ are obtained based on the same set of training data, they are highly dependent on each other. As a consequence, conditioning θ_t ($t \geq 1$), the gradients computed by the good workers are no longer *i.i.d.*. This is in sharp contrast to [2].

To deal with this interdependency and unspecified behavior of adversarial workers, we instead view each gradient as a *function* of model parameter θ, and aim to robustly estimate the mean of the *infinite-dimensional* gradient function – the true population gradient function. This poses significant challenges in proving the desired robustness guarantees and estimation error bounds. To this end, we establish a uniform concentration of sample covariance matrix of gradient functions. To get a near-optimal uniform concentration bound, we develop a new matrix concentration inequality.

[5] The notion of "high-dimension" here does NOT refer to the setting where $d \gg N$.

[6] Recall that bounded random variables fall within the sub-gaussian family.

1.2 Further Related Work

Recent years have witnessed a flurry of research on securing distributed machine learning algorithms against adversarial attacks. Here we can only hope to cover a fraction of them we see most relevant. See [2, 4, 7–10, 13, 31, 38] and references therein for more details. Both [4, 31] considered a pure optimization framework and characterizations of statistical performance of the learning outputs are left open; whereas [8] studied the same statistical learning framework as ours. In particular, [8] proposed an algorithm that converges in logarithmic rounds to an estimation error $O(\sqrt{dq/N})$ for $q \geq 1$, which is suboptimal up to a multiplicative factor of $\sqrt{q}$. In the low dimensional regime where $d = O(1)$, the concurrent work [38] obtains an order-optimal error rate based on coordinate-wise median and trimmed mean, but the dependency of error rate on dimension d is highly suboptimal and is even inferior to the result in [8]. In this work, we focus on the more general regime of model dimension d.

Notation. We use standard big O notations, e.g., for any sequences $\{a_n\}$ and $\{b_n\}$, $a_n = O(b_n)$ or $a_n \lesssim b_n$ if there is an absolute constant $C > 0$ such that $a_n/b_n \leq C$.

2 SYSTEM MODEL

Let X denote the input data generated from some *unknown* distribution P. Let $\Theta \subset \mathbb{R}^d$ denote the set of all possible model parameters. We consider a risk function $f : \mathcal{X} \times \Theta \to \mathbb{R}$, where $f(x, \theta)$ measures the risk induced by a realization x of the data under the model parameter choice θ. A classical example of the above statistical learning framework is linear regression, where $x = (w, y) \in \mathbb{R}^{d-1} \times \mathbb{R}$ is the feature-response pair and $f(x, \theta) = \frac{1}{2}\left(\langle w, \theta \rangle - y\right)^2$. The learner is interested in learning θ^* which minimizes the *population risk* i.e.,

$$\theta^* \in \arg\min_{\theta \in \Theta} F(\theta) \triangleq \mathbb{E}\left[f(X, \theta)\right] \tag{1}$$

– assuming that $F(\theta) \triangleq \mathbb{E}\left[f(X, \theta)\right]$ is well-defined for every $\theta \in \Theta$. The model choice θ^* is optimal in the sense that it minimizes the expected risk to pay if it is used for prediction.

If the population risk F were known, then θ^* might be computed by solving the minimization problem in (1). In statistical learning, however, the distribution P (thus the population risk F) is typically unknown; instead, training data is available for learning θ^*. Formally, we assume that there exist N i.i.d. data points $X_i \overset{\text{i.i.d.}}{\sim} P$ in the decentralized learning system wherein the training data is kept locally by data providers and cannot be accessed by the learner directly. The learner can only request those providers to compute gradient-like quantities of their locally kept data, as is the case in Federated Learning. We refer to those data providers as workers, as they can be viewed as “recruited” by the learner. We assume there are m workers, and the N data points are distributed evenly across the m workers. Specifically, the index set $[N]$ is partitioned into m subsets S_j such that $|\mathcal{S}_j| = N/m \triangleq n$, and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for $i \neq j$. [7] Notably, n is often much smaller than the model dimension d.

The learner communicates with the workers in synchronous communication rounds, but non-faulty workers do not communicate with each other. We leave the asynchronous communication as one future direction. We use the Byzantine fault model [20] to capture the unreliability and potential malicious behaviors of the workers. It is assumed that up to q out of the m workers suffer Byzantine faults and thus behave arbitrarily and possibly maliciously. Those faulty workers are referred to as Byzantine workers. The arbitrarily faulty behavior arises when the workers are reprogrammed by the system adversary. We assume the learner knows the upper bound q – a standard assumption in literature [7, 10, 29]. Nevertheless, an effective and efficient learning algorithm that does not call for

[7] It would be interesting to consider more general data partitions, and we leave this as one future direction.

the knowledge of q as input is highly desirable. The set of Byzantine workers is allowed to *change* between communication rounds; the adversary can choose different sets of workers to control across communication rounds. Byzantine workers are assumed to have *complete knowledge* of the system, including the total number of workers m, all N data points over the whole system, the programs that the workers are supposed to run, the program run by the learner, and the realization of the random bits generated by the learner. Moreover, Byzantine workers can collude. Nevertheless, when the adversary gives up the control of a worker, this worker recovers and becomes normal immediately. Note that this mobile Byzantine fault model is more general than the most classic Byzantine fault model, where the set of Byzantine workers is fixed throughout an execution.

3 OUR ALGORITHM AND MAIN RESULTS

A standard approach to estimate θ^* in statistical learning is via empirical risk minimization. Given N independent copies $X_1, \cdots, X_N$ of X, the empirical risk function is a random function over Θ defined as $(1/N) \sum_{i=1}^{N} f(X_i, \theta)$ for all $\theta \in \Theta$. By the functional law of large numbers, the empirical risk function converges uniformly to the population risk function $F(\theta)$ in probability as sample size $N \to \infty$.[8] As a consequence, we expect the minimizer of the empirical risk function (which is random) also converges to the population risk minimizer θ^* in probability. While it may be possible to secure the empirical risk minimization using some “robust" versions of empirical risk functions [7, 31], the characterizations of the estimation error are either unavailable or too loose. Moreover, in our distributed setting, it is costly to transmit the local empirical risk functions. Similar observation is made in [25]. In this paper, we take a different approach: Instead of robustifying the empirical risk functions, we aim at robustifying the *learning process*. Specifically, we focus on securing the gradient descent method against the interruption caused by the Byzantine workers during model training. We focus on gradient descent as it is one of the most important and fundamental algorithms in machine learning [3, 28]. At a high level, many machine learning problems are solved by minimizing certain appropriate risk (cost) function using gradient descent [23].

Recall that $\nabla F(\theta_{t-1})$ is the gradient of the population risk at θ_{t-1}, η is some fixed stepsize, and θ_0 is the given initial guess of θ^*. For the perfect gradient descent method, i.e.,

$$\theta_t = \theta_{t-1} - \eta \times \nabla F(\theta_{t-1}), \tag{2}$$

to converge exponentially fast, the following standard assumption is often adopted [6].

Assumption 1. *The population risk function $F : \Theta \to \mathbb{R}$ is M-strongly convex, and differentiable over Θ with L-Lipschitz gradient. That is, for all $\theta, \theta' \in \Theta$,*

$$F(\theta') \geq F(\theta) + \langle \nabla F(\theta),\ \theta' - \theta \rangle + \frac{M}{2} \left\| \theta' - \theta \right\|_2^2,$$
$$\left\| \nabla F(\theta) - \nabla F(\theta') \right\|_2 \leq L \left\| \theta - \theta' \right\|_2.$$

Note that both M and L may scale in d – the dimension of θ. Though we assume strong-convexity of the population risk, the empirical risk can be highly non-convex. Detailed comments on strong convexity assumption can be found in Remark 4.

Our approximate gradient descent method is given by Algorithm 1.

If worker j is non-faulty at round t, on receipt of θ_{t-1}, it computes its local gradient at θ_{t-1} and reports the computed gradient $g_j(\theta_{t-1})$ to the learner; if worker j is Byzantine faulty at round t, it

[8] See [26, 27] and reference therein for details about uniform convergence of functions.

Algorithm 1 Approximate Gradient Descent Method: Round $t \geq 1$

```
The learner:
1: Initialization: Let $\theta_0$ be an arbitrary point in $\Theta$. Let $\eta = \frac{M}{2L^2}$.
2: Broadcast the current model iterate $\theta_{t-1}$ to all workers;
3: Wait to receive all the gradients reported by the $m$ workers; Let $g_j(\theta_{t-1})$ denote the value received from worker $j$.
   If no message from worker $j$ is received, set $g_j(\theta_{t-1})$ to be some arbitrary value;
4: Aggregate gradients: Pass the received gradients to a gradient aggregator $\mathcal{R}$ to obtain an aggregated gradient $G(\theta_{t-1})$, i.e.,
   $$G(\theta_{t-1}) \leftarrow \mathcal{R}(g_1(\theta_{t-1}), \cdots, g_j(\theta_{t-1}), \cdots, g_m(\theta_{t-1})). \quad (3)$$
5: Update: $\theta_t \leftarrow \theta_{t-1} - \eta \times G(\theta_{t-1})$;
Worker j:
1: On receipt of $\theta_{t-1}$, compute the gradient at $\theta_{t-1}$, i.e., $\frac{1}{n}\sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta_{t-1})$;
2: Send $\frac{1}{n}\sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta_{t-1})$ back to the learner;
```

reports an arbitrary value to the learner. Formally,

$$g_j(\theta_{t-1}) = \begin{cases} \frac{1}{n}\sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta_{t-1}), & \text{if } j \text{ is non-faulty at round } t; \\ \star, & \text{otherwise,} \end{cases}$$

where $\star$ denotes an arbitrary value. Notably, the Byzantine workers might use all the information of the system to determine what value to report. The learner aggregates the received gradients via a gradient aggregator $\mathcal{R}$ (an algorithmic function) to obtain an approximate gradient (line 4 of Algorithm 1).

We use a gradient aggregator that originates from robust mean estimation [29]. For ease of exposition, we postpone the presentation of this gradient aggregator after stating our main results.

3.1 Main Results

To characterize the statistical estimation error rate of our proposed algorithm, we adopt some assumptions. Similar assumptions are made in [8] and [38]. We illustrate our results by applying them to the classical linear regression and logistic regression problems in Section 5.

Though we successfully relax the distributional assumption from sub-gaussian to sub-exponential, no evidence so far hints that no further relaxation is possible. In fact, our analysis framework developed in this work is not tied to sub-exponential assumptions. With different gradient distributional assumptions such as bounded second moment, one can follow our analysis roadmap (Lemma 2) to obtain different uniform concentration bounds for the sample covariance matrix of gradient vectors, which in turn implies different error bounds to our robust gradient descent method. Detailed comments on our distributional assumptions can be found in Remark 5.

The concurrent work [11] attempts to prove uniform approximation bounds under bounded second moment assumption. However, their proof, in its current form, only holds for sub-gaussian distributions. See the proof of [11, Prop B.5] for details. More technical comparisons with the concurrent papers [11, 25] can be found in Remark 3.

Let $S^{d-1} = \left\{v \in \mathbb{R}^d : \|v\|_2 = 1\right\}$ denote the unit Euclidean sphere.

Assumption 2. *The sample gradient at the optimal model parameter θ^*, i.e., $\nabla f(X,\theta^*)$, is sub-exponential with constants (σ_1,α_1), i.e., for every unit vector $v \in S^{d-1}$,*

$$\mathbb{E}\left[\exp\left(\lambda\langle\nabla f(X,\theta^*),v\rangle\right)\right] \le e^{\sigma_1^2\lambda^2/2}, \quad \forall|\lambda| \le \frac{1}{\alpha_1}.$$

We further assume the Lipschitz continuity of the sample gradient functions.

Assumption 3. *There exists an L' such that*

$$\|\nabla f(X,\theta) - \nabla f(X,\theta')\|_2 \ \le\ L'\,\|\theta-\theta'\|_2 \ \forall\ \theta,\theta' \in \Theta.$$

For applications where Assumption 3 does not hold deterministically, it suffices to have Assumption 3 hold with high probability for all training data. Notably, L' may be much larger than L and even scale polynomially in N and d. However, L' will affect our results only by logarithmic factors $\log L'$.

Next define the gradient difference function

$$h(X,\theta) \ =\ \nabla f(X,\theta) - \nabla f(X,\theta^*) - (\nabla F(\theta) - \nabla F(\theta^*))\,. \tag{4}$$

Note that $h(X,\theta)/\,\|\theta-\theta^*\|_2$ characterizes the change rate of $f(X,\theta)-\nabla F(\theta)$ from $f(X,\theta^*)-\nabla F(\theta^*)$; hence it can be viewed as a local Lipschitz parameter with respect to θ^*.

Assumption 4. *The local Lipschitz parameter $h(X,\theta)/\,\|\theta-\theta^*\|_2$ is sub-exponential with constants (σ_2,α_2), i.e.., for every $\theta \in \Theta$ and $v \in S^{d-1}$,*

$$\mathbb{E}\left[\exp\left(\frac{\lambda\langle h(X,\theta),v\rangle}{\|\theta-\theta^*\|}\right)\right] \le e^{\sigma_2^2\lambda^2/2}, \quad \forall|\lambda| \le \frac{1}{\alpha_2}.$$

Notably, Assumption 4 assumes a concentration of the *local* Lipschitz parameter with respect to θ^*, instead of a *global* Lipschitz parameter.

Again, our analysis may not be tied to sub-exponential assumptions. Now we are ready to present our main results.

Theorem 3.1. *Suppose Assumptions 1–4 hold. Assume that $\log(L+L') = O(\log(Nd))$ and $\Theta \subset \{\theta\ :\ \|\theta-\theta^*\|_2\ \le\ r\}$ for some positive parameter r such that $\log r\ =\ O(\log(Nd))$. Suppose that $N \ge cd^2\log^8(Nd))$ and $N \ge cq$ for a sufficiently large constant c, and that $4q \le m \le e^{\sqrt{d}}$. Further assume that $M \ge 1$. Then there exists a gradient aggregator $\mathcal{R}$ such that with probability at least $1-3e^{-\sqrt{d}}$, the iterates $\{\theta_t\}$ given by Algorithm 1 satisfy*

$$\|\theta_t-\theta^*\|_2 \lesssim \left(1-\frac{M^2}{16L^2}\right)^t \|\theta_0-\theta^*\|_2 + \sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}.$$

Note that in Theorem 3.1, the Lipschitz parameters L, L', and the size of the search space r are allowed to scale even polynomially in N and d. The estimation error $O(\sqrt{q/N}+\sqrt{d/N})$ in Theorem 3.1 significantly improves the previous results ($O(\sqrt{dq/N})$ for $q \ge 1$) [8]. Recall that even in the failure-free and centralized setting the minimax-optimal error rate is $O(\sqrt{d/N})$. Thus, an immediate consequence of Theorem 3.1 is that as long as $q\ =\ O(d)$, our proposed algorithm achieves the optimal error rate $O(\sqrt{d/N})$. The sample complexity $N \ge cd^2\log^8(Nd)$ appears to be suboptimal at first glance. However, it turns out that under sub-exponential distributional assumption, if we rely on uniform concentration of the sample covariance matrices, this sample complexity is order optimal up to logarithmic factors. See Remark 2 for details.

Remark 1. *One concurrent work [38] uses coordinate-wise median and trimmed mean and obtains an estimation error $rd(q/(m\sqrt{n})+1/\sqrt{N})$ up to logarithmic factors – noting that the radius of the* *model parameter space r is typically on the order of $\sqrt{d}$. This error rate is shown to be order-optimal in the low dimensional regime where $d = O(1)$ [38], but turns out to scale poorly in dimension d.*

As an important ingredient of our proof of Theorem 3.1, we establish a *uniform* concentration of the sample covariance matrix of gradients. Recall that in our problem local sample gradients $\frac{1}{n}\sum_{i\in\mathcal{S}_j}\nabla f(X_i,\theta)$'s are sub-exponential random vectors. Standard routine to bounding the spectral norm of the sample covariance matrix is available, see [33, Theorem 5.44] and [1, Corollary 3.8] for example. However, it turns out that using these existing results, the uniform concentration bound obtained is far from being optimal. To this end, we develop a new concentration inequality for matrices with i.i.d. sub-exponential column vectors. As can be seen later, this new inequality leads to a near-optimal uniform bound.

Theorem 3.2. *Let A be a $d\times m$ matrix whose columns A_j are independent and identically distributed sub-exponential, zero-mean random vectors in $\mathbb{R}^d$ with parameters (σ,α). Assume that*

$$\sigma/\alpha = \Omega(1). \tag{5}$$

Then with probability at least $1-\delta$,

$$\|A\|_2 \le c\left(\sigma\sqrt{m} + \sigma\phi\left(d+\log\frac{1}{\delta}\right) + \alpha\phi^2\left(d+\log\frac{1}{\delta}\right)\right),$$

where c is a universal positive constant and $\phi(x):\mathbb{R}\to\mathbb{R}$ is a function given by $\phi(x) = \sqrt{x}\log^{3/2}(x)$.

If A has sub-Gaussian columns, i.e., $\alpha = 0$, then the upper bound in Theorem 3.2 matches the sub-Gaussian matrix concentration inequality [35][Theorem 5.39] up to logarithmic factors. If $\sigma,\alpha = \Theta(1)$ and $\log(1/\delta) = d$, Theorem 3.2 implies that with probability at least $1-e^{-d}$, $\|A\|_2 \lesssim \sqrt{m} + d\log^3 d$, which is tight up to logarithmic factors; whereas the analogous bound implied by standard concentration inequality [1, Corollary 3.8] is on the order of $\sqrt{md}+d$. See Remark 10 in Section 4.1 for details.

With the performance guarantee of our robust aggregator $\mathcal{R}$ (formally stated in the next subsection) and Theorem 3.2, it can be shown that the approximate gradients used in the robust gradient descent update (Algorithm 1 with the chosen robust aggregator $\mathcal{R}$) are good uniformly over $\theta\in\Theta$.

Theorem 3.3. *Suppose Assumptions 1–4 hold. Assume that $\log(L+L') = O(\log(Nd))$ and $\Theta\subset\{\theta : \|\theta-\theta^*\|_2 \le r\}$ for some positive parameter r such that $\log r = O(\log(Nd))$. Suppose $N \ge cd^2\log^8(Nd))$ for a sufficiently large constant c and $m \le e^{\sqrt{d}}$. Let $G(\theta)$ (for each $\theta\in\Theta$) be the aggregated gradient returned by Algorithm 2. Then with probability at least $1-3e^{-\sqrt{d}}$,*

$$\begin{aligned}\|G(\theta)-\nabla F(\theta)\|_2 \;\lesssim\; &\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\log^2(Nd)\right)\|\theta-\theta^*\|_2\\ &+\sqrt{\frac{q}{N}}+\sqrt{\frac{d}{N}}\end{aligned}$$

holds for all $\theta\in\Theta$.

Remark 2. *Theorem 3.3 requires the total sample size $N \gtrsim d^2$ (ignoring the logarithmic factors), which is due to our sub-exponential assumption of local Lipschitz parameter $h(X,\theta)/\|\theta-\theta^*\|_2$. This sample size requirement $N\gtrsim d^2$ is inevitable as can be seen from the linear regression example. (cf. Remark 11). If instead $h(X,\theta)/\|\theta-\theta^*\|_2$ is assumed to be sub-Gaussian, then $N\gtrsim d$ suffices.*

Remark 3. *The robust gradient estimation has also been studied in two concurrent papers [11, 25] for $m = N$. Under an ϵ-contamination model with $\epsilon = q/N$, [25, Lemma 1] proves a* point-wise *approximation bound to $\nabla F(\theta)$ for a given θ, which scales as $\sqrt{q/N} + (d/N)^{3/8} + q^{1/4}\sqrt{d/N}$ up* *to logarithmic factors. In contrast, a uniform approximation bound similar to ours is proved in [11, Prop B.5]. However, their bound only holds under the stringent condition $q \gg d^2(L')^2$, where L' is the Lipschitz continuity parameter of the sample gradient function given in Assumption 3. Note that L' may scale polynomially in N, d. For instance, in the standard linear regression model, $L' = \Omega(d)$ (See Lemma 5.1). In fact, [11, Prop B.5] assumes that $\|\nabla f(X,\theta) - \nabla F(\theta)\|_2$ is bounded by L' and hence the proof follows from a straightforward application of Hoeffding's inequality plus an ϵ-net argument.*

In contrast, with Assumption 4, we obtain a tighter uniform approximation bound via the new matrix concentration inequality given in Theorem 3.2.

Remark 4 (Strongly-convex assumption). *We next explain the strong convexity assumption assumed in our paper. First, we clarify that we only require* the population risk *to be strongly-convex, while the empirical risk (sample version) could be highly non-convex. Second, we point out that it is possible to enforce the strong convexity of the population risk by introducing proper regularization. For example, we add an extra ℓ_2 norm regularization to the quadratic loss in ridge regression. Thirdly, while our results do not directly apply to settings where the population risk is not strongly convex, our results are still important for the following two-fold reasons: (1) while many learning problems are highly non-convex globally, they sometimes satisfy certain* restricted strong convexity *properties (the Hessian matrix are strictly positive definite in certain regions or directions around the optimal model parameter) [24] and thus gradient descent schemes are still able to converge to the optimal model parameter [21]. Therefore, our robust gradient descent results can still be applied to these settings; (2) It is possible to extend our results to non-convex settings by combining our robust gradient descent methods with proper saddle-points escaping schemes; such extension has been pursued recently in a follow-up work [39].*

Remark 5. *Finally, we explain the sub-exponential assumption on the gradient vectors. First, while sub-gaussian* data distribution *is commonly assumed in statistical learning literature, the resulting* gradients *may be sub-exponential, as we illustrated in the simple linear regression example. Loosely speaking, this is due to the fact that the gradients may involve x^2 term, which is sub-exponential even if x is sub-Gaussian. Similar phenomenon also occurs in logistic regression. Thus, it is important to consider sub-exponential or even heavier-tailed* gradient distribution *when we try to robustify a learning procedure such as gradient descent. Second, our analysis framework developed in this work is not tied to sub-exponential assumptions. With different gradient distributional assumptions such as bounded second moment, one can follow our analysis roadmap (Lemma 2) to obtain different uniform concentration bounds for the sample covariance matrix of gradient vectors which in turn implies different error bounds to our robust gradient descent method. Thirdly, certain distributional assumptions are inevitable to some extent, in order to show the learning procedures perform well on the average case beyond the worst case guarantees [35].*

3.2 Robust Gradient Aggregator

In this subsection, we present the robust gradient aggregator $\mathcal{R}$ used in Algorithm 1. We present the aggregator in the setup of robust mean estimation.

Let $\mathcal{S} = \{y_1, \cdots, y_m\}$ be the true sample. Define $\mu_{\mathcal{S}} = \frac{1}{m}\sum_{i=1}^m y_i$ as the sample mean on $\mathcal{S}$. Let $\{\widehat{y}_1, \cdots, \widehat{y}_m\} \subseteq \mathbb{R}^d$ be the observed sample, which is obtained from $\mathcal{S}$ by adversarially corrupting up to $q = \epsilon m$ data points. We use an iterative filtering algorithm proposed in [29], formally presented in Algorithm 2. At a high level, by solving (6) and (7) for a saddle point (W, U), Algorithm 2 iteratively finds a direction (given by U^*) along which all data points are spread out the most, and filters away data points which have large residual errors projected along this direction (given by (8)). See Appendix C for detailed discussions.

Algorithm 2 Iterative Filtering for Robust Mean Estimation [29]

```
        Input: Sample $\{\widehat{y}_1, \cdots, \widehat{y}_m\} \subseteq \mathbb{R}^d$, $1 - \alpha \triangleq \epsilon \in [0, \frac{1}{4})$, and $\sigma > 0$.
        Initialization: $\mathcal{A} \leftarrow \{1, \cdots, m\}$, $c_i \leftarrow 1$ and $\tau_i \leftarrow 0$ for all $i \in \mathcal{A}$.
 1: while true do
 2:     For $W \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$ and $U \in \mathbb{R}^{d \times d}$, define a cost function $\psi : (W, U) \to \mathbb{R}$ as:
            $$\psi(W, U) = \sum_{i \in \mathcal{A}} c_i \Big(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji}\Big)^\top U \Big(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji}\Big).$$
        Let $W^*$ be a minimizer to the following convex program:
            $$\min_{\substack{0 \le W_{ji} \le \frac{4-\alpha}{\alpha(2+\alpha)m} \\ \sum_{j \in \mathcal{A}} W_{ji} = 1}} \max_{\substack{U \succeq 0 \\ \mathsf{Tr}(U) \le 1}} \psi(W, U) \tag{6}$$
        and $U^*$ be a maximizer to the following convex program:
            $$\max_{\substack{U \succeq 0 \\ \mathsf{Tr}(U) \le 1}} \min_{\substack{0 \le W_{ji} \le \frac{4-\alpha}{\alpha(2+\alpha)m} \\ \sum_{j \in \mathcal{A}} W_{ji} = 1}} \psi(W, U) \tag{7}$$
 3:     For $i \in \mathcal{A}$,
            $$\tau_i \leftarrow \Big(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W^*_{ji}\Big)^\top U^* \Big(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W^*_{ji}\Big). \tag{8}$$
 4:     if $\sum_{i \in \mathcal{A}} c_i \tau_i > 8m\sigma^2$ then
 5:         For $i \in \mathcal{A}$, $c_i \leftarrow \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i$, where $\tau_{\max} = \max_{i \in \mathcal{A}} \tau_i$.
 6:         $\mathcal{A} \leftarrow \mathcal{A} / \left\{i : c_i \le \frac{1}{2}\right\}$.
 7:     else
 8:         Break while-loop.
 9:     end if
10: end while
11: return $\widehat{\mu} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \widehat{y}_i$.
```

Given the corrupted sample $\{\widehat{y}_1, \cdots, \widehat{y}_m\}$, ϵ, and σ, Algorithm 2 *deterministically* outputs an estimate $\widehat{\mu}$ that differs from the true *sample mean* by at most a bounded distance, formally stated in Lemma 3.4.

Lemma 3.4. *[29] Suppose that*

$$\left\| \frac{1}{m} \sum_{i \in \mathcal{S}} (y_i - \mu_{\mathcal{S}}) (y_i - \mu_{\mathcal{S}})^\top \right\|_2 \le \sigma^2. \tag{9}$$

Then for $q/m = \epsilon \le \frac{1}{4}$, Algorithm 2 outputs a parameter $\widehat{\mu}$ such that

$$\|\widehat{\mu} - \mu_{\mathcal{S}}\|_2 = O(\sigma \sqrt{\epsilon}). \tag{10}$$

Condition (9) ensures that the uncorrupted data points y_i's are well concentrated around the sample mean $\mu_{\mathcal{S}}$ in every direction. If there are large residual errors found by Step 3 of Algorithm 2, they are likely caused by the corrupted data points rather than the good data points.

Remark 6. *Note that condition* (9) *is slightly different from that in [29]: the summation is taken over the entire true sample $\mathcal{S}$ rather than a subset of sample. We make this modification in order to include the regime $q/m = o(1)$. For completeness, we present the proof of Lemma 3.4 in Appendix C.*

Also, the termination of Algorithm 2 requires the knowledge of σ; however, in the setup of statistical learning, this might further call for the knowledge of θ^*, which is not practical. Considering this, we have an alternative termination condition which does not need to know any parameter other than ϵ. See Appendix C.3 for details.

Formally, we use Algorithm 2 as our robust gradient aggregator $\mathcal{R}$ with inputs

$$\widehat{y}_1(\theta) = g_1(\theta), \cdots, \widehat{y}_m(\theta) = g_m(\theta),$$

where $g_1(\theta), \cdots, g_m(\theta)$ are the local gradient functions computed by the m workers, among which up to q reported gradient functions may not be the true local gradient functions. The true m local gradient functions are given by

$$y_1(\theta) = \tfrac{1}{n} \textstyle\sum_{i \in \mathcal{S}_1} \nabla f(X_i, \theta), \cdots, y_m(\theta) = \tfrac{1}{n} \textstyle\sum_{i \in \mathcal{S}_m} \nabla f(X_i, \theta). \tag{11}$$

– recalling that $|\mathcal{S}_j| = n = \frac{N}{m}$ for each $j \in [m]$. The true sample mean $\mu_{\mathcal{S}}(\theta)$ is

$$\mu_{\mathcal{S}}(\theta) = \frac{1}{m} \sum_{j=1}^{m} y_j(\theta) = \frac{1}{N} \sum_{i=1}^{N} \nabla f(X_i, \theta), \tag{12}$$

and the population mean $\mu(\theta)$ is $\nabla F(\theta)$.

Note that other robust mean estimation algorithms given in [10, 19] might also suffice for our purpose, and we would like to explore these different robust gradient aggregation schemes in the future.

4 MAIN ANALYSIS

Before presenting our main analysis, we briefly discuss two important implications of Lemma 3.4 in the robust mean estimation setup.

Remark 7. *In Lemma 3.4, the estimation error bound (10) is in terms of $\|\widehat{\mu} - \mu_{\mathcal{S}}\|_2$. Let μ be the true mean of the unknown underlying distribution. We can easily deduce an estimation error bound in terms of $\|\widehat{\mu} - \mu\|_2$ from the following triangle inequality*

$$\|\widehat{\mu} - \mu\|_2 \le \|\widehat{\mu} - \mu_{\mathcal{S}}\|_2 + \|\mu_{\mathcal{S}} - \mu\|_2 = O\left(\sigma\sqrt{\epsilon}\right) + \|\mu_{\mathcal{S}} - \mu\|_2 .$$

Thus, to characterize the estimation error $\|\widehat{\mu} - \mu\|_2$, it is enough to control the spectral norm of the true *sample covariance matrix $\left\| \frac{1}{m} \sum_{i \in \mathcal{S}} (y_i - \mu_{\mathcal{S}}) (y_i - \mu_{\mathcal{S}})^\top \right\|_2$ and the deviation of the empirical average $\|\mu_{\mathcal{S}} - \mu\|_2$ – the latter of which is standard.*

Remark 8. *Note that*

$$\begin{aligned}
&\left\| \frac{1}{m} \sum_{i \in \mathcal{S}} (y_i - \mu_{\mathcal{S}}) (y_i - \mu_{\mathcal{S}})^\top \right\|_2 \\
&= \frac{1}{m} \left\| \left([y_1, \cdots, y_m] - \mu_{\mathcal{S}} \mathbf{1}_m^\top \right) \left([y_1, \cdots, y_m] - \mu_{\mathcal{S}} \mathbf{1}_m^T \right)^\top \right\|_2 \\
&= \frac{1}{m} \left\| [y_1, \cdots, y_m] - \mu_{\mathcal{S}} \mathbf{1}_m^\top \right\|_2^2 \\
&\le \frac{1}{m} \left(\left\| [y_1, \cdots, y_m] - \mu \mathbf{1}_m^\top \right\|_2 + \sqrt{m} \|\mu - \mu_{\mathcal{S}}\|_2 \right)^2 ,
\end{aligned}$$

where $\mathbf{1}_m \in \mathbb{R}^m$ is an all-ones vector. Therefore, to derive σ in condition (9), *it is enough to bound* $\|\mu - \mu_{\mathcal{S}}\|_2$ *and* $\frac{1}{\sqrt{m}} \|[y_1, \cdots, y_m] - \mu \mathbf{1}_m^\top\|_2$.

Back to our statistical learning problem, as discussed in Remarks 7 and 8, to guarantee that the aggregated gradient is close to the true gradient uniformly over all $\theta \in \Theta$, it suffices to bound

$$\left\| \tfrac{1}{N} \textstyle\sum_{i=1}^N \nabla f(X_i, \theta) - \nabla F(\theta) \right\|_2 \tag{13}$$

and

$$\frac{1}{\sqrt{m}} \Big\| \Big[\tfrac{1}{n} \textstyle\sum_{i \in \mathcal{S}_1} \nabla f(X_i, \theta) - \nabla F(\theta), \cdots, \tfrac{1}{n} \textstyle\sum_{i \in \mathcal{S}_m} \nabla f(X_i, \theta) - \nabla F(\theta) \Big] \Big\| \tag{14}$$

uniformly over all $\theta \in \Theta$. Getting a uniform bound to (13) involves standard concentration of sum of i.i.d. random vectors and is relatively easy. The main challenge is to uniformly bound (14), for which we develop a (nearly tight) matrix concentration inequality.

4.1 New Matrix Concentration Inequality: Theorem 3.2 and its Proof

The aim of this subsection is to present our main technical tool to derive a tight uniform bound to (14). This subsection is independent from the rest of the paper and can be skipped at the first reading.

For any fixed θ, the matrix in (14) is of independent columns; standard routine to bound (14) point-wise is available, see [33, Theorem 5.44] and [1, Corollary 3.8] for example. To get a uniform concentration result, we can use $\epsilon-$net argument to extend the concentration of a fixed θ to uniform over all $\theta \in \Theta$. Nevertheless, using these standard matrix concentration results, the uniform concentration bound obtained is far from being optimal, as we explain next.

The following theorem is, to the best of our knowledge, a state-of-the-art concentration inequality for matrices with sub-exponential columns [1, Corollary 3.8].

Theorem 4.1. *Let A be a $d \times m$ matrix whose columns A_j are i.i.d., zero-mean, sub-exponential random vectors in $\mathbb{R}^d$ with the scaling parameters σ and α. Assume that $\sigma, \alpha = O(1)$ and $m \le e^{\sqrt{d}}$. There are absolute positive constants C and c such that for every $K \ge 1$, with probability at least $1 - e^{-cK\sqrt{d}}$,*

$$\|A\|_2 \le CK \left(\sqrt{m} + \sqrt{d} \right).$$

Note that assuming $m \le e^{\sqrt{d}}$ only loses minimal generality in the high-dimensional regime. The above theorem is tight up to constant factors when the tail probability is on the order of $e^{-\sqrt{d}}$, i.e., $K = \Theta(1)$, see [1, Remark 3.7] for a proof. However, in our problem, to guarantee a uniform bound to (14) via an $\epsilon-$net argument, we need a tail probability on the order of e^{-d}, i.e., $K \approx \sqrt{d}$. In this case, Theorem 4.1 yields an upper bound on the order of $\sqrt{md} + d$. Using matrix Bernstein's inequality given by [33, Theorem 5.44] instead, we can obtain an alternative upper bound $O(\sqrt{m} + d^{3/2})$. Both of these two upper bounds turn out to be loose. To this end, we develop a new matrix concentration inequality, proving a nearly-tight upper bound on the order of $\sqrt{m} + d$ up to logarithmic factors.

A key step in deriving a concentration inequality for matrices with sub-exponential random vectors is to obtain a large deviation inequality for the sum of independent random variables whose tails decay *slower* than sub-exponential random variables. Note that in this case, the moment generating function may not exist and thus we cannot follow the standard approach to obtain a large deviation inequality by invoking the Chernoff bound. To circumvent this, we partition the support of a real-valued random variable Y into countably many finite segments, and write Y as a summation of component random variables, each of which is supported on its corresponding segment. Due to the fact that each segment is of finite length, we can apply Bennett's inequality for bounded random variables (cf. Lemma A.1). Then we take a union bound to arrive at a concentration result of the original Y. Some additional care is needed in choosing the partition. Our proof is inspired by Proposition 2.1.9 and Excercise 2.1.7 in [32].

Lemma 4.2. *Let Y be a random variable whose tail probability satisfies*

$$\mathbb{P}\{|Y| \geq t\} \leq \exp(-E(t)),$$

where $E(t) : \mathbb{R}_+ \to [0, \infty]$ is a non-decreasing function. Suppose that

$$E(t)/t \text{ is monotone in } t, \tag{15}$$

and there exists $t_0 \geq e^2$ such that for all $t \geq t_0$ and all k with $4(k+1)^2 e^k \geq t$,

$$E(e^{k-1}) \geq 2(2k + 4\log(k+1) + \log 2 - \log t). \tag{16}$$

Let $Y_1, \cdots, Y_m$ be m independent copies of Y. If $E(t)/t$ is non-decreasing, then

$$\begin{aligned} &\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}[Y]\right| \geq mt\right\} \\ &\leq 2\log(mt)\exp\left(-\frac{m}{4(\log(mt)+1)^2} E\left(\frac{t}{4e\log^2 t}\right)\right) \\ &\quad + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right); \end{aligned} \tag{17}$$

if $E(t)/t$ is non-increasing, then

$$\begin{aligned} &\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}[Y]\right| \geq mt\right\} \\ &\leq 2\log(mt)\exp\left(-\frac{1}{4e(\log(mt)+1)^2} E\left(\frac{mt}{e}\right)\right) \\ &\quad + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right). \end{aligned} \tag{18}$$

Remark 9. *To illustrate the upper bound* (17)*, let us consider the following special cases.*

Case 1: Suppose Y is sub-Gaussian. In this case, $E(t) = ct^2$ for a universal constant $c > 0$. Thus, there exists a universal constant $t_0 \geq e^2$ such that both (16) *and* (15) *hold. Then* (17) *gives the desired sub-Gaussian tail bound $e^{-\Omega(mt^2)}$ up to logarithmic factors in the exponent.*

Case 2: Suppose Y is sub-exponential. In this case, $E(t) = ct$ for a universal constant c. Thus, there exists $t_0 \geq e^2$ that only depends on c such that both (16) *and* (15) *hold. Then* (17) *gives the desired sub-exponential tail bound $e^{-\Omega(mt)}$ up to logarithmic factors in the exponent.*

Case 3: Suppose $Y = Z^2$, where Z is sub-exponential. In this case, $E(t) = c\sqrt{t}$ for a universal constant $c > 0$. Thus, there exists $t_0 \geq e^2$ that only depends on c such that both (16) *and* (15) *hold. Then* (18) *gives a tail bound $e^{-\Omega(\sqrt{mt})}$ up to logarithmic factors in the exponent.*

Despite the fact that Lemma 4.2 is loose up to logarithmic factors compared to the standard sub-gaussian and sub-exponential random variables, Lemma 4.2 applies to much larger family than the sub-gaussian distributions, and requires much less structure on the distributions. In particular, Lemma 4.2 does not require the existence of moment generating function.

The proof of Lemma 4.2 can be found in Appendix A. Lemma 4.2 is our key machinery to obtain the concentration inequality for matrices with i.i.d. sub-exponential random vectors given in Theorem 3.2. We restate the theorem below for ease of reference.

Theorem (Theorem 3.2). *Let A be a $d\times m$ matrix whose columns A_j are independent and identically distributed sub-exponential, zero-mean random vectors in $\mathbb{R}^d$ with parameters (σ,α). Assume that*

$$\sigma/\alpha = \Omega(1). \tag{19}$$

Then with probability at least $1-\delta$,

$$\|A\|_2 \le c\left(\sigma\sqrt{m} + \sigma\phi\left(d+\log\frac{1}{\delta}\right) + \alpha\phi^2\left(d+\log\frac{1}{\delta}\right)\right),$$

where c is a universal positive constant and $\phi(x):\mathbb{R}\to\mathbb{R}$ is a function given by $\phi(x) = \sqrt{x}\log^{3/2}(x)$.

Remark 10. *We discuss two consequences of Theorem 3.2.*

Suppose $\alpha = 0$. In this case, A has sub-Gaussian columns, and Theorem 3.2 implies that

$$\|A\|_2 \;\lesssim\; \sigma\left(\sqrt{m} + \sqrt{d+\log\frac{1}{\delta}}\log^{3/2}\left(d+\log\frac{1}{\delta}\right)\right),$$

which matches the sub-Gaussian matrix concentration inequality [35, Theorem 5.39] up to logarithmic factors.

Suppose $\sigma,\alpha = \Theta(1)$, and $\log(1/\delta) = d$. In this case, we get that with probability at least $1-e^{-d}$,

$$\|A\|_2 \;\lesssim\; \sqrt{m} + d\log^3 d \qquad \textit{implied by Theorem 3.2}\,, \tag{20}$$

whereas the analogous bound implied by Theorem 4.1 is on the order of $\sqrt{md}+d$. Using matrix Bernstein's inequality given by [33, Theorem 5.44] instead, we can obtain an alternative upper bound $O(\sqrt{m}+d^{3/2})$. The upper bound (20) is tight up to logarithmic factors; see Appendix B for a proof.

The proof of Theorem 3.2 also uses the following standard concentration inequality for sum of independent sub-exponential random variables. In particular, we use this concentration inequality to get a concentration bound at θ^*.

Lemma 4.3. *[36, Proposition 2.2] Let $Y_1,\ldots,Y_m$ denote a sequence of independent random variables, where Y_j's are sub-exponential with scaling parameters (σ_j,α_j) and mean 0. Then $\sum_{j=1}^m Y_j$ is sub-exponential with scaling parameters (σ,α), where $\sigma^2 = \sum_{j=1}^m \sigma_j^2$ and $\alpha = \max_{1\le j\le m}\alpha_j$. Moreover,*

$$\mathbb{P}\left\{\sum_{j=1}^m Y_j \ge t\right\} \le \begin{cases}\exp\left(-\frac{t^2}{2\sigma^2}\right) & \textit{if } 0\le t\le \sigma^2/\alpha;\\ \exp\left(-\frac{t}{2\alpha}\right) & \textit{o.w.}\end{cases}$$

The following lemma gives an upper bound to the spectral norm of the covariance matrix of a sub-exponential random vector.

Lemma 4.4. *Let $Y\in\mathbb{R}^d$ denote a zero-mean, sub-exponential random vector with scaling parameters (σ,α), and Σ denote its covariance matrix $\Sigma = \mathbb{E}\left[YY^\top\right]$. Then*

$$\|\Sigma\|_2 \le 4\sigma^2 + 16\alpha^2.$$

Proof. First recall that

$$\|\Sigma\|_2 \;=\; \sup_{v\in S^{d-1}} v^\top\Sigma v = \sup_{v\in S^{d-1}} v^\top\mathbb{E}\left[YY^\top\right]v = \sup_{v\in S^{d-1}} \mathbb{E}\left[\langle Y,v\rangle^2\right].$$

For each unit vector v, from [35, Exercise 1.2.3], we have

$$\begin{aligned}\mathbb{E}\left[\langle Y,v\rangle^2\right] &= \int_0^\infty 2t\,\mathbb{P}\left\{|\langle Y,v\rangle|\geq t\right\}dt\\ &\leq \int_0^\infty 4t\ \exp\left(-\frac{1}{2}\min\left\{\frac{t^2}{\sigma^2},\frac{t}{\alpha}\right\}\right)dt\\ &\leq 4\sigma^2+16\alpha^2. \end{aligned}\tag{21}$$

Note that the above upper bound is independent of v. The lemma follows by combining the last two displayed equations. □

Now, we are ready to present the proof of Theorem 3.2.

Proof of Theorem 3.2. Recall $\Sigma = \mathbb{E}\left[A_1 A_1^\top\right]$. Then

$$\|A\|_2^2 = \left\|AA^\top\right\|_2 \leq \left\|AA^\top - m\Sigma\right\|_2 + m\,\|\Sigma\|_2\,.$$

In view of Lemma 4.4, we have $\|\Sigma\|_2 \leq 4\sigma^2 + 16\alpha^2$. It remains to bound $\|AA^\top - m\Sigma\|_2$. Note that

$$\begin{aligned}\left\|AA^\top - m\Sigma\right\|_2 &= \sup_{v\in S^{d-1}}\left|v^\top\left(AA^\top - m\Sigma\right)v\right|\\ &= \sup_{v\in S^{d-1}}\left|\sum_{j=1}^m\left(\left\langle A_j,v\right\rangle^2 - \mathbb{E}\left[\left\langle A_j,v\right\rangle^2\right]\right)\right|.\end{aligned}$$

Fix a $v\in S^{d-1}$. Note that $\left\langle A_j, v\right\rangle$ is zero-mean sub-exponential random variable with parameter (σ,α). For $j=1,\cdots,m$, define

$$Y_j = \left\langle A_j,v\right\rangle^2/\sigma^2. \tag{22}$$

It follows from Lemma 4.3 that

$$\mathbb{P}\left\{|Y_j|\geq t\right\} = \mathbb{P}\left\{\left|\left\langle A_j,v\right\rangle\right|\geq \sigma\sqrt{t}\right\} \leq 2\exp\left(-\min\left\{\frac{t}{2},\frac{\sigma\sqrt{t}}{2\alpha}\right\}\right),$$

We apply Lemma 4.2 to $Y_1,\cdots,Y_m$ with

$$E(t) = \min\left\{\frac{t}{2},\frac{\sigma\sqrt{t}}{2\alpha}\right\} - \log 2,$$

which is non-decreasing in t. By assumption $\sigma/\alpha = \Omega(1)$, it follows that $E(t)$ scales as $\sqrt{t}$ in t. Thus there exists $t_0\geq e^2$ such that (16) holds. In addition, $E(t)/t$ is non-increasing. Therefore, (18) in Lemma 4.2 applies, i.e., for all $t\geq t_0$,

$$\begin{aligned}&\mathbb{P}\left\{\left|\sum_{j=1}^m\left(Y_j - \mathbb{E}\left[Y_j\right]\right)\right|\geq mt\right\}\\ &\leq 2\log(mt)\exp\left(-\frac{1}{4e\log^2(emt)}E\left(\frac{mt}{e}\right)\right) + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right)\\ &\leq 4\log(mt)\exp\left(-\frac{1}{4e\log^2(emt)}E\left(\frac{mt}{e}\right)\right).\end{aligned}\tag{23}$$

Next, we apply ϵ-net argument. Let $\mathcal{N}_{\frac{1}{4}}$ be the $\frac{1}{4}$–net of the unit sphere S^{d-1}. From [33, Lemma 5.2], we know that $\left|\mathcal{N}_{\frac{1}{4}}\right| \le 9^d$. In addition, it follows from [33, Lemma 5.4] that

$$\left\|AA^\top - \Sigma\right\|_2 \le 2 \sup_{v\in\mathcal{N}_{\frac{1}{4}}} \left|\sum_{j=1}^m \left(\left\langle A_j, v\right\rangle^2 - \mathbb{E}\left[\left\langle A_j, v\right\rangle^2\right]\right)\right|.$$

Hence,

$$\begin{aligned}
&\mathbb{P}\left\{\left\|AA^\top - \Sigma\right\|_2 \ge 2\sigma^2 mt\right\}\\
&\le \mathbb{P}\left\{\sup_{v\in\mathcal{N}_{\frac{1}{4}}} \left|\sum_{j=1}^m \left(\left\langle A_j, v\right\rangle^2 - \mathbb{E}\left[\left\langle A_j, v\right\rangle^2\right]\right)\right| \ge \sigma^2 mt\right\}\\
&\le \left|\mathcal{N}_{\frac{1}{4}}\right| \mathbb{P}\left\{\left|\sum_{j=1}^m \left(\left\langle A_j, v\right\rangle^2 - \mathbb{E}\left[\left\langle A_j, v\right\rangle^2\right]\right)\right| \ge \sigma^2 mt\right\}\\
&\le 9^d \mathbb{P}\left\{\left|\sum_{j=1}^m \left(Y_j - \mathbb{E}\left[Y_j\right]\right)\right| \ge mt\right\} \text{ by definition of } Y_j \text{ in (22)}\\
&\le \exp\left(-\frac{1}{4e\log^2(emt)} E\left(\frac{mt}{e}\right) + \log 4 + \log\log(mt) + d\log 9\right),
\end{aligned}$$

where the last inequality holds by (23). To complete the proof, we need to choose mt so that the right hand side of the last inequality is smaller than δ. In other words, we need to find $x \ge mt_0$ such that

$$\frac{1}{4e\log^2(ex)} E(x/e) - \log\log x \ge \log\frac{4}{\delta} + d\log 9 \triangleq a.$$

One such x is given by

$$x = c\left(a\log^3 a + \frac{\alpha^2}{\sigma^2} a^2 \log^6 a + m\right),$$

where c is a sufficiently large constant. Therefore, we choose

$$\begin{aligned}
mt = c\Bigg(&\left(d + \log\frac{1}{\delta}\right)\log^3\left(d + \log\frac{1}{\delta}\right)\\
&+ \frac{\alpha^2}{\sigma^2}\left(d + \log\frac{1}{\delta}\right)^2 \log^6\left(d + \log\frac{1}{\delta}\right) + m\Bigg).
\end{aligned}$$

The theorem follows by taking the square root of mt. □

4.2 Proof of Theorem 3.3

With Lemma 3.4 and Theorem 3.2, we are ready to prove Theorem 3.3. Recall that we need to bound (13) and (14) uniformly for all $\theta \in \Theta$. Bounding (13) uniformly is relatively easy and has been done in previous work [8, Proposition 3.8].

Proposition 4.5. *[8, Proposition 3.8] Consider the same setup as Theorem 3.3. Assume that* $N = \Omega(d\log(Nd))$. *Then with probability at least* $1 - e^{-d}$,

$$\left\|\frac{1}{N}\sum_{i=1}^N \nabla f(X_i, \theta) - \nabla F(\theta)\right\|_2 \lesssim \Delta_2 \left\|\theta - \theta^*\right\|_2 + \Delta_1, \ \forall\, \theta \in \Theta,$$

where

$$\Delta_1 \triangleq \sqrt{\frac{d}{N}}, \quad \text{and} \quad \Delta_2 \triangleq \sqrt{\frac{d\log(Nd)}{N}}.$$

It remains to bound (14) uniformly over all $\theta \in \Theta$. For notational convenience, let

$$G(X_{\mathcal{S}},\theta) \triangleq \frac{1}{\sqrt{m}}\Big[\tfrac{1}{n}\textstyle\sum_{i\in\mathcal{S}_1}\nabla f(X_i,\theta) - \nabla F(\theta), \cdots, \\ \tfrac{1}{n}\textstyle\sum_{i\in\mathcal{S}_m}\nabla f(X_i,\theta) - \nabla F(\theta)\Big]. \tag{24}$$

Proposition 4.6. *Consider the same setup as Theorem 3.3. With probability at least* $1 - 2e^{-\sqrt{d}}$, *for all* $\theta \in \Theta$

$$\|G(X_{\mathcal{S}},\theta^*)\|_2 \lesssim \Delta_3 \tag{25}$$

$$\|G(X_{\mathcal{S}},\theta) - G(X_{\mathcal{S}},\theta^*)\|_2 \lesssim \Delta_4\|\theta-\theta^*\|_2 + \frac{1}{\sqrt{n}}, \tag{26}$$

where

$$\begin{aligned}\Delta_3 &\triangleq \frac{1}{\sqrt{n}} + \sqrt{\frac{d}{N}},\\ \Delta_4 &\triangleq \frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}}\phi\left(2d + d\log\left(1 + r\sqrt{n}(L+L')\right)\right)\\ &\quad + \frac{1}{\sqrt{Nn}}\phi^2\left(2d + d\log\left(1 + r\sqrt{n}(L+L')\right)\right),\end{aligned}$$

and $\phi(x) = \sqrt{x}\log^{3/2}(x)$. *It follows from triangle inequality that*

$$\|G(X_{\mathcal{S}},\theta)\|_2 \lesssim \Delta_4\|\theta-\theta^*\|_2 + \Delta_3, \ \forall\theta\in\Theta.$$

Remark 11. *The uniform upper bound* Δ_4 *in (26) depends linearly in d (ignoring logarithmic factors). Such linear dependency is inevitable as can be seen from the standard linear regression model given in Section 5. In this setting,* $\nabla f(X_i,\theta) = w_iw_i^\top(\theta-\theta^*) - w_i\zeta_i$ *and* $\nabla F(\theta) = \theta - \theta^*$, *where* $w_i \overset{i.i.d.}{\sim} \mathcal{N}(0,\mathbf{I})$ *and* $\zeta_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ *independent of* w_i*'s. For simplicity, assume* $n = 1$ *and* $m = N$. *Then*

$$\begin{aligned}&\sup_{\theta\in S^{d-1}} \|G(X_{\mathcal{S}},\theta) - G(X_{\mathcal{S}},\theta^*)\|_2\\ &\ge \sup_{\theta\in S^{d-1}} \frac{1}{\sqrt{N}}\left\|(w_1w_1^\top - \mathbf{I})(\theta-\theta^*)\right\|_2\\ &= \frac{1}{\sqrt{N}}\left(\|w_1\|_2^2 - 1\right)\|\theta-\theta^*\|_2\\ &= O_P\left(\frac{d}{\sqrt{N}}\right)\|\theta-\theta^*\|_2,\end{aligned}$$

where $O_P\left(\frac{d}{\sqrt{N}}\right)\|\theta-\theta^*\|_2$ *denotes* $O\left(\frac{d}{\sqrt{N}}\right)\|\theta-\theta^*\|_2$ *holds with high probability. The first equality follows by choosing* $\theta-\theta^*$ *parallel to* w_1, *and the last equality holds by the concentration of* χ^2 *distribution.*

Proof of Proposition 4.6. We prove the two bounds in (26) individually.

Bounding $\|G(X_{\mathcal{S}},\theta^*)\|_2$*:* It follows from Assumption 2 that the columns of $G(X_{\mathcal{S}},\theta^*)$ are i.i.d. sub-exponential random vectors in $\mathbb{R}^d$ with mean 0 and scaling parameters $\sigma_1/\sqrt{nm}$ and $\alpha_1/(n\sqrt{m})$, where σ_1 and α_1 are two absolute constants. Therefore, the columns of the scaled matrix $\sqrt{N}\,G(X_{\mathcal{S}},\theta^*)$ are i.i.d. sub-exponential random vectors $\mathbb{R}^d$ with mean 0 and scaling parameters σ_1 and $\alpha_1/\sqrt{n}$ – recalling that $N = nm$. Applying Theorem 4.1 to $A = \sqrt{N}\,G(X_{\mathcal{S}},\theta^*)$, we get that with probability at least $1 - e^{-\sqrt{d}}$,

$$\|G(X_{\mathcal{S}},\theta^*)\|_2 = \frac{1}{\sqrt{N}}\|A\|_2 \lesssim \frac{1}{\sqrt{N}}\left(\sqrt{m}+\sqrt{d}\right) = \frac{1}{\sqrt{n}} + \sqrt{\frac{d}{N}}. \tag{27}$$

Bounding $\|G(X_{\mathcal{S}},\theta) - G(X_{\mathcal{S}},\theta^*)\|_2$ *for a fixed* $\theta \in \Theta$*:* For notational convenience, define

$$\begin{aligned} H(X_{\mathcal{S}},\theta) &\triangleq G(X_{\mathcal{S}},\theta) - G(X_{\mathcal{S}},\theta^*) \\ &= \frac{1}{\sqrt{m}}\left[\frac{1}{n}\sum_{i\in\mathcal{S}_1} h(X_i,\theta), \cdots, \frac{1}{n}\sum_{i\in\mathcal{S}_m} h(X_i,\theta)\right], \end{aligned} \tag{28}$$

where recall from (4) that the gradient difference function $h(X,\cdot)$ is defined as

$$h(X,\theta) = \nabla f(X,\theta) - \nabla f(X,\theta^*) - (\nabla F(\theta) - \nabla F(\theta^*)).$$

It follows from Assumption 4 that the columns of $H(X_{\mathcal{S}},\theta)/\|\theta-\theta^*\|_2$ are i.i.d. sub-exponential random vectors in $\mathbb{R}^d$ with mean 0 and scaling parameters $\sigma_2/\sqrt{nm}$ and $\alpha_2/(n\sqrt{m})$, where σ_2 and α_2 are two absolute constants. Recall that $N = nm$. Applying Theorem 3.2 to $H(X_{\mathcal{S}},\theta)/\|\theta-\theta^*\|_2$, we know that for any fixed θ, with probability at least $1-\delta$,

$$\begin{aligned} &\|H(X_{\mathcal{S}},\theta)\|_2 \\ &\lesssim \left(\frac{\sigma_2}{\sqrt{n}} + \frac{\sigma_2}{\sqrt{N}}\phi\left(d+\log\frac{1}{\delta}\right) + \frac{\alpha_2}{\sqrt{Nn}}\phi^2\left(d+\log\frac{1}{\delta}\right)\right)\|\theta-\theta^*\|_2 \\ &\lesssim \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}}\phi\left(d+\log\frac{1}{\delta}\right) + \frac{1}{\sqrt{Nn}}\phi^2\left(d+\log\frac{1}{\delta}\right)\right)\|\theta-\theta^*\|_2, \end{aligned} \tag{29}$$

where we used $\phi(x) = \sqrt{x}\log^{3/2}(x)$, $\sigma_2 = O(1)$, and $\alpha_2 = O(1)$.

ϵ-net argument: We apply ϵ-net argument to extend the point convergence in (29) to the uniform convergence over Θ. In particular, let $\mathcal{N}_{\epsilon_0}$ be an ϵ_0-cover of $\Theta = \{\theta : \|\theta-\theta^*\|_2 \le r\}$ with

$$\epsilon_0 = \frac{1}{\sqrt{n}(L+L')}.$$

By [33, Lemma 5.2], we have

$$\log|\mathcal{N}_{\epsilon_0}| \le d\log(1+2r/\epsilon_0) = d\log\left(1+2r\sqrt{n}(L+L')\right).$$

By (29) and the union bound, we get that with probability at least $1-\delta$, for all $\theta \in \mathcal{N}_{\epsilon_0}$,

$$\begin{aligned} \|H(X_{\mathcal{S}},\theta)\|_2 \lesssim \|\theta-\theta^*\|_2 &\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}}\phi\left(d+\log\frac{|\mathcal{N}_{\epsilon_0}|}{\delta}\right)\right. \\ &\left.+ \frac{1}{\sqrt{Nn}}\phi^2\left(d+\log\frac{|\mathcal{N}_{\epsilon_0}|}{\delta}\right)\right). \end{aligned} \tag{30}$$

So far, we have shown the uniform convergence over net $\mathcal{N}_{\epsilon_0}$. Next, we extend this uniform convergence to the entire set Θ.

For any $\theta \in \Theta$, there exists a $\theta_k \in \mathcal{N}_{\epsilon_0}$ such that $\|\theta - \theta_k\|_2 \le \epsilon_0$. By triangle inequality,

$$\|H(X_{\mathcal{S}}, \theta)\|_2 \le \|H(X_{\mathcal{S}}, \theta_k)\|_2 + \|H(X_{\mathcal{S}}, \theta) - H(X_{\mathcal{S}}, \theta_k)\|_2 .$$

Note that

$$\begin{aligned} & \|H(X_{\mathcal{S}}, \theta) - H(X_{\mathcal{S}}, \theta_k)\|_2 \\ & \le \|H(X_{\mathcal{S}}, \theta) - H(X_{\mathcal{S}}, \theta_k)\|_{\mathrm{F}} \\ & \overset{(a)}{\le} \frac{1}{n} \max_{1 \le j \le m} \left\| \sum_{i \in \mathcal{S}_j} \left(h(X_i, \theta) - h(X_i, \theta_k)\right) \right\|_2 \\ & \overset{(b)}{\le} (L + L') \|\theta - \theta_k\|_2 \le (L + L')\epsilon_0 = \frac{1}{\sqrt{n}}, \end{aligned} \tag{31}$$

where (a) follows because the Frobenius norm $\|A\|_{\mathrm{F}}^2 = \sum_j \|A_j\|^2 \le m \max_j \|A_j\|^2$; (b) holds because

$$\begin{aligned} & \frac{1}{n} \left\| \sum_{i \in \mathcal{S}_j} \left(h(X_i, \theta) - h(X_i, \theta_k)\right) \right\|_2 \\ & \le \frac{1}{n} \sum_{i \in \mathcal{S}_j} \|h(X_i, \theta) - h(X_i, \theta_k)\|_2 \le (L + L') \|\theta - \theta_k\|_2 , \end{aligned}$$

in view of Assumption 1 and Assumption 3.

Combining (30) and (31), we have that with probability at least $1 - \delta$, for any $\theta \in \Theta$,

$$\begin{aligned} \|H(X_{\mathcal{S}}, \theta)\|_2 & \le \|H(X_{\mathcal{S}}, \theta_k)\|_2 + \frac{1}{\sqrt{n}} \\ & \lesssim \|\theta - \theta^*\|_2 \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}} \phi \left(d + \log \frac{|\mathcal{N}_{\epsilon_0}|}{\delta} \right) \right. \\ & \qquad \left. + \frac{1}{\sqrt{Nn}} \phi^2 \left(d + \log \frac{|\mathcal{N}_{\epsilon_0}|}{\delta} \right) \right) + \frac{1}{\sqrt{n}}. \end{aligned}$$

Choosing $\delta = e^{-d}$, we get that with probability at least $1 - e^{-d}$, for all $\theta \in \Theta$,

$$\begin{aligned} & \|H(X_{\mathcal{S}}, \theta)\|_2 \\ & \lesssim \|\theta - \theta^*\|_2 \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}} \phi \left(2d + d \log \left(1 + r\sqrt{n}(L + L') \right) \right) \right. \\ & \qquad \left. + \frac{1}{\sqrt{Nn}} \phi^2 \left(2d + d \log \left(1 + r\sqrt{n}(L + L') \right) \right) \right) + \frac{1}{\sqrt{n}}. \end{aligned} \tag{32}$$

Putting all pieces together. Combing (27) and (32), we conclude Proposition 4.6.

□

Finish the proof of Theorem 3.3:

Let $\mathcal{E}_1$ and $\mathcal{E}_2$ denote the two events on which the conclusions in Proposition 4.5 and Proposition 4.6 hold, respectively. It follows from Proposition 4.5 and Proposition 4.6 that $\mathbb{P}\{\mathcal{E}_1 \cap \mathcal{E}_2\} \ge 1 - 3e^{-\sqrt{d}}$.

Recall that we use Algorithm 2 as our robust gradient aggregator $\mathcal{R}$ with input $\widehat{y}_j(\theta)$ given by the local gradient function $g_j(\theta)$ at worker j. We will apply Lemma 3.4 with $y_j(\theta) = \frac{1}{n} \sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta)$ as per (11). Then the true mean $\mu = \nabla F(\theta)$ and the true sample mean $\mu_{\mathcal{S}} = \frac{1}{N} \sum_{i=1}^{N} \nabla f(X_i, \theta)$ as per (12).

On event $\mathcal{E}_1 \cap \mathcal{E}_2$, in view of Remark 8, for each $\theta \in \Theta$, condition (9) in Lemma 3.4 is satisfied with

$$\sigma = (\Delta_4 + \Delta_2) \left\| \theta - \theta^* \right\|_2 + \Delta_1 + \Delta_3, \tag{33}$$

where Δ_i's are given in Proposition 4.5 and Proposition 4.6.

Therefore, in view of Remark 7, it follows from Lemma 3.4 that for each $\theta \in \Theta$, the output $G(\theta)$ of the gradient aggregator $\mathcal{R}$ satisfies

$$\begin{aligned} \| G(\theta) - \nabla F(\theta) \|_2 &\lesssim \sqrt{\frac{q}{m}} \left[(\Delta_4 + \Delta_2) \| \theta - \theta^* \|_2 + \Delta_1 + \Delta_3 \right] \\ &\quad + \Delta_2 \| \theta - \theta^* \|_2 + \Delta_1 \\ &\lesssim \left(\sqrt{\frac{q}{m}} \Delta_4 + \Delta_2 \right) \| \theta - \theta^* \|_2 + \sqrt{\frac{q}{m}} \Delta_3 + \Delta_1. \end{aligned}$$

Recall that $\phi(x) = \sqrt{x} \log^{3/2}(x)$, $\log(L + L') = O(\log(Nd))$, $\Theta \subset \{\theta : \|\theta - \theta^*\|_2 \le r\}$ for some positive parameter r such that $\log r = O(\log(Nd))$, and that $N = \Omega(d^2 \log^8(Nd))$. Then,

$$\begin{aligned} \Delta_4 &= \frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}} \phi\left(2d + d \log\left(1 + r\sqrt{n}(L + L') \right) \right) \\ &\quad + \frac{1}{\sqrt{Nn}} \phi^2 \left(2d + d \log\left(1 + r\sqrt{n}(L + L') \right) \right) \\ &\overset{(a)}{\lesssim} \frac{1}{\sqrt{n}} + \sqrt{\frac{d}{N}} \log^2(Nd) + \frac{1}{\sqrt{n}} \lesssim \frac{1}{\sqrt{n}} + \sqrt{\frac{d}{N}} \log^2(Nd), \end{aligned}$$

where in (a) we used the assumption $N = \Omega(d^2 \log^8(Nd))$.

Combining the last two displayed equations together with the expressions of $\Delta_1, \Delta_2, \Delta_3$ in Proposition 4.5 and Proposition 4.6, we get that for each $\theta \in \Theta$,

$$\begin{aligned} &\| G(\theta) - \nabla F(\theta) \|_2 \\ &\lesssim \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{qd}{mN}} \log^2(Nd) + \sqrt{\frac{d \log(Nd)}{N}} \right) \| \theta - \theta^* \|_2 \\ &\quad + \sqrt{\frac{q}{N}} + \sqrt{\frac{qd}{mN}} + \sqrt{\frac{d}{N}} \\ &\lesssim \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}} \log^2(Nd) \right) \| \theta - \theta^* \|_2 + \sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}, \end{aligned}$$

completing the proof of Theorem 3.3.

4.3 Proof of Theorem 3.1

Proof. From Theorem 3.3, we know that there exists a constant c_0 such that with probability at least $1 - 3e^{-\sqrt{d}}$, for all $\theta \in \Theta$,

$$\begin{aligned} &\| G(\theta) - \nabla F(\theta) \|_2 \\ &\le c_0 \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}} \log^2(Nd) \right) \| \theta - \theta^* \|_2 + c_0 \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}} \right). \end{aligned} \tag{34}$$

Thus, with probability at least $1 - 3e^{-\sqrt{d}}$, we have that for every $t \geq 1$,

$$\begin{aligned}\|\theta_t - \theta^*\|_2 &= \|\theta_{t-1} - \eta G(\theta_{t-1}) - \theta^*\|_2 \\ &= \|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^* + \eta \left(\nabla F(\theta_{t-1}) - G(\theta_{t-1})\right)\|_2 \\ &\leq \|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^*\|_2 + \eta \left\|\left(\nabla F(\theta_{t-1}) - G(\theta_{t-1})\right)\right\|_2 \\ &\leq \sqrt{1 - \frac{M^2}{4L^2}} \|\theta_{t-1} - \theta^*\|_2 + \eta \left\|\left(\nabla F(\theta_{t-1}) - G(\theta_{t-1})\right)\right\|_2 \\ &\leq \rho \|\theta_{t-1} - \theta^*\|_2 + c_0 \frac{M}{2L^2}\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right), \end{aligned} \tag{35}$$

where the second inequality follows from the standard convergence analysis of perfect gradient descent, see, e.g., [8, Lemma 3.2]; the last inequality follows from (34), $\eta = M/(2L^2)$, and

$$\rho \triangleq \sqrt{1 - \frac{M^2}{4L^2}} + c_0 \frac{M}{2L^2}\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}} \log^2(Nd)\right).$$

Then applying a standard telescoping argument to (35) yields that

$$\|\theta_t - \theta^*\|_2 \leq \rho^t \|\theta_0 - \theta^*\|_2 + \frac{c_0 M}{2L^2(1-\rho)}\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right). \tag{36}$$

There exists a constant c such that if $N \geq cd^2 \log^8(Nd)$ and $N \geq cq$, then

$$c_0\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}} \log^2(Nd)\right) \leq \frac{1}{8}.$$

Consequently,

$$\rho \leq \sqrt{1 - \frac{M^2}{4L^2}} + \frac{M}{16L^2} \leq 1 - \frac{M^2}{8L^2} + \frac{M}{16L^2} \leq 1 - \frac{M^2}{16L^2},$$

where the last inequality follows from the assumption that $M \geq 1$. Combining the last displayed equation with (36) yields that

$$\|\theta_t - \theta^*\|_2 \leq \left(1 - \frac{M^2}{16L^2}\right)^t \|\theta_0 - \theta^*\|_2 + 8c_0\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right),$$

completing the proof of Theorem 3.1.

□

5 APPLICATIONS TO LINEAR REGRESSION AND LOGISTIC REGRESSION

In this section, we illustrate our general results by applying them to the classical linear regression and logistic regression problems.

5.1 Application to Linear Regression

Let $X_i = (w_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ denote the input data and define the risk function $f(X_i, \theta) = \frac{1}{2}\left(\langle w_i, \theta\rangle - y_i\right)^2$. For simplicity, we assume that y_i is indeed generated from a linear model:

$$y_i = \langle w_i, \theta^*\rangle + \zeta_i,$$

where θ^* is an unknown true model parameter, $w_i \sim N(0, \mathbf{I})$ is the covariate vector whose covariance matrix is assumed to be identity, and $\zeta_i \sim N(0, 1)$ is i.i.d. additive Gaussian noise independent of

w_i's. Intuitively, the inner product $\langle w_i, \theta^* \rangle$ can be viewed as a linear “measurement" of θ^* – the signal; and ζ_i is the additive noise.

The population risk function F is given by

$$F(\theta) \triangleq \mathbb{E}\left[f(X,\theta)\right] = \mathbb{E}\left[\frac{1}{2}\left(\langle w,\theta\rangle - y\right)^2\right]$$
$$= \mathbb{E}\left[\frac{1}{2}\left(\langle w,\theta\rangle - \langle w,\theta^*\rangle - \zeta\right)^2\right] = \frac{1}{2}\|\theta-\theta^*\|_2^2 + \frac{1}{2},$$

for which θ^* is indeed the unique minimum. The population gradient function is $\nabla_\theta F(\theta) = \theta - \theta^*$. It is easy to see that the population risk function F is L-Lipschitz continuous with $L = 1$, and M-strongly convex with $M = 1$. Hence, Assumption 1 is satisfied with $M = L = 1$; and the stepsize $\eta = M/(2L^2) = 1/2$.

For a given random sample $X = (w, y)$, the associated random gradient is given by

$$\nabla f(X,\theta) = w\left(\langle w,\theta\rangle - y\right) = w\langle w, \theta-\theta^*\rangle - w\zeta,$$

where $w \sim \mathcal{N}(0, \mathbf{I})$ and $\zeta \sim \mathcal{N}(0,1)$ that is independent of w.

The following lemma verifies that Assumption 2–Assumption 4 are satisfied with appropriate parameters.

Lemma 5.1. *Under the linear regression model, the sample gradient function* $\nabla f(X,\cdot)$ *satisfies*
(1) Assumption 2 with $\sigma_1 = \sqrt{2}$ *and* $\alpha_1 = \sqrt{2}$*,*
(2) and Assumption 4 with $\sigma_2 = \sqrt{8}$ *and* $\alpha_2 = 8$.
Moreover, with probability $1 - e^{-d}$*, Assumption 3 holds with* $L' = 3d + 2\log N + 2\sqrt{d(d+\log N)}$ *for all* $\{\nabla f(X_i,\cdot)\}_{i=1}^N$.

Proof. Claims (1) and (2) have been proved in Lemma 4.1 [8]. It remains to prove the last claim. Under the linear regression model

$$\|\nabla f(X,\theta) - \nabla f(X,\theta')\|_2 = \left\|ww^\top(\theta-\theta')\right\|_2 \le \|w\|_2^2\,\|\theta-\theta'\|_2\,.$$

Hence, it suffices to show

$$\mathbb{P}\left\{\max_{i=1}^N \|w_i\|_2^2 \ge 3d + 2\log N + 2\sqrt{d(d+\log N)}\right\} \le e^{-d}. \tag{37}$$

Note that for each $i \in [N]$, $\|w_i\|_2^2 \sim \chi^2(d)$. Using the following tail bound:

$$\mathbb{P}\left\{\chi^2(d) \ge d + 2\sqrt{dt} + 2t\right\} \le e^{-t}$$

with $t = d + \log N$, we get that

$$\mathbb{P}\left\{\|w_i\|_2^2 \ge 3d + 2\log N + 2\sqrt{d(d+\log N)}\right\} \le \frac{1}{N}e^{-d}.$$

Then the desired (37) follows by the union bound. □

As an immediate corollary of Theorem 3.1, our Byzantine-resilient Gradient Descent method can robustly solve the linear regression problem exponentially fast with high probability.

Corollary 5.2 (Linear regression). *Under the aforementioned least-squares model for linear regression, assume* $\Theta \subset \{\theta : \|\theta - \theta^*\| \le r\}$ *for* $r > 0$ *such that* $\log r = O(\log(Nd))$. *Suppose that* $N \ge cd^2\log^8(Nd))$ *and* $N \ge cq$ *for a sufficiently large constant c, and that* $m \le e^{\sqrt{d}}$. *Then with* *probability at least* $1 - 3e^{-\sqrt{d}}$*, the iterates* $\{\theta_t\}$ *given by Algorithm 1 with Robust Gradient Aggregator Algorithm 2 satisify*

$$\|\theta_t - \theta^*\|_2 \lesssim \left(\frac{15}{16}\right)^t \|\theta_0 - \theta^*\|_2 + \sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}, \ \forall t \geq 0.$$

5.2 Application to Logistic Regression

Here we consider the binary logistic regression problem [14, Section 4.4], where we assume that y_i is generated as follows:

$$y_i \overset{\text{i.i.d.}}{\sim} \text{Bern}\left(\frac{1}{1 + e^{-w_i^\top \theta^*}}\right),$$

where $\theta^* \in \mathbb{R}^d$ is an unknown true model parameter, $w_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ is the observed feature vector, and $y_i \in \{0, 1\}$ is the observed class label. Intuitively, logistic regression tries to model the log likelihood ratio

$$\log \frac{\mathbb{P}\{y_i = 1 | w_i\}}{\mathbb{P}\{y_i = 0 | w_i\}}$$

as a linear function $w_i^\top \theta$ of w_i. Let $X_i = (w_i, y_1) \in \mathbb{R}^d \times \{0, 1\}$ denote the input data and define the risk function $f(X_i, \theta)$ as the negative log likelihood function

$$\begin{aligned} f(X_i, \theta) &= -\log \mathbb{P}\{y_i \mid w_i;\ \theta\} \\ &= -\mathbf{1}_{\{y_i = 1\}} \log \frac{1}{1 + e^{-w_i^\top \theta}} - \mathbf{1}_{\{y_i = 0\}} \log \frac{e^{-w_i^\top \theta}}{1 + e^{-w_i^\top \theta}} \\ &= (1 - y_i) w_i^\top \theta + \log\left(1 + e^{-w_i^\top \theta}\right). \end{aligned}$$

It follows that the population risk function F is given by

$$\begin{aligned} F(\theta) &\triangleq \mathbb{E}\left[f(X, \theta)\right] \\ &= \mathbb{E}_{w \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)} \left[\frac{e^{-w^\top \theta^*}}{1 + e^{-w^\top \theta^*}} w^\top \theta + \log\left(1 + e^{-w^\top \theta}\right)\right]. \end{aligned}$$

The population gradient function is given by

$$\nabla_\theta F(\theta) = \mathbb{E}_{w \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)} \left[w\left(\frac{e^{-w^\top \theta^*}}{1 + e^{-w^\top \theta^*}} - \frac{e^{-w^\top \theta}}{1 + e^{-w^\top \theta}}\right)\right].$$

The population Hessian function is given by

$$\nabla_\theta^2 F(\theta) = \mathbb{E}_{w \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)} \left[\frac{w w^\top}{1 + e^{w^\top \theta}}\right].$$

It can be seen that the population Hessian function is strictly positive definite and hence $F(\theta)$ is strictly convex for which θ^* is indeed the unique minimum. Moreover, for a given random sample $X = (w, y)$, the associated random gradient is given by

$$\nabla f(X, \theta) = w\left[(1 - y) - \frac{e^{-w^\top \theta}}{1 + e^{-w^\top \theta}}\right].$$

When σ is small and θ is restricted within a ball of small radius, then $w^\top \theta \sim \mathcal{N}(0, \sigma^2 \|\theta\|_2^2)$ is typically small. In this case, we can approximate $e^{-w^\top \theta}/(1 + e^{-w^\top \theta})$ by its first-order Taylor series

$1/2 - w^\top \theta/4$. As a consequence,

$$\nabla f(X,\theta) \approx w\left[\frac{1}{2} - y + \frac{1}{4}w^\top \theta\right],$$

which resembles the gradient vector in the simple linear regression.

6 SUMMARY AND FUTURE DIRECTIONS

This present paper intersects two main areas of research: fault-tolerant distributed computing and statistical machine learning. In particular, we consider a machine learning scenario where a model is trained in a distributed but unsecured environment. Armed with a robust mean estimation primitive, we secure the gradient descent method against adversarial interruptions, even in high dimensions. Our secured gradient descent converges to the true model parameter exponentially fast up to an estimation error $O(\sqrt{q/N} + \sqrt{d/N})$ – matching the minimax-optimal error rate in the failure-free setting as long as the number of faulty workers $q = O(d)$. A key ingredient in our analysis is a uniform concentration of the sample covariance matrix of gradient functions.

There are many interesting future directions to explore, and we list a few as follows.

- We have shown the optimal error rate is $O(\sqrt{d/N})$ when $q = O(d)$. However, the optimal error rate remains elusive when $q \gg d \gg 1$.
- The present paper assumes the population risk function $F(\theta)$ is convex. In many contemporary machine learning applications, the population risk function is often non-convex. It would be interesting to extend our results to the non-convex setting. A crucial question is how to escape saddle points with robustly aggregated gradients. This direction has been recently pursued in [39].
- It would be interesting to see how the choice of robust mean estimation building block affects the performance of the stochastic optimization algorithm in terms of computation, estimation error, probability error, etc.
- Note that in this work, we consider full gradient descent under which each worker computes the local gradient based on the *entire* local sample (all n data points). Since n is small, the computational burdens of the workers are reasonable. It has been demonstrated numerically in [18] that in the adversary-free setting, there is a performance improvement when each worker performs a few epochs of SGD before the model updates are aggregated. Whether there will be similar performance improvement in our adversary-prone setting is unclear.
- So far, we consider synchronous distributed systems, wherein the learner communicates with the workers in synchronous communication rounds. It would be interesting to see how asynchrony affects the learning performance.
- We assume each worker reports the entire gradient vector in each round. Some applications may call for even more communication-efficient algorithms. It would be interesting to see if, rather than the entire gradient vector, it suffices for each worker to report partial gradient vector in each round.

A PROOF OF LEMMA 4.2

We first quote a classical concentration inequality for sum of independent, bounded random variables.

Lemma A.1 (Bennett's inequality). *Let $Y_1, \cdots, Y_m$ be independent random variables. Assume that $\left|Y_j - \mathbb{E}\left[Y_j\right]\right| \le B$ almost surely for every j. Then for any $t > 0$, we have*

$$\mathbb{P}\left\{\sum_{j=1}^{m}(Y_j - \mathbb{E}\left[Y_j\right]) \ge t\right\} \le \exp\left(-\frac{\sigma^2}{B^2} \cdot h\left(\frac{Bt}{\sigma^2}\right)\right),$$

where $\sigma^2 = \sum_{j=1}^{m} \mathrm{var}(Y_j)$ is the variance of the sum, and

$$h(u) = (1+u)\log(1+u) - u.$$

Proof of Lemma 4.2. We use the idea of truncation. In this proof, we adopt the convention that $\frac{1}{0} = +\infty$.

For each copy $j = 1, \cdots, m$, we partition Y_j into countably many pieces as follows: Let

$$\begin{aligned} Y_{j,0} &= Y_j \mathbf{1}_{\{|Y_j| \le 1\}} \\ Y_{j,k} &= Y_j \mathbf{1}_{\{e^{k-1} \le |Y_j| \le e^k\}}, \text{ for } k = 1, 2, \ldots \end{aligned}$$

It is easy to see that

$$Y_j = \sum_{k=0}^{\infty} Y_{j,k}, \text{ for } j = 1, \cdots, m.$$

Let $S = \sum_{j=1}^{m} Y_j$. We have

$$S = \sum_{j=1}^{m} Y_j = \sum_{j=1}^{m}\left(\sum_{k=0}^{\infty} Y_{j,k}\right) = \sum_{k=0}^{\infty}\sum_{j=1}^{m} Y_{j,k} = \sum_{k=0}^{\infty} S_k,$$

where $S_k \triangleq \sum_{j=1}^{m} Y_{j,k}$, for $k = 0, 1, \cdots$. Thus,

$$\begin{aligned} \mathbb{P}\left\{\left|\sum_{j=1}^{m} Y_j - m\mathbb{E}\left[Y\right]\right| > mt\right\} &= \mathbb{P}\left\{|S - \mathbb{E}\left[S\right]| > mt\right\} \\ &= \mathbb{P}\left\{\left|\sum_{k=0}^{\infty}(S_k - \mathbb{E}\left[S_k\right])\right| > mt\right\}. \end{aligned}$$

To bound $\mathbb{P}\left\{\left|\sum_{j=1}^{m} Y_j - m\mathbb{E}\left[Y\right]\right| > mt\right\}$ for a given t, our plan is to find a sequence of t_k (which depends on t) such that

$$\{|S - \mathbb{E}\left[S\right]| > mt\} \subseteq \cup_{k=0}^{\infty}\{|S_k - \mathbb{E}\left[S_k\right]| > mt_k\}, \tag{38}$$

and

$$\mathbb{P}\{|S_k - \mathbb{E}\left[S_k\right]| > mt_k\}$$

is small enough to apply the union bound over all k.

In this proof, we choose $t_k = \frac{t}{2(k+1)^2}$ for $k = 0, 1, \cdots$. It is easy to see that (38) holds.

Next, we bound $\mathbb{P}\{|S_k - \mathbb{E}\left[S_k\right]| > mt_k\}$ for each k. For given $t \ge t_0$, define

$$k_0 \triangleq \inf\left\{k \in \mathbb{Z} : 4e^k(k+1)^2 \ge t\right\}. \tag{39}$$

We are particularly interested in the setting when $t \ge t_0 \ge e^2$, which implies that

$$1 \le k_0 \le \log t - 1, \tag{40}$$

noting that $4e^{\log t - 1}(\log t - 1 + 1)^2 \ge t$.

Case 1: $0 \le k \le k_0 - 1$. It is easy to see that when $t \ge t_0 \ge e^2$, $k_0 \ge 1$. Thus, case 1 is well posed. As per the definition of (39), for all $0 \le k \le k_0 - 1$, it holds that $4e^k(k+1)^2 < t$. That is,

$$2e^k < \frac{t}{2(k+1)^2} = t_k. \tag{41}$$

On the other hand, by construction of $Y_{j,k}$ we have deterministically

$$\left|Y_{j,k} - \mathbb{E}\left[Y_{j,k}\right]\right| \le 2e^k, \text{ for all } k. \tag{42}$$

Thus

$$\begin{aligned}|S_k - \mathbb{E}[S_k]| &= \left|\sum_{j=1}^m Y_{j,k} - \mathbb{E}\left[\sum_{j=1}^m Y_{j,k}\right]\right| \\ &\le \sum_{j=1}^m \left|Y_{j,k} - \mathbb{E}\left[Y_{j,k}\right]\right| \le 2me^k \text{ for all } k,\end{aligned}$$

i.e.,

$$\mathbb{P}\left\{|S_k - \mathbb{E}[S_k]| > 2me^k\right\} = 0 \text{ for all } k.$$

By (41), we have that when $0 \le k \le k_0 - 1$,

$$\mathbb{P}\{|S_k - \mathbb{E}[S_k]| > mt_k\} \le \mathbb{P}\left\{|S_k - \mathbb{E}[S_k]| > 2me^k\right\} = 0. \tag{43}$$

Case 2: $k_0 \le k \le \log(mt)$. For each k in this range, we will apply Bennett's inequality given in Lemma A.1.

From (42), we know that for any fixed k, the random variable $\left|Y_{j,k} - \mathbb{E}\left[Y_{j,k}\right]\right| \le 2e^k$. The variance of $Y_{j,k}$ can be bounded as follows: for $k \ge 1$

$$\begin{aligned}\mathrm{var}(Y_{j,k}) &\le \mathbb{E}\left[Y_{j,k}^2\right] \\ &\le e^{2k}\mathbb{P}\left\{\left|Y_j\right| \ge e^{k-1}\right\} \\ &\le e^{2k}\exp\left(-E\left(e^{k-1}\right)\right).\end{aligned} \tag{44}$$

For notational convenience, define

$$\sigma_k^2 \triangleq e^{2k}\exp\left(-E(e^{k-1})\right). \tag{45}$$

To see that σ_k^2 is well-defined, recall that we adopt the convention that $\frac{1}{0} = \infty$ and $\exp(-\infty) = 0$.

For each k in this case, i.e., $k_0 \le k \le \log(mt)$, by Lemma A.1, we get that

$$\begin{aligned}&\mathbb{P}\{|S_k - \mathbb{E}[S_k]| \ge mt_k\} \\ &= \mathbb{P}\left\{\left|\sum_{j=1}^m (Y_{j,k} - \mathbb{E}\left[Y_{j,k}\right])\right| \ge mt_k\right\} \\ &\le 2\exp\left(-\frac{\sum_{j=1}^m \mathrm{var}(Y_{j,k})}{e^{2(k+1)}} \cdot h\left(\frac{e^{(k+1)}mt_k}{\sum_{j=1}^m \mathrm{var}(Y_{j,k})}\right)\right),\end{aligned}$$

Note that when $u > 0$, it holds that $h(u) \geq u \log(u/e)$, so we have that

$$\begin{aligned}
&\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\
&\leq 2\exp\left(-\frac{\sum_{j=1}^m \mathsf{var}(Y_{j,k})}{e^{2(k+1)}} \cdot \frac{e^{(k+1)}mt_k}{\sum_{j=1}^m \mathsf{var}(Y_{j,k})} \log\left(\frac{e^{(k+1)}mt_k}{e\sum_{j=1}^m \mathsf{var}(Y_{j,k})}\right)\right) \\
&= 2\exp\left(-\frac{mt_k}{e^{(k+1)}} \log\left(\frac{e^k mt_k}{\sum_{j=1}^m \mathsf{var}(Y_{j,k})}\right)\right) \\
&\leq 2\exp\left(-\frac{mt_k}{e^{(k+1)}} \log\left(\frac{e^k t_k}{\sigma_k^2}\right)\right),
\end{aligned} \tag{46}$$

where the last inequality follows from the fact that $\sum_{j=1}^m \mathsf{var}(Y_{j,k}) \leq m\sigma_k^2$. We proceed to bound $\log\left(\frac{e^k t_k}{\sigma_k^2}\right)$ using the assumption (16):

$$\begin{aligned}
\log\left(\frac{e^k t_k}{\sigma_k^2}\right) &= \log\left(\frac{e^k t}{2(k+1)^2 e^{2k} \exp\left(-E\left(e^{k-1}\right)\right)}\right) \\
&= \log\left(\frac{t}{2(k+1)^2 e^{k} \exp\left(-E\left(e^{k-1}\right)\right)}\right) \\
&= \log t - \left(\log 2 + 2\log(k+1) + k - E\left(e^{k-1}\right)\right) \\
&= E(e^{k-1}) - (\log 2 + 2\log(k+1) + k - \log t) \\
&\overset{(a)}{\geq} \frac{1}{2}E(e^{k-1}) + (2k + 4\log(k+1) + \log 2 - \log t) \\
&\quad - (\log 2 + 2\log(k+1) + k - \log t) \\
&\geq \frac{1}{2}E(e^{k-1}) + 2\log(k+1) + k,
\end{aligned} \tag{47}$$

where inequality (a) holds due to the assumption (16). Combining the last displayed equation with (46) yields

$$\begin{aligned}
&\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\
&\leq 2\exp\left(-\frac{mt_k}{2e^{(k+1)}} E(e^{k-1})\right) \\
&= 2\exp\left(-\frac{mt}{4(k+1)^2 e^{(k+1)}} E(e^{k-1})\right) \\
&\leq 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 e^{(k+1)}} E(e^{k-1})\right),
\end{aligned} \tag{48}$$

where the last inequality holds because in the case under consideration, $k_0 \leq k \leq \log(mt)$. To proceed, we use the monotonicity assumption of $E(t)/t$. If $E(t)/t$ is non-decreasing (increasing), we can bound (48) as

$$\begin{aligned}
&\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\
&\overset{(a)}{\leq} 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 e^{(k_0+1)}} E(e^{k_0-1})\right) \\
&\overset{(b)}{\leq} 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 t} E\left(\frac{t}{4e(k_0+1)^2}\right)\right) \\
&\overset{(c)}{\leq} 2\exp\left(-\frac{m}{4(\log(mt)+1)^2} E\left(\frac{t}{4e\log^2 t}\right)\right),
\end{aligned} \tag{49}$$

where (a) holds because $k_0 \leq k \leq \log(mt)$; (b) holds because $k_0 \leq \log t - 1$, $4e^{k_0}(k_0+1)^2 \geq t$, and that $E(\cdot)$ is non-decreasing; (c) follows from $k_0 \leq \log t - 1$, and that $E(\cdot)$ is non-decreasing.

If $E(t)/t$ is non-increasing, we can bound (48) as

$$\begin{aligned}
&\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\
&\leq 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 e^{\log(mt)+1}} E(e^{\log(mt)-1})\right) \\
&= 2\exp\left(-\frac{1}{4e(\log(mt)+1)^2} E\left(\frac{mt}{e}\right)\right).
\end{aligned} \tag{50}$$

Case 3: $k \geq \log(mt)$. In this case, we use the Chebyshev's inequality:

$$\begin{aligned}
\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \leq \frac{\sigma_k^2}{t_k} &= \exp\left(-\log\frac{t_k}{\sigma_k^2}\right) \\
&\overset{(a)}{\leq} \frac{1}{(k+1)^2}\exp\left(-\frac{1}{2}E(e^{k-1})\right) \\
&\leq \frac{1}{(k+1)^2}\exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right),
\end{aligned} \tag{51}$$

where (a) follows from (47); the last inequality follows from the fact that $E(u)$ is increasing (non-decreasing) in u.

For a fix t, summing over all $k \in \mathbb{N}$, we have

$$\begin{aligned}
&\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} \\
&\leq \sum_{k=0}^{\infty} \mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\
&= \sum_{k=0}^{k_0-1} \mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} + \sum_{k=k_0}^{\lfloor\log(mt)\rfloor} \mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\
&\quad + \sum_{\lceil\log(mt)\rceil}^{\infty} \mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\
&\leq 0 + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right) + \sum_{k=k_0}^{\lfloor\log(mt)\rfloor} \mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\}.
\end{aligned}$$

Therefore, we have that if $E(t)/t$ is non-decreasing,

$$\begin{aligned}&\mathbb{P}\left\{\left|\sum_{j=1}^{m} Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\}\\ &\leq 2\log(mt)\exp\left(-\frac{m}{4(\log(mt)+1)^2}E\left(\frac{t}{4e\log^2 t}\right)\right)\\ &\quad+\exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right);\end{aligned}$$

if $E(t)/t$ is non-increasing,

$$\begin{aligned}&\mathbb{P}\left\{\left|\sum_{j=1}^{m} Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\}\\ &\leq 2\log(mt)\exp\left(-\frac{1}{4e(\log(mt)+1)^2}E\left(\frac{mt}{e}\right)\right)\\ &\quad+\exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right).\end{aligned}$$

□

B TIGHTNESS OF UPPER BOUND (20)

To see the upper bound (20) is tight up to logarithmic factors, consider an example, where A_j's are i.i.d. isotropic Laplace distribution with the density function given by $f(x) = \prod_{i=1}^{d}\left[(1/\sqrt{2})\exp\left(-\sqrt{2}x_i\right)\right]$ for $x \in \mathbb{R}^d$. In this case, note that

$$\left\{\|A\|_2 \geq \max\{\sqrt{m/2}, d\}\right\} \supseteq \left\{|A_{11}| \geq d \text{ and } \sum_{j=1}^{m} A_{2j}^2 \geq m/2\right\}.$$

Since

$$\mathbb{P}\left\{|A_{11}| \geq d\right\} = \int_{|t|\geq d}\frac{1}{\sqrt{2}}\exp\left(-\sqrt{2}t\right)\mathrm{d}t = \exp\left(-\sqrt{2}d\right),$$

and by Chebyshev's inequality,

$$\mathbb{P}\left\{\sum_{j=1}^{m} A_{2j}^2 \geq m/2\right\} \geq 1 - O(1/m) \geq \frac{1}{2}$$

for m suffciently large, and A_{11} is independent of $\sum_{j=1}^{m} A_{2j}^2$, it follows that

$$\begin{aligned}\mathbb{P}\left\{\|A\|_2 \geq \max\{\sqrt{m/2}, d\}\right\} &\geq \mathbb{P}\left\{|A_{11}| \geq d \text{ and } \sum_{j=1}^{m} A_{2j}^2 \geq m/2\right\}\\ &\geq \frac{1}{2}\exp\left(-\sqrt{2}d\right).\end{aligned}$$

C ROBUST MEAN ESTIMATION

Robust gradient aggregation is closely related to robust mean estimation, formally stated next.

Definition C.1 (Robust mean estimation). Let $\mathcal{S} = \{y_1, \cdots, y_m\}$ be a sample of size m, wherein each of the data point y_i is generated independently from an unknown distribution. Among those m data points, up to $q = \epsilon m$ of them may be adversarially corrupted. Let $\{\widehat{y}_1, \cdots, \widehat{y}_m\}$ be the observed sample. The goal is to estimate the true mean of the unknown distribution when only corrupted sample $\{\widehat{y}_1, \cdots, \widehat{y}_m\}$ is accessible.

The adversarially corrupted data may affect the mean estimation in the following two ways: (i) extreme magnitudes and/or (ii) extreme directions. The adversarial magnitudes are relatively easy to "detected" and removed. For instance, a simple trimming/pruning procedure may suffice [9, Section 4.3.1]. Dealing with adversarially extreme directions is more challenging.

If the true sample mean/center were known, then those adversarially extreme directions would be "identified" by finding the eigenvectors corresponding to large eigenvalues of the sample covariance matrix; hence the corrupted data points would be filtered away by projecting along these extremre directions. However, the true sample mean/center is unknown in reality. It turns out that we can approximate the center by representing each data point by sufficiently many other data points evenly, as per Step 2 of Algorithm 2.

To gain some intuition on how it works, let us first consider the ideal setting where the data sample is corruption-free, i.e., all the data points are generated from the same underlying distribution. If the spectral norm of the sample covariance matrix is bounded, then we expect these data points are well concentrated around the sample mean and hence "similar" to each other.

When up to an ϵ fraction of the data sample is adversarially corrupted, as long as ϵ is small enough, there still exists a large collection of uncorrupted data points that are close to the true sample mean and similar to each other. Thus, each of them can be approximately represented as a convex combination of sufficiently many other data points so that the convex coefficients are approximately uniform over this collection of data. Hence, the corrupted data is more responsible for the approximation error of representation. Thus, the direction which maximizes the approximation error of representation is likely to the adversarially extreme direction.

The Step 2 of Algorithm 2 works precisely along this idea. In particular, (6) aims to find good center approximation through representation, while (7) aims to find the extreme direction. By solving (6) and (7) for a saddle point (W, U), Algorithm 2 iteratively finds a direction (given by U^*) along which all data points are spread out the most, and filters away data points which have large residual errors projected along this direction (given by (8)).

For completeness, next we present the proof of Lemma 3.4– the robustness guarantee of Algorithm 2.

For ease of exposition, in the sequel, we let

$$\alpha \triangleq 1 - \epsilon \quad \text{and} \quad \widetilde{\sigma}^2 = 2\sigma^2.$$

We first need a minimax identity between the min-max problem (6) and max-min problem (7). For $W \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$ and $U \in \mathbb{R}^{d \times d}$, recall the function $\psi : (W, U) \to \mathbb{R}$ defined as:

$$\psi(W, U) = \sum_{i \in \mathcal{A}} c_i \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right)^{\top} U \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right).$$

Also, let $\mathcal{W}$ denote the set of all column stochastic matrices $W \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$ such that $0 \le W_{ji} \le \frac{4-\alpha}{\alpha(2+\alpha)m}$, and $\mathcal{U}$ denote the set of all positive semi-definite matrices $U \in \mathbb{R}^{d \times d}$ such that $\mathsf{Tr}(U) \le 1$. Then the min-max program (6) can be rewritten as

$$\begin{aligned} W^* &\in \arg\min_{W \in \mathcal{W}} \max_{U \in \mathcal{U}} \psi(W, U) \\ &= \arg\min_{W \in \mathcal{W}} \left\| \sum_{i \in \mathcal{A}} c_i \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right) \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right)^{\top} \right\| \end{aligned} \tag{52}$$

and the max-min program (7) can be rewritten as

$$U^* \in \arg\max_{U \in \mathcal{U}} \min_{W \in \mathcal{W}} \psi(W, U).$$

Note that $\psi(W, U)$ is convex in W for a fixed U and concave (in fact linear) in U for a fixed W. By von Neumann's minimax theorem, we have

$$\min_{W \in \mathcal{W}} \max_{U \in \mathcal{U}} \psi(W, U) = \max_{U \in \mathcal{U}} \min_{W \in \mathcal{W}} \psi(W, U) = \psi(W^*, U^*).$$

Moreover, (W^*, U^*) is a saddle point, i.e.,

$$W^* \in \arg\min_{W \in \mathcal{W}} \psi(W, U^*), \tag{53}$$

$$U^* \in \arg\max_{U \in \mathcal{U}} \psi(W^*, U). \tag{54}$$

The saddle point properties (53) and (54) are crucial to prove Lemma 3.4.

Moreover, by condition (9), the underlying true sample $\mathcal{S}$ (of size m) satisfies the following condition:

$$\left\| \frac{1}{m} \sum_{i=1}^{m} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\|_2 \le \sigma^2,$$

where $\mu_{\mathcal{S}} = \frac{1}{m} \sum_{i=1}^{m} y_i$. Recall that up to q points in $\mathcal{S}$ are corrupted. Let $\mathcal{S}_0 \subseteq \mathcal{S}$ be a subset of uncorrupted subset of $\mathcal{S}$ of size $m - q = (1 - \epsilon)m = \alpha m$. Notably, since q is only an upper bound on the number of corrupted data points, the choice of subset $\mathcal{S}_0$ may not be unique. Nevertheless, for any choice of subset $\mathcal{S}_0$, the following holds:

$$\begin{aligned}
&\left\| \frac{1}{|\mathcal{S}_0|} \sum_{i \in \mathcal{S}_0} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\|_2 \\
&= \frac{1}{|\mathcal{S}_0|} \left\| \sum_{i \in \mathcal{S}_0} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\|_2 \\
&\le \frac{1}{|\mathcal{S}_0|} \left\| \sum_{i=1}^{m} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\|_2 \\
&\le \frac{1}{\alpha} \sigma^2 \le 2\sigma^2,
\end{aligned} \tag{55}$$

where the last inequality follows because by assumption, $\alpha = 1 - \epsilon \ge \frac{3}{4} \ge \frac{1}{2}$.

As commented in Subsection 3.2, Algorithm 2 terminates within m iterations. For ease of exposition, we use $t = 1, 2, \cdots$ to denote the iteration number. We use $c_i(t)$, $\tau_i(t)$, and $\mathcal{A}(t)$ to denote the quantities of interest at iteration t. Note that weights c_i and set $\mathcal{A}$ may be updated throughout an iteration. Therefore, we use $\mathcal{A}'(t)$ and $c_i'(t)$ to denote the updated quantities at the end of iteration t. Note that $c_i'(t-1) = c_i(t)$ and $\mathcal{A}'(t-1) = \mathcal{A}(t)$.

C.1 Two auxiliary lemmas

We first show that when Algorithm 2 terminates, most of data points in $\mathcal{S}_0$ are remained in $\mathcal{A}$.

Lemma C.2. *For every iteration $t \geq 1$ in the* **while***–loop of Algorithm 2,*

$$\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} c_i(t)\tau_i(t) \;\leq\; \alpha m\widetilde{\sigma}^2 \tag{56}$$

$$\sum_{i\in\mathcal{S}_0}(1-c_i(t)) \leq \frac{\alpha}{4}\sum_{i=1}^{m}(1-c_i(t)) \tag{57}$$

$$|\mathcal{S}_0\cap\mathcal{A}(t)| \geq \frac{\alpha(2+\alpha)m}{4-\alpha}. \tag{58}$$

Intuitively, Lemma C.2 says that in every iteration: (1) the summation of the projected residual error over the non-corrupted data is small; (2) the weights of non-corrupted data points are reduced by a relatively small amount; (3) and more importantly, most non-corrupted data points are not removed.

Proof of Lemma C.2. The proof is by induction on (57) and (58). Note that the induction hypotheses do not include (56). Recall that we use $t = 1, \cdots$ to denote the iteration number in the **while**–loop.

Base case: $t = 1$. Note that $\mathcal{A}(1) = [m]$, and $c_i(1) = 1$ for all $i \in \mathcal{A}(1)$. Therefore, (57) and (58) hold for $t = 1$ trivially.

Induction Step: Suppose (57) and (58) hold for t, and the **while**– has not terminate at iteration t. We aim to show (57) and (58) hold for $t+1$.

We first prove (56) holds for t. Recall that

$$\tau_i(t) = \left(y_i - \sum_{j\in\mathcal{A}(t)}\widehat{y}_j W_{ji}(t)\right)^{\top} U(t)\left(y_i - \sum_{j\in\mathcal{A}(t)}\widehat{y}_j W_{ji}(t)\right),$$

where $W(t)$ is a minimizer to (6) and $U(t)$ is a maximizer to (7) at iteration t, respectively. Since $(W(t), U(t))$ is a saddle point, it follows from (53) that $W(t) \in \arg\min_{W\in\mathcal{W}} \psi(W, U(t))$. Moreover, this minimization is decoupled over all data points in $\mathcal{A}(t)$ and hence each column of $W(t)$ is optimized independently. Therefore, by letting $W_{*i}(t)$ denote the column of $W(t)$ corresponding to $i \in \mathcal{A}(t)$, we have

$$\begin{aligned} W_{*i}(t) \in \arg\min_{w} & \left(y_i - \sum_{j\in\mathcal{A}(t)}\widehat{y}_j w_j\right)^{\top} U(t)\left(y_i - \sum_{j\in\mathcal{A}(t)}\widehat{y}_j w_j\right)\\ \text{s. t.} & \sum_{j\in\mathcal{A}(t)} w_j = 1\\ & 0 \leq w_j \leq \frac{4-\alpha}{\alpha(2+\alpha)m}. \end{aligned} \tag{59}$$

Let $\widetilde{w} \in \mathbb{R}^{|\mathcal{A}(t)|}$ be the column stochastic vector such that

$$\widetilde{w}_j \triangleq \frac{\mathbf{1}_{\{j\in\mathcal{S}_0\cap\mathcal{A}(t)\}}}{|\mathcal{S}_0\cap\mathcal{A}(t)|},\ \forall\, j\in\mathcal{A}(t).$$

By the induction hypothesis, $\widetilde{w}$ is feasible to (59). Let $Y_{\mathcal{A}(t)} \in \mathbb{R}^{d\times n}$ be the matrix with $\widehat{y}_i$ with $i \in \mathcal{A}(t)$ as columns. Moreover,

$$Y_{\mathcal{A}(t)}\widetilde{w} = \sum_{j\in\mathcal{A}(t)}\widehat{y}_j\widetilde{w}_j = \frac{1}{|\mathcal{S}_0\cap\mathcal{A}(t)|}\sum_{j\in\mathcal{S}_0\cap\mathcal{A}(t)} y_j \;\triangleq\; \mu_{\mathcal{S}_0\cap\mathcal{A}(t)}.$$

Thus, we have

$$\begin{aligned}
&\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} c_i(t)\tau_i(t)\\
&\overset{(a)}{\le} \sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} c_i(t)(y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)})^\top U(t)\left(y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)}\right)\\
&\overset{(b)}{\le} \sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} (y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)})^\top U(t)\left(y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)}\right)\\
&\overset{(c)}{\le} \sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} (y_i-\mu_{\mathcal{S}})^\top U(t)\left(y_i-\mu_{\mathcal{S}}\right)\\
&\le \sum_{i\in\mathcal{S}_0} (y_i-\mu_{\mathcal{S}})^\top U(t)\left(y_i-\mu_{\mathcal{S}}\right)\\
&\overset{(d)}{\le} \mathsf{Tr}\left(U(t)\right)\left\|\sum_{i\in\mathcal{S}_0}(y_i-\mu_{\mathcal{S}})(y_i-\mu_{\mathcal{S}})^\top\right\|_2\\
&\overset{(e)}{\le} \alpha m\widetilde{\sigma}^2,
\end{aligned}$$

where (a) holds by the optimality of $W_{*i}(t)$ to (59); (b) holds because $c_i(t)\le 1$ and $U(t)\succeq 0$; (c) holds because

$$\mu_{\mathcal{S}_0\cap\mathcal{A}(t)}=\frac{1}{|\mathcal{S}_0\cap\mathcal{A}(t)|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} y_i$$

is a minimizer of the quadratic form

$$\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} (y_i-u)^\top U(t)\left(y_i-u\right),$$

as a function of u; (d) holds because $|\langle A,B\rangle|\le\|A\|_2\,\|B\|_*$, where $\|B\|_*$ is the sum of singlular values of B and $\|B\|_*=\mathsf{Tr}(B)$ when $B\succeq 0$; (e) follows by (55) and the facts that $|\mathcal{S}_0|\le\alpha m$ and $\mathsf{Tr}(U(t))=1$.

Next we prove (57) and (58). Since by induction hypothesis the **while**–loop has not terminate at iteration t, it follows that

$$\sum_{i\in\mathcal{A}(t)} c_i(t)\tau_i(t)>4m\widetilde{\sigma}^2. \tag{60}$$

Note that the weights of the data points that do not lie in $\mathcal{A}(t)$ are not updated in iteration t, i.e., $c_i'(t)=c_i(t)$ for $i\notin\mathcal{A}(t)$. As a consequence, we have

$$\begin{aligned}
&\sum_{i\in\mathcal{S}_0}(1-c_i'(t))\\
&=\sum_{i\in\mathcal{S}_0}(1-c_i(t))+\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)}(c_i(t)-c_i'(t))\\
&\le\frac{\alpha}{4}\sum_{i=1}^{m}(1-c_i(t))+\frac{1}{\tau_{\max}(t)}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)}\tau_i(t)c_i(t),
\end{aligned}\tag{61}$$

where the last inequality follows from induction hypothesis. Furthermore, we have

$$\frac{1}{\tau_{\max}(t)}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)}\tau_i(t)c_i(t)\overset{(a)}{\le}\frac{1}{\tau_{\max}(t)}\alpha m\widetilde{\sigma}^2$$
$$\overset{(b)}{<}\frac{\alpha}{4\tau_{\max}(t)}\sum_{i\in\mathcal{A}(t)}\tau_i(t)c_i(t),$$

where (a) holds because we have shown that (56) holds for t; (b) follows from (60).

Thus, (61) can be further bounded as

$$\begin{aligned}
&\sum_{i\in\mathcal{S}_0}(1-c_i'(t))\\
&\le\frac{\alpha}{4}\sum_{i=1}^{m}(1-c_i(t))+\frac{\alpha}{4\tau_{\max}(t)}\sum_{i\in\mathcal{A}(t)}\tau_i(t)c_i(t)\\
&=\frac{\alpha}{4}\left(\sum_{i\notin\mathcal{A}(t)}(1-c_i(t))+\sum_{i\in\mathcal{A}(t)}(1-c_i(t))+\frac{1}{\tau_{\max}(t)}\sum_{i\in\mathcal{A}(t)}\tau_i(t)c_i(t)\right)\\
&=\frac{\alpha}{4}\left(\sum_{i\notin\mathcal{A}(t)}(1-c_i'(t))+\sum_{i\in\mathcal{A}(t)}\left(1-\left(1-\frac{\tau_i(t)}{\tau_{\max}(t)}\right)c_i(t)\right)\right)\\
&=\frac{\alpha}{4}\sum_{i=1}^{m}(1-c_i'(t)),
\end{aligned}$$

proving (57) for $t+1$. We rewrite (57) for $t+1$ as

$$\sum_{i\in\mathcal{S}_0}(1-c_i'(t))\le\frac{\alpha}{4-\alpha}\sum_{i\notin\mathcal{S}_0}(1-c_i'(t))\,.$$

One the one hand, we have

$$\sum_{i\notin\mathcal{S}_0}(1-c_i'(t))\le|\mathcal{S}_0^c|\le(1-\alpha)m.$$

On the other hand,

$$\sum_{i\in\mathcal{S}_0}(1-c_i'(t))\ge\sum_{i\in\mathcal{S}_0\setminus\mathcal{A}'(t)}(1-c_i'(t))\ge\frac{1}{2}\left|\mathcal{S}_0\setminus\mathcal{A}'(t)\right|,$$

where the last inequality holds from the fact that $c_i'(t)\le 1/2$ for all $i\notin\mathcal{A}'(t)$ – by the data removal criterion in Algorithm 2. Combining the last three displayed equations, we get that

$$|\mathcal{S}_0\setminus\mathcal{A}'(t)|\le\frac{2\alpha(1-\alpha)}{4-\alpha}m,$$

proving (57) for $t+1$. The proof of Lemma C.2 is complete. □

Let W be the minimizer of (6) when the **while**–loop terminates. Let W_1 be the result of zeroing out all singular values of W that are greater than 0.9.

LEMMA C.3. *The matrix $W_0=(W-W_1)(I-W_1)^{-1}$ is a column stochastic matrix, and the rank of the weight matrix W_0 is one.*

Remark 12. *Let $X_{\mathcal{A}} \subseteq \mathbb{R}^{d\times|\mathcal{A}|}$ be the data matrix with columns being the data points in $\mathcal{A}$. Let $Z = X_{\mathcal{A}} W_0$. Since W_0 is rank one, all the $|\mathcal{A}|$ columns in the matrix Z are identical. Denote*

$$Z = [\widetilde{\mu}, \cdots, \widetilde{\mu}] \,. \tag{62}$$

Then $\widetilde{\mu}$ is a weighted average of the points in $\mathcal{A}$.

Proof. We first show that W_0 is a column stochastic matrix:

$$\begin{aligned} \mathbf{1}^\top W_0 &= \mathbf{1}^\top (W - W_1)(I - W_1)^{-1} \overset{(a)}{=} (\mathbf{1}^\top - \mathbf{1}^\top W_1)(I - W_1)^{-1} \\ &= \mathbf{1}^\top (I - W_1)(I - W_1)^{-1} = \mathbf{1}^\top, \end{aligned}$$

where (a) follows because W is column stochastic.

Next we show that rank of W_0 is one. From (6), we know that $\|W\|_F^2 \le \frac{4-\alpha}{\alpha(2+\alpha)}$. To see this,

$$\begin{aligned} \|W\|_F^2 &= \sum_{i,j\in\mathcal{A}} W_{ji}^2 \\ &\le \sum_{i,j\in\mathcal{A}} \left(W_{ji} \cdot \max_{i,j\in\mathcal{A}} W_{ji} \right) \\ &\le \left(\sum_{i,j\in\mathcal{A}} W_{ji} \right) \frac{4-\alpha}{\alpha(2+\alpha)m} \\ &\le \frac{4-\alpha}{\alpha(2+\alpha)}. \end{aligned}$$

When $\alpha \ge \frac{3}{4}$,

$$\frac{4-\alpha}{\alpha(2+\alpha)} \le \frac{52}{33} < 2 \times 0.9^2.$$

Hence, at most one singular value of W can be greater than 0.9. Moreover, since W is column stochastic, its largest singular value is at least 1. Thus, $W - W_1$ is of rank one. As a consequence, W_0 is of rank one. □

C.2 Proof of Lemma 3.4

Proof. Recall that our goal is to show

$$\|\mu_{\mathcal{S}} - \widehat{\mu}\|_2 = O(\sigma\sqrt{1-\alpha}),$$

where $\widehat{\mu} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \widehat{y}_i$ is the algorithm output. Recall $Y_{\mathcal{A}} \subseteq \mathbb{R}^{d\times|\mathcal{A}|}$ is the data matrix with columns being the data points in $\mathcal{A}$. In view of Remark 12, columns of $Z = Y_{\mathcal{A}} W_0$ are identical and denoted by $\widetilde{\mu}$. Our proof is divided into two steps:

Step 1: We first show that points in $\mathcal{A}$ are clustered around the center $\widetilde{\mu}$. In addition, by (58) in Lemma C.2, the set $\mathcal{A}$ mainly consists of uncorrupted data. As a consequence, we are able to show that

$$\widehat{\mu} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \widehat{y}_i \approx \frac{1}{|\mathcal{S}_0 \cap \mathcal{A}|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} \widehat{y}_i = \frac{1}{|\mathcal{S}_0 \cap \mathcal{A}|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} y_i. \tag{63}$$

Step 2: By (55), points in $\mathcal{S}_0$ are clustered around the center $\mu_{\mathcal{S}}$. In addition, by (58) in Lemma C.2, most of the points in $\mathcal{S}_0$ have been preserved. Thus we are able to show that

$$\mu_{\mathcal{S}} = \frac{1}{m}\sum_{i=1}^{m} y_i \approx \frac{1}{|\mathcal{S}_0 \cap \mathcal{A}|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} y_i. \tag{64}$$

Putting these two pieces together, the proof of Lemma 3.4 is complete.

Step 1: We show (63).

When the **while**–loop terminates, in view of (52), we have

$$\left\|Y_{\mathcal{A}}(I-W)\mathrm{diag}\left\{(c_{\mathcal{A}})^{\frac{1}{2}}\right\}\right\|_2 \le 2\sqrt{m}\widetilde{\sigma}., \tag{65}$$

where $\mathrm{diag}\left\{(c_{\mathcal{A}})^{\frac{1}{2}}\right\}$ is the diagonal matrix with diagonal entries given by $\{c_i^{1/2}\}_{i\in\mathcal{A}}$. We will show that $\widehat{y}_i \approx \widetilde{\mu}$ for all $i \in \mathcal{A}$. For this purpose, it is enough to show $\|Y_{\mathcal{A}} - Z\|_2$ is small:

$$\begin{aligned}
\left\|Y_{\mathcal{A}} - \widetilde{\mu}\mathbf{1}^T\right\|_2 &= \|Y_{\mathcal{A}} - Z\|_2 \\
&= \|Y_{\mathcal{A}} - Y_{\mathcal{A}}W_0\|_2 \\
&= \left\|Y_{\mathcal{A}}(I-W_1)(I-W_1)^{-1} - Y_{\mathcal{A}}(W-W_1)(I-W_1)^{-1}\right\|_2 \\
&= \left\|Y_{\mathcal{A}}(I-W)(I-W_1)^{-1}\right\|_2 \\
&\le \|Y_{\mathcal{A}}(I-W)\|_2 \left\|(I-W_1)^{-1}\right\|_2 \\
&\overset{(a)}{\le} \|Y_{\mathcal{A}}(I-W)\|_2 \times 10 \\
&\overset{(b)}{\le} 10\sqrt{2}\left\|Y_{\mathcal{A}}(I-W)\mathrm{diag}\left\{(c_{\mathcal{A}})^{\frac{1}{2}}\right\}\right\|_2 \\
&\overset{(c)}{\le} 20\sqrt{2m}\widetilde{\sigma},
\end{aligned}$$

where (a) holds because the largest singular value of W_1 is at most 0.9; (b) holds because $c_i \ge \frac{1}{2}$ for all $i \in \mathcal{A}$; (c) follows from (65).

Fix any $0 < \epsilon' < 1/2$. Let $\mathcal{T} \subseteq \mathcal{A}$ such that $|\mathcal{T}| \ge (1-\epsilon')|\mathcal{A}|$. We have

$$\begin{aligned}
&\left\|\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\widehat{y}_i - \widehat{\mu}\right\|_2 \\
&= \left\|\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\widehat{y}_i - \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\widehat{y}_i\right\|_2 \\
&= \left\|\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}(\widehat{y}_i - \widetilde{\mu}) - \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}(\widehat{y}_i - \widetilde{\mu})\right\|_2 \\
&= \left\|\left(\frac{1}{|\mathcal{T}|} - \frac{1}{|\mathcal{A}|}\right)\sum_{i\in\mathcal{T}}(\widehat{y}_i - \widetilde{\mu}) - \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}/\mathcal{T}}(\widehat{y}_i - \widetilde{\mu})\right\|_2 \\
&\overset{(a)}{\le} \frac{|\mathcal{A}| - |\mathcal{T}|}{|\mathcal{T}|\,|\mathcal{A}|}\|[Y_{\mathcal{A}} - Z]_{\mathcal{T}}\,\mathbf{1}\|_2 + \frac{1}{|\mathcal{A}|}\left\|[Y_{\mathcal{A}} - Z]_{\mathcal{A}/\mathcal{T}}\,\mathbf{1}\right\|_2 \\
&= \left(\frac{|\mathcal{A}| - |\mathcal{T}|}{\sqrt{|\mathcal{T}|}\,|\mathcal{A}|} + \frac{\sqrt{|\mathcal{A}/\mathcal{T}|}}{|\mathcal{A}|}\right)\|Y_{\mathcal{A}} - Z\|_2 \\
&\le 80\sqrt{2}\widetilde{\sigma}\sqrt{\epsilon'},
\end{aligned} \tag{66}$$

where $[Y_{\mathcal{A}} - Z]_{\mathcal{T}}$ denotes the submatrix of $Y_{\mathcal{A}} - Z$ – restricting to columns in $\mathcal{T}$, and $\mathbf{1} \in \mathbb{R}^{|\mathcal{T}|}$; the last inequality holds because $\epsilon' < 1/2$ and

$$|\mathcal{A}| \ge |\mathcal{A} \cap \mathcal{S}_0| \ge \frac{\alpha(2+\alpha)}{4-\alpha}m.$$

Note that

$$\frac{\alpha(2+\alpha)}{4-\alpha} \geq 1 - \frac{5}{3}(1-\alpha) \Leftrightarrow (\alpha-1)^2 \geq 0.$$

Thus, $|\mathcal{A} - \mathcal{A} \cap \mathcal{S}_0| \leq \frac{5}{3}(1-\alpha)m$. Choosing $\mathcal{T} = \mathcal{A} \cap \mathcal{S}_0$, we obtain

$$\|\mu_{\mathcal{S}_0 \cap \mathcal{A}} - \widehat{\mu}\|_2 \leq 80\sqrt{2}\widetilde{\sigma}\sqrt{5(1-\alpha)/3} \leq 160\widetilde{\sigma}\sqrt{1-\alpha} = O(\widetilde{\sigma}\sqrt{1-\alpha}). \tag{67}$$

Step 2: We show (64)**.** The proof of (64) is similar to that of (63).

Recall that $\mu_{\mathcal{S}} = \frac{1}{m}\sum_{i=1}^m y_i$ and that

$$\mu_{\mathcal{S}_0 \cap \mathcal{A}} = \frac{1}{|\mathcal{S}_0 \cap \mathcal{A}|} \sum_{i \in \mathcal{S}_0 \cap \mathcal{A}} y_i.$$

We have

$$\begin{aligned} \|\mu_{\mathcal{S}} - \mu_{\mathcal{S}_0 \cap \mathcal{A}}\|_2 &= \left\| \mu_{\mathcal{S}} - \frac{1}{|\mathcal{A} \cap \mathcal{S}_0|} \sum_{i \in \mathcal{S}_0 \cap \mathcal{A}} y_i \right\|_2 \\ &= \left\| \frac{1}{|\mathcal{A} \cap \mathcal{S}_0|} \sum_{i \in \mathcal{A} \cap \mathcal{S}_0} (y_i - \mu_{\mathcal{S}}) \right\|_2 \\ &= \frac{1}{|\mathcal{A} \cap \mathcal{S}_0|} \|[Y_{\mathcal{A} \cap \mathcal{S}_0} - \mu_{\mathcal{S}}]\mathbf{1}\|_2 \\ &\leq \frac{\sqrt{|\mathcal{S}_0|}}{\sqrt{|\mathcal{A} \cap \mathcal{S}_0|}} \widetilde{\sigma} \\ &\leq \sqrt{\frac{4-\alpha}{\alpha(2+\alpha)}} \sqrt{1-\alpha}\widetilde{\sigma} \leq \sqrt{2(1-\alpha)}\widetilde{\sigma}. \end{aligned}$$

□

C.3 Alternative Termination Condition of Algorithm 2

Recall that the termination of Algorithm 2 relies on the knowledge of σ. Since σ depends on $\|\theta - \theta^*\|_2$ according to (33) in our robust gradient aggregation setting, the learner needs to know a priori $\|\theta - \theta^*\|_2$ for all θ, which may not be possible. Nevertheless, it turns out that the termination condition of Algorithm 2 can be replaced by checking the cardinality of set $|\mathcal{A}|$, formally stated as follows:

If

$$\left| \mathcal{A} \setminus \left\{ i : \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i \leq \frac{1}{2} \right\} \right| \geq \frac{\alpha(2+\alpha)m}{4-\alpha},$$

we update $c_i \leftarrow \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i$ and remove $\left\{ i : \ c_i \leq \frac{1}{2} \right\}$ from $\mathcal{A}$; otherwise, we break the **while**–loop.

Similar to the original Algorithm 2, in the modified Algorithm 2, in each iteration of the **while**–loop at least one point will be removed. Thus, the modified Algorithm 2 terminates in at most m iterations. We next prove the conclusion of Lemma 3.4 still holds after this modification.

Suppose the modified Algorithm 2 terminates at iteration t^*. By the modified code we know $|\mathcal{A}(t^*)| \geq \frac{\alpha(2+\alpha)m}{4-\alpha}$; otherwise, the algorithm terminates earlier than t^*. By the termination condition, we also know that

$$\left| \mathcal{A}(t^*) - \left\{ i : \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i \leq \frac{1}{2} \right\} \right| < \frac{\alpha(2+\alpha)m}{4-\alpha}. \tag{68}$$

Claim 1. *There exists an iteration $t' \le t^*$ such that*

$$\sum_{i \in \mathcal{A}(t')} c_i(t')\tau_i(t') \le 8m\sigma^2.$$

Proof. We prove by contradiction. Suppose

$$\sum_{i \in \mathcal{A}(t)} c_i(t)\tau_i(t) > 8m\sigma^2, \quad \forall t \le t^*. \tag{69}$$

Note that the modified Algorithm 2 and the original Algorithm 2 differ only in their termination conditions. Recall that the original termination condition is only used in the proof of Lemma C.2 to conclude that (60) holds when the **while**-loop does not terminate. Thus, under the hypothesis (given in the last displayed equation), Lemma C.2 still holds. It follows that

$$\begin{aligned}
&\left|\mathcal{A}(t^*) - \left\{i : \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i \le \frac{1}{2}\right\}\right| \\
&\ge \left|\mathcal{S}_0 \cap \left(\mathcal{A}(t^*) - \left\{i : \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i \le \frac{1}{2}\right\}\right)\right| \\
&\ge \frac{\alpha(2+\alpha)m}{4-\alpha},
\end{aligned}$$

which leads to a contradiction. □

Since $\mathcal{A}(t)$ is monotone decreasing, it follows that $\mathcal{A}(t^*) \subseteq \mathcal{A}(t')$. Moreover,

$$|\mathcal{A}(t^*)| \ge \frac{\alpha(2+\alpha)m}{4-\alpha} \ge \frac{\alpha(2+\alpha)}{4-\alpha}\,|\mathcal{A}(t')| \ge \left(1 - \frac{5}{3}(1-\alpha)\right)|\mathcal{A}(t')|.$$

By (66), we know

$$\begin{aligned}
&\left\|\frac{1}{|\mathcal{A}(t^*)|}\sum_{i \in \mathcal{A}(t^*)} \widehat{y}_i - \frac{1}{|\mathcal{A}(t')|}\sum_{i \in \mathcal{A}(t')} \widehat{y}_i\right\|_2 \\
&\le 80\sqrt{2}\widetilde{\sigma}\sqrt{\frac{5}{3}(1-\alpha)} = O(\sigma\sqrt{1-\alpha}).
\end{aligned}$$

From Lemma 3.4, we know

$$\left\|\frac{1}{|\mathcal{A}(t')|}\sum_{i \in \mathcal{A}(t')} \widehat{y}_i - \mu_{\mathcal{S}}\right\|_2 = O(\sigma\sqrt{1-\alpha}).$$

Combining the last two displayed equations, we have

$$\left\|\frac{1}{|\mathcal{A}(t^*)|}\sum_{i \in \mathcal{A}(t^*)} \widehat{y}_i - \mu_{\mathcal{S}}\right\|_2 = O(\sigma\sqrt{1-\alpha}).$$

ACKNOWLEDGMENTS

L. Su was supported in part by the NSF Science & Technology Center for Science of Information Grant CCF-0939370. J. Xu was supported in part by the NSF Grant CCF-1755960.

REFERENCES

[1] Radosław Adamczak, Alexander Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. 2010. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *Journal of the American Mathematical Society* 23, 2 (2010), 535–561.

[2] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. 2018. Byzantine Stochastic Gradient Descent. *arXiv preprint arXiv:1803.08917* (2018).

[3] Dimitri P Bertsekas and Athena Scientific. 2015. *Convex optimization algorithms.* Athena Scientific Belmont.

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Byzantine-Tolerant Machine Learning. *arXiv preprint arXiv:1703.02757* (2017).

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.

[6] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization.* Cambridge university press.

[7] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. 2017. Learning from Untrusted Data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017).* ACM, New York, NY, USA, 47–60. https://doi.org/10.1145/3055399.3055491

[8] Yudong Chen, Lili Su, and Jiaming Xu. 2017. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 2, Article 44 (Dec. 2017), 25 pages. https://doi.org/10.1145/3154503

[9] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. 2016. Robust Estimators in High Dimensions without the Computational Intractability. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS).* 655–664. https://doi.org/10.1109/FOCS.2016.85

[10] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. 2017. Being Robust (in High Dimensions) Can Be Practical. *CoRR* abs/1703.00893 (2017). arXiv:1703.00893 http://arxiv.org/abs/1703.00893

[11] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. 2018. Sever: A Robust Meta-Algorithm for Stochastic Optimization. *arXiv preprint arXiv:1803.02815* (2018).

[12] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2014. Privacy Aware Learning. *J. ACM* 61, 6, Article 38 (Dec. 2014), 57 pages. https://doi.org/10.1145/2666468

[13] Jiashi Feng, Huan Xu, and Shie Mannor. 2014. Distributed Robust Learning. *arXiv preprint arXiv:1409.5937* (2014).

[14] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics.

[15] Peter J Huber. 2011. Robust statistics. In *International Encyclopedia of Statistical Science.* Springer, 1248–1251.

[16] Adam Klivans, Pravesh K Kothari, and Raghu Meka. 2018. Efficient Algorithms for Outlier-Robust Regression. *arXiv preprint arXiv:1803.03241* (2018).

[17] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. 2015. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575* (2015).

[18] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning.* https://arxiv.org/abs/1610.05492

[19] Kevin A Lai, Anup B Rao, and Santosh Vempala. 2016. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on.* IEEE, 665–674.

[20] Nancy A. Lynch. 1996. *Distributed Algorithms.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[21] Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin ChenâĂą. 2017. Implicit Regularization in Nonconvex Statistical Estimation: Gradient Descent Converges Linearly for Phase Retrieval, Matrix Completion and Blind Deconvolution. *arXiv preprint arXiv:1711.10467* (2017).

[22] Brendan McMahan and Daniel Ramage. 2017. Federated Learning: Collaborative Machine Learning without Centralized Training Data. https://research.googleblog.com/2017/04/federated-learning-collaborative.html. (April 2017). Accessed: 2017-04-06.

[23] Song Mei, Yu Bai, and Andrea Montanari. 2016. The landscape of empirical risk for non-convex losses. *arXiv preprint arXiv:1607.06534* (2016).

[24] Sahand Negahban and Martin J Wainwright. 2011. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research* 13, 1 (2011), 1665–1697.

[25] Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. 2018. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485* (2018).

[26] Maxim Raginsky. [n. d.]. ECE 543: Statistical Learning Theory Bruce Hajek. ([n. d.]).

[27] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms.* Cambridge university press.

[28] Alex Smola and SVN Vishwanathan. 2008. Introduction to machine learning. *Cambridge University, UK* 32 (2008), 34.

[29] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. 2018. Resilience: A Criterion for Learning in the Presence of Arbitrary Outliers. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018) (Leibniz International Proceedings in Informatics (LIPIcs))*, Anna R. Karlin (Ed.), Vol. 94. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 45:1–45:21. https://doi.org/10.4230/LIPIcs.ITCS.2018.45

[30] Lili Su. 2017. *Defending distributed systems against adversarial attacks: Consensus, consensus-based learning, and statistical learning*. Ph.D. Dissertation. University of Illinois at Urbana-Champaign.

[31] Lili Su and Nitin H. Vaidya. 2016. Fault-Tolerant Multi-Agent Optimization: Optimal Iterative Distributed Algorithms. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC '16)*. ACM, New York, NY, USA, 425–434. https://doi.org/10.1145/2933057.2933105

[32] T. Tao. 2012. *Topics in random matrix theory*. American Mathematical Society, Providence, RI, USA.

[33] Roman Vershynin. 2010. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027* (2010).

[34] Roman Vershynin. 2012. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability* 25, 3 (2012), 655–686.

[35] Roman Vershynin. 2018. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge university press.

[36] Martin Wainwright. 2015. Basic tail and concentration bounds. *URl: https://www. stat. berkeley. edu/.../Chap2_TailBounds_Jan22_2015. pdf (visited on 12/31/2017)* (2015).

[37] Yihong Wu. 2017. Lecture Notes on Information-theoretic Methods For High-dimensional Statistics. (April 2017). http://www.stat.yale.edu/ yw562/teaching/it-stats.pdf.

[38] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. *arXiv preprint arXiv:1803.01498* (2018).

[39] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. 2018. Defending Against Saddle Point Attack in Byzantine-Robust Distributed Learning. *arXiv preprint arXiv:1806.05358* (2018).

[40] Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. 2013. Communication-Efficient Algorithms for Statistical Optimization. *Journal of Machine Learning Research* 14 (2013), 3321–3363. http://jmlr.org/papers/v14/zhang13b.html

Received December 2018; revised January 2019; accepted February 2019